

# 統計解析ソフト R のスクリプト集

[Ver. 16.0]

本冊子では、データファイルとして CSV (コンマ区切り) (\*.csv) で保存した Shift JIS コードの csv ファイルを用いることを前提としており、`read.table` や `write.table` において、`fileEncoding = "shift-jis"` オプションを付けています。UTF-8 CSV ファイルを用いる場合は、このオプションを削除してください。

2025年4月

名古屋大学 大学院教育発達科学研究科・教育学部

心理発達科学専攻 計量心理学領域

石井研究室

## はじめに

この冊子は、心理学や教育学の研究でよく用いられる統計手法に関して、統計解析ソフト「R」のスクリプト（プログラム）をまとめたもので、授業の補助教材として作成されたものです。

Rは、スクリプトと言われるプログラムを書いてそれを実行する、という使い方をします。例えば、クロス表を表示するとしたら、スクリプト画面に `table(d1$x1, d1$x2)` と書き、この部分を選択して実行します。すると、出力画面に、実行したスクリプトとその結果が表示されます。

ここで、`table` がゴシック体になっていますが、これは、`table` がRの関数であることを表しています。スクリプト画面上でこのような表示がされるわけではありません。この冊子においては、Rの関数とそうでないもの（分析者が指定する変数名など）を区別するために、このような表示をしています。

本冊子では、重要と思われるコメントもゴシック体にしてあります。これは基本的に和文で、「#」を前に書いてコメントアウトしてあるので（Rは認識しない）、それがRの関数でないことはすぐわかると思います。

正しいスクリプトでないとソフトは動かず、エラーとなります。正しいスクリプトを書いて結果が表示されると、とても嬉しい気持ちになります。また、スクリプトを書くのに慣れてくるにつれ、分析法を考える力が増し、統計分析力が向上します。

本冊子はRの実行に主眼を置いているため、分析結果をどのように解釈するかについてはほとんど触れていません。統計分析の手法や結果の解釈については、別の成書にあたってください。参考までに、本冊子と関連の強い書籍を挙げておきます。1冊目には、本スクリプト集と同じデータを用いた分析例が出てきます。

人間科学のための統計分析—ここに安心があるすべての人のために— 医歯薬出版 2014  
<http://www.amazon.co.jp/dp/4263731611/>

統計分析のここが知りたい—保健・看護・心理・教育系研究のまとめ方— 文光堂 2005  
<http://www.amazon.co.jp/dp/4830644605/>

Essentials of Statistical Analysis and Psychometrics in Human Sciences.  
 See Sein Publishing:Myanmar, 2019 「人間科学のための統計分析」の英語版)

公認心理師カリキュラム準拠 臨床統計学 [心理学統計法・心理学研究法] 医歯薬出版 2020  
<https://www.amazon.co.jp/dp/4263266358/>



Rは開発環境が公開されているため、多くの研究者がボランティアで開発に関わっている、世界的に定評のあるフリーソフトです。新しい統計手法がすぐに取り入れられるという利点がありますが、同じ分析をするのでも開発者が複数いて、多数の関数が存在することがあったり、古い関数が使えなくなったりするという欠点もあります。ユーザー自身が使いやすい関数を選び、時折、知識を更新する必要があります。

Rは日々開発されていますので、この冊子の内容もすぐに古いものになってしまいます。ですので、こんな関数があるとか、ここの記述は間違ってる、これはもう使えないなど、気づいたことがありましたら、教えて頂ければ幸いです。

問い合わせ先 ((a)を@に変えて下さい)  
 ishii.lab.nuedu.psychometrics(a)gmail.com

石井研究室Webサイト  
<http://www.educa.nagoya-u.ac.jp/~ishii-h/>

石井 秀宗

## 目次

はじめに .....	2
<b>1 Rを使うために</b>	
Rの概要 .....	10
Rを使うための準備 .....	11
管理者権限について .....	12
Rのインストール .....	14
Rの起動と終了 .....	18
パッケージのインストール .....	19
パッケージの読み込み・取り外し .....	20
スクリプトの新規作成・保存・読み込み .....	21
スクリプトの実行 .....	23
オブジェクトの確認・消去 .....	24
ヘルプの使い方 .....	25
スクリプトの外部ファイルへのコピー .....	26
出力結果の外部保存 — Wordファイルなどに保存 .....	27
出力結果の外部保存 — sink関数 .....	28
図の外部保存 — Wordファイルなどに貼りつけ .....	30
図の外部保存 — PDFファイルに保存 .....	31
出力結果をExcelで整形保存 .....	32
Rの情報源 .....	35
Mac版のRで作業ディレクトリを設定 .....	36
Mac版のRでread.table関数でCSVファイルを読み込み .....	38
Mac版のRで図を描いたときに日本語を正しく表示させる .....	39
R Studioについて .....	40
Google ColaboratoryでRを使うための準備 .....	42
Google ColaboratoryでRを使う .....	44
Google ColaboratoryにCSVデータをアップロードする .....	45
Google Colaboratoryに日本語フォントをインストールする .....	46
Google Colaboratoryのノートブックを保存する .....	47
Google Colaboratoryのノートブックを共有する .....	48
<b>2 Rのスクリプト作成時の注意点とエラーへの対処</b>	
スクリプト作成にあたっての一般的注意 .....	50
スクリプトの冒頭に作業メモリを空にするコマンドを書く	
バックスラッシュと円マーク	
コメントを書くときは # を使う	
半角カナ文字を使わない	
全角文字もなるべく使わない	
スクリプトの各行の先頭に「>」「+」「1:」などを書かない	
データファイルの変数名とRスクリプトの変数名を一致させる	
Rの予約語をオブジェクト名に用いない	
pi, T, Fなどをオブジェクト名に用いない	
代入結果を表示するときは当該のスクリプト部分を () で囲う	
改行の位置に気をつける	
欠測のあるデータの分析について	
四捨五入について	
作成したスクリプトを少しずつ実行して確認する	
よくある質問・エラー .....	56
データが読み込めない (rm, setwd, read.table)	
添え字が許される範囲外です と表示される	
作成した覚えのない変数が存在する	
incomplete final line found by readTableHeader と表示される	
結果の数値に e-01 のような表記が付く	
パッケージが開けない	
パッケージの関数が使えない	
エラーが発生したときの対処 .....	59
半角カナ文字を使っていませんか？	
お使いのPCのOSは何ですか？	
Excelでデータを保存したCSVデータと、Rでの読み込み形式は合っていますか？	
作業メモリ上に余計なデータが保存されていませんか？	

実行するスクリプトの範囲選択は正しいですか？  
 データのあるフォルダを正しく設定していますか？  
 データファイル名は正しいですか？  
 変数名を間違えていませんか？  
 データファイルの変数名にスペースが入っていませんか？  
 全角記号が混ざっていませんか？  
 スペルミスをしていませんか？  
 スクリプトを書くときに「>」「+」「1:」などを左端に書いていませんか？  
 大文字と小文字を間違えていませんか？  
 記号を間違えていませんか？  
 カンマを忘れていませんか？  
 カンマを多く入れていませんか？  
 カンマとピリオドを間違えていませんか？  
 文字列をダブルクォーテーションで囲うのを忘れていませんか？  
 カッコをつけ忘れていませんか？  
 カッコの種類を間違えていませんか？  
 カッコで括らなければならないのを忘れていませんか？  
 カッコをつける位置を間違えていませんか？  
 オプションは正しく書かれていますか？  
 if - else文を書くとき、if 文の終わりと else 文の始まりの間に改行してませんか？  
 1行に複数の命令文を書いていませんか？  
 1つの数式を複数行に分けて書くとき、次の行の先頭が+ - \* / などの記号になっていませんか？  
 パッケージは正しくインストールされていますか？  
 R Studioを使っていますか？  
 Rが複数個、起動していませんか？  
 コンピュータが再起動待ちの状態になっていませんか？  
 全角文字を使っていることが原因のこともあります  
 パソコンのユーザー名が全角文字になっていることが原因のこともあります  
 Rやコンピュータを再起動することも有効です  
 Rを再インストールすることも有効です

### 3 データの作成・読み込み・保存

Excelにおけるデータファイルの作成 .....	66
Excelを使わずにCSVファイルを作成 .....	67
ExcelにおけるデータファイルのCSV保存 .....	68
作業ディレクトリの指定 .....	69
データの読み込み .....	70
読み込んだデータの確認 .....	72
R上のデータのCSV保存 .....	73
UTF-8 CSVファイルをExcelで読み込む方法 .....	75
Excelデータファイルの読み込み・保存 .....	76

### 4 データ操作

欠測値の取り扱い .....	80
データ値に基づいたカテゴリ変数の生成 — 離散データ .....	84
データ値に基づいたカテゴリ変数の生成 — 連続データ .....	85
カテゴリ変数の再カテゴリ化 — 水準の合併 .....	86
不要な水準の削除・必要な水準の追加 .....	87
回答データの採点 .....	88
逆転項目のデータ処理 .....	90
合計得点・平均得点の計算 .....	91
データの標準化・中心化・偏差値化 .....	92
行名・列名（変数名）・要素名の指定 .....	94
行や列の抽出・削除 .....	96
データの並べ替え .....	100
データの結合 .....	101
ロングフォーマットデータ・ワイドフォーマットデータ — 対応のあるデータのフォーマット変換 .....	104
データ型の確認・変換 .....	109
データ構造の確認・変換 .....	110
文字型の数字を数値型の数値に変換 .....	111
factor型のデータ値の扱い .....	113

### 5 統計図表

度数分布表 — 質的変数 .....	116
度数分布表 — 量的変数 .....	117



クロス集計表	118
割合なども入ったクロス集計表	120
群別のクロス集計表	123
フラットなクロス集計表	1124
フラットなクロス集計表から通常のクロス集計表を作成	127
円グラフ	129
帯グラフ	130
棒グラフ	133
ヒストグラム	136
2群の積み上げヒストグラム	138
箱ひげ図	139
折れ線グラフ	140
散布図	141
ひまわり図	142
バブルプロット	143
ラベルつき散布図	144
グラフオプション	145
ggplotを使った作図	148
度数を比較する棒グラフ — ggplot	150
平均値を比較する棒グラフ — ggplot	151
帯グラフ — ggplot	152
ヒストグラム — ggplot	153
箱ひげ図 — ggplot	154
折れ線グラフ — ggplot	155
散布図 — ggplot	156

## 6 量的変数の要約統計量

平均, 標準偏差, 最大値, 中央値, 最小値, 分位数	158
群別の平均, 標準偏差, 最大値, 中央値, 最小値, 分位数	159
要約統計量 — describe, describeBy	161
2要因以上の群別要約統計量	163
共分散・相関係数	164
群別の共分散・相関係数	165
偏相関係数	167
四分相関係数・多分相関係数	169
アルファ係数	171
オメガ係数	173
級内相関係数	174

## 7 1群・2群の平均値に関する推測 — t 検定 —

1群の平均値の推測	176
対応のある2群の平均値の比較	177
対応のある2群の平均値差の効果量の推定	178
対応のない2群の平均値の比較 — 素データ	179
対応のない2群の平均値の比較 — 要約統計量	180
対応のない2群の平均値差の効果量の推定	181
平均値の非劣性・同等性の検証 — 素データ	184
平均値の非劣性・同等性の検証 — 要約統計量	186

## 8 分散に関する推測

2群の分散の比較 — 素データ	188
2群の分散の比較 — 要約統計量	189
多群の分散の比較 — 素データ	190
多群の分散の比較 — 要約統計量	191

## 9 多群の平均値に関する推測 — 分散分析・多重比較 —

1つの被験者間要因(1 B)	193
1つの被験者内要因(1 W)	200
2つの被験者間要因(2 B)	204
1つの被験者間要因と1つの被験者内要因(1 B 1 W)	213
2つの被験者内要因(2 W)	219
3つの被験者間要因(3 B)	224
2つの被験者間要因と1つの被験者内要因(2 B 1 W)	235

1つの被験者間要因と2つの被験者内要因(1 B 2 W) .....	241
3つの被験者内要因(3 W) .....	248
<b>1 0 分布の位置に関する推測 — ノンパラメトリック法 —</b>	
対応のある2群の分布の位置の比較 — ウィルコクソンの符号順位検定 .....	256
対応のある多群の分布の位置の比較 — フリッドマンの検定 .....	258
対応のない2群の分布の位置の比較 — ウィルコクソンの順位和検定(マン・ホイットニーの検定) ..	260
対応のない多群の分布の位置の比較 — クラスカル・ウォリスの検定 .....	262
<b>1 1 相関係数に関する推測</b>	
相関係数の推測 .....	265
相関係数行列の各要素の検定 .....	266
2群の相関係数の差の検定 .....	267
2群の相関係数の差の信頼区間 .....	268
多群の相関係数の等質性の推測 .....	270
<b>1 2 分割表に関する推測 — <math>\chi^2</math> 検定・フィッシャーの正確検定 —</b>	
2×2表 — ファイ係数, カイ2乗検定, 尤度比検定, フィッシャーの正確検定 .....	274
r×c表 — クラメル の 連関係数, カイ2乗検定, 尤度比検定 .....	277
ピアソン残差・デビアンズ残差 .....	280
2名の評定者の評定の一致度 — Cohenのカップパ係数 .....	283
複数名の評定者の評定の一致度 — Fleissのカップパ係数 .....	285
<b>1 3 比率に関する推測</b>	
1群の比率の推測 — 二項検定 .....	287
対応のある2群の比率の比較 — マクネマーの検定 .....	288
対応のある多群の比率の比較 — コ克蘭のQ検定 .....	290
リスク差・リスク比・オッズ比の推測 .....	292
対応のない多群の比率の等質性の推測 .....	297
<b>1 4 回帰分析</b>	
単回帰分析 — lm関数 .....	300
重回帰分析 — lm関数 .....	302
重回帰分析 — glm関数 .....	305
重回帰分析 — sem関数 .....	307
<b>1 5 パス解析</b>	
パス解析 — specifyEquations関数 .....	313
パス解析 — specifyModel関数 .....	317
多母集団のパス解析 .....	322
多母集団パス解析におけるパス係数の検定 .....	327
<b>1 6 因子分析</b>	
確認的因子分析 .....	334
2次因子分析 .....	340
階層因子分析 .....	345
探索的因子分析 .....	351
<b>1 7 潜在変数のある構造方程式モデリング(共分散構造分析)</b>	
多重指標モデル .....	355
潜在変数のある構造方程式モデリング — semパッケージ .....	360
潜在変数のある構造方程式モデリング — lavaanパッケージ .....	366
成長曲線モデル — lavaan パッケージ .....	370
多母集団同時分析 — semパッケージ .....	373
多母集団同時分析 — lavaanパッケージ .....	382

等値制約 — semパッケージ	387
等値制約 — lavaanパッケージ	395

## 18 さまざまな多変量データ解析

クラスター分析	400
主成分分析	404
主成分得点を用いた重回帰分析	406
主成分回帰分析・PLS回帰分析	409
線形判別分析	412
数量化Ⅰ類	414
数量化Ⅱ類	417
数量化Ⅲ類	419
対応分析	421
多重対応分析	424
ロジスティック回帰分析 — 素データ	428
ロジスティック回帰分析 — 集計データ	431
多項ロジスティック回帰分析	434
対数線形モデル — glm関数	437
対数線形モデル — loglm関数	444
カウントデータの分析	451

## 19 階層線形モデル（マルチレベルモデル）

階層線形モデルの例	454
Rで階層線形モデル分析をするためのデータ構造	455
階層線形モデル — lmer関数	456
階層線形モデル — lme関数	463

## 20 検定力分析

検定力分析 — 2群の平均値	470
検定力分析 — 相関係数	472
検定力分析 — クロス集計表	474
検定力分析 — 2群の比率	476
検定力分析 — 1つの被験者間要因(1 B)	479
検定力分析 — 1つの被験者内要因(1 W)	482
検定力分析 — 2つの被験者間要因(2 B)	485
検定力分析 — 1つの被験者間要因と1つの被験者内要因(1 B 1 W)	489
検定力分析 — 2つの被験者内要因(2 W)	494

## 21 信頼区間に基づく標本サイズの推定

信頼区間に基づく標本サイズの推定 — 1群の平均値	500
信頼区間に基づく標本サイズの推定 — 2群の平均値	501
信頼区間に基づく標本サイズの推定 — 相関係数	502
信頼区間に基づく標本サイズの推定 — 比率	503

## 22 項目分析

項目分析 — 多肢選択式解答データ	505
項目分析 — 解答類型データ	515
項目分析 — itemanalysisパッケージ	525

## 23 項目応答理論

項目パラメタの推定 — 1PL, 2PL, 3PLモデル	530
特性関数・情報関数の図示	534
特性値の推定 — 項目パラメタを推定する場合	537
特性値の推定 — 項目パラメタが既知の場合	540
局所独立性の確認	543
等化 — 共通項目法	546
D I F分析	552

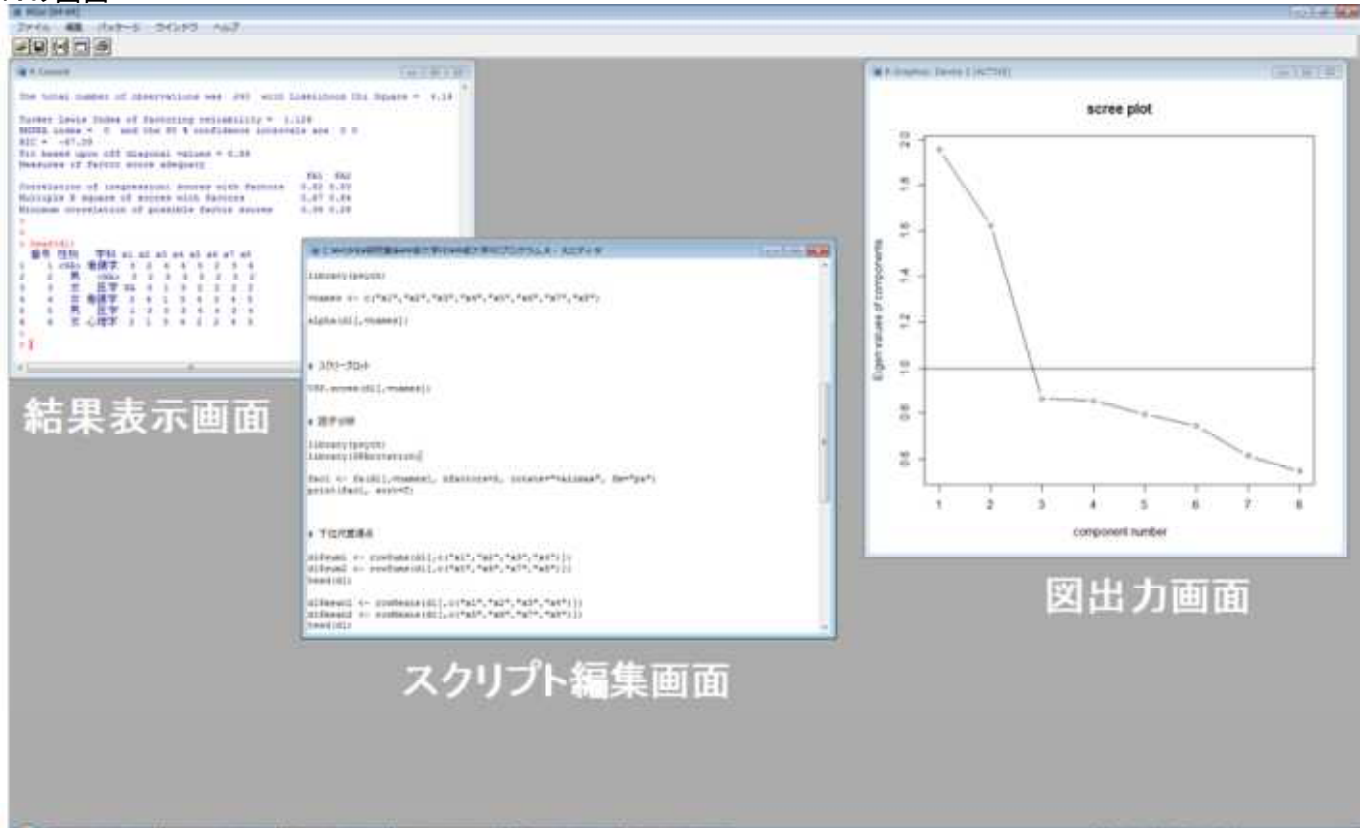
## 2 4 付録

文字列関数 — 文字列の切り出し・結合・検索・置換 .....	559
演算記号・算術関数 .....	562
切り上げ・切り下げ・四捨五入 .....	564
集合関数 .....	566
比較演算子 .....	567
論理演算子 .....	568
ベクトルの作成 .....	569
行列の作成 .....	570
ベクトル・行列の要素の評価 .....	572
行列演算 .....	574
確率関数 .....	577
多変量正規乱数の発生 .....	578
項目反応モデルに基づく0/1データの発生 .....	580
制御コマンド .....	583
コマンドを生成して実行 .....	586
一括分析・総当たりの分析 .....	589
関数の作成 .....	592
スクリプトの呼び出し .....	593
いくつかの図をまとめてPDFファイルに出力 .....	594
正規分布と標準得点の対応関係の図を描く .....	596
統計的有意性検定の概念図を描く .....	600

# 1 Rを使うために

## Rの概要

## Rの画面



## どういうものか

- 統計分析や統計図表の作成ができる
- SPSS や EXCELより, SAS や FORTRAN に近い
- スクリプト (プログラム) を書く
- 本体とパッケージからなる
  - 本体 データ管理, 編集, 領域普遍的な分析
  - パッケージ 領域固有的な分析, 便利機能

## デメリット

- 取っ付きが悪い
- 関数名や文法を知らないと使えない
- 正しくスクリプト (プログラム) を書かないと動かない
- パッケージが山のようにあって, どれを使えば良いのか調べるのが大変
- 同じ分析を実行するパッケージや関数が複数あったりする
  - 分散分析 aov, anova, Anova
  - 共分散構造分析 sem, lavaan
- 異なるパッケージの中に, 同じ名前の関数があつたりする
- パッケージ, 関数によって, 仮定や数値計算法の違いなどから, 結果が多少異なることがある

## メリット

- フリーソフト
- 世界標準
- 世界中のボランティア開発者が, よってたかってパッケージを開発
- 新しい分析機能がどんどん追加されていく
- 信頼性が高い
- 分析のスクリプト (プログラム) を保存
  - 同じ分析を簡単に再現できる
  - SPSSは分析結果を保存し, 過程を保存しないので, 過去の分析を再現するのが難しいことがある
  - Rは, 分析結果ではなく, スクリプト (分析方法, 過程) を保存するので, 結果を容易に再現できる

## Rを使うための準備

### Rを使うにあたって必要なもの

- ・インターネットにつながるパソコン  
Rをダウンロード・実行するのに必要  
Windows, Mac, Linux いずれでもよい  
(本スクリプト集は基本的にWindows環境で作成しています)
- ・エクセルなどデータ入力できるソフト  
データをCSV形式で保存するのに使う  
他の形式(エクセル形式など)でも構わないのだが、汎用性の高いCSV形式を推奨  
表入力・CSV保存できるソフトなら何でも構わない
- \* Google Colaboratory を使って、インターネット上で R を使うこともできます。その場合は、Google アカウントが必要です。  
Google Colaboratory でRを実行する場合は、OSの違いはあまり意識しないで済みますが、インターネットへの常時接続が必要である、CSVデータを毎回アップロードしなければならない、利用時間に制約があるなど、いくつか不便な点もあります。

### Rを使うにあたって必要なこと

- ・所定のWebサイトからRファイルをダウンロードしてパソコンにRをインストールする  
フリーソフトなので料金はかからない  
安全なソフトとされている
- ・エクセル等でデータを入力しCSV形式で保存する  
データのある場所が、Rの作業領域になるので、処理の速さが気にならないければUSBなどの外部メモリに保存しておくで良い。(バックアップは別にとっておく)
- ・Rで分析を行うための、スクリプト(プログラム)の書き方、関数名を覚える  
データの読み込み、書き出しのためのスクリプト  
分析のためのスクリプト  
データ加工のためのスクリプト など  
おそらくこの部分が一番のハードルになる

### ファイルの拡張子を表示させる

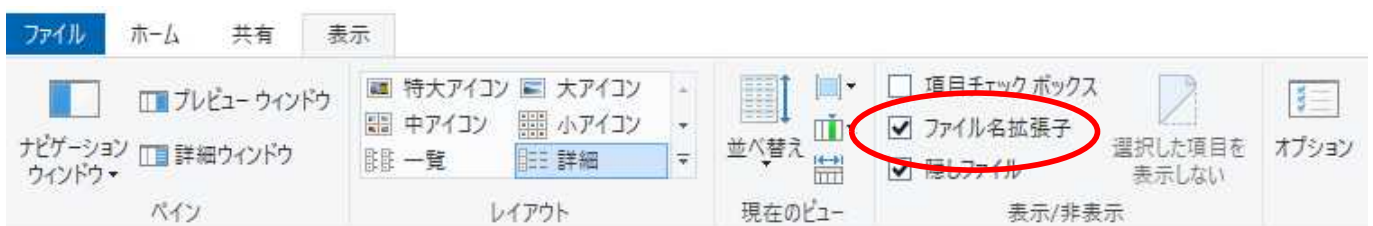
たいていの場合、コンピュータ上のファイル名には、拡張子というものがついています。～.pdf, ～.docxの.pdfや.docxなどです。ファイルの属性を表すのに用いられています。

Windowsのエクスプローラは、初期設定では拡張子を表示しません。この状態だと、～の部分と同じファイル名のファイルが、PDFファイルなのかWordファイルなのか、表示されるファイル名だけでは分からなくなります。

そこで、エクスプローラにおいて、拡張子も表示させるように設定を変更しておきます。

エクスプローラ → 表示 → ファイル名拡張子をクリック としてください。そうすると、拡張子がついたファイル名が表示されます。

ファイルの種類を確認するためにも、拡張子を常に表示させることをお勧めします。





## 管理者権限について

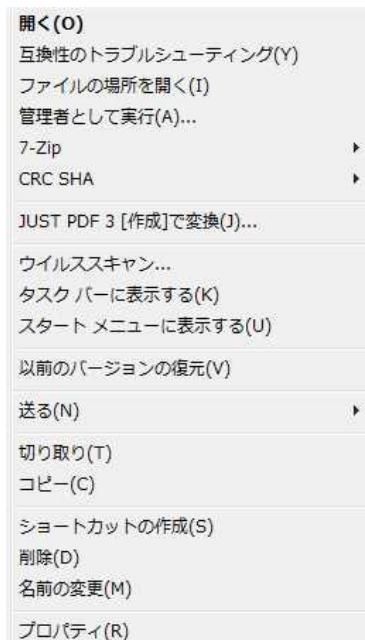
Windowsマシンの場合、管理者としてパソコンを実行していないとRをインストールできないことがある。このような場合は、パソコンの管理者にRをインストールしてもらうか、自分にも管理者権限を持たせてもらう必要がある。Macの場合は、管理者権限がないとRを実行できないので、管理者権限を意識することはまずない。

管理者権限を持っていなくてもRを使うことはできる。しかし、Rを管理者権限として実行できていないと、パッケージをインストールした際、Rのライブラリフォルダに保存されず、別のフォルダにインストールされる場合がある。例えば、パッケージのインストールを行った際、以下のような警告メッセージが出ることもある。

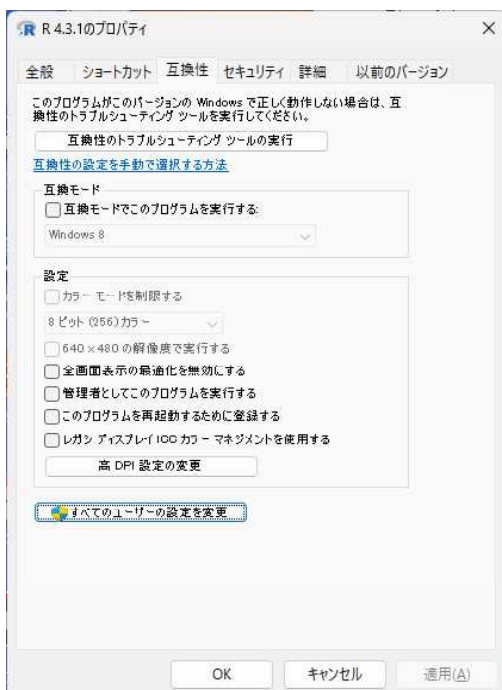
```
Warning in install.packages(NULL, .libPaths()[1L], dependencies = NA, type = type) :
'lib = "C:/Program Files/R/R-3.2.0/library"' は書き込み可能ではありません
ダウンロードされたパッケージは、以下にあります
C:\Users\¥○○○¥AppData¥Local¥Temp¥RtmpWi0uAi¥downloaded_packages
```

このような状況では、パッケージはR本体とは別のフォルダにインストールされているため、library()関数で読み込もうとしても、「○○というパッケージはありません」というエラーが出る。

上のような状況を解決するためには、管理者権限のあるユーザーとしてRを実行して、パッケージをインストールする。Rを管理者権限として実行するためには、Rのプロパティを変更する。



デスクトップ上のRのアイコン上で右クリックし、左のようなメニューを表示し、メニューの中から、「プロパティ」を選択する。



「プロパティ」を選択すると、左のようなウィンドウが表示される。

プロパティウィンドウにある複数のタブの中から、「互換性」タブを選択する。

「すべてのユーザーの設定を変更」のボタンをクリックする。

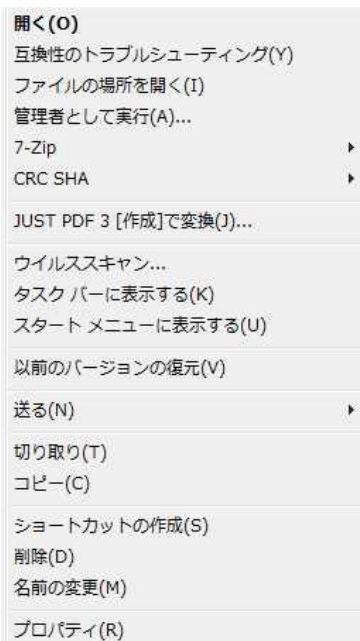


「管理者としてこのプログラムを実行する」のチェックボックスにチェックを入れ、「適用」を選択する。

表示させたすべてのウィンドウについて「OK」を選択して閉じる。

次にRを起動したとき、「コンピュータへの変更を許可しますか？」と聞かれたら「はい」として続行する。

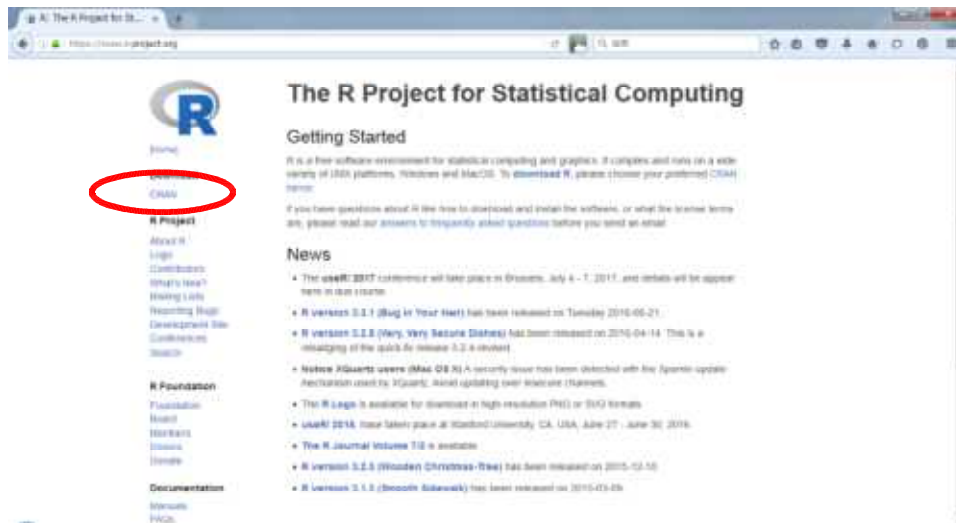
これで、管理者としての実行が可能となるので、この状態で、パッケージのインストールをすればよい。



インストール後、Rを起動するとき、デスクトップ上のRのアイコンをクリックして起動しない場合は、Rのアイコン上で右クリックし、「管理者として実行」を選択する。

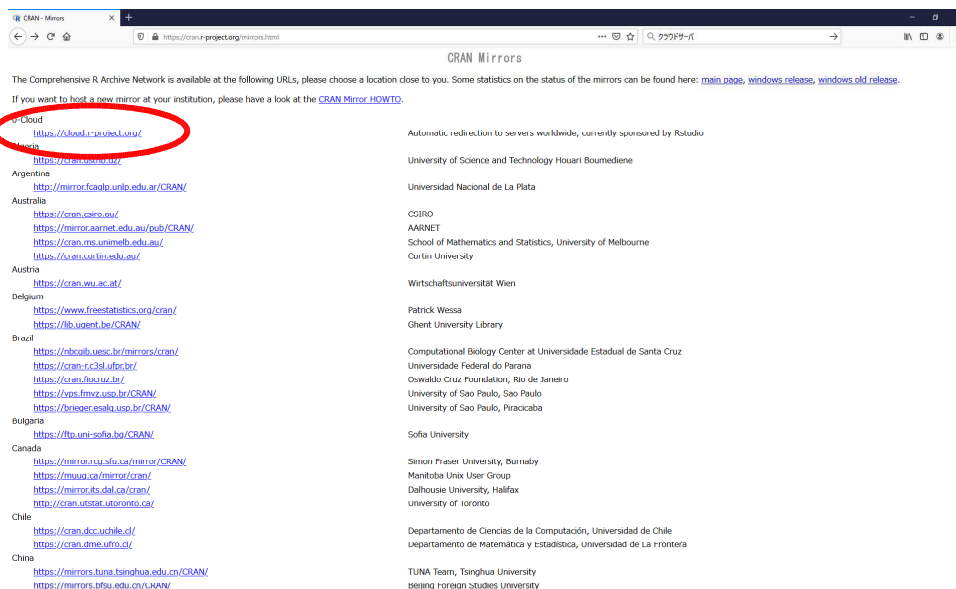
## Rのインストール

【注意】 管理者または管理者権限を持ったユーザーとしてパソコンを起動（ログイン）している必要がある。  
管理者権限については前項参照。

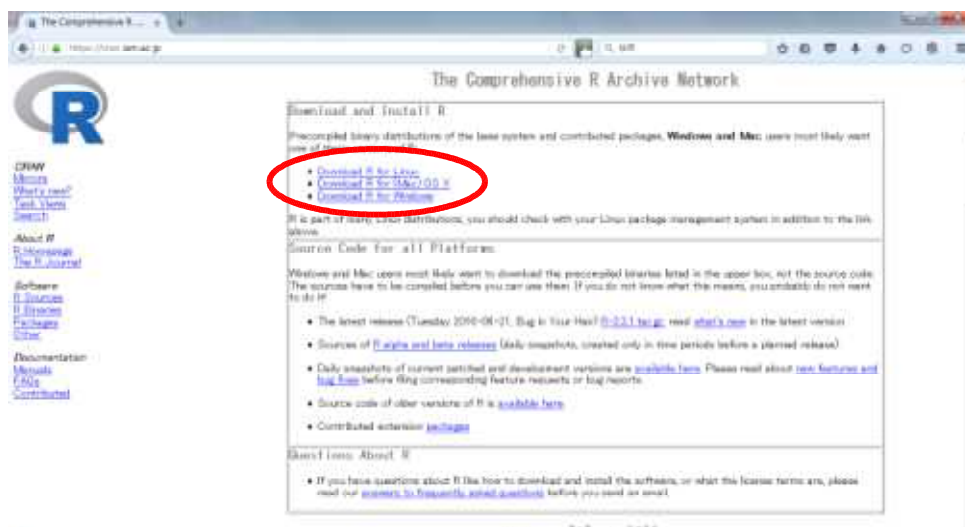


<http://www.r-project.org/>  
を開く。

Downloadの下にある CRAN を  
クリックして、最寄りのミラ  
ーサイトを選ぶ。



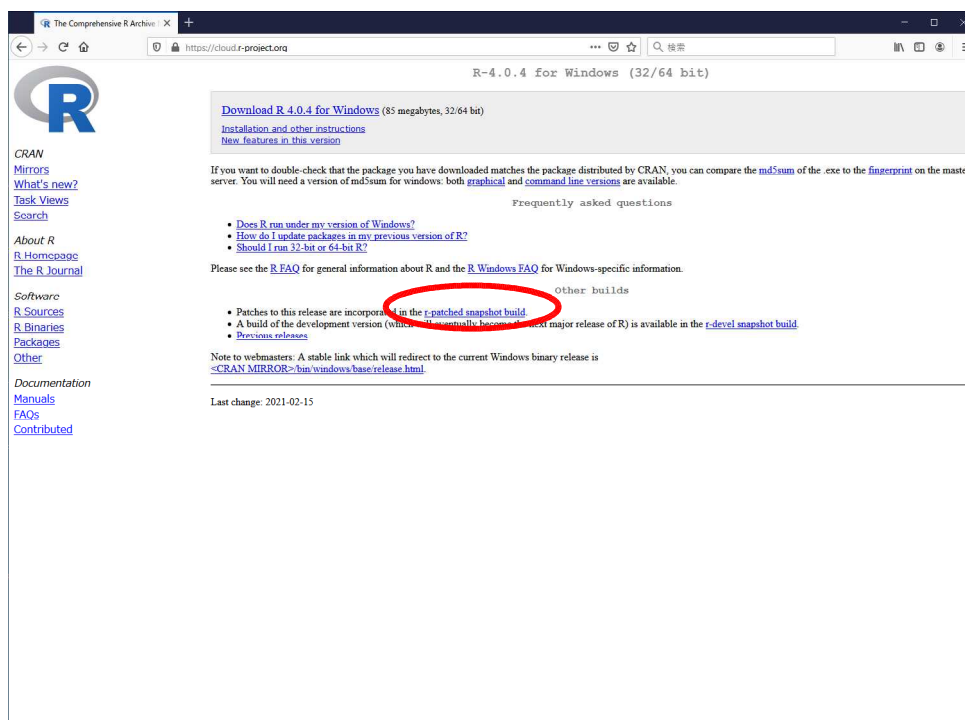
一番上の 0-cloud, もしくは,  
Japan の The Institute of S  
tatistical Mathematics, Tok  
yo からダウンロードするのが  
分かりやすい。



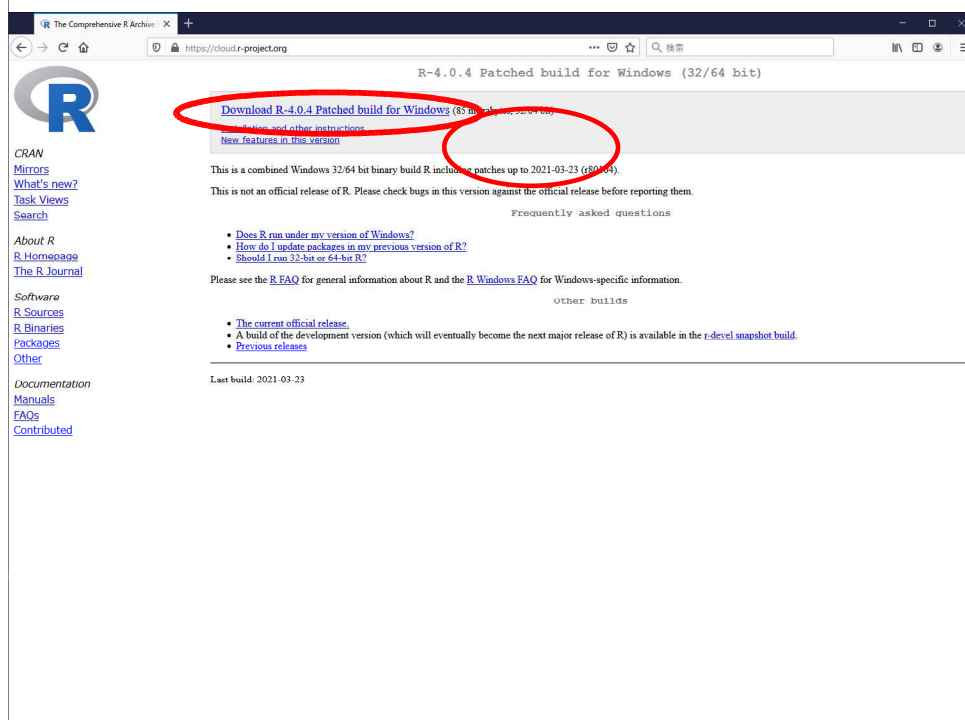
Download and Instal R で,  
Mac, Windows, Linux のうち,  
自分の使っているコンピュ  
ータにあうものを選ぶ。



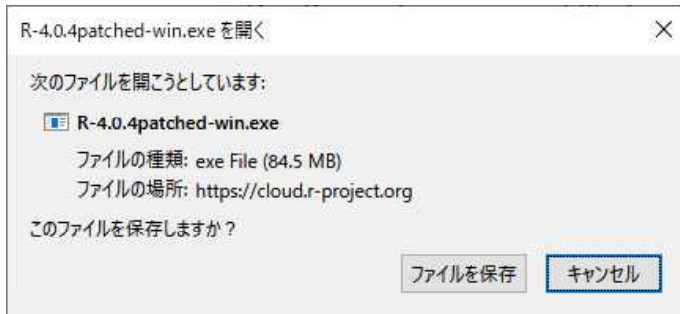
baseを選択する。



当初リリースの不具合を修正したPatched buildがあればクリックする。



一番上のDownload R... を選ぶ。



ファイルを保存するか聞いてくるので、保存するを選ぶ。

ダウンロードしたファイルが、「ダウンロード」というフォルダか、「デスクトップ」か、どこかにあるはずなので、探し出す。

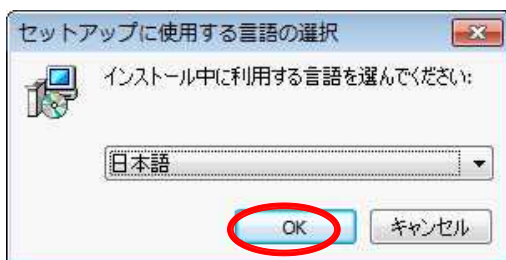
デスクトップに保存した場合



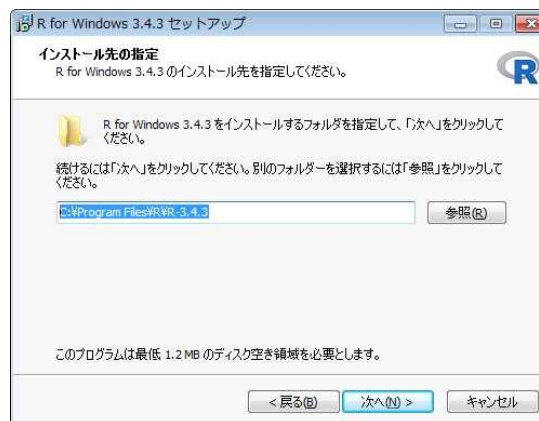
ダウンロードフォルダ等に保存した場合



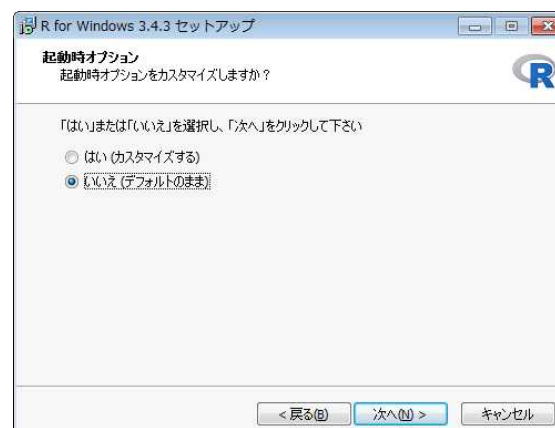
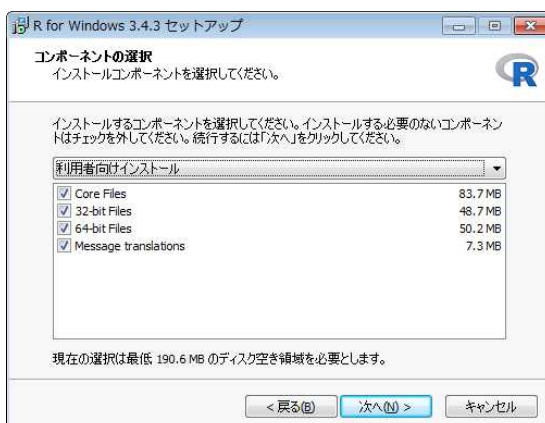
ダウンロードしたファイル「R-〇〇.exe」をダブルクリックして、インストールを開始する。〇〇のところは、RのバージョンやOSなどにより異なる。



利用言語を選びOKとする

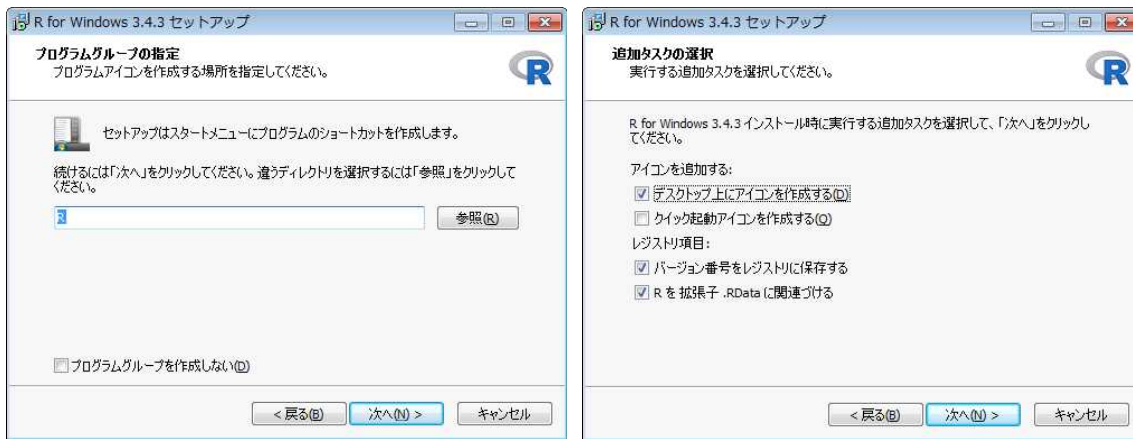


特に構わなければ  
次へ 次へ 次へ  
...

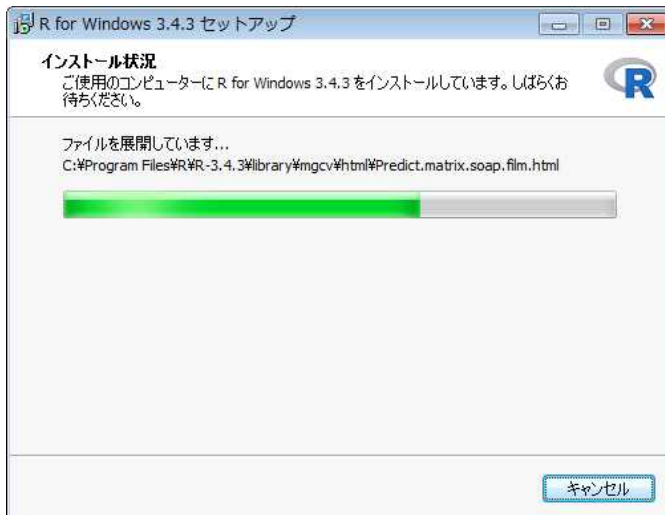


32bitマシンか 64bit マシンか選択できるときは、自分のあてはまるほうを選ぶ。(あてはまらないほうのチェックを外す)

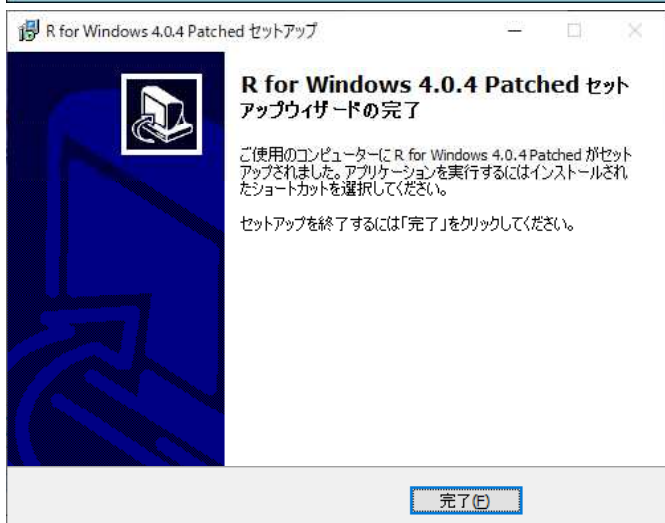




「デスクトップ上にアイコンを作成する」にチェックを入れる



インストール中の画面



インストール完了の画面

インストールができれば、画面上にこんなアイコンができています。



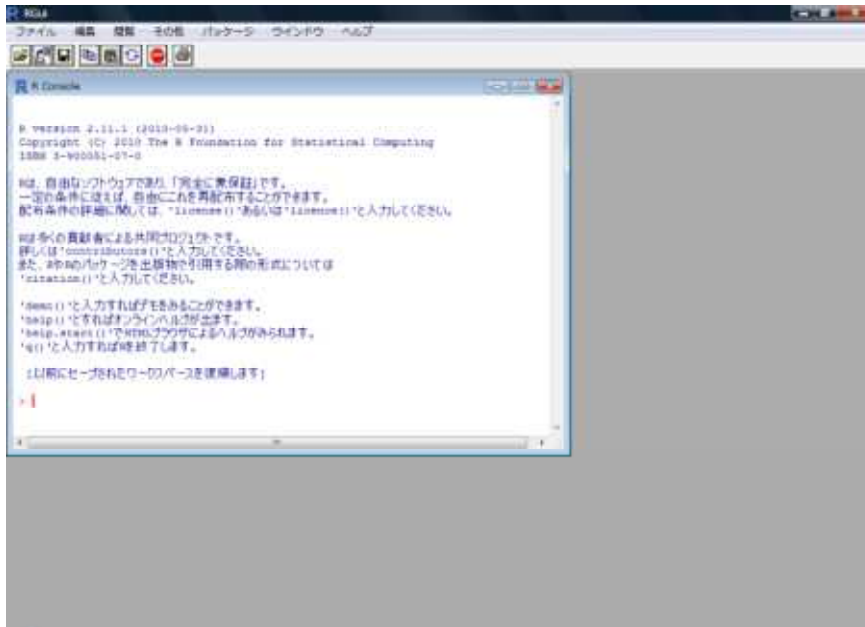
以上でRのダウンロードは終了。

## Rの起動と終了



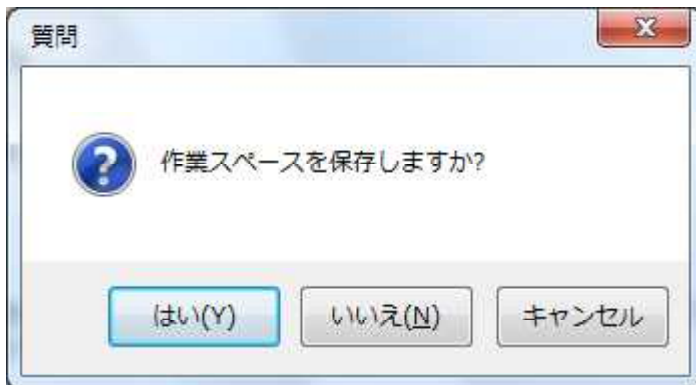
画面上にある左にあるようなアイコンをクリックすると、Rが起動する。

アイコンがない場合には、スタートメニューから起動する。



Rが起動すると左図の画面が開き、Rが起動する。

右上の×を推すと、Rが閉じる。



Rを閉じるとき、作業内容を保存するかどうか聞いてくるので、「はい」か「いいえ」かを答える。

「はい」を選択すると「.RData」という拡張子のついたファイルが作成される。そのファイルを開くとRが起動し、そのファイルを作成した段階のメモリ状態となる（スクリプトは出てこない）。

とくに必要なければ、ここは「いいえ」が良い。



最新のスクリプトが保存されていないときは、最新のスクリプトを保存するか聞いてくるので、「はい」か「いいえ」かを答える。「はい」とした場合には、スクリプトの保存が行われる。

その後、改めてRを終了する。



## パッケージのインストール

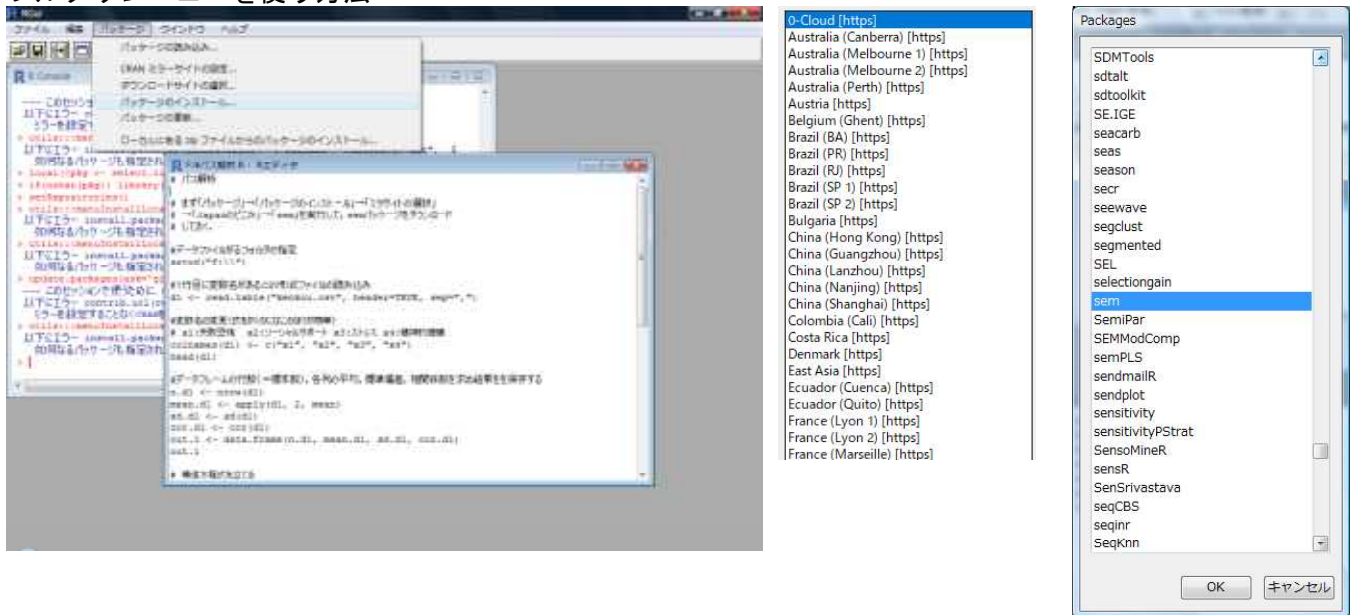
【注意】 管理者または管理者権限を持ったユーザーとしてパソコンを起動し、管理者としてRを起動している必要がある。

Rには、本体の他にパッケージというものがあり、パッケージに含まれる関数を使うためには、まずパッケージのインストールをする必要がある。ただし、パッケージにも、MASSのようにRをインストールしたときに同時にコンピュータにインストールされるものと、psychやsemのようにRをインストールしたときにはコンピュータにインストールされないものがある。以下は、後者のパッケージのインストールの説明である。

パッケージのインストールは、原則として当該コンピュータで一度だけやればよいが、Rのバージョンアップをしたり、システムの復元をしたりすると、過去のパッケージがインストールされていない状態に戻ることがあるので、その場合は改めてインストールする。

パッケージに含まれる各コマンドについては、Rをインストールしたフォルダの中にそのパッケージ名のフォルダができていれば、その中にパッケージのマニュアルがあるので参照するとよい。

## プルダウンメニューを使う方法



- (1) 「パッケージ」→「パッケージのインストール」とする。
- (2) ミラーサイトを設定する画面が出てくるので、0-cloud, もしくは, JapanのTokyoなど最寄りのミラーサイトを選択する。
- (3) パッケージのリストが出てくるので、必要なパッケージ (psych や sem など) を選ぶ。

## Rのコマンドを使う方法

- (1) Rのコンソール画面上に以下を入力し実行する  
`install.packages(c("パッケージ名1", "パッケージ名2", ...), dependencies = TRUE)`

パッケージ名のところは, psych や sem など, 自分が必要とするパッケージの名前を書く。  
 c()の中に複数のパッケージ名を書くと, 複数のパッケージを一括してインストールすることができる。  
 dependencies = TRUE と指定しておくと, 依存するパッケージもインストールしてくれる

- (2) ミラーサイトを設定する画面が出てくるので, 0-cloud, もしくは, JapanのTokyoなど最寄りのミラーサイトを選択する。

## 本スクリプト集で使用するパッケージの一括インストール

```
install.packages(c("agricolae", "ca", "car", "coefficientalpha", "compute.es", "corpcor", "CVST",
  "descr", "DescTools", "difR", "effectsiz", "epitools", "FactoMineR", "fdth", "ggplot2", "gmodels",
  "GPArotation", "irr", "irtoys", "itemanalysis", "lavaan", "lme4", "lsr", "ltm", "MBESS", "nlme",
  "openxlsx", "phia", "pls", "polycor", "psych", "pwr", "sem", "vcd", "VGAM"),
  dependencies=TRUE)
```

## パッケージの読み込み・取り外し

パッケージは、Rをインストールしたときに同時にインストールされるものでも、後からインストールしたものでも、読み込みをしてからでないと使用できない。

インストールはできたはずなのに、パッケージを読み込めない場合は、インストールがうまくできていない可能性がある。その場合は、管理者としてRを起動させて、パッケージをインストールするとうまく行くことが多い（次項参照）。

### 読み込み

`library(パッケージ名)`

上記スクリプトを書いて実行すると、そのパッケージが読み込まれる。

パッケージ名を"で囲わない。

読み込み後なら、ヘルプを使って、そのパッケージに含まれる関数の使い方を読むことができる。

### 取り外し

`detach("package:パッケージ名")`

異なるパッケージが同一の関数名を使用しているときに、その2つのパッケージを読み込むと、どちらか一方のパッケージの関数が使えなくなる。これを避けるために、不要なほうのパッケージを取り外す。

一度取り外したパッケージでも、`library()` でまた読み込むことができる（再インストールする必要はない）。

```
# パッケージの読み込み・取り外し

# sem パッケージの読み込み
library(sem)

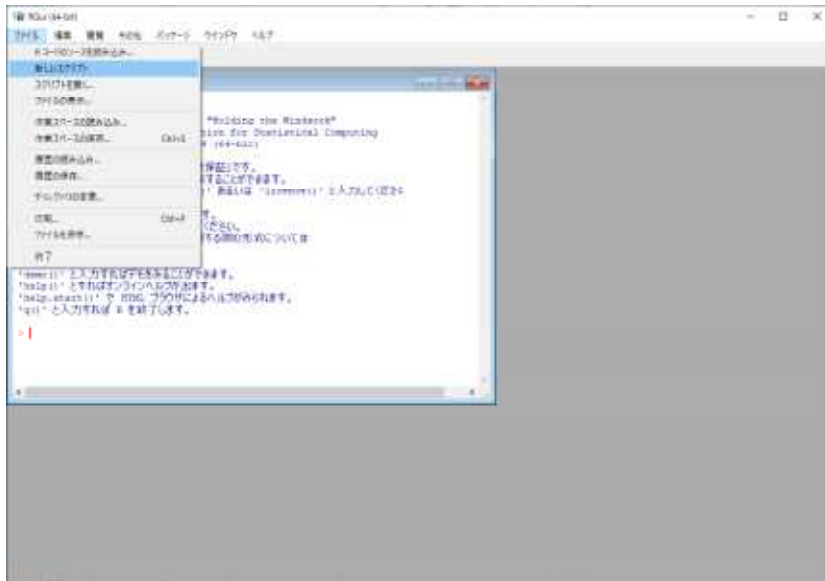
# lavaan パッケージの読み込み
library(lavaan)

# どちらにも sem というコマンドがあるので、
# このあと sem パッケージのほうを使うためには、
# lavaan パッケージを取り外す。
detach("package:lavaan")
```

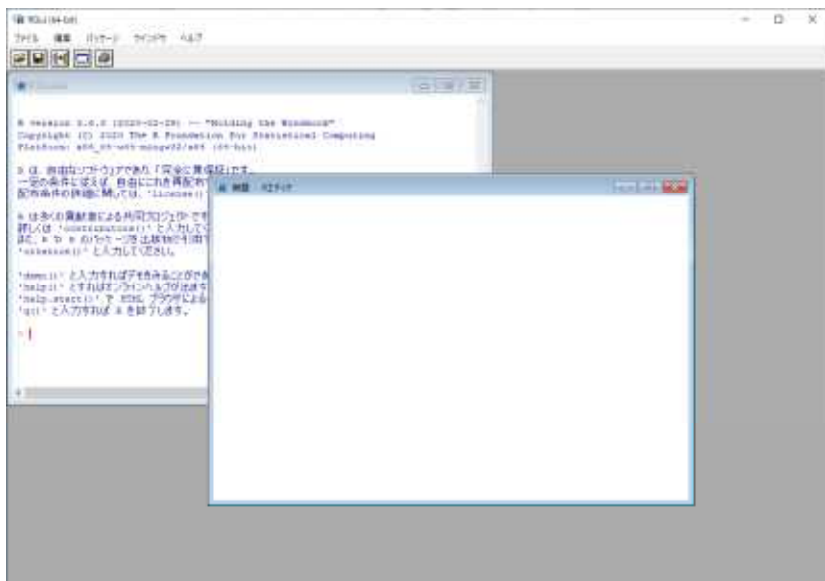
## スクリプトの新規作成・保存・読み込み

Rは、スクリプトと言われるプログラムを書いてそれを実行する、という使い方をします。  
スクリプトは、Rエディタで開き、実行する範囲を指定して実行する。  
実行結果は、コンソール（Console）画面や、グラフィクス（Graphics）画面に出力される。

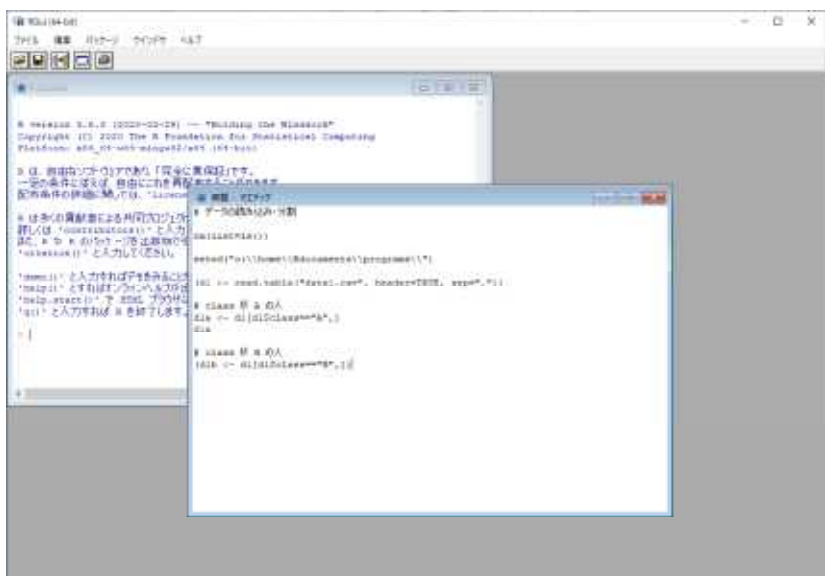
### 新規作成



スクリプトを最初を書くときは、Rを起動し、「ファイル」→「新しいスクリプト」を選択する。

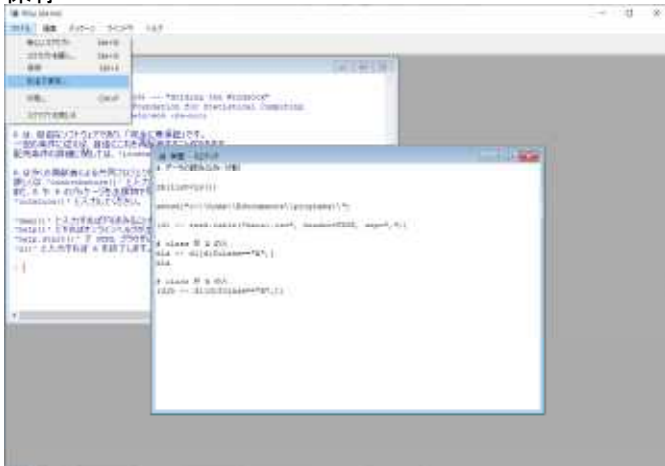


スクリプトを書く画面（Rエディタ）が表示されるので、ここにスクリプト（プログラム）を書く。

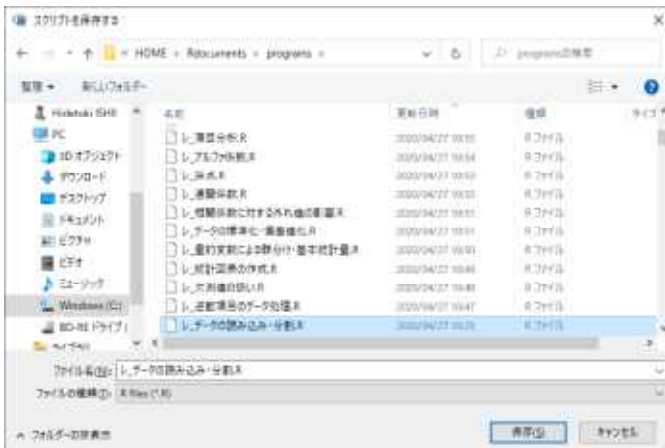


スクリプトを書いたところ

## 保存



スクリプト（プログラム）を書いたら「ファイル」→「別名で保存」とする。

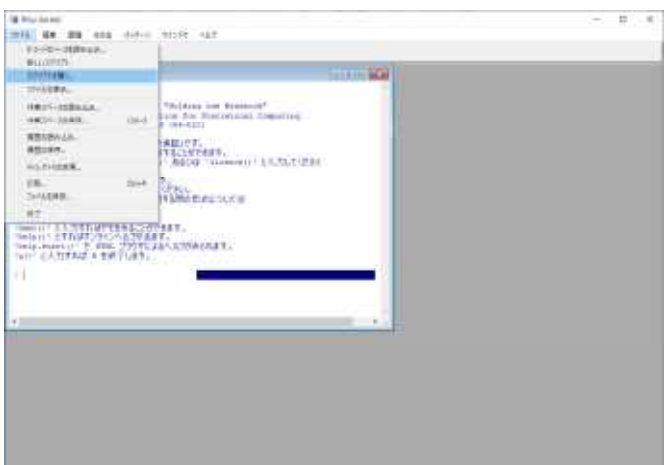


保存するフォルダを設定する画面が開くので、ドライブ、フォルダを選択移動し、ファイル名のところに「ファイル名」を入力する。

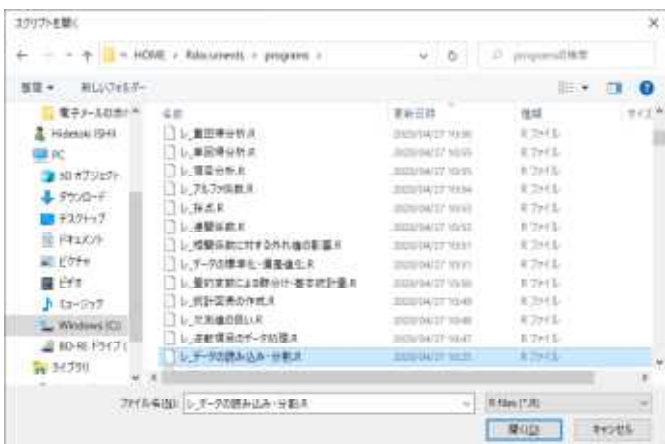
ファイル名は、変な記号を使わなければ、基本的に何でも良い。

保存すると、自動的に「.R」という拡張子がつく。

## 読み込み



スクリプトを開いたあとや、Rを終了した後などに、保存したスクリプトを再び開くときは、Rを起動し、「ファイル」→「スクリプトを開く」とする。

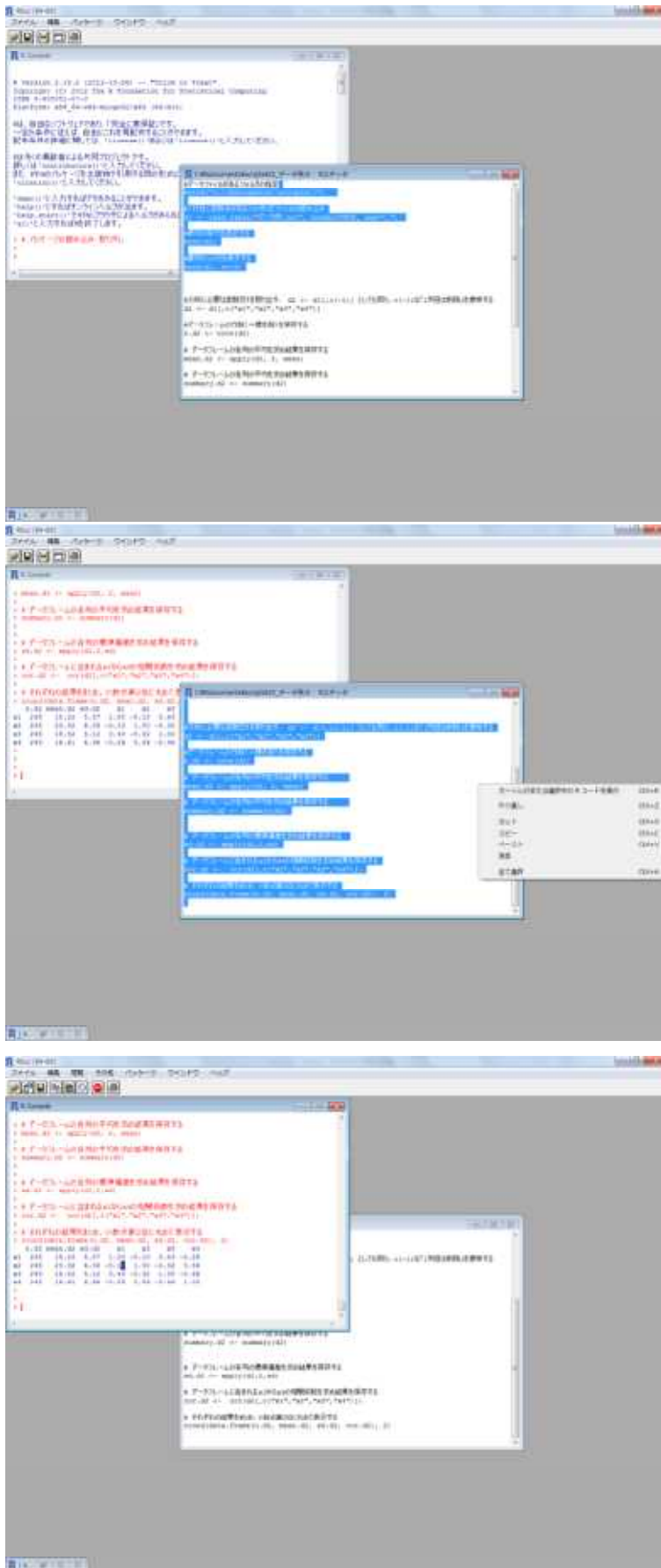


開くファイルを指定する画面が開くので、ドライブ、フォルダを選択移動し、目的のファイルを選択して、「開く」を押す。

読み込んだスクリプトを再編集して保存するときも、「別名で保存」からファイル名を指定して保存するほうが間違いがない。「上書き保存」ではファイル名が文字化けすることがある。

同名保存する場合、「既に存在します。上書きしますか?」と訊いてくるので、良ければ「OK」を押す。

## スクリプトの実行



**スクリプトの実行したい部分を選択する。** スクリプト全体を選択するのであれば、スクリプトエディタ上で右クリックし、「全て選択」を選ぶ。

スクリプトエディタ上で右クリックし、「カーソル行または選択中のRコードを実行」を選ぶ。

スクリプトが実行され、結果が表示される。

スクリプトとともに、コンソール画面に結果が表示される。

実行されたスクリプトは、コンソール画面上において、「>」が左端について表示される。

図は、グラフィクス画面に出力される。

スクリプトは、毎回すべて実行する必要はありません。実行したい部分だけを選択して実行することができます。

ただし、前の分析結果が残っていたりして、古い分析結果が出力されてしまう危険性もあります。

スクリプトがすべて完成したら、一度、すべてを選択して実行することをお勧めします。

**オブジェクトの確認・消去**

Rでは、計算や分析の結果をメモリ上に“オブジェクト”として再利用することができる。

オブジェクトを生成するには、たとえば「`r1 <- c(1, 2, 3)`」とすると、`r1`という名前で、`1, 2, 3` を要素に持つベクトルが生成される。

例えば、スクリプトにミスがあるのに、同名の古いオブジェクトが残っていると、過去の結果がいつまでも使われて、ミスに気がつかないことがある。

そこで、いまメモリ上にあるオブジェクト名を確認したり、オブジェクトを消去したりする必要がある。新しく分析を始めるときは、いったんメモリ上のオブジェクトを消去しておいたほうがよい。

**メモリ上にあるオブジェクト名を確認**

```
ls()
```

**メモリ上にあるすべてのオブジェクトを消去**

```
rm(list=ls())
```

```
> r1 <- c(1, 2, 3)
> r1
[1] 1 2 3
```

```
> ls()
[1] "r1"
```

```
> rm(list=ls())
```

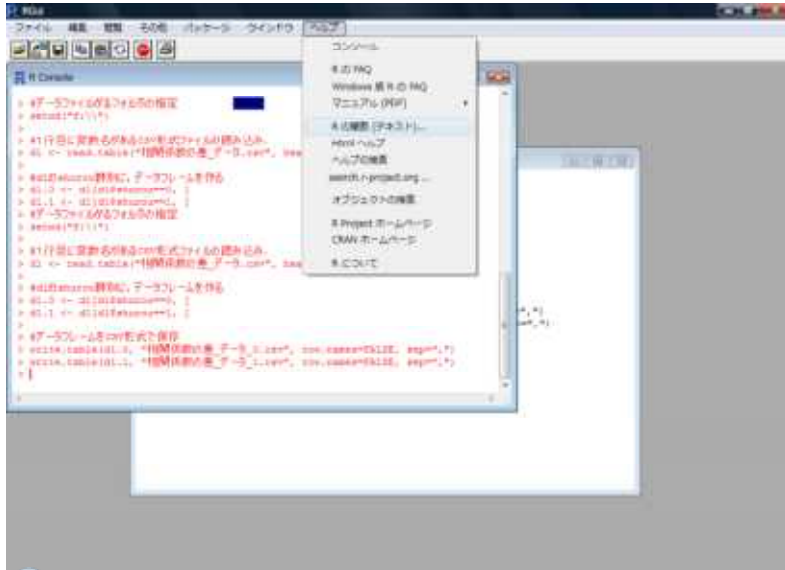
```
> ls()
character(0)
```

```
> r1
エラー： オブジェクト 'r1' がありません
```

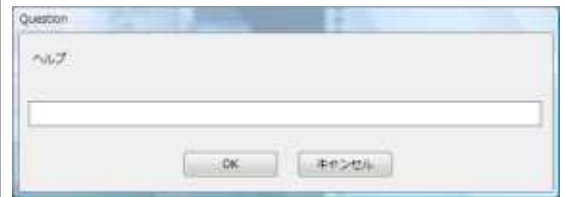


## ヘルプの使い方

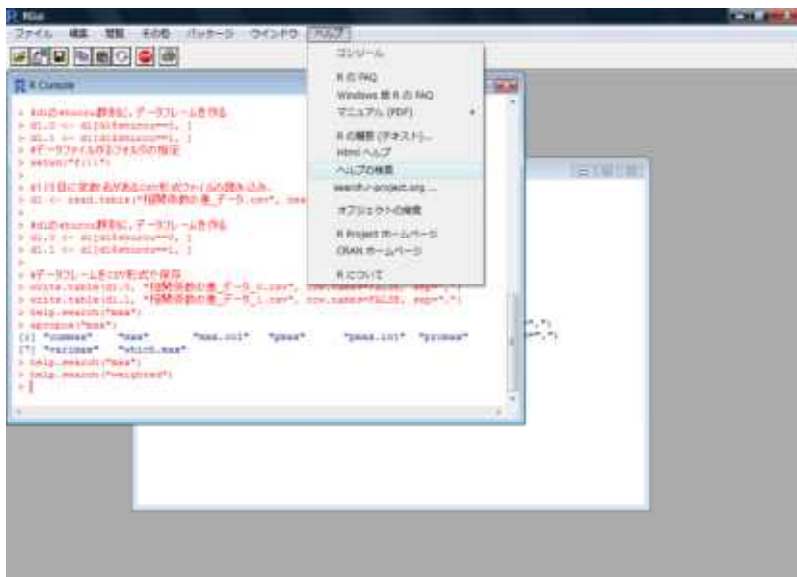
## 使うコマンド(関数名)のスペルが分かっている場合



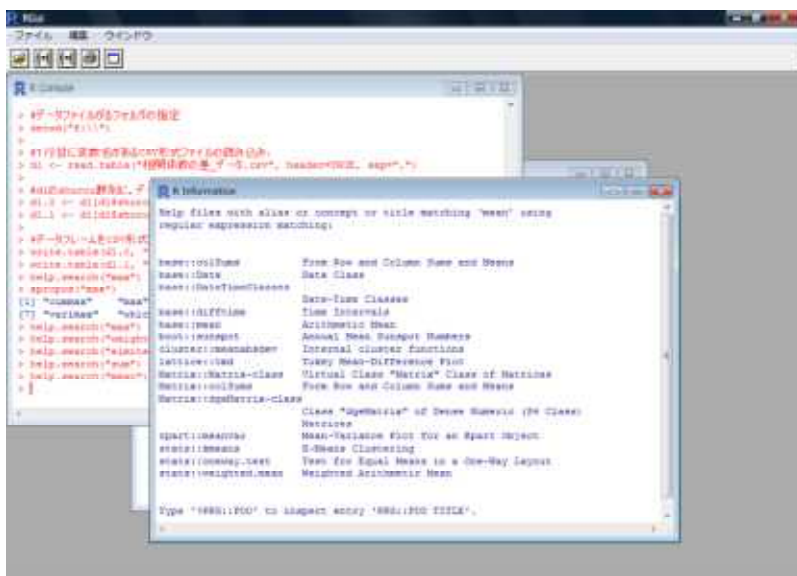
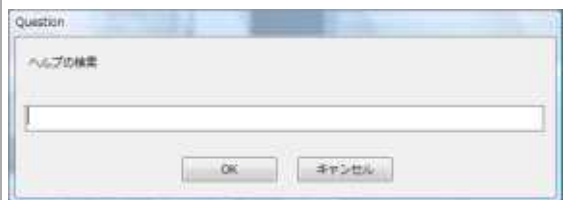
「ヘルプ」→「Rの関数 (テキスト)」とすると、次の入力画面が表示されるので、コマンドのスペルを入力して使い方を調べればよい。



## 使うべきコマンド(関数名)がよくわからない場合: キーワード検索



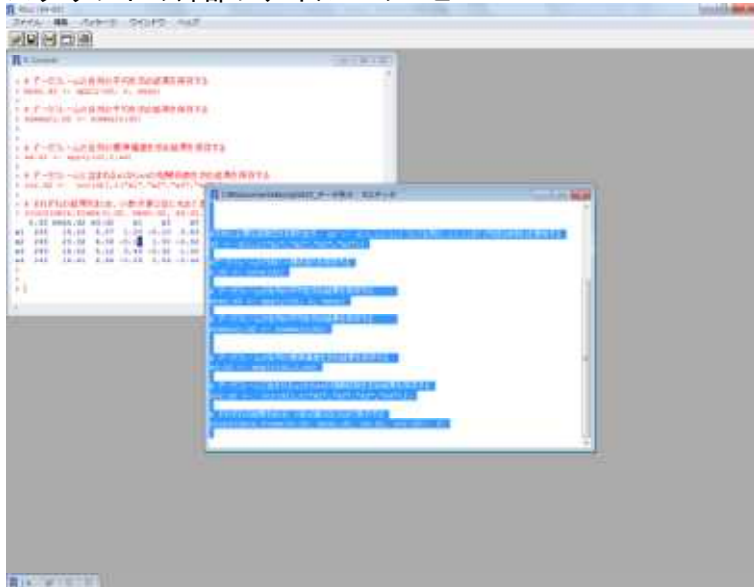
「ヘルプ」→「ヘルプの検索」とすると、次の入力画面が表示されるので、キーワードを入力する。



たとえば、「mean」というキーワードを入力すると、「mean」を含むヘルプ文書の一覧を表示してくれるので、その中からコマンド(関数名)を選んで、上の方法で関数のヘルプを見る。



## スクリプトの外部ファイルへのコピー



スクリプトの実行したい部分を選択する。スクリプト全体を選択するのであれば、スクリプトエディタ上で右クリックし「全て選択」を選ぶ。

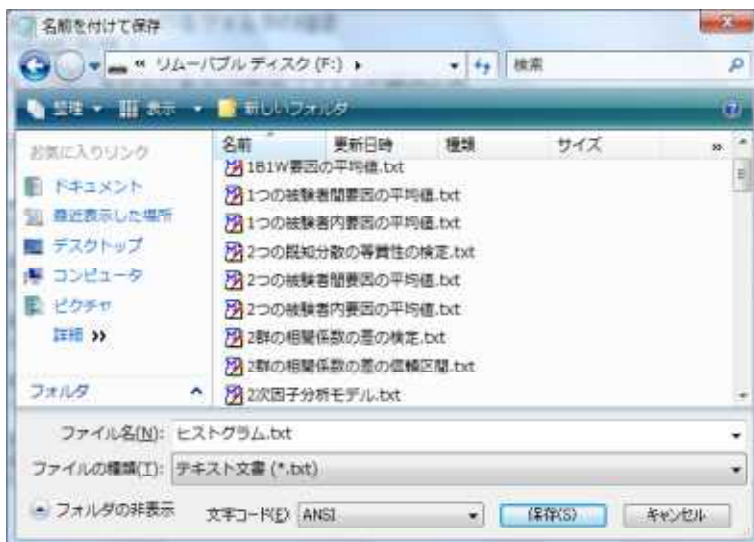
スクリプトエディタ上で右クリックし「コピー」を選ぶか、「CTRL+C」キーを押す。

```
#最初の 15 行を表示する
head(d1, n=15)

#分析に必要な変数だけを取り出す。 d2 <- d1[c(1),] としても同じ。 c(1)は
#データフレームの行数 (= 標本数) を保存する
n.d2 <- nrow(d2)

# データフレームの各列の平均を求め結果を保存する
mean.d2 <- apply(d2, 2, mean)
```

Wordなどを起動し、スクリプトをペースト（貼り付け）する。（一太郎などでも構わない）

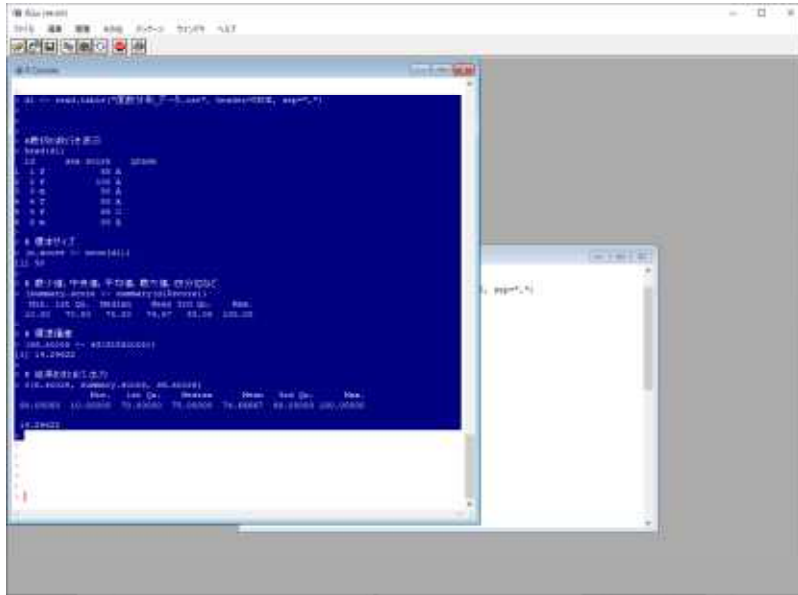


ドライブ、フォルダを移動し、「ファイル」→「名前をつけて保存」を選んで、ファイル名のところに、任意の名前を書いて「保存」する。

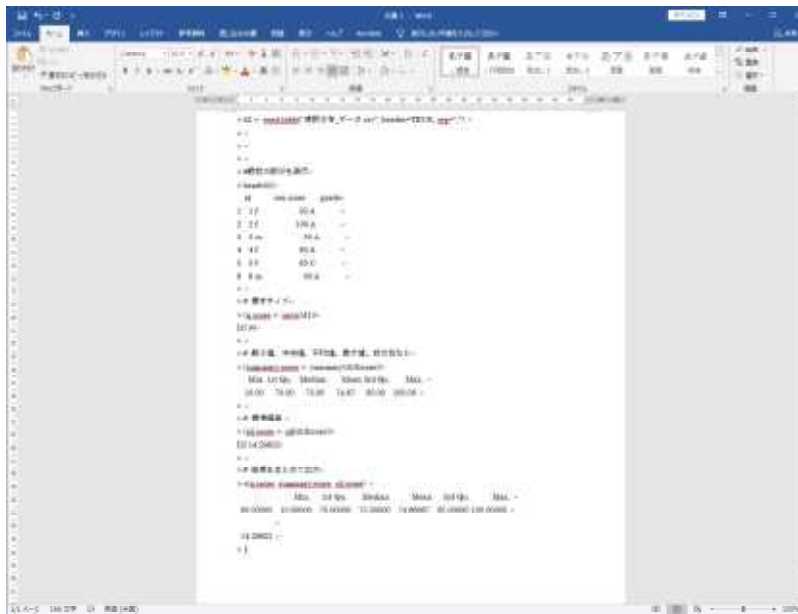
## 出力結果の外部保存 — Wordファイルなどに保存

コピー・ペーストを使う。

## コピー・ペーストを使う方法



Rコンソール画面の該当箇所を範囲選択して、「編集」→「コピー」とするか、「CTRL+C」キーを押す。



Wordなど、適当なソフトを起動し、ペースト（貼り付け）する。



「ファイル」→「名前をつけて保存」と進み、ドライブ、フォルダを指定して、ファイル名のところに任意の名前を書いて「保存」する。

## 出力結果の外部保存 — sink関数

## sink関数を使う方法

sink("ファイル名.txt")

Rスクリプト

sink()

結果のみが保存され、スクリプト部分は保存されない。  
 コンソール画面には、結果は表示されない。  
 Append=T オプションをつけないと、ファイルは上書きされる。  
 ファイルが開いていると保存されない。

```
> setwd("d:¥¥")
> d1 <- read.table("度数分布_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
>
> # sink で結果をファイルに出力するようにする
> sink("sinkout1.txt")
>
> #最初の数行を表示
> head(d1)
>
> # 標本サイズ
> (n.score <- nrow(d1))
>
> # 最小値, 中央値, 平均値, 最大値, 四分位など
> (summary.score <- summary(d1$score))
>
> # 標準偏差
> (sd.score <- sd(d1$score))
>
> # 結果をまとめて出力
> c(n.score, summary.score, sd.score)
>
> # ファイルを閉じる
> sink()
```

	A	B	C	D
1	id	sex	score	grade
2	1	f	85	A
3	2	f	100	A
4	3	m	80	A
5	4	f	80	A
6	5	f	65	C
7	6	m	80	A
8	7	m	75	B
9	8	m	100	A
10	9	f	65	C
11	10	m	55	D
12	11	f	90	A
13	12	m	80	A
14	13	f	65	C
15	14	f	75	B
16	15	m	70	B
17	16	f	70	B
18	17	f	75	B
19	18	m	65	C
20	19	m	50	D
21	20	f	95	A

```
> # コンソール画面には結果は表示されない
```

```
> #最初の数行を表示
> head(d1)
>
> # 標本サイズ
> (n.score <- nrow(d1))
>
> # 最小値, 中央値, 平均値, 最大値, 四分位など
> (summary.score <- summary(d1$score))
>
> # 標準偏差
> (sd.score <- sd(d1$score))
>
> # 結果をまとめて出力
> c(n.score, summary.score, sd.score)
>
```

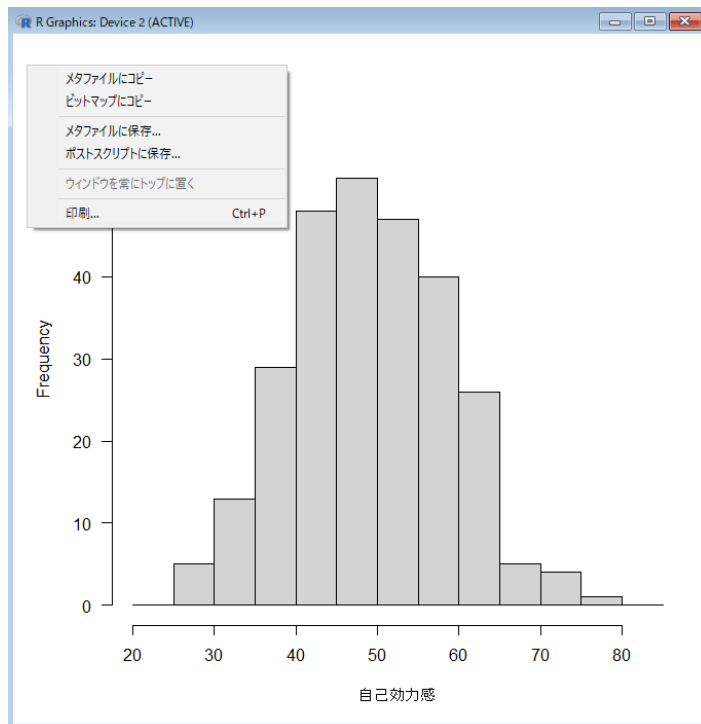
Wordなどを使って，出力結果を確認する

```

┌ id      sex score  grade↵
1  1 f      85 A     ↵
2  2 f     100 A     ↵
3  3 m      80 A     ↵
4  4 f      80 A     ↵
5  5 f      65 C     ↵
6  6 m      80 A     ↵
[1] 90↵
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max. ↵
      10.00  70.00   75.00   74.67  80.00  100.00 ↵
[1] 14.29622↵
           Min.    1st Qu.    Median      Mean   3rd Qu.      Max.
90.000000  10.00000  70.00000  75.00000  74.67000  80.00000 100.00000  14.29622

```

## 図の外部保存 — Wordファイルなどに貼りつけ

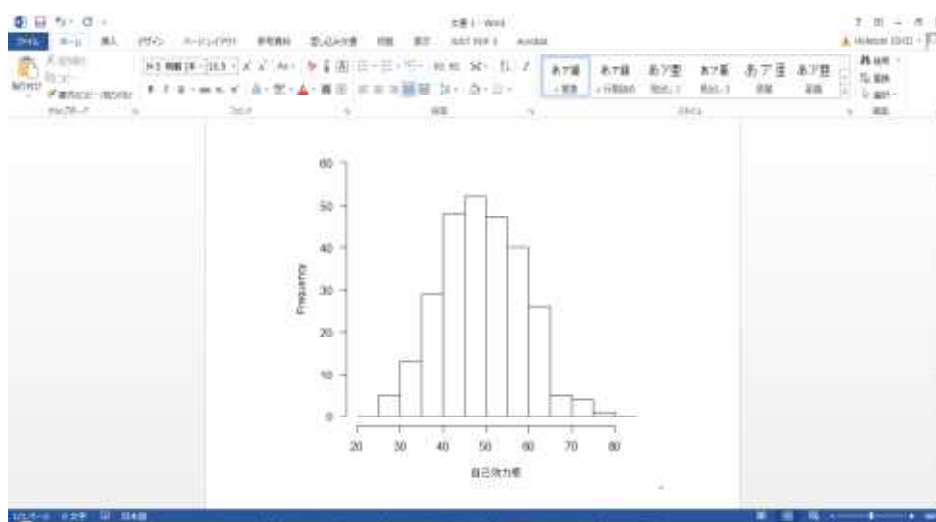


グラフィックス画面上で右クリックし、「メタファイルにコピー」を選択する。

もしくは、図上で左クリックし、「CTRL」＋「C」キー（macの場合は「Command」＋「C」）キーを押す。

貼り付け先によっては、「メタファイルにコピー」だと背景が透明になり、「CTRL」＋「C」だと背景が白になる。

メタファイルに保存やポストスクリプトに保存を選択すると、画像ファイルとして保存することができる。



Wordなどを起動し、ペースト（貼り付け）し、保存する。

もしくは、「CTRL」＋「V」キー（macの場合は「Command」＋「V」）キーを押して貼り付ける。

図を貼り付けたファイルをWordファイルとして保存する。

同様にして、1つのWordファイルに何枚も図を貼り付けることができる。

画像ファイルとして保存した図をWordに貼り付けたい場合は、次のようにする。

- 1) エクスプローラを開き、画像ファイルが保存されているフォルダを表示する。
- 2) Wordを起動する。新規にファイルを開くか、既存のファイルを開く。
- 3) Wordの画面上に、エクスプローラから画像ファイルをドラッグ&ドロップする。これで画像がWord上にコピーされるはず。

## 図の外部保存 — PDFファイルに保存

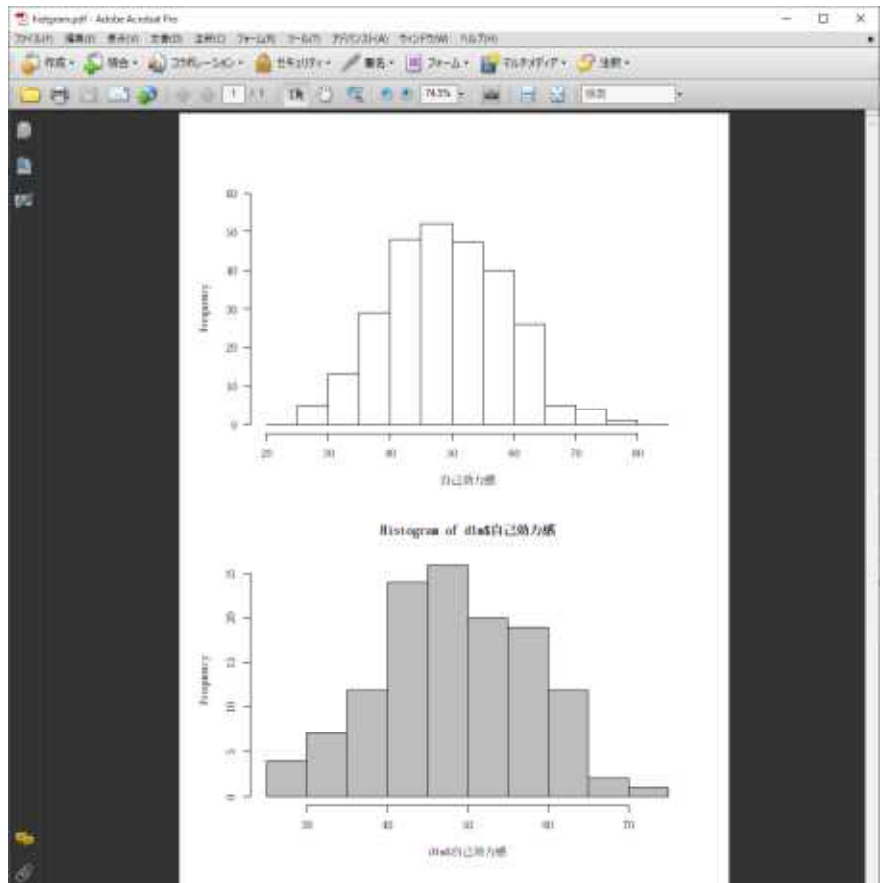
図の出力先をBMP, JPEG, PNG, TIFF, PDF形式ファイルにする `bmp`, `jpeg`, `png`, `tiff`, `pdf` 関数もある.  
図を作成する関数の前にこれらの関数を置いておくと、指定された形式のファイルに図が出力ができる.

```
pdf("ファイル名.pdf", paper="a4", width=8, height=16, family="Japan1")
layout(matrix(c(1, 2, 3, 4, 5, 6), 3, 2, byrow=TRUE))
```

# 図を作成するスクリプトをここに書く

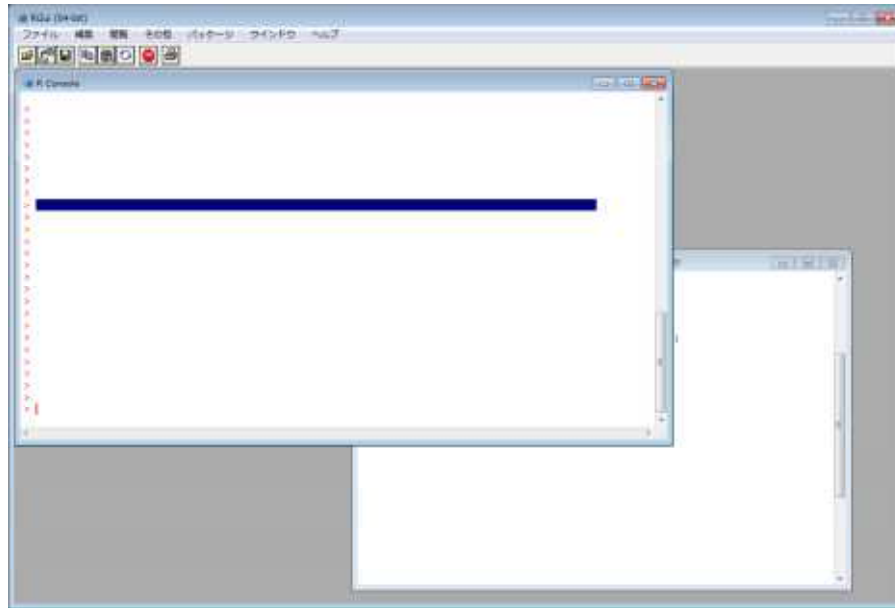
```
layout(1)
dev.off()
```

```
> setwd("e:¥¥¥課題¥¥¥")
>
>
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1     1      20Y1 看護学  女          高          49        23 就職
2     2      20Y2 心理学  男          低          57        29 就職
3     3      20Y1  医学  女          高          42        23 進学
4     4      20Y1 看護学  女          高          41        23 就職
5     5      20Y2  医学  男          低          41        22 就職
6     6      20Y1 心理学  女          低          47        24 就職
>
> # 男だけのデータ
> d1m <- d1[d1$性別=="男",]
>
> # PDF ファイルに図を保存
> pdf("histgram.pdf", paper="a4", width=8, height=16, family="Japan1")
> layout(matrix(c(1, 2), 2, 1, byrow=TRUE))
>
> hist(d1$自己効力感, breaks=seq(20, 85, by=5), ylim=c(0, 60), las=1, xlab="自己効力感", main="")
> hist(d1m$自己効力感, col="gray")
>
> layout(1)
> dev.off()
windows
  2
>
```

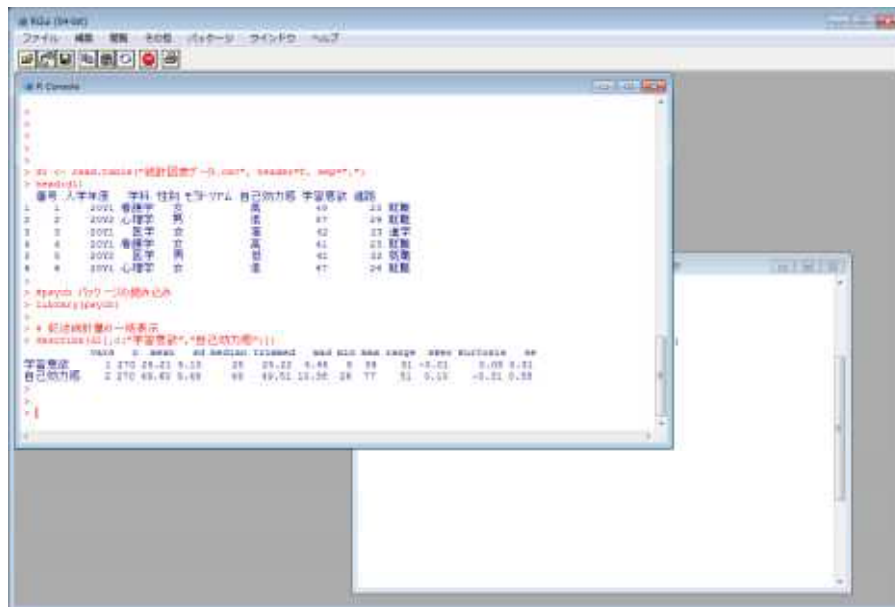


## 出力結果をエクセルで整形保存

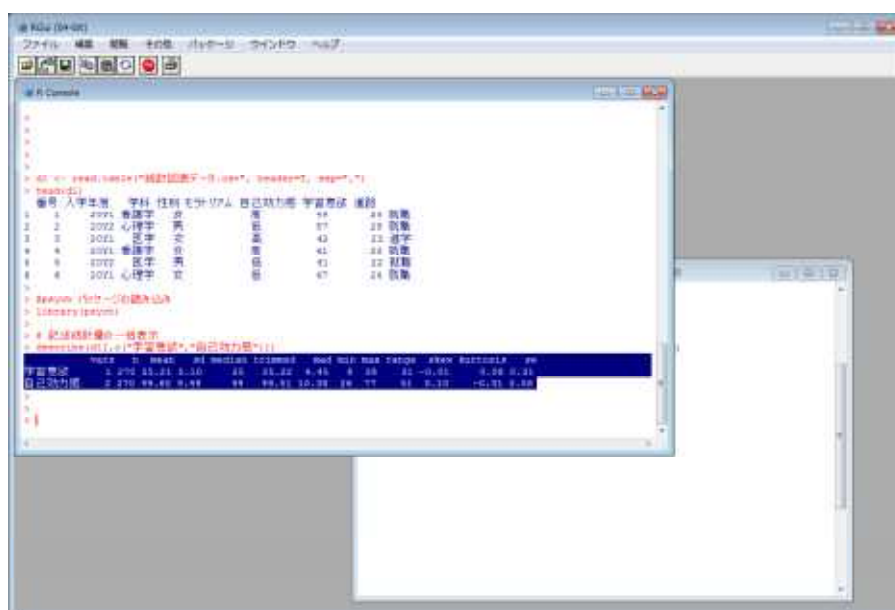
表などは、エクセルの機能を使って、見やすい形に整形保存することができる。



R Consoleウィンドウの大きさを調整して、表が綺麗に表示されるようにしておく。



R スクリプトを実行し、実行結果をR console画面に表示させる。

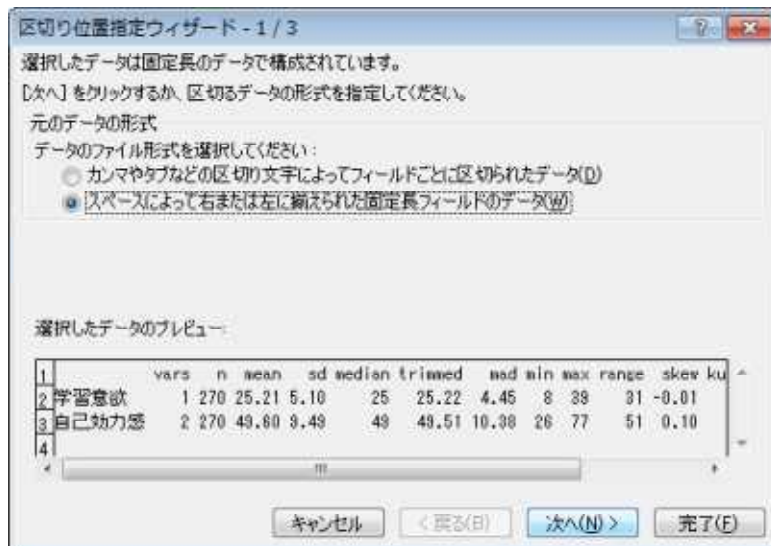
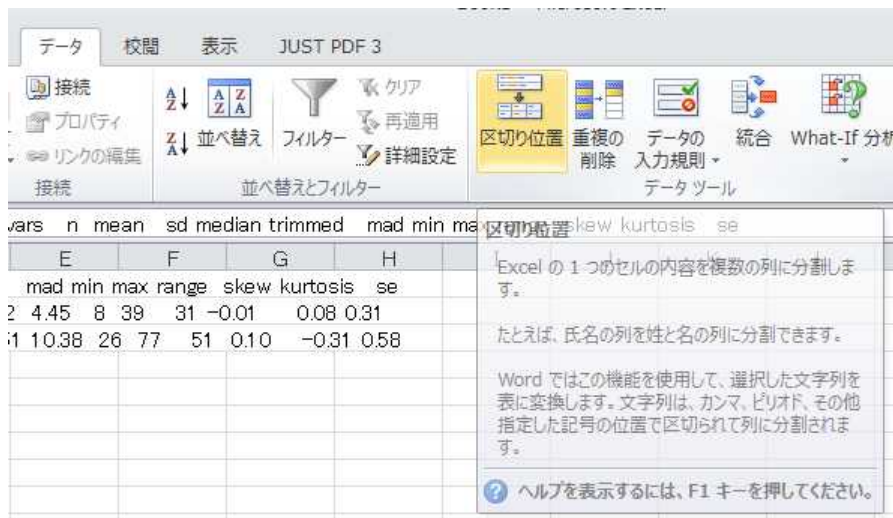


エクセルに貼り付けたい部分を範囲選択してコピーする。

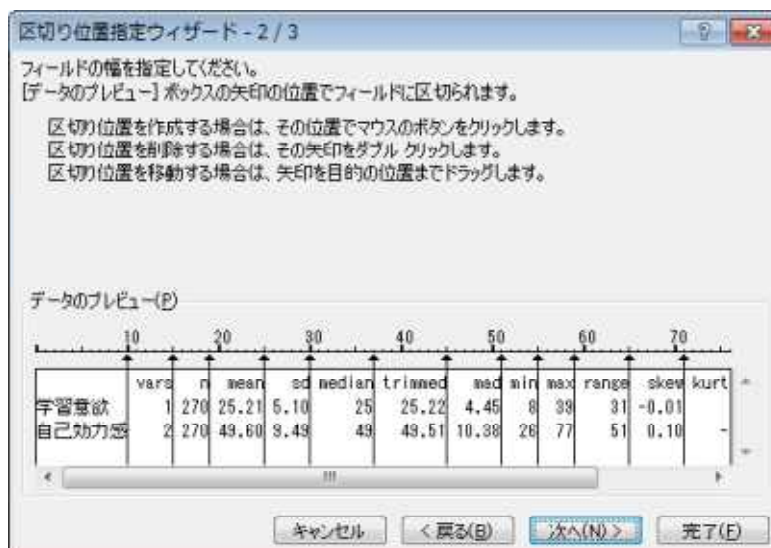


	A	B	C	D	E	F	G	H						
1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se	
2	学習意欲	1	270	25.21	5.10	25	25.22	4.45	8	39	31	-0.01	0.08	0.31
3	自己効力感	2	270	49.60	9.49	49	49.51	10.38	26	77	51	0.10	-0.31	0.58

コピーした部分をエクセルに貼り付ける。



データを区切る形式を聞かれるので、「スペース」を選び、次に進む。



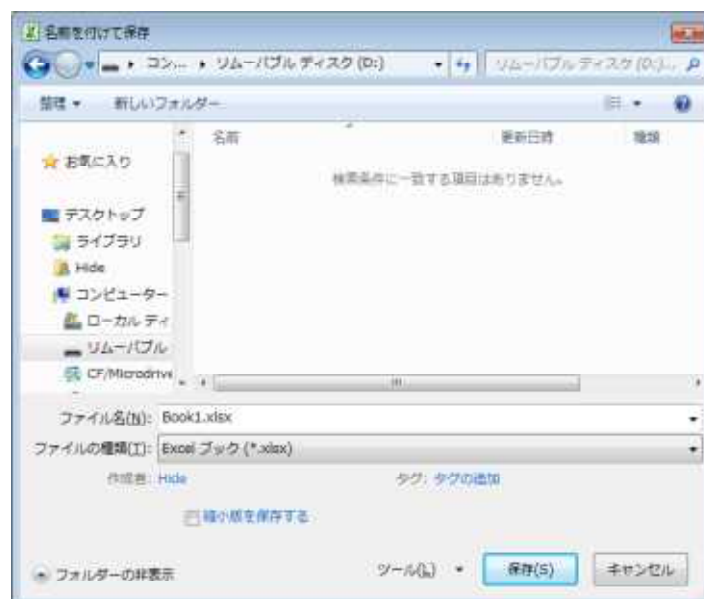
区切り位置が表示されるので、修正の必要があれば区切り位置を調節する。修正の必要がなければ、次に進む。



データ形式（標準，文字列，日付など）の確認画面が表示されるので，修正の必要があれば修正する。必要なければ，完了する。

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1		vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
2	学習意欲	1	270	25.21	5.1	25	25.22	4.45	8	39	31	-0.01	0.08	0.31
3	自己効力感	2	270	49.6	9.49	49	49.51	10.38	26	77	51	0.1	-0.31	0.58
4														
5														

データがセルごとに区切られ，見やすい表の形式となる。



エクセルでファイル保存する。

## Rの情報源



公式ホームページ  
<http://www.r-project.org/>



日本語のRの情報集積ページ  
<http://www.okadajp.org/RWiki/>



青木繁伸先生のRのページ  
<http://aoki2.si.gunma-u.ac.jp/R/index.html>



服部環先生のご著書

心理・教育のための  
 Rによるデータ解析  
 福村出版  
 2011/04



山田・村井・杉澤先生のご著書

Rによる心理データ解析  
 ナカニシヤ出版  
 2015/9/10

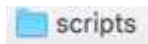
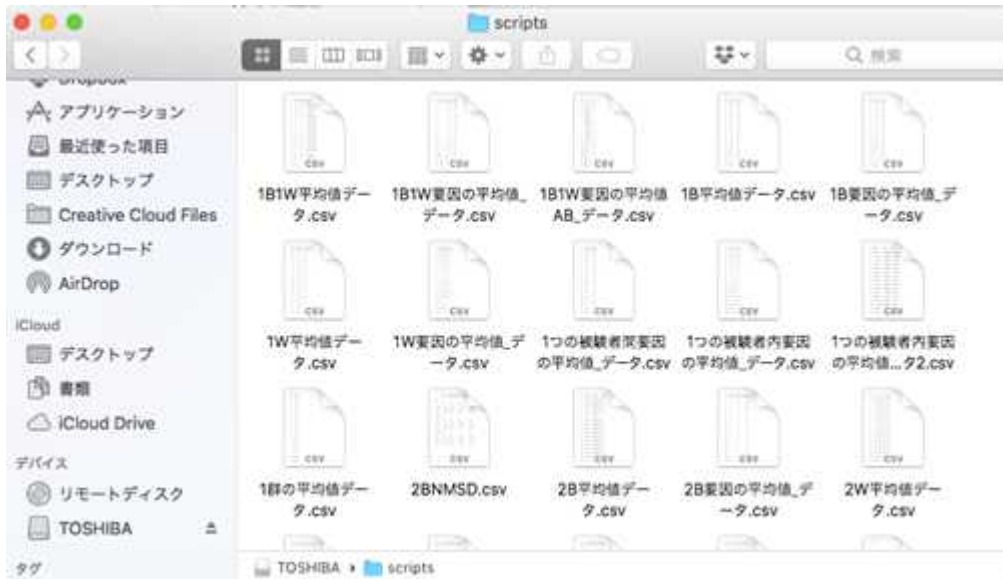
他にもたくさんあります。

## Mac版のRで作業ディレクトリを設定

## 2通りある

## setwdを使って作業ディレクトリを指定する方法

下図のように、あるフォルダにCSVファイルが入っている場合、



のアイコン 

を押し、そのままRのスクリプト作成画面へ移動して離す。




すると、フォルダの絶対パス：/Volumes/TOSHIBA/scripts が得られる。これをダブルクォーテーションで括って、setwd()の中に入れる。

```
setwd("/Volumes/TOSHIBA/scripts")
```

これを実行する。

## 【重要！】

MacOS Mojaveでは、フォルダアイコンを押してRのスクリプト作成画面へ貼り付けることができない。その場合は、設定アイコン  の中から「○○のパス名をコピー」を選択し、Rのスクリプトエディタに貼り付けられる。また、作業ディレクトリとして指定するフォルダを選択の上、「Option」+「Command」+「C」を同時に押すことにより、絶対パス名をコピーすることもできる。

Macの場合は、必ず半角スラッシュ「/」で始める。Windowsで作成したスクリプトを、Macでそのまま通したり、Windows流にバックスラッシュ「¥」でフォルダの階層を表現するとエラーが出る。

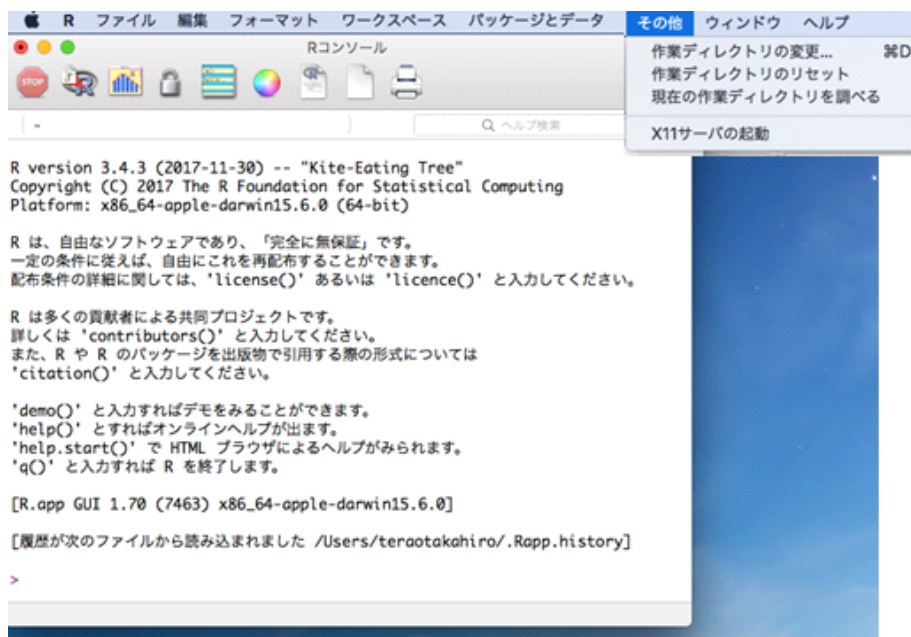
半角スラッシュは、1つでも2つでも認識する。

```
setwd("/Volumes/TOSHIBA/scripts")
setwd("//Volumes//TOSHIBA//scripts")
```

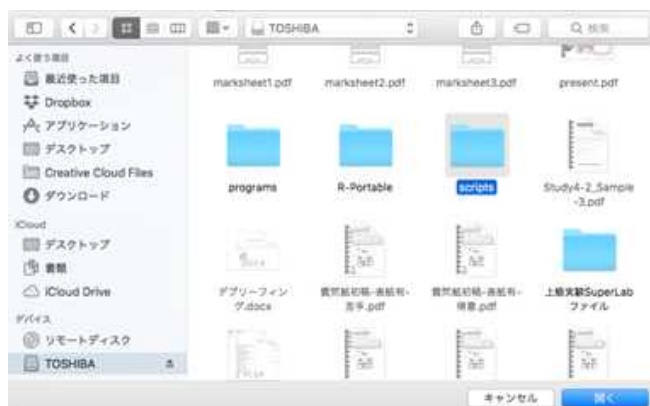
どちらでもよい。

## ツールバーから作業ディレクトリを設定する方法

Mac版Rのツールバーの中の「その他」タブから、「作業ディレクトリの変更」を選択する。



フォルダの選択画面がポップアップするので、CSVファイルの場所を選択し、「開く」を押す。



これでOK。

**Mac版のRでread.table関数でCSVファイルを読み込み**

Windowsと同じようにしてread.table関数でCSVファイルを読み込もうとすると、「不正なマルチバイト文字があります」というエラーが出ることもある。このようなエラーが出た場合は、エンコーディング方法を明示してRに伝える必要がある。

read.tableのオプションで、fileEncoding="cp932" を指定する。  
(write.table関数等でも同様の指定が必要なことがある。)

```
> setwd("//Volumes//TOSHIBA//scripts")
> d1 <- read.table("対応のある2群の平均値データ.csv", header=TRUE, sep=",")
      make.names(col.names, unique = TRUE) でエラー:
      ' <8b><b3><97>{<89>?<da>' に不正なマルチバイト文字があります
```

**# エラーが発生**

```
> d1 <- read.table("対応のある2群の平均値データ.csv", header=TRUE, fileEncoding="cp932", sep=",")
```

**# 今度はうまく行った**



**Mac版のRで図を描いたときに日本語を正しく表示させる**

Mac版のRで図を描いたとき、日本語文字（全角文字）が□□と表示されることがある。このような場合は、`par`関数を使って図を描く前にフォントを指定しておく。

```
par(family="HiraKakuProN-W3")
```

フォント名（HiraKakuProN-W3）は他のものでもよい。

**Mac OS版のRスクリプトファイルをWindows版Rで読む場合のマルチバイト文字の文字化けへの対応**

Mac版のRでスクリプトファイルを作成した人が、Windowsマシンをもつ人とスクリプトを共有する際、日本語文字列が文字化けする。

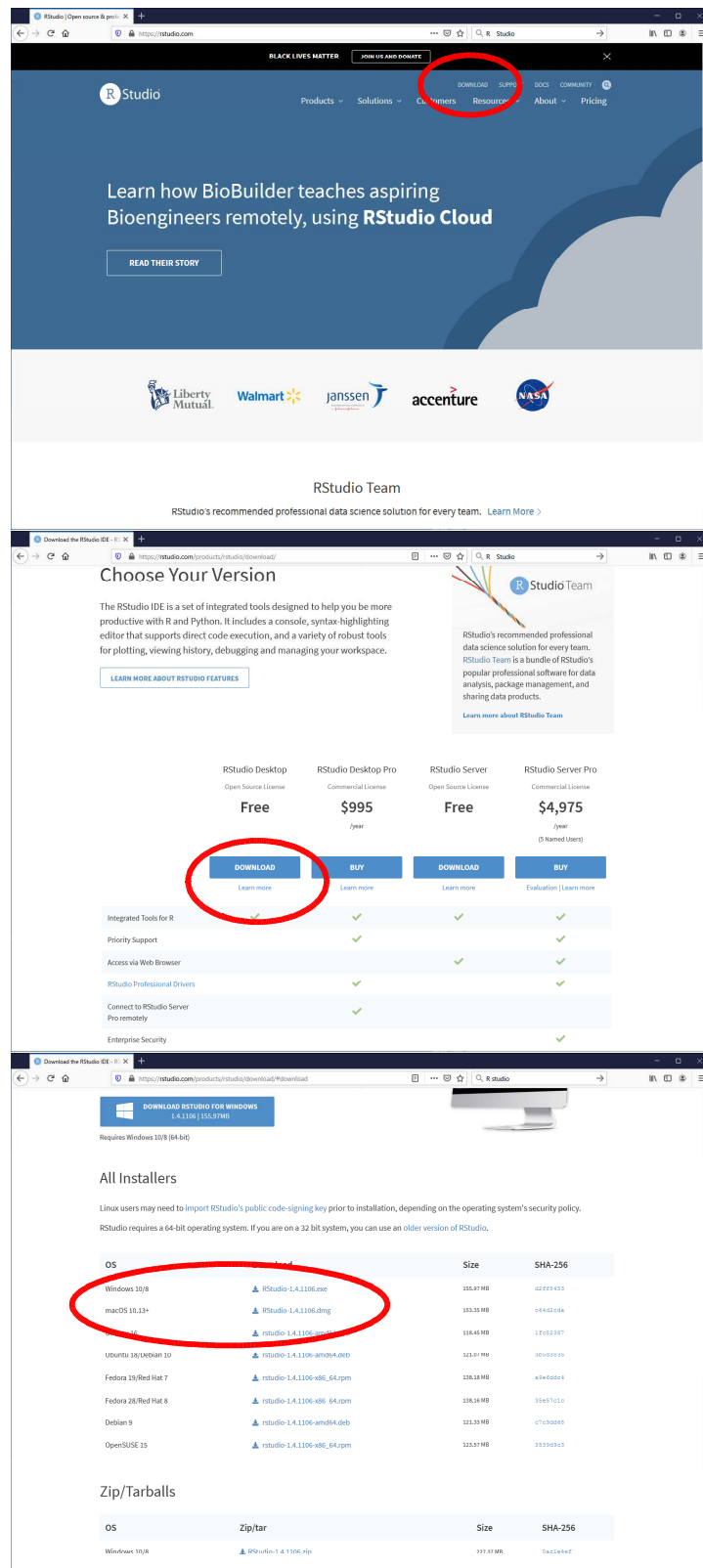
これは、Mac版のRのスクリプトエディタがUTF-8形式でエンコーディングするのに対し、Windows版のエディタではCP932形式でエンコーディングすることによる。

このようなときは、Mac版Rで作成したスクリプトファイルをテキストエディタ（ex. メモ帳）で読み込み、エンコーディング方法に「UTF-8 (BOM付き)」を選択する。

## R Studioについて

Rを実行するにはR本体があれば良いのであるが、より操作性を高めるものとして、R StudioというIDE（統合開発環境）が提供されている。R Studio を使うと、`setwd`の設定やパッケージのインストールをメニューで操作できたり、グラフが扱いやすくなったりする。他にもいろいろ便利な機能がある。

本スクリプト集は、R Studio を使わず、Rだけで実行できるように書かれているが、R Studio のインストール方法は以下の通りである。R Studioの使い方などは、別の情報源から収集のこと。

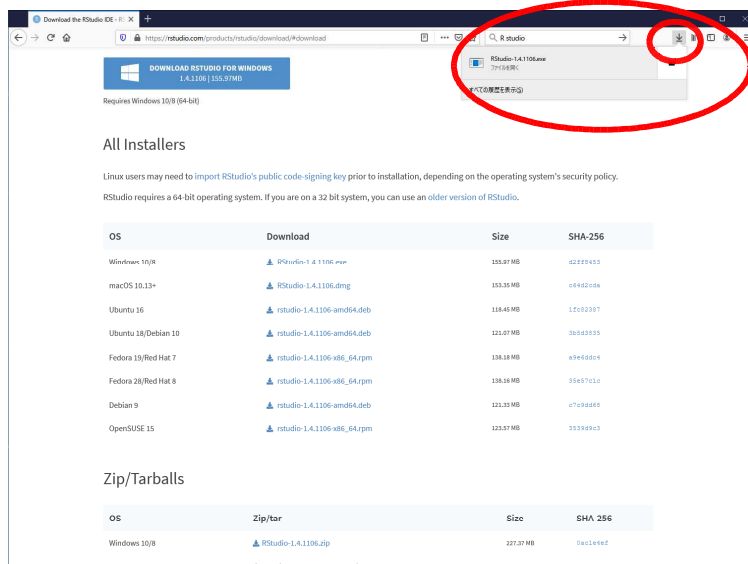


<https://rstudio.com/> で R Studio のホームページにアクセスし、DOWNLOADを選択する。

Free版（または製品版）をDOWNLOAD

インストーラファイルを選択してダウンロード



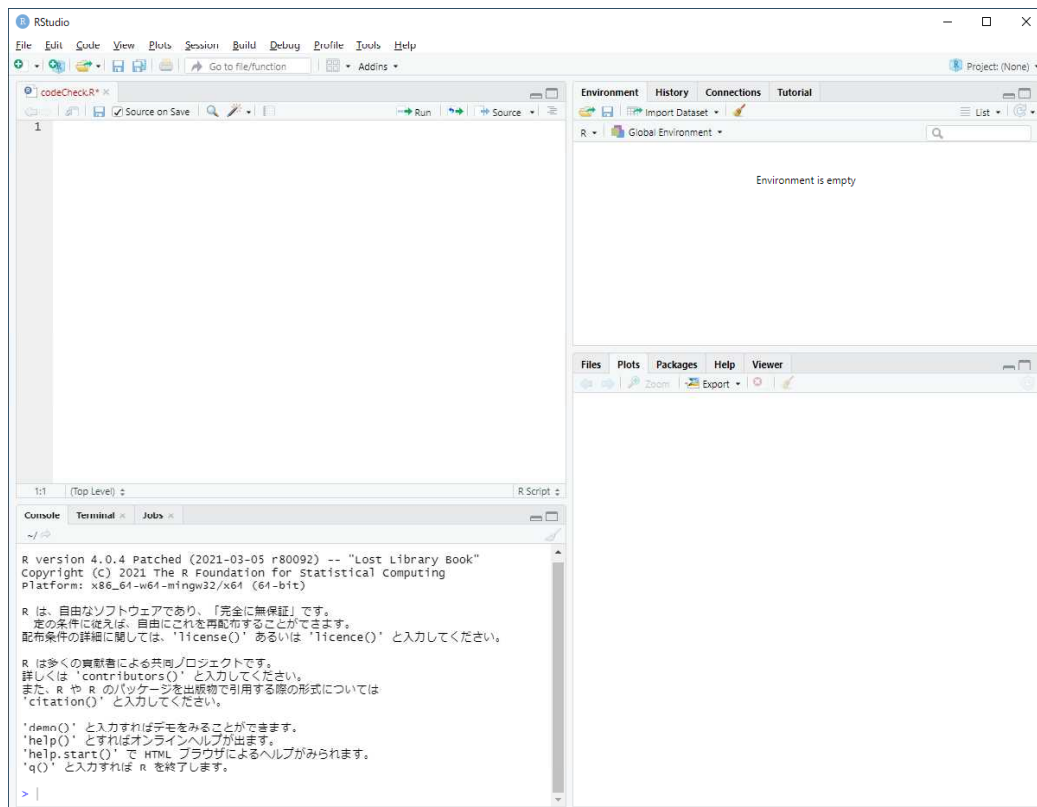


ダウンロードの進捗状況を表示するボタンを押し、ダウンロードしたファイルを実行

指示に従ってインストール

アプリケーションメニューにR Studioが登録される。

## R Studioを起動したところ



## Google ColaboratoryでRを使うための準備

インターネットに接続して、Google Colaboratory を使って R を実行することも可能。  
分析結果をURLで他の人と共有することができる。  
データを毎回アップロードしなければならない。

## Google アカウントが必要

以下のページから作成できる。既に持っている場合はそれが使える  
<https://accounts.google.com/signup>

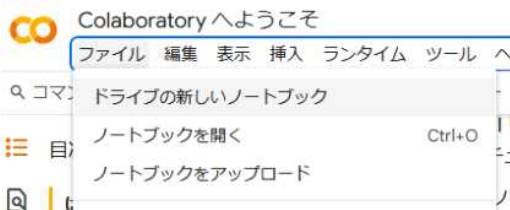
## 環境

パソコンが必要。OSは何でもよい。  
ブラウザは Google Chrome か Firefox を用いる。無い場合はダウンロードする  
Microsoft Edge や Internet Explore では動作しない。

## 設定

以下のURLからGoogle Colaboratoryにアクセスする  
<https://colab.research.google.com>

Googleアカウントでログイン → グレー領域をクリック → ファイル → ドライブの新しいノートブック



→ 新しいノートブックが開く（少し時間がかかる）

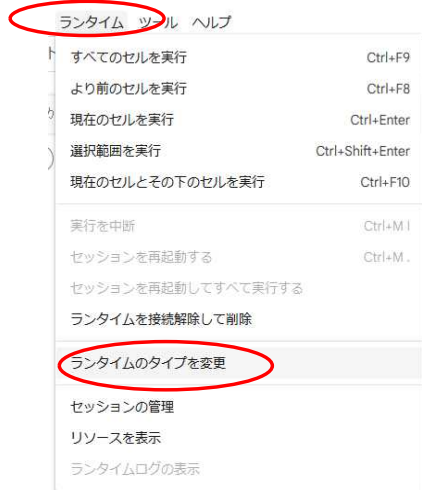


ノートブック名をクリック → ファイル名を「〇〇. ipynb」 のようにを変更する → ファイル → 保存

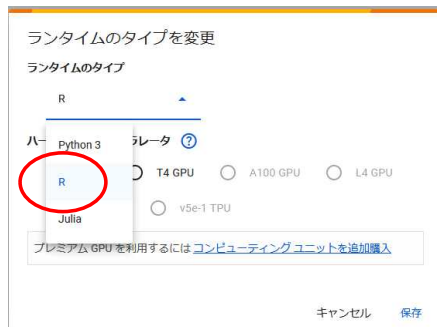


## ランタイムのタイプを変更

新しく開いたウインドウで、ランタイム → ランタイムのタイプを変更 を選択



ランタイムのタイプとしてRを選択 → 保存



これでRを使える環境になる

## Google ColaboratoryでRを使う


Google Colaboratory にログイン

→ 新しいノートブックを開いてランタイムのタイプをRに変更する

または

→ 保存してあるノートブックを開く

グレーのエリアがスクリプトを書く領域

実行するには、Shift+Enter を押すか  マークを押す

「+コード」を押すと、次のスクリプト入力領域が生成される

+コード +テキスト

```
[2] 1+2
     3-4
     5*8
     7/8
     9^2
     sqrt(5)

3
-1
30
0.875
81
2.2360679774997
```

 |

**【重要】** Google Colaboratoy では、起動する度にパッケージをインストールしなければならない。  
MASS, ggplotなどはプレインストールされているので、改めてインストールする必要はない。

```
install.packages("descr")
install.packages("fdth")
install.packages("psych")
```

## Google ColaboratoryにCSVデータをアップロードする

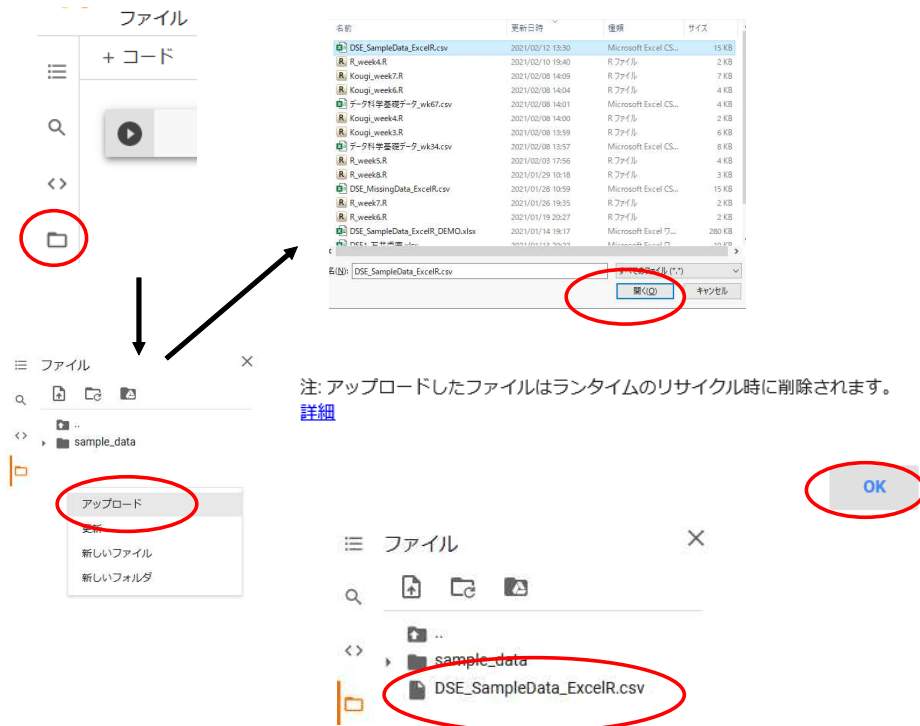
Google Colaboratory にログイン

\*\*\*\*\*.ipynb ファイルを読み込む

左バーにあるファイルアイコンをクリック → 少し待つ → ファイルエリア内で右クリック → アップロードを選択

保存したCSVファイルを選択 → 開く → OK → ファイルが /contentディレクトリにアップロードされる

【アップロードしたファイルは、ログアウトすると削除されてしまうので、次回ログイン時も、改めてアップロードする必要がある】



## データの読み込み・表示

```
rm(list=ls()) # 作業メモリを消去する
read.table("CSVファイル名", オプション) # データを読み込む
head(データフレーム名) # 先頭6行のデータを表示する
```

# Shift-JIS コードのファイルを読み込むときは、オプションに `fileEncoding="shift-jis"` を設定する。

```
rm(list=ls())
d1 <- read.table("DSE_SampleData_ExcelR.csv", header=TRUE, sep=";", encoding="UTF-8")
head(d1)
```

	変数名	変数の型	データ	unitID	year	area	future	club	motiveL	motiveE	stress	fear	supp
1		<int>	1	20X1	理系	進学	0	17	18	20	22		
2		<int>	2	20X2	文系	就職	1	16	14	23	48		

## Google Colaboratoryに日本語フォントをインストールする

図で日本語文字を表示するためには、日本語フォントをインストールする必要がある

`system("apt-get -y install fonts-ipafont-gothic")` でフォントをインストールする

`systemfonts::system_fonts()` でフォントを確認する

`par(family= "フォント名")` で、図で使うフォントを指定する  
IPAGothic や IPAMinchoなどを使う

```
system("apt-get -y install fonts-ipafont-gothic")
systemfonts::system_fonts()
```

path	index	name	family	style	weight
<chr>	<int>	<chr>	<chr>	<chr>	<ord>
'ipamp.ttf	0	IPAPMincho	IPAPMincho	Regular	normal
is-Bold.ttf	0	LiberationSans-Bold	Liberation Sans	Bold	bold
o-Bold.ttf	0	LiberationMono-Bold	Liberation Mono	Bold	bold
-gothic.ttf	0	IPAGothic	IPAGothic	Regular	normal

```
par(family="IPAGothic")
```

## Google Colaboratoryのノートブックを保存する

Google Colaboratory で書いたスクリプトと分析結果は、ノートブック（.ipynbファイル）としてファイルに保存することができる。

ファイル → 保存



ノートブックをUSBやパソコンに保存する場合：

ファイル → .ipynbをダウンロード → ファイルを保存 → ドライブやフォルダを指定して保存



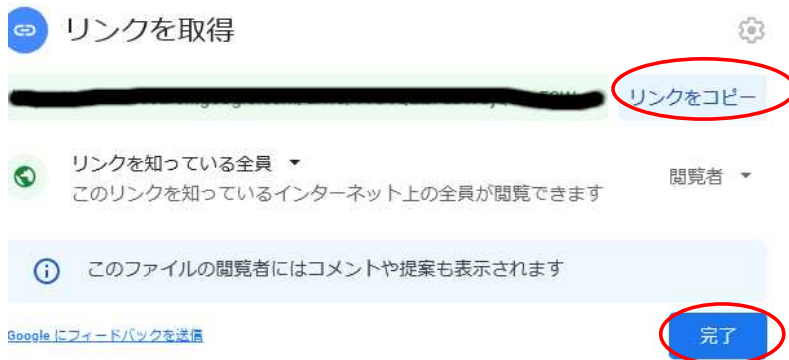


## Google Colaboratoryのノートブックを共有する

共有 をクリック → リンクを知っている全員に変更 をクリック

リンクをコピー → 完了

ノートブックを共有したい人にリンク先URLを伝える



## 2 Rのスクリプト作成時の注意点と エラーへの対処

## スクリプト作成にあたっての一般的注意

### スクリプトの冒頭に作業メモリを空にするコマンドを書く

Rでは、以前に扱ったデータを保存しておいて、次にまた利用できるという機能があります。Rを終了するときに、「作業スペースを保存しますか？」で「はい」と答えると「`~.RData`」というファイルが作成されます。この「`~.RData`」というファイルを開くと、前に使っていたときのように、データが作業メモリ上に復活します。

便利なのですが、スクリプトにエラーがある場合に問題を起こすことがあります。スクリプトにエラーがあると、メモリ上のデータは上書きされません。しかし、何らかのデータがメモリ上に残っていればそれが使われるため、エラーがあっても実行結果が表示されてしまいます。たとえば、新しいデータを読み込んで分析結果を表示させるはずだったのに、古いデータの分析結果が残っていてそれが表示されてしまうというようなことが起こります。ですので、Rを終了するときに「作業スペースを保存しますか？」は「いいえ」と答えることをお勧めします。

作業メモリ上のデータを消去するため、Rスクリプトの先頭に、次の一文をいれておくことを推奨します。

```
rm(list=ls())
```

これを実行すると、メモリ上のデータが消去されます。`rm`は `remove`、`ls()` はすべてのリスト（作業メモリにあるデータ）を意味します。つまり、上のスクリプトは、消去しなさい(`rm`)、対象リストは(`list=`)、すべてのリスト(`ls()`)です、という命令になります。

### バックスラッシュと円マーク

Rで作業ディレクトリを設定するとき、バックスラッシュ `\` もしくは円マーク `¥` が出てきます（実際はいずれも半角）。同じ記号を **2つ入れます**（詳しくは、本スクリプト集の「**作業ディレクトリの設定**」の項を参照）。

コンピュータでは、バックスラッシュと円マークに同じコード番号を使っているため、その番号をどう翻訳するかで表示が変わってきます。

コード番号を記号に変換する翻訳体系として、UTF とか、Shift JIS などがあり、どの体系を使うかで、表示される記号が異なってきます。

コード番号は同じなので、スクリプト上の記号がどちらになっても、問題はありません。

Rに限らず、コンピュータでソフトを使っているとき、コード翻訳体系を変えようとまく動かないこともありますので、ソフトが指定してくるコード体系のままにしておいたほうが無難です。

つまり、バックスラッシュで表示されるならそのまま、円マークで表示されるならそのまま、コピーペーストして変わるなら変わったままにしておきます。

とりあえずは慣れて下さい。どうしても気になるときはコード体系を変えることが考えられますが、他に影響がでる可能性もあるので注意して下さい。

### コメントを書くときは # を使う

スクリプト中でコメントを書きたいこともあります。そのような場合は、シャープ記号 `#`（半角）を用います。Rは、`#`で始まる文を読み飛ばします。例えば、

```
# 統計学レポート課題3
```

と冒頭に書いておくと、そのスクリプトが何のためのものか分かります。また、

```
xa <- abs(x) # xの絶対値
```

のように、命令文の右にコメントを書くことも可能です。

## 半角カナ文字を使わない

Rは世界中で使われているソフトなので、日本語対応が完全でない場合があります。とくに半角カナ文字は特殊で、半角カナ文字を使用していることでエラーになることも多くあります。

それゆえ、データ、変数名、ファイル名、フォルダ名等のいずれにおいても、半角カナ文字は使用しないことを強く推奨します。どうしてもカナ文字を使う必要がある場合は、全角カナ文字を用いるようにします。

## 全角文字もなるべく使わない

半角カナ文字と同様、全角文字も使わないに越したことはありません。全角文字を使用していることでエラーになることがあります。Rを使うものについては、データ、変数名、ファイル名、フォルダ名等のいずれにおいても、なるべく全角文字は使わず、半角英数文字を使うことをお勧めします。

スクリプト画面においては、全角のスペースやカッコと、半角のスペースやカッコとの見分けがつきにくく、エラーのもとになります。ファイル名や変数名で全角文字を使用したときは、とくに注意が必要です。

なお、変数名については、ごく簡単な英単語を除き、ローマ字表記するのが分かりやすいです。

## スクリプトの各行の先頭に「>」「+」「1:」などを書かない

スクリプトを実行すると、コンソール画面では、スクリプトの左端に「>」「+」「1:」などの記号をつけて表示します。「>」は各命令文の始まり、「+」や「1:」は命令文中の改行を意味します。コンソール画面の出力を参考にしてスクリプトを書くときは、これらの記号を外して（書かないで）、スクリプトを書くようにしてください。

## データファイルの変数名とRスクリプトの変数名を一致させる

データファイル（CSVファイル）の変数名と、Rスクリプトの変数名が一致していないと、エラーになります。注意しなければならないのは、CSVファイルの変数名におけるスペース（半角、全角）やハイフン（-）は、Rでは「.」（半角ピリオド）に変換されることです。

たとえば、CSVファイルでの「x 2」（半角x 半角スペース 半角2）という変数名は、R上では「x.2」（半角X 半角ピリオド 半角2）となります。Rで「x 2」という変数を使おうとしても、「x 2」という変数はないので（あるのは x.2）、エラーとなります。この場合は、CSVファイルの変数名を x2 とするか、Rのスクリプトを x.2 とするかの、いずれかを行う必要があります。

このように混乱のもとになるので、データファイルの変数名にはスペースやハイフンを入れず、なるべく英数半角文字で書くことをお勧めします。ただし、先頭に数字を使うことはできません。

なお、例えば「情緒的サポート」の変数名を、support.emotion, supportEmotion, support\_emotion などと書くように決めておくと、「道具的サポート」は、support.tangible, supportTangible, support\_tangible のように統一的に表記することができ、何の変数であるかが分かりやすくなります。

## Rの予約語をオブジェクト名に用いない

if, for, TRUE, NA など予約語と言われるものは用途が限定されているので、たとえば、if <- 1 のように値を代入しようとしてもエラーになります。Rの予約語は以下の通りです。

for	in	if	else	repeat	while	next	break
TRUE	FALSE	NULL	Inf	NaN	NA	function	
NA_integer_		NA_real_		NA_complex_		NA_character_	

**pi, T, Fなどをオブジェクト名に用いない**

また、pi, T, F はそれぞれ、円周率, TRUE, FALSE を意味する記号として初期設定されているので、これらの記号をオブジェクト名に使うのは避けるべきです。たとえば、pi はそのままでは円周率の値ですが、以下のように1という値を代入してしまうと、以降、円周率として使えなくなってしまいます。

```
> pi
[1] 3.141593
>
> pi <- 1
>
> pi
[1] 1
```

**代入結果を表示するときは当該のスクリプト部分を () で囲う**

Rでは、命令文を実行した結果（データ、値、分析結果など）を「<-」を用いて代入保存することができます。例えば、

```
x <- mean(c(1, 2, 3, 4))
```

とすると、x には 1, 2, 3, 4 という4つのデータの平均値 2.5 が代入されます。しかし、これを実行しても、コンソール画面上では、

```
> x <- mean(c(1, 2, 3, 4))
```

と表示されるのみで、xの値は表示されません。代入することしか命令されていないので、それを行ったらそれで終わりなのです。

結果を代入して、さらに、その値をコンソール画面上に表示させたいときは、

```
(x <- mean(c(1, 2, 3, 4)))
```

のように、命令文全体を () で囲います。こうすると、

```
> (x <- mean(c(1, 2, 3, 4)))
[1] 2.5
```

のように結果が表示されます。

**改行の位置に気をつける**

スクリプトにおける改行のルールは基本的に以下の通りです。

- **1つの行に複数の命令文を書くことは、基本的にはできない**

例えば、head(d1) (d1の最初の数行を表示) と、nrow(d1) (d1の行数を表示) という2つの命令文を、

```
head(d1) nrow(d1)
```

と書いてはいけないということです。

- **1つの行に複数の命令文を書くときは、セミコロンなどで文を区切りを明示する**

上記の場合、

```
head(d1); nrow(d1)
```

とのように、間を ; で明示すれば、ここが切れ目だとRが理解することができるので、1つの行に複数の命令文を書くことができます。

- ・ひとかたまりの単語（や項）の途中で改行しない

例えば、 $10^{\text{rdgts}}$ （10のrdgts乗）というひとかたまりの項は、どこに改行をしてもいけないということです。これは、英単語を途中で改行してはいけないのと同じです。

- ・1つの命令文を複数の行に分けるときは、要素の切れ目で行変える

例えば、

```
d1$gr <- cut(d1$x1, breaks=c(-Inf, med1, Inf), right=F, labels=c("L", "H"), ordered_result=TRUE)
```

というスクリプトであれば、

```
d1$gr <- cut(d1$x1, breaks=c(-Inf, med1, Inf), right=F,
             labels=c("L", "H"), ordered_result=TRUE)
```

などとします。なお、改行したスクリプトを実行すると、コンソール画面では、

```
> d1$gr <- cut(d1$x1, breaks=c(-Inf, med1, Inf), right=F,
+             labels=c("L", "H"), ordered_result=TRUE)
```

のように表示され、スクリプトの1行目に「>」、2行目以降に「+」がついて表示されます（「1:」のような番号がつくこともあります）。この「>」や「+」（および「1:」など）をスクリプトに書くとエラーになりますので、気をつけてください。

- ・計算式では、演算記号の後で改行する

例えば、 $1+2+3+4+5$ を2行に分けるとき

```
1+2+3
+4+5
```

としてしまうと、 $1+2+3$  と、 $4+5$  という2つの計算式を入力したことになってしまい、実行すると 6 と 9 という2つの結果が出力されます。このような場合は、

```
1+2+3+
4+5
```

と1行目は「+」の後で改行し、まだ式は完結していないことを示した上で2行目を書きます。そうすると1つの計算式と認識され、15 という結果が得られます。

- ・for文、if文等における改行

条件式は丸カッコ ( )、命令文は波カッコ { } で括ります。命令文はセミコロン ; で区切って同じ行に書いても構いません。

if-else文では、if文の最後の } と else { は同じ行に書きます。

```
for(条件式){
  命令文
  命令文
  ...
}
```

```
if(条件式){
  命令文
  命令文
  ...
} else {
  命令文
  命令文
  ...
}
```

## 欠測のあるデータの分析について

実際にデータを収集したときは、多くの場合で欠測データが生じます。欠測のあるデータをRで分析する場合、多くの関数は自動的に欠測データを除外して分析しますが、いくつかの関数では NA (Not Available) という結果を返したり、欠測の処理の仕方によって異なる結果を出力するので注意が必要です。詳しくは、本スクリプト集の「**欠測値の扱い**」の項を参照してください。

## 四捨五入について

Rで四捨五入を行う関数として `round` 関数があります。

`round(数値, 小数点以下桁数)`

と設定することにより、指定した桁数までに数値を四捨五入します。桁数を省略したときは整数になります。しかし、`round`関数について注意しなければならないのは、

```
> round(1.5)
[1] 2
```

```
> round(2.5)
[1] 2
```

のようになるところです。通常の四捨五入なら 2, 3 という結果になるはずですが、Rの`round`関数では、`~5` という数値があったとき、`~`が偶数なら切り捨て、`~`が奇数なら切り上げという計算をします。

通常の四捨五入を行うためには次のようにします。

`sign(x)*(floor(abs(x)*10p + 0.5)/10p)`

`sign`は`x`の符号(+, -)を与える関数、`floor`は小数点以下を切り捨てる関数、`abs`は絶対値を求める関数、`p`は小数点以下の桁数です。詳しくは、本スクリプト集の「**切り上げ, 切り下げ, 四捨五入**」の項を参照してください。

## 作成したスクリプトを少しずつ実行して確認する

スクリプトをすべて書いてから実行するよりも、少し書いてはその部分を実行することを繰り返したほうが、エラーを早く見つけることができます。確認にあたって、いつも上から全部実行する必要はなく、必要な箇所だけを選択して実行します。

エラー修正（プログラムのデバッグ）にあたっては、上から順に必要な箇所を選択し、実行結果を確かめて行きます。1命令文ごとに実行し、結果を確認することをお勧めします。

`for`文は複数行にわたりますので、例えば `i <- 1` のように初期値を設定しておいて、`for`文の `{}` 中を1命令文ずつ実行していきます。

`if`文や`ifelse`文では、条件式だけを実行して `TRUE` か `FALSE` の値が得られるか確認します。データに欠測値があると `NA` という結果になり、エラーの原因となります。

図を複数描く場合も、1つずつ実行しないと最後の図しか画面上に残らなくなります。たとえば、データを読み込んで、クロス表、円グラフ、帯グラフをそれぞれ作成したい場合は、まず、データやパッケージを取り込みます。これは共通部分で、全体を通して1回だけ行えば十分です。

```
rm(list=ls())
setwd("d:¥¥Rreport¥¥")
dl <- read.table("度数分布_データ.csv", header=TRUE, sep=";", file=Encoding="shift-jis")
library(descr)
```

そのあとクロス表を作成します。



```
ctd1 <- CrossTable(d1$sex, d1$grade)
```

円グラフは次のスクリプトを実行することにより作成できます。

```
t.grade <- table(d1$grade)  
pie(t.grade, clockwise=TRUE)
```

帯グラフは以下を実行することにより作成されます。

```
t2 <- table(d1$grade, d1$sex)  
p2 <- prop.table(t2, 2)  
barplot(as.matrix(p2), horiz=TRUE, beside=FALSE, las=1, ylim=c(0,1), width=0.3, legend.text=T)  
segments(0, 0, 0, 1)
```

クロス表、円グラフ、帯グラフの手順はそれぞれ独立していますので、それぞれ当該のスクリプトだけを範囲選択して実行します。

## よくある質問・エラー

## データが読み込めない (rm, setwd, read.table)

Rを使い始めの頃に良くあるのが、データの読み込みエラーです。rm, setwd, read.table等の関数でエラーが発生している可能性があります。それぞれのエラーの原因と対処法について説明します。

USBに「Report」というフォルダを作り、その中に「data1.csv」というcsvデータファイル(Shift JIS コード)があるとします。

USBドライブが「D」ドライブなら、データ読み込みのスク립トは次のようになります。

```
rm(list=ls())
setwd("D:¥¥Report¥¥")
d1 <- read.table("data1.csv", header=TRUE, sep=",", file=Encoding="shift-jis")
```

ここで、もし rm の行に間違いがあったら、  
「... は名前あるいは文字列を含んでいなくてはなりません。」  
のようなエラーが出ます。rm(list=ls()) と正しく書かれているか確認してください。

もし setwd の行に間違いがあったら、  
「setwd("d:¥¥Rreport¥¥") でエラー: 作業ディレクトリを変更できません。」  
のようなエラーがでます。いまの場合、フォルダ名を「R\_report」と書き間違えていますので、「Report」と修正します。

もし read.table の行のファイル名に間違いがあったら、  
「file(file, "rt") でエラー: コネクションを開くことができません  
追加情報: 警告メッセージ:file(file, "rt") で:ファイル 'data.csv' を開くことができません: No such file or directory」  
のようなエラーがでます。いまはファイル名を「data.csv」と間違えていますので、「data1.csv」とファイル名を正しく書き直します。

read.table の header や sep など、オプションの指定に間違いがあると、そのオプションは無視されるかエラーになります。エラーメッセージを表示してくれれば分かりますが、無視されたり、エラーメッセージを表示することなく実行できない場合は、オプションに誤りがないか、自分で検討をつけ対応する必要があります。古い結果しか出力されない、オプションを書き換えても結果が変わらない、エラーは出ないのに実行結果に問題があるなどの場合は、オプションが正しく書かれているか確認してください。

フォルダ名、ファイル名は、Windowsであればエクスプローラで確認することができます。気をつけなければならないのは、エクスプローラでは最初、拡張子というものを表示していないことです。

.csvなどは拡張子と言って、そのファイルがどのような形式か、どのソフトに関連しているかを表す記号です。エクスプローラの上のほうにある「表示」をクリックして、「ファイル名拡張子」をクリックを入れて下さい。そうすると、拡張子がついたファイル名が表示されます。

拡張子がついたファイル名を表示したとき、ファイル名が「data1.csv.csv」などとなっていたら、ファイル名をクリックして「data1.csv.csv」のように書き換えてください。

## 添え字が許される範囲外です と表示される

たとえば、以下のようなエラーが出たとします。

```
> describeBy(d1$tekiou, list(d1$future), mat=TRUE, digits=2)
`[<-`(*tmp*, var, group + 1, value = dim.names[[group]][[groupi]]) でエラー:
添え字が許される範囲外です
```

このような場合は、指定したデータや変数に何らかの問題があります。たいていの場合、そういう名前の変数が無いことにより、このエラーが生じています。データ名や変数名が正しいか、よく確認してください。

他の原因として、データがすべてNA、群分け変数 (d1\$future) にNAが含まれるなどの場合もあります。変数 (d1\$tekiou や d1\$future) の中身を確認してください。

スク립トにエラーはなくても、同じスク립トを複数回実行するとエラーの原因をつくることもあります。スク립トを先頭から実行しなおして下さい。

## 作成した覚えのない変数が存在する

データファイルで、想定範囲外の列のセルに何らかのデータをRで読み込んだときそのような列に対してRは自動で X, X.1, X.2, X.3 のような変数名をつけます。また、データファイルに同じ変数名が複数あるときも、同様の番号づけを行います。

X, X.1, X.2, X.3 のような作成した覚えのない変数がR上でできているときは、重複している変数名があればそれらを修正してください。また、想定される範囲外のセルにデータがあると考えられるときは、範囲外の行や列について、ある程度の領域を指定して削除を実行し、不要なデータを証拠してください。

## incomplete final line found by readTableHeader と表示される

Rでread.table関数を実行した際、incomplete final line found by readTableHeader というエラーが出ることがあります。主にApple社製のパソコン（Mac）を使っている場合に出るメッセージです。

変数名やファイル名に全角文字があると、このメッセージが出ることがあります。変数名やファイル名を半角英数文字に変えてみてください。

それでもこのメッセージが出る場合は、「テキストエディット」などのエディタをまず起動し、そこから目的のCSVファイルを開いて、最終行の末尾に改行記号を入れて保存してください。詳しくは、本スクリプト集の「Excelを使わずにCSVファイルを作成する方法」の項を参照してください。

## 結果の数値に e-01 のような表記が付く

分析結果の出力が、例えば次のようになっているとします（重回帰分析の結果の一部です）。

Coefficients:

	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	4.497e-17	4.571e-02	0.000	1.0000
stress	4.509e-01	5.361e-02	8.411	3.61e-15 ***
kyoufu	1.210e-01	5.048e-02	2.397	0.0173 *
support	-3.508e-01	4.907e-02	-7.150	1.03e-11 ***

ここで、Estimate（推定値）の 4.497e-17, 4.509e-01 などの値はそれぞれ、  
 $4.497e-17 = 4.497 \times 10^{-17}$  乗 =  $0.000000000000000004497 \approx 0$   
 $4.509e-01 = 4.509 \times 10^{-1}$  乗 =  $0.4509 \approx 0.45$   
 を意味します。

e は指数 (exponential) のことで、そのうしろにくる数値が指数であることを表しています。もし、eのうしろが +03 となっていたら、10の3乗（×1000）ということになります。

4.497e-17 は実質的には0ですが、数学的には0ではありません。コンピュータは数値計算で解を求めるので、数学的に厳密に正しい値ではなく、このような値を出力することがあります。

## パッケージが開けない

パッケージはインストールできたが、library関数でそのパッケージを利用可能にしようとしても、開けないというトラブルがたまに発生します。いくつか原因が考えられますが、その1つとして、OSのアップデート（システムの更新）が未完了であることがあります。

Windowsであれば、スタートメニュー → 設定 → 更新とセキュリティ → 更新プログラムのチェック を行って、システムを最新の状態にしてください。おそらくパソコンの再起動が必要になります。OS（オペレーティングシステム）を最新の状態にして、Rを使うようにしてください。

## パッケージの関数が使えない

パッケージに含まれる関数を使おうとすると反応しない（そのような関数はみつからないと言われる）ことがあります。

まずは、当該のパッケージをきちんとインストールしているか、library関数でそのパッケージを使える状態にしているか、スクリプトに誤記はないかなどを確認してください。

パッケージの更新を行って下さい。Rでパッケージ → パッケージの更新 → パッケージを選択してOK を実行してください。

同じ関数名が他のパッケージにもあって、他のパッケージの関数が作動している場合があります。例えば、semという関数は、semパッケージにもlavaanにもあります。使わないほうのパッケージを取り外すことにより、使いたいパッケージの関数を使うことができます。以下のスクリプトを実行して取り外します。

```
detach("package:使わないパッケージ")
```

これらを行っていてもパッケージの関数が使えないことがあります。そのようなときの原因の1つとして、ユーザー名の問題が考えられます。全角文字のユーザー名だと、このようなエラーが生じることがあります。この場合は、半角英数文字（先頭はアルファベット）のユーザーを新たに作成してください。

コントロールパネル → ユーザーアカウント → 別のアカウントの管理 → PC設定で新しいユーザーを追加 → その他のユーザーをこのPCに追加、と進んで、半角英数ユーザー名で新しいアカウントを作成してください。管理者権限をもたせることをお勧めします。

アカウントができたなら、新しいユーザー名でサインインしてください。それで、パッケージも含めてRが正常に動くなら、今後Rを使うときはそのアカウントを使ってください。

新しいアカウントでもRがうまく動かないときは、いったんRをアンインストールしてください。

コントロールパネル → プログラム → プログラムのアンインストール、と進んで、Rを選んでクリックします。アンインストールするか聞いて来たら、指示に従ってアンインストールしてください。

そして、新しいアカウントを使って、Rおよびパッケージをインストールしてください。パッケージの更新、もやってください。それでパッケージも含めてRが正常に動くなら、今後Rを使うときはそのアカウントを使ってください。

## エラーが発生したときの対処

スクリプトを実行してエラーが発生するときは、ほとんどの場合、スクリプトに何らかの誤りがあります。また、Rが全然動かない（エラーすら出ない）ということも時々起こります。このような状態について、よくある誤りと対処法を説明します。なお、ある程度修正を試みてもうまくいかないときは、修正を繰り返すよりも、スクリプトをすべて新しく書き直したほうが早いです。

## 半角カナ文字を使っていませんか？

半角カナ文字は特殊な文字なので、Rのように世界中で使われるソフトには不向きです。データ、変数名、ファイル名、フォルダ名、ユーザ名等のすべてにおいて、使わないで下さい。

## お使いのPCのOSは何ですか？

本スクリプト集は主としてWindowsマシンを前提に書かれています。MacのパソコンでRを使用する場合には、作業ディレクトリを設定する方法や、read.table関数でCSVファイルを読み込む方法などが異なります。具体的な方法は、本スクリプト集の「Mac OSでRを使用するときの注意点」の項を参照してください。

## エクセルでデータを保存したCSVデータと、Rでの読み込み形式は合っていますか？

コンピュータが数値、文字、記号を認識するためには、文字コードというものがようになります。文字コードには、Shift JIS や UTF-8など、いくつか種類があります。

エクセルで「CSV（コンマ区切り）(\*.csv)」で保存した場合は、文字コードは Shift JIS になります。このデータをRで読み込むには、

```
d1 <- read.table("data1.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

とします。

エクセルで「CSV UTF-8（コンマ区切り）(\*.csv)」で保存した場合は、文字コードは UTF-8 になります。このデータをRで読み込むには、

```
d1 <- read.table("data1.csv", header=TRUE, sep=";", fileEncoding="UTF-8")
```

とします。fileEncoding オプションは不要です。Rは、ver. 4.2からUTF-8を標準採用するようになりました。

どちらもうまく行かない場合は、CSVファイルにBOMというものがついている可能性が考えられます。BOMは「バイトオーダーマーク」というもので、データの先頭に、文字コードがUTF-8であることを意味する記号が付加されます。このデータをRで読み込むには、

```
d1 <- read.table("data1.csv", header=TRUE, sep=";", fileEncoding="UTF-8-BOM")
```

とします。

本スクリプト集の「データの読み込み」の項も参照してください。

## 作業メモリ上に余計なデータが保存されていませんか？

Rでは、計算や分析の結果を作業メモリ上に「オブジェクト」として置いておき、再利用することができます。スクリプトにエラーがあるのに、同名の古いオブジェクトが残っていると、過去の結果が使われて、ミスに気づかないことがあります。また、新しく分析結果を保存したいのに、古い分析結果の後ろに新しい分析結果が追加されてしまうこともあります。

これらの事態を避けるために、分析を始めるときは、いったんメモリ上のオブジェクトを消去しておきます。メモリを消去するには、以下のスクリプトを実行します。

```
rm(list=ls())
```

rmはメモリ消去をする関数、ls()はメモリ上にあるすべてのオブジェクト名を返す関数です。

## 実行するスクリプトの範囲選択は正しいですか？

スクリプト全体を実行する場合は、スクリプト画面上でマウスを右クリックし、「全て選択」を選択→「カーソル行または選択中のRコードを実行」を選択、で実行できます。

選択した範囲のスクリプトだけを実行する場合は、マウスの左ボタンを押しながら範囲を選択して左ボタンを放し、スクリプト画面上でマウスを右クリックして「カーソル行または選択中のRコードを実行」を選択、で実行できます。

ここで、実行範囲を適切に選択しないと、エラーが生じます。行の途中から範囲指定する、最後のカッコを入れてないなどです。また、最後の改行を含んでいないため、最後の行が実行されないということもよくあります。範囲指定するときは、最後の行の次の行（空白行）まで範囲指定するようにします。

例)  
`setwd("D:¥¥Report¥¥")` を選択すべきところを `setwd("D:¥¥Report¥¥"`  
 # 末尾の ) と 改行 が含まれていない

## データのあるフォルダを正しく設定していますか？

データを読み込めないときによくあるエラーとして、`setwd()` の指定がうまくできていないことがあります。データの入っているドライブ名 (c, d, e など) と、フォルダ名 (Report など) を正しく指定して下さい。ドライブ名の後ろは : (コロン)、フォルダ名の後ろは ¥¥ (円マークを2つ) を入れるのを忘れないでください。カッコ内全体をダブルクォーテーションで囲うのも忘れないでください。なお、フォルダ名は、Windowsであればエクスプローラで確認することができます。

例)  
 データファイルが、USBメモリのReportというフォルダに入っていて、USBがDドライブの場合  
`setwd("D:¥¥Report¥¥")`      # 「D」は「d」でも構いません。

## データファイル名は正しいですか？

データを読み込めないときよくある別のエラーとして、`read.table()` でファイル名が正しく書かれていないことがあります。ファイル名が data1.csv のときに、データを読み込むスクリプトは次のようになります。  
`d1 <- read.table("data1.csv", header=TRUE, sep=",", fileEncoding="shift-jis")`

なお、ファイル名については、「良くある質問・エラー」の『データを読み込めない』の項も参照してください。

## 変数名を間違えていませんか？

データファイル (CSVファイル) で指定した変数名と、Rのスクリプトで書いている変数名が一致しないとエラーになります。変数名を確認し、正しい変数名を書いてください。

Rにおいて、`read.table()` など関数と呼ばれるものは綴りが決まっていますが、変数名やデータ名は自分で綴りを指定します。これらはなるべく英数半角文字で書くことをお勧めします。ただし、先頭に数字を使うことはできません。

## データファイルの変数名にスペースが入っていませんか？

データファイルの変数名に半角スペース、全角スペースがあるとき、Rでは「.」(半角ピリオド) に変換されます。あまり好ましくないのが、変数名にスペースを入れるべきではありません。

また、データファイルで変数名が無い場合 (予想外のセルにデータが入っている場合など)、Rは自動的に X, X.1, X.2, X.3 のような変数名をつけます。

例)  
 X 1 が X.1

## 全角記号が混ざっていませんか？

Rは全角文字と半角文字を区別します。しかし、スクリプト画面では、とくに英数字やスペースについて、全角文字と半角文字の見分けがつきにくくなっています。それゆえ、半角で書かなければならないところが全角になっているのに気がつかず、エラーとなることがしばしばあります。

例)  
`d1 <-` を `d1 <-`      # d1の後ろが全角スペースになっている

`<-` を `<-`                    # 代入記号 (半角`<-`) が全角になっている  
`x1` を `x 1`                    # 半角`x1` を全角`x 1` にしている。  
`d1[d1$class=="A",]` を `d1[d1$class=="A",]`                    # 最後の `]` が `」` になっている

### スペルミスをしていませんか？

スペルミスがあれば、エラーになるか、正しく動きません。

例)

```

x1 を x l                    # 1 (いち) と l (エル) を間違えている
colnames を colname        # sをつけ忘れている

```

### スクリプトを書くときに「>」「+」「1:」などを左端に書いていませんか？

スクリプトを実行すると、コンソール画面には、実行されたスクリプトが表示されます。その際、左端に「>」「+」「1:」などの記号が付加されます。コンソール画面の出力を参考にしてスクリプトを書く場合、左端のこれらの記号は、スクリプトには書かないようにする必要があります。

### 大文字と小文字を間違えていませんか？

変数名や関数名で大文字と小文字を間違えても、エラーになります。  
(ドライブ名、フォルダ名、ファイル名は、大文字小文字の区別はありません。)

例)

```

rowSums を rowsums
TRUE を True

```

### 記号を間違えていませんか？

\$マーク、論理式、数式、などの記号を間違えると、エラーや計算間違いとなります。

例)

```

d1[d1$class=="A",] を d1[d1%class=="A",]        # $ を % としている
d1[d1$sex=="f",] を d1[d1$sex="f",]            # == を = としている
<- を <    または    <=                    # 代入記号「<-」を、大小関係を表す「<」や「<=」としている

```

### カンマを忘れていませんか？

配列の行と列の区切り、変数や関数のオプションを複数指定するときなどは、要素間をカンマで区切りなければなりません。

例)

```

d1[d1$class=="A",] を d1[d1$class=="A"]
table(d1$x1, d1$x2) を table(d1$x1 d1$x2)
c(x1, x2, x3) を c(x1 x2 x3)

```

### カンマを多く入れていますか？

余計なカンマが入っていても、エラーになります。

例)

```

table(d1$class, d1$grade) を table(d1$class,, d1$grade)

```

### カンマとピリオドを間違えていませんか？

カンマを打つべきところをピリオドにすると、エラーになります。逆も同様です。



例)  
`ylim=c(1.5, 2.5)` を `ylim=c(1.5, 2.5)`      # 1.5 の後ろのカンマをピリオドにしている  
`ylim=c(1.5, 2.5)` を `ylim=c(1,5, 2.5)`      # 1.5 と書くべきところを 1 カンマ 5 としている

### 文字列をダブルクォーテーションで囲うのを忘れていませんか？

変数名、ファイル名、記号などの文字列は " " で囲います。片方または両方を付け忘れると、エラーになります。

例)  
`("D:¥¥Report¥¥")` を `("D:¥¥Rreport¥¥")`  
`"data1.csv"` を `data1.csv`  
`=="A"` を `==A`      # 「"A"」はAという文字、「A」はAという変数の意味になります。

### カッコをつけ忘れていませんか？

開くカッコに対して、必ず閉じるカッコがなければなりません。

例)  
`round(data.frame(n.d1, mean.d1),2)` を `round(data.frame(n.d1, mean.d1, 2)`  
# 開くカッコが2つあるのに、閉じるカッコが1つしかない

### カッコの種類を間違えていませんか？

Rはカッコの種類を区別します。開くカッコと閉じるカッコは、同じ種類でなければなりません。

例)  
`d1[, items]` を `d1[, items)`

### カッコで括らなければならないのを忘れていませんか？

if文の条件式はカッコで括らなければなりません。また、計算結果を代入値として使いたいときに、計算部分をカッコで括らなければならないことがあります。

開くカッコと閉じるカッコが同じ行にないとエラーになる場合があります。その場合は、長い文になっても、途中で改行せず1行のままにしておく必要があります。

例)  
`if (x==5)` を `if x==5`  
`xlim=c((xmin -.5), (xmax + .5))` を `xlim=c(xmin -.5, xmax + .5)`  
`for(i in 1:(n+3))` を `for(i in 1:n+3)`

### カッコをつける位置を間違えていませんか？

開くカッコと閉じるカッコの数があっても、つける位置が間違っていると、エラーになったり、意図とは異なる結果になったりします。

例)  
`floor(round(24.5)+0.5)` を `floor(round(24.5+0.5))`      # エラーになる  
`floor(round(24.5)+0.5)` を `floor(round(24.5+0.5))`      # 異なる結果になる

### オプションは正しく書かれていますか？

`header=TRUE` などオプションと言われるものに誤りがある場合、Rは関数によって、エラーを出すか、間違いのあるオプションを無視します。

エラーが表示される場合はそこで止まりますので、間違いを修正します。  
無視される場合は、何も言わずに以前の分析結果を表示することがありますので、オプションを変えても結果が変わらない場合は、オプション指定に誤りがあることに自分で気がつく必要があります。

Rのヘルプを使って、関数の書式やオプションの指定法などを調べるのも有効な方法です。ヘルプの使い方については、本スクリプト集の「ヘルプの使い方」の項を参照してください。なお、パッケージに含まれている関数は、そのパッケージをlibrary()で呼び出してからでないと、ヘルプをみることができません。

if - else文を書くとき、if 文の終わりと else 文の始まりの間で改行してませんか？

if - else文 では、else文はif文の終わりと同じ行に書かなくてはなりません。

例)

```
if (x==5) {y <- 1} else y <- 0 を
if (x==5) {y <- 1}
else y <- 0
```

1行に複数の命令文を書いていませんか？

プログラムを書くとき、1つの行に複数の命令文を続けて書くことはできません。コンピュータには、どこが文の切れ目か分からないからです。どうしても1行に複数の命令文を書きたいときは、セミコロン「;」で区切るようにします。

例)

```
head(d1)
nrow(d1)
を
head(d1) nrow(d1)          # head(d1); nrow(d1) ならOK
```

1つの数式を複数行に分けて書くとき、次の行の先頭が+ - \* / などの記号になっていませんか？

数式を途中で改行するとき、改行した次の行の先頭が + - \* / などになっているとエラーになることがあります。

パッケージは正しくインストールされていますか？

library()でパッケージを呼び出す際、パッケージがインストールされていないと呼び出すことはできません。パッケージを利用する際は、あらかじめインストールしておく必要があります。本スクリプト集の「パッケージのインストール」の項などを参照して、パッケージをインストールしてください。

パッケージのインストールは、Rをインストールした後に1回行えば、あとは何度でも利用できます。ただし、異なるバージョンのRをインストールした場合は、パッケージも改めてインストールする必要があります。

なお、パッケージをインストールしたはずなのに、library()でパッケージを開けないことがあります。これは、パッケージをインストールするとき、Rが自動的に指定するフォルダの不具合で、通常とは異なるディレクトリにパッケージがインストールされるためです。この場合は、Rのインストールからやり直すかと解決することが多いです。

R Studioを使っていますか？

R Studioを使っている場合、まれに原因不明のエラーが生じます。他にエラーの原因が思い当たらない場合は、R Studioは使わずに、R本体を起動してRを実行してください。

Rが複数個、起動していませんか？

Rが複数個起動していて、いま操作していない別のRが何かの待ち状態になっていると、いま操作しているRも動かないことがあります。

Rが複数個起動していることは、画面のタスクバーに複数個のRのタブがあることなどで確認できます。こ

れが原因でRが動かないときは、いま操作していない別のRの待ち状態を解消するか終了して下さい。

### コンピュータが再起動待ちの状態になっていませんか？

コンピュータを使っている間に、システムの自動アップデートなどが行われて、コンピュータの再起動待ち状態になっていると、Rが動かないことがあります。

この場合は、Rを含めすべてのプログラムを終了して、コンピュータを再起動してシステムのアップデートを行ってから、Rを再度実行してください。システムの再起動後、システムの更新版のチェックを行って、最新の状態にしてからRを実行することをお勧めします。

### 全角文字を使っていることが原因のこともあります

全角文字に対応していない関数を使う場合などは、半角だけでなく、全角文字もエラーの原因となります。スペルミスなど、他に思い当たる原因がない場合は、これが原因になっている可能性があります。データ、変数名等を半角英数文字にしてください。

### パソコンのユーザー名が全角文字になっていることが原因のこともあります

パソコンのユーザー名が全角文字だと、エラーが出る、または、動かない関数があります。この場合は、管理者権限のある半角英数文字のユーザーアカウントを作成し、そのアカウントでパソコンにサインインして、Rを使ってください。

### Rやコンピュータを再起動することも有効です

再起動待ちでもなく、原因は分からないけどとにかくRが動かなくなったという場合も、Rを終了して再起動、コンピュータを再起動などすることにより、動くことがあります。

### Rを再インストールすることも有効です

何をやってもエラーが出るという場合、Rを再インストールしたら解決したという事例も、パッケージを利用する場合などに、まれにあります。

### 3 データの作成・読み込み・保存

## Excelにおけるデータファイルの作成

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	id	x1	x2	x3	x4												
2	1	23	28	23	17												
3	2	18	22	23	14												
4	3	15	12	15	13												
5	4	12	16	22	25												
6	5	5	24	19	28												
7	6	14	16	15	16												
8	7	17	29	9	22												
9	8	17	11	25	6												
10	9	24	11	28	8												
11	10	14	14	18	10												
12	11	21	14	22	13												
13	12	21	18	13	9												
14	13	16	11	19	13												
15	14	19	26	18	21												
16	15	10	15	10	15												
17	16	12	23	16	32												
18	17	19	14	17	12												
19	18	10	25	21	29												
20	19	17	25	14	20												
21	20	20	16	16	15												
22	21	20	10	22	19												
23	22	21	10	15	23												
24	23	13	24	20	21												
25	24	17	20	18	16												
26	25	10	31	15	13												
27	26	23	14	29	6												
28	27	18	23	17	15												
29	28	9	24	21	17												
30	29	12	24	27	9												
31	30	20	8	21	8												
32	31	12	26	23	18												
33	32	10	23	18	20												
34	33	16	21	15	18												
35	34	22	32	5	27												
36	35	17	15	18	13												
37	36	15	14	25	5												
38	37	10	24	8	18												
39	38	21	4	32	6												
40	39	20	22	19	23												
41	40	22	15	15	6												
42	41	21	24	20	23												
43	42	14	17	24	18												

Excelなどを使ってデータを入力する。

`setwd` で指定する作業ディレクトリにデータファイルを置いておくのがわかりやすくよい。

1行目に変数名を入れておく。x1, x2のようなものでもよいし, stress, support, のように内容がわかるような変数名でもいい。

1列目にはidを入れておくと, 入力ミスの検索や質問紙との照合などに役立つ。id変数の変数名を大文字で「ID」としないほうが無難である。

和文フォントも使えないわけではないが, ときどきRで問題がおきるので, 英文フォントにしておいたほうが無難。(Rは世界中のボランティアが開発しているソフトなので, 言語のローカル性を考慮していない追加オプションが存在する可能性がある)

欠測の場合は空白セルにしておくといよい(Rで読み込むと自動的にNAとなるが, 最初からNAと入れるよりは, 空白にしておいたほうがよい)。

## Excelを使わずにCSVファイルを作成

Excelを使わずにCSVファイルを作成したい場合、たとえば、「メモ帳」や「テキストエディット」などを使ってデータをCSV保存する方法があります。

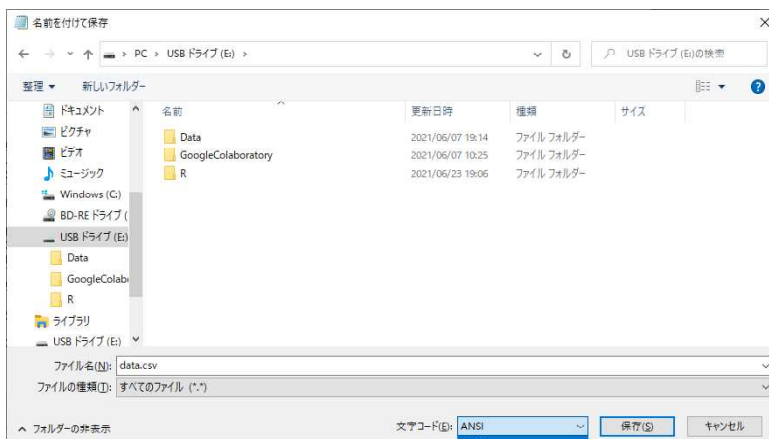
Windowsの場合、まずWindowsアクセサリからメモ帳を開き、データを「半角カンマ」で区切って入力します。Macの場合はテキストエディットを用います。

最終行まで入力したら、さらに改行し、カーソルを最終行の下に移動させます。

ファイル → 名前を付けて保存 → ファイルを保存するフォルダを選択 → ファイル名に「data1.csv」などを入力 → ファイルの種類を「すべてのファイル」に変更 → 文字コードを「ANSI」に変更 → 保存 とします。

これで、指定したフォルダにCSVデータファイルが保存されるはずです。

CSVは「comma separated value」の略なので、テキストエディタと言われるもの（メモ帳など）で、データを半角カンマで区切って入力し、ファイル名を「〇〇.csv」として保存すれば、Excelを使わなくても作成できます。



## ExcelにおけるデータファイルのCSV保存

データはCSV形式で保存しておくのが汎用性が高くよい。

ExcelでCSV保存する際、2通りの文字コードがある。どちらにも長短があるので、使いやすさなどを考えて、自分が良いと思うほうのCSVファイルで保存する。

【重要】本スクリプト集では、CSV（コンマ区切り）(\*.csv) で保存した Shift JIS コードのcsvファイル  
を前提として記載する。  
read.table, write.table において、fileEncoding="shift-jis" を付ける。

## CSV（コンマ区切り）(\*.csv)で保存した場合

文字コード：Shift JIS

Microsoft社が開発したコードで、日本国内のソフトであれば不自由を感じることはまずないが、外国製のソフトでは文字化けすることがある。

日本語環境のRで読み込むときに、encoding= というオプションを付ける必要がない。

Excelで保存した場合、「BOMなし」のcsvファイルになり、Rで読み込んだとき、第1変数の変数名に余計な記号が付くことはない。

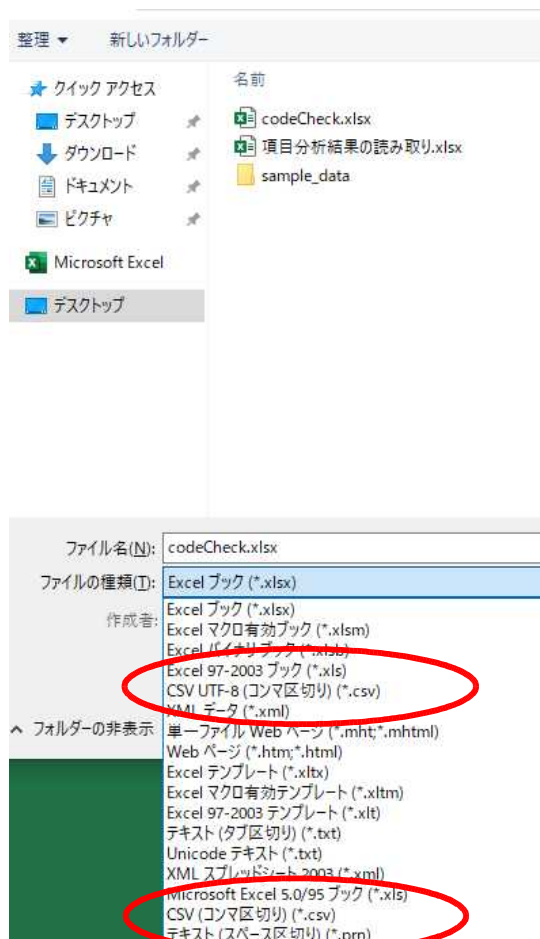
## CSV UTF-8(コンマ区切り) (\*.csv)で保存した場合

文字コード：UTF-8 コード

世界中で普及しているコードなので、多くのソフトが対応しており、外国製のソフトでも文字化けする可能性が低い。

日本語環境のRで読み込むときは、encoding="UTF-8" というオプションを付けなければならない。

Excelで保存した場合は「BOM付き」というcsvファイルになり、Rで読み込んだとき、第1変数の変数名に余計な記号が付く場合がある。元の変数名に戻すには、colnames 関数を使って変数名を修正する。



CSV UTF-8(コンマ区切り) (\*.csv)

CSV（コンマ区切り）(\*.csv)

## 作業ディレクトリの指定

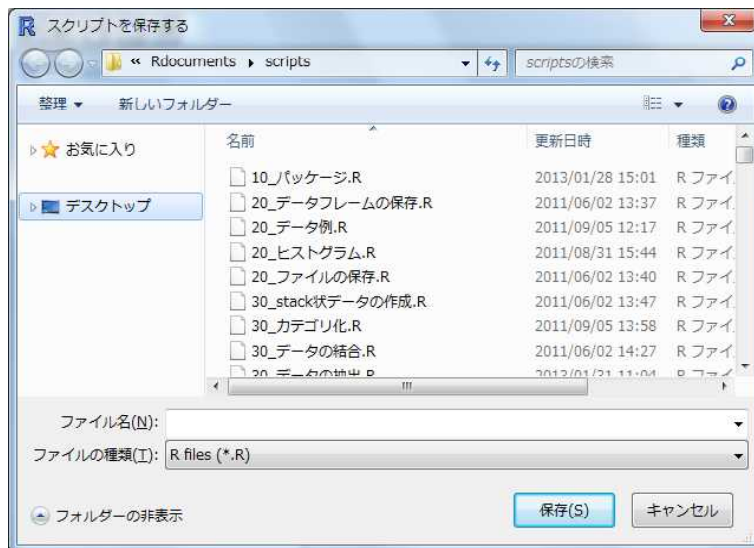
`setwd("ドライブ名¥¥フォルダ名¥¥…¥¥")`

「f:¥¥data¥¥」のように「¥」マーク（画面上では「\」になることも多い）を2つずつ入れて区切る。指定したディレクトリを「"」で囲む。

Windowsでは、¥¥の代わりに /（スラッシュ）を1つ書いて「f:/data/」とすることもできる。

Mac の場合は、ディレクトリの指定の仕方が異なるので注意（「Mac版のRで作業ディレクトリを設定する方法」参照）。

> #データファイルがあるフォルダの指定  
> `setwd("d:¥¥")`  
>



`setwd` を実行しておくと、「ファイル」→「(別名で) 保存」などとしたとき、`setwd`で指定されたディレクトリが最初から表示される。

データフレームの保存（後述）も`setwd`で作業ディレクトリを指定してあれば、その場所に保存される。

`setwd` を実行していないと、「ファイル」→「(別名で) 保存」などとしたとき、個人のドキュメントファイルディレクトリが表示される。ファイルを保存すべきドライブ¥¥フォルダ¥¥まで、いちいち移動しなければならない。



## データの読み込み

## CSV (コンマ区切り) (\*.csv) の場合 (Shift JIS コード)

```
データフレーム名 <- read.table("ファイル名.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

## CSV UTF-8(コンマ区切り) (\*.csv) の場合 (UTF-8コード)

```
データフレーム名 <- read.table("ファイル名.csv", header=TRUE, sep=";")
```

header=TRUE: 1行目に変数名があるという指定

sep=";": データはカンマ区切りのテキストファイル (つまりCSV形式) という指定

Shift-JISコードの場合は, fileEncoding="shift-jis" というオプションをつける.

**【重要】** 全角文字を使わなければ, fileEncodingオプションは不要であるが, ExcelがShift-JISコードである間は, とにかく fileEncoding="shift-jis" を付けておいたほうが無難。

ある列の値を行名に設定したいときは, row.names="変数名" または 列番号 とする.

変数名 (header) が 1セル左にずれたデータを読み込むと, 1列目のデータが自動的に行名になる.

Mac の場合は, ディレクトリの指定の仕方が異なるので注意 (「Mac OSでRを使用するときの注意点」参照)。

read.csv という関数や, Excelファイルを読み込む関数などもある.

```
> setwd("d:¥¥")
>
> # Shift JIS 形式のcsvファイルを読み込む
> d1 <- read.table("データ例2.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

```
> #最初の数行を表示
> head(d1)
```

```
  id x1 x2  x3 x4
1  1 23  a  心  ア
2  2 18  b  教  ア
3  3 15  a  医  ウ
4  4 12  c  看  ウ
5  5  5  b  看  イ
6  6 14  b  心  エ
```

	A	B	C	D	E	F
1	id	x1	x2	x3	x4	
2	1	23	a	心理	ア	
3	2	18	b	教育	ア	
4	3	15	a	医学	ウ	
5	4	12	c	看護	ウ	
6	5	5	b	看護	イ	
7	6	14	b	心理	エ	
8	7	17	a	医学	エ	
9	8	17	d	教育	ア	
10	9	24	b	医学	ア	
11	10	14	c	心理	イ	
12						

## &gt; #UTF-8形式のCSVファイルを読み込む

```
> d2 <- read.table("データ例UTF8.csv", header=TRUE, sep=";")
> head(d2)
```

```
  id x1 x2  x3 x4
1  1 23  a  心  ア
2  2 18  b  教  ア
3  3 15  a  医  ウ
4  4 12  c  看  ウ
5  5  5  b  看  イ
6  6 14  b  心  エ
```

> # 変数名が1セル左にずれたデータの読み込み

> d4 <- read.table("データ例\_rowあり.csv", header=TRUE, sep="," , fileEncoding="shift-jis")

> head(d4)

```

  x1 x2  x3 x4
1 23  a  心理 ア
2 18  b  教育 ア
3 15  a  医学 ウ
4 12  c  看護 ウ
5  5  b  看護 イ
6 14  b  心理 エ

```

1列目が行番号のデータ  
変数名が1セル左にずれている

	A	B	C	D	E	F
1	x1	x2	x3	x4		
2	1	23	a	心理	ア	
3	2	18	b	教育	ア	
4	3	15	a	医学	ウ	
5	4	12	c	看護	ウ	
6	5	5	b	看護	イ	
7	6	14	b	心理	エ	
8	7	17	a	医学	エ	
9	8	17	d	教育	ア	
10	9	24	b	医学	ア	
11	10	14	c	心理	イ	

> # write.table で row.names=TRUE にした場合, 1行目に変数名, 1列目に行名が保存されるが,

> # 行名に対する変数名は無いため, 変数名が1セル左にずれて保存される.

> # このようなデータを読み込んだ場合, 1行目にある変数名は自動的に右にずれて2列目以降の変数名

> # になる。(ズレが自動的に解消される)

> # ある列のデータをを行名にした読み込み

> p.all <- read.table("irt\_項目パラメタデータ.csv",  
> row.names="item", header=TRUE, sep="," , fileEncoding="shift-jis")

> p.all

```

      aj      bj  cj      # item 列の値が行名になっている
s03 0.7519081 -1.3479054 0
s04 0.6336668 -2.6447011 0
s05 1.2141256 -1.0870115 0
s06 1.4940753 -0.9263137 0
s07 1.7233574 -1.7290162 0
s08 0.9815908  3.1497698 0
s09 1.0866067 -0.3036178 0
s12 1.4498429 -0.1380151 0
s14 1.0159834 -0.2558882 0
s15 0.9046749 -0.4216555 0
s16 1.4752955  0.2216158 0

```

	A	B	C	D
1	item	aj	bj	cj
2	s03	0.752	-1.348	0
3	s04	0.634	-2.645	0
4	s05	1.214	-1.087	0
5	s06	1.494	-0.926	0
6	s07	1.723	-1.729	0
7	s08	0.982	3.150	0
8	s09	1.087	-0.304	0
9	s12	1.450	-0.138	0
10	s14	1.016	-0.256	0
11	s15	0.905	-0.422	0
12	s16	1.475	0.222	0

```

> p.all[1,1]
[1] 0.7519081

```

> # p.allというデータの(1,1)要素は0.75…であり, s03ではない. s03は1行目に付けられた行名

## 読み込んだデータの確認

head(データフレーム名)

データフレームの最初の6行を表示する.  
n=10 など行数を指定するオプションを付けることもできる.

```
> setwd("d:¥¥")
>
> d1 <- read.table("データ例2.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
```

&gt; # 最初の数行を表示

```
> head(d1)
  id x1 x2 x3 x4
1  1 23  a 心理 ア
2  2 18  b 教育 ア
3  3 15  a 医学 ウ
4  4 12  c 看護 ウ
5  5  5  b 看護 イ
6  6 14  b 心理 エ
```

&gt; # データの行数

```
> nrow(d1)
[1] 10
```

&gt; # データの列数

```
> ncol(d1)
[1] 5
```

&gt; # 変数名

```
> colnames(d1)
[1] "id" "x1" "x2" "x3" "x4"
>
```

	A	B	C	D	E	F
1	id	x1	x2	x3	x4	
2	1	23	a	心理	ア	
3	2	18	b	教育	ア	
4	3	15	a	医学	ウ	
5	4	12	c	看護	ウ	
6	5	5	b	看護	イ	
7	6	14	b	心理	エ	
8	7	17	a	医学	エ	
9	8	17	d	教育	ア	
10	9	24	b	医学	ア	
11	10	14	c	心理	イ	
12						

## R上のデータのCSV保存

## SHIFT JIS CSVファイルとして保存

```
write.table(データ名, "ファイル名.csv", row.names=FALSE, sep="," , fileEncoding="shift-jis")
```

row.names=FALSE にしておくと、行名は保存されない。

sep="," でデータをカンマ区切りのテキストファイル（つまりCSV形式）で出力する指定をしている。

fileEncoding="shift-jis" でShift-JISコードで保存される。Excelで文字化けせずに開ける。

setwd で作業ディレクトリを設定していない場合は、意に反したディレクトリに保存されることもある。

```
> setwd("d:¥¥")
>
> d1 <- read.table("データ例2.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
>
```

```
> head(d1)
  id x1 x2  x3 x4
1  1 23  a  心  ア
2  2 18  b  教  ア
3  3 15  a  医  ウ
4  4 12  c  看  ウ
5  5  5  b  看  イ
6  6 14  b  心  エ
```

	A	B	C	D	E	F
1	id	x1	x2	x3	x4	
2	1	23	a	心理	ア	
3	2	18	b	教育	ア	
4	3	15	a	医学	ウ	
5	4	12	c	看護	ウ	
6	5	5	b	看護	イ	
7	6	14	b	心理	エ	
8	7	17	a	医学	エ	
9	8	17	d	教育	ア	
10	9	24	b	医学	ア	
11	10	14	c	心理	イ	
12						

```
> #分析に必要な変数だけを取り出す
> d2 <- d1[,c("x1","x2","x3","x4")]
>
```

```
> head(d2)
```

```
  x1 x2  x3 x4
1 23  a  心  ア
2 18  b  教  ア
3 15  a  医  ウ
4 12  c  看  ウ
5  5  b  看  イ
6 14  b  心  エ
```

# id変数が削除されている。

# d2 <- d1[,c(-1)] としても同じ。c(-1)は「1列目は削除」を意味する

## &gt; # データの保存

```
> # row.names=FALSEの場合
```

```
> write.table(d2, "データ例2_rowなし.csv", row.names=FALSE, sep="," , fileEncoding="shift-jis")
```

```
> # row.names=TRUEの場合
```

```
> write.table(d2, "データ例2_rowあり.csv", row.names=TRUE, sep="," , fileEncoding="shift-jis")
```

## 新たに保存したファイル

行番号なし

変数名とデータの列が一致

	A	B	C	D
1	x1	x2	x3	x4
2	23	a	心理	ア
3	18	b	教育	ア
4	15	a	医学	ウ
5	12	c	看護	ウ
6	5	b	看護	イ
7	14	b	心理	エ
8	17	a	医学	エ
9	17	d	教育	ア
10	24	b	医学	ア
11	14	c	心理	イ

行番号あり

1 列目は行番号。変数名が 1 セル左にずれる。

	A	B	C	D	E
1	x1	x2	x3	x4	
2	1	23	a	心理	ア
3	2	18	b	教育	ア
4	3	15	a	医学	ウ
5	4	12	c	看護	ウ
6	5	5	b	看護	イ
7	6	14	b	心理	エ
8	7	17	a	医学	エ
9	8	17	d	教育	ア
10	9	24	b	医学	ア
11	10	14	c	心理	イ

## UTF-8 CSVファイルとして保存

```
write.table(データフレーム名, "ファイル名.csv", row.names=FALSE, sep=",")
```

```
> setwd("d:¥¥¥")
>
> d1 <- read.table("データ例2.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> head(d1)
  id x1 x2  x3 x4
1  1 23  a 心理 ア
2  2 18  b 教育 ア
3  3 15  a 医学 ウ
4  4 12  c 看護 ウ
5  5  5  b 看護 イ
6  6 14  b 心理 エ
>
> #分析に必要な変数だけを取り出す
> d2 <- d1[,c("x1", "x2", "x3", "x4")]
>
> head(d2)
```

	A	B	C	D	E	F
1	id	x1	x2	x3	x4	
2		1	23 a	心理	ア	
3		2	18 b	教育	ア	
4		3	15 a	医学	ウ	
5		4	12 c	看護	ウ	
6		5	5 b	看護	イ	
7		6	14 b	心理	エ	
8		7	17 a	医学	エ	
9		8	17 d	教育	ア	
10		9	24 b	医学	ア	
11		10	14 c	心理	イ	
12						

```
  x1 x2  x3 x4      # id変数が削除されている。
1 23  a 心理 ア      # d2 <- d1[,c(-1)] としても同じ。 c(-1)は「1列目は削除」を意味する
2 18  b 教育 ア
3 15  a 医学 ウ
4 12  c 看護 ウ
5  5  b 看護 イ
6 14  b 心理 エ
```

## &gt; # UTF-8 CSVファイルとして保存

```
> write.table(d2, "データ例2_UTF8.csv", row.names=FALSE, sep=",")
```

	A	B	C	D
1	x1	x2	x3	x4
2		23 a	蠢・炊	繚「
3		18 b	謨呵ご	繚「
4		15 a	蛹サ蟄ヲ	繚ヲ
5		12 c	達玖キ	繚ヲ
6		5 b	達玖キ	繚、
7		14 b	蠢・炊	繚イ
8		17 a	蛹サ蟄ヲ	繚イ
9		17 d	謨呵ご	繚「
10		24 b	蛹サ蟄ヲ	繚「
11		14 c	蠢・炊	繚、

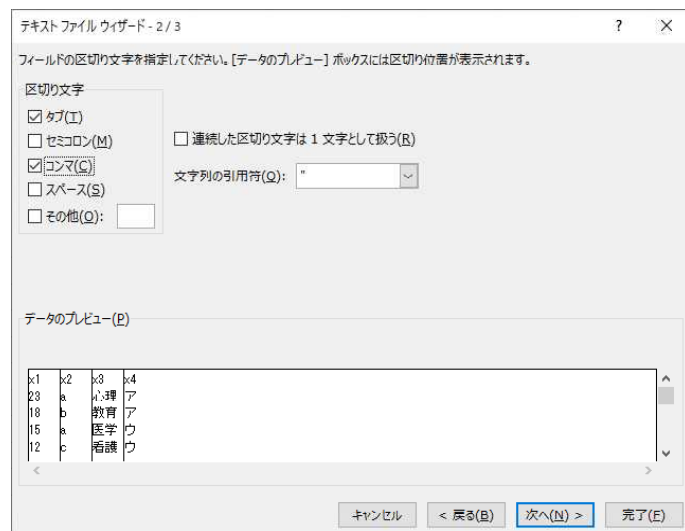
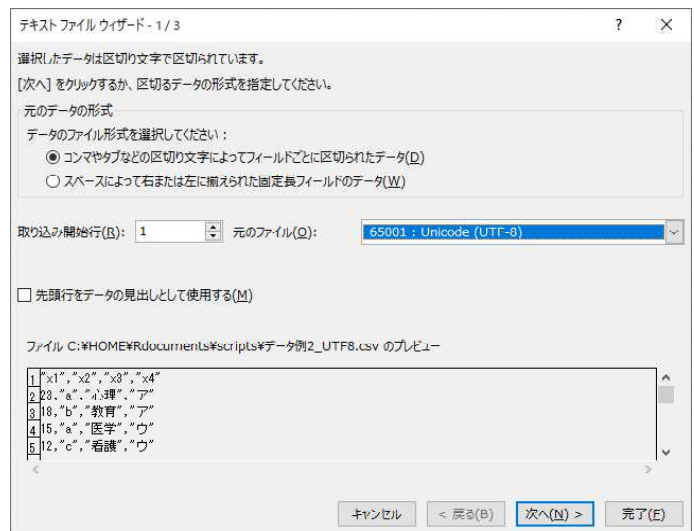
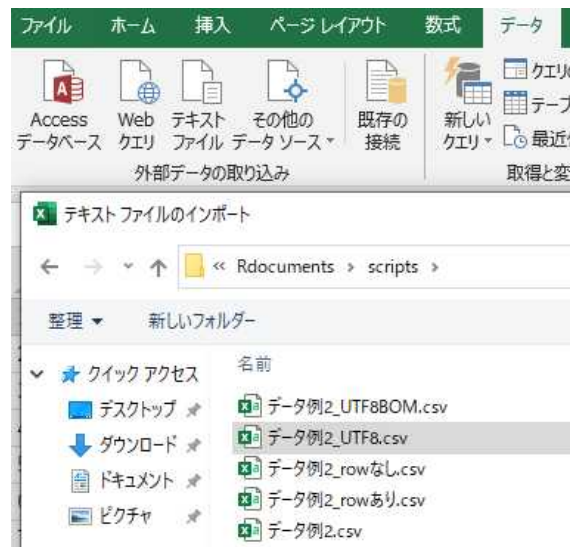
# Excelで開くと全角文字が文字化けする

## UTF-8 CSVファイルをExcelで読み込む方法

Excelを起動する

- 空白のブック
- データ
- テキストファイル
- 目的のファイルを選択して
- インポート
- 元のファイル」65001: Unicode (UTF-8)
- 次へ
- 「区切り文字」でコンマにチェック
- 次へ
- 完了
- OK

文字化けしないでデータが読み込まれる。これを保存すれば次からはExcelではきちんと読み込める



	A	B	C	D
1	x1	x2	x3	x4
2	23	a	心理	ア
3	18	b	教育	ア
4	15	a	医学	ウ
5	12	c	看護	ウ
6	5	b	看護	イ
7	14	b	心理	エ
8	17	a	医学	エ
9	17	d	教育	ア
10	24	b	医学	ア
11	14	c	心理	イ
12				

## Excel形式データファイルの読み込み・保存

Excel形式（xlsx）で保存したデータをCSVに変換せず、Excel形式のままRで読み込むことも可能である。

```
library("openxlsx")
```

```
データフレーム名 <- read.xlsx("Excelファイル名.xlsx")
```

## 【重要！】

あらかじめopenxlsxパッケージをインストールしておく必要がある。

CSV形式のファイルを読み込んだ場合、文字変数の空白セルは「空白」というデータとして認識される。

ExcelV形式のファイルを読み込んだ場合、文字変数の空白セルは欠測データとして認識される。

例えば、多枝選択における複数選択問題など、「,」（カンマ）を含むデータを読み込む場合は、CSV形式だと文字変数の欠測値の指定が必要になるが、Excel形式であれば、欠測値の指定をしなくても、空白セルを欠測値として読み込む。

## CSVデータ

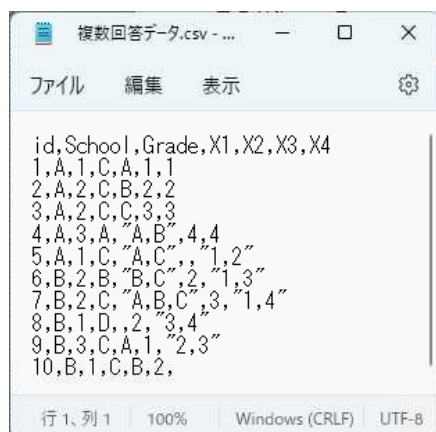
	A	B	C	D	E	F	G
1	id	School	Grade	X1	X2	X3	X4
2	1	A		1	C	A	1
3	2	A		2	C	B	2
4	3	A		2	C	C	3
5	4	A		3	A	A,B	4
6	5	A		1	C	A,C	1,2
7	6	B		2	B	B,C	2 1,3
8	7	B		2	C	A,B,C	3 1,4
9	8	B		1	D		2 3,4
10	9	B		3	C	A	1 2,3
11	10	B		1	C	B	2

## Excel (xlsx) データ

	A	B	C	D	E	F	G
1	id	School	Grade	X1	X2	X3	X4
2	1	A		1	C	A	1
3	2	A		2	C	B	2
4	3	A		2	C	C	3
5	4	A		3	A	A,B	4
6	5	A		1	C	A,C	1,2
7	6	B		2	B	B,C	2 1,3
8	7	B		2	C	A,B,C	3 1,4
9	8	B		1	D		2 3,4
10	9	B		3	C	A	1 2,3
11	10	B		1	C	B	2

Excelで、CSVデータとExcelデータをを開いた場合、見かけ上は同じである。

## CSVデータをメモ帳で開いた場合



CSVデータをメモ帳で開くと、「,」（カンマ）を含むデータは「"」で囲まれていることが分かる。



```
> rm(list=ls())
>
> setwd("d:¥¥Rreports¥¥")
>
```

### > # CSVデータの読み込み

```
> (d1 <- read.table("複数回答データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis"))
```

```
   id School Grade X1    X2 X3  X4
1    1      A     1  C    A  1   1
2    2      A     2  C    B  2   2
3    3      A     2  C    C  3   3
4    4      A     3  A   A, B 4   4
5    5      A     1  C   A, C NA 1, 2
6    6      B     2  B   B, C 2 1, 3
7    7      B     2  C A, B, C 3 1, 4
8    8      B     1  D        2 3, 4
9    9      B     3  C    A  1 2, 3
10  10      B     1  C    B  2
```

# 文字変数の空白は「空白」データ,  
# 数値変数の空白は欠測値になっている。

### > # パッケージの読み込み

```
> library("openxlsx")
```

### > # Excelデータの読み込み

```
> (d2 <- read.xlsx("複数回答データ.xlsx"))
```

```
   id School Grade X1    X2 X3  X4
1    1      A     1  C    A  1   1
2    2      A     2  C    B  2   2
3    3      A     2  C    C  3   3
4    4      A     3  A   A, B 4   4
5    5      A     1  C   A, C NA 1, 2
6    6      B     2  B   B, C 2 1, 3
7    7      B     2  C A, B, C 3 1, 4
8    8      B     1  D  <NA> 2 3, 4
9    9      B     3  C    A  1 2, 3
10  10      B     1  C    B  2 <NA>
```

# 文字変数も数値変数も、空白は欠測値になっている

### > # 文字変数における空白セルの扱いの確認

```
> table(d1$X2, useNA="ifany")
```

```
      A  A, B A, B, C  A, C  B  B, C  C
1      2      1      1      1  2      1  1
```

```
> table(d2$X2, useNA="ifany")
```

```
      A  A, B A, B, C  A, C  B  B, C  C  <NA>
2      1      1      1      2  1      1  1
```

### > # CSVファイル保存

```
> write.table(d1, "複数回答データ_R.csv", row.names=FALSE, sep=",", fileEncoding="shift-jis")
```

### > # Excelファイル保存

```
> write.xlsx(d2, "複数回答データ_R.xlsx")
```



## 複数回答データ\_R.csv

	A	B	C	D	E	F	G
1	id	School	Grade	X1	X2	X3	X4
2		1 A		1 C	A		1
3		2 A		2 C	B		2
4		3 A		2 C	C		3
5		4 A		3 A	A,B		4
6		5 A		1 C	A,C	NA	1,2
7		6 B		2 B	B,C		2 1,3
8		7 B		2 C	A,B,C		3 1,4
9		8 B		1 D			2 3,4
10		9 B		3 C	A		1 2,3
11		10 B		1 C	B		2

CSV形式で保存した場合、文字変数の空白は「空白」データ、数値変数の欠測は「NA」という文字列として保存される。

## 複数回答データ\_R.xlsx

	A	B	C	D	E	F	G
1	id	School	Grade	X1	X2	X3	X4
2		1 A		1 C	A		1 1
3		2 A		2 C	B		2 2
4		3 A		2 C	C		3 3
5		4 A		3 A	A,B		4 4
6		5 A		1 C	A,C		1,2
7		6 B		2 B	B,C		2 1,3
8		7 B		2 C	A,B,C		3 1,4
9		8 B		1 D			2 3,4
10		9 B		3 C	A		1 2,3
11		10 B		1 C	B		2

Excel形式で保存した場合、文字変数および数値変数の欠測は「空白」データとして保存される。

## 4 データ操作

## 欠測値の取り扱い

CSVデータにおいて、欠測値は空白セルにしておく。それをRで読み込むと自動的にNA (Not Available) となる。

## 文字列データにおける空白を欠測値とする

```
na.strings=""
```

データ読み込み時、read.tableのオプションに na.strings="" と書いて設定する。  
そうしないと、空白も1つのカテゴリとして認識してしまう。

```
> d1 <- read.table("欠測値文字データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

```
> head(d1)
```

```
  番号 性別  学科 x1 x2 x3
1    1    看護学  3  2  4
2    2  男    看護学  3  2  3
3    3  女    医学 NA  3  1
4    4  女    看護学  3  4  1
5    5  男    医学  1  3  3
6    6  女    心理学  2  1  3
```

	A	B	C	D	E	F
1	番号	性別	学科	x1	x2	x3
2	1		看護学	3	2	4
3	2	男		3	2	3
4	3	女	医学		3	1
5	4	女	看護学	3	4	1
6	5	男	医学	1	3	3
7	6	女	心理学	2	1	3

```
> d1$性別 <- as.factor(d1$性別)
```

```
> levels(d1$性別)
```

```
[1] "" "女" "男" # na.strings="" がないので、空白も1つのカテゴリとして認識されている
```

```
>
```

```
>
```

```
> d1 <- read.table("欠測値文字データ.csv", header=TRUE, sep=";", na.strings="",  
fileEncoding="shift-jis")
```

```
> head(d1)
```

```
  番号 性別  学科 x1 x2 x3
1    1 <NA> 看護学  3  2  4
2    2  男 <NA>  3  2  3
3    3  女    医学 NA  3  1
4    4  女    看護学  3  4  1
5    5  男    医学  1  3  3
6    6  女    心理学  2  1  3
```

```
> d1$性別 <- as.factor(d1$性別)
```

```
> levels(d1$性別)
```

```
[1] "女" "男" # na.strings="" があるので、空白は欠測値として認識されている
```

## データの各要素が欠測値か否かを確認する

```
is.na(データ)
```

当該要素が欠測値ならTRUE、欠測値でなければFALSEという値を返す。

データには、データフレーム、ベクトル、及び、それらの要素を指定することができる。

## if 文、ifelse文での欠測値の扱い

```
if( is.na(データ) != TRUE ){ if(条件式) }
```

```
ifelse( is.na(データ), ..., ...)
```

データに欠測値があるとき、「データ==…」の評価結果は、TRUEでもFALSEでもなくNAとなり、

if文やifelse文がエラーとなる。

例えば、不等式  $x1 < 1$  を評価するとき、 $x1$ がNAの場合、 $x1 < 1$  はTRUEになってしまうことがある。

そこで、is.na(データ)でNAかどうかを評価して、TRUEまたはFALSEを返して、先に進む。

## どの行に欠測値があるかを確認する complete.cases(データフレーム名)

欠測値がない行はTRUE, 欠測値がある行はFALSEという値を返す.

## 欠測値があることを許さない関数を使う場合 データフレーム名 <- na.omit(データフレーム名)

欠測値のある行を取り除いたデータフレームを作成して関数に入れる.

## 欠測値の除外をオプションで指定できる関数の場合

関数(..., na.rm=TRUE)

関数(..., use="pairwise.complete.obs")

na.rm=TRUE とすると, 当該変数において, 欠測値のある行は除外して計算する. すべての関数で必要な訳でもないが (自動的に欠測値のある行を除外する関数もある), na.rm が効かない関数もある.

cov や cor 関数では, use オプションで指定する.

use: "everything"

当該2変数に欠測値がある場合, その箇所だけNAとなる

"complete.obs"

1つでも欠測値のある行を除外してすべての値を計算

"pairwise.complete.obs"

当該2変数に欠測値がある場合, その箇所だけ欠測値を除外して計算

```
> setwd("d:¥¥¥")
> d1 <- read.table("欠測データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> d1
```

```
  x1 x2 x3 x4
1  NA  3  2  3
2   3  3  2  3
3   3  3  2  1
4   3  3  2  2
5   3  1  2  3
6   2  3  3  3
7   2  3  2  3
8   3  2  2  3
9   3  3  2  2
10  2  2  2  1
11  3  3  2  2
12  2  3  2  1
13  1  3  2  3
14  2  3  2  1
15 NA NA  2  2    # 1行目のx1, 15行目のx1, x2に欠測値がある
>
>
```

	x1	x2	x3	x4
1				
2		3	2	3
3	3	3	2	3
4	3	3	2	1
5	3	3	2	2
6	3	1	2	3
7	2	3	3	3
8	2	3	2	3
9	3	2	2	3
10	3	3	2	2
11	2	2	2	1
12	3	3	2	2
13	2	3	2	1
14	1	3	2	3
15	2	3	2	1
16			2	2
17				

```
> # 各要素欠測値かどうかの確認
```

```
> is.na(d1)
      x1      x2      x3      x4
[1,] TRUE FALSE FALSE FALSE
[2,] FALSE FALSE FALSE FALSE
[3,] FALSE FALSE FALSE FALSE
[4,] FALSE FALSE FALSE FALSE
[5,] FALSE FALSE FALSE FALSE
[6,] FALSE FALSE FALSE FALSE
```

# 1行目のx1, 20行目のx1, x2に欠測値があることがわかる

```
[7,] FALSE FALSE FALSE FALSE
[8,] FALSE FALSE FALSE FALSE
[9,] FALSE FALSE FALSE FALSE
[10,] FALSE FALSE FALSE FALSE
[11,] FALSE FALSE FALSE FALSE
[12,] FALSE FALSE FALSE FALSE
[13,] FALSE FALSE FALSE FALSE
[14,] FALSE FALSE FALSE FALSE
[15,] TRUE TRUE FALSE FALSE
>
>
```

> #欠測値のある行の確認

> complete.cases(d1)

```
[1] FALSE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE TRUE
[13] TRUE TRUE FALSE
```

# 1行目と20行目がFALSEになっており、欠測値があることがわかる

>

> # 欠測値のある行を削除したデータフレームの作成

> d2 <- na.omit(d1)

> d2

```
  x1 x2 x3 x4
2   3  3  2  3
3   3  3  2  1
4   3  3  2  2
5   3  1  2  3
6   2  3  3  3
7   2  3  2  3
8   3  2  2  3
9   3  3  2  2
10  2  2  2  1
11  3  3  2  2
12  2  3  2  1
13  1  3  2  3
14  2  3  2  1
```

# d1から1行目と15行目が削除されている

>

>

> # 平均値

> #欠測値を除外しない場合

> round(colMeans(d1), 2)

```
  x1  x2  x3  x4
NA   NA 2.07 2.20
```

# 欠測値のある変数の平均は計算されない

>

> #当該変数において、欠測値を除外した場合

> round(colMeans(d1, na.rm=TRUE), 2)

```
  x1  x2  x3  x4
2.46 2.71 2.07 2.20
```

# 変数ごとに、欠測値を除外して平均値を計算

>

> # 欠測値のある行を削除した場合

> round(colMeans(d2), 2)

```
  x1  x2  x3  x4
2.46 2.69 2.08 2.15
```

# 1つでも欠測値のある行を削除してしまったので、  
# x3, x4の平均値も変わってしまう

>

>

```
> # 相関係数行列
> # 欠測値を除外しない場合
> round(cor(d1), 2)
```

	x1	x2	x3	x4
x1	1	NA	NA	NA
x2	NA	1	NA	NA
x3	NA	NA	1.00	0.26
x4	NA	NA	0.26	1.00

```
>
```

# 欠測値のある変数の相関係数は計算されない

```
> # 当該変数において、欠測値を除外した場合
> round(cor(d1, use="pairwise.complete.obs"), 2)
```

	x1	x2	x3	x4
x1	1.00	-0.23	-0.21	0.01
x2	-0.23	1.00	0.13	-0.16
x3	-0.21	0.13	1.00	0.26
x4	0.01	-0.16	0.26	1.00

```
>
```

# 変数ペアごとに欠測値を除外して相関係数を計算

```
> # 欠測値のある行を削除した場合
> round(cor(d1, use="complete.obs"), 2)
```

	x1	x2	x3	x4
x1	1.00	-0.23	-0.21	0.01
x2	-0.23	1.00	0.15	-0.20
x3	-0.21	0.15	1.00	0.28
x4	0.01	-0.20	0.28	1.00

```
>
```

# 1つでも欠測値のある行を削除してしまうので、  
# 他の相関係数の値が変わってしまう

```
> # または
> round(cor(d2), 2)
```

	x1	x2	x3	x4
x1	1.00	-0.23	-0.21	0.01
x2	-0.23	1.00	0.15	-0.20
x3	-0.21	0.15	1.00	0.28
x4	0.01	-0.20	0.28	1.00

```
> # 度数表, クロス表
```

```
> # 欠測値を無視する
> table(d1$x1, d1$x4)
```

```
  1 2 3
1 0 0 1
2 3 0 2
3 1 3 3
```

```
>
> # 欠測を常に1つのカテゴリとして扱う
> table(d1$x1, d1$x4, useNA="always")
```

```
  1 2 3 <NA>
1  0 0 1  0
2  3 0 2  0
3  1 3 3  0
<NA> 0 1 1  0
```

```
>
> # 欠測がある場合のみ1つのカテゴリとして扱う
> table(d1$x1, d1$x4, useNA="ifany")
```

```
  1 2 3
1  0 0 1
2  3 0 2
3  1 3 3
<NA> 0 1 1
```

## データ値に基づいたカテゴリ変数の生成 — 離散データ

各データ値に、新しくカテゴリ値を対応させる

```
新変数名 <- factor(元変数名, levels=c(データ値), labels=c(カテゴリ値))
```

levelsの各データ値に対して、labelsでカテゴリ値を付与していく。  
levelsの異なるデータ値に対して、同じlabels値を与えることも可能。

```
> setwd("d:¥¥")
```

```
> d1 <- read.table("統計図表データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> head(d1)
```

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

```
> # 性別データからsexデータを作成
```

```
> table(d1$性別)
```

```
女 男
143 127
```

```
> d1$sex <- factor(d1$性別, levels=c("男","女"), labels=c("M", "F"))
```

```
> table(d1$性別, d1$sex, useNA="ifany")
```

	M	F
女	0	143
男	127	0

```
> # 学科データからareaデータを作成
```

```
> table(d1$学科)
```

```
医学 看護学 心理学
76 100 94
```

```
>
> d1$area <- factor(d1$学科, levels=c("医学","看護学","心理学"),
+ labels=c("medical","comedical","comedical"))
```

```
> table(d1$学科, d1$area, useNA="ifany")
```

	medical	comedical
医学	76	0
看護学	0	100
心理学	0	94

```
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

## データ値に基づいたカテゴリ変数の生成 — 連続データ

連続変数データをいくつかの階級に分割して、各階級にカテゴリ値を対応させる

新変数名 <- cut(元変数名,breaks=c(分割点),right=FALSE, labels=c(カテゴリ名), ordered\_result=TRUE)

分割点は、 $-\text{Inf}$ , ...,  $\text{Inf}$  で指定する。

right=FALSE とすると、分割点の右端の値は含まない。TRUE (初期設定) とすると、右端の値を含む。

分割点で分割される区間分のカテゴリ名を指定する ( $-\text{Inf}$ ,  $\text{Inf}$  を含んだ分割点の個数 - 1)。

```
> setwd("d:¥¥¥")
> d1 <- read.table("合計得点_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id x1 x2 x3 x4 x5 x6 x7 x8
1  1  3  3  2  3  1  2  3  1
2  2  3  3  2  3  2  3  3  2
3  3  3  3  2  1  1  1  1  1
4  4  3  3  2  2  1  2  3  1
5  5  3  3  3  3  3  3  3  3
6  6  3  3  2  3  2  2  2  2
>
```

	A	B	C	D	E	F	G	H	I	J
1	id	x1	x2	x3	x4	x5	x6	x7	x8	goukei
2	1	3	3	2	3	1	2	3	1	18
3	2	3	3	2	3	2	3	3	2	21
4	3	3	3	2	1	1	1	1	1	13
5	4	3	3	2	2	1	2	3	1	17
6	5	3	3	3	3	3	3	3	3	24
7	6	3	3	2	3	2	2	2	2	19
8	7	2	3	3	3	3	2	3	3	22
9	8	2	3	2	3	2	2	3	3	20
10	9	3	2	2	3	1	2	2	2	17
11	10	3	3	3	3	1	2	2	3	20
12	11	3	3	2	2	2	2	3	3	20
13	12	2	2	2	1	2	2	1	2	14
14	13	3	3	2	3	2	3	3	3	22
15	14	3	3	2	2	2	2	3	3	20
16	15	2	3	2	1	1	1	2	1	13
17	16	3	3	2	3	1	3	3	3	21
18	17	3	3	2	3	1	2	3	2	19
19	18	3	3	3	3	2	2	1	2	19
20	19	2	3	2	1	1	1	1	2	13
21	20	2	3	2	2	2	3	2	3	19

```
> # 変数リスト名
> list.goukei <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8")

> # 合計得点の計算
> d1$goukei <- rowSums(d1[, list.goukei])
>

> # 合計得点を3群に分けるカテゴリデータを生成
> d1$gun <- cut(d1$goukei, breaks=c(-Inf, 14, 20, Inf), labels=c("L", "M", "H"),
+               ordered_result=TRUE)

> #対応表の表示
> table(d1$gun, d1$goukei)
```

```
      9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24
L    3  5  5  7 14 13  0  0  0  0  0  0  0  0  0  0
M    0  0  0  0  0  0 11 20 30 26 19 29  0  0  0  0
H    0  0  0  0  0  0  0  0  0  0  0  0 16 22 16 12
```

breaks=c(-Inf, 14, 20, Inf), labels=c("L", "M", "H")でやっていること

goukei得点に基づいて、

```
-∞ < goukei <= 14   なら gun = "L"
14  < goukei <= 20   なら gun = "M"
20  < goukei <= +∞   なら gun = "H"
```

のようにgun変数を作成しています。table(d1\$gun, d1\$goukei) で、群分けが正しく行われているか、gun と goukei の関係をみて確認しています。

ordered\_result=TRUE は、表などを書くとき、カテゴリの順番を L, M, H の順で表示させるための指定です。



## カテゴリ変数の再カテゴリ化 — 水準の合併

## 元のカテゴリの確認

levels(変数名)

元の変数はfactor型である必要がある。もしfactor型でなければ、変数名 <- as.factor(変数名) として、factor型に変換しておく。

## 再カテゴリ化

levels(変数名) &lt;- c(元のカテゴリに対応させる新しいカテゴリの並び)

元のカテゴリの並び順に、新しいカテゴリを割り当てていく。

再カテゴリ化する変数は、元の変数に上書きしてもよいし、新しい変数を作成してもよい。

```
> setwd("d:¥¥¥")
> d1 <- read.table("カテゴリ化_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  seibetsu age
1         0 29
2         0 28
3         1 30
4         1 30
5         1 33
6         1 29
>
```

```
> #データのカテゴリ化
> d1$age2 <- cut(d1$age, ,right=FALSE,
+               breaks=c(-Inf, 10, 20, 30, 40, 50, Inf),
+               labels=c("0_9", "10_19", "20_29", "30_39", "40_49", "50_"))
>
```

## &gt; #カテゴリ化の結果

```
> table(d1$age2)

0_9 10_19 20_29 30_39 40_49 50_
  0     30    132     97     11     0
```

## &gt; #再カテゴリ化

```
> # 元の変数のカテゴリ確認
> levels(d1$age2)
[1] "0_9" "10_19" "20_29" "30_39" "40_49" "50_"
```

## &gt; # 新しい変数の作成

```
> d1$age3 <- d1$age2
```

## &gt; # 新しい変数のカテゴリを再カテゴリ化

```
> levels(d1$age3) <- c("nonage", "nonage", "adult", "adult", "adult", "adult")
>
```

## &gt; # 再カテゴリ化の結果

```
> table(d1$age2, d1$age3)
```

```
      nonage adult
0_9         0     0
10_19        30     0
20_29         0    132
30_39         0     97
40_49         0     11
50_          0      0
```

```
>
```

	A	B
1	seibetsu	age
2	0	29
3	0	28
4	1	30
5	1	30
6	1	33
7	1	29
8	0	24
9	1	35
10	1	38
11	0	27
12	1	21
13	1	38
14	1	41
15	0	30
16	1	30
17	0	30
18	1	34
19	1	27
20	0	20

## 不要な水準の削除・必要な水準の追加

as.vector(変数名) または as.factor(as.vector(変数名))

```
> setwd("d:¥¥")
> d1 <- read.table("カテゴリ化_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  seibetsu age
1         0 29
2         0 28
3         1 30
4         1 30
5         1 33
6         1 29
```

	A	B
1	seibetsu	age
2	0	29
3	0	28
4	1	30
5	1	30
6	1	33
7	1	29
8	0	24
9	1	35
10	1	38
11	0	27
12	1	21
13	1	38
14	1	41
15	0	30
16	1	30
17	0	30
18	1	34
19	1	27
20	0	20

```
> # データのカテゴリ化
> d1$age2 <- cut(d1$age, , right=FALSE, breaks=c(-Inf, 10, 20, 30, 40, 50, Inf),
+               labels=c("0_9", "10_19", "20_29", "30_39", "40_49", "50_"))
```

```
> # 新しいカテゴリ変数の作成
> d1$age3 <- d1$age2
> levels(d1$age3) <- c("nonage", "nonage", "adult", "adult", "adult", "adult")
```

```
> # 再カテゴリ化の結果
> table(d1$age2, d1$age3)
```

```
      nonage adult
0_9         0     0
10_19       30     0
20_29        0   132
30_39        0    97
40_49        0    11
50_          0     0
```

```
> # adult だけのデータ
> d2 <- d1[d1$age3=="adult",]
```

```
> # adult だけのデータのage2の度数分布
> table(d2$age2)
```

```
0_9 10_19 20_29 30_39 40_49 50_
0    0    132    97    11    0
```

# adult だけのデータであるから、0\_9 および  
# 10\_19 という水準はもはや不要であるが、  
# もとのデータの水準名を引き継いでいるため、  
# これらの水準が残っている

```
> # 不要な水準の削除
> # 一度ベクトル型にしてから、(必要があれば) factorに戻す
> d2$age2a <- as.factor(as.vector(d2$age2))
> table(d2$age2a)
```

```
20_29 30_39 40_49      # 0_9 と 10_19, さらに度数が0だった 50_ という水準が無くなっている
132    97    11
```

```
> # 必要な水準の追加
> # 50_ というカテゴリが必要なら、levels に50_を追加する
> d2$age2b <- factor(d2$age2a, levels=c("20_29", "30_39", "40_49", "50_"))
> table(d2$age2b)
```

```
20_29 30_39 40_49 50_      # 50_というデータは無いが、度数 0 として認識されている
132    97    11    0
```

## 回答データの採点

```
採点変数 <- ifelse(正答の条件式, 点数, 誤答の点数)
```

最初に無回答の処理をする

誤答のところに、準正答の式を入れることも可能

正答の条件式は、複数の条件を & (and) や | (or) で設定することも可能

```
> setwd("d:¥¥")
> d1 <- read.table("採点データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d1
```

```
  id class sex x1 x2 x3 x4 x5
1   1    a   f  4  1  1  3  2
2   2    b   m  1  1  2  3  4
3   3    a   m  2  3  4  3  4
4   4    a   m  4  2  3  2  1
5   5    b   f  3  3  2  4  3
6   6    b   f  1  4  2  4  2
7   7    b   f NA  3  1  1  4
8   8    a   m  1  2  4  2  3
9   9    a   f NA NA NA NA NA
10  10    b   m  3  4  3  1  4
```

id	class	sex	x1	x2	x3	x4	x5
1	a	f	4	1	1	3	2
2	b	m	1	1	2	3	4
3	a	m	2	3	4	3	4
4	a	m	4	2	3	2	1
5	b	f	3	3	2	4	3
6	b	f	1	4	2	4	2
7	b	f	NA	3	1	1	4
8	a	m	1	2	4	2	3
9	a	f	NA	NA	NA	NA	NA
10	b	m	3	4	3	1	4

```
> # 採点
```

```
> # スコア変数を s1などとして作成
```

```
>
```

```
> # x1: 1 が正答
```

```
> d1$s1 <- ifelse(is.na(d1$x1), 0, ifelse(d1$x1==1, 1, 0))
```

```
>
```

```
> # x2: 1 が正答, 2 が準正答
```

```
> d1$s2 <- ifelse(is.na(d1$x2), 0, ifelse(d1$x2==1, 1, ifelse(d1$x2==2, 0.5, 0)))
```

```
>
```

```
> # x3: 1 または 4 が正答
```

```
> d1$s3 <- ifelse(is.na(d1$x3), 0, ifelse((d1$x3==1 | d1$x3==4), 1, 0))
```

```
>
```

```
> # x4, x5: x4 が 3, かつ, x5 が 4 が正答
```

```
> d1$x4_5 <- paste(d1$x4, d1$x5, sep="")
```

```
> d1$s4_5 <- ifelse(d1$x4_5 == "34", 1, 0)
```

```
>
```

```
> # 採点結果の確認
```

```
> table(d1$x1, d1$s1, useNA="always")
```

```
      0 1 <NA>
1      0 3    0
2      1 0    0
3      2 0    0
4      2 0    0
<NA> 2 0    0
```

```
>
```

```
> table(d1$x2, d1$s2, useNA="always")
```

```
      0 0.5 1 <NA>
1      0  0 2    0
2      0  2 0    0
3      3  0 0    0
4      2  0 0    0
<NA> 1  0 0    0
```

```
>
```

```
> table(d1$x3, d1$s3, useNA="always")
```

```
      0 1 <NA>
1      0 2    0
2      3 0    0
3      2 0    0
4      0 2    0
<NA> 1 0    0
```

```
>
```

```
> table(d1$x4_5, d1$s4_5, useNA="always")
```

```
      0 1 <NA>
14     2 0    0
21     1 0    0
23     1 0    0
32     1 0    0
34     0 2    0
42     1 0    0
43     1 0    0
NANA   1 0    0
<NA>   0 0    0
```

```
>
```

## 逆転項目のデータ処理

## 新しい変数を作る場合

新しい変数名 <- カテゴリ最小値 + カテゴリ最大値 - 元の変数名

## 上書きする場合

変数名 <- カテゴリ最小値 + カテゴリ最大値 - 変数名

1～5の5段階評価なら、1+5-変数名 = 6-変数名

0～4の4段階評価なら、0+4-変数名 = 4-変数名 など

```
> rm(list=ls())
```

```
> setwd("d:¥¥")
```

```
> d1 <- read.table("逆転項目_データ_欠測あり.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

```
> head(d1)
```

```
  id x1 x2 x3          # x2は逆転項目である
```

```
1  1  3 NA  5
2  2  3  3  3
3  3  5  3  4
4  4  4  1  5
5  5  5  1  4
6  6  3  2  1
```

```
>
```

```
>
```

```
> # 項目間相関係数
```

```
> cor(d1[,c("x1", "x2", "x3")], use="pairwise.complete.obs")
```

```
          x1          x2          x3
x1  1.0000000 -0.5975349  0.6863536
x2 -0.5975349  1.0000000 -0.7695711
x3  0.6863536 -0.7695711  1.0000000
```

```
>
```

```
>
```

```
> #カテゴリの上限, 下限
```

```
> minx <- 1
```

```
> maxx <- 5
```

```
>
```

```
>
```

```
> # 項目得点の逆転した変数の作成
```

```
> d1$x2r <- ifelse(is.na(d1$x2), NA, (minx + maxx - d1$x2))
```

```
>
```

```
>
```

```
> # 逆転の確認
```

```
> # 1→5, 2→4, ..., 5→1
```

```
> table(d1$x2, d1$x2r, useNA="ifany", dnn=c("x2", "x2r"))
```

```
      x2r
x2     1 2 3 4 5 <NA>
1      0 0 0 0 3    0
2      0 0 0 4 0    0
3      0 0 4 0 0    0
4      0 5 0 0 0    0
5      2 0 0 0 0    0
<NA> 0 0 0 0 0    2
```

```
>
```

```
>
```

```
> # データフレームの確認
```

```
> head(d1)
```

```
  id x1 x2 x3 x2r
1  1  3 NA  5  NA
2  2  3  3  3   3
3  3  5  3  4   3
4  4  4  1  5   5
5  5  5  1  4   5
6  6  3  2  1   4
```

```
>
```

```
>
```

```
> # データフレームの保存
```

```
> write.table(d1, "逆転済み_データ.csv", sep=";", row.names=FALSE, fileEncoding="shift-jis")
```

```
>
```

	A	B	C	D
1	id	x1	x2	x3
2	1	3		5
3	2	3	3	3
4	3	5	3	4
5	4	4	1	5
6	5	5	1	4
7	6	3	2	1
8	7	2		2
9	8	2	1	4
10	9	1	2	3
11	10	4	3	3
12	11	1	4	1
13	12	3	2	5
14	13	2	5	1
15	14	1	4	2

## 合計得点・平均得点の計算

## 合計得点

```
変数リスト名 <- c("変数名1", "変数名2", ..., "変数名p")
合計得点変数 <- rowSums(データフレーム名[, 変数リスト名])
```

## 平均得点（合計得点／項目数）

```
変数リスト名 <- c("変数名1", "変数名2", ..., "変数名p")
平均得点変数 <- rowMeans(データフレーム名[, 変数リスト名], na.rm=TRUE)
```

```
> rm(list=ls())

> setwd("d:¥¥")

>
> d1 <- read.table("逆転項目_データ_欠測あり.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  id x1 x2 x3          # x2は逆転項目である
1  1  3 NA  5
2  2  3  3  3
3  3  5  3  4
4  4  4  1  5
5  5  5  1  4
6  6  3  2  1
>
>
> #カテゴリの上限, 下限
> minx <- 1
> maxx <- 5
>
>
> # 項目得点の逆転した変数の作成
> d1$x2r <- ifelse(is.na(d1$x2), NA, (minx + maxx - d1$x2))
>
>
> # 得点を合計する変数の指定
> items <- c("x1", "x2r", "x3")
>
>
> # 合計得点の計算
> d1$xsum <- rowSums(d1[, items])
>
>
> #平均得点の計算
> d1$xmean <- rowMeans(d1[, items], na.rm=TRUE)
>
>
> # 結果の確認
> head(d1)
  id x1 x2 x3 x2r xsum  xmean
1  1  3 NA  5  NA  NA 4.000000
2  2  3  3  3   3   9 3.000000
3  3  5  3  4   3  12 4.000000
4  4  4  1  5   5  14 4.666667
5  5  5  1  4   5  14 4.666667
6  6  3  2  1   4   8 2.666667

# 合計得点は、欠測データを除外していないので、
# 欠測のある回答者の合計得点は、欠測になる。

# 平均得点は、欠測データを除外して平均を求めて
# いるので、欠測のある回答者の平均得点は、欠測
# していない項目得点の平均となる。

> # データフレームの保存
> write.table(d1, "合計得点_結果.csv", sep="," , row.names=FALSE, fileEncoding="shift-jis")
```

	A	B	C	D
1	id	x1	x2	x3
2	1	3		5
3	2	3	3	3
4	3	5	3	4
5	4	4	1	5
6	5	5	1	4
7	6	3	2	1
8	7	2		2
9	8	2	1	4
10	9	1	2	3
11	10	4	3	3
12	11	1	4	1
13	12	3	2	5
14	13	2	5	1
15	14	1	4	2

## データの標準化・中心化・偏差値化

scale(データフレーム名)

特に指定しなければ、変数ごとに、平均=0、標準偏差=1 に標準化  
 center=c(値1, 値2, ...) を指定すると、平均値が(値1, 値2, ...) だけずれる。  
 scale=FALSE とすると、標準偏差はもとのままとなる。

```
> setwd("d:¥¥¥")
> d1 <- read.table("データ例.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  id x1 x2 x3 x4
1  1 23 28 23 17
2  2 18 22 23 14
3  3 15 12 15 13
4  4 12 16 22 25
5  5  5 24 13 28
6  6 14 16 15 16
>
>
>
>
> # もとのデータの記述統計量
> dtmp <- d1[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cov(dtmp)
> cnames <- paste("c", colnames(ctmp), sep="")
> rtmp <- cor(dtmp)
> rnames <- paste("r", colnames(rtmp), sep="")
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp, rtmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", cnames, rnames)
> ktmp
   N Mean  SD  cx1  cx2  cx3  cx4  rx1  rx2  rx3  rx4
x1 245 15.22 5.37 28.80 -3.38 11.69 -9.70 1.00 -0.10 0.43 -0.26
x2 245 20.32 6.08 -3.38 37.00 -10.06 23.80 -0.10 1.00 -0.32 0.56
x3 245 18.52 5.12 11.69 -10.06 26.18 -16.35 0.43 -0.32 1.00 -0.46
x4 245 16.61 6.98 -9.70 23.80 -16.35 48.66 -0.26 0.56 -0.46 1.00
>
>
>
```

	A	B	C	D	E
1	id	x1	x2	x3	x4
2	1	23	28	23	17
3	2	18	22	23	14
4	3	15	12	15	13
5	4	12	16	22	25
6	5	5	24	13	28
7	6	14	16	15	16
8	7	17	29	8	22
9	8	17	11	25	6
10	9	24	11	28	8
11	10	14	14	18	10
12	11	21	14	22	13
13	12	21	18	13	9
14	13	16	11	19	13
15	14	19	26	18	21
16	15	10	15	10	15
17	16	12	23	16	32
18	17	19	14	17	12
19	18	10	25	21	29
20	19	17	25	14	20
21	20	20	16	16	15

```
> # データの標準化
> d2 <- scale(d1[, c(-1)])
>
> # 標準化したデータの記述統計量
> dtmp <- d2
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cov(dtmp)
> cnames <- paste("c", colnames(ctmp), sep="")
> rtmp <- cor(dtmp)
> rnames <- paste("r", colnames(rtmp), sep="")
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp, rtmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", cnames, rnames)
> ktmp
   N Mean SD  cx1  cx2  cx3  cx4  rx1  rx2  rx3  rx4
x1 245  0  1 1.00 -0.10 0.43 -0.26 1.00 -0.10 0.43 -0.26
x2 245  0  1 -0.10 1.00 -0.32 0.56 -0.10 1.00 -0.32 0.56
x3 245  0  1 0.43 -0.32 1.00 -0.46 0.43 -0.32 1.00 -0.46
x4 245  0  1 -0.26 0.56 -0.46 1.00 -0.26 0.56 -0.46 1.00
>
```

```

>
>
> #データの中心化
> d3 <- scale(d1[, c(-1)], scale=F)
>

> # 中心化したデータの記述統計量
> dtmp <- d3
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cov(dtmp)
> cnames <- paste("c", colnames(ctmp), sep="")
> rtmp <- cor(dtmp)
> rnames <- paste("r", colnames(rtmp), sep="")
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp, rtmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", cnames, rnames)
> ktmp
  N Mean  SD  cx1  cx2  cx3  cx4  rx1  rx2  rx3  rx4
x1 245   0 5.37 28.80 -3.38 11.69 -9.70  1.00 -0.10  0.43 -0.26
x2 245   0 6.08 -3.38 37.00 -10.06 23.80 -0.10  1.00 -0.32  0.56
x3 245   0 5.12 11.69 -10.06 26.18 -16.35  0.43 -0.32  1.00 -0.46
x4 245   0 6.98 -9.70 23.80 -16.35 48.66 -0.26  0.56 -0.46  1.00
>
>

> #データの偏差値化
> d4 <- scale(d1[, c(-1)])*10 + 50
>

> # 偏差値化したデータの記述統計量
> dtmp <- d4
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cov(dtmp)
> cnames <- paste("c", colnames(ctmp), sep="")
> rtmp <- cor(dtmp)
> rnames <- paste("r", colnames(rtmp), sep="")
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp, rtmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", cnames, rnames)
> ktmp
  N Mean SD  cx1  cx2  cx3  cx4  rx1  rx2  rx3  rx4
x1 245  50 10 100.00 -10.34 42.55 -25.92  1.00 -0.10  0.43 -0.26
x2 245  50 10 -10.34 100.00 -32.32 56.08 -0.10  1.00 -0.32  0.56
x3 245  50 10 42.55 -32.32 100.00 -45.80  0.43 -0.32  1.00 -0.46
x4 245  50 10 -25.92 56.08 -45.80 100.00 -0.26  0.56 -0.46  1.00
>
>
>

```



## 行名・列名（変数名）・要素名の指定

## 行名を指定

```
rownames(データフレーム名) <- c(変数名の並び)
```

## 列名（変数名）を指定

```
colnames(データフレーム名) <- c(変数名の並び)
```

## 特定列の名前を指定

```
colnames(データフレーム名)[当該変数の番号] <- "新しい変数名"
```

colnamesは2列以上からなるデータフレームには有効だが、ベクトルには無効である。  
ベクトルの場合はベクトル名が変数名である。

## 要素名の指定

```
names(ベクトル名) <- c(変数名の並び)
```

```
> setwd("d:¥¥¥")
> d1 <- read.table("制御_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d1
```

```
  x1 x2 x3
1  NA  3  2
2   3  3  2
3   1  3  2
4   3  3  2
5   3  3  3
6   3  3  2
7   2  3  3
8   2  3  2
9   3  3  2
10  2  3  2
11  1  3  2
12  3  3  2
13  1  3  3
14  2  3  2
15 NA NA  2
```

	A	B	C
1	x1	x2	x3
2		3	2
3	3	3	2
4	1	3	2
5	3	3	2
6	3	3	3
7	3	3	2
8	2	3	3
9	2	3	2
10	3	3	2
11	2	3	2
12	1	3	2
13	3	3	2
14	1	3	3
15	2	3	2
16			2
17			

```
> # d1の行数
> (nr <- nrow(d1))
[1] 15
> (nc <- ncol(d1))
[1] 3
>
>
```

```
> # d1の変数名
> (cnames1 <- colnames(d1))
[1] "x1" "x2" "x3"
>
>
```

## &gt; # d1の行名列名を指定

```
> rownames(d1) <- c("a", "b", "c", "d", "e", "f", "g", "h", "i", "j",
+                  "k", "l", "m", "n", "o")
> colnames(d1) <- c("c1", "c2", "c3")
> d1
  c1 c2 c3
a NA  3  2
b  3  3  2
c  1  3  2
d  3  3  2
e  3  3  3
f  3  3  2
g  2  3  3
h  2  3  2
```

```

i  3  3  2
j  2  3  2
k  1  3  2
l  3  3  2
m  1  3  3
n  2  3  2
o NA NA  2
>
>

```

```

> # 3番目の変数名をe3に変更
> colnames(d1)[3] <- "e3"
> head(d1)
  c1 c2 e3
a NA  3  2
b  3  3  2
c  1  3  2
d  3  3  2
e  3  3  3
f  3  3  2

```

```

> # ベクトルに要素名を指定
> (r1 <- c(1, 2, 3))
[1] 1 2 3

> names(r1) <- c("v1", "v2", "v3")
> r1
v1 v2 v3
1  2  3
>

```

## 行や列の抽出・削除

## 行操作

## 連続した指定行のみ抽出

データフレーム名[開始行:終了行, ]

## 行番号で指定した行のみ抽出

データフレーム名[c(行番号1, 行番号2…), ]

## 行番号で指定した行のみ削除

データフレーム名[c(-行番号1, -行番号2…), ]

## 数値変数の値が、ある値である行のみ抽出

データフレーム名[データフレーム名\$変数名==値, ]

## 数値変数の値が、ある値である行のみ削除

データフレーム名[データフレーム名\$変数名!=値, ]

## 文字変数の値が、ある値である行のみ抽出

データフレーム名[grepl("値", データフレーム名\$変数名, fixed=FALSE), ]

## 文字変数の値が、ある値である行のみ削除

データフレーム名[grepl("値", データフレーム名\$変数名, fixed=FALSE)==FALSE, ]

## subset関数またはsubsetオプションを使う方法

subset(データフレーム名, 条件式)

条件にあう行を抽出

## 列操作

## 連続した指定列のみ抽出

データフレーム名[, 開始列:終了列]

## 列番号で指定した列のみ抽出

データフレーム名[, c(列番号1, 列番号2…)]

## 列番号で指定した列のみ削除

データフレーム名[, c(-列番号1, -列番号2…)]

## 変数名で指定した列のみ抽出

データフレーム名[, c("変数名1", "変数名2"…)]

## 変数名で指定した列のみ削除

データフレーム名[, (colnames(データフレーム名) %in% c("変数名1", "変数名2"…))==FALSE]

&gt; setwd("d:¥¥¥")

&gt; d1 &lt;- read.table("データの抽出\_データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")

&gt; d1

	id	sex	x1	x2	y
1	1	f	1	2	6
2	2	f	3	3	4
3	3	m	5	3	7
4	4	m	2	1	5
5	5	f	2	5	6
6	6	m	3	2	5
7	7	f	4	5	9
8	8	m	2	1	6
9	9		1	3	5
10	10	f	4	4	7

```
> # 指定行のみ抽出
```

```
> (d2 <- d1[1:2,])
  id sex x1 x2 y
1  1  f  1  2 6
2  2  f  3  3 4
```

```
> (d2 <- d1[c(1,3,5),])
```

```
  id sex x1 x2 y
1  1  f  1  2 6
3  3  m  5  3 7
5  5  f  2  5 6
```

```
> # subset関数を使う方法
```

```
> (d2 <- subset(d1, d1$x1==1))
```

```
  id sex x1 x2 y
1  1  f  1  2 6
9  9     1  3 5
```

```
> (d2 <- subset(d1, grepl("f", d1$sex, fixed=FALSE)))
```

```
  id sex x1 x2 y
1  1  f  1  2 6
2  2  f  3  3 4
5  5  f  2  5 6
7  7  f  4  5 9
10 10  f  4  4 7
```

```
> # 指定行のみ削除
```

```
> (d2 <- d1[c(-1,-3,-5),])
```

```
  id sex x1 x2 y
2  2  f  3  3 4
4  4  m  2  1 5
6  6  m  3  2 5
7  7  f  4  5 9
8  8  m  2  1 6
9  9     1  3 5
10 10  f  4  4 7
```

```
> # subset関数を使う方法
```

```
> (d2 <- subset(d1, d1$x1!=1))
```

```
  id sex x1 x2 y
2  2  f  3  3 4
3  3  m  5  3 7
4  4  m  2  1 5
5  5  f  2  5 6
6  6  m  3  2 5
7  7  f  4  5 9
8  8  m  2  1 6
10 10  f  4  4 7
```

```
> (d2 <- subset(d1, grepl("f", d1$sex, fixed=FALSE)==FALSE))
```

```
  id sex x1 x2 y
3  3  m  5  3 7
4  4  m  2  1 5
6  6  m  3  2 5
8  8  m  2  1 6
9  9     1  3 5
```

	A	B	C	D	E
1	id	sex	x1	x2	y
2	1	f	1	2	6
3	2	f	3	3	4
4	3	m	5	3	7
5	4	m	2	1	5
6	5	f	2	5	6
7	6	m	3	2	5
8	7	f	4	5	9
9	8	m	2	1	6
10	9		1	3	5
11	10	f	4	4	7
12					

> # 変数の値が、ある値である行のみを抽出

```
> (d2 <- d1[d1$x1==1,])           # d1 の x1 の値が 1 のとき TRUE, それ以外るとき FALSE で、
  id sex x1 x2 y                 # TRUE の行だけが残る
1  1  f  1  2  6
9  9    1  3  5
```

```
> (d2 <- d1[grepl("f", d1$sex, fixed=FALSE),])
  id sex x1 x2 y
1  1  f  1  2  6
2  2  f  3  3  4
5  5  f  2  5  6
7  7  f  4  5  9
10 10  f  4  4  7           # d1 の sex に f が含まれるとき TRUE, それ以外るとき FALSE で、
                           # TRUE の行だけが残る
```

> # 変数の値が、ある値である行のみを削除

```
> (d2 <- d1[d1$x1!=1,])
  id sex x1 x2 y
2  2  f  3  3  4
3  3  m  5  3  7
4  4  m  2  1  5
5  5  f  2  5  6
6  6  m  3  2  5
7  7  f  4  5  9
8  8  m  2  1  6
10 10  f  4  4  7

> (d2 <- d1[grepl("f", d1$sex, fixed=FALSE)==FALSE,])
  id sex x1 x2 y
3  3  m  5  3  7
4  4  m  2  1  5
6  6  m  3  2  5
8  8  m  2  1  6
9  9    1  3  5
```

> # 指定列のみ抽出

```
> (d2 <- d1[, c(3, 5)])
  x1 y
1  1  6
2  3  4
3  5  7
4  2  5
5  2  6
6  3  5
7  4  9
8  2  6
9  1  5
10 4  7

> (d2 <- d1[, c("x1", "y")])
  x1 y
1  1  6
2  3  4
3  5  7
4  2  5
5  2  6
6  3  5
7  4  9
8  2  6
9  1  5
10 4  7
```

> # 指定列のみ削除

> (d2 <- d1[,c(-3,-5)])

	id	sex	x2
1	1	f	2
2	2	f	3
3	3	m	3
4	4	m	1
5	5	f	5
6	6	m	2
7	7	f	5
8	8	m	1
9	9		3
10	10	f	4

> (d2 <- d1[, (colnames(d1) %in% c("x1", "y"))==FALSE])

	id	sex	x2
1	1	f	2
2	2	f	3
3	3	m	3
4	4	m	1
5	5	f	5
6	6	m	2
7	7	f	5
8	8	m	1
9	9		3
10	10	f	4

>

## データの並べ替え

データフレーム名[order(データフレーム名\$並べ替え変数名1, データフレーム名\$並べ替え変数名2, ...),]

decreasing=TRUE とすると降順に並べ替える

変数名の前に「- (マイナス)」を付けても、降順にできる。1つの変数で降順、別の変数で昇順にしたい場合などに便利。

na.last = TRUE とするとNAは最後尾. = FALSE とするとNAは先頭. = NA とするとNAは削除

```
> setwd("d:¥¥")
> d1 <- read.table("採点データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")

> d1
  id class sex x1 x2 x3 x4 x5
1   1     a   f  4  1  1  3  2
2   2     b   m  1  1  2  3  4
3   3     a   m  2  3  4  3  4
4   4     a   m  4  2  3  2  1
5   5     b   f  3  3  2  4  3
6   6     b   f  1  4  2  4  2
7   7     b   f NA  3  1  1  4
8   8     a   m  1  2  4  2  3
9   9     a   f NA NA NA NA NA
10 10     b   m  3  4  3  1  4
>
> # class で並べ替え
> (d2 <- d1[order(d1$class),])
  id class sex x1 x2 x3 x4 x5
1   1     a   f  4  1  1  3  2
3   3     a   m  2  3  4  3  4
4   4     a   m  4  2  3  2  1
8   8     a   m  1  2  4  2  3
9   9     a   f NA NA NA NA NA
2   2     b   m  1  1  2  3  4
5   5     b   f  3  3  2  4  3
6   6     b   f  1  4  2  4  2
7   7     b   f NA  3  1  1  4
10 10     b   m  3  4  3  1  4
>
>
> # class で降順に並べ替え
> (d2 <- d1[order(d1$class, decreasing=TRUE),])
  id class sex x1 x2 x3 x4 x5
2   2     b   m  1  1  2  3  4
5   5     b   f  3  3  2  4  3
6   6     b   f  1  4  2  4  2
7   7     b   f NA  3  1  1  4
10 10     b   m  3  4  3  1  4
1   1     a   f  4  1  1  3  2
3   3     a   m  2  3  4  3  4
4   4     a   m  4  2  3  2  1
8   8     a   m  1  2  4  2  3
9   9     a   f NA NA NA NA NA
>
>
> # class で昇順, sex 降順 で並べ替え
> (d2 <- d1[order(d1$class, -as.numeric(as.factor(d1$sex))),])
  id class sex x1 x2 x3 x4 x5
3   3     a   m  2  3  4  3  4
4   4     a   m  4  2  3  2  1
8   8     a   m  1  2  4  2  3
1   1     a   f  4  1  1  3  2
9   9     a   f NA NA NA NA NA
2   2     b   m  1  1  2  3  4
10 10     b   m  3  4  3  1  4
5   5     b   f  3  3  2  4  3
6   6     b   f  1  4  2  4  2
7   7     b   f NA  3  1  1  4
```

# 文字変数に「-」は効かないので、  
# factor化したうえで、  
# 数値に換えて評価している。  
# fが1, mが2になっている。

## データの結合

## データを縦に繋げる

```
rbind(データ名1, データ名2)
```

データ名1とデータ名2の列変数が対応していなければならない。

## データを横に繋げる

## 各行に同じオブザベーションのデータがある場合

```
data.frame(データ名1, データ名2)
```

```
cbind(データ名1, データ名2)
```

`data.frame` を使うと同じ変数名には添え字が付け加えられる。データフレームが作られる。  
`cbind` を使うと同じ変数名はそのままにされる。データフレーム or 行列 が作られる。

## 各行に同じオブザベーションのデータがない場合

```
merge(データ名1, データ名2, by="変数名", all=FALSE/TRUE)
```

`all=TRUE` とすると、`by`で指定された変数が非対応のデータも結合する。`all=` を指定しないか `all=FALSE` とすると、`by`で指定された変数が対応するデータのみを結合する。

データ1とデータ2で、対応させる変数の名前が異なる場合は、`by. x="データ1での変数名"`、`by. y="データ2での変数名"`と指定する。

```
> setwd("d:¥¥")
>
> d.a1 <- read.table("データ結合A1.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> d.a2 <- read.table("データ結合A2.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

```
> d.a1
  id sex age x1 x2
1  1  M  28  A 17
2  2  F  22  B 14
3  3  M  24  B 13
4  4  M  25  A 25
5  5  M  24  A 28
```

```
> d.a2
  id sex age x1 x2
1  6  F  18  A 16
2  7  F  29  A 22
3  8  M  25  C  6
4  9  M  23  D  8
5 10  F  19  A 10
>
```

## &gt; #データフレームを縦に繋げる

```
> da <- rbind(d.a1, d.a2)
```

```
> da
  id sex age x1 x2
1  1  M  28  A 17
2  2  F  22  B 14
3  3  M  24  B 13
4  4  M  25  A 25
5  5  M  24  A 28
6  6  F  18  A 16
7  7  F  29  A 22
8  8  M  25  C  6
9  9  M  23  D  8
10 10  F  19  A 10
>
>
```

データ結合A1.csv

A	B	C	D	E
id	sex	age	x1	x2
1	M	28	A	17
2	F	22	B	14
3	M	24	B	13
4	M	25	A	25
5	M	24	A	28

データ結合A2.csv

A	B	C	D	E
id	sex	age	x1	x2
6	F	18	A	16
7	F	29	A	22
8	M	25	C	6
9	M	23	D	8
10	F	19	A	10



```
> d.b1 <- read.table("データ結合B1.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> d.b2 <- read.table("データ結合B2.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> d.b3 <- read.table("データ結合B3.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

```
> d.b1
  id sex age
1  1  M  28
2  2  F  22
3  3  M  24
4  4  M  25
5  5  M  24
6  6  F  18
7  7  F  29
8  8  M  25
9  9  M  23
10 10  F  19
```

データ結合B1.csv

	A	B	C
id	sex	age	
1	M		28
2	F		22
3	M		24
4	M		25
5	M		24
6	F		18
7	F		29
8	M		25
9	M		23
10	F		19

```
> d.b2
  id x1 x2
1  1  A 17
2  2  B 14
3  3  B 13
4  4  A 25
5  5  A 28
6  6  A 16
7  7  A 22
8  8  C  6
9  9  D  8
10 10  A 10
```

データ結合B2.csv

	A	B	C
id	x1	x2	
1	A		17
2	B		14
3	B		13
4	A		25
5	A		28
6	A		16
7	A		22
8	C		6
9	D		8
10	A		10

```
> d.b3
  id x1 x2
1  1  A 17
2  3  B 13
3  4  A 25
4  7  A 22
5 10  A 10
>
```

```
> #データフレームを横に繋げる
> #横に繋げた結果がデータフレーム
> d.bd <- data.frame(d.b1, d.b2)
> d.bd
```

```
  id sex age id.1 x1 x2
1  1  M  28   1  A 17
2  2  F  22   2  B 14
3  3  M  24   3  B 13
4  4  M  25   4  A 25
5  5  M  24   5  A 28
6  6  F  18   6  A 16
7  7  F  29   7  A 22
8  8  M  25   8  C  6
9  9  M  23   9  D  8
10 10  F  19  10  A 10
>
```

```
> #横に繋げた結果がデータフレームまたは行列
> d.bc2 <- cbind(d.b1, d.b2)
> d.bc2
```

```
  id sex age id x1 x2
1  1  M  28  1  A 17
2  2  F  22  2  B 14
3  3  M  24  3  B 13
4  4  M  25  4  A 25
5  5  M  24  5  A 28
6  6  F  18  6  A 16
7  7  F  29  7  A 22
8  8  M  25  8  C  6
9  9  M  23  9  D  8
10 10  F  19 10  A 10
>
```

データ結合B3.csv

	A	B	C
id	x1	x2	
1	A		17
3	B		13
4	A		25
7	A		22
10	A		10

```

> #byで指定された変数が対応するデータのみを結合
> d.bm <- merge(d.b1, d.b3, by="id")
> d.bm
  id sex age x1 x2
1   1   M  28  A 17
2   3   M  24  B 13
3   4   M  25  A 25
4   7   F  29  A 22
5  10   F  19  A 10
>

```

# 非対応行は削除される

```

> #byで指定された変数が非対応のデータも結合
> d.bmc <- merge(d.b1, d.b3, by="id", all=TRUE)
> d.bmc
  id sex age  x1 x2
1   1   M  28   A 17
2   2   F  22 <NA> NA
3   3   M  24   B 13
4   4   M  25   A 25
5   5   M  24 <NA> NA
6   6   F  18 <NA> NA
7   7   F  29   A 22
8   8   M  25 <NA> NA
9   9   M  23 <NA> NA
10 10   F  19   A 10
>

```

# 非対応行のデータはNA (Not Available, 欠測値) となる

## ロングフォーマットデータ・ワイドフォーマットデータ — 対応のあるデータのフォーマット変換

Rで対応のあるデータを扱う際、

- ・対応のあるデータをすべて縦に並べた1つの従属変数
- ・どの研究参加者のデータかを表す変数
- ・どの条件の下でのデータかを表す変数

からなるデータに、フォーマットを変換する必要があることがある。

多くの場合、対応のあるデータはWideフォーマットという形式で入力されている。  
これをR上で、Longフォーマットといわれる形式に変換する。

## stack, unstack関数を使う方法

## Long format

```
データフレーム名 <- stack(Wideフォーマットデータ名, select=c("水準名1", "水準名2", ...))
```

## Wide format

```
データフレーム名 <- unstack(Longフォーマットデータ名, 従属変数~独立変数)
```

## reshape関数を使う方法

## Long format

```
データフレーム名 <- reshape(data=wideデータ名, idvar="id変数名", v.names="従属変数につける名前",
  varying=c("対応のある変数名1", "対応のある変数名2", ...),
  timevar="時点(条件)を表す変数名", times=c("時点1", "時点2", ...),
  direction="long")
```

## Wide format

```
データフレーム名 <- reshape(data=longデータ名, idvar="id変数名", v.names="従属変数名",
  varying=c("対応のある変数名1", "対応のある変数名2", ...),
  timevar="時点(条件)を表す変数名",
  direction="wide")
```

```
> rm(list=ls())
> setwd("d:¥¥¥Rdocuments¥¥¥")
>
> # Wide formata data ファイルの読み込み
> d1 <- read.table("wide_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d1
  id gender t1 t2 t3
1  1      F  2  1  1
2  2      F  4  3  2
3  3      M  6  5  8
4  4      M  8  7 10
5  5      F 10  9 16
```

	id	gender	t1	t2	t3
	1	F	2	1	1
	2	F	4	3	2
	3	M	6	5	8
	4	M	8	7	10
	5	F	10	9	16

## &gt; # long format データの作成

## &gt; # stack関数を使う方法

```
> (d2 <- stack(d1))
```

```
  values ind
1      1  id
2      2  id
3      3  id
4      4  id
5      5  id
6      F gender
7      F gender
8      M gender
9      M gender
10     F gender
11     2   t1
12     4   t1
13     6   t1
```

```

14      8      t1
15     10      t1
16      1      t2
17      3      t2
18      5      t2
19      7      t2
20      9      t2
21      1      t3
22      2      t3
23      8      t3
24     10      t3
25     16      t3

```

```
> (d3 <- stack(d1[, c("t1", "t2", "t3")]))
```

```

  values ind
1       2  t1
2       4  t1
3       6  t1
4       8  t1
5      10  t1
6       1  t2
7       3  t2
8       5  t2
9       7  t2
10      9  t2
11      1  t3
12      2  t3
13      8  t3
14     10  t3
15     16  t3

```

```
> (d3s <- stack(d1, select=c("t1", "t2")))
```

```

  values ind
1       2  t1
2       4  t1
3       6  t1
4       8  t1
5      10  t1
6       1  t2
7       3  t2
8       5  t2
9       7  t2
10      9  t2

```

```
> # 属性変数も加えた long format data の作成
```

```
> # 既存の long format data を使う方法 1
```

```

> d4 <- data.frame(d1$id, d1$gender, d3)
> colnames(d4) <- c("id", "gender", "y", "x")
> d4$id <- as.factor(d4$id)
> d4$y <- as.factor(d4$y)
> d4

```

```

  id gender y x
1   1     F 2 t1
2   2     F 4 t1
3   3     M 6 t1
4   4     M 8 t1
5   5     F 10 t1
6   1     F 1 t2
7   2     F 3 t2
8   3     M 5 t2
9   4     M 7 t2
10  5     F 9 t2
11  1     F 1 t3
12  2     F 2 t3
13  3     M 8 t3
14  4     M 10 t3
15  5     F 16 t3

```

```

> # 既存の long format data を使う方法2
> d5 <- as.matrix(d1[, c("t1", "t2", "t3")])
> cnames <- colnames(d5)
> x <- as.factor(col(d5))
> x <- factor(x, levels=seq(1:length(cnames)), labels=cnames)
> id <- as.factor(d1$id)
> gender <- as.factor(d1$gender)
> (d6 <- data.frame(id, gender, y=as.vector(d5), x))
  id gender  y  x
1  1      F  2 t1
2  2      F  4 t1
3  3      M  6 t1
4  4      M  8 t1
5  5      F 10 t1
6  1      F  1 t2
7  2      F  3 t2
8  3      M  5 t2
9  4      M  7 t2
10 5      F  9 t2
11 1      F  1 t3
12 2      F  2 t3
13 3      M  8 t3
14 4      M 10 t3
15 5      F 16 t3

> # reshape関数を使う方法
> (d7 <- reshape(data=d1, idvar="id", varying=c("t1", "t2", "t3"), v.names="y",
+               timevar="x", times=c("t1", "t2", "t3"), direction="long"))
  id gender  x  y
1. t1  1      F t1  2
2. t1  2      F t1  4
3. t1  3      M t1  6
4. t1  4      M t1  8
5. t1  5      F t1 10
1. t2  1      F t2  1
2. t2  2      F t2  3
3. t2  3      M t2  5
4. t2  4      M t2  7
5. t2  5      F t2  9
1. t3  1      F t3  1
2. t3  2      F t3  2
3. t3  3      M t3  8
4. t3  4      M t3 10
5. t3  5      F t3 16

```

```

> # wide format データの作成

```

```

> # unstack 関数を使う方法
> d2

```

```

  values  ind
1      1   id
2      2   id
3      3   id
4      4   id
5      5   id
6      F gender
7      F gender
8      M gender
9      M gender
10     F gender
11     2    t1
12     4    t1

```

```

13      6      t1
14      8      t1
15     10      t1
16      1      t2
17      3      t2
18      5      t2
19      7      t2
20      9      t2
21      1      t3
22      2      t3
23      8      t3
24     10      t3
25     16      t3

```

```

> (d2u <- unstack(d2))
  id gender t1 t2 t3
1  1      F  2  1  1
2  2      F  4  3  2
3  3      M  6  5  8
4  4      M  8  7 10
5  5      F 10  9 16

```

```

> d3
  values ind
1      2  t1
2      4  t1
3      6  t1
4      8  t1
5     10  t1
6      1  t2
7      3  t2
8      5  t2
9      7  t2
10     9  t2
11     1  t3
12     2  t3
13     8  t3
14    10  t3
15    16  t3

```

```

> (d3u <- unstack(d3))
  t1 t2 t3
1  2  1  1
2  4  3  2
3  6  5  8
4  8  7 10
5 10  9 16

```

> # 属性変数も加えた long format data の読み込み

> d8 <- read.table("long\_データ2.csv", header=TRUE, sep=";", fileEncoding="shift-jis")

```

> d8
  id gender y x
1  1      F 2 t1
2  2      F 4 t1
3  3      M 6 t1
4  4      M 8 t1
5  5      F 10 t1
6  1      F 1 t2
7  2      F 3 t2
8  3      M 5 t2
9  4      M 7 t2
10 5      F 9 t2
11 1      F 1 t3
12 2      F 2 t3
13 3      M 8 t3
14 4      M 10 t3
15 5      F 16 t3

```

	A	B	C	D
1	id	gender	y	x
2	1	F	2	t1
3	2	F	4	t1
4	3	M	6	t1
5	4	M	8	t1
6	5	F	10	t1
7	1	F	1	t2
8	2	F	3	t2
9	3	M	5	t2
10	4	M	7	t2
11	5	F	9	t2
12	1	F	1	t3
13	2	F	2	t3
14	3	M	8	t3
15	4	M	10	t3
16	5	F	16	t3

```
> (d8u <- unstack(d8, y~x))
  t1 t2 t3
1  2  1  1
2  4  3  2
3  6  5  8
4  8  7 10
5 10  9 16
```

```
> # 属性変数も加えた wide format data の作成
```

```
> # 既存の wide format data を使う方法
```

```
> id <- as.factor(unique(d8$id))
```

```
> gender <- as.factor(d8$gender[id])
```

```
> (d8w <- data.frame(id, gender, d8u))
```

```
  id gender t1 t2 t3
1  1      F  2  1  1
2  2      F  4  3  2
3  3      M  6  5  8
4  4      M  8  7 10
5  5      F 10  9 16
```

```
> # reshape関数を使う方法
```

```
> (d8r <- reshape(data=d8, idvar="id", varying=c("t1","t2","t3"), v.names="y",
+               timevar="x", direction="wide"))
```

```
  id gender t1 t2 t3
1  1      F  2  1  1
2  2      F  4  3  2
3  3      M  6  5  8
4  4      M  8  7 10
5  5      F 10  9 16
```

## データ型の確認・変換

データ値が、数値か文字かなどの類型

数値型  
複素数型  
文字型  
論理型

```
> x <- 0
> y <- 5+0i
> z <- "4"
> w <- TRUE
>
>
> # データ型の確認
```

```
> mode(x)
[1] "numeric"

> is.numeric(x)
[1] TRUE

> is.complex(y)
[1] TRUE

> is.character(z)
[1] TRUE

> is.logical(w)
[1] TRUE
```

```
> # データ型の変換
>
> # as. ~ を使うと、その時だけ型が変わる
> as.numeric(z)
[1] 4
> z
[1] "4"
>
> as.complex(z)
[1] 4+0i
> z
[1] "4"
# z は文字のままなので、虚数表示にならない
>
> # mode関数を使うと、変数の型そのものが変わる
> mode(z) <- "numeric"
> z
[1] 4
>
> as.complex(z)
[1] 4+0i
> z
[1] 4
# z は数値になっているので、虚数表示になる
>
```



## データ構造の確認・変換

データの並びがどのような構造になっているかの類型

ベクトル	: 要素の型が同一な, データの1次元の並び
行列	: 要素の型が同一な, データの2次元の並びで, 行及び列の要素数がそれぞれ等しい
配列	: 要素の型が同一な, データの3次元以上の行列
リスト	: 異なる構造のデータをひとまとまりにしたもの
データフレーム	: 2次元の行列状だが各列のデータ構造は異なっても良い. 各行, 各列はラベルを持つ
順序なし因子	: 異なる要素の値をカテゴリとするカテゴリカル変数
順序付き因子	: 異なる要素の値をカテゴリとし, カテゴリ間に順序関係のあるカテゴリカル変数

```
> x <- c(1, 2)
>

> # データ構造の確認
> is.vector(x)
[1] TRUE

> is.matrix(x)
[1] FALSE

> is.array(x)
[1] FALSE

> is.list(x)
[1] FALSE

> is.data.frame(x)
[1] FALSE

> is.factor(x)
[1] FALSE

> is.ordered(x)
[1] FALSE

> # データ構造の変換
> as.vector(x)
[1] 1 2

> as.matrix(x)
      [,1]
[1,]    1
[2,]    2

> as.array(x)
[1] 1 2

> as.list(x)
[[1]]
[1] 1

[[2]]
[1] 2

> as.data.frame(x)
  x
1 1
2 2

> as.factor(x)
[1] 1 2
Levels: 1 2

> as.ordered(x)
[1] 1 2
Levels: 1 < 2
```

## 文字型の数字を数値型の数値に変換

```
行列名 <- as.matrix(データフレーム名)
storage.mode(行列名) <- "データ型名"
データフレーム名 <- as.data.frame(行列名)
```

## データ型名

"logical", "integer", "double", "complex", "raw", "character", "list", "expression",  
"name", "symbol", "function"

integer : 整数型, double : 倍精度型, numeric : 数値型, complex : 複素数型, character : 文字型,  
logical : 論理型

型変換したいデータを行列構造にして、すべての変数、データの型を同一にしておき、データ型の変換を行う。最後にデータフレーム構造に戻しておいたほうがよい。

文字型データを数値型にすると、数字は数値に変換されるが、文字は欠測値になる。

```
> setwd("d:¥¥")
> d1 <- read.table("型変換_データ1.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> d1
```

```
  id sex x1 x2 x3
1  1  F  1  2  4
2  2  F  2  4  3
3  3  M  4  2  5
4  4  M  1  5  2
5  5  F  3  2  2
6  6  F  4  2  3
7  7  M  5  3  6
>
```

	A	B	C	D	E
1	id	sex	x1	x2	x3
2	1	F	1	2	4
3	2	F	2	4	3
4	3	M	4	2	5
5	4	M	1	5	2
6	5	F	3	2	2
7	6	F	4	2	3
8	7	M	5	3	6

```
> d2 <- read.table("型変換_データ2.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> d2
```

```
  id sex x1 x2 x3
1  1  F  1  2  5
2  2  F  2  4  3
3  3  M  4  2  5
4  4  M  1  5  2
5  5  F  3  2  2
6  6  F  4  2  a
7  7  M  5 3,6  6
>
>
```

	A	B	C	D	E
1	id	sex	x1	x2	x3
2	1	F	1	2	5
3	2	F	2	4	3
4	3	M	4	2	5
5	4	M	1	5	2
6	5	F	3	2	2
7	6	F	4	2	a
8	7	M	5	3,6	6

> # d1は全てのデータが数値型なので、そのまま計算できる

```
> d1$xt <- rowSums(d1[,c("x1", "x2", "x3")])
```

```
> d1
  id sex x1 x2 x3 xt
1  1  F  1  2  4  7
2  2  F  2  4  3  9
3  3  M  4  2  5 11
4  4  M  1  5  2  8
5  5  F  3  2  2  7
6  6  F  4  2  3  9
7  7  M  5  3  6 14
>
```

> # d2は、数値と文字が混在しているので、そのままでは計算できない

```
> d2$xt <- rowSums(d2[,c("x1", "x2", "x3")])
```

以下にエラー rowSums(d2[,c("x1", "x2", "x3")]) :  
'x' は数値でなければなりません

```
>
>
```

```

> # データ型の変換
> # 行列構造にしてデータ型をそろえる
> d3 <- d2[,c("x1", "x2", "x3")]
> d3 <- as.matrix(d3)
> mode(d3)
[1] "character"

> d3
      x1 x2 x3
[1,] "1" "2" "5"
[2,] "2" "4" "3"
[3,] "4" "2" "5"
[4,] "1" "5" "2"
[5,] "3" "2" "2"
[6,] "4" "2" "a"
[7,] "5" "3,6" "6"
# すべて文字型のデータになっている

> # データ型を数値型にする. 文字はNAに変換される
> storage.mode(d3) <- "numeric"
警告メッセージ:
In storage.mode(d3) <- "numeric" : 強制変換により NA が生成されました

> d3
      x1 x2 x3
[1,]  1  2  5
[2,]  2  4  3
[3,]  4  2  5
[4,]  1  5  2
[5,]  3  2  2
[6,]  4  2 NA
[7,]  5 NA  6

> mode(d3)
[1] "numeric"

> # データフレーム構造に戻しておく
> d3 <- as.data.frame(d3)
> d3
      x1 x2 x3
1  1  2  5
2  2  4  3
3  4  2  5
4  1  5  2
5  3  2  2
6  4  2 NA
7  5 NA  6

> # もとのd2のかたちに戻す
> d2 <- data.frame(d2[,c("id", "sex")], d3)
>

> # 数値型になったデータで計算をする
> d2$xt <- rowSums(d2[,c("x1", "x2", "x3")])
> d2
  id sex x1 x2 x3 xt
1  1  F  1  2  5  8
2  2  F  2  4  3  9
3  3  M  4  2  5 11
4  4  M  1  5  2  8
5  5  F  3  2  2  7
6  6  F  4  2 NA NA
7  7  M  5 NA  6 NA

```

## factor型のデータ値の扱い

ifelse文などにおいて、factor型変数のデータ値を参照しようとする、データ値ではなく水準番号に値が置き換わってしまうことがある。このような場合は、factor型のデータを、character型に変えて参照する。

オブジェクト名 <- as.character(factor型データ)

```
> setwd("d:¥¥")
> d1 <- read.table("データの抽出_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d1
  id sex x1 x2 y
1  1  f  1  2 6
2  2  f  3  3 4
3  3  m  5  3 7
4  4  m  2  1 5
5  5  f  2  5 6
6  6  m  3  2 5
7  7  f  4  5 9
8  8  m  2  1 6
9  9   1  3 5
10 10  f  4  4 7
>
```

> # データ型の確認

```
> is.factor(d1$sex)
```

```
[1] TRUE
```

```
>
```

> # カテゴリ値の確認

> # 空白=1, m=2, f=3という、水準番号とカテゴリ値の対応になっている

```
> d1$sex
```

```
[1] f f m m f m f m f
```

```
Levels: f m
```

```
>
```

> # 空白=Unknown, その他はもとのままにしたい

> # factor型変数をそのまま参照

> # 要素番号がデータになってしまう

```
> d1$sex1 <- ifelse(d1$sex=="", "Unkown", d1$sex)
```

```
> d1
```

```
  id sex x1 x2 y  sex1
1  1  f  1  2 6     2
2  2  f  3  3 4     2
3  3  m  5  3 7     3
4  4  m  2  1 5     3
5  5  f  2  5 6     2
6  6  m  3  2 5     3
7  7  f  4  5 9     2
8  8  m  2  1 6     3
9  9   1  3 5 Unkown
10 10  f  4  4 7     2
>
```

> # factor型をcharacter型に変換して参照

```
> #
```

```
> d1$sex1 <- ifelse(d1$sex=="", "Unkown", as.character(d1$sex))
```

```
> d1
```

```
  id sex x1 x2 y  sex1
1  1  f  1  2 6     f
2  2  f  3  3 4     f
3  3  m  5  3 7     m
4  4  m  2  1 5     m
5  5  f  2  5 6     f
6  6  m  3  2 5     m
7  7  f  4  5 9     f
8  8  m  2  1 6     m
9  9   1  3 5 Unkown
10 10  f  4  4 7     f
```

```
> # b は character型
> (b <- "B")
[1] "B"
```

```
> # fb は factor型
> (fb <- as.factor(b))
[1] B
Levels: B
```

```
> # abc は 要素を10, 20, 30とするベクトル
> abc <- c(10, 20, 30)
```

```
> # abc の要素の名前はA, B, C
> names(abc) <- c("A", "B", "C")
> abc
  A  B  C
10 20 30
```

```
> # abcの中で、名前がbに対応する要素
> abc[b]
  B
20
```

```
> # abcの中で、名前がfbの値の水準番号に対応する要素
> abc[fb]
  A
10
```

```
> # abcの中で、名前がfbの値に対応する要素
> abc[as.character(fb)]
  B
20
```

## 5 統計図表

## 度数分布表 — 質的変数

質的変数

table(変数名)

prop.table(テーブル名)

table: 度数.

prop.table: 割合.

&gt; rm(list=ls())

&gt; setwd("d:¥¥")

&gt; d1 &lt;- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")

&gt; head(d1)

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

&gt; # 質的データの度数分布表

&gt; (t1 &lt;- table(d1\$学科))

医学	看護学	心理学
76	100	94

&gt; # 度数の降順

&gt; (t2 &lt;- t1[order(t1, decreasing=TRUE)])

看護学	心理学	医学
100	94	76

&gt; # 一定度数以上

&gt; (t3 &lt;- t1[t1&gt;=80])

看護学	心理学
100	94

&gt; # 割合

&gt; (p1 &lt;- prop.table(t1)\*100)

医学	看護学	心理学
28.14815	37.03704	34.81481

&gt; # 度数を割合の同時表示

&gt; rbind(t1, p1)

医学	看護学	心理学
t1 76.00000	100.00000	94.00000
p1 28.14815	37.03704	34.81481

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

## 度数分布表 — 量的変数

## 量的変数

library(fdth)

オブジェクト名 &lt;- fdt(変数名, start=始まりの値, end=終わりの値, h=階級幅)

あらかじめ fdth パッケージをインストールしておく必要がある。

&gt; rm(list=ls())

&gt; setwd("d:¥¥¥")

&gt; d1 &lt;- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")

&gt; head(d1)

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

&gt; # 量的変数の度数分布表

&gt; library(fdth)

&gt; (t1 &lt;- fdt(d1\$自己効力感, start=25, end=80, h=5))

Class	limits	f	rf	rf(%)	cf	cf(%)
[25, 30)	4	0.01	1.48	4	1.48	
[30, 35)	8	0.03	2.96	12	4.44	
[35, 40)	29	0.11	10.74	41	15.19	
[40, 45)	42	0.16	15.56	83	30.74	
[45, 50)	54	0.20	20.00	137	50.74	
[50, 55)	50	0.19	18.52	187	69.26	
[55, 60)	40	0.15	14.81	227	84.07	
[60, 65)	28	0.10	10.37	255	94.44	
[65, 70)	9	0.03	3.33	264	97.78	
[70, 75)	5	0.02	1.85	269	99.63	
[75, 80)	1	0.00	0.37	270	100.00	

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職



## クロス集計表

## 度数だけのクロス表

table(変数名1, 変数名2)

## 行周辺度数 (行和)

margin.table(テーブル名, 1)

## 行和に対する割合

prop.table(テーブル名, 1)

## 列周辺度数 (列和)

margin.table(テーブル名, 2)

## 列和に対する割合

prop.table(テーブル名, 2)

## 総周辺度数 (総和)

margin.table(テーブル名)

## 総和に対する割合

prop.table(テーブル名)

## 次元名 (変数名) の指定

table 関数でdnnオプションを使う

dnn=list("行名", "列名")

作成した表にあとから次元名を付ける

names(dimnames(テーブル名)) &lt;- c("変数名1", "変数名2", ...)

次元名を指定すると、行名、列名を表示することができる (指定しないと表示されない)。

```
> setwd("d:¥¥")
> d1 <- read.table("度数分布_データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
```

```
  id sex score grade
1  1 f    85  A
2  2 f   100  A
3  3 m    80  A
4  4 f    80  A
5  5 f    65  C
6  6 m    80  A
```

	A	B	C	D
1	id	sex	score	grade
2	1	f	85	A
3	2	f	100	A
4	3	m	80	A
5	4	f	80	A
6	5	f	65	C
7	6	m	80	A
8	7	m	75	B
9	8	m	100	A
10	9	f	65	C
11	10	m	55	D
12	11	f	90	A
13	12	m	80	A
14	13	f	65	C
15	14	f	75	B
16	15	m	70	B
17	16	f	70	B
18	17	f	75	B
19	18	m	65	C
20	19	m	50	D
21	20	f	95	A

```
> # 度数だけのクロス表
> t1 <- table(d1$sex, d1$grade, dnn=list("sex", "grade"))
> t1
```

```
      grade
sex      A      B      C      D
f         25     18      7      1
m         15     12      8      4
```

## &gt; # 行周辺度数 (行和)

&gt; margin.table(t1, 1)

```
sex
f      m
51     39
```

## &gt; # 行和に対する割合

&gt; prop.table(t1, 1)

```
      grade
sex      A      B      C      D
f 0.49019608 0.35294118 0.13725490 0.01960784
m 0.38461538 0.30769231 0.20512821 0.10256410
```

```
>
```

```
> # 列周辺度数(列和)
> margin.table(t1, 2)
```

```
grade
A      B      C      D
40     30     15      5
```

```
>
```

```
> # 列和に対する割合
> prop.table(t1, 2)
```

```
grade
sex    A      B      C      D
f      0.6250000 0.6000000 0.4666667 0.2000000
m      0.3750000 0.4000000 0.5333333 0.8000000
```

```
>
```

```
> # 総周辺度数(総和)
> margin.table(t1)
```

```
[1] 90
```

```
>
```

```
> # 総和に対する割合
> prop.table(t1)
```

```
grade
sex    A      B      C      D
f      0.27777778 0.20000000 0.07777778 0.01111111
m      0.16666667 0.13333333 0.08888889 0.04444444
```

```
>
```

```
>
```

```
>
```

## 割合なども入ったクロス集計表

## gmodels パッケージの CrossTable 関数を使う方法

```
library(gmodels)
CrossTable(変数1, 変数2)
```

あらかじめ gmodels パッケージをインストールしておく必要がある。  
 expected=TRUE とすると、期待度数と、カイ 2 乗値検定の結果を表示する。  
 fisher=TRUE や mcnemar=TURE とすると、フィッシャーの直接検定やマクネマーの検定をしてくれる。

## descr パッケージの CrossTable 関数を使う方法

```
library(descr)
CrossTable(変数1, 変数2)
```

あらかじめ descr パッケージをインストールしておく必要がある。  
 デフォルトで期待値の表示と、カイ 2 乗検定をしてくれる。  
 fisher=TRUE や mcnemar=TURE とすると、フィッシャーの直接検定やマクネマーの検定をしてくれる。

## オリジナル関数を使う方法

```
cross.table(変数名1, 変数名2)
```

R 起動後に、一度関数部分を実行しておく必要がある。  
 digit=桁数で、セルパーセントの有効桁数を指定できる。

## クロス集計表の警告メッセージについて

クロス集計表を作成したとき、次のようなメッセージが表示される場合があります。」

警告メッセージ:

```
chisq.test(tab, correct = FALSE, ...) で:  
Chi-squared approximation may be incorrect
```

これは、データ数が少ないために表示される警告で、エラーではありません。クロス表の検定を行うとき使う「カイ2乗統計量」のカイ2乗分布へのあてはまりが良くないという警告です。

警告が出る原因はデータ数が少ないことによります。大まかな目安ですが、1つ1つのセルに少なくとも5個、できれば10個以上のデータがないと、カイ2乗統計量はあてはまりが悪くなります。

```
> setwd("d:¥¥")
> d1 <- read.table("度数分布_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> table(d1$sex, d1$grade)
```

	A	B	C	D
f	25	18	7	1
m	15	12	8	4

```
>
```

	A	B	C	D
1	id	sex	score	grade
2	1	f	85	A
3	2	f	100	A
4	3	m	80	A
5	4	f	80	A
6	5	f	65	C
7	6	m	80	A
8	7	m	75	B
9	8	m	100	A
10	9	f	65	C
11	10	m	55	D
12	11	f	90	A
13	12	m	80	A
14	13	f	65	C
15	14	f	75	B
16	15	m	70	B
17	16	f	70	B
18	17	f	75	B
19	18	m	65	C
20	19	m	50	D
21	20	f	95	A

```
> #gmodels パッケージの CrossTable 関数を使う方法
> library(gmodels)
> CrossTable(d1$sex, d1$grade, expected=TRUE)
```

Cell Contents	
	N
	Expected N
	Chi-square contribution
	N / Row Total
	N / Col Total
	N / Table Total

Total Observations in Table: 90

d1\$sex	d1\$grade	A	B	C	D	Row Total
f		25	18	7	1	51
		22.667	17.000	8.500	2.833	
		0.240	0.059	0.265	1.186	
		0.490	0.353	0.137	0.020	0.567
		0.625	0.600	0.467	0.200	
		0.278	0.200	0.078	0.011	
m		15	12	8	4	39
		17.333	13.000	6.500	2.167	
		0.314	0.077	0.346	1.551	
		0.385	0.308	0.205	0.103	0.433
		0.375	0.400	0.533	0.800	
		0.167	0.133	0.089	0.044	
Column Total		40	30	15	5	90
		0.444	0.333	0.167	0.056	

Statistics for All Table Factors

Pearson's Chi-squared test

Chi^2 = 4.038462 d.f. = 3 p = 0.2573408

警告メッセージ:

In chisq.test(t, correct = FALSE, ...) :  
Chi-squared approximation may be incorrect

```
> detach("package:gmodels")
>
```

```
> #descr パッケージの CrossTable関数を使う方法
> library(descr)
> (ctd1 <-CrossTable(d1$sex, d1$grade))
```

Cell Contents

N
Expected N
Chi-square contribution
N / Row Total
N / Col Total
N / Table Total

d1\$sex	d1\$grade				Total
	A	B	C	D	
f	25	18	7	1	51
	22.7	17.0	8.5	2.8	
	0.240	0.059	0.265	1.186	
	0.490	0.353	0.137	0.020	0.567
	0.625	0.600	0.467	0.200	
	0.278	0.200	0.078	0.011	
m	15	12	8	4	39
	17.3	13.0	6.5	2.2	
	0.314	0.077	0.346	1.551	
	0.385	0.308	0.205	0.103	0.433
	0.375	0.400	0.533	0.800	
	0.167	0.133	0.089	0.044	
Total	40	30	15	5	90
	0.444	0.333	0.167	0.056	

Statistics for All Table Factors

Pearson's Chi-squared test

Chi^2 = 4.038462      d.f. = 3      p = 0.2573408

警告メッセージ:

In chisq.test(t, correct = FALSE, ...) :  
Chi-squared approximation may be incorrect

## 群別のクロス集計表

## 度数だけのクロス表

table(変数名1, 変数名2, 群分け変数名)

## 周辺度数

新データフレーム名 &lt;- データフレーム名[データフレーム名\$群分け変数==値, ]

margin.table(新データフレーム名)

## 割合

新データフレーム名 &lt;- データフレーム名[データフレーム名\$群分け変数==値, ]

prop.table(新データフレーム名)

```
> setwd("d:¥¥¥")
> d1 <- read.table("連関係数_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
>
```

## &gt; #度数だけのクロス表

```
> (t1 <- table(d1$method, d1$grade, d1$sex))
```

```
, , = F
      A B C D
1  5 19 5 1
2 12 13 4 5
```

```
, , = M
      A B C D
1  5 10 1 2
2  9  5 4 0
```

```
>
> d1F <- d1[d1$sex=="F", ]
> d1M <- d1[d1$sex=="M", ]
> t1F <- table(d1F$method, d1F$grade)
> t1M <- table(d1M$method, d1M$grade)
```

## &gt; #行周辺度数, 行和に対する割合

```
> margin.table(t1F, 1)
```

```
1 2
30 34
> prop.table(t1F, 1)
```

```
      A      B      C      D
1 0.1666667 0.6333333 0.1666667 0.0333333
2 0.3529412 0.3823529 0.1176471 0.1470588
```

## &gt; #列周辺度数, 列和に対する割合

```
> margin.table(t1F, 2)
```

```
      A B C D
17 32  9  6
> prop.table(t1F, 2)
```

```
      A      B      C      D
1 0.2941176 0.5937500 0.5555556 0.1666667
2 0.7058824 0.4062500 0.4444444 0.8333333
```

## &gt; #総周辺度数, 総和に対する割合

```
> margin.table(t1F)
```

```
[1] 64
> prop.table(t1F)
```

```
      A      B      C      D
1 0.078125 0.296875 0.078125 0.015625
2 0.187500 0.203125 0.062500 0.078125
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	F	1	A
6	5	M	1	B
7	6	M	2	A
8	7	F	2	B
9	8	F	2	B
10	9	F	2	B
11	10	F	1	B
12	11	M	2	B
13	12	F	2	A
14	13	M	1	D
15	14	F	2	C
16	15	F	2	D
17	16	F	2	B
18	17	F	2	A
19	18	F	2	B
20	19	F	2	B
21	20	F	1	B

## フラットなクロス集計表

## 元データから作成する場合

```
ftable(データフレーム名[, c("変数名1", "変数名2", ...)], row.vars=c("変数名1", "変数名2", ...))
```

row.vars で行にカテゴリを並べる変数を指定する. col.vars というオプションをある.

ftable 関数で作られる表では, カテゴリ名が初回のみ表示され, 2回目以降は省略される.

すべての行にカテゴリ名が入ったデータを作成するには, ftable で作成した表をデータフレーム化する. その際, 後ろの変数から順にソートされてしまうので, 前の変数からソートし直すと比較しやすい.

## 既にあるクロス表から作成する場合

```
ftable(列変数名 ~ 行変数名1 + 行変数名2 + ..., data=テーブル名)
```

```
ftable(. ~ 行変数名1 + 行変数名2 + ..., data=テーブル名)
```

フラット化する変数を行変数, フラット化しない変数を列変数に入れる.

すべての変数についてフラット化したい場合は左辺に「.」(ピリオド)を入れる.

```
> setwd("d:¥¥¥")
> d1 <- read.table("連関係数_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  id sex method grade
1  1  F      1      B
2  2  M      1      B
3  3  F      2      A
4  4  F      1      A
5  5  M      1      B
6  6  M      2      A
```

## &gt; # データからフラットなクロス表をつくる

```
> (ft1 <- ftable(d1[, c("sex", "method", "grade")], row.vars=c("sex", "method", "grade")))
```

```
sex method grade
F    1      A      5
      B     19
      C      5
      D      1
      2      A     12
      B     13
      C      4
      D      5
M    1      A      5
      B     10
      C      1
      D      2
      2      A      9
      B      5
      C      4
      D      0
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	F	1	A
6	5	M	1	B
7	6	M	2	A
8	7	F	2	B
9	8	F	2	B
10	9	F	2	B
11	10	F	1	B
12	11	M	2	B
13	12	F	2	A
14	13	M	1	D
15	14	F	2	C
16	15	F	2	D
17	16	F	2	B
18	17	F	2	A
19	18	F	2	B
20	19	F	2	B
21	20	F	1	B

# 表なのでカテゴリの値は最小限しか表示されない

```
> # フラットなクロス表をデータ行列に変換
```

```
> (fd1 <- as.data.frame(ft1))
```

	sex	method	grade	Freq
1	F	1	A	5
2	M	1	A	5
3	F	2	A	12
4	M	2	A	9
5	F	1	B	19
6	M	1	B	10
7	F	2	B	13
8	M	2	B	5
9	F	1	C	5
10	M	1	C	1
11	F	2	C	4
12	M	2	C	4
13	F	1	D	1
14	M	1	D	2
15	F	2	D	5
16	M	2	D	0

```
# データなのでカテゴリの値がすべての行に入っている
```

```
> # ソートし直す
```

```
> (fd1 <- fd1[order(fd1$sex, fd1$method, fd1$grade),])
```

	sex	method	grade	Freq
1	F	1	A	5
5	F	1	B	19
9	F	1	C	5
13	F	1	D	1
3	F	2	A	12
7	F	2	B	13
11	F	2	C	4
15	F	2	D	5
2	M	1	A	5
6	M	1	B	10
10	M	1	C	1
14	M	1	D	2
4	M	2	A	9
8	M	2	B	5
12	M	2	C	4
16	M	2	D	0

```
> # 既にあるクロス表からフラットなクロス表を作成
```

```
> # クロス表
```

```
> (t1 <- table(d1[,c("method", "grade", "sex")], dnn=list("method", "grade", "sex")))
```

```
, , sex = F
```

		grade			
method		A	B	C	D
	1	5	19	5	1
	2	12	13	4	5

```
, , sex = M
```

		grade			
method		A	B	C	D
	1	5	10	1	2
	2	9	5	4	0



## &gt; # フラットなクロス表

> **ftable**(grade ~ sex + method, data=t1)

	sex	method	grade	A	B	C	D
F	1			5	19	5	1
	2			12	13	4	5
M	1			5	10	1	2
	2			9	5	4	0

> (ft2 <- **ftable**(. ~ sex + method + grade, data=t1))

sex	method	grade	
F	1	A	5
		B	19
		C	5
		D	1
	2	A	12
		B	13
		C	4
		D	5
M	1	A	5
		B	10
		C	1
		D	2
	2	A	9
		B	5
		C	4
		D	0

> (fd2<- **as.data.frame**(ft2))

	sex	method	grade	Freq
1	F	1	A	5
2	M	1	A	5
3	F	2	A	12
4	M	2	A	9
5	F	1	B	19
6	M	1	B	10
7	F	2	B	13
8	M	2	B	5
9	F	1	C	5
10	M	1	C	1
11	F	2	C	4
12	M	2	C	4
13	F	1	D	1
14	M	1	D	2
15	F	2	D	5
16	M	2	D	0

## フラットなクロス集計表から通常のクロス集計表を作成

`xtabs`(度数変数名 ~ 変数名1 + 変数名2 + ..., `data`=フラットなクロス表名)

フラットなクロス表は、`fTable` で作成した表でも、データフレームでもよい。

```
> setwd("d:¥¥")
> d1 <- read.table("連関係数_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id sex method grade
1  1  F      1      B
2  2  M      1      B
3  3  F      2      A
4  4  F      1      A
5  5  M      1      B
6  6  M      2      A
>
>
```

```
> # フラットなクロス表
> (ft1 <- fTable(d1[,c("sex", "method", "grade")], row.vars=c("sex", "method", "grade")))
```

```
sex method grade
F      1      A      5
      B     19
      C      5
      D      1
      2      A     12
      B     13
      C      4
      D      5
M      1      A      5
      B     10
      C      1
      D      2
      2      A      9
      B      5
      C      4
      D      0
>
>
```

```
> # フラットなクロス表をデータ行列に変換
> (fd1 <- as.data.frame(ft1))
```

```
1  F      1  A      5
2  M      1  A      5
3  F      2  A     12
4  M      2  A      9
5  F      1  B     19
6  M      1  B     10
7  F      2  B     13
8  M      2  B      5
9  F      1  C      5
10 M      1  C      1
11 F      2  C      4
12 M      2  C      4
13 F      1  D      1
14 M      1  D      2
15 F      2  D      5
16 M      2  D      0
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	F	1	A
6	5	M	1	B
7	6	M	2	A
8	7	F	2	B
9	8	F	2	B
10	9	F	2	B
11	10	F	1	B
12	11	M	2	B
13	12	F	2	A
14	13	M	1	D
15	14	F	2	C
16	15	F	2	D
17	16	F	2	B
18	17	F	2	A
19	18	F	2	B
20	19	F	2	B
21	20	F	1	B

> # フラットなクロス表からクロス表を作成  
 > # フラットなクロス表でも、それをデータフレーム化したものでも、クロス表が作成できる

> xtabs(Freq~method+grade+sex, data=ft1)

, , sex = F

	grade			
method	A	B	C	D
1	5	19	5	1
2	12	13	4	5

, , sex = M

	grade			
method	A	B	C	D
1	5	10	1	2
2	9	5	4	0

> xtabs(Freq~method+grade+sex, data=fd1)

, , sex = F

	grade			
method	A	B	C	D
1	5	19	5	1
2	12	13	4	5

, , sex = M

	grade			
method	A	B	C	D
1	5	10	1	2
2	9	5	4	0

>

## 円グラフ

## 円グラフ

pie(テーブル名, clockwise=TRUE, col=色名)

```
> setwd("d:¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学   女          高          49        23 就職
2    2      20Y2 心理学   男          低          57        29 就職
3    3      20Y1 医学     女          高          42        23 進学
4    4      20Y1 看護学   女          高          41        23 就職
5    5      20Y2 医学     男          低          41        22 就職
6    6      20Y1 心理学   女          低          47        24 就職
>

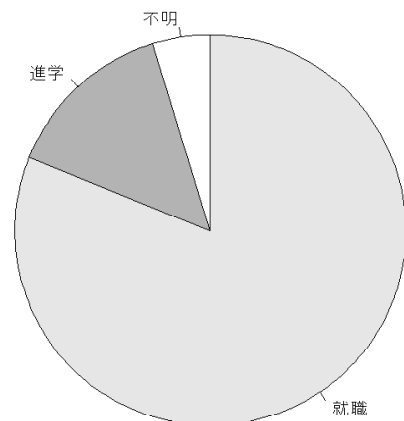
> #度数
> (t1 <- table(d1$進路))

就職 進学 不明
219   38   13
>
```

```
> # 円グラフ
> par(mar=c(1, 1, 1, 1))
> pie(t1, clockwise=TRUE, col=gray(c(0.9, 0.7, 1.0)), cex=1.4)
>
```

```
> #度数の割合
> (p1 <- prop.table(t1))

就職      進学      不明
0.8111111 0.1407407 0.04814815
>
```



	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

## 帯グラフ

## 帯グラフ

```
barplot(as.matrix(割合テーブル名), horiz=TRUE, beside=FALSE,
        ylim=c(0, 1), width=0.3, col=色名, legend.text=TRUE)
```

ylim: 帯グラフ領域全体の高さ  
width: 帯の幅  
cex: 文字の大きさの調整  
beside: 従属変数の水準別に棒を分けるか  
horizontal: 横向きグラフ  
las: 軸ラベルの向き. 1は水平  
segments: 直線を引く

```
> setwd("d:¥¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=";", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度 学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学 女          高          49        23 就職
2    2      20Y2 心理学 男          低          57        29 就職
3    3      20Y1 医学 女          高          42        23 進学
4    4      20Y1 看護学 女          高          41        23 就職
5    5      20Y2 医学 男          低          41        22 就職
6    6      20Y1 心理学 女          低          47        24 就職
>
```

```
> #度数
> (t1 <- table(d1$進路))
```

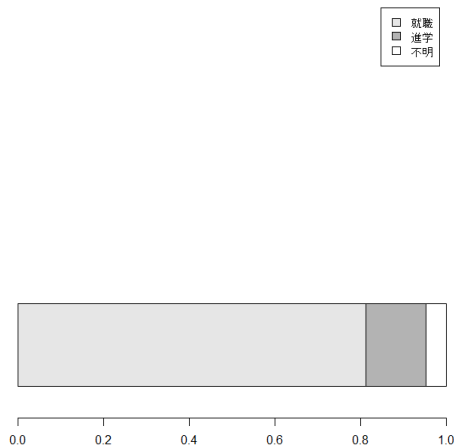
```
就職 進学 不明
219   38   13
>
```

```
> #度数の割合
> (p1 <- prop.table(t1))
```

```
      就職      進学      不明
0.8111111 0.1407407 0.04814815
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

```
> # 帯グラフ
> # 度数の割合ベクトルを行列に変換するのがポイント
> par(mar=c(3, 4, 3, 1))
> barplot(as.matrix(p1), horiz=TRUE, beside=FALSE, las=1,
+   ylim=c(0, 1.4), width=0.3, col=gray(c(0.9, 0.7, 1.0)), legend.text=T)
>
```



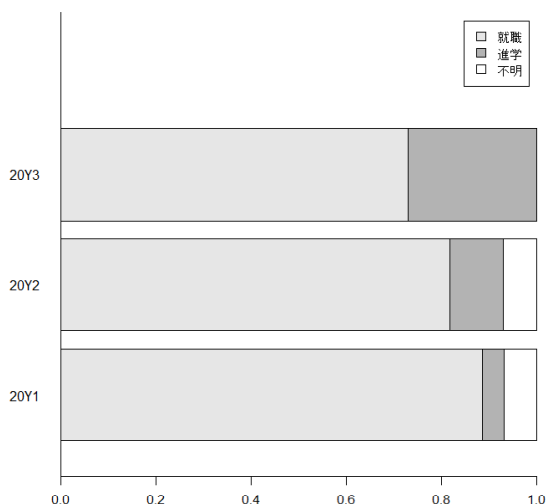
```
> # 年度別の帯グラフ
> # 列ごとに割合を算出
> (t2 <- table(d1$進路, d1$入学年度))
```

	20Y1	20Y2	20Y3
就職	77	80	62
進学	4	11	23
不明	6	7	0

```
> (p2 <- prop.table(t2, 2))
```

	20Y1	20Y2	20Y3
就職	0.88505747	0.81632653	0.72941176
進学	0.04597701	0.11224490	0.27058824
不明	0.06896552	0.07142857	0.00000000

```
> par(mar=c(3, 4, 3, 1))
> barplot(as.matrix(p2), horiz=TRUE, beside=FALSE, las=1,
+   ylim=c(0, 1.4), width=0.3, col=gray(c(0.9, 0.7, 1.0)), legend.text=T)
> segments(0, -1, 0, 2)
```



## segments()について

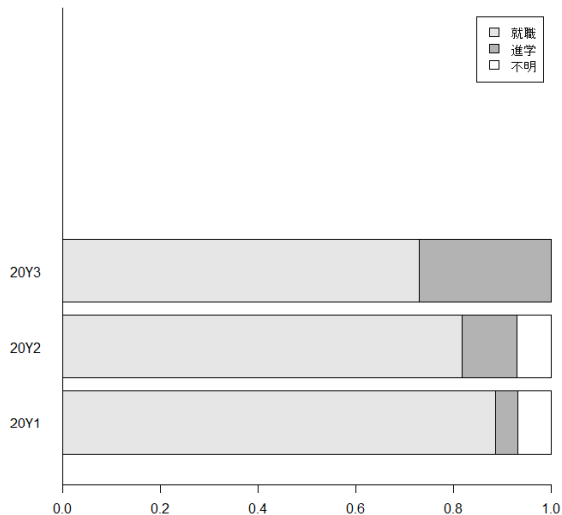
segments()は、既存の図に直線を書き加えるコマンドです。カッコ内の4つの値のうち、最初の2つが始点の座標、うしろの2つが終点の座標を表します。

segments(0, -1, 0, 2)の場合、(0, -1)から(0, 2)に至る線分を引くということになります。始点と終点のx座標がともに0なので、縦の直線になります。

## 帯グラフの凡例と帯が重なることについて

画面の大きさによっては、帯グラフの帯と凡例が重なってしまうことがあります。帯グラフの凡例が帯に重ならないようにするには、帯の幅を狭くすることが考えられます。例えば、width=0.3としているところを、width=0.2 などとすると、帯の幅が狭くなり、帯と凡例が離れるようになります。

```
> # 帯の幅を狭くする
> barplot(as.matrix(p2), horiz=TRUE, beside=FALSE, las=1,
+   ylim=c(0, 1.4), width=0.2, col=gray(c(0.9, 0.7, 1.0)), legend.text=T)
> segments(0, -1, 0, 2)
```



## 棒グラフ

barplot(変数 (または行列) 名, ...)

## 群別 (変数別) のグラフの比較

barplot(変数 (または行列) 名, beside=TRUE, legend=TRUE, space=棒の間の大きさ)

## オプション

width : 棒の幅を設定する (横軸に関する他のオプションと一緒に使って初めて有効となる)

space : 棒間の間隔を設定する

offset : 縦軸の開始点を設定する. 表示するデータからも, 当該量を引いておく必要がある. ほかにも, 文字の大きさを変えるオプションなど, いろいろ使える.

```
> setwd("d:¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
```

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

```
> # 度数
> (t1 <- table(d1$進路))
```

就職	進学	不明
219	38	13

```
> # 棒グラフ
> par(mar=c(5, 5, 3, 1))
> barplot(as.matrix(t1), xlim=c(0, 5), ylim=c(0, 250), ylab="Frequency",
+ las=1, beside=TRUE, space=0.5, col=gray(c(0.9, 0.7, 1.0)), legend=TRUE)
> segments(-1, 0, 5, 0)
>
```

```
> # 学科 × 進路のクロス表
> (t2 <- table(d1$学科, d1$進路))
```

	就職	進学	不明
医学	64	10	2
看護学	80	13	7
心理学	75	15	4

```
> # 進路ごとに学科を比較した棒グラフ
> par(mar=c(5, 5, 3, 1))
> barplot(as.matrix(t2), xlim=c(0, 14), ylim=c(0, 80), ylab="Frequency",
+ las=1, beside=TRUE, col=gray(c(0.9, 0.7, 1.0)), legend=TRUE)
> segments(-1, 0, 14, 0)
>
```

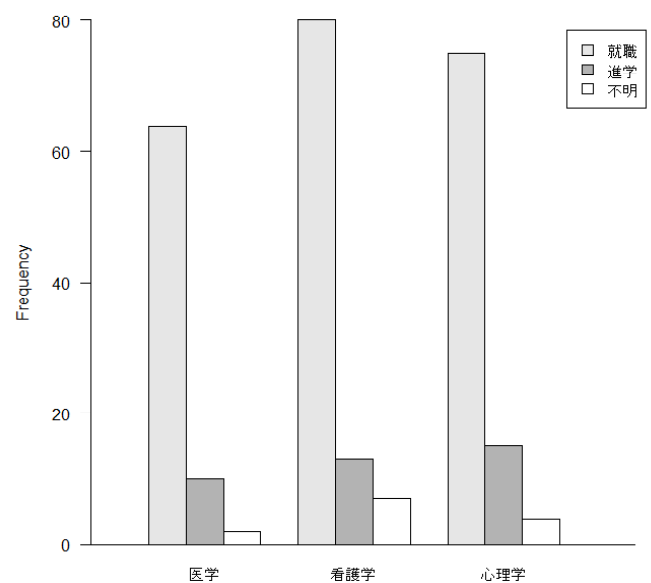
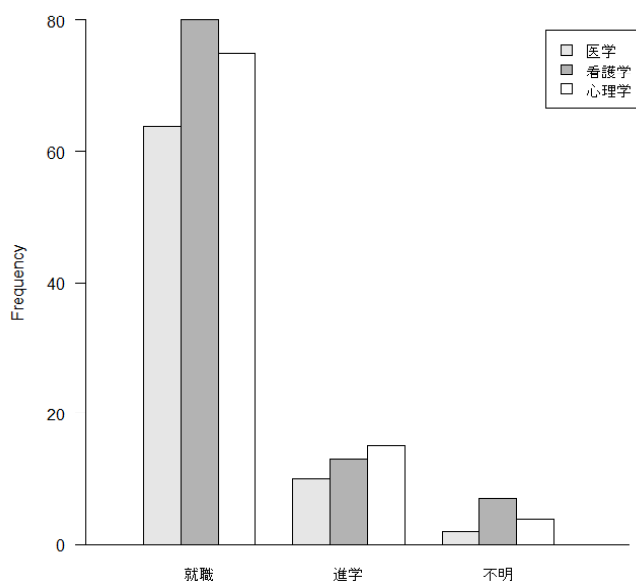
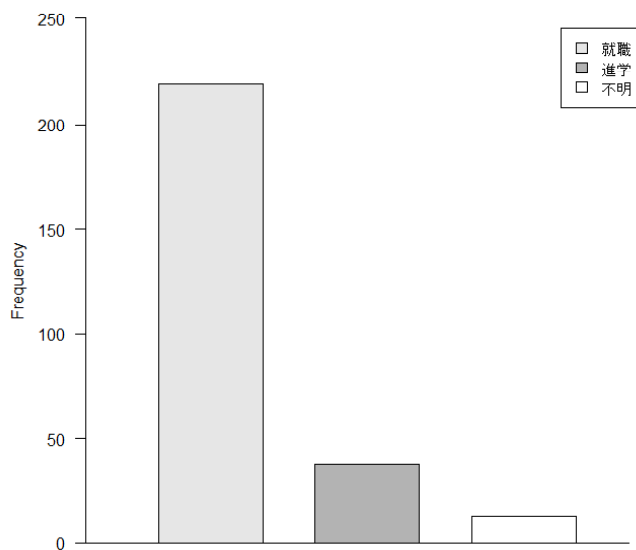
```
> # 進路 × 学科のクロス表
> (t3 <- table(d1$進路, d1$学科))
```

	医学	看護学	心理学
就職	64	80	75
進学	10	13	15
不明	2	7	4

```
> # 学科ごとに進路を比較した棒グラフ
> par(mar=c(5, 5, 3, 1))
> barplot(as.matrix(t3), xlim=c(0, 14), ylim=c(0, 80), ylab="Frequency",
+ las=1, beside=TRUE, col=gray(c(0.9, 0.7, 1.0)), legend=TRUE)
> segments(-1, 0, 14, 0)
```



	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モトリウム	自己効力感	学習意欲	進路
2		1 20Y1	看護学	女	高	49	23	就職
3		2 20Y2	心理学	男	低	57	29	就職
4		3 20Y1	医学	女	高	42	23	進学
5		4 20Y1	看護学	女	高	41	23	就職
6		5 20Y2	医学	男	低	41	22	就職
7		6 20Y1	心理学	女	低	47	24	就職
8		7 20Y3	看護学	女	高	41	26	就職
9		8 20Y2	医学	男	低	43	29	就職
10		9 20Y2	看護学	女	低	54	21	就職
11		10 20Y3	医学	女	高	55	30	進学
12		11 20Y1	心理学	男	低	66	25	就職
13		12 20Y1	心理学	女	低	38	30	就職
14		13 20Y1	心理学	女	低	45	29	就職
15		14 20Y1	医学	男	高	53	23	就職
16		15 20Y1	心理学	男	高	47	25	就職



```
> # 棒の間隔を調整したグラフを描く方法
> # space オプションを利用する
```

```
> # あるテスト得点について、性別、自己効力感別（3段階）、および、学校別の平均値が以下だったとする.
```

```
>
```

```
> # 性別
```

```
> mF <- 63; mM <- 61
```

```
>
```

```
> # 自己効力感
```

```
> m1 <- 60; m2 <- 64; m3 <- 62
```

```
>
```

```
> # 学校別
```

```
> mA <- 61; mB <- 63; mC <- 62; mD <- 62
```

```
>
```

```
> # 各要因の平均値の棒グラフを、同時に表示する
```

```
> mgraph <- c(mF, mM, m1, m2, m3, mA, mB, mC, mD)
```

```
> names(mgraph) <- c("F", "M", "1", "2", "3", "A", "B", "C", "D")
```

```
>
```

```
> barplot(mgraph, xlim=c(1, 18), ylim=c(0, 70), ylab="Mean",
+ las=1, col=gray(c(0.4, 1.0, 0.4, 0.7, 1.0, 0.4, 0.6, 0.8, 1.0)), cex.names=0.8,
+ legend=F, width=1.2, space=c(2, 0, 2, 0, 0, 2, 0, 0, 0), main="各群の平均点")
> segments(0, 0, 20, 0)
>
```

```
> # 縦軸の原点をずらした図を描く方法
```

```
> # offsetオプションを利用する
```

```
> # テスト得点を間隔尺度と考えると、縦軸を0から始める必要はなく、
```

```
> # もっと見やすいグラフを書くことも考えられる.
```

```
> # 縦軸の範囲を、50 から 70 にすることを考える.
```

```
>
```

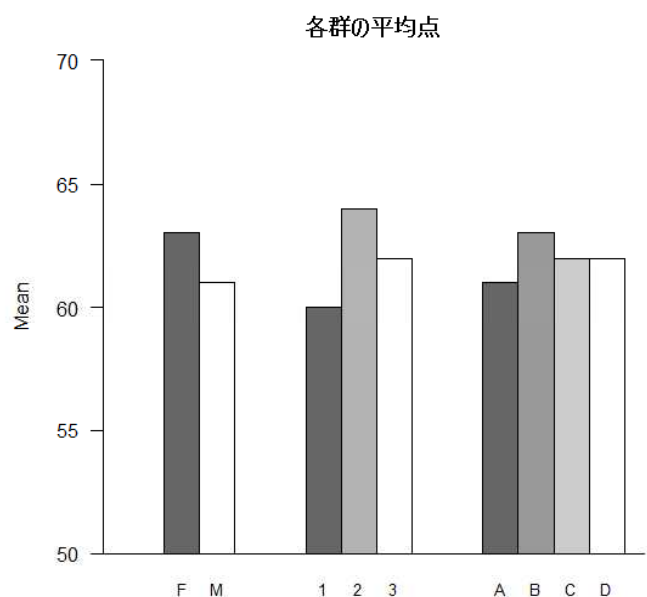
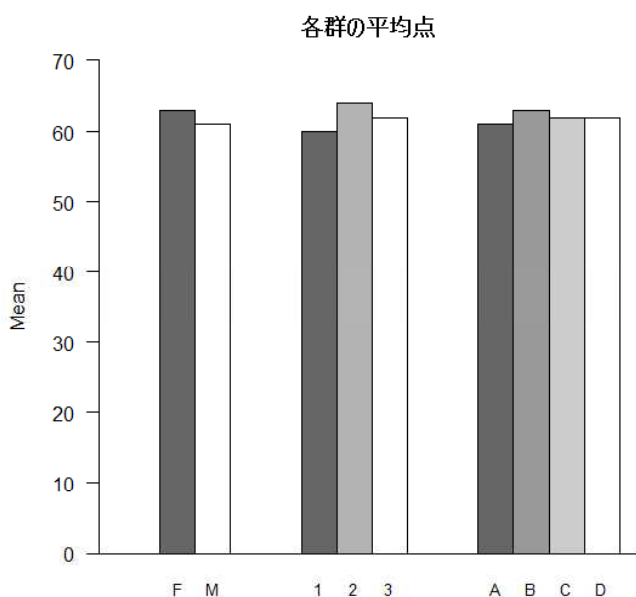
```
> voffset <- 50
```

```
> mgraph <- (c(mF, mM, m1, m2, m3, mA, mB, mC, mD) - voffset)
```

```
> names(mgraph) <- c("F", "M", "1", "2", "3", "A", "B", "C", "D")
```

```
>
```

```
> barplot(mgraph, xlim=c(1, 18), ylim=c(voffset, 70), ylab="Mean", offset=voffset,
+ las=1, col=gray(c(0.4, 1.0, 0.4, 0.7, 1.0, 0.4, 0.6, 0.8, 1.0)), cex.names=0.8,
+ legend=F, width=1.2, space=c(2, 0, 2, 0, 0, 2, 0, 0, 0), main="各群の平均点")
> segments(0, voffset, 20, voffset)
>
```



## ヒストグラム

hist(変数名, オプション群)

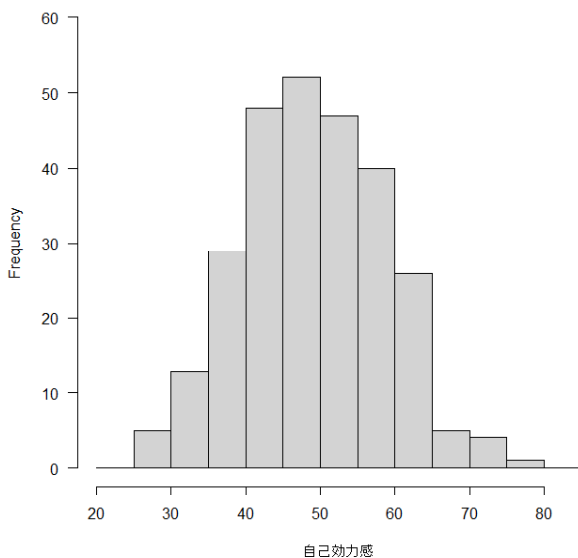
breaks=seq(a, b, by=c) : x軸の値を, aからbまで増分cで区切っていく  
 ylim=c(a, b) : y軸の範囲をaからbまでとする.  
 las=1 : 目盛りの値を水平にする.  
 xlab="aa" : x軸のラベルをaaとする.

```
> setwd("d:¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
```

番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23 就職
2	2	20Y2	心理学	男	低	57	29 就職
3	3	20Y1	医学	女	高	42	23 進学
4	4	20Y1	看護学	女	高	41	23 就職
5	5	20Y2	医学	男	低	41	22 就職
6	6	20Y1	心理学	女	低	47	24 就職

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2		1 20Y1	看護学	女	高	49	23	就職
3		2 20Y2	心理学	男	低	57	29	就職
4		3 20Y1	医学	女	高	42	23	進学
5		4 20Y1	看護学	女	高	41	23	就職
6		5 20Y2	医学	男	低	41	22	就職
7		6 20Y1	心理学	女	低	47	24	就職
8		7 20Y3	看護学	女	高	41	26	就職
9		8 20Y2	医学	男	低	43	29	就職
10		9 20Y2	看護学	女	低	54	21	就職
11		10 20Y3	医学	女	高	55	30	進学
12		11 20Y1	心理学	男	低	66	25	就職
13		12 20Y1	心理学	女	低	38	30	就職
14		13 20Y1	心理学	女	低	45	29	就職
15		14 20Y1	医学	男	高	53	23	就職
16		15 20Y1	心理学	男	高	47	25	就職

```
> # ヒストグラム
> par(mar=c(5, 5, 3, 1))
> hist(d1$自己効力感, breaks=seq(20, 85, by=5), ylim=c(0, 60), las=1, xlab="自己効力感", main="")
```



`breaks=seq(20, 85, by=5)`について

`seq(x, y, by=z)` は、 $x$ から $y$ まで、 $z$ 点刻みの値を作る関数です。よって、`seq(20, 85, by=5)`は、(10, 25, 30, ..., 85)という数値を算出します。

`hist` 関数の `break` オプションは、各階級の下限と上限の値を指定するオプションでう。上の例だと

$20 < x \leq 25$

$25 < x \leq 30$

...

$80 < x \leq 85$

という階級をつくります。

ヒストグラムの横軸の目盛は、基本的に整数値のところに付けられるので、(20, 30, ..., 80)のところに目盛がつけられます。

これに対して、`seq(17.5, 87.5, by=5)` とした場合は、

$17.5 < x \leq 22.5$

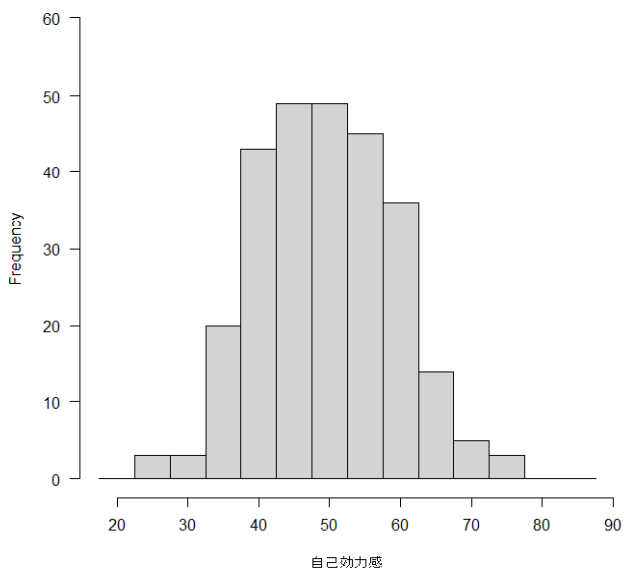
$22.5 < x \leq 27.5$

...

$82.5 < x \leq 87.5$

という階級が作られます。この場合は、棒の中央 (20, 30, ..., 80) に目盛が来るようになります。R

```
> hist(d1$自己効力感, breaks=seq(17.5, 87.5, by=5), ylim=c(0, 60), las=1, xlab="自己効力感", main="")
```



## 2群の積み上げヒストグラム

hist(変数名, オプション群)

hist(一方の群のみのデータ\$変数名, col="色指定", add=TRUE)

全体のヒストグラムを描いた上に、一方の群のヒストグラムを重ね合わせる。

breaks=seq(a, b, by=c): x軸の値を, aからbまで増分cで区切っていく

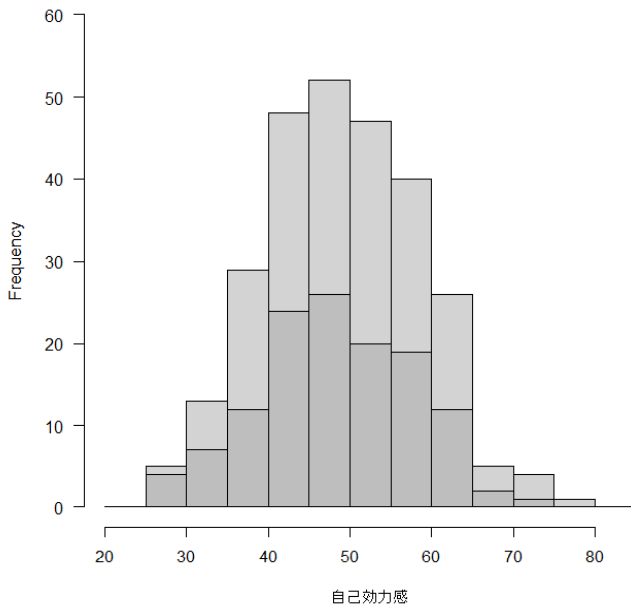
ylim=c(a, b): y軸の範囲をaからbまでとする。

las=1: 目盛りの値を水平にする。

xlab="aa ": x軸のラベルをaaとする。

```
> setwd("d:¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学  女         高          49         23 就職
2    2      20Y2 心理学  男         低          57         29 就職
3    3      20Y1  医学  女         高          42         23 進学
4    4      20Y1 看護学  女         高          41         23 就職
5    5      20Y2  医学  男         低          41         22 就職
6    6      20Y1 心理学  女         低          47         24 就職
>
>
> # 男だけのデータ
> d1m <- d1[d1$性別=="男",]
>
>
> # 全体のヒストグラムに、男だけのヒストグラムを重ね合わせ
> par(mar=c(5, 5, 3, 1))
> hist(d1$自己効力感, breaks=seq(20, 85, by=5), ylim=c(0, 60), las=1, xlab="自己効力感", main="")
> hist(d1m$自己効力感, col="gray", add=TRUE)
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職



## 箱ひげ図

## 1変数の箱ひげ図

boxplot(変数名)

## 群別の箱ひげ図

boxplot(変数名~群変数名, data=データフレーム名)

## クラスごとの群別箱ひげ図

boxplot(変数名~群変数名, data=データフレーム名, subset=クラス変数名==クラス値)

## 対応のある変数の箱ひげ図

boxplot(対応のある変数を入れたデータフレーム名)

```
> setwd("d:¥¥")
> d1 <- read.table("箱ひげ図_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
>
```

## #1変数の箱ひげ図

boxplot(d1\$pre)

&gt;

## #群別の箱ひげ図

boxplot(pre~seibetsu, data=d1)

&gt;

## #クラスごとの群別箱ひげ図

boxplot(pre~seibetsu, data=d1, subset=class=="A", main="class=A")

boxplot(pre~seibetsu, data=d1, subset=class=="B", main="class=B")

&gt;

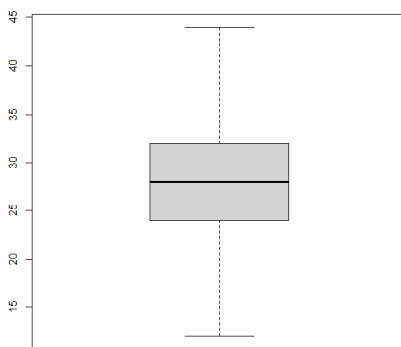
## #対応のある変数の箱ひげ図

d2 &lt;- d1[,c("pre", "post1", "post2")]

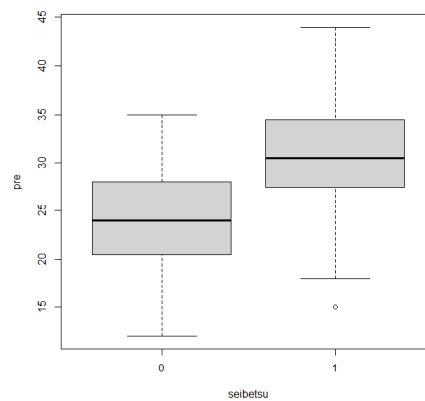
boxplot(d2)

	A	B	C	D	E
1	class	seibetsu	pre	post1	post2
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	28	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

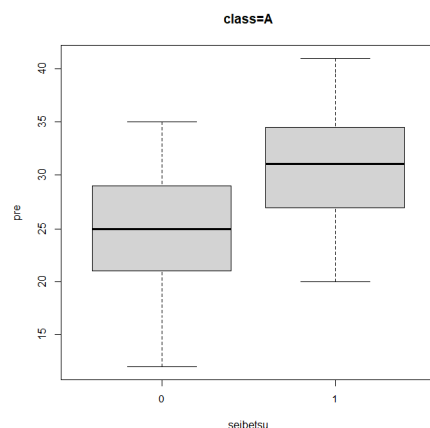
1変数の箱ひげ図



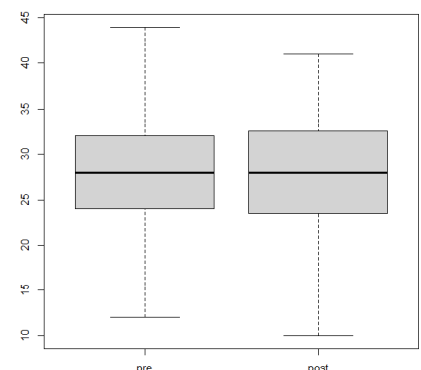
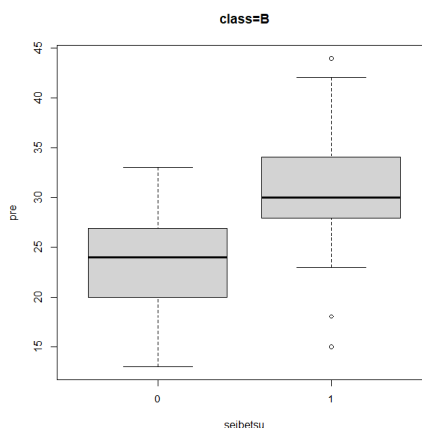
群別の箱ひげ図



クラスごとの群別箱ひげ図



対応のある変数の箱ひげ図



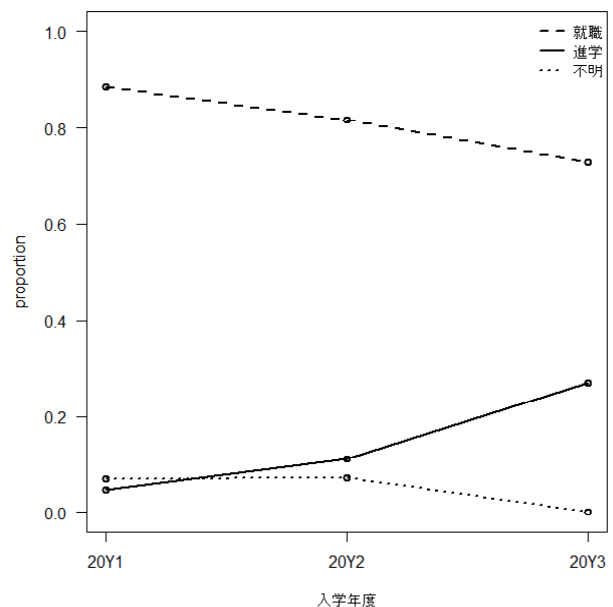
## 折れ線グラフ

plot(変数ベクトル名, type="タイプ名", オプション名)

type : 1…折れ線, b…○と線分(分離), o…○と線分(結合), c…分離した折れ線  
 lty : 線種. 1実線. 2破線. 3点線.  
 lwd : 線の太さ.  
 xaxt="n" : x軸とラベルを表示しない.  
 yaxt="n" : y軸とラベルを表示しない.

```
> setwd("d:¥¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学  女      高          49        23 就職
2    2      20Y2 心理学  男      低          57        29 就職
3    3      20Y1  医学  女      高          42        23 進学
4    4      20Y1 看護学  女      高          41        23 就職
5    5      20Y2  医学  男      低          41        22 就職
6    6      20Y1 心理学  女      低          47        24 就職
>
> # 折れ線グラフ
> par(mar=c(5,5,3,3))
> t1 <- table(d1$進路, d1$入学年度)
> p1 <- prop.table(t1, 2)
> plot(p1["就職",], type="o", ylim=c(0,1), las=1, xlab="入学年度", ylab="proportion", lty=2, lwd=
2, xaxt="n")
> par(new=TRUE)
> plot(p1["進学",], type="o", ylim=c(0,1), xlab="", ylab="", lty=1, lwd=2, xaxt="n", yaxt="n")
> par(new=TRUE)
> plot(p1["不明",], type="o", ylim=c(0,1), xlab="", ylab="", lty=3, lwd=2, xaxt="n", yaxt="n")
> axis(1, c(1,2,3), colnames(p1))
>
> # 凡例
> text(3, 1.0, "就職")
> text(3, 0.96, "進学")
> text(3, 0.92, "不明")
> segments(2.8, 1.0, 2.9, 1.0, lty=2, lwd=2)
> segments(2.8, .96, 2.9, .96, lty=1, lwd=2)
> segments(2.8, .92, 2.9, .92, lty=3, lwd=2)
>
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2		1 20Y1	看護学	女	高	49	23	就職
3		2 20Y2	心理学	男	低	57	29	就職
4		3 20Y1	医学	女	高	42	23	進学
5		4 20Y1	看護学	女	高	41	23	就職
6		5 20Y2	医学	男	低	41	22	就職
7		6 20Y1	心理学	女	低	47	24	就職
8		7 20Y3	看護学	女	高	41	26	就職
9		8 20Y2	医学	男	低	43	29	就職
10		9 20Y2	看護学	女	低	54	21	就職
11		10 20Y3	医学	女	高	55	30	進学
12		11 20Y1	心理学	男	低	66	25	就職
13		12 20Y1	心理学	女	低	38	30	就職
14		13 20Y1	心理学	女	低	45	29	就職
15		14 20Y1	医学	男	高	53	23	就職
16		15 20Y1	心理学	男	高	47	25	就職



## 散布図

### 2変数の散布図

`plot(x軸変数名, y軸変数名, pch=マーク番号(または"記号"))`

### 多変量の散布図

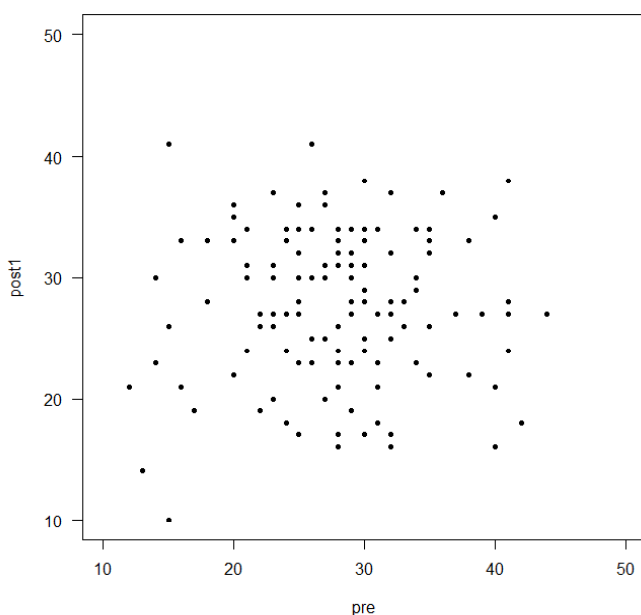
`pairs(データフレーム (または行列) 名, pch=マーク番号(または"記号"))`

```
> setwd("d:¥¥¥")
> d1 <- read.table("グラフ_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  class seibetsu pre post1 post2
1     A         0  29    34    30
2     A         0  28    24    32
3     A         1  30    24    33
4     A         1  30    28    31
5     A         1  33    26    34
6     A         1  29    28    34
>
```

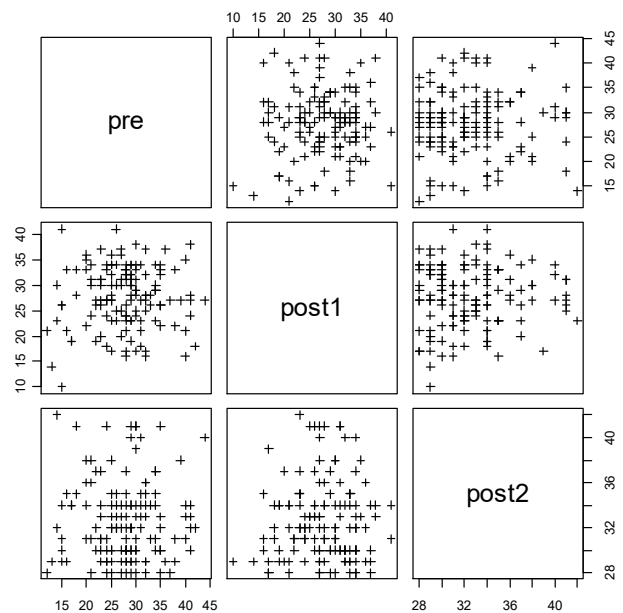
	A	B	C	D	E
1	class	seibetsu	pre	post1	post2
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

```
> #2変数の散布図
> plot(d1$pre, d1$post1, pch=20, las=1, xlim=c(10,50), ylim=c(10,50),
+      xlab="pre", ylab="post1")
>
>
> #多変量の散布図
> d2 <- d1[,c("pre", "post1", "post2")]
> pairs(d2, pch=3)
>
```

2変数の散布図



多変量の散布図





## ひまわり図

座標の重なりを花卉で表示した散布図

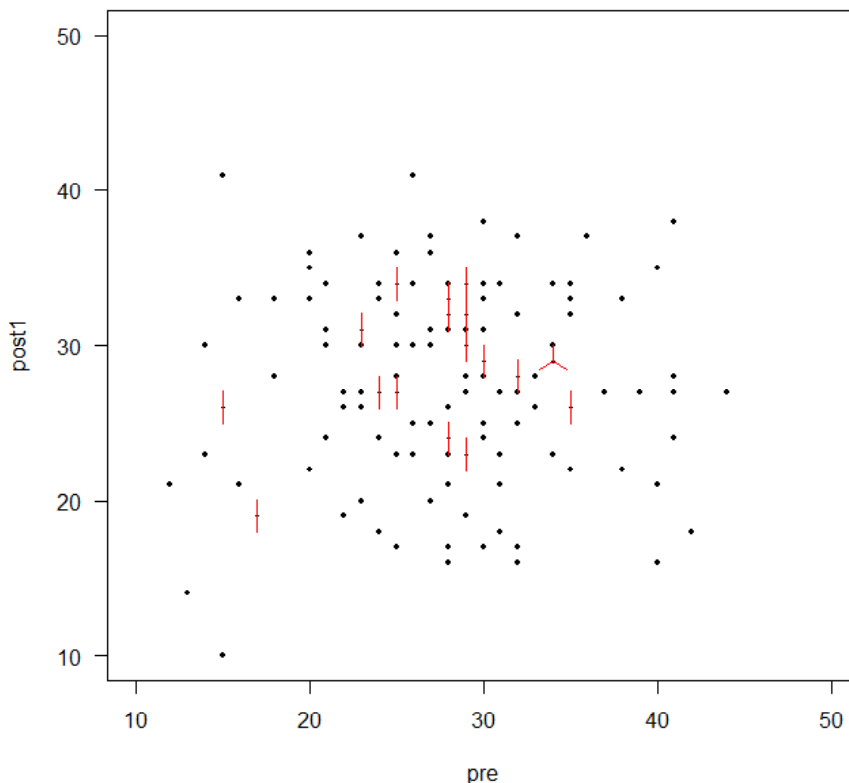
sunflowerplot(x軸変数名, y軸変数名, pch=マーク番号(または"記号"))

```

> setwd("d:¥¥¥")
> d1 <- read.table("グラフ_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  class seibetsu pre post1 post2
1     A         0 29    34    30
2     A         0 28    24    32
3     A         1 30    24    33
4     A         1 30    28    31
5     A         1 33    26    34
6     A         1 29    28    34
>

> sunflowerplot(d1$pre, d1$post1, pch=20, las=1, xlim=c(10,50),
+               ylim=c(10,50), xlab="pre", ylab="post1")

```



	A	B	C	D	E
1	class	seibetsu	pre	post1	post2
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

## バブルプロット

座標の重なりを円の大きさで表示した散布図

`symbols(x軸変数名, y軸変数名, circles=円の半径, inches=小さな値, bg="塗りつぶす色")`

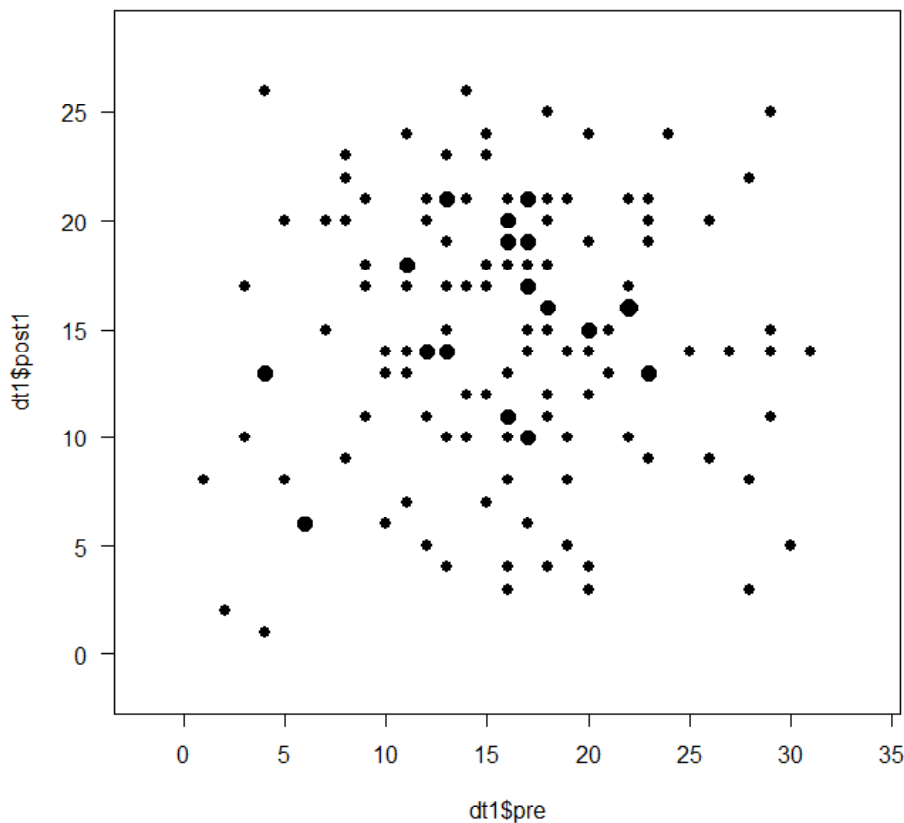
```
> setwd("d:¥¥")
> d1 <- read.table("グラフ_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  class seibetsu pre post1 post2
1     A         0  29   34   30
2     A         0  28   24   32
3     A         1  30   24   33
4     A         1  30   28   31
5     A         1  33   26   34
6     A         1  29   28   34
>
```

## # フラットなテーブル

```
> ft1 <- ftable(d1[,c("pre", "post1")], row.vars=c("pre", "post1"))
> dt1 <- as.data.frame(ft1)
> dt1 <- dt1[dt1$Freq>0,]          # 度数が 0 の座標を削除
```

## # バブルプロット

```
> symbols(dt1$pre, dt1$post1, circles=sqrt(dt1$Freq/pi), inches=0.07, bg="black", las=1)
```



	A	B	C	D	E
1	class	seibetsu	pre	post1	post2
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

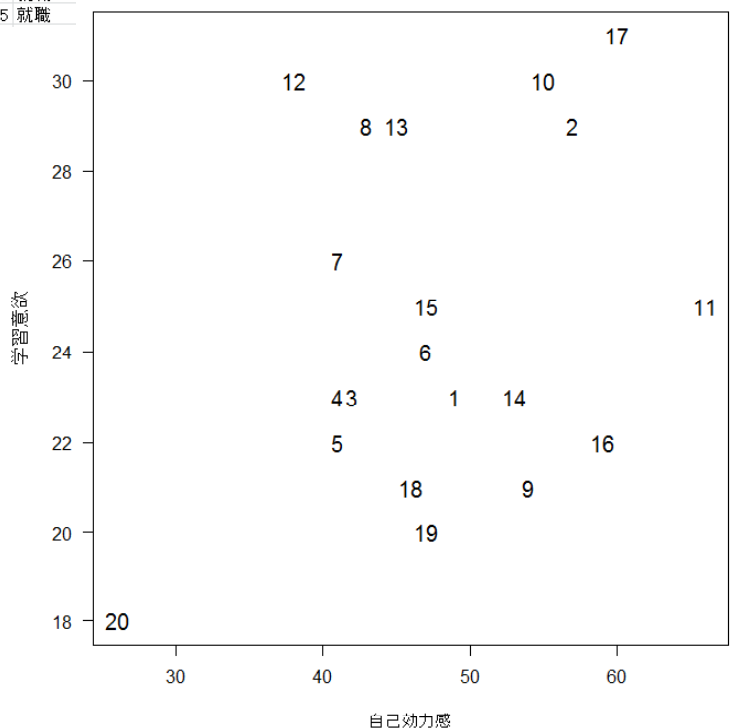
## ラベルつき散布図

```
plot(x軸変数名, y軸変数名, pch="", オプション)
```

```
text(x軸変数名, y軸変数名, labels=ラベル変数名)
```

```
> setwd("d:¥¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学   女         高          49        23 就職
2    2      20Y2 心理学   男         低          57        29 就職
3    3      20Y1  医学   女         高          42        23 進学
4    4      20Y1 看護学   女         高          41        23 就職
5    5      20Y2  医学   男         低          41        22 就職
6    6      20Y1 心理学   女         低          47        24 就職
>
>
>
> # 見やすくするため, 番号1~20のデータのみ抽出
> d2 <- d1[d1$番号<= 20,]
>
>
> # 散布図の作成
> par(mar=c(4, 4, 1, 1))
>
> # プロットエリアの作成
> plot(d2$自己効力感, d2$学習意欲, las=1, pch="", xlab="自己効力感", ylab="学習意欲")
>
> # ラベルの座標表示
> text(d2$自己効力感, d2$学習意欲, labels=d2$番号, cex=1.25) # cex ラベル文字の大きさ調整
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職



## グラフィックオプション

### par関数を用いて指定

「ヘルプ」→「Rの関数(テキスト)」で「par」を検索すると、グラフに関するさまざまなオプションが表示される。以下はその中の一部である。

図の配列 `par(mfrow=c(行数, 列数))` 図を何行何列に配列するかを指定する。

余白の行数 `par(mar=c(5, 5, 3, 1))` など。 `c()` の中は、下、左、上、右の行数を指定

グラフの重ね合わせ `plot(グラフ1); par(new=TRUE); plot(グラフ2)`

Macで日本語を表示させる `par(family="フォント名")` (フォント名はHiraKakuProN-W3などを指定)

### 個々の作図関数のオプションとして指定

関数によって使えるオプションと使えないオプションがある。

マーク `pch=マーク番号` または“記号”

番号 (0~25, 33~126) により, ○, ●, △, ▲, □, ■, ◆, + などの記号が選べる。

色 `col="色名"`

`colors()` とすると色名がたくさん表示される。black, blue, gray, green, pink, red, yellow など。

タイトル `main="タイトル"`

軸ラベル `xlab="x軸ラベル", ylab="y軸ラベル"`

軸ラベルの向き `las=0, 1, 2, or 3` (軸に平行, 水平, 軸に垂直, 垂直)

軸の表示 `axes=TRUE` (デフォルト)。FALSEにすると軸を自動的に書かない。

枠の表示 `box()` または `frame.plot=TRUE` (デフォルト)。枠線を描く。

軸の範囲 `xlim=c(下限, 上限), ylim=c(下限, 上限)`

軸を書く `axis()` 自動設定しないで手動で書く。

目盛りや目盛りの値を指定できる。side=1:x軸下, 2:y軸左, 3:x軸上, 4:y軸右

棒の幅 `width=幅の大きさ` (xlim, ylimを指定時に有効)

横書き `horiz=TRUE` (デフォルトはFALSE)

並列配置 `beside=TRUE` (FALSEにすると積み上げグラフ)

凡例 `legend=TRUE` (FALSEにすると凡例非表示)

グラフの重ね合わせ `add=TRUE` (hist関数など)

`matplot(グラフ1); matpoints(グラフ2);` とする方法もある。

他にもいろいろなオプションがある。

```
> setwd("d:¥¥¥")
> d1 <- read.table("グラフ_データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
>
```

> #データの準備

```
> d1a <- d1[d1$class=="A",]
```

```
> d1b <- d1[d1$class=="B",]
```

```
>
```

```
> m3 <- sapply(d1[,c("pre", "post1", "post2")], tapply, list(d1$seibetsu, d1$class), mean)
```

```
> rownames(m3) <- c("A0", "A1", "B0", "B1")
```

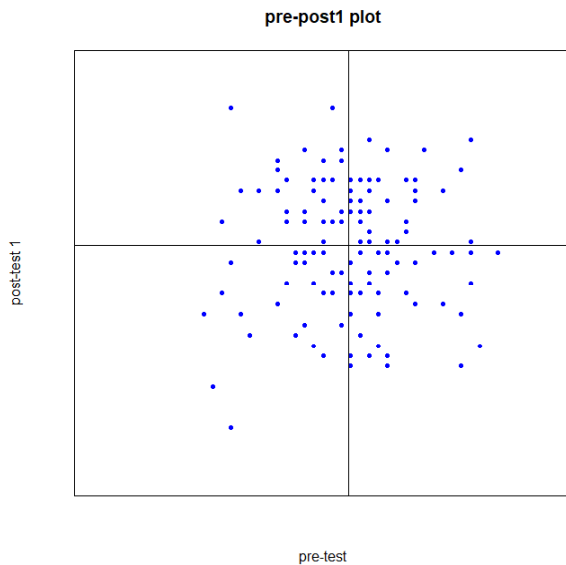
```
> m4 <- t(m3)
```

```
> m4
```

	A	B	C	D	E
1	class seibetsu pre post1 post2				
2	A	0	29	34	30
3	A	0	28	24	32
4	A	1	30	24	33
5	A	1	30	28	31
6	A	1	33	26	34
7	A	1	29	28	34
8	A	0	24	33	28
9	A	1	35	32	30
10	A	1	38	33	29
11	A	0	27	25	33
12	A	1	21	30	38
13	A	1	38	22	31
14	A	1	41	24	32
15	A	0	30	31	41
16	A	1	30	25	41
17	A	0	30	34	29
18	A	1	34	34	37
19	A	1	27	36	32
20	A	0	20	36	36

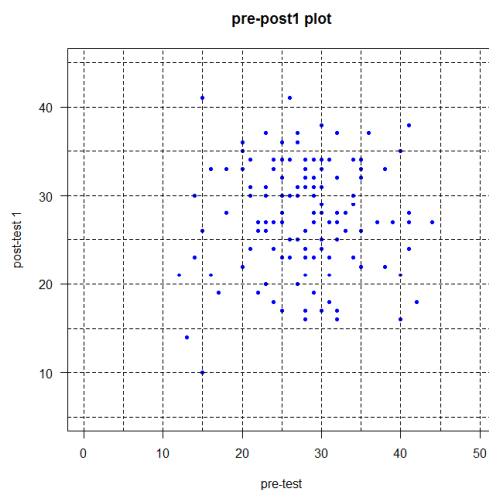
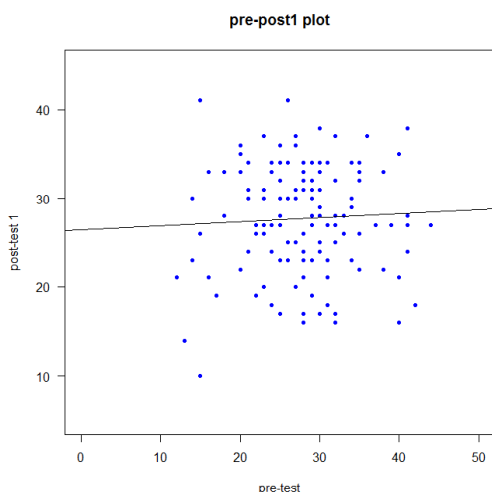
	A0	A1	B0	B1
pre	24.56	30.71429	23.20588	31.09756
post1	28.64	27.77143	27.14706	27.46341
post2	32.56	32.94286	32.52941	32.51220

```
> #マーク, 色, タイトル, 軸ラベル, 軸の範囲の設定, 軸ラベルの非表示, 線分の挿入
> plot(d1$pre, d1$post1, pch=20, col="blue", las=1, axes=FALSE, frame.plot=TRUE,
+      main="pre-post1 plot", xlab="pre-test", ylab="post-test 1",
+      xlim=c(0,50), ylim=c(5,45))
> segments(mean(d1$pre), 0, mean(d1$pre), 50) # 始点と終点のx,y座標を指定して線分を描く
> segments(-10, mean(d1$post1), 60, mean(d1$post1))
```



```
> # 散布図に回帰直線を乗せる
> plot(d1$pre, d1$post1, pch=20, col="blue", las=1,
+      main="pre-post1 plot", xlab="pre-test", ylab="post-test 1",
+      xlim=c(0,50), ylim=c(5,45))
> abline(lm(post1~pre, data=d1))
```

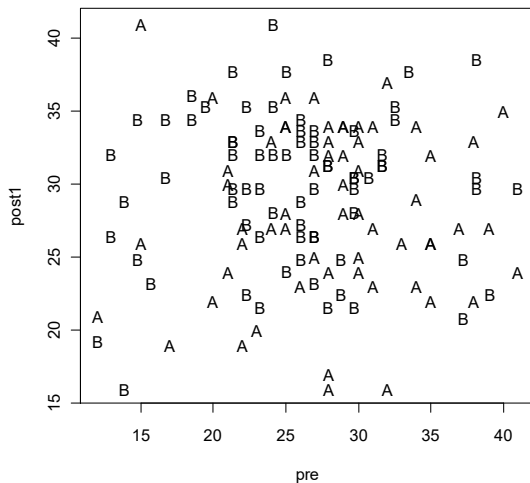
```
> # グラフに格子線を引く
> plot(d1$pre, d1$post1, pch=20, col="blue", las=1,
+      main="pre-post1 plot", xlab="pre-test", ylab="post-test 1",
+      xlim=c(0,50), ylim=c(5,45))
> x <- seq(0, 50, by=5)
> y <- seq(5, 45, by=5)
> abline(h=y, v=x, lty=2) # lty で線の種類を指定
```



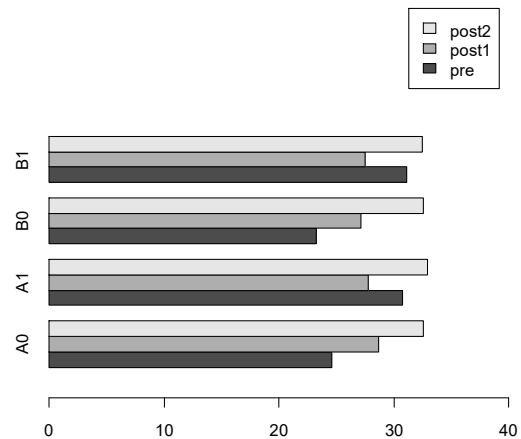
```
> #2群の散布図
> plot(dla$pre, dla$post1, pch="A", xlab="pre", ylab="post1")
> par(new=TRUE)
> plot(dlb$pre, dlb$post1, pch="B", axes=FALSE, xlab="", ylab="")
```

```
> #棒の幅, 横書き, 軸の範囲, 並列配置, 凡例の設定
> barplot(m4,width=0.5, horiz=TRUE, xlim=c(0,40), ylim=c(0,12),
+        beside=TRUE, legend=TRUE)
```

2群の散布図の重ね合わせ

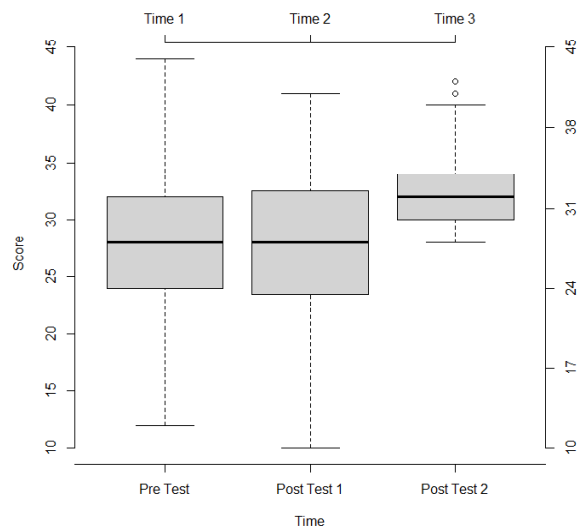


棒の幅, 横書き, 軸の範囲, 並列配置, 凡例の設定



```
> #軸を手動で書く
```

```
> # グラフの目盛りラベルを明示的に書く
> boxplot(d1[,3:5], axes=FALSE, xlab="Time", ylab="Score")
>
> # side = 1:x軸下. 1から3目盛りまでに, c()で指定したラベルを書く
> axis(side=1, 0:4, c("", "Pre Test", "Post Test 1", "Post Test 2", ""))
>
> # side = 2: y軸左
> axis(side=2)
>
> # side = 3: x軸上
> axis(side=3, 1:3, c("Time 1", "Time 2", "Time 3"))
>
> # side = 4: y軸右
> axis(side=4, seq(10, 45, by=7))
>
```



## ggplotを使った作図

ggplot2というパッケージにあるggplotという関数を使うと、より複雑できれいなグラフを描くことができる。基本的には、ggplot()関数のうしろに、必要な関数を + でつなげて描画する。

## ggplot2パッケージをインストールする

```
library(ggplot2)
```

パッケージを読み込む

```
ggplot(data=データ名, aes())
```

グラフを描くキャンパスの大枠を設定する

aes() 内のオプション

x = x軸変数名

y = y軸変数名 (指定しなければ、x軸変数の度数がy軸になる)

group = 群分け変数名

color = 色分け変数名 (線, ドット)

fill = 色分け変数名 (領域)

shape = 記号分け変数名 (ドット)

```
geom_bar()
```

棒グラフ

stat = "identity": 値そのもので棒の長さを指定する。他に"summary" (平均) などがある

position = "dodge": 群別に棒 (領域) を分ける

position = "fill": 帯グラフ化する (x軸の値ごとに、y軸を最大幅まで取る)

```
geom_errorbar(aes())
```

エラーバー

aes() 内のオプション

ymin = エラーバーの上限

ymin = エラーバーの下限

width = エラーバーの横棒の幅

position = position\_dodge(width = 0.9): 複数のエラーバー間の間隔を調整

```
geom_histogram()
```

ヒストグラム

breaks = seq(上限, 下限, by=階級幅)

bins = 棒の数

```
geom_boxplot()
```

箱ひげ図

```
geom_line()
```

折れ線グラフ

linetype = 値: 線種

linewidth = 値: 線の太さ

```
geom_point()
```

散布図

position = position\_jitter(): ドットの重なりをずらす

```
geom_smooth()
```

近似曲線

method = "lm": 回帰直線

se=F: 誤差の表示 (TRUE or FALSE)

```
xlab("ラベル名")
```

x軸のラベルを設定する

```
ylab("ラベル名")
```

y軸のラベルを設定する

```
labs()
```

ラベルを細かく設定する

title="タイトル"

subtitle="サブタイトル"

x="x軸ラベル": xlab()と同じ役割

y="y軸ラベル": ylab()と同じ役割

caption="キャプション"

fill="凡例": aesにおいて fill で色分けした場合

color="凡例": aesにおいて color で色分けした場合

`scale_y_continuous()`

y軸の範囲を設定する

`limits = c(下限値, 上限値)` : 下限と上限を設定する。データの範囲を超えるとエラーになる場合がある

`labels = scales::percent` : y軸をパーセント表示にする

`theme_bw()`, `theme_classic()`, `theme_light`, `theme_test()` など

背景色や目盛の表示などを設定する

カッコ内でオプションを指定することもできる

`facet_grid()` ( ~群分け変数名 )

群別に図を描く

`nrow = 値` : 指定された行数に収まるように配列する

`ncol = 値` : 指定された列数に収まるように配列する

`facet_grid()`

群別に図を描く

`rows=vars(群分け変数名)` : 群ごとの図を **行** を変えて表示する

`cols=vars(群分け変数名)` : 群ごとの図を **列** を変えて表示する

行変数名 ~ 列変数名 : グラフを群別に分けて描いて配列表示する際の群分け変数を指定する

`coord_flip()`

x軸とy軸の入れ替え

\* スクリプトにおいて、関数を「+」でつなぐことを忘れない（コンソール画面上の「+」とは異なる）。

\* 関数の間で改行するときは、上の行の末尾に「+」をつけて、スクリプトが続くことを明示する

```
> rm(list=ls())
```

```
> setwd("d:¥¥Rdocuments¥¥scripts¥¥")
```

```
> # ggplot2パッケージの呼び出し
```

```
> library(ggplot2)
```

```
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
```

```
> head(d1)
```

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

```
>
```

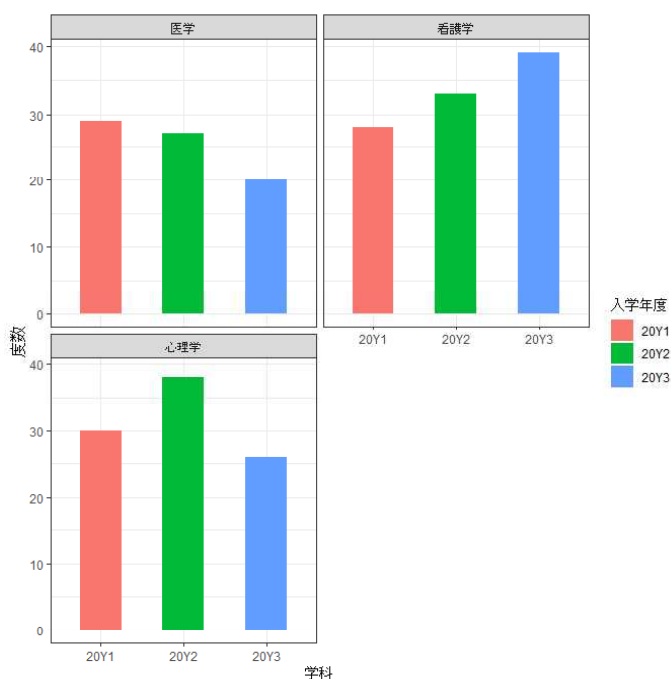
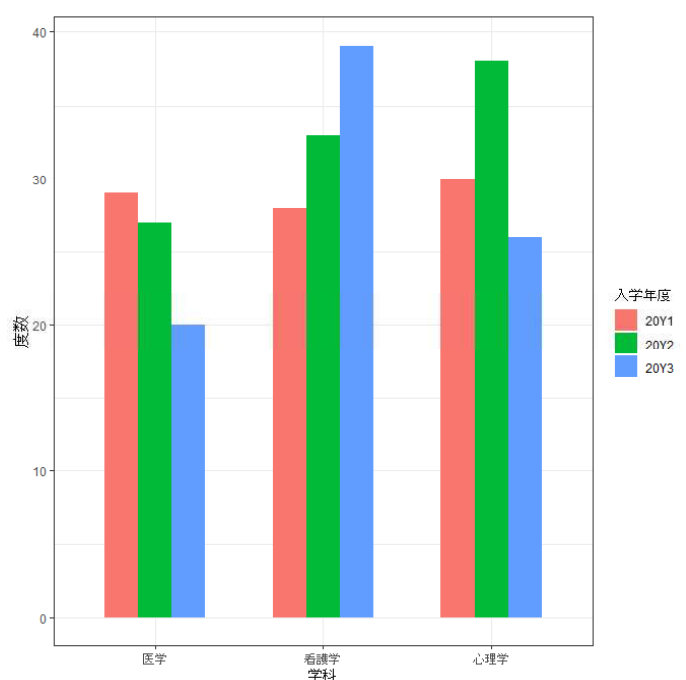
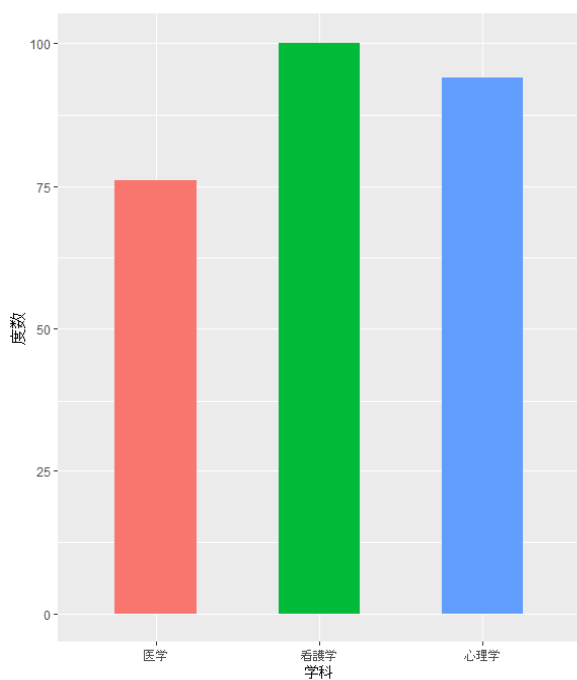


## 度数を比較する棒グラフ — ggplot

```
> ggplot(data=d1, aes(x = 学科, fill = 学科)) +
+   geom_bar(width=0.5) +
+   xlab("学科") + ylab("度数")
>

> # 入学年度別に棒を分ける
> ggplot(data=d1, aes(x = 学科, fill = 入学年度)) +
+   geom_bar(position = "dodge", width=0.6) +
+   xlab("学科") + ylab("度数")
+   theme_bw()

> # 学科別に図を分ける
> ggplot(data=d1, aes(x = 入学年度, fill = 入学年度)) +
+   geom_bar(position = "dodge", width=0.5) +
+   xlab("学科") + ylab("度数") +
+   theme_bw() +
+   facet_wrap(~ 学科, nrow=2)
```



## 平均値を比較する棒グラフ — ggplot

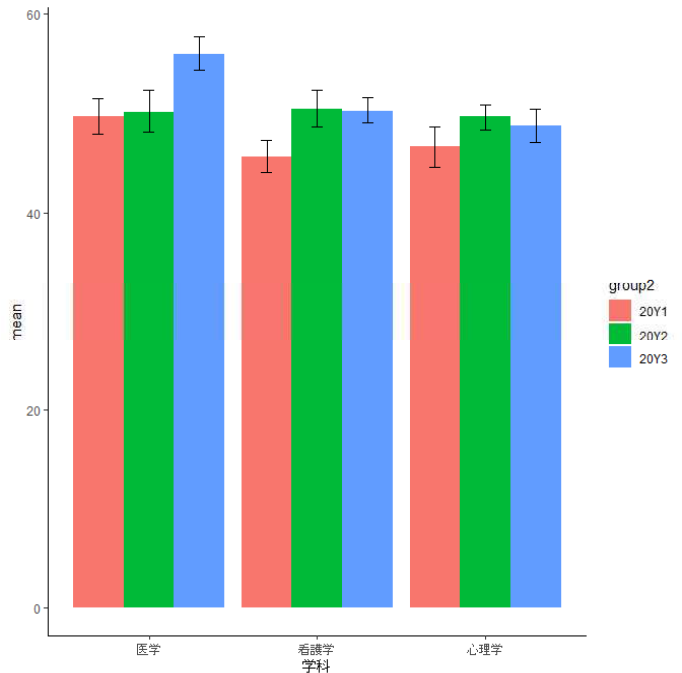
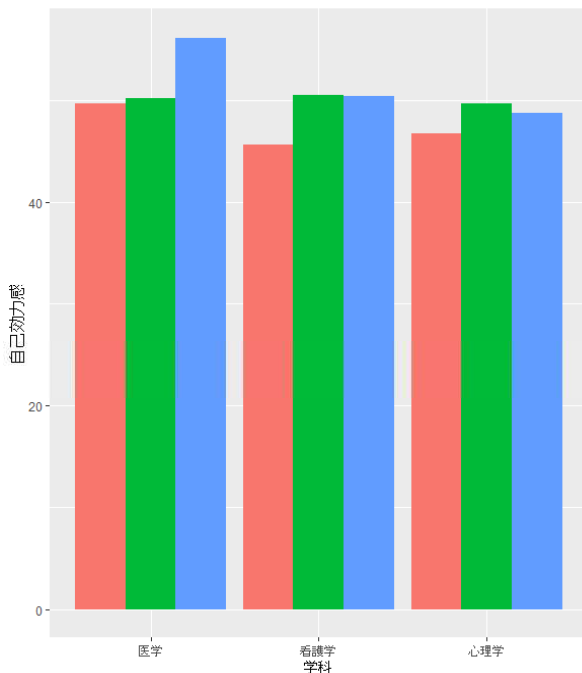
```
> ggplot(data=d1, aes(x = 学科, y = 自己効力感, fill = 入学年度)) +
+   geom_bar(stat = "summary", position = "dodge") +
+   xlab("学科")
```

## &gt; # エラーバー付棒グラフ

```
> library(psych)
> (ds1 <- describeBy(d1$自己効力感, group=list(d1$学科, d1$入学年度), mat=T, digits=2))
```

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	医学	20Y1	1	29	49.76	9.83	50.0	49.32	11.86	35	73	38	0.41	-0.83	1.83
X12	2	看護学	20Y1	1	28	45.68	8.67	45.0	44.96	5.93	33	68	35	0.97	0.51	1.64
X13	3	心理学	20Y1	1	30	46.73	10.91	47.5	46.92	14.08	26	66	40	-0.09	-1.22	1.99
X14	4	医学	20Y2	1	27	50.26	10.81	52.0	50.87	8.90	26	73	47	-0.50	-0.01	2.08
X15	5	看護学	20Y2	1	33	50.55	10.71	49.0	49.85	11.86	34	77	43	0.55	-0.43	1.86
X16	6	心理学	20Y2	1	38	49.71	7.90	48.5	49.41	9.64	38	70	32	0.37	-0.69	1.28
X17	7	医学	20Y3	1	20	56.00	7.67	54.5	55.75	8.15	41	72	31	0.20	-0.69	1.72
X18	8	看護学	20Y3	1	39	50.38	8.11	52.0	50.48	8.90	35	65	30	-0.14	-1.08	1.30
X19	9	心理学	20Y3	1	26	48.81	8.58	49.5	48.95	6.67	29	65	36	-0.28	-0.31	1.68

```
> ggplot(data=ds1, aes(x = group1, y = mean, fill = group2)) +
+   geom_bar(stat = "identity", position = "dodge") +
+   geom_errorbar(aes(ymin = mean - se, ymax = mean + se),
+   width = 0.2, position = position_dodge(width = 0.9)) +
+   xlab("学科") +
+   theme_classic()
```

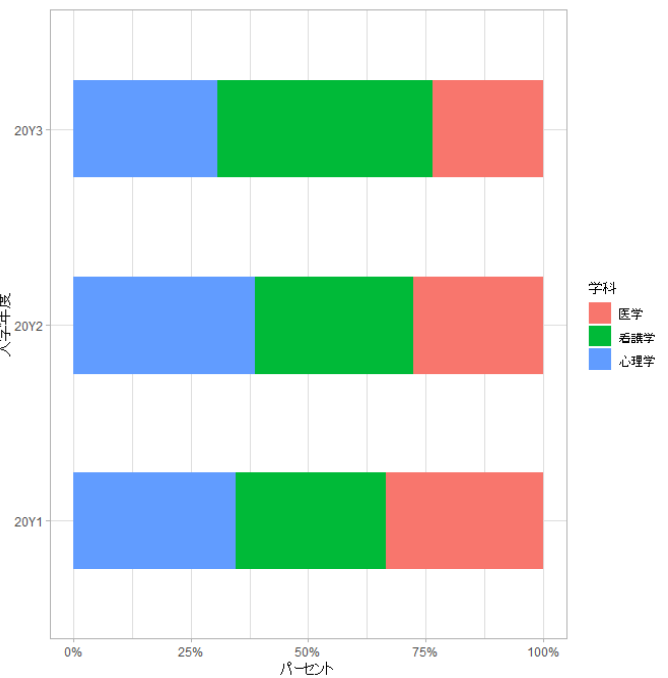
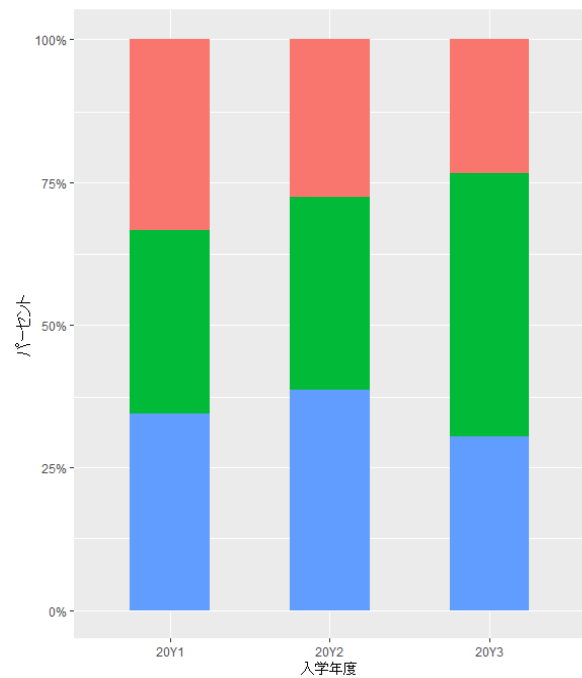


## 帯グラフ — ggplot

```
> ggplot(data=d1, aes(x = 入学年度, fill = 学科)) +
+   geom_bar(position="fill", width=0.5) +
+   ylab("パーセント") +
+   scale_y_continuous(labels = scales::percent)
```

## # 横向き

```
> ggplot(data=d1, aes(x = 入学年度, fill = 学科)) +
+   geom_bar(position="fill", width=0.5) +
+   ylab("パーセント") +
+   scale_y_continuous(labels = scales::percent) +
+   coord_flip() +
+   theme_light()
```



## ヒストグラム — ggplot

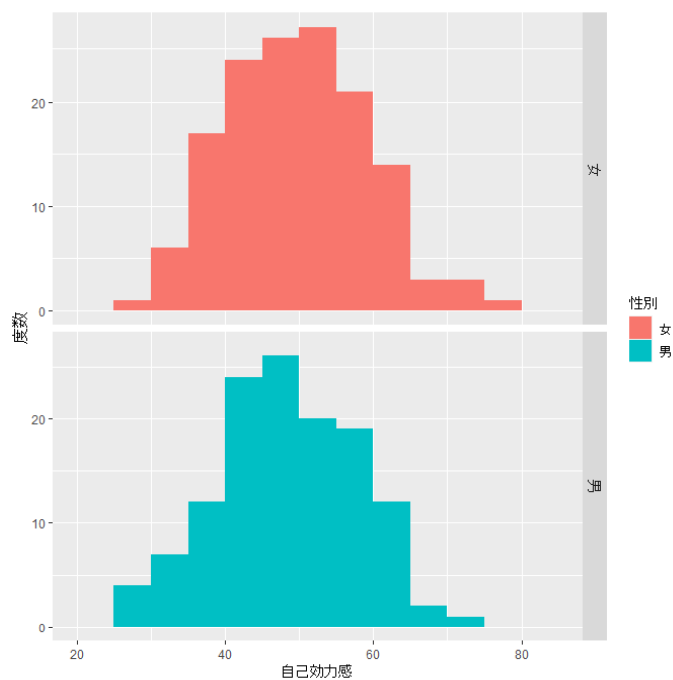
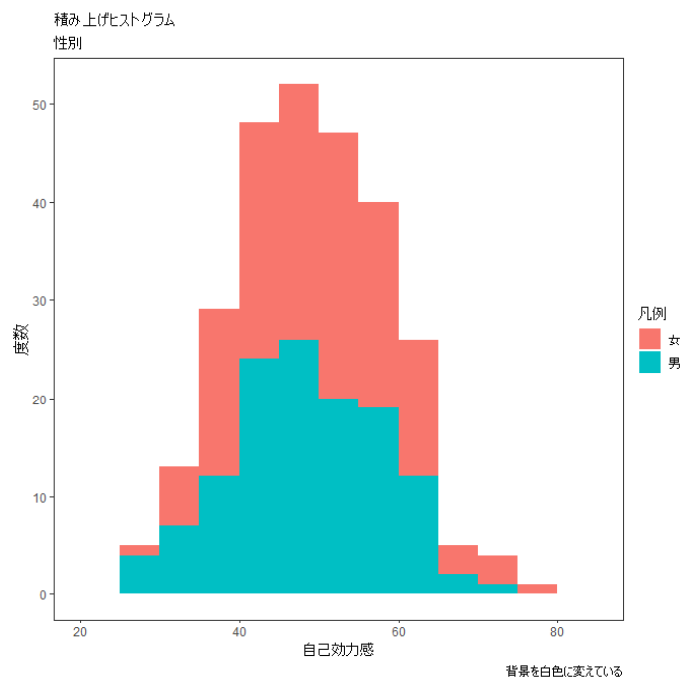
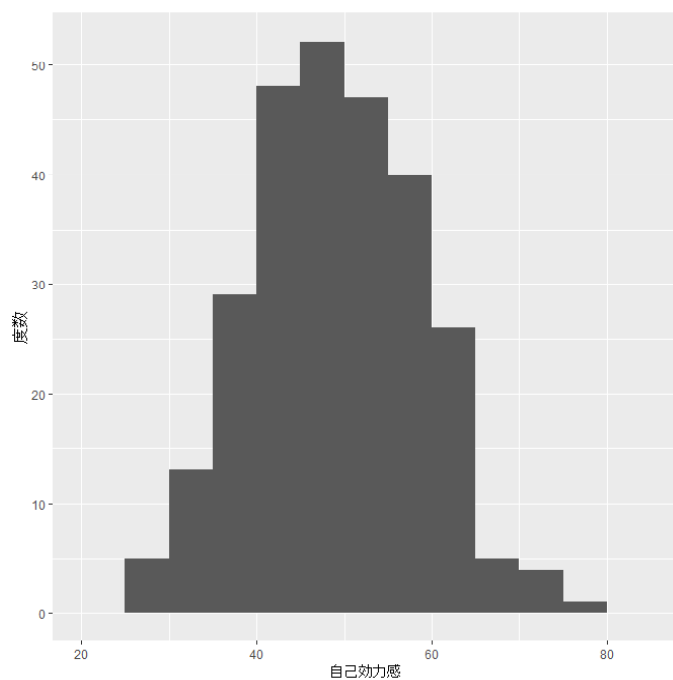
```
> ggplot(data=d1, aes(x=自己効力感)) +
+   geom_histogram(breaks=seq(20, 85, by=5)) +
+   xlab("自己効力感")+ ylab("度数")
```

## &gt; # 積み上げヒストグラム

```
> ggplot(data=d1, aes(x=自己効力感, fill=性別)) +
+   geom_histogram(breaks=seq(20, 85, by=5)) +
+   labs(title="積み上げヒストグラム", subtitle="性別", x="自己効力感", y="度数",
+         caption="背景を白色に変えている", fill="凡例") +
+   theme_test()
```

## &gt; # 群別ヒストグラム

```
> ggplot(data=d1, aes(x=自己効力感, fill=性別)) +
+   geom_histogram(breaks=seq(20, 85, by=5)) +
+   xlab("自己効力感")+ ylab("度数") +
+   facet_grid(rows=vars(性別))
```



## 箱ひげ図 — ggplot

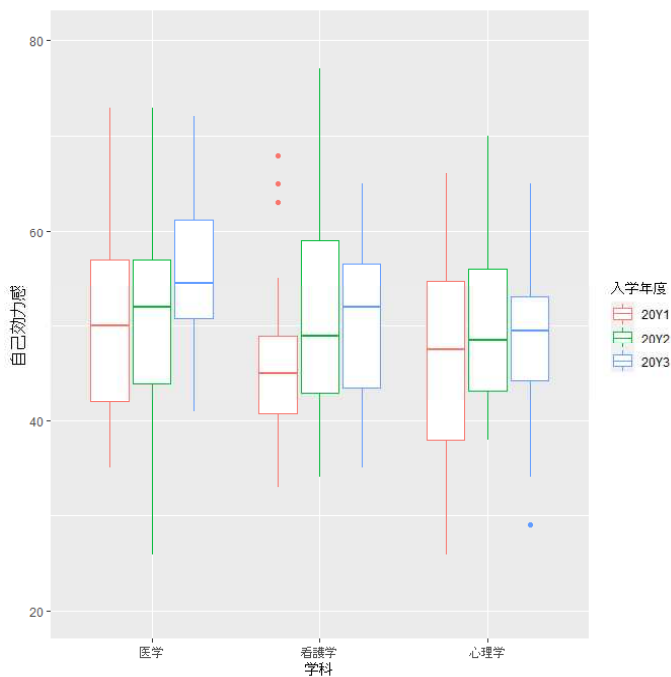
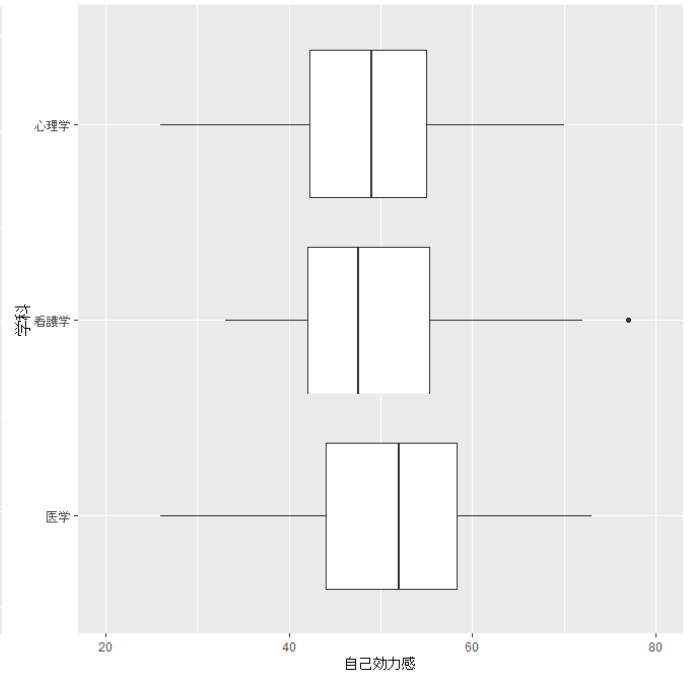
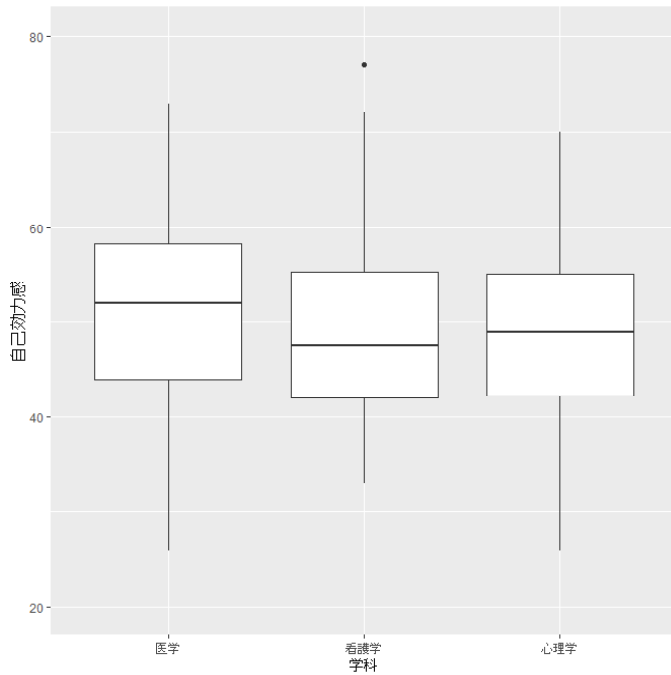
```

> ggplot(data=d1, aes(x = 学科, y = 自己効力感)) +
+   geom_boxplot() + ylim(20, 80)

> # x軸とy軸の入れ替え
> ggplot(data=d1, aes(x = 学科, y = 自己効力感)) +
+   geom_boxplot() + ylim(20, 80) + coord_flip()

> # 群分け
> ggplot(data=d1, aes(x = 学科, y = 自己効力感, color=入学年度)) +
+   geom_boxplot() + ylim(20, 80)

```



## 折れ線グラフ — ggplot

```
> # フラットなテーブルを作成してデータフレーム化
```

```
> (ft1 <- as.data.frame(ftable(d1[,c("進路","入学年度")], row.vars=c("進路","入学年度"))))
```

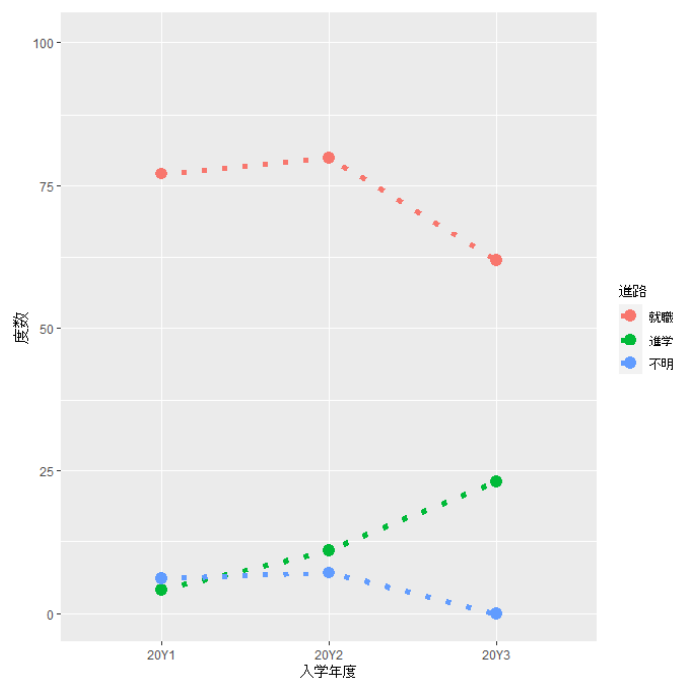
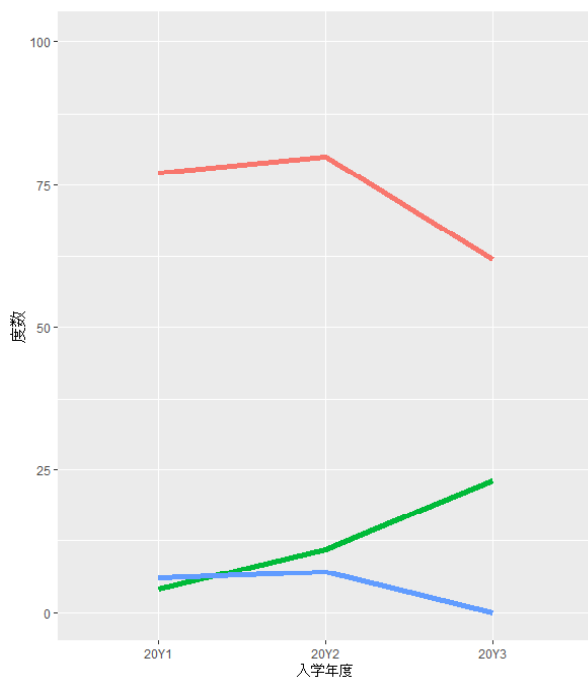
	進路	入学年度	Freq
1	就職	20Y1	77
2	進学	20Y1	4
3	不明	20Y1	6
4	就職	20Y2	80
5	進学	20Y2	11
6	不明	20Y2	7
7	就職	20Y3	62
8	進学	20Y3	23
9	不明	20Y3	0

```
> # 折れ線グラフ
```

```
>
> ggplot(data=ft1, aes(x=入学年度, y=Freq, group=進路, color=進路)) +
+   geom_line(linewidth=2) +
+   xlab("入学年度")+ ylab("度数") +
+   scale_y_continuous(limits = c(0, 100))
>
```

```
> # 線種を変え、点をつける
```

```
> ggplot(data=ft1, aes(x=入学年度, y=Freq, group=進路, color=進路)) +
+   geom_line(linewidth=2, linetype=3) + geom_point(size=4) +
+   xlab("入学年度")+ ylab("度数") +
+   scale_y_continuous(limits = c(0, 100))
```



## 散布図 — ggplot

```
> ggplot(data=d1, aes(x = 自己効力感, y = 学習意欲, color = 性別)) +  
+   geom_point()
```

```
# ドットの重なりをずらした散布図
```

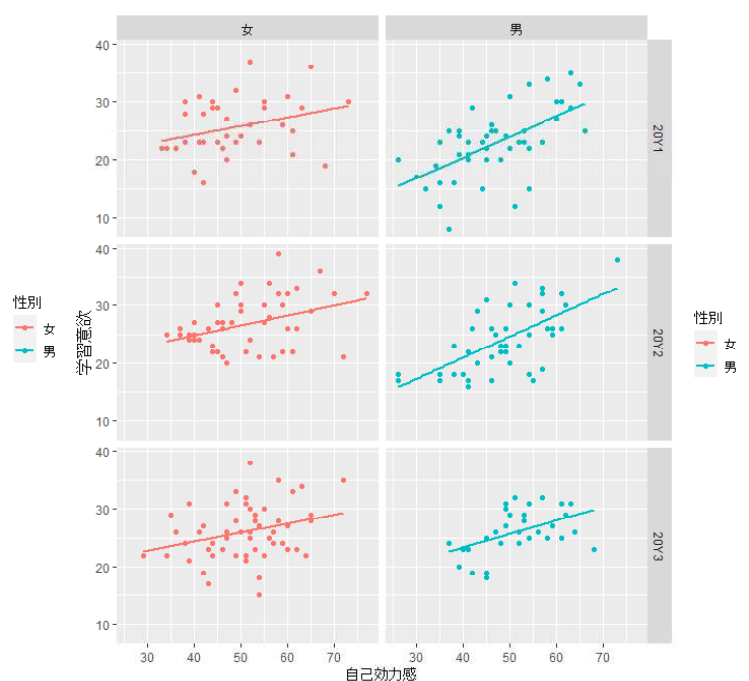
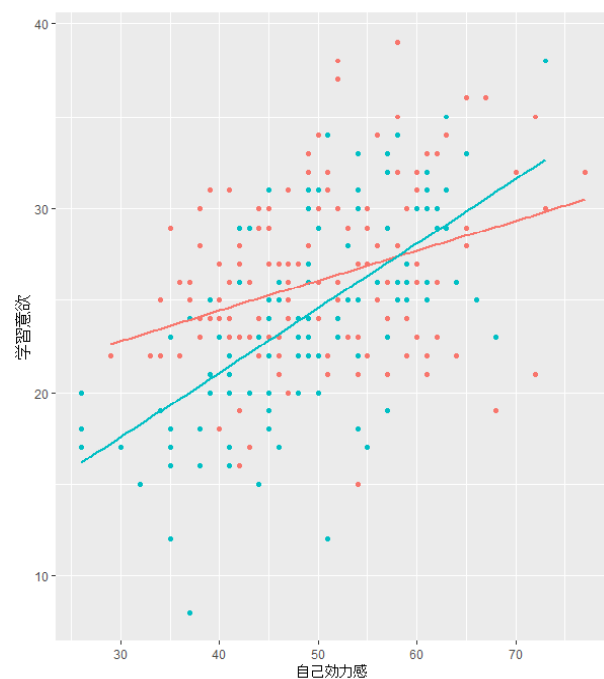
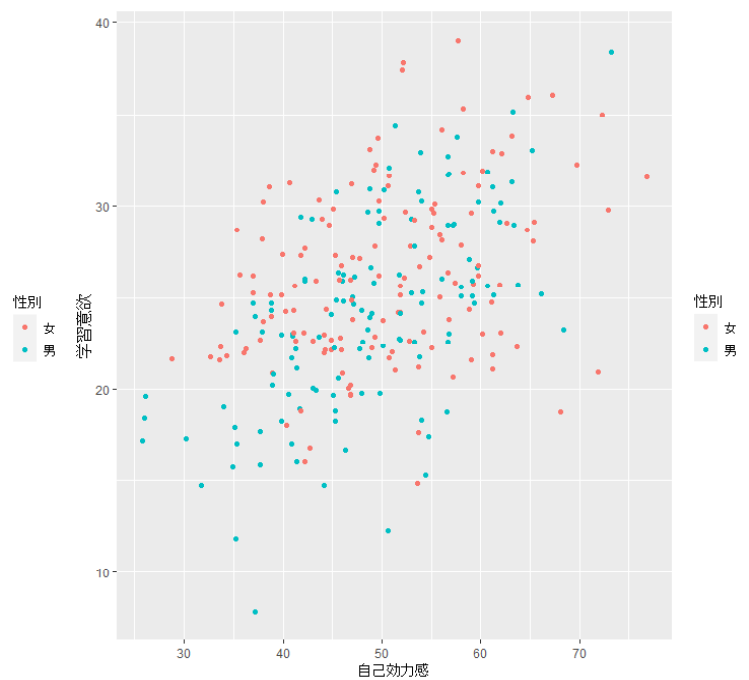
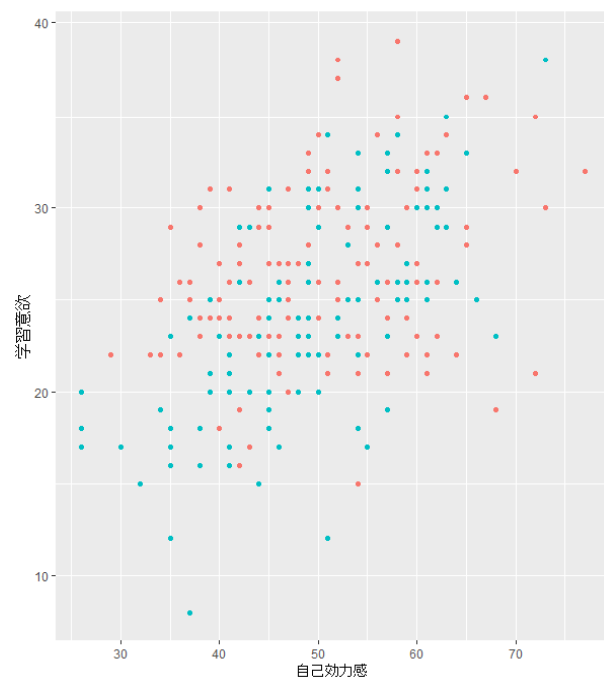
```
> ggplot(data=d1, aes(x = 自己効力感, y = 学習意欲, color= 性別)) +  
+   geom_point(position = position_jitter())
```

```
> # 回帰直線を乗せる
```

```
> ggplot(data=d1, aes(x = 自己効力感, y = 学習意欲, color= 性別)) +  
+   geom_point() + geom_smooth(method = "lm", se=F)
```

```
> # 年度別, 性別に図を分ける
```

```
> ggplot(data=d1, aes(x = 自己効力感, y = 学習意欲, color=性別)) +  
+   geom_point() + geom_smooth(method = "lm", se=F) +  
+   facet_grid(入学年度~性別)
```



## 6 量的変数の要約統計量



平均, 標準偏差, 最大値, 中央値, 最小値, 分位数

```
summary(データフレーム名)      または      apply(データフレーム名, 2, summary)
colMeans(データフレーム名)     または      apply(データフレーム名, 2, mean)
sapply(データフレーム名, sd)   または      apply(データフレーム名, 2, sd)
```

summary : 最小値, 四分位数, 最大値, 平均値  
colMeans : 各列の平均値  
mean : 平均値  
sd : 標準偏差  
max : 最大値  
median : 中央値  
min : 最小値  
quantile(値) : 値で指定された累積比率のところのデータ  
na.rm=TRUE : mean や sd などにおいて, 欠測値があるときはそれを除外する指定

```
> setwd("d:¥¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度 学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1    20Y1 看護学 女      高      49      23 就職
2    2    20Y2 心理学 男      低      57      29 就職
3    3    20Y1 医学 女      高      42      23 進学
4    4    20Y1 看護学 女      高      41      23 就職
5    5    20Y2 医学 男      低      41      22 就職
6    6    20Y1 心理学 女      低      47      24 就職
>

> #分析に必要な変数だけを取り出す
> dtmp <- d1[,c("自己効力感","学習意欲")]
>
```

```
> # 標本サイズ
> (ntmp <- nrow(dtmp))
[1] 270
>
```

```
> # 各変数の平均
> (mtmp <- colMeans(dtmp))
自己効力感  学習意欲
  49.60000   25.20741
>
```

```
> # 各変数の標準偏差
> (sd.d2 <- apply(dtmp, 2, sd))
自己効力感  学習意欲
  9.492866   5.099160
>
```

```
> # 各変数の最小値, 四分位数, 最大値
> (summarytmp <- summary(dtmp))
  自己効力感  学習意欲
Min.   :26.0   Min.   : 8.00
1st Qu.:42.0   1st Qu.:22.00
Median :49.0   Median :25.00
Mean   :49.6   Mean   :25.21
3rd Qu.:57.0   3rd Qu.:29.00
Max.   :77.0   Max.   :39.00
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

群別の平均, 標準偏差, 最大値, 中央値, 最小値, 分位数

集計したい変数が1つだけの場合

`tapply`(集計したい変数名, 群分け変数名, 関数, 関数のオプション群)

集計したい変数は1つしか指定できない。

集計したい変数が1つ以上の場合

`by`を使う方法

`by`(データフレーム名, 群分け変数名, 関数, 関数のオプション群)

`aggregate`を使う方法

`aggregate`(データフレーム名, `list`(群分け変数名), 関数, 関数のオプション群)

`sapply`と`tapply`を組み合わせて使う方法

`sapply`(データフレーム名, `tapply`, 群分け変数名, 関数, 関数のオプション群)

データフレームは, 集計したい変数だけを入れたものにするか, 集計したい変数を指定する。

```
> setwd("d:¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学  女          高          49        23 就職
2    2      20Y2 心理学  男          低          57        29 就職
3    3      20Y1  医学  女          高          42        23 進学
4    4      20Y1 看護学  女          高          41        23 就職
5    5      20Y2  医学  男          低          41        22 就職
6    6      20Y1 心理学  女          低          47        24 就職
>
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2		1 20Y1	看護学	女	高	49	23	就職
3		2 20Y2	心理学	男	低	57	29	就職
4		3 20Y1	医学	女	高	42	23	進学
5		4 20Y1	看護学	女	高	41	23	就職
6		5 20Y2	医学	男	低	41	22	就職
7		6 20Y1	心理学	女	低	47	24	就職
8		7 20Y3	看護学	女	高	41	26	就職
9		8 20Y2	医学	男	低	43	29	就職
10		9 20Y2	看護学	女	低	54	21	就職
11		10 20Y3	医学	女	高	55	30	進学
12		11 20Y1	心理学	男	低	66	25	就職
13		12 20Y1	心理学	女	低	38	30	就職
14		13 20Y1	心理学	女	低	45	29	就職
15		14 20Y1	医学	男	高	53	23	就職
16		15 20Y1	心理学	男	高	47	25	就職

> # 各群の標本サイズ

> `table(d1$入学年度)`

20Y1 20Y2 20Y3

87 98 85

>

> # `tapply`を使う方法

> `tapply(d1$学習意欲, d1$入学年度, summary)`

\$`20Y1`

Min. 1st Qu. Median Mean 3rd Qu. Max.  
8.00 21.50 23.00 24.08 28.50 37.00

\$`20Y2`

Min. 1st Qu. Median Mean 3rd Qu. Max.  
16.00 22.00 25.50 25.48 29.00 39.00

\$`20Y3`

Min. 1st Qu. Median Mean 3rd Qu. Max.  
15.00 23.00 26.00 26.05 29.00 38.00

> `tapply(d1$学習意欲, d1$入学年度, sd)`

20Y1 20Y2 20Y3

5.528401 5.105545 4.445184

```

>
> # byを使う方法
> by(d1[,c("学習意欲")], d1$入学年度, summary)
d1$入学年度: 20Y1
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   8.00   21.50   23.00   24.08   28.50   37.00
-----
d1$入学年度: 20Y2
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  16.00   22.00   25.50   25.48   29.00   39.00
-----
d1$入学年度: 20Y3
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  15.00   23.00   26.00   26.05   29.00   38.00

> by(d1[,c("学習意欲")], d1$入学年度, sd)
d1$入学年度: 20Y1
[1] 5.528401
-----
d1$入学年度: 20Y2
[1] 5.105545
-----
d1$入学年度: 20Y3
[1] 4.445184
>

> # aggregateを使う方法
> aggregate(d1[,c("学習意欲")], list(d1$入学年度), summary)
  Group.1 x. Min.  x. 1st Qu.  x. Median x. Mean x. 3rd Qu.  x. Max.
1    20Y1   8.00    21.50    23.00  24.08    28.50   37.00
2    20Y2  16.00    22.00    25.50  25.48    29.00   39.00
3    20Y3  15.00    23.00    26.00  26.05    29.00   38.00

> aggregate(d1[,c("学習意欲")], list(d1$入学年度), sd)
  Group.1      x
1    20Y1 5.528401
2    20Y2 5.105545
3    20Y3 4.445184
>

> # sapply と tapply を組み合わせて使う方法
> sapply(d1[,c("学習意欲", "自己効力感")], tapply, d1$入学年度, mean)
      学習意欲 自己効力感
20Y1 24.08046   47.40230
20Y2 25.47959   50.14286
20Y3 26.04706   51.22353

> sapply(d1[,c("学習意欲", "自己効力感")], tapply, d1$入学年度, sd)
      学習意欲 自己効力感
20Y1 5.528401   9.908253
20Y2 5.105545   9.649529
20Y3 4.445184   8.516617
>
>
>
> # 群別にデータフレームを分割する
> d1Y1 <- d1[d1$入学年度=="20Y1", ]
> d1Y2 <- d1[d1$入学年度=="20Y2", ]
> d1Y3 <- d1[d1$入学年度=="20Y3", ]
>
> # 各群における記述統計量
> nrow(d1Y1)
[1] 87

> summary(d1Y1[,c("学習意欲")])
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   8.00   21.50   23.00   24.08   28.50   37.00

> sd(d1Y1[,c("学習意欲")])
[1] 5.528401

```

## 要約統計量 — describe, describeBy

```
library(psych)
describe(データフレーム名[, c("変数名1", "変数名2", ...)])
```

## 群ごとの表示

## byを使う方法

```
library(psych)
by(データフレーム名, 群分け変数名, describe)
```

## describeByを使う方法

```
library(psych)
describeBy(データフレーム名, 群分け変数名)
```

あらかじめpsychパッケージをインストールしておく必要がある。

mat=TRUEと指定すると行列表記になる。

list(群分け変数1, 群分け変数2)などとして、入れ子状に複数の群分け変数を指定することもできる。

## 出力内容

item name            変数名  
 item number        標本の大きさ  
 number of valid cases 有効数  
 mean                算術平均  
 standard deviation   標準偏差 (分母=n-1)  
 trimmed mean (with trim defaulting to .1) 調整平均(指定した割合の両端データを削除したときの平均)  
 median (standard or interpolated) 中央値  
 mad: median absolute deviation (from the median)  $1.4826 \times [\text{中央値からの絶対偏差}] \text{の中央値}$   
 minimum            最小値  
 maximum            最大値  
 skew                歪度  
 kurtosis            尖度  
 standard error      標準誤差

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

```
> setwd("d:¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  番号 入学年度 学科 性別 モラトリアム 自己効力感 学習意欲 進路
1     1     20Y1 看護学 女         高           49       23 就職
2     2     20Y2 心理学 男         低           57       29 就職
3     3     20Y1 医学   女         高           42       23 進学
4     4     20Y1 看護学 女         高           41       23 就職
5     5     20Y2 医学   男         低           41       22 就職
6     6     20Y1 心理学 女         低           47       24 就職
>
```

```
> #psych パッケージの読み込み
> library(psych)
>
```

## &gt; # 記述統計量の一括表示

&gt; describe(d1[,c("学習意欲","自己効力感")])

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	270	25.21	5.10	25	25.22	4.45	8	39	31	-0.01	0.08	0.31
自己効力感	2	270	49.60	9.49	49	49.51	10.38	26	77	51	0.10	-0.31	0.58

## &gt; # 群別の記述統計量の一括表示

## &gt; # byを使う方法

&gt; by(d1[,c("学習意欲","自己効力感")], d1\$入学年度, describe)

d1\$入学年度: 20Y1

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	87	24.08	5.53	23	24.15	4.45	8	37	29	-0.12	0.16	0.59
自己効力感	2	87	47.40	9.91	46	47.04	10.38	26	73	47	0.32	-0.58	1.06

d1\$入学年度: 20Y2

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	98	25.48	5.11	25.5	25.36	5.19	16	39	23	0.26	-0.46	0.52
自己効力感	2	98	50.14	9.65	49.5	49.98	11.12	26	77	51	0.16	0.04	0.97

d1\$入学年度: 20Y3

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	85	26.05	4.45	26	26.04	4.45	15	38	23	0.10	-0.17	0.48
自己効力感	2	85	51.22	8.52	52	51.35	8.90	29	72	43	-0.14	-0.34	0.92

## &gt; # describeByを使う方法

&gt; describeBy(d1[,c("学習意欲","自己効力感")], d1\$入学年度)

Descriptive statistics by group

group: 20Y1

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	87	24.08	5.53	23	24.15	4.45	8	37	29	-0.12	0.16	0.59
自己効力感	2	87	47.40	9.91	46	47.04	10.38	26	73	47	0.32	-0.58	1.06

group: 20Y2

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	98	25.48	5.11	25.5	25.36	5.19	16	39	23	0.26	-0.46	0.52
自己効力感	2	98	50.14	9.65	49.5	49.98	11.12	26	77	51	0.16	0.04	0.97

group: 20Y3

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲	1	85	26.05	4.45	26	26.04	4.45	15	38	23	0.10	-0.17	0.48
自己効力感	2	85	51.22	8.52	52	51.35	8.90	29	72	43	-0.14	-0.34	0.92

## &gt; # mat=TRUEを指定して次のように表記することも可能

&gt; describeBy(d1[,c("学習意欲","自己効力感")], d1\$入学年度, mat=TRUE, digits=2)

	item	group	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲1	1	20Y1	1	87	24.08	5.53	23.0	24.15	4.45	8	37	29	-0.12	0.16	0.59
学習意欲2	2	20Y2	1	98	25.48	5.11	25.5	25.36	5.19	16	39	23	0.26	-0.46	0.52
学習意欲3	3	20Y3	1	85	26.05	4.45	26.0	26.04	4.45	15	38	23	0.10	-0.17	0.48
自己効力感1	4	20Y1	2	87	47.40	9.91	46.0	47.04	10.38	26	73	47	0.32	-0.58	1.06
自己効力感2	5	20Y2	2	98	50.14	9.65	49.5	49.98	11.12	26	77	51	0.16	0.04	0.97
自己効力感3	6	20Y3	2	85	51.22	8.52	52.0	51.35	8.90	29	72	43	-0.14	-0.34	0.92

## 2 要因以上の群別要約統計量

## describeBy関数を使う方法

library(psych)

describeBy(変数名, list(群分け変数1, 群分け変数2, ...), mat=TRUE, digits=2)

予めpsychパッケージをインストールしておく必要がある。

mat=TRUE というオプションを指定することにより, 出力結果を行列表記にすることができる。

&gt; # list() を使って群分け変数を入れ子状に設定することも可能

&gt; describeBy(d1[,c("学習意欲")], list(d1\$入学年度, d1\$学科), mat=TRUE, digits=2)

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	20Y1	医学	1	29	25.07	6.54	25.0	25.52	5.93	8	35	27	-0.72	0.09	1.21
X12	2	20Y2	医学	1	27	26.81	5.95	26.0	26.61	5.93	17	39	22	0.04	-0.76	1.14
X13	3	20Y3	医学	1	20	27.50	3.87	28.0	27.44	4.45	22	35	13	0.06	-1.35	0.87
X14	4	20Y1	看護学	1	28	23.86	4.47	23.0	23.58	2.97	16	36	20	0.82	0.30	0.84
X15	5	20Y2	看護学	1	33	24.52	4.33	25.0	24.48	4.45	16	36	20	0.18	0.09	0.75
X16	6	20Y3	看護学	1	39	25.13	4.70	25.0	24.97	4.45	15	38	23	0.34	0.20	0.75
X17	7	20Y1	心理学	1	30	23.33	5.40	23.0	23.33	4.45	12	37	25	0.15	-0.13	0.99
X18	8	20Y2	心理学	1	38	25.37	5.01	24.5	25.34	5.19	17	34	17	0.20	-1.10	0.81
X19	9	20Y3	心理学	1	26	26.31	4.28	26.0	26.36	4.45	17	35	18	0.01	-0.49	0.84

> # 群分け変数2(学科)でまず群分けし, さらに, 群分け変数1(年度)で群分けして, 各群の記述統計量を  
 > # 計算

> # 記述統計量を計算する変数を1つ(学習意欲)しか設定していないので, 変数名は X1 とされている。  
 > # X11 はその第1群, X12はその第2群, という変数表示になっている。

&gt; # 記述統計量を計算する変数を2つ以上かけば, 変数名に数字がついて表示される。

&gt; # 変数が2つ以上ある場合

&gt;

&gt; describeBy(d1[,c("学習意欲", "自己効力感")], list(d1\$入学年度, d1\$学科), mat=TRUE, digits=2)

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
学習意欲1	1	20Y1	医学	1	29	25.07	6.54	25.0	25.52	5.93	8	35	27	-0.72	0.09	1.21
学習意欲2	2	20Y2	医学	1	27	26.81	5.95	26.0	26.61	5.93	17	39	22	0.04	-0.76	1.14
学習意欲3	3	20Y3	医学	1	20	27.50	3.87	28.0	27.44	4.45	22	35	13	0.06	-1.35	0.87
学習意欲4	4	20Y1	看護学	1	28	23.86	4.47	23.0	23.58	2.97	16	36	20	0.82	0.30	0.84
学習意欲5	5	20Y2	看護学	1	33	24.52	4.33	25.0	24.48	4.45	16	36	20	0.18	0.09	0.75
学習意欲6	6	20Y3	看護学	1	39	25.13	4.70	25.0	24.97	4.45	15	38	23	0.34	0.20	0.75
学習意欲7	7	20Y1	心理学	1	30	23.33	5.40	23.0	23.33	4.45	12	37	25	0.15	-0.13	0.99
学習意欲8	8	20Y2	心理学	1	38	25.37	5.01	24.5	25.34	5.19	17	34	17	0.20	-1.10	0.81
学習意欲9	9	20Y3	心理学	1	26	26.31	4.28	26.0	26.36	4.45	17	35	18	0.01	-0.49	0.84
自己効力感1	10	20Y1	医学	2	29	49.76	9.83	50.0	49.32	11.86	35	73	38	0.41	-0.83	1.83
自己効力感2	11	20Y2	医学	2	27	50.26	10.81	52.0	50.87	8.90	26	73	47	-0.50	-0.01	2.08
自己効力感3	12	20Y3	医学	2	20	56.00	7.67	54.5	55.75	8.15	41	72	31	0.20	-0.69	1.72
自己効力感4	13	20Y1	看護学	2	28	45.68	8.67	45.0	44.96	5.93	33	68	35	0.97	0.51	1.64
自己効力感5	14	20Y2	看護学	2	33	50.55	10.71	49.0	49.85	11.86	34	77	43	0.55	-0.43	1.86
自己効力感6	15	20Y3	看護学	2	39	50.38	8.11	52.0	50.48	8.90	35	65	30	-0.14	-1.08	1.30
自己効力感7	16	20Y1	心理学	2	30	46.73	10.91	47.5	46.92	14.08	26	66	40	-0.09	-1.22	1.99
自己効力感8	17	20Y2	心理学	2	38	49.71	7.90	48.5	49.41	9.64	38	70	32	0.37	-0.69	1.28
自己効力感9	18	20Y3	心理学	2	26	48.81	8.58	49.5	48.95	6.67	29	65	36	-0.28	-0.31	1.68

&gt;

## 共分散・相関係数

不偏共分散行列 (n-1で割る)  
 cov(データフレーム名)

## 相関係数行列

cor(データフレーム名, method="算出方法", use="欠測値の扱い方法")

```
method: "pearson" (default), "spearman", or "kendall"
use: "everything"      当該2変数に欠測値がある場合, その箇所の値だけNAとなる.
    "complete.obs"     1つでも欠測値のある行を除外してすべての値を計算.
    "pairwise.complete.obs" 当該2変数に欠測値がある場合, その箇所だけ欠測値を除外して計算.
```

```
> setwd("d:¥¥")
> d1 <- read.table("偏相関データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  grade kutsu math
1      3    21  49
2      2    20  36
3      3    20  47
4      3    20  54
5      6    26  73
6      1    17   2
```

```
>
> # 共分散
> cov(d1)
```

```
      grade      kutsu      math
grade  2.926421  4.573579  30.80100
kutsu  4.573579  9.073567  47.37457
math  30.801003  47.374571  400.18394
```

```
# grade の分散 = 2.926421
# grade と kutsu の共分散 = 4.573579
# kutsu と math の共分散 = 47.47457
```

```
> # ピアソンの積率相関係数
> cor(d1)
```

```
      grade      kutsu      math
grade  1.0000000  0.887562  0.9000499
kutsu  0.8875620  1.0000000  0.7861880
math   0.9000499  0.786188  1.0000000
```

```
# grade と kutsu の相関係数 = 0.887562
# kutsu と math の相関係数 = 0.9000499
```

```
> # スピアマンの順位相関係数
> cor(d1, method="spearman")
```

```
      grade      kutsu      math
grade  1.0000000  0.9004303  0.9094415
kutsu  0.9004303  1.0000000  0.8158021
math   0.9094415  0.8158021  1.0000000
```

```
> # ケンドールの順位相関係数
> cor(d1, method="kendall")
```

```
      grade      kutsu      math
grade  1.0000000  0.7881925  0.7766475
kutsu  0.7881925  1.0000000  0.6314266
math   0.7766475  0.6314266  1.0000000
```

	A	B	C
1	grade	kutsu	math
2	3	21	49
3	2	20	36
4	3	20	47
5	3	20	54
6	6	26	73
7	1	17	2
8	1	21	25
9	3	19	57
10	6	24	73
11	4	23	58
12	2	18	27
13	2	16	39
14	4	21	66
15	3	21	44
16	5	24	71
17	4	21	59
18	5	24	69
19	1	16	21
20	1	19	31
21	6	24	81

## 群別の共分散・相関係数

不偏共分散行列 (n-1で割る)  
**by**(データフレーム名, 群別変数名, **cov**)

相関係数行列  
**by**(データフレーム名, 群別変数名, **cor**, **use**="欠測値の扱い方法")

method: "pearson" (default), "spearman", or "kendall"  
 use: "everything" 当該2変数に欠測値がある場合, その箇所の値だけNAとなる.  
       "complete.obs" 1つでも欠測値のある行を除外してすべての値を計算.  
       "pairwise.complete.obs" 当該2変数に欠測値がある場合, その箇所だけ欠測値を除外して計算.

データフレームは, 群別変数も含めた必要な変数だけにしておく.

```
> setwd("d:¥¥")
> d1 <- read.table("偏相関データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  grade kutsu math
1      3    21  49
2      2    20  36
3      3    20  47
4      3    20  54
5      6    26  73
6      1    17   2
>
>
```

```
> # 各群の標本サイズ
> table(d1$grade)
```

```
 1  2  3  4  5  6
50 50 50 50 50 50
>
```

```
> # 各群の平均
> aggregate(d1[,c("kutsu", "math")], list(d1$grade), mean)
  Group.1 kutsu  math
1      1 17.22 23.64
2      2 18.54 32.58
3      3 20.26 46.54
4      4 21.64 55.50
5      5 23.38 66.72
6      6 24.98 75.04
>
```

```
> # 各群のSD
> sapply(d1[,c("kutsu", "math")], tapply, d1$grade, sd)
      kutsu      math
1 1.266362 11.035675
2 1.501156  6.931413
3 1.208980  8.447219
4 1.396205  8.576784
5 1.496799  7.946351
6 1.477588  8.787654
>
```

```
> # 各群の共分散行列
> by(d1[,c("kutsu", "math")], d1$grade, cov)
d1$grade: 1
      kutsu      math
kutsu 1.603673 -2.164082
math -2.164082 121.786122
```

	A	B	C
1	grade	kutsu	math
2	3	21	49
3	2	20	36
4	3	20	47
5	3	20	54
6	6	26	73
7	1	17	2
8	1	21	25
9	3	19	57
10	6	24	73
11	4	23	58
12	2	18	27
13	2	16	39
14	4	21	66
15	3	21	44
16	5	24	71
17	4	21	59
18	5	24	69
19	1	16	21
20	1	19	31
21	6	24	81



```

d1$grade: 2
      kutsu      math
kutsu  2.253469 -2.748163
math   -2.748163 48.044490
-----
d1$grade: 3
      kutsu      math
kutsu  1.4616327 -0.6534694
math   -0.6534694 71.3555102
-----
d1$grade: 4
      kutsu      math
kutsu  1.949388  2.591837
math   2.591837 73.561224
-----
d1$grade: 5
      kutsu      math
kutsu  2.2404082 -0.8914286
math   -0.8914286 63.1444898
-----
d1$grade: 6
      kutsu      math
kutsu  2.1832653 -0.9583673
math   -0.9583673 77.2228571
>

```

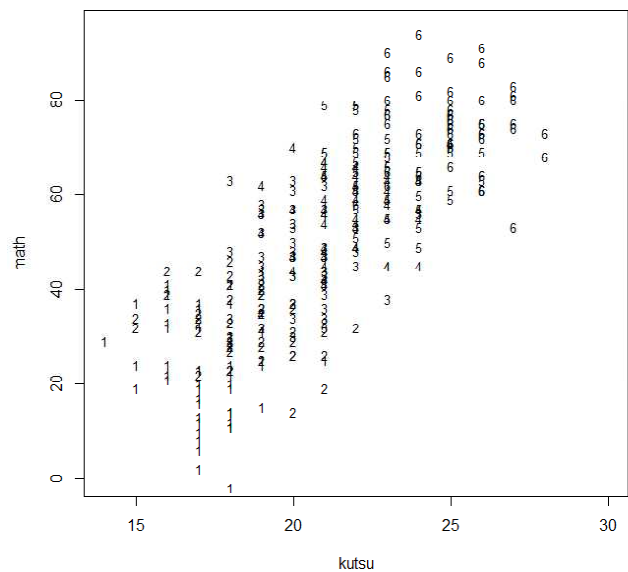
> # 各群の相関係数行列

```
> by(d1[,c("kutsu", "math")], d1$grade, cor)
```

```

d1$grade: 1
      kutsu      math
kutsu  1.000000 -0.154852
math   -0.154852  1.000000
-----
d1$grade: 2
      kutsu      math
kutsu  1.0000000 -0.2641161
math   -0.2641161  1.0000000
-----
d1$grade: 3
      kutsu      math
kutsu  1.00000000 -0.06398709
math   -0.06398709  1.00000000
-----
d1$grade: 4
      kutsu      math
kutsu  1.0000000 0.2164383
math   0.2164383  1.0000000
-----
d1$grade: 5
      kutsu      math
kutsu  1.00000000 -0.07494717
math   -0.07494717  1.00000000
-----
d1$grade: 6
      kutsu      math
kutsu  1.0000000 -0.0738084
math   -0.0738084  1.0000000
>
>
>

```



> # 重ね合わせ散布図

```

> plot(d1$kutsu, d1$math, pch="", axes=T, xlim=c(14, 30), ylim=c(0, 95),
+       xlab="kutsu", ylab="math")
> text(d1$kutsu, d1$math, labels=d1$grade, cex=0.8)

```

## 偏相関係数

## 影響を除く変数を指定した偏相関係数

```
library(psych)
partial.r(data, c(偏相関係数を求めたい変数群の列番号), c(影響を除きたい変数群の列番号))
```

あらかじめpsychパッケージをインストールしておく必要がある。  
 data はデータフレームでもよいし、相関係数行列でもよい。  
 偏相関係数を求めたい変数が2つ以上ある場合は、列番号をカンマで区切る。  
 影響を除きたい変数が2つ以上ある場合は、列番号をカンマで区切る。

## 当該の2変数以外の全ての変数の影響を除いた偏相関係数行列

```
library(corpcor)
cor2pcor(相関係数行列)
```

あらかじめcorpcorパッケージをインストールしておく必要がある。

```
> setwd("d:¥¥")
> d1 <- read.table("偏相関データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  kutsu math grade
1    21   49    3
2    20   36    2
3    20   47    3
4    20   54    3
5    26   73    6
6    17    2    1
>
>
```

	A	B	C
1	grade	kutsu	math
2	3	21	49
3	2	20	36
4	3	20	47
5	3	20	54
6	6	26	73
7	1	17	2
8	1	21	25
9	3	19	57
10	6	24	73
11	4	23	58
12	2	18	27
13	2	16	39
14	4	21	66
15	3	21	44
16	5	24	71
17	4	21	59
18	5	24	69
19	1	16	21
20	1	19	31
21	6	24	81

```
> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean   SD kutsu math grade
kutsu 300 21.0  3.01  1.00 0.79  0.89
math  300 50.0 20.00  0.79 1.00  0.90
grade 300  3.5  1.71  0.89 0.90  1.00
```

```
> # 偏相関係数
> library(psych)
> partial.r(d1, c(2,3), c(1))
partial correlations
      grade math
grade  1.00 0.71
math   0.71 1.00
```

```
> partial.r(d1, c(1,3), c(2))
partial correlations
      grade kutsu
grade  1.00  0.67
kutsu  0.67  1.00
```

```
> partial.r(d1, c(1,2), c(3))
partial correlations
      kutsu math
kutsu  1.00 -0.06
math  -0.06  1.00
>
```

```
> # 当該の2変数以外の全ての変数の影響を除いた偏相関係数  
>
```

```
> #corpcorパッケージのcor2pcor関数を使う方法
```

```
> library(corpcor)
```

```
> cor2pcor(ctmp)
```

```
      [,1]      [,2]      [,3]  
[1,]  1.0000000 -0.0630700  0.6682002  
[2,] -0.0630700  1.0000000  0.7104308  
[3,]  0.6682002  0.7104308  1.0000000
```

## 四分相関係数・多分相関係数

```
library(psych)
library(polycor)
オブジェクト名1 <- polychoric(データフレーム名)
```

```
相関係数行列だけの取り出し
オブジェクト名1$rho
```

あらかじめpsychとpolycorパッケージをインストールしておく必要がある。  
データフレーム以外に、テーブルを代入したりすることもできる。

```
> setwd("d:¥¥")
> d1 <- read.table("2値_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  x1 x2 x3 x4 x5 x6 x7 x8
1  0  0  1  1  1  0  0  1
2  0  0  1  1  0  0  1  0
3  0  0  0  0  0  0  0  0
4  0  1  0  0  1  0  1  1
5  0  1  1  1  1  1  0  1
6  0  0  0  1  0  0  1  0
>
```

```
> # 標本サイズ, 平均, 標準偏差
> dtmp <- d1
> ntmp <- nrow(d1)
> mtmp <- apply(d1, 2, mean)
> stmp <- apply(d1, 2, sd)
> ktmp <- round(data.frame(ntmp, mtmp, stmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD")
> ktmp
```

	N	Mean	SD
x1	346	0.48	0.5
x2	346	0.54	0.5
x3	346	0.51	0.5
x4	346	0.53	0.5
x5	346	0.49	0.5
x6	346	0.53	0.5
x7	346	0.53	0.5
x8	346	0.51	0.5

	A	B	C	D	E	F	G	H
1	x1	x2	x3	x4	x5	x6	x7	x8
2	0	0	1	1	1	0	0	1
3	0	0	1	1	0	0	1	0
4	0	0	0	0	0	0	0	0
5	0	1	0	0	1	0	1	1
6	0	1	1	1	1	1	0	1
7	0	0	0	1	0	0	1	0
8	0	1	0	0	0	1	0	0
9	1	1	1	1	0	1	0	0
10	1	0	1	1	1	1	0	0
11	1	1	1	0	0	0	1	1
12	1	1	0	0	0	0	1	1
13	1	1	0	1	0	1	0	1
14	0	1	0	0	1	0	0	0
15	0	1	0	1	1	1	1	1
16	0	0	1	0	0	1	1	1
17	1	1	0	1	1	1	0	0
18	1	1	1	1	1	0	1	1
19	1	1	1	0	1	1	0	1
20	1	1	1	0	0	0	0	1
21	1	1	0	0	1	1	1	1

```
> #積率相関係数
> round(cor(dtmp), 2)
```

	x1	x2	x3	x4	x5	x6	x7	x8
x1	1.00	0.18	0.25	0.16	0.03	0.05	0.05	0.01
x2	0.18	1.00	0.19	0.05	0.07	0.07	0.01	0.10
x3	0.25	0.19	1.00	0.14	0.09	0.04	0.03	0.07
x4	0.16	0.05	0.14	1.00	0.07	0.01	-0.02	0.00
x5	0.03	0.07	0.09	0.07	1.00	0.13	0.17	0.24
x6	0.05	0.07	0.04	0.01	0.13	1.00	0.10	0.03
x7	0.05	0.01	0.03	-0.02	0.17	0.10	1.00	0.12
x8	0.01	0.10	0.07	0.00	0.24	0.03	0.12	1.00

```
> #四分相関係数・多分相関係数
> library(psych)
> library(polycor)
```

```
> (ptcor.d1 <- polychoric(d1))
Call: polychoric(x = d1)
Polychoric correlations
```

	x1	x2	x3	x4	x5	x6	x7	x8
x1	1.00							
x2	0.28	1.00						
x3	0.38	0.29	1.00					
x4	0.25	0.07	0.22	1.00				
x5	0.05	0.11	0.15	0.11	1.00			
x6	0.08	0.10	0.07	0.01	0.21	1.00		
x7	0.09	0.02	0.04	-0.03	0.27	0.16	1.00	
x8	0.02	0.16	0.11	0.00	0.36	0.04	0.18	1.00

```
with tau of
0
```

```
x1 0.051
x2 -0.094
x3 -0.036
x4 -0.080
x5 0.014
x6 -0.073
x7 -0.065
x8 -0.014
>
```

```
> # 相関係数行列だけの取り出し
```

```
> pcor.d1 <- ptcor.d1$rho
```

```
> round(pcor.d1, 2)
```

	x1	x2	x3	x4	x5	x6	x7	x8
x1	1.00	0.28	0.38	0.25	0.05	0.08	0.09	0.02
x2	0.28	1.00	0.29	0.07	0.11	0.10	0.02	0.16
x3	0.38	0.29	1.00	0.22	0.15	0.07	0.04	0.11
x4	0.25	0.07	0.22	1.00	0.11	0.01	-0.03	0.00
x5	0.05	0.11	0.15	0.11	1.00	0.21	0.27	0.36
x6	0.08	0.10	0.07	0.01	0.21	1.00	0.16	0.04
x7	0.09	0.02	0.04	-0.03	0.27	0.16	1.00	0.18
x8	0.02	0.16	0.11	0.00	0.36	0.04	0.18	1.00

```
>
```

```
>
```

## アルファ係数

変数リスト名 <- c("変数名1", "変数名2", ..., "変数名p")

library(psych)

alpha(データフレーム名[, 変数リスト名])

あらかじめpsychパッケージをインストールしておく必要がある。

## 出力内容

total	a list containing
raw_alpha	alpha based upon the covariances
std.alpha	The standardized alpha based upon the correlations
G6(smc)	Guttman's Lambda 6 reliability
average_r	The average interitem correlation
median_r	The median interitem correlation
mean	For data matrices, the mean of the scale formed by averaging or summing the items (depending upon the cumulative option)
sd	For data matrices, the standard deviation of the total score
alpha.drop	A data frame with all of the above for the case of each item being removed one by one.
item.stats	A data frame including
n	number of complete cases for the item
raw.r	The correlation of each item with the total score, not corrected for item overlap.
std.r	The correlation of each item with the total score (not corrected for item overlap) if the items were all standardized
r.cor	Item whole correlation corrected for item overlap and scale reliability
r.drop	Item whole correlation for this item against the scale without this item
mean	for data matrices, the mean of each item
sd	For data matrices, the standard deviation of each item
response.freq	For data matrices, the frequency of each item response (if less than 20)
scores	Scores are by default simply the average response for all items that a participant took. If cumulative=TRUE, then these are sum scores. Note, this is dangerous if there are lots of missing values.
boot.ci	The lower, median, and upper ranges of the 95% confidence interval based upon the bootstrap.
boot	a 6 column by n.iter matrix of boot strapped resampled values
Unidim	An index of unidimensionality
Fit	The fit of the off diagonal matrix

coefficientパッケージの中にも $\alpha$ 係数を求めるalphaという関数があるが、項目分析をやってくれない。

```
> setwd("d:¥¥")
```

```
> d1 <- read.table("統計分析力尺度データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> head(d1)
```

	番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	y1	y2	y3	y4	y5	y6	y7	y8	y9	y10	xt	yt
1	1	3	2	1	2	2	4	4	3	4	4	3	2	1	2	1	4	3	3	3	4	29	26
2	2	3	3	4	3	2	2	2	2	4	1	2	2	4	2	2	1	1	1	3	1	26	19
3	3	4	3	3	4	1	3	4	2	5	1	4	3	4	4	1	3	4	2	5	1	30	31
4	4	5	5	5	3	3	4	4	2	4	4	5	4	5	2	2	3	4	2	3	3	39	33
5	5	3	3	4	2	2	3	3	3	4	1	3	4	4	3	2	4	3	4	5	1	28	33
6	6	2	1	4	1	1	3	3	2	5	1	3	2	4	2	2	5	4	3	5	2	23	32

	統計	数学	批判的思考力	国語	自己効力感	
1	51	48		28	72	61
2	74	53		26	66	53
3	48	60		35	71	48
4	67	68		27	67	48
5	55	49		30	66	49
6	74	63		36	83	37

```
> #変数リスト名
```

```
> inames <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")
```

```
> # α 係数の推定
> library(psych)
> alpha(d1[, inames])
```

Reliability analysis

Call: alpha(x = d1[, inames], check.keys = FALSE)

```
raw_alpha std.alpha G6(smc) average_r S/N ase mean sd
0.84      0.84      0.84      0.35 5.3 0.02 3.1 0.61
```

```
lower alpha upper      95% confidence boundaries
0.8 0.84 0.88
```

Reliability if an item is dropped:

```
raw_alpha std.alpha G6(smc) average_r S/N alpha se
x1      0.82      0.82      0.82      0.34 4.7 0.022
x2      0.82      0.82      0.82      0.34 4.6 0.022
x3      0.83      0.83      0.83      0.34 4.7 0.022
x4      0.82      0.82      0.82      0.34 4.6 0.022
x5      0.82      0.82      0.82      0.34 4.6 0.022
x6      0.82      0.82      0.82      0.34 4.7 0.022
x7      0.83      0.83      0.83      0.35 4.8 0.022
x8      0.83      0.83      0.83      0.35 4.8 0.022
x9      0.83      0.83      0.83      0.35 4.9 0.022
x10     0.83      0.83      0.83      0.36 5.0 0.022
```

Item statistics

```
      n      r r.cor r.drop mean sd
x1 365 0.67 0.62 0.57 4.0 0.92
x2 365 0.67 0.63 0.57 3.1 0.96
x3 365 0.65 0.60 0.54 4.1 0.85
x4 365 0.68 0.64 0.57 3.0 1.07
x5 365 0.67 0.63 0.57 2.2 0.93
x6 365 0.65 0.61 0.54 3.0 1.01
x7 365 0.62 0.56 0.51 3.1 0.99
x8 365 0.63 0.58 0.53 2.2 0.89
x9 365 0.59 0.52 0.48 3.9 0.90
x10 365 0.57 0.51 0.46 2.1 0.95
```

Non missing response frequency for each item

```
      1      2      3      4      5 miss
x1 0.01 0.05 0.23 0.36 0.35 0
x2 0.04 0.22 0.42 0.25 0.07 0
x3 0.01 0.03 0.22 0.40 0.35 0
x4 0.09 0.23 0.36 0.24 0.08 0
x5 0.25 0.41 0.25 0.09 0.01 0
x6 0.06 0.24 0.38 0.25 0.07 0
x7 0.05 0.22 0.36 0.30 0.07 0
x8 0.25 0.42 0.25 0.07 0.00 0
x9 0.01 0.06 0.24 0.40 0.29 0
x10 0.30 0.37 0.25 0.08 0.01 0
```

>

	A	B	C	D	E	F	G	H	I	J	K
1	番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
2	1	3	2	1	2	2	4	4	3	4	4
3	2	3	3	4	3	2	2	2	2	4	1
4	3	4	3	3	4	1	3	4	2	5	1
5	4	5	5	5	3	3	4	4	2	4	4
6	5	3	3	4	2	2	3	3	3	4	1
7	6	2	1	4	1	1	3	3	2	5	1
8	7	5	3	5	5	4	5	5	3	5	3
9	8	4	3	4	3	2	3	2	2	4	1
10	9	4	2	4	2	2	2	2	2	4	2
11	10	4	5	5	4	4	3	3	3	4	3
12	11	4	2	4	4	2	4	4	2	4	3
13	12	5	3	4	5	2	3	3	2	3	2
14	13	3	3	5	3	3	2	3	2	3	2
15	14	4	4	5	3	3	3	4	2	3	1
16	15	4	2	4	3	1	4	2	1	3	3
17	16	4	3	4	1	1	2	1	1	3	1
18	17	3	2	4	3	2	2	2	2	3	2
19	18	3	3	5	2	2	3	2	1	4	1
20	19	3	3	5	2	2	4	4	2	4	1
21	20	3	2	5	1	2	3	3	1	4	2

## アルファ係数の警告メッセージについて

アルファ係数を求めるにあたって、次のような警告メッセージが出ることがあります。

Some items (...) were negatively correlated with the total scale and probably should be reversed.  
To do this, run the function again with the 'check.keys=TRUE' option Reliability analysis

これは、いわゆる逆転項目が混入している疑いがあるというメッセージです。他の多くの項目と負の相関がある項目があるということです。

対処法としては、実際に逆転項目であれば、得点の変換（逆転処理）を行ってアルファ係数を求めるようにします。逆転項目でなければ、いま出ている結果を受け入れることになります。

逆転処理後のアルファ係数を求めるオプション設定として check.keys=TRUE があります。これは、逆転項目の疑いがある変数について、自動で得点変換を行ってアルファ係数を算出するものですが、元のデータを変換する訳ではないので、このオプションを使うのは、お勧めできません。

## オメガ係数

変数リスト名 <- c("変数名1", "変数名2", ..., "変数名p")

library(coefficientsalpha)

オブジェクト名 <- omega(データフレーム名[, 変数リスト名], se=TRUE)

summary.omega(オブジェクト名)

あらかじめcoefficientsalphaパッケージをインストールしておく必要がある。  
se=TRUEをつけておかないと、summary.omegaで信頼区間を計算してくれない。  
逆転項目は、あらかじめスコアを正しい方向に修正しておく必要がある。

psychパッケージにもomegaという関数があるが、因子分析を使うものである。

```
> setwd("d:¥¥¥")
> d1 <- read.table("統計分析力尺度データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> head(d1)
```

```
  番号 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 xt yt
1    1  1  3  2  1  2  2  4  4  3  4  4  3  2  1  2  1  4  3  3  3  4 29 26
2    2  2  3  3  4  3  2  2  2  2  4  1  2  2  4  2  2  1  1  1  3  1 26 19
3    3  3  4  3  3  4  1  3  4  2  5  1  4  3  4  4  1  3  4  2  5  1 30 31
4    4  4  5  5  5  3  3  4  4  2  4  4  5  4  5  2  2  3  4  2  3  3 39 33
5    5  5  3  3  4  2  2  3  3  3  4  1  3  4  4  3  2  4  3  4  5  1 28 33
6    6  6  2  1  4  1  1  3  3  2  5  1  3  2  4  2  2  5  4  3  5  2 23 32
```

```
  統計 数学 批判的思考力 国語 自己効力感
1    51   48           28   72         61
2    74   53           26   66         53
3    48   60           35   71         48
4    67   68           27   67         48
5    55   49           30   66         49
6    74   63           36   83         37
>
```

```
> #変数リスト名
```

```
> inames <- c("x1", "x2", "x3", "x4", "x5",
+            "x6", "x7", "x8", "x9", "x10")
```

```
> # omega 係数の推定
```

```
> library(coefficientsalpha)
```

```
> o1 <- omega(d1[, inames], se=T)
```

```
Test of homogeneity
```

```
The robust F statistic is 3.288
```

```
with a p-value 0
```

```
**The F test rejected homogeneity**
```

```
The omega is 0.8409224 with the standard error 0.01255086.
```

```
About 6.85% of cases were downweighted.
```

```
> # 信頼区間の計算
```

```
> summary.omega(o1)
```

```
The estimated omega is
```

```
omega          0.841
se             0.013
p-value (omega>0) 0.000
Confidence interval 0.816      0.866
```

```
Test of homogeneity
```

```
The robust F statistic is 3.288
```

```
with a p-value 0
```

```
Note. The robust test rejected the assumption of homogeneity.
```

	A	B	C	D	E	F	G	H	I	J	K
1	番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
2	1	3	2	1	2	2	4	4	3	4	4
3	2	3	3	4	3	2	2	2	2	4	1
4	3	4	3	3	4	1	3	4	2	5	1
5	4	5	5	5	3	3	4	4	2	4	4
6	5	3	3	4	2	2	3	3	3	4	1
7	6	2	1	4	1	1	3	3	2	5	1
8	7	5	3	5	5	4	5	5	3	5	3
9	8	4	3	4	3	2	3	2	2	4	1
10	9	4	2	4	2	2	2	2	2	4	2
11	10	4	5	5	4	4	3	3	3	4	3
12	11	4	2	4	4	2	4	4	2	4	3
13	12	5	3	4	5	2	3	3	2	3	2
14	13	3	3	5	3	3	2	3	2	3	2
15	14	4	4	5	3	3	3	4	2	3	1
16	15	4	2	4	3	1	4	2	1	3	3
17	16	4	3	4	1	1	2	1	1	3	1
18	17	3	2	4	3	2	2	2	2	3	2
19	18	3	3	5	2	2	3	2	1	4	1
20	19	3	3	5	2	2	4	4	2	4	1
21	20	3	2	5	1	2	3	3	1	4	2



## 級内相関係数

```
library(psych)
ICC(データフレーム名)
```

あらかじめpsychパッケージをインストールしておく必要がある。

ICC1: 被験者内要因 (k回評価) の効果を誤差に含めるモデルにおける, 評価の一致度  
 ICC2: 被験者内要因 (k回評価) の効果を変量効果とするモデルにおける, 評価の一致度  
 ICC3: 被験者内要因 (k回評価) の効果を固定効果とするモデルにおける, 評価の一致度  
 ICC1k: 被験者内要因 (k回評価) の効果を誤差に含めるモデルにおける, k回評価の平均値の信頼性  
 ICC2k: 被験者内要因 (k回評価) の効果を変量効果とするモデルにおける, k回評価の平均値の信頼性  
 ICC3k: 被験者内要因 (k回評価) の効果を固定効果とするモデルにおける, k回評価の平均値の信頼性

```
> setwd("d:¥¥")
> d1 <- read.table("統計分析力尺度データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 xt yt
1    1  1  3  2  1  2  2  4  4  3  4  4  3  2  1  2  1  4  3  3  3  4 29 26
2    2  2  3  3  4  3  2  2  2  2  4  1  2  2  4  2  2  1  1  1  3  1 26 19
3    3  3  4  3  3  4  1  3  4  2  5  1  4  3  4  4  1  3  4  2  5  1 30 31
4    4  4  5  5  5  3  3  4  4  2  4  4  5  4  5  2  2  3  4  2  3  3 39 33
5    5  5  3  3  4  2  2  3  3  3  4  1  3  4  4  3  2  4  3  4  5  1 28 33
6    6  6  2  1  4  1  1  3  3  2  5  1  3  2  4  2  2  5  4  3  5  2 23 32
  統計 数学 批判的思考力 国語 自己効力感
1    51   48           28   72         61
2    74   53           26   66         53
3    48   60           35   71         48
4    67   68           27   67         48
5    55   49           30   66         49
6    74   63           36   83         37
>
```

```
> #変数リスト名
> inames <- c("x1", "x2", "x3", "x4", "x5",
+            "x6", "x7", "x8", "x9", "x10")
>
>
```

```
> # 級内相関係数の推定
> library(psych)
> ICC(d1[, inames])
Call: ICC(x = d1[, inames])
```

```
Intraclass correlation coefficients
```

	type	ICC	F	df1	df2	p	lower bound	upper bound
Single_raters_absolute	ICC1	0.18	3.2	364	3285	0	0.15	0.22
Single_random_raters	ICC2	0.21	6.3	364	3276	0	0.13	0.29
Single_fixed_raters	ICC3	0.34	6.3	364	3276	0	0.31	0.39
Average_raters_absolute	ICC1k	0.69	3.2	364	3285	0	0.64	0.73
Average_random_raters	ICC2k	0.73	6.3	364	3276	0	0.61	0.81
Average_fixed_raters	ICC3k	<u>0.84</u>	6.3	364	3276	0	0.81	0.86

Number of subjects = 365      Number of Judges = 10>

```
> # ICC3Kはα係数に等しい
> library(psych)
> alpha(d1[, inames])
```

```
Reliability analysis
Call: alpha(x = d1[, inames], check.keys = FALSE)
```

	raw_alpha	std.alpha	G6(smc)	average_r	S/N	ase	mean	sd
	<u>0.84</u>	0.84	0.84	0.35	5.3	0.02	3.1	0.61

lower alpha upper      95% confidence boundaries  
 0.8 0.84 0.88

## 7 1群・2群の平均値に関する推測 — t検定 —

## 1 群の平均値の推測

t.test(平均値を比較する変数, mu=母平均値)

母平均値 mu=0 の場合は「mu=0」を省略することもできる

```
> setwd("d:¥¥¥")
> d1 <- read.table("1群の平均値データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  id score
1  1  116
2  2  105
3  3  102
4  4  104
5  5   93
6  6  111
>
>
```

	A	B
1	id	score
2	1	116
3	2	105
4	3	102
5	4	104
6	5	93
7	6	111
8	7	117
9	8	98
10	9	109
11	10	101
12	11	102
13	12	113
14	13	76
15	14	126
16	15	93
17	16	109
18	17	78
19	18	93
20	19	125
21	20	108

```
> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
   N   Mean   SD   id score
id  50 25.50 14.58 1.00  0.25
score 50 108.72 15.07 0.25  1.00
>
>
```

```
> # 「H0:  $\mu_0=100$ 」の検定
> t.test(d1$score, mu=100)
```

One Sample t-test

```
data: d1$score
t = 4.0916, df = 49, p-value = 0.0001593
alternative hypothesis: true mean is not equal to 100
95 percent confidence interval:
 104.4372 113.0028
sample estimates:
mean of x
 108.72
>
>
```

```
> # 「H0:  $\mu_0=95$ 」の検定
> t.test(d1$score, mu=95)
```

One Sample t-test

```
data: d1$score
t = 6.4376, df = 49, p-value = 4.933e-08
alternative hypothesis: true mean is not equal to 95
95 percent confidence interval:
 104.4372 113.0028
sample estimates:
mean of x
 108.72
```

## 対応のある2群の平均値の比較

t.test(平均値を比較する変数1, 平均値を比較する変数2, paired=TRUE)

paired=TRUE で対応のある検定であることを指定. これは省略できない.

```
> setwd("d:¥¥")
> d1 <- read.table("対応のある2群の平均値データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  教養科目 専門科目
1       17       18
2       16       14
3       16       18
4       16       15
5       18       18
6       21       20
>
>
> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD 教養科目 専門科目
教養科目 237 16.72 2.12      1.00      0.43
専門科目 237 17.01 2.40      0.43      1.00
>
>
> # 対応のある平均値の差の検定 (対応のある t 検定) を行う
> t.test(d1$教養科目, d1$専門科目, paired=TRUE)
```

	A	B
1	教養科目	専門科目
2	17	18
3	16	14
4	16	18
5	16	15
6	18	18
7	21	20
8	17	16
9	16	17
10	19	17
11	18	20
12	17	16
13	14	16
14	20	17
15	15	19
16	17	18
17	15	15
18	16	18
19	18	19
20	17	20
21	13	19

Paired t-test

```
data: d1$教養科目 and d1$専門科目
t = -1.8515, df = 236, p-value = 0.06534
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.6009154  0.0186369
sample estimates:
mean of the differences
 -0.2911392
```

>

## 対応のある2群の平均値差の効果量の推定

effectsizeパッケージのcohens\_dまたはhedges\_g関数を使う方法

library(effectsize)

cohens\_d(変数1, 変数2, paired=TRUE)

cohens\_d(Pair(変数1, 変数2)~1, data=データフレーム名)

hedges\_g(変数1, 変数2, paired=TRUE)

hedges\_g(Pair(変数1, 変数2)~1, data=データフレーム名)

あらかじめeffectsizeパッケージをインストールしておく必要がある。  
cohens\_d はコーエンのd, hedges\_g はヘッジズのgを求める。

&gt; setwd("d:¥¥")

&gt; d1 &lt;- read.table("対応のある2群の平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")

&gt; head(d1)

	教養科目	専門科目
1	17	18
2	16	14
3	16	18
4	16	15
5	18	18
6	21	20

&gt; # 各群の記述統計量

&gt; library(psych)

&gt; (stats &lt;- describe(d1[,c("教養科目", "専門科目")]))

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
教養科目	1	237	16.72	2.12	17	16.73	2.97	12	21	9	-0.09	-0.57	0.14
専門科目	2	237	17.01	2.40	17	17.01	2.97	10	24	14	0.00	-0.14	0.16

&gt; # 効果量の推定

&gt; # effectsizeパッケージのcohens\_dまたはhedges\_g関数を使う方法

&gt; library(effectsize)

&gt; cohens\_d(d1\$教養科目, d1\$専門科目, paired=TRUE)

Cohen's d | 95% CI

-----

-0.12 | [-0.25, 0.01]

&gt; cohens\_d(Pair(d1[, "教養科目"], d1[, "専門科目"])~1, data=d1)

Cohen's d | 95% CI

-----

-0.12 | [-0.25, 0.01]

&gt; hedges\_g(d1\$教養科目, d1\$専門科目, paired=TRUE)

Hedges' g | 95% CI

-----

-0.12 | [-0.25, 0.01]

&gt; hedges\_g(Pair(d1[, "教養科目"], d1[, "専門科目"])~1, data=d1)

Hedges' g | 95% CI

-----

-0.12 | [-0.25, 0.01]

## 対応のない2群の平均値の比較 — 素データ

t.test(平均値を比較する変数 ~ 群分け変数, データフレーム名) または  
t.test(平均値を比較する変数 1, 平均値を比較する変数 2, paired=FALSE)

paired=FALSE で対応のない検定であることを指定. これは省略することも可能.

```
> setwd("d:¥¥¥")
> d1 <- read.table("対応のない2群の平均値データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  番号 説明 不安
1    1 ビデオ 32
2    2 ビデオ 23
3    3 ビデオ 28
4    4 ビデオ 29
5    5 ビデオ 20
6    6 ビデオ 33
>
>
```

```
> # 記述統計量
> library(psych)
> describeBy(d1[, c("不安")], d1$説明)
group: ビデオ
  vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
1     1 103 27.04 4.32    27   27.12 4.45  17 37   20 -0.1    -0.44 0.43
-----
group: 看護師
  vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
1     1  94 25.44 6.22    25   25.43 5.93   9 40   31 -0.03   -0.15 0.64
>
>
```

```
> # 対応のない2群の平均値の検定
> # 等分散性を仮定しない場合
> t.test(不安 ~ 説明, d1)
```

Welch Two Sample t-test

```
data: 不安 by 説明
t = 2.082, df = 163.823, p-value = 0.0389
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.08272468 3.12260480
sample estimates:
mean in group ビデオ mean in group 看護師
      27.03883         25.43617
```

>

```
> # 等分散性を仮定した場合
> t.test(不安 ~ 説明, d1, var.equal=TRUE)
```

Two Sample t-test

```
data: 不安 by 説明
t = 2.1158, df = 195, p-value = 0.03563
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1087512 3.0965783
sample estimates:
mean in group ビデオ mean in group 看護師
      27.03883         25.43617
```

	A	B	C
1	番号	説明	不安
2	1	ビデオ	26
3	2	看護師	24
4	3	ビデオ	29
5	4	看護師	23
6	5	ビデオ	25
7	6	看護師	23
8	7	ビデオ	37
9	8	看護師	26
10	9	看護師	20
11	10	ビデオ	36
12	11	ビデオ	28
13	12	ビデオ	27
14	13	ビデオ	27
15	14	ビデオ	33
16	15	看護師	38
17	16	ビデオ	30
18	17	看護師	23
19	18	看護師	28
20	19	ビデオ	28
21	20	ビデオ	23

## 対応のない2群の平均値の比較 — 要約統計量

各群の人数, 平均, 標準偏差 (n1, m1, s1, n2, m2, s2) を求め, 以下の式に代入

```
dm <- m1 - m2
df <- n1 + n2 - 2
se <- sqrt(((n1-1)*s1*s1+(n2-1)*s2*s2)/(n1 + n2-2)*(1/n1 + 1/n2))
tValue <- dm/se
pValue <- 2 *(1 - pt(abs(tValue), df))
t95 <- qt(0.975, df)
L95 <- dm - t95 * se
U95 <- dm + t95 * se
data.frame(m1, m2, dm, df, tValue, pValue, L95, U95)
```

先行研究等で, 要約統計量しか分からないデータであっても, この方法を使えば, 平均値の比較ができる.  
先行研究を対照群に設定するときなどに, この計算方法を用いると良い.

```
> setwd("d:¥¥¥")
> d1 <- read.table("対応のない2群の平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
番号 説明 不安
1 1 ビデオ 32
2 2 ビデオ 23
3 3 ビデオ 28
4 4 ビデオ 29
5 5 ビデオ 20
6 6 ビデオ 33
```

	A	B	C
1	番号	説明	不安
2	1	ビデオ	26
3	2	看護師	24
4	3	ビデオ	29
5	4	看護師	23
6	5	ビデオ	25
7	6	看護師	23
8	7	ビデオ	37
9	8	看護師	26
10	9	看護師	20
11	10	ビデオ	36
12	11	ビデオ	28
13	12	ビデオ	27
14	13	ビデオ	27
15	14	ビデオ	33
16	15	看護師	38
17	16	ビデオ	30
18	17	看護師	23
19	18	看護師	28
20	19	ビデオ	28
21	20	ビデオ	23

```
> # 各群のデータを抽出
```

```
> data1 <- d1[d1$説明=="ビデオ", "不安"]
> data2 <- d1[d1$説明=="看護師", "不安"]
>
>
```

```
> # 記述統計量
```

```
> (n1 <- length(data1))
[1] 103
> (m1 <- mean(data1))
[1] 27.03883
> (s1 <- sd(data1))
[1] 4.315777
> (n2 <- length(data2))
[1] 94
> (m2 <- mean(data2))
[1] 25.43617
> (s2 <- sd(data2))
[1] 6.220947
>
>
```

```
> # 平均値の差の検定, 信頼区間
```

```
dm <- m1 - m2
df <- n1 + n2 - 2
se <- sqrt(((n1-1)*s1*s1+(n2-1)*s2*s2)/(n1 + n2-2)*(1/n1 + 1/n2))
tValue <- dm/se
pValue <- 2 *(1 - pt(abs(tValue), df))
t95 <- qt(0.975, df)
L95 <- dm - t95 * se
U95 <- dm + t95 * se
data.frame(m1, m2, dm, df, tValue, pValue, L95, U95)
```

```
      m1      m2      dm  df  tValue  pValue  L95  U95
1 27.03883 25.43617 1.602665 195 2.115773 0.03563318 0.1087512 3.096578
>
```

## 対応のない2群の平均値差の効果量の推定

compute.esパッケージのmesまたはmes2関数を使う方法

```
library(compute.es)
mes(m.1=m1, m.2=m2, sd.1=s1, sd.2=s2, n.1=n1, n.2=n2)
```

```
s.pool <- sqrt((n1*s1^2+n2*s2^2)/(n1+n2-2))
mes2(m.1=m1, m.2=m2, s.pooled=s.pool, n.1=n1, n.2=n2)
```

あらかじめcompute.esパッケージをインストールしておく必要がある。

mesは2群のSDの値が分かっているとき、mes2はプールされたSDしか分からないときに使う。

mesもmes2も、プールされたSDを分母に使う。

effectsizeパッケージのcohens\_dまたはhedges\_g関数を使う方法

```
library(effectsize)
cohens_d(変数~群分け変数, data=データフレーム名)
```

```
cohens_d(変数~群分け変数, data=データフレーム名, pooled_sd=FALSE)
```

```
hedges_g(変数~群分け変数, data=データフレーム名)
```

```
hedges_g(変数~群分け変数, data=データフレーム名, pooled_sd=FALSE)
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

cohens\_d はコーエンのd, hedges\_g はヘッジズのgを求める。

pooled\_sd=FALSE とすると、プールされたSDではなく、各群のSDの平均値をSDの値として用いる。

```
> setwd("d:¥¥")
> d1 <- read.table("対応のない2群の平均値データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  番号  説明  不安
1    1 ビデオ  26
2    2 看護師  24
3    3 ビデオ  29
4    4 看護師  23
5    5 ビデオ  25
6    6 看護師  23
>
>
> # 各群の記述統計量
> library(psych)
> (stats <- describeBy(d1[,c("不安")], list(d1$説明), mat=TRUE, digits=2))
  item group1 vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
X11    1 ビデオ    1 103 27.04 4.32    27   27.12 4.45   17 37    20 -0.10   -0.44 0.43
X12    2 看護師    1  94 25.44 6.22    25   25.43 5.93    9 40    31 -0.03   -0.15 0.64
>
>
> # 記述統計量の保存
> (m1 <- stats$mean[1])
[1] 27.04
> (s1 <- stats$sd[1])
[1] 4.32
> (n1 <- stats$n[1])
[1] 103
> (m2 <- stats$mean[2])
[1] 25.44
> (s2 <- stats$sd[2])
[1] 6.22
> (n2 <- stats$n[2])
[1] 94
>
```



```
> # 対応のない2群の平均値の検定
> # 等分散性を仮定した場合
> (res1 <- t.test(data1, data2, paired=FALSE, var.equal=TRUE))
```

Two Sample t-test

```
data: data1 and data2
t = 2.1158, df = 195, p-value = 0.03563
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1087512 3.0965783
sample estimates:
mean of x mean of y
 27.03883  25.43617
```

```
> # 効果量の推定
> # compute.esパッケージのmesまたはmes2関数を使う方法
> library(compute.es)
> mes(m.1=m1, m.2=m2, sd.1=s1, sd.2=s2, n.1=n1, n.2=n2)
Mean Differences ES:
```

```
d [ 95 %CI] = 0.3 [ 0.02 , 0.58 ]
var(d) = 0.02
p-value(d) = 0.04
U3(d) = 61.86 %
CLES(d) = 58.45 %
Cliff's Delta = 0.17
```

```
g [ 95 %CI] = 0.3 [ 0.02 , 0.58 ]
var(g) = 0.02
p-value(g) = 0.04
U3(g) = 61.82 %
CLES(g) = 58.42 %
```

Correlation ES:

```
r [ 95 %CI] = 0.15 [ 0.01 , 0.28 ]
var(r) = 0
p-value(r) = 0.04
```

```
z [ 95 %CI] = 0.15 [ 0.01 , 0.29 ]
var(z) = 0.01
p-value(z) = 0.04
```

Odds Ratio ES:

```
OR [ 95 %CI] = 1.73 [ 1.03 , 2.89 ]
p-value(OR) = 0.04
```

```
Log OR [ 95 %CI] = 0.55 [ 0.03 , 1.06 ]
var(log OR) = 0.07
p-value(Log OR) = 0.04
```

Other:

```
NNT = 10.56
Total N = 197
```

```
> # プールされたSDを使う方法
> s.pool <- sqrt((n1*s1^2+n2*s2^2)/(n1+n2-2))
> mes2(m.1=m1, m.2=m2, s.pooled=s.pool, n.1=n1, n.2=n2)
Mean Differences ES:
```

```
d [ 95 %CI] = 0.3 [ 0.02 , 0.58 ]
var(d) = 0.02
p-value(d) = 0.04
U3(d) = 61.8 %
CLES(d) = 58.41 %
Cliff's Delta = 0.17
```

	A	B	C
1	番号	説明	不安
2	1	ビデオ	26
3	2	看護師	24
4	3	ビデオ	29
5	4	看護師	23
6	5	ビデオ	25
7	6	看護師	23
8	7	ビデオ	37
9	8	看護師	26
10	9	看護師	20
11	10	ビデオ	36
12	11	ビデオ	28
13	12	ビデオ	27
14	13	ビデオ	27
15	14	ビデオ	33
16	15	看護師	38
17	16	ビデオ	30
18	17	看護師	23
19	18	看護師	28
20	19	ビデオ	28
21	20	ビデオ	23

```

g [ 95 %CI] = 0.3 [ 0.02 , 0.58 ]
var(g) = 0.02
p-value(g) = 0.04
U3(g) = 61.76 %
CLES(g) = 58.37 %

```

Correlation ES:

```

r [ 95 %CI] = 0.15 [ 0.01 , 0.28 ]
var(r) = 0
p-value(r) = 0.04

z [ 95 %CI] = 0.15 [ 0.01 , 0.29 ]
var(z) = 0.01
p-value(z) = 0.04

```

Odds Ratio ES:

```

OR [ 95 %CI] = 1.72 [ 1.03 , 2.88 ]
p-value(OR) = 0.04

Log OR [ 95 %CI] = 0.54 [ 0.03 , 1.06 ]
var(logOR) = 0.07
p-value(Log OR) = 0.04

```

Other:

```

NNT = 10.62
Total N = 197

```

```

>
>

```

```

> # effectsizeパッケージのcohens_dまたはhedges_g関数を使う方法
> library(effectsize)

```

```

> cohens_d(不安~説明, data=d1)
Cohen's d |          95% CI
-----|-----
0.30      | [0.02, 0.58]

```

```

- Estimated using pooled SD.>

```

```

> cohens_d(不安~説明, data=d1, pooled_sd=FALSE)
Cohen's d |          95% CI
-----|-----
0.30      | [0.02, 0.58]

```

```

- Estimated using un-pooled SD.>

```

```

> hedges_g(不安~説明, data=d1)
Hedges' g |          95% CI
-----|-----
0.30      | [0.02, 0.58]

```

```

- Estimated using pooled SD.>

```

```

> hedges_g(不安~説明, data=d1, pooled_sd=FALSE)
Hedges' g |          95% CI
-----|-----
0.30      | [0.02, 0.58]

```

```

- Estimated using un-pooled SD.>

```

## 平均値の非劣性・同等性の検証 — 素データ

```
t.test(実験群データ, 参照群データ, paired=FALSE, conf.level=0.95)
t.test(実験群データ, 参照群データ, paired=FALSE, conf.level=0.90)
```

conf.level のデフォルトは0.95なので、95%CIのときは省略してもよい。  
 平均値差の95%信頼区間、90%信頼区間 をそれぞれ推定して、大小比較する。  
 非劣性マージンはsd/3とするのが一般的。プールした分散を使うか、対照群のsdを使うかは場合による。

```
> rm(list=ls())
> setwd("d:¥¥Report¥¥")

> d1 <- read.table("平均値の非劣性データ2.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 補習 成績
1    1   なし   57
2    2   なし   55
3    3   あり   49
4    4   なし   52
5    5   なし   57
6    6   あり   55
>
```

```
> # 記述統計量
> library(psych)
> describeBy(d1$成績, d1$補習, mat=TRUE, digits=2)
  item group l vars  n mean  sd median trimmed mad min max range skew kurtosis  se
X11    1   あり    1 215 60.27 9.82    60   59.96 8.9  31  88    57 0.19  -0.08 0.67
X12    2   なし    1 185 58.82 9.14    58   58.81 8.9  36  83    47 0.06   0.00 0.67
>
```

```
> # 非劣性を確認したい群データ
> d11 <- d1[d1$補習=="あり",]
>
```

```
> # 参照群データ
> d10 <- d1[d1$補習=="なし",]
>
```

```
> # 平均値差
> mean(d11$成績) - mean(d10$成績)
[1] 1.453551
>
```

```
> # 非劣性マージン： 参照群のSDの1/3
> sd(d10$成績)/3
[1] 3.047505
```

```
> # 平均値の非劣性の検証
```

```
> # 95%信頼区間の推定
> t.test(d11$成績, d10$成績, paired=FALSE, conf.level=0.95)
```

Welch Two Sample t-test

```
data: d11$成績 and d10$成績
t = 1.5318, df = 395.52, p-value = 0.1264
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.4120211  3.3191235
sample estimates:
mean of x mean of y
 60.26977  58.81622
```

```
>
```

	A	B	C
1	番号	補習	成績
2	1	なし	57
3	2	なし	55
4	3	あり	49
5	4	なし	52
6	5	なし	57
7	6	あり	55
8	7	あり	66
9	8	なし	70
10	9	あり	69
11	10	なし	72
12	11	あり	54
13	12	なし	49
14	13	あり	55
15	14	なし	61
16	15	あり	52
17	16	あり	81
18	17	あり	49
19	18	なし	44
20	19	あり	71
21	20	あり	55

> #90%信頼区間の推定

> t.test(d11\$成績, d10\$成績, paired=FALSE, conf.level=0.90)

Welch Two Sample t-test

data: d11\$成績 and d10\$成績

t = 1.5318, df = 395.52, p-value = 0.1264

alternative hypothesis: true difference in means is not equal to 0

90 percent confidence interval:

-0.110960 3.018062

sample estimates:

mean of x mean of y

60.26977 58.81622

標本平均は、実験群60.27、参照群58.82、平均値差は1.45である。

平均値差の検定のp値は0.13で統計的に有意でなく、平均値差に差があるとは言えない。

95%CIの下限 (-0.41)  $\geq$  -非劣性マージン (-3.05) なので、非劣性が言える。

90%CIの下限 (-0.11)  $<$  0 なので、同等性は言えない。

(もし、同等性が言えたら、優越性の判断も行うこと。)

実験群の平均値は、参照群の平均値に比べ遜色ない程度（非劣性）とは言えるが、同等とは言えない。

## 平均値の非劣性・同等性の検証 — 要約統計量

## 自作関数を使う

doutou.m.unpaired(人数0, 平均0, SD0, 人数1, 平均1, SD1)

m1 が m0 に劣らないと言えるかどうかを検討する  
非劣性マージンdを指定することもできる。指定しなければ全体のSDを使う。

```
# -----
doutou.m.unpaired<- function(n0, m0, s0, n1, m1, s1, d=NA) {
  dm <- m1 - m0
  n <- n0 + n1
  df <- n - 2
  s <- sqrt(((n0-1)*s0*s0+(n1-1)*s1*s1)/df)
  se <- s * sqrt(1/n0+1/n1)
  t95 <- qt(0.975, df)
  t90 <- qt(0.950, df)
  L95 <- dm - t95 * se
  H95 <- dm + t95 * se
  L90 <- dm - t90 * se
  H90 <- dm + t90 * se
  ZERO <- 0
  if (is.na(d)) d <- s/3
  MINUS_D <- -d
  if (L95 < MINUS_D) KEKKA <- "非劣性は言えない"
  else if ((L95 <= ZERO) & (L90 < ZERO)) KEKKA <- "非劣性まで言える"
  else if ((L95 <= ZERO) & (L90 >= ZERO)) KEKKA <- "同等以上まで言える"
  else if (L95 > ZERO) KEKKA <- "優越性まで言える"
  else KEKKA <- "Error"

  title <- "Equality Test : equality of means"
  statistic.1 <- data.frame(n0, m0, s0, n1, m1, s1, m1-m0, s, d)
  statistic.2 <- data.frame(MINUS_D, L95, L90, ZERO, H90, H95)
  out <- list(title, statistic.1, statistic.2, KEKKA)
  return(out)
}
# -----
>
>
> # 非劣性を検討する群
> n1 <- 215
> m1 <- 60.27
> s1 <- 9.82
>
> # 参照群
> n0 <- 185
> m0 <- 58.82
> s0 <- 9.14
>
> # 非劣性マージン
> d <- s0/3
>
> doutou.m.unpaired(n0, m0, s0, n1, m1, s1, d)
[[1]]
[1] "Equality Test : equality of means"

[[2]]
      n0      m0      s0      n1      m1      s1 m1...m0      s      d
1 185 58.82 9.14 215 60.27 9.82      1.45 9.511672 3.046667

[[3]]
      MINUS_D      L95      L90 ZERO      H90      H95
1 -3.046667 -0.4252214 -0.1226102      0 3.02261 3.325221

[[4]]
[1] "非劣性まで言える"
```

## 8 分散に関する推測

## 2 群の分散の比較 — 素データ

`var.test`(変数1, 変数2)      または      `var.test`(変数 ~ 群分け変数, データフレーム名)

対応のないt検定においては, 等分散性が仮定できるか否かが問題となる. 等分散性の検定を行って分散が等しいという帰無仮説が保持されれば (棄却されなければ), t検定において `var.equal=TRUE` という指定をすることにより, 検出力を高めることができる.

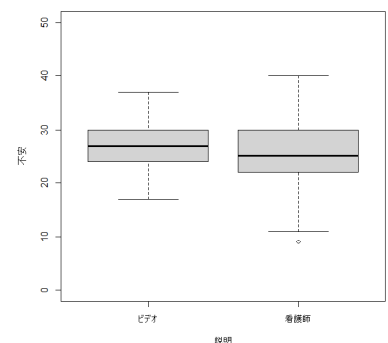
よくわからないときは, 等分散性を仮定しない (何も指定しない). その場合, 自由度は小数点以下を切り捨てて整数にしておいたほうが無難.

```
> setwd("d:¥¥")
> d1 <- read.table("対応のない2群の平均値データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  番号 説明 不安
1    1 ビデオ 26
2    2 看護師 24
3    3 ビデオ 29
4    4 看護師 23
5    5 ビデオ 25
6    6 看護師 23
>
>
> # 記述統計量
> library(psych)
> describeBy(d1[,c("不安")], d1$説明)
group: ビデオ
  vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
1     1 103 27.04 4.32    27   27.12 4.45  17  37   20 -0.1   -0.44 0.43
-----
group: 看護師
  vars   n mean   sd median trimmed  mad min max range skew kurtosis   se
1     1  94 25.44 6.22    25   25.43 5.93   9  40   31 -0.03  -0.15 0.64
>
> # 分散
> tapply(d1[,c("不安")], d1$説明, var)
ビデオ 看護師
18.62593 38.70018
>
> # 等分散性の検定
> var.test(不安 ~ 説明, d1)

      F test to compare two variances

data: 不安 by 説明
F = 0.4813, num df = 102, denom df = 93, p-value = 0.0003419
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.3220954 0.7164806
sample estimates:
ratio of variances
 0.4812879
>
>
> #箱ひげ図
> boxplot(不安 ~ 説明, d1, ylim=c(0, 50))
>
```

	A	B	C
1	番号	説明	不安
2	1	ビデオ	26
3	2	看護師	24
4	3	ビデオ	29
5	4	看護師	23
6	5	ビデオ	25
7	6	看護師	23
8	7	ビデオ	37
9	8	看護師	26
10	9	看護師	20
11	10	ビデオ	36
12	11	ビデオ	28
13	12	ビデオ	27
14	13	ビデオ	27
15	14	ビデオ	33
16	15	看護師	38
17	16	ビデオ	30
18	17	看護師	23
19	18	看護師	28
20	19	ビデオ	28
21	20	ビデオ	23



## 2 群の分散の比較 — 要約統計量

自作関数を使う

known.var.test(人数ベクトル, 分散の値ベクトル)

```
#-----
known.var.test <- function(n, u) {
  if (u[1] < u[2])
    {temp <- u[1]; u[1]<- u[2]; u[2] <- temp; df1 <- n[2]-1; df2 <- n[1]-1}
  else { df1 <- n[1]-1; df2 <- n[2]-1}
  v <- u*u
  F <- u[1]^2/u[2]^2
  p.value <- round(2 *(1 - pf(F, df1, df2)),4)
  title <- "Equality Test : equality of two variances"
  statistic.1 <- data.frame(n, u, v)
  statistic.2 <- data.frame(F, df1, df2, p.value)

  out <- list(title, statistic.1, statistic.2)
  return(out)
}
# -----
>
> #各群の人数
> n <- c(103, 94)
>
> #各群の標準偏差（不偏分散の平方根）
> u <- c(4.32, 6.22)
>

> known.var.test(n, u)
[[1]]
[1] "Equality Test : equality of two variances"

[[2]]
      n      u      v
1 103 6.22 38.6884
2  94 4.32 18.6624

[[3]]
      F df1 df2 p.value
1 2.073067  93 102  4e-04
```



## 多群の分散の比較 — 素データ

`bartlett.test(変数1, 変数2)` または `bartlett.test(変数 ~ 群分け変数, データフレーム名)`

分散分析においては、等分散性が仮定できるか否かが問題となることがある。等分散性の検定を行って分散が等しいという帰無仮説が保持されれば（棄却されなければ）、分散分析において `var.equal=TRUE` という指定をすることにより、検出力を高めることができる。

よくわからないときは、等分散性を仮定しない（何も指定しない）。その場合、自由度は小数点以下を切り捨てて整数にしておいたほうが無難。

```
> setwd("d:¥¥")
> d1 <- read.table("1B平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 group utsu
1    1    60   23
2    2    50   24
3    3    50   28
4    4    40   29
5    5    60   36
6    6    50   22
>
```

```
> # 記述統計量
> library(psych)
> describeBy(d1[,c("utsu")], d1$group)
group: 40
  vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
1    1  73 26.56 5.71    26   26.59 5.93  14  41   27 0.08   -0.38 0.67
-----
group: 50
  vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
1    1  81 29.26 7.09    29   29.25 7.41  13  47   34 0.03   -0.3 0.79
-----
group: 60
  vars  n  mean   sd median trimmed  mad min max range skew kurtosis   se
1    1  84 27.51 6.69    27   27.43 7.41  11  44   33 0.1   -0.53 0.73
>
> # 分散
> tapply(d1[,c("utsu")], d1$group, var)
      40      50      60
32.61073 50.31944 44.80709
>

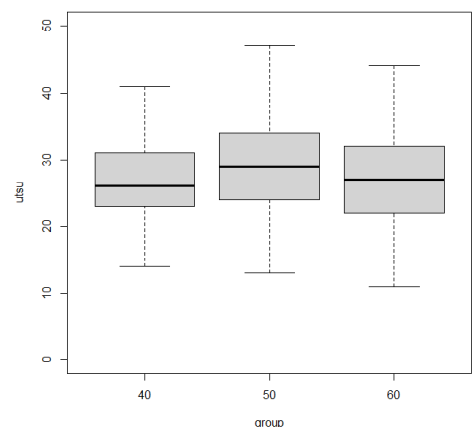
> #等分散性の検定
> bartlett.test(utsu ~ group, d1)
```

Bartlett test of homogeneity of variances

data: utsu by group  
Bartlett's K-squared = 3.6091, df = 2, p-value = 0.1646

```
>
> #箱ひげ図
> boxplot(utsu ~ group, d1, ylim=c(0, 50))
>
>
>
```

	A	B	C
1	番号	group	utsu
2	1	60	23
3	2	50	24
4	3	50	28
5	4	40	29
6	5	60	36
7	6	50	22
8	7	60	36
9	8	50	23
10	9	60	28
11	10	60	27
12	11	60	27
13	12	50	36
14	13	50	33
15	14	40	32
16	15	40	22
17	16	60	44
18	17	50	25
19	18	60	32
20	19	60	20
21	20	40	28



## 多群の分散の比較 — 要約統計量

自作関数を使う

`known.bartlett.test`(人数ベクトル, 分散の値ベクトル)

```

#-----
known.bartlett.test <- function(n, u) {
  v <- u*u
  n1 <- n-1
  sum.Q<- sum(n1 * v)
  q <- length(n)
  df <- q-1
  f <- sum(n)-q
  X.2 <- f*log(sum.Q/f) - sum(n1 * log(v))
  B <- X.2 / (1+1/3/df*(sum(1/n1)-1/f))
  p.value <- round(1 - pchisq(B, df), 4)
  title <- "Equality Test : equality of variances"
  statistic.1 <- data.frame(n, u, v)
  statistic.2 <- data.frame(B, df, p.value)
  out <- list(title, statistic.1, statistic.2)
  return(out)
}
#-----

>
> #各群の人数
> n <- c(73, 81, 84)
>
> #各群の標準偏差（不偏分散の平方根）
> u <- c(5.71, 7.09, 6.69)
>

> known.bartlett.test(n, u)
[[1]]
[1] "Equality Test : equality of variances"

[[2]]
      n      u      v
1 73 5.71 32.6041
2 81 7.09 50.2681
3 84 6.69 44.7561

[[3]]
      B df p.value
1 3.594569 2 0.1657

>
>

```

## 9 多群の平均値に関する推測 — 分散分析・多重比較 —

## 1つの被験者間要因(1B)

## 独立変数の型変換

独立変数 <- as.factor(独立変数)

## 【重要！！】

独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。  
factor型にしていないと、間違った分散分析をした結果を出力してしまう。  
間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

## 1つの被験者間要因の分散分析

## aovを使う方法

aov(従属変数 ~ 独立変数, データフレーム名)

## anovaを使う方法

anova(lm(従属変数 ~ 独立変数, データフレーム名))

## Anovaを使う方法

library(car)

オブジェクト名 <- lm(従属変数 ~ 独立変数, データフレーム名)

Anova(オブジェクト名)

## oneway.testを使う方法(等分散性を仮定する)

oneway.test(従属変数 ~ 独立変数, データフレーム名, var.equal=TRUE)

## oneway.testを使う方法(等分散性を仮定しない)

oneway.test(従属変数 ~ 独立変数, データフレーム名)

上記のうち、Anova, aov, anova(lm()) は等分散性を仮定して分析する。oneway.testは等分散性を仮定することも、仮定しないことも指定できる。

## 多重比較

## テューキー法

TukeyHSD(aov(従属変数 ~ 独立変数, データフレーム名))

または

TukeyHSD(aovの出力結果)

## シェッフェ方法

library(agricolae)

scheffe.test(aov(従属変数 ~ 独立変数, データフレーム名), "要因名")

または

scheffe.test(aovかlmの出力結果, "要因名")

あらかじめagricolaeパッケージをインストールしておく必要がある。

group=FALSE を指定すると、対比較の検定をする。

## ボンフェロニ法

pairwise.t.test(従属変数, 独立変数, p.adjust.method="bonferroni")

p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"

## 効果量

各効果について、 $\eta^2$ 乗, 偏 $\eta^2$ 乗,  $\omega^2$ 乗の全部または一部を計算する。

## effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

library(effectsize)

eta\_squared(分散分析結果オブジェクト, alternative="two.sided")

omega\_squared(分散分析結果オブジェクト, alternative="two.sided")

あらかじめeffectsizeパッケージをインストールしておく必要がある。

分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

aov, anova, Anova, oneway.test関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際、alternative="two.sided" を指定しておく、下限と上限を推定する信頼区間となる。(指定しない場合、上限は1.00となる。)

## DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
EtaSq(aovオブジェクト)
```

あらかじめDescToolsパッケージをインストールしておく必要がある。  
aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。  
typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

## lsrパッケージのetaSquared関数を使う方法

```
library(lsr)
etaSquared(aovオブジェクト)
```

あらかじめlsrパッケージをインストールしておく必要がある。  
aov関数で分散分析を実行した結果オブジェクトをetaSquaredに入れる。  
typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

```
> setwd("i:\Rdocuments\scripts")
> dl <- read.table("1B平均値データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(dl)
  番号 group utsu
1     1    60   23
2     2    50   24
3     3    50   28
4     4    40   29
5     5    60   36
6     6    50   22
>
```

## &gt; # 各群の記述統計量

```
> library(psych)
> describeBy(dl$utsu, dl$group, mat=TRUE, digits=2)
  item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1     40    1  73 26.56 5.71    26  26.59 5.93  14  41   27 0.08  -0.38 0.67
X12    2     50    1  81 29.26 7.09    29  29.25 7.41  13  47   34 0.03  -0.30 0.79
X13    3     60    1  84 27.51 6.69    27  27.43 7.41  11  44   33 0.10  -0.53 0.73
>
```

## &gt; # 全体の記述統計量

```
> describe(dl$utsu)
vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
1     1 238 27.82 6.62    27  27.74 7.41  11  47   36 0.14  -0.27 0.43
>
```

## &gt; # まとめて出力

```
> dtmp <- NULL
> dtmp$x <- dl$utsu
> dtmp$group <- dl$group
>
> dtmp <- data.frame(dtmp)
> ntmp <- as.matrix(table(dtmp$group))
> mtmp <- tapply(dtmp$x, dtmp$group, mean)
> stmp <- tapply(dtmp$x, dtmp$group, sd)
> itmp <- tapply(dtmp$x, dtmp$group, min)
> etmp <- tapply(dtmp$x, dtmp$group, median)
> atmp <- tapply(dtmp$x, dtmp$group, max)
> nta <- nrow(dtmp)
> mta <- mean(dtmp$x)
> sta <- sd(dtmp$x)
> ita <- min(dtmp$x)
> eta <- median(dtmp$x)
```

	A	B	C
1	番号	group	utsu
2	1	60	23
3	2	50	24
4	3	50	28
5	4	40	29
6	5	60	36
7	6	50	22
8	7	60	36
9	8	50	23
10	9	60	28
11	10	60	27
12	11	60	27
13	12	50	36
14	13	50	33
15	14	40	32
16	15	40	22
17	16	60	44
18	17	50	25
19	18	60	32
20	19	60	20
21	20	40	28

```

> ata <- max(dtmp$x)
> otmp <- round(data.frame(ntmp, mtmp, stmp, itmp, etmp, atmp), 2)
> ota <- round(data.frame(nta, mta, sta, ita, eta, ata), 2)
> rownames(ota) <- "Total"
> colnames(otmp) <- colnames(ota) <- c("N", "Mean", "SD", "Min", "Median", "Max")
> otmp <- rbind(otmp, ota)
> otmp

```

```

      N Mean  SD Min Median Max
40    73 26.56 5.71 14     26 41
50    81 29.26 7.09 13     29 47
60    84 27.51 6.69 11     27 44
Total 238 27.82 6.62 11     27 47
>

```

```

> #分散分析
> #aovを使う方法
> d1$group <- as.factor(d1$group)
> result.aov <- aov(utsu ~ group, data=d1)
> summary(result.aov)
      Df Sum Sq Mean Sq F value Pr(>F)
group    2    291   145.67   3.392 0.0353 *
Residuals 235 10093    42.95
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

```

> #anovaを使う方法
> d1$group <- as.factor(d1$group)
> mod.1 <- lm(utsu ~ group, data=d1)
> (result.anova <- anova(mod.1))

```

Analysis of Variance Table

```

Response: utsu
      Df Sum Sq Mean Sq F value Pr(>F)
group    2    291.3   145.675   3.392 0.0353 *
Residuals 235 10092.5    42.947
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

```

> #Anovaを使う方法
> library(car)
> d1$group <- as.factor(d1$group)
> mod.1 <- lm(utsu ~ group, data=d1)
> (result.Anova <- Anova(mod.1))

```

Anova Table (Type II tests)

```

Response: utsu
      Sum Sq Df F value Pr(>F)
group    291.3    2    3.392 0.0353 *
Residuals 10092.5 235
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

```

> #oneway.testを使う方法（等分散性を仮定する）
> d1$group <- as.factor(d1$group)
> result.oneway.equal <- oneway.test(utsu ~ group, data=d1, var.equal=TRUE)
> result.oneway.equal

```

## One-way analysis of means

```
data: utsu and group
F = 3.392, num df = 2, denom df = 235, p-value = 0.0353
>
```

```
> #oneway.testを使う方法（等分散性を仮定しない）
> d1$group <- as.factor(d1$group)
> (result.oneway <- oneway.test(utsu ~ group, data=d1))
```

## One-way analysis of means (not assuming equal variances)

```
data: utsu and group
F = 3.4125, num df = 2.000, denom df = 156.372, p-value = 0.03543
```

## &gt; #多重比較

```
> #テューキーの方法
> (mc.tukey <- TukeyHSD(result.aov))
```

```
Tukey multiple comparisons of means
95% family-wise confidence level
```

```
Fit: aov(formula = utsu ~ group, data = d1)
```

```
$group
      diff      lwr      upr    p adj
50-40  2.6976154  0.2030911  5.1921398 0.0304917
60-40  0.9502609 -1.5230578  3.4235796 0.6369201
60-50 -1.7473545 -4.1544379  0.6597289 0.2027422
```

```
> #シェッフェの方法
> library(agricolae)
```

```
> # group=FALSE を指定したシェッフェの方法
> (mc.scheffe <- scheffe.test(result.aov, "group", group=FALSE))
$statistics
  MSerror Df      F      Mean      CV
42.94688 235 3.034248 27.81513 23.56051
```

```
$parameters
  test name.t ntr alpha
Scheffe group  3  0.05
```

```
$means
      utsu      std  r Min Max Q25 Q50 Q75
40 26.56164 5.710581 73  14  41  23  26  31
50 29.25926 7.093620 81  13  47  24  29  34
60 27.51190 6.693809 84  11  44  22  27  32
```

```
$comparison
      Difference pvalue sig      LCL      UCL
40 - 50 -2.6976154 0.0404  * -5.3029455 -0.09228539
40 - 60 -0.9502609 0.6637    -3.5334434  1.63292150
50 - 60  1.7473545 0.2330    -0.7666505  4.26135954
```

```
$groups
NULL
```

```
attr(,"class")
[1] "group"
```

```

> # group=TRUE を指定したシェッフェの方法
> (mc.scheffe <- scheffe.test(result.aov, "group", group=TRUE))
$statistics
      MSerror Df      F      Mean      CV
42.94688 235 3.034248 27.81513 23.56051

$parameters
      test name.t ntr alpha
Scheffe group    3  0.05

$means
      utsu      std  r Min Max Q25 Q50 Q75
40 26.56164 5.710581 73  14  41  23  26  31
50 29.25926 7.093620 81  13  47  24  29  34
60 27.51190 6.693809 84  11  44  22  27  32

$comparison
NULL

$groups
      utsu groups
50 29.25926      a
60 27.51190     ab
40 26.56164      b

attr(,"class")
[1] "group"

```

```

> #ボンフェロニの方法
> (mc.bon <- pairwise.t.test(d1$utsu, d1$group, p.adjust.method="bonferroni"))

```

Pairwise comparisons using t tests with pooled SD

data: d1\$utsu and d1\$group

```

      40      50
50 0.034 -
60 1.000 0.265

```

```

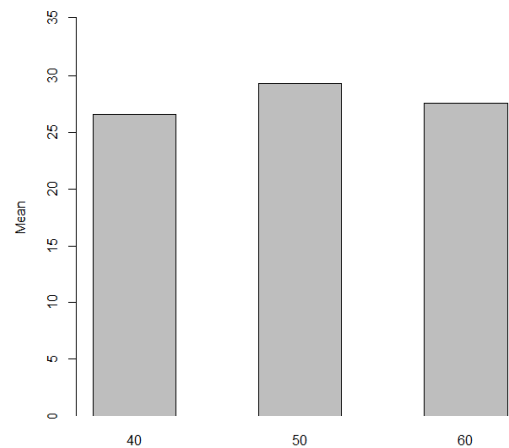
P value adjustment method: bonferroni
>
>

```

```

> #平均値を比較する棒グラフの作成
> barplot(mtemp, ylim=c(0, 35), ylab="Mean", space=1)
> segments(0, 0, 7, 0)
>

```





## &gt; #効果量

```
> # eta_squared関数, omega_squared関数を使う方法
> library(effectsize)
```

```
> eta_squared(result.aov, alternative="two.sided")
```

For one-way between subjects designs, partial eta squared is equivalent to eta squared. Returning eta squared.

```
# Effect Size for ANOVA
```

Parameter	Eta2	95% CI
group	0.03	[0.00, 0.08]

```
> eta_squared(result.anova, alternative="two.sided")
```

For one-way between subjects designs, partial eta squared is equivalent to eta squared. Returning eta squared.

```
# Effect Size for ANOVA
```

Parameter	Eta2	95% CI
group	0.03	[0.00, 0.08]

```
> eta_squared(result.Anova, alternative="two.sided")
```

For one-way between subjects designs, partial eta squared is equivalent to eta squared. Returning eta squared.

```
# Effect Size for ANOVA
```

Parameter	Eta2	95% CI
group	0.03	[0.00, 0.08]

```
> eta_squared(result.oneway.equal, alternative="two.sided")
```

For one-way between subjects designs, partial eta squared is equivalent to eta squared. Returning eta squared.

```
# Effect Size for ANOVA
```

Eta2	95% CI
0.03	[0.00, 0.08]

```
> eta_squared(result.oneway, alternative="two.sided")
```

For one-way between subjects designs, partial eta squared is equivalent to eta squared. Returning eta squared.

```
# Effect Size for ANOVA
```

Eta2	95% CI
0.04	[0.00, 0.11]

```
`警告メッセージ:
```

```
`var.equal = FALSE` - effect size is an approximation.
```

```
> omega_squared(result.aov, alternative="two.sided")
For one-way between subjects designs, partial omega squared is equivalent to omega squared. Returning
omega squared.
# Effect Size for ANOVA
```

Parameter	Omega2	95% CI
group	0.02	[0.00, 0.06]

```
> omega_squared(result.anova, alternative="two.sided")
For one-way between subjects designs, partial omega squared is equivalent to omega squared. Returning
omega squared.
# Effect Size for ANOVA
```

Parameter	Omega2	95% CI
group	0.02	[0.00, 0.06]

```
> omega_squared(result.Anova, alternative="two.sided")
For one-way between subjects designs, partial omega squared is equivalent to omega squared. Returning
omega squared.
# Effect Size for ANOVA
```

Parameter	Omega2	95% CI
group	0.02	[0.00, 0.06]

```
> omega_squared(result.oneway.equal, alternative="two.sided")
For one-way between subjects designs, partial omega squared is equivalent to omega squared. Returning
omega squared.
# Effect Size for ANOVA
```

Omega2	95% CI
0.02	[0.00, 0.06]

```
> omega_squared(result.oneway, alternative="two.sided")
For one-way between subjects designs, partial omega squared is equivalent to omega squared. Returning
omega squared.
# Effect Size for ANOVA
```

Omega2	95% CI
0.03	[0.00, 0.09]

```
警告メッセージ:
`var.equal = FALSE` - effect size is an approximation.
```

```
> #EtaSq関数を使う方法
```

```
> library(DescTools)
> EtaSq(result.aov)
eta.sq eta.sq.part
group 0.02805788 0.02805788
```

```
> #EtaSquared関数を使う方法
```

```
> library(lsr)
> etaSquared(result.aov)
eta.sq eta.sq.part
group 0.02805788 0.02805788
```

## 1つの被験者内要因(1W)

## aovを使う方法

まずstackデータを作成

```
d3 <- stack(被験者内要因の各水準を列とするデータだけのデータフレーム名)
d4 <- data.frame(d3, 被験者IDの変数名)
colnames(d4) <- c("y", "x", "id")
d4$x <- as.factor(d4$x)
d4$id <- as.factor(d4$id)
```

```
aov(y ~ x + id, data=d4)
```

「球面性の仮定」が満たされることを前提としている。

## 【重要！！】

独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。factor型にしていないと、間違った分散分析をした結果を出力してしまう。間違っているかどうかは、「独立変数の自由度Dfの値が“水準数-1”になっているか否か」などでわかる。

## Anovaを使う方法

```
library(car)
被験者内要因名 <- factor(c("条件1", "条件2", "条件3", ...))
全被験者内要因 <- data.frame(被験者内要因名)
lmオブジェクト名 <- lm(cbind(条件1の従属変数, 条件2の従属変数, ...)~1, データフレーム名)
Anovaオブジェクト名 <- Anova(lmオブジェクト名, idata=全被験者内要因, idesign=~被験者内要因名)
summary(Anovaオブジェクト名, multivariate=FALSE)
```

「球面性の仮定」の検討をして、修正した結果も表示する。

あらかじめcarパッケージをインストールしておく必要がある。

被験者内要因名をfactor型にしておく。

被験者間要因はないので、線形モデル(lm)の独立変数(~の右側)は固定値1にしておく。

summaryでmultivariate=TRUEとしておくと、多変量分散分析の結果も表示してくれる。

## 多重比較

## ボンフェロニ法

```
pairwise.t.test(従属変数, 独立変数, p.adjust.method="bonferroni", paired=TRUE)
```

p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"  
の値を変えれば、他の方法での多重比較も行える

## 効果量

各効果について、 $\eta^2$ 乗、偏 $\eta^2$ 乗、 $\omega^2$ 乗の全部または一部を計算する。

## effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```
library(effectsize)
eta_squared(分散分析オブジェクト, alternative="two.sided")
omega_squared(分散分析オブジェクト, alternative="two.sided")
```

あらかじめsjstatsパッケージをインストールしておく必要がある。

aov, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際、alternative="two.sided"を指定しておくと、下限と上限を推定する信頼区間となる。(指定しない場合、上限は1.00となる。)

## DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
EtaSq(aovオブジェクト)
```

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをEtaSqに入れる。

typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

## lsrパッケージのetaSquared関数を使う方法

library(lsr)

etaSquared(aovオブジェクト)

あらかじめlsrパッケージをインストールしておく必要がある。  
 aov関数で分散分析を実行した結果オブジェクトをetaSquaredに入れる。  
 typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

```
> setwd("d:¥¥")
> d1 <- read.table("1W平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  Team Rank Cooperate
1   13   15         14
2   15   15         10
3   11   13         13
4   14   14         18
5   15   10         14
6   11   11         14
>
>
> # 各群の人数・平均値・標準偏差
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD Team Rank Cooperate
Team  164 12.42 1.63 1.00 0.21      0.20
Rank  164 12.12 1.57 0.21 1.00      0.21
Cooperate 164 13.30 1.63 0.20 0.21      1.00
>

> #stackデータの作成
> d3 <- stack(dtmp)
> did <- c(1:nrow(dtmp))
> d4 <- data.frame(d3, did)
> colnames(d4) <- c("y", "x", "id")
> d4$x <- as.factor(d4$x)
> d4$id <- as.factor(d4$id)
> head(d4)
   y    x id
1 13 Team  1
2 15 Team  2
3 11 Team  3
4 14 Team  4
5 15 Team  5
6 11 Team  6
>
>

> #分散分析
> #aovを使う方法
> result.aov <- aov(y ~ x + id, data=d4)
> summary(result.aov)

      Df Sum Sq Mean Sq F value    Pr(>F)
x         2   124.1    62.05   30.229 9.05e-13 ***
id        163   595.1     3.65    1.778 6.35e-06 ***
Residuals   326   669.2     2.05
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```
> # Anovaを使う方法
> library(car)
> fac.a <- factor(c("Team", "Rank", "Cooperate"))
> youin <- data.frame(fac.a)
> mod.1 <- lm(cbind(Team, Rank, Cooperate)~1, d1)
> result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.a)
Note: model has only an intercept; equivalent type-III tests substituted.
> summary(result.Anova, multivariate=FALSE)
```

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

	SS	num Df	Error SS	den Df	F	Pr(>F)
(Intercept)	78307	1	595.06	163	21449.790	< 2.2e-16 ***
fac.a	124	2	669.22	326	30.229	9.054e-13 ***

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Mauchly Tests for Sphericity

	Test statistic	p-value
fac.a	0.99895	0.91826

Greenhouse-Geisser and Huynh-Feldt Corrections  
for Departure from Sphericity

	GG eps	Pr(>F[GG])
fac.a	0.99895	9.284e-13 ***

---  
 Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	HF eps	Pr(>F[HF])
fac.a	1.011338	9.053813e-13

警告メッセージ:  
 In summary.Anova.mlm(result.Anova, multivariate = FALSE) :  
 HF eps > 1 treated as 1  
 >  
 >

```
> # 多重比較
> pairwise.t.test(d4$y, d4$x, p.adjust.method="bonferroni", paired=TRUE)
```

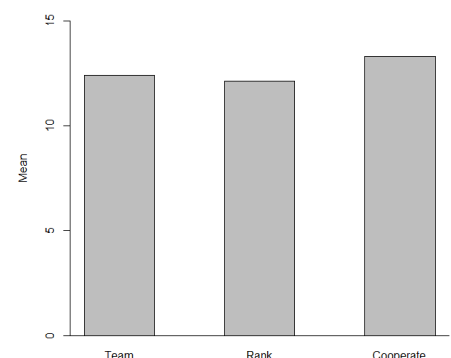
Pairwise comparisons using paired t tests

data: d4\$y and d4\$x

	Team	Rank
Rank	0.17	-
Cooperate	4.3e-07	1.0e-11

P value adjustment method: bonferroni >

```
> #平均値を比較するグラフ
> barplot(colMeans(dtmp), ylim=c(0, 16), ylab="Mean", space=1)
> segments(0, 0, 7, 0)
>
>
```



> # 効果量

```
> #eta_squared, omega_squared関数を使う方法
> library(effectsize)
> eta_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type I)
```

Parameter	Eta2 (partial)	95% CI
x	0.16	[0.09, 0.23]
id	0.47	[0.16, 0.54]

```
> omega_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type I)
```

Parameter	Omega2 (partial)	95% CI
x	0.11	[0.05, 0.17]
id	0.21	[0.00, 0.00]

```
> eta_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type III)
```

Parameter	Eta2 (partial)	95% CI
fac.a	0.16	[0.09, 0.23]

```
> omega_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type III)
```

Parameter	Omega2 (partial)	95% CI
fac.a	0.09	[0.03, 0.15]

> #EtaSq関数を使う方法

```
> library(DescTools)
> EtaSq(result.aov)
      eta.sq eta.sq.part
x  0.08939073  0.1564409
id 0.42859736  0.4706710
>
```

> #lsr関数を使う方法

```
> library(lsr)
> etaSquared(result.aov)
      eta.sq eta.sq.part
x  0.08939073  0.1564409
id 0.42859736  0.4706710
```

**2つの被験者間要因(2B)**

```

データフレーム名： d1
被験者ID：          id
被験者間要因1：     x1
被験者間要因2：     x2
従属変数：          y

```

**独立変数の型変換**

```

d1$x1 <- as.factor(d1$x1)
d1$x2 <- as.factor(d1$x2)

```

**【重要！！】**

独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。  
factor型にしていないと、間違った分散分析をした結果を出力してしまう。  
間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

**aovを使う方法**

```

lmオブジェクト名 <- lm(y ~ x1 * x2, data=d1)
aovオブジェクト名 <- aov(lmオブジェクト名)
summary(aovオブジェクト名)

```

**anovaを使う方法**

```

lmオブジェクト名 <- lm(y ~ x1 * x2, data=d1)
anovaオブジェクト名 <- anova(lmオブジェクト名)
anovaオブジェクト名

```

**Anovaを使う方法**

```

library(car)
lmオブジェクト名 <- lm(y ~ x1 * x2, d1)
Anova オブジェクト名 <- Anova(lmオブジェクト名)
Anova オブジェクト名

```

あらかじめcarパッケージをインストールしておく必要がある。

被験者間要因が2つあるので、線形モデル(lm)の独立変数を「独立変数1 \* 独立変数2」とする。

**多重比較 テューキー法**

```
TukeyHSD(aov(y ~ x1 * x2, d1) または TukeyHSD(aovの出力結果))
```

**効果量**

各効果について、 $\eta^2$ 乗、偏 $\eta^2$ 乗、 $\omega^2$ 乗の全部または一部を計算する。

**effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法**

```

library(effectsize)
eta_squared(分散分析オブジェクト, alternative="two.sided")
omega_squared(分散分析オブジェクト, alternative="two.sided")

```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, anova, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際、alternative="two.sided"を指定しておく、下限と上限を推定する信頼区間となる。(指定しない場合、上限は1.00となる。)

**DescToolsパッケージのEtaSq関数を使う方法**

```

library(DescTools)
EtaSq(aovオブジェクト)

```

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。

typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

## lsrパッケージのetaSquared関数を使う方法

library(lsr)

etaSquared(aovオブジェクト)

あらかじめlsrパッケージをインストールしておく必要がある。  
 aov関数で分散分析を実行した結果オブジェクトをetaSquaredに入れる。  
 typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

## 単純効果の検定

library(phia)

testInteractions(lmオブジェクト名, fixed="変数名1", across="変数名2")

または

testInteractions(lmオブジェクト名, fixed="変数名1", pairwise="変数名2")

あらかじめphiaパッケージをインストールしておく必要がある。  
 fixedで指定した変数の水準ごとに、acrossで指定した変数の単純主効果の検定、pairwiseで指定した変数の任意の2水準間の効果の検定を行う。

## 調整済み平均値の推定と作図

(オブジェクト名 &lt;- interactionMeans(lmオブジェクト名))

plot(オブジェクト名)

一方の変数の各水準における、他方の変数の平均値を比較した折れ線グラフを作図する

```
> setwd("i:\Rdocuments\scripts")
> d1 <- read.table("2B平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id work  future tekiou
1  1  few  shingaku    17
2  2  few  shingaku    13
3  3  few  shingaku    12
4  4  few  shingaku    14
5  5  few  shushoku    10
6  6  few  shushoku    12
>
>
```

## &gt; # 独立変数をfactor型に変換

```
> d1$work <- factor(d1$work, levels=c("many", "few"), labels=c("Many", "Few"))
> d1$future <- factor(d1$future, levels=c("shingaku", "shushoku", "mitei"),
+                      labels=c("Shingaku", "Shushoku", "Mitei"))
```

```
> head(d1)
  id work  future tekiou
1  1 Few Shingaku    17
2  2 Few Shingaku    13
3  3 Few Shingaku    12
4  4 Few Shingaku    14
5  5 Few Shushoku    10
6  6 Few Shushoku    12
>
```

## &gt; # 基本統計量 (psychパッケージのdescribeBy関数を使う方法)

library(psych)

&gt;

	A	B	C	D
1	id	work	future	tekiou
2		1 few	shingaku	17
3		2 few	shingaku	13
4		3 few	shingaku	12
5		4 few	shingaku	14
6		5 few	shushoku	10
7		6 few	shushoku	12
8		7 few	shushoku	12
9		8 few	shushoku	11
10		9 few	mitei	13
11		10 few	mitei	13



```

> # セル単位
> (stats_c <- describeBy(d1$tekiou, list(d1$work, d1$future), mat=TRUE, digits=2))
  item group1 group2 vars  n mean  sd median trimmed mad min max range skew kurtosis se
X11    1   Many Shingaku   1 24 14.38 1.61  14.0   14.30 1.48 12 18    6  0.43   -0.57 0.33
X12    2    Few Shingaku   1 16 14.19 2.48  14.5   14.29 2.22  9 18    9 -0.42   -0.84 0.62
X13    3   Many Shushoku   1 64 12.84 1.77  13.0   12.79 1.48  9 17    8  0.28   -0.22 0.22
X14    4    Few Shushoku   1 80 12.29 2.06  12.0   12.12 1.48  9 19   10  0.65    0.21 0.23
X15    5   Many Mitei     1 24 10.38 2.00  10.5   10.40 2.22  6 14    8 -0.15   -0.41 0.41
X16    6    Few Mitei     1 18 12.00 1.78  12.0   12.00 1.48  9 15    6  0.06   -0.92 0.42
>
> # 変数単位
> describeBy(d1$tekiou, list(d1$work), mat=TRUE, digits=2)
  item group1 vars  n mean  sd median trimmed mad min max range skew kurtosis se
X11    1   Many   1 112 12.64 2.22    13   12.67 1.48  6 18    12 -0.19    0.16 0.21
X12    2    Few   1 114 12.51 2.18    12   12.38 2.22  9 19    10  0.50   -0.23 0.20
>
> describeBy(d1$tekiou, list(d1$future), mat=TRUE, digits=2)
  item group1 vars  n mean  sd median trimmed mad min max range skew kurtosis se
X11    1 Shingaku   1  40 14.30 1.98    14   14.34 1.48  9 18    9 -0.25   -0.09 0.31
X12    2 Shushoku   1 144 12.53 1.95    12   12.43 1.48  9 19   10  0.46    0.02 0.16
X13    3 Mitei     1  42 11.07 2.05    11   11.12 1.48  6 15    9 -0.16   -0.28 0.32
>
> # 全体
> describe(d1$tekiou)
  vars  n mean  sd median trimmed mad min max range skew kurtosis se
X1     1 226 12.58 2.19    12   12.52 2.97  6 19    13 0.15   -0.05 0.15

> #分散分析
> # aovを使う方法
> mod.1 <- lm(tekiou~work*future, data=d1)
> result.aov <- aov(mod.1)
> summary(result.aov)
          Df Sum Sq Mean Sq F value    Pr(>F)
work          1      1.0     1.02    0.269  0.60448
future        2    213.6   106.81   28.294 1.16e-11 ***
work:future    2     38.1    19.04    5.043  0.00722 **
Residuals    220    830.5     3.78
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
>
>
> # anovaを使う方法
> mod.1 <- lm(tekiou~work*future, data=d1)
> result.anova <- anova(mod.1)
> result.anova
Analysis of Variance Table

Response: tekiou
          Df Sum Sq Mean Sq F value    Pr(>F)
work          1      1.02    1.016    0.2691  0.604483
future        2    213.62   106.810   28.2936 1.161e-11 ***
work:future    2     38.07    19.036    5.0427  0.007223 **
Residuals    220    830.51     3.775
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
>
>
> # Anovaを使う方法
> library(car)
> mod.1 <- lm(tekiou~work*future, data=d1)
> result.Anova <- Anova(mod.1)
> result.Anova

Anova Table (Type II tests)

```

```

Response: tekiou
      Sum Sq Df F value    Pr(>F)
work      0.43  1  0.1130  0.737057
future    213.62  2 28.2936 1.161e-11 ***
work:future 38.07  2  5.0427  0.007223 **
Residuals  830.51 220
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
>

```

> #多重比較

> #テューキーの方法

```

> (mc.tukey <- TukeyHSD(result.aov)))
  Tukey multiple comparisons of means
    95% family-wise confidence level

```

```
Fit: aov(formula = mod.1)
```

```

$work
      diff      lwr      upr      p adj
Few-Many -0.1340852 -0.6435314 0.375361 0.6044829

```

```

$future
      diff      lwr      upr      p adj
Shushoku-Shingaku -1.74442 -2.563854 -0.9249864 3.10e-06
Mitei-Shingaku    -3.22474 -4.237645 -2.2118362 0.00e+00
Mitei-Shushoku    -1.48032 -2.284340 -0.6763006 6.31e-05

```

```

$`work:future`
      diff      lwr      upr      p adj
Few:Shingaku-Many:Shingaku -0.18750 -1.9903740 1.6153740 0.9996764
Many:Shushoku-Many:Shingaku -1.53125 -2.8682971 -0.1942029 0.0145212
Few:Shushoku-Many:Shingaku -2.08750 -3.3875709 -0.7874291 0.0000963
Many:Mitei-Many:Shingaku    -4.00000 -5.6125395 -2.3874605 0.0000000
Few:Mitei-Many:Shingaku     -2.37500 -4.1167417 -0.6332583 0.0016313
Many:Shushoku-Few:Shingaku  -1.34375 -2.9050847  0.2175847 0.1363042
Few:Shushoku-Few:Shingaku   -1.90000 -3.4297893 -0.3702107 0.0057706
Many:Mitei-Few:Shingaku     -3.81250 -5.6153740 -2.0096260 0.0000001
Few:Mitei-Few:Shingaku      -2.18750 -4.1068059 -0.2681941 0.0152697
Few:Shushoku-Many:Shushoku  -0.55625 -1.4930508  0.3805508 0.5285578
Many:Mitei-Many:Shushoku    -2.46875 -3.8057971 -1.1317029 0.0000040
Few:Mitei-Many:Shushoku     -0.84375 -2.3340773  0.6465773 0.5811320
Many:Mitei-Few:Shushoku     -1.91250 -3.2125709 -0.6124291 0.0004879
Few:Mitei-Few:Shushoku      -0.28750 -1.7447457  1.1697457 0.9930198
Few:Mitei-Many:Mitei        1.62500 -0.1167417  3.3667417 0.0829213

```

> #シェッフェの方法

```
> library(agricolae)
```

```
> (mc.scheffe <- scheffe.test(result.aov, "work"))
```

```

$statistics
  MSerror Df      F      Mean      CV
3.775057 220 3.884075 12.57522 15.45063

```

```

$parameters
  test name.t ntr alpha
Scheffe  work  2  0.05

```

```

$means
      tekiou      std      r Min Max Q25 Q50 Q75
Few 12.50877 2.178924 114    9  19  11  12  14
Many 12.64286 2.217283 112    6  18  11  13  14

```

```

$comparison
      Difference pvalue sig      LCL      UCL
Few - Many -0.1340852 0.6045    -0.6435314 0.375361

$groups
NULL

attr(,"class")
[1] "group"

> (mc.scheffe <- scheffe.test(result.aov, "future"))
$statistics
      MSerror Df      F      Mean      CV
3.775057 220 3.036898 12.57522 15.45063

$parameters
      test name.t ntr alpha
Scheffe future   3  0.05

$means
      tekiou      std      r Min Max Q25 Q50 Q75
Mitei  11.07143 2.052873  42   6  15  10  11 12.00
Shingaku 14.30000 1.976788  40   9  18  13  14 15.25
Shushoku 12.53472 1.950124 144   9  19  11  12 14.00

$comparison
      Difference pvalue sig      LCL      UCL
Mitei - Shingaku -3.228571 0e+00 *** -4.2864702 -2.1706726
Mitei - Shushoku -1.463294 1e-04 *** -2.3030290 -0.6235583
Shingaku - Shushoku 1.765278 0e+00 ***  0.9094438  2.6211117

$groups
NULL

attr(,"class")
[1] "group"

> # group=FALSE を指定したシェッフェの方法
> (mc.scheffe <- scheffe.test(result.aov, "work", group=FALSE))
$statistics
      MSerror Df      F      Mean      CV
3.775057 220 3.884075 12.57522 15.45063

$parameters
      test name.t ntr alpha
Scheffe  work   2  0.05

$means
      tekiou      std      r Min Max Q25 Q50 Q75
Few  12.50877 2.178924 114   9  19  11  12 14
Many 12.64286 2.217283 112   6  18  11  13 14

$comparison
      Difference pvalue sig      LCL      UCL
Few - Many -0.1340852 0.6045    -0.6435314 0.375361

$groups
NULL

attr(,"class")
[1] "group"

```

```
> (mc.scheffe <- scheffe.test(result.aov, "future", group=FALSE))
$statistics
      MSerror Df      F      Mean      CV
      3.775057 220 3.036898 12.57522 15.45063

$parameters
      test name.t ntr alpha
      Scheffe future   3  0.05

$means
      tekiou      std      r Min Max Q25 Q50  Q75
Mitei    11.07143 2.052873  42   6  15  10  11 12.00
Shingaku 14.30000 1.976788  40   9  18  13  14 15.25
Shushoku 12.53472 1.950124 144   9  19  11  12 14.00

$comparison
      Difference pvalue sig      LCL      UCL
Mitei - Shingaku -3.228571 0e+00 *** -4.2864702 -2.1706726
Mitei - Shushoku -1.463294 1e-04 *** -2.3030290 -0.6235583
Shingaku - Shushoku 1.765278 0e+00 ***  0.9094438  2.6211117

$groups
NULL

attr(,"class")
[1] "group"
```

### > #ボンフェロニの方法

```
> (mc.bon.tekiou <- pairwise.t.test(d1$tekiou, d1$work, p.adjust.method="bonferroni"))
```

Pairwise comparisons using t tests with pooled SD

data: d1\$tekiou and d1\$work

```
      Many
Few 0.65
```

P value adjustment method: bonferroni

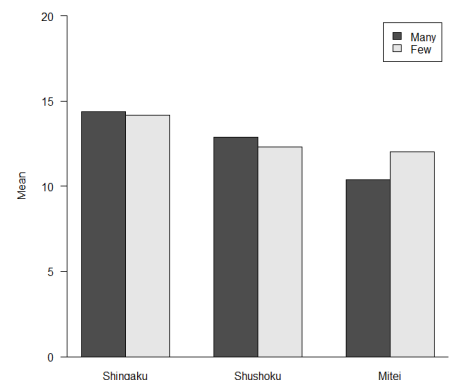
```
> (mc.bon.gakunen <- pairwise.t.test(d1$tekiou, d1$future, p.adjust.method="bonferroni"))
```

Pairwise comparisons using t tests with pooled SD

data: d1\$tekiou and d1\$future

```
      Shingaku Shushoku
Shushoku 3.4e-06 -
Mitei    8.1e-12 1e-04
```

P value adjustment method: bonferroni



### > #平均値を比較する棒グラフの作成

```
> mlt <- tapply(d1[, "tekiou"], list(d1[, "work"], d1[, "future"]), mean)
> barplot(mlt, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean")
```

> # 効果量

> #eta\_squared, omega\_squared関数を使う方法

> library(effectsize)

> eta\_squared(result.aov, alternative="two.sided")

# Effect Size for ANOVA (Type I)

Parameter	Eta2 (partial)	95% CI
work	1.22e-03	[0.00, 0.03]
future	0.20	[0.12, 0.29]
work:future	0.04	[0.00, 0.10]

> omega\_squared(result.aov, alternative="two.sided")

# Effect Size for ANOVA (Type I)

Parameter	Omega2 (partial)	95% CI
work	0.00	[0.00, 0.00]
future	0.19	[0.11, 0.28]
work:future	0.03	[0.00, 0.09]

> eta\_squared(result.anova, alternative="two.sided")

# Effect Size for ANOVA (Type I)

Parameter	Eta2 (partial)	95% CI
work	1.22e-03	[0.00, 0.03]
future	0.20	[0.12, 0.29]
work:future	0.04	[0.00, 0.10]

> omega\_squared(result.anova, alternative="two.sided")

# Effect Size for ANOVA (Type I)

Parameter	Omega2 (partial)	95% CI
work	0.00	[0.00, 0.00]
future	0.19	[0.11, 0.28]
work:future	0.03	[0.00, 0.09]

>

> eta\_squared(result.Anova, alternative="two.sided")

# Effect Size for ANOVA (Type II)

Parameter	Eta2 (partial)	95% CI
work	5.13e-04	[0.00, 0.02]
future	0.20	[0.12, 0.29]
work:future	0.04	[0.00, 0.10]

> omega\_squared(result.Anova, alternative="two.sided")

# Effect Size for ANOVA (Type II)

Parameter	Omega2 (partial)	95% CI
work	0.00	[0.00, 0.00]
future	0.19	[0.11, 0.28]
work:future	0.03	[0.00, 0.09]

>

>

```
> #EtaSq関数を使う方法
```

```
> library(DescTools)
```

```
> EtaSq(result.aov)
```

```
      eta.sq  eta.sq.part
work      0.0003938638 0.0005134451
future    0.1972081460 0.2045909323
work:future 0.0351479090 0.0438332936
```

```
> #etaSquared関数を使う方法
```

```
> library(lsr)
```

```
> etaSquared(result.aov)
```

```
      eta.sq  eta.sq.part
work      0.0003938638 0.0005134451
future    0.1972081460 0.2045909323
work:future 0.0351479090 0.0438332936
```

```
>
```

```
> # 単純効果
```

```
> library(phia)
```

```
>
```

```
> # workの各水準におけるfutureの主効果
```

```
> testInteractions(mod.1, fixed="work", across="future")
```

```
F Test:
```

```
P-value adjustment method: holm
```

	future1	future2	Df	Sum of Sq	F	Pr(>F)	
Many	4.0000	2.4688	2	198.03	26.228	1.216e-10	***
Few	2.1875	0.2875	2	53.67	7.108	0.00102	**
Residuals			220	830.51			

```
----
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

```
>
```

```
> # workの各水準におけるfuture2水準間の効果
```

```
> testInteractions(mod.1, fixed="work", pairwise="future")
```

```
F Test:
```

```
P-value adjustment method: holm
```

	Value	Df	Sum of Sq	F	Pr(>F)	
Shingaku-Shushoku : Many	1.5313	1	40.93	10.8412	0.003469	**
Shingaku-Mitei : Many	4.0000	1	192.00	50.8602	8.473e-11	***
Shushoku-Mitei : Many	2.4687	1	106.38	28.1799	1.348e-06	***
Shingaku-Shushoku : Few	1.9000	1	48.13	12.7504	0.001748	**
Shingaku-Mitei : Few	2.1875	1	40.53	10.7371	0.003469	**
Shushoku-Mitei : Few	0.2875	1	1.21	0.3217	0.571149	
Residuals		220	830.51			

```
----
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

```
>
```

```
> # futureの各水準におけるworkの主効果
```

```
> testInteractions(mod.1, fixed="future", across="work")
```

```
F Test:
```

```
P-value adjustment method: holm
```

	Value	Df	Sum of Sq	F	Pr(>F)	
Shingaku	0.18750	1	0.34	0.0894	0.7652	
Shushoku	0.55625	1	11.00	2.9142	0.1784	
Mitei	-1.62500	1	27.16	7.1948	0.0236	*
Residuals		220	830.51			

```
----
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
```

```
>
```

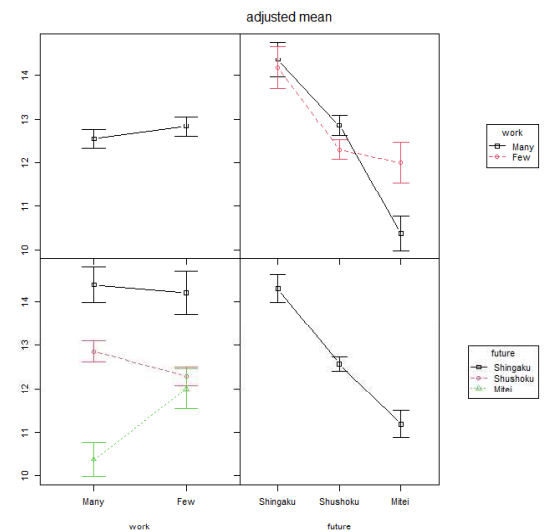
```
> # futureの各水準におけるwork2水準間の効果
> testInteractions(mod.1, fixed="future", pairwise="work")
F Test:
P-value adjustment method: holm
      Value Df Sum of Sq      F Pr(>F)
Many-Few : Shingaku  0.18750   1      0.34 0.0894 0.7652
Many-Few : Shushoku  0.55625   1     11.00 2.9142 0.1784
Many-Few : Mitei    -1.62500   1     27.16 7.1948 0.0236 *
Residuals                220     830.51

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1
>
>
```

```
> # モデルによって調整された各セルの平均値とその概観
```

```
> (mn.dladj <- interactionMeans(mod.1))
      work  future adjusted mean std. error
1 Many Shingaku    14.37500    0.3966031
2 Few  Shingaku    14.18750    0.4857376
3 Many Shushoku    12.84375    0.2428688
4 Few  Shushoku    12.28750    0.2172285
5 Many Mitei      10.37500    0.3966031
6 Few  Mitei      12.00000    0.4579578
```

```
> plot(mn.dladj)
>
```



## 1つの被験者間要因と1つの被験者内要因(1B1W)

## stackデータの作成

被験者内要因の水準ごとにstackデータを作成して、最後に縦に繋げる

```
vn.x <- c("mh1", "mh2")
d3 <- NULL
for(i in vn.x) {
  d2 <- d1
  d2$x <- i
  d2$y <- d1[,i]
  d3 <- rbind(d3, d2)
}
d4 <- d3[, c("id", "group", "x", "y")]
d4$id <- as.factor(d4$id)
d4$group <- as.factor(d4$group)
d4$x <- as.factor(d4$x)
head(d4)
```

データフレーム名 : d4  
 被験者ID : id  
 被験者間要因変数 : group  
 被験者内要因変数 : x  
 従属変数 : y

## 【重要！！】

被験者idも含め、独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。

factor型にしていないと、間違った分散分析をした結果を出力してしまう。

間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

## 分散分析

## aovを使う方法

```
result.aov <- aov(y ~ group * x + Error(id + id:group + id:group:x), data=d4)
summary(result.aov)
```

「球面性の仮定」が満たされることを前提としている。

## Anovaを使う方法

データフレーム名 : d1  
 被験者ID : id  
 被験者間要因変数 : group  
 被験者内要因名 : fac.b <- factor(c("b1", "b2", "b3"))  
 全被験者内要因 : youin  
 従属変数 : y

```
library(car)
d1$group <- as.factor(d1$group)
fac.b <- factor(c("b1", "b2", "b3"))
youin <- data.frame(fac.b)
mod.1 <- lm(cbind(b1, b2, b3) ~ group, c1)
result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.b)
summary(result.Anova, multivariate=FALSE)
```

「球面性の仮定」の検討をして、修正した結果も表示する。

あらかじめcarパッケージをインストールしておく必要がある。

multivariate=TRUE としておく、多変量分散分析の結果も表示してくれる。

## 多重比較

## 被験者間要因

```
pairwise.t.test(d1$y, d1$group, p.adjust.method="bonferroni")
```

## 被験者内要因

```
pairwise.t.test(d1$y, d1$x, p.adjust.method="bonferroni", paired=TRUE)
```

p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"



## 効果量

effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```
library(effectsize)
```

```
eta_squared(分散分析オブジェクト, alternative="two.sided")
```

```
omega_squared(分散分析オブジェクト, alternative="two.sided")
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際, alternative="two.sided" を指定しておく, 下限と上限を推定する信頼区間となる。(指定しない場合, 上限は1.00となる。)

DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
```

```
EtaSq(aovオブジェクト, type=1)
```

各効果について,  $\eta^2$  乗, 偏  $\eta^2$  乗, 一般化  $\eta^2$  乗の値を計算する。

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。

typeオプションでtype=1と指定する。(デフォルトはtype=2)

```
> setwd("d:¥¥")
> d1 <- read.table("1B1W平均値データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
```

```
> head(d1)
  id mh1 mh2   group
1  1  11  18 experiment
2  2  18  15 experiment
3  3  16  14   control
4  4   9  16 experiment
5  5  17  16 experiment
6  6   9  17 experiment
```

```
>
>
```

```
> # stackデータの作成
> d2 <- d1[, c("mh1", "mh2")]
> d3 <- stack(d2)
> d4 <- data.frame(d1$id, d1$group, d3)
> colnames(d4) <- c("id", "group", "y", "x")
> head(d4)
```

```
  id   group y   x
1  1 experiment 11 mh1
2  2 experiment 18 mh1
3  3   control 16 mh1
4  4 experiment  9 mh1
5  5 experiment 17 mh1
6  6 experiment  9 mh1
```

```
>
>
```

```
> # 独立変数をfactor型に変換
> d4$id <- factor(d4$id)
> d4$group <- factor(d4$group)
> d4$x <- factor(d4$x)
>
```

```
> # 基本統計量
> library(psych)
```

1	id	mh1	mh2	group
2	1	11	18	experiment
3	2	18	15	experiment
4	3	16	14	control
5	4	9	16	experiment
6	5	17	16	experiment
7	6	9	17	experiment
8	7	12	17	experiment
9	8	12	17	experiment
10	9	11	16	experiment
11	10	10	13	experiment
12	11	16	16	experiment
13	12	9	11	experiment
14	13	14	15	control
15	14	12	13	experiment
16	15	13	12	experiment
17	16	13	11	experiment
18	17	14	16	control
19	18	11	15	experiment
20	19	14	16	control
21	20	16	15	control

```

>
> # セル単位
> describeBy(d4$y, list(d4$x, d4$group), mat=TRUE, digits=2)
  item group1 group2 vars n mean sd median trimmed mad min max
X11 1 mh1 control 1 50 15.48 1.67 15 15.53 1.48 12 19
X12 2 mh2 control 1 50 15.32 1.88 15 15.30 1.48 11 20
X13 3 mh1 experiment 1 47 12.26 2.31 12 12.15 1.48 8 18
X14 4 mh2 experiment 1 47 14.53 2.38 15 14.56 2.97 10 20
  range skew kurtosis se
X11 7 -0.17 -0.59 0.24
X12 9 0.09 -0.28 0.27
X13 10 0.47 -0.33 0.34
X14 10 -0.15 -0.53 0.35

> # 変数単位
> describeBy(d4$y, list(d4$x), mat=TRUE, digits=2)
  item group1 vars n mean sd median trimmed mad min max range skew
X11 1 mh1 1 97 13.92 2.57 14 14.01 2.97 8 19 11 -0.23
X12 2 mh2 1 97 14.94 2.16 15 14.99 1.48 10 20 10 -0.19
  kurtosis se
X11 -0.84 0.26
X12 -0.15 0.22

> describeBy(d4$y, list(d4$group), mat=TRUE, digits=2)
  item group1 vars n mean sd median trimmed mad min max range
X11 1 control 1 100 15.40 1.77 15 15.41 1.48 11 20 9
X12 2 experiment 1 94 13.39 2.60 13 13.37 2.97 8 20 12
  skew kurtosis se
X11 -0.04 -0.33 0.18
X12 0.14 -0.74 0.27
>

> # 全体
> describe(d4$y)
  vars n mean sd median trimmed mad min max range skew kurtosis se
X1 1 194 14.43 2.42 15 14.51 2.97 8 20 12 -0.31 -0.42 0.17
>

> # 分散分析
> #aovを使う方法
> result.aov <- aov(y ~ group * x + Error(id + id:group + id:group:x), data=d4)

警告メッセージ:
In aov(y ~ group * x + Error(id + id:group + id:group:x), data = d4) :
  Error() モデルは特異です

> summary(result.aov)

Error: id
      Df Sum Sq Mean Sq F value Pr(>F)
group 1 195.1 195.05 39.18 1.11e-08 ***
Residuals 95 472.9 4.98
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:group:x
      Df Sum Sq Mean Sq F value Pr(>F)
x 1 50.5 50.52 14.07 0.000303 ***
group:x 1 71.9 71.92 20.03 2.12e-05 ***
Residuals 95 341.1 3.59
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

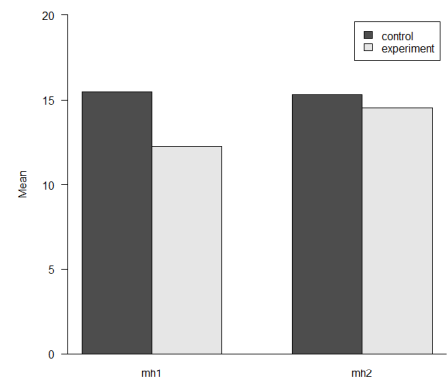
```
> # Anovaを使う方法
> library(car)
> d1$group <- as.factor(d1$group)
> fac.b <- factor(c("mh1", "mh2"))
> youin <- data.frame(fac.b)
> mod.1 <- lm(cbind(mh1, mh2) ~ group, data=d1)
> result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.b)
> summary(result.Anova, multivariate=FALSE)
```

Univariate Type II Repeated-Measures ANOVA Assuming Sphericity

	SS	num Df	Error SS	den Df	F	Pr(>F)
(Intercept)	40384	1	472.94	95	8111.948	< 2.2e-16 ***
group	195	1	472.94	95	39.181	1.111e-08 ***
fac.b	51	1	341.06	95	14.072	0.0003026 ***
group:fac.b	72	1	341.06	95	20.032	2.115e-05 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

```
> # 平均値を比較する棒グラフの作成
> m4t <- tapply(d4$y, list(d4$group$, d4$x), mean)
> barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean")
> segments(0, 0, 10, 0)
>
```



```
> # ボンフェロニ法を用いた多重比較
> d4mh1 <- subset(d4, x=="mh1")
> d4mh2 <- subset(d4, x=="mh2")
> d4control <- subset(d4, group=="control")
> d4experiment <- subset(d4, group=="experiment")
>
> # mh1でのcontrol群とexperiment群の比較
> pairwise.t.test(d4mh1$y, d4mh1$group, paired=FALSE, p.adjust.method="bonferroni")
```

Pairwise comparisons using t tests with pooled SD

data: d4mh1\$y and d4mh1\$group

control	
experiment	4.4e-12

P value adjustment method: bonferroni

```
> # mh2でのcontrol群とexperiment群の比較
> pairwise.t.test(d4mh2$y, d4mh2$group, paired=FALSE, p.adjust.method="bonferroni")
```

Pairwise comparisons using t tests with pooled SD

data: d4mh2\$y and d4mh2\$group

control	
experiment	0.072

P value adjustment method: bonferroni

```
> # control群のmh1とMh2の比較
> pairwise.t.test(d4control$y, d4control$x, paired=TRUE, p.adjust.method="bonferroni")

Pairwise comparisons using paired t tests

data: d4control$y and d4control$x

      mh1
mh2 0.63

P value adjustment method: bonferroni

> # experiment群のmh1とMh2の比較
> pairwise.t.test(d4experiment$y, d4experiment$x, paired=TRUE, p.adjust.method="bonferroni")

Pairwise comparisons using paired t tests

data: d4experiment$y and d4experiment$x

      mh1
mh2 5e-06

P value adjustment method: bonferroni
```

## > # 効果量

```
> #eta_squared, omega_squared関数を使う方法
> library(effectsize)
> eta_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Eta2 (partial)	95% CI
id	group	0.29	[0.15, 0.43]
id:group:x	x	0.13	[0.03, 0.26]
id:group:x	group:x	0.17	[0.06, 0.31]

```
> omega_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Omega2 (partial)	95% CI
id	group	0.28	[0.14, 0.42]
id:group:x	x	0.05	[0.00, 0.16]
id:group:x	group:x	0.08	[0.01, 0.20]

```
> eta_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type II)
```

Parameter	Eta2 (partial)	95% CI
group	0.29	[0.15, 0.43]
fac.b	0.13	[0.03, 0.26]
group:fac.b	0.17	[0.06, 0.31]

```
> omega_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type II)
```

Parameter	Omega2 (partial)	95% CI
group	0.28	[0.14, 0.42]
fac.b	0.05	[0.00, 0.16]
group:fac.b	0.08	[0.01, 0.20]

```
> #EtaSq関数を使う方法
> library(DescTools)
> EtaSq(result.aov, type=1)

      eta.sq eta.sq.part eta.sq.gen
group  0.17238648  0.2920008 0.19330377
x      0.04464965  0.1290165 0.05843784
group:x 0.06355980  0.1741425 0.08117845
>
>
```

## 2つの被験者内要因(2W)

### stackデータの作成

条件の組合せごとに各被験者の反応を縦にならべる。すべての条件の組合せについて縦に繋げる。

```
d2 <- d1[-1]
d3 <- stack(d2)
d3$x1 <- substr(d3$ind, 1, 2)
d3$x2 <- substr(d3$ind, 3, 4)
d3$id <- d1$id
d4 <- d3[, c("id", "values", "x1", "x2")]
colnames(d4) <- c("id", "y", "x1", "x2")
d4$id <- as.factor(d4$id)
d4$x1 <- as.factor(d4$x1)
d4$x2 <- as.factor(d4$x2)
```

### 独立変数をfactor型に変換

```
データフレーム名 : d4
被験者ID          : id
被験者内要因1     : x1
被験者内要因2     : x2
従属変数          : y
```

### 【重要！！】

被験者idも含め、独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをしておく。

factor型にしていないと、間違った分散分析をした結果を出力してしまう。

間違っているかどうかは、「独立変数の自由度Dfの値が“水準数-1”になっているか否か」などでわかる。

### 分散分析

#### aovを使う方法

```
result.aov <- aov(y ~ x1 * x2 + Error(id + id:x1 + id:x2 + id:x1:x2), data=d4)
summary(result.aov)
```

「球面性の仮定」が満たされることを前提としている。

### Anovaを使う方法

```
データフレーム名 : d1
被験者ID          : id
被験者内要因名1   : fac.a <- c(a1, a2)
被験者内要因名2   : fac.b <- c(b1, b2, b3)
全被験者内要因    : youin
従属変数          : y
```

### library(car)

```
fac.a <- factor(c("a1", "a1", "a1", "a2", "a2", "a2"))
fac.b <- factor(c("b1", "b2", "b3", "b1", "b2", "b3"))
youin <- data.frame(fac.a, fac.b)
mod.1 <- lm(cbind(a1b1, a1b2, a1b3, a2b1, a2b2, a2b3)~1, d1)
result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.a*fac.b)
summary(result.Anova, multivariate=FALSE)
```

「球面性の仮定」の検討をして、修正した結果も表示する。

あらかじめ car パッケージをインストールしておく必要がある。

multivariate=TRUE としておくと、多変量分散分析の結果も表示してくれる。

### 多重比較

#### 被験者内要因

```
pairwise.t.test(d4$y, d4$x1, p.adjust.method="bonferroni", paired=TRUE)
pairwise.t.test(d4$y, d4$x2, p.adjust.method="bonferroni", paired=TRUE)
```

p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"

## 効果量

effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```
library(effectsize)
```

```
eta_squared(分散分析オブジェクト, alternative="two.sided")
```

```
omega_squared(分散分析オブジェクト, alternative="two.sided")
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際, alternative="two.sided" を指定しておく, 下限と上限を推定する信頼区間となる。(指定しない場合, 上限は1.00となる。)

DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
```

```
EtaSq(aovオブジェクト, type=1)
```

各効果について,  $\eta^2$ 乗, 偏 $\eta^2$ 乗, 一般化 $\eta^2$ 乗を計算する。

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。

typeオプションでtype=1と指定する。(デフォルトはtype=2)

```
> setwd("d:¥¥¥")
> d1 <- read.table("2W平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d2 <- d1[-1]
```

```
> head(d2)
```

```
  cs cd ns nd
1 34 22 36 22
2 36 21 33 17
3 34 25 36 23
4 36 29 32 21
5 36 25 34 21
6 37 23 33 24
```

```
>
>
```

```
> #stackデータの作成
```

```
> d3 <- stack(d2)
```

```
> d3$x1 <- substr(d3$ind, 1, 1)
```

```
> d3$x2 <- substr(d3$ind, 2, 2)
```

```
> d3$id <- d1$id
```

```
> d4 <- d3[, c("id", "values", "x1", "x2")]
```

```
> colnames(d4) <- c("id", "y", "x1", "x2")
```

```
> # 独立変数をfactor型に変換
```

```
> d4$id <- factor(d4$id)
```

```
> d4$x1 <- factor(d4$x1)
```

```
> d4$x2 <- factor(d4$x2)
```

```
> head(d4)
```

```
  id y x1 x2
1  1 34  c  s
2  2 36  c  s
3  3 34  c  s
4  4 36  c  s
5  5 36  c  s
6  6 37  c  s
```

```
>
>
```

```
> # 各群の人数・平均値・標準偏差
```

1	id	cs	cd	ns	nd
2	1	34	22	36	22
3	2	36	21	33	17
4	3	34	25	36	23
5	4	36	29	32	21
6	5	36	25	34	21
7	6	37	23	33	24
8	7	34	22	33	21
9	8	34	25	38	22
10	9	34	24	29	21
11	10	33	25	34	22
12	11	38	22	35	22
13	12	37	22	32	18
14	13	35	26	29	25
15	14	31	24	32	20
16	15	35	24	33	19
17	16	36	23	31	16
18	17	34	26	36	18
19	18	37	26	33	21
20	19	37	26	35	28
21	20	37	23	33	22

```

> # 基本統計量
> library(psych)
>
> # セル単位
> describeBy(d4$y, list(d4$x1, d4$x2), mat=TRUE, digits=2)
      item group1 group2 vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
X11    1      c      d    1 138 23.86 2.70     24   23.90 2.97  15  30    15 -0.32    0.29 0.23
X12    2      n      d    1 138 21.83 2.58     22   21.91 2.97  14  28    14 -0.28    0.11 0.22
X13    3      c      s    1 138 35.06 2.39     35   35.09 1.48  28  44    16  0.01    1.12 0.20
X14    4      n      s    1 138 33.08 2.56     33   33.02 2.97  28  40    12  0.21   -0.48 0.22
>

> # 変数単位
> describeBy(d4$y, list(d4$x1), mat=TRUE, digits=2)
      item group1 vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
X11    1      c    1 276 29.46 6.16     29   29.54 8.9  15  44    29 -0.07   -1.32 0.37
X12    2      n    1 276 27.46 6.19     28   27.47 8.9  14  40    26 -0.01   -1.32 0.37

> describeBy(d4$y, list(d4$x2), mat=TRUE, digits=2)
      item group1 vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
X11    1      d    1 276 22.84 2.83     23   22.89 2.97  14  30    16 -0.21    0.10 0.17
X12    2      s    1 276 34.07 2.67     34   34.09 2.97  28  44    16  0.03    0.01 0.16
>

> # 全体
> describe(d4$y)
      vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
X1      1 552 28.46 6.25     29   28.51 8.9  14  44    30 -0.04   -1.27 0.27
>

> # 分散分析

> # aovを使う方法
> result.aov <- aov(y ~ x1 * x2 + Error(id + id:x1 + id:x2 + id:x1:x2), data=d4)
> summary(result.aov)

Error: id
      Df Sum Sq Mean Sq F value Pr(>F)
Residuals 137    1461    10.66

Error: id:x1
      Df Sum Sq Mean Sq F value Pr(>F)
x1      1    552    552.0   98.21 <2e-16 ***
Residuals 137    770     5.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:x2
      Df Sum Sq Mean Sq F value Pr(>F)
x2      1 17387  17387   3184 <2e-16 ***
Residuals 137    748     5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:x1:x2
      Df Sum Sq Mean Sq F value Pr(>F)
x1:x2   1    0.1    0.065   0.014 0.905
Residuals 137 618.9    4.518

> # Anovaを使う方法
> library(car)
> fac.a <- factor(c("c", "c", "n", "n"))
> fac.b <- factor(c("s", "d", "s", "d"))
> youin <- data.frame(fac.a, fac.b)
> mod.l <- lm(cbind( cs, cd, ns, nd)~1, d1)
> result.Anova <- Anova(mod.l, idata=youin, idesign=~fac.a*fac.b)

```



Note: model has only an intercept; equivalent type-III tests substituted.  
 > summary(result.Anova, multivariate=FALSE)

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

	SS	num Df	Error SS	den Df	F	Pr(>F)
(Intercept)	446995	1	1460.96	137	41916.5937	<2e-16 ***
fac.a	552	1	770.00	137	98.2130	<2e-16 ***
fac.b	17387	1	748.04	137	3184.3565	<2e-16 ***
fac.a:fac.b	0	1	618.93	137	0.0144	0.9045

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

> #ボンフェロニ法を用いた多重比較

> d4x1\_c <- subset(d4, x1=="c")

> d4x1\_n <- subset(d4, x1=="n")

> d4x2\_d <- subset(d4, x2=="d")

> d4x2\_s <- subset(d4, x2=="s")

>

> # x1=cでのx2=d群とx2=s群の比較

> pairwise.t.test(d4x1\_c\$y, d4x1\_c\$x2, paired=TRUE, p.adjust.method="bonferroni")

Pairwise comparisons using paired t tests

data: d4x1\_c\$y and d4x1\_c\$x2

d  
s <2e-16

P value adjustment method: bonferroni

>

> # x1=nでのx2=d群とx2=s群の比較

> pairwise.t.test(d4x1\_n\$y, d4x1\_n\$x2, paired=TRUE, p.adjust.method="bonferroni")

Pairwise comparisons using paired t tests

data: d4x1\_n\$y and d4x1\_n\$x2

d  
s <2e-16

P value adjustment method: bonferroni

>

> # x2=dでのx1=c群とx1=n群の比較

> pairwise.t.test(d4x2\_d\$y, d4x2\_d\$x1, paired=TRUE, p.adjust.method="bonferroni")

Pairwise comparisons using paired t tests

data: d4x2\_d\$y and d4x2\_d\$x1

c  
n 2.7e-11

P value adjustment method: bonferroni

>

> # x2=sでのx1=c群とx1=n群の比較

> pairwise.t.test(d4x2\_s\$y, d4x2\_s\$x1, paired=TRUE, p.adjust.method="bonferroni")

Pairwise comparisons using paired t tests

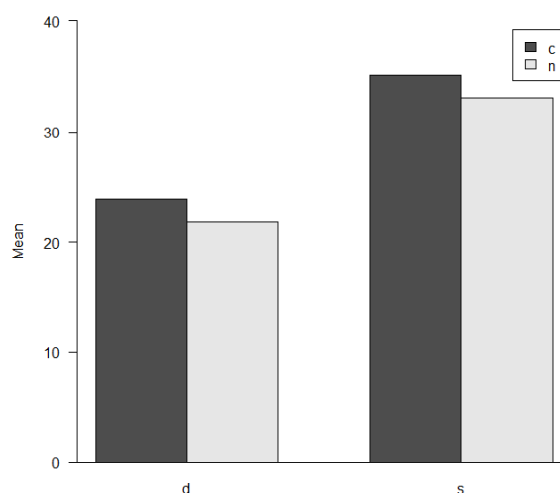
data: d4x2\_s\$y and d4x2\_s\$x1

```
c
n 6.6e-12
```

```
P value adjustment method: bonferroni
>
```

```
> #平均値を比較する棒グラフの作成
```

```
> m4t <- tapply(d4$y, list(d4$x1, d4$x2), mean)
> barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 40), las=1, ylab="Mean")
> segments(0, 0, 10, 0)
```



```
> # 効果量
```

```
> #eta_squared, omega_squared関数を使う方法
> library(effectsiz)
```

```
> eta_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Eta2 (partial)	95% CI
id:x1	x1	0.42	[0.30, 0.52]
id:x2	x2	0.96	[0.95, 0.97]
id:x1:x2	x1:x2	1.05e-04	[0.00, 0.02]

```
> omega_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Omega2 (partial)	95% CI
id:x1	x1	0.20	[0.09, 0.31]
id:x2	x2	0.89	[0.85, 0.91]
id:x1:x2	x1:x2	0.00	[0.00, 0.00]

```
> eta_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type III)
```

Parameter	Eta2 (partial)	95% CI
fac.a	0.42	[0.30, 0.52]
fac.b	0.96	[0.95, 0.97]
fac.a:fac.b	1.05e-04	[0.00, 0.02]

```
> omega_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type III)
```

Parameter	Omega2 (partial)	95% CI
fac.a	0.20	[0.09, 0.31]
fac.b	0.89	[0.85, 0.91]
fac.a:fac.b	0.00	[0.00, 0.00]

```
> #EtaSq関数を使う方法
```

```
> library(DescTools)
```

```
> EtaSq(result.aov, type=1)
```

```
      eta.sq  eta.sq.part  eta.sq.gen
x1  2.563036e-02  0.4175491679  1.330144e-01
x2  8.073083e-01  0.9587517931  8.285468e-01
x1:x2 3.028162e-06  0.0001053593  1.812605e-05
```

## 3つの被験者間要因(3B)

```

データフレーム名： d1
被験者ID：          id
被験者間要因1：     A
被験者間要因2：     B
被験者間要因3：     C
従属変数：          Y

```

## 独立変数の型変換

```

d1$A <- as.factor(d1$A)
d1$B <- as.factor(d1$B)
d1$C <- as.factor(d1$C)

```

## 【重要！！】

独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。  
factor型にしていないと、間違った分散分析をした結果を出力してしまう。  
間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

## 分散分析

## aovを使う方法

```

lmオブジェクト名 <- lm(Y ~ A*B*C, data=d1)
aovオブジェクト名 <- aov(lmオブジェクト名)
summary(aovオブジェクト名)

```

## anovaを使う方法

```

lmオブジェクト名 <- lm(Y ~ A*B*C, data=d1)
anovaオブジェクト名 <- anova(lmオブジェクト名)
anovaオブジェクト名

```

## Anovaを使う方法

```

library(car)
lmオブジェクト名 <- lm(Y~A*B*C, data=d1)
Anova オブジェクト名 <- Anova(lmオブジェクト名)
Anova オブジェクト名

```

あらかじめcarパッケージをインストールしておく必要がある。

被験者間要因が2つあるので、線形モデル(lm)の独立変数を「独立変数1 \* 独立変数2」とする。

## 多重比較 テューキー法

```
TukeyHSD(aov(Y ~ A*B*C, data=d1) または TukeyHSD(aovの出力結果))
```

## 効果量

各効果について、 $\eta^2$ 乗、偏 $\eta^2$ 乗、 $\omega^2$ 乗の全部または一部を計算する。

## effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```

library(effectsize)
eta_squared(分散分析オブジェクト, alternative="two.sided")
omega_squared(分散分析オブジェクト, alternative="two.sided")

```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, anova, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際、alternative="two.sided" を指定しておく、下限と上限を推定する信頼区間となる。(指定しない場合、上限は1.00となる。)

## DescToolsパッケージのEtaSq関数を使う方法

```

library(DescTools)
EtaSq(aovオブジェクト)

```

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをEtaSqに入れる。

typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

## lsrパッケージのetaSquared関数を使う方法

library(lsr)

etaSquared(aovオブジェクト)

あらかじめlsrパッケージをインストールしておく必要がある。  
 aov関数で分散分析を実行した結果オブジェクトをetaSquaredに入れる。  
 typeオプションで平方和のタイプを選択できる。デフォルトはタイプ2。

```
> setwd("d:¥¥")
> d1 <- read.table("3B平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id  A  B  C  Y
1  1 a1 b1 c1  8
2  2 a1 b1 c1  6
3  3 a1 b1 c1  8
4  4 a1 b1 c1  9
5  5 a1 b1 c1  8
6  6 a1 b1 c2  9
>
>
> # 独立変数をfactor型に変換
> d1$id <- factor(d1$id)
> d1$A <- factor(d1$A)
> d1$B <- factor(d1$B)
> d1$C <- factor(d1$C)
> head(d1)
  id  A  B  C  Y
1  1 a1 b1 c1  8
2  2 a1 b1 c1  6
3  3 a1 b1 c1  8
4  4 a1 b1 c1  9
5  5 a1 b1 c1  8
6  6 a1 b1 c2  9
>
>
>
> # 各群の人数・平均値・標準偏差
> library(psych)
> (tABC <- describeBy(d1$Y, list(d1$A, d1$B, d1$C), mat=TRUE, digits=2))
```

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se	
X11	1	a1	b1	c1	1	5	7.8	1.10	8	7.8	0.00	6	9	3	-0.62	-1.25	0.49
X12	2	a2	b1	c1	1	5	2.8	0.84	3	2.8	1.48	2	4	2	0.25	-1.82	0.37
X13	3	a1	b2	c1	1	5	9.4	2.07	10	9.4	1.48	6	11	5	-0.69	-1.41	0.93
X14	4	a2	b2	c1	1	5	3.8	2.49	3	3.8	1.48	2	8	6	0.80	-1.27	1.11
X15	5	a1	b3	c1	1	5	7.2	0.84	7	7.2	1.48	6	8	2	-0.25	-1.82	0.37
X16	6	a2	b3	c1	1	5	13.0	1.87	12	13.0	1.48	11	15	4	0.18	-2.18	0.84
X17	7	a1	b1	c2	1	5	7.2	1.30	7	7.2	1.48	6	9	3	0.26	-1.96	0.58
X18	8	a2	b1	c2	1	5	4.4	1.52	4	4.4	0.00	3	7	4	0.84	-1.12	0.68
X19	9	a1	b2	c2	1	5	6.0	2.55	6	6.0	1.48	3	10	7	0.43	-1.40	1.14
X110	10	a2	b2	c2	1	5	7.6	2.88	6	7.6	1.48	5	12	7	0.52	-1.73	1.29
X111	11	a1	b3	c2	1	5	7.0	1.00	7	7.0	1.48	6	8	2	0.00	-2.20	0.45
X112	12	a2	b3	c2	1	5	13.4	1.67	13	13.4	2.97	11	15	4	-0.25	-1.82	0.75

```
>
> (tAB <- describeBy(d1$Y, list(d1$A, d1$B), mat=TRUE, digits=2))
```

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	a1	b1	1	10	7.5	1.18	8.0	7.50	1.48	6	9	3	-0.18	-1.67	0.37
X12	2	a2	b1	1	10	3.6	1.43	3.5	3.38	0.74	2	7	5	1.05	0.52	0.45
X13	3	a1	b2	1	10	7.7	2.83	7.5	7.88	3.71	3	11	8	-0.18	-1.66	0.90
X14	4	a2	b2	1	10	5.7	3.23	5.5	5.38	3.71	2	12	10	0.51	-1.03	1.02
X15	5	a1	b3	1	10	7.1	0.88	7.0	7.12	1.48	6	8	2	-0.16	-1.81	0.28
X16	6	a2	b3	1	10	13.2	1.69	13.0	13.25	2.97	11	15	4	-0.03	-1.85	0.53

```

>
> (tAC <- describeBy(d1$Y, list(d1$A, d1$C), mat=TRUE, digits=2))
      item group1 group2 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11     1      a1      c1    1 15 8.13 1.64      8   8.08 1.48   6 11    5 0.34   -1.04 0.42
X12     2      a2      c1    1 15 6.53 5.05      4   6.23 2.97   2 15   13 0.57   -1.50 1.31
X13     3      a1      c2    1 15 6.73 1.71      6   6.77 1.48   3 10    7 -0.09   -0.25 0.44
X14     4      a2      c2    1 15 8.47 4.32      7   8.38 4.45   3 15   12 0.25   -1.67 1.12
>
> (tBC <- describeBy(d1$Y, list(d1$B, d1$C), mat=TRUE, digits=2))
      item group1 group2 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11     1      b1      c1    1 10  5.3 2.79      5.0   5.25 4.45   2  9    7 0.05   -1.94 0.88
X12     2      b2      c1    1 10  6.6 3.66      7.0   6.62 5.19   2 11    9 -0.07   -1.87 1.16
X13     3      b3      c1    1 10 10.1 3.35      9.5  10.00 3.71   6 15    9 0.27   -1.66 1.06
X14     4      b1      c2    1 10  5.8 1.99      6.0   5.75 2.97   3  9    6 0.09   -1.56 0.63
X15     5      b2      c2    1 10  6.8 2.70      6.0   6.62 1.48   3 12    9 0.59   -0.95 0.85
X16     6      b3      c2    1 10 10.2 3.61      9.5  10.12 5.19   6 15    9 0.13   -1.85 1.14
>
> (tA <- describeBy(d1$Y, list(d1$A), mat=TRUE, digits=2))
      item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11     1      a1    1 30 7.43 1.79      7.5   7.38 2.22   3 11    8 0.05   -0.14 0.33
X12     2      a2    1 30 7.50 4.73      6.0   7.25 5.19   2 15   13 0.35   -1.52 0.86
>
> (tB <- describeBy(d1$Y, list(d1$B), mat=TRUE, digits=2))
      item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11     1      b1    1 20  5.55 2.37      6.0   5.56 2.97   2  9    7 -0.03   -1.57 0.53
X12     2      b2    1 20  6.70 3.13      6.0   6.69 4.45   2 12   10 0.11   -1.34 0.70
X13     3      b3    1 20 10.15 3.39      9.5  10.06 3.71   6 15    9 0.22   -1.62 0.76
>
> (tC <- describeBy(d1$Y, list(d1$C), mat=TRUE, digits=2))
      item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11     1      c1    1 30 7.33 3.78      8.0   7.17 4.45   2 15   13 0.22   -0.88 0.69
X12     2      c2    1 30 7.60 3.35      6.5   7.29 2.22   3 15   12 0.77   -0.40 0.61
>
> (tall <- describe(d1$Y))
      vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X1       1 60 7.47 3.54      7   7.25 2.97   2 15   13 0.44   -0.55 0.46
>
>
>
> #分散分析
> # aovを使う方法
> mod.1 <- lm(Y~A*B*C, data=d1)
> result.aov <- aov(mod.1)
> summary(result.aov)
      Df Sum Sq Mean Sq F value    Pr(>F)
A         1    0.07    0.07    0.020  0.88700
B         2 229.23 114.62   35.087 4.07e-10 ***
C         1    1.07    1.07    0.327  0.57038
A:B       2 282.03 141.02   43.168 1.88e-11 ***
A:C       1  41.67   41.67   12.755  0.00082 ***
B:C       2    0.43    0.22    0.066  0.93591
A:B:C     2  29.63   14.82    4.536  0.01569 *
Residuals 48 156.80    3.27
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>
> # anovaを使う方法
> mod.1 <- lm(Y~A*B*C, data=d1)
> result.anova <- anova(mod.1)
> result.anova
Analysis of Variance Table

```

```

Response: Y
      Df Sum Sq Mean Sq F value    Pr(>F)
A       1   0.067    0.067   0.0204 0.8870010
B       2 229.233   114.617  35.0867 4.067e-10 ***
C       1   1.067    1.067   0.3265 0.5703753
A:B     2 282.033   141.017  43.1684 1.875e-11 ***
A:C     1  41.667   41.667  12.7551 0.0008196 ***
B:C     2   0.433    0.217   0.0663 0.9359109
A:B:C   2  29.633   14.817   4.5357 0.0156942 *
Residuals 48 156.800    3.267
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

```

> # Anovaを使う方法
> library(car)
> mod.1 <- lm(Y~A*B*C, data=d1)
> result.Anova <- Anova(mod.1)
> result.Anova
Anova Table (Type II tests)

```

```

Response: Y
      Sum Sq Df F value    Pr(>F)
A       0.067  1  0.0204 0.8870010
B     229.233  2 35.0867 4.067e-10 ***
C       1.067  1  0.3265 0.5703753
A:B    282.033  2 43.1684 1.875e-11 ***
A:C     41.667  1 12.7551 0.0008196 ***
B:C      0.433  2  0.0663 0.9359109
A:B:C   29.633  2  4.5357 0.0156942 *
Residuals 156.800 48
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>
>

```

```

> #多重比較
> #テューキーの方法
> (mc.tukey <- TukeyHSD(result.aov))
Tukey multiple comparisons of means
 95% family-wise confidence level

```

```
Fit: aov(formula = mod.1)
```

```
$A
      diff      lwr      upr    p adj
a2-a1 0.0666667 -0.8716296 1.004963 0.887001

```

```
$B
      diff      lwr      upr    p adj
b2-b1 1.15 -0.2322808 2.532281 0.1202186
b3-b1 4.60  3.2177192 5.982281 0.0000000
b3-b2 3.45  2.0677192 4.832281 0.0000007

```

```
$C
      diff      lwr      upr    p adj
c2-c1 0.2666667 -0.6716296 1.204963 0.5703753

```

```
$`A:B`
      diff      lwr      upr    p adj
a2:b1-a1:b1 -3.9 -6.2989208 -1.5010792 0.0002027
a1:b2-a1:b1  0.2 -2.1989208  2.5989208 0.9998650
a2:b2-a1:b1 -1.8 -4.1989208  0.5989208 0.2447676
a1:b3-a1:b1 -0.4 -2.7989208  1.9989208 0.9961341
a2:b3-a1:b1  5.7  3.3010792  8.0989208 0.0000001
a1:b2-a2:b1  4.1  1.7010792  6.4989208 0.0000886
a2:b2-a2:b1  2.1 -0.2989208  4.4989208 0.1174879
a1:b3-a2:b1  3.5  1.1010792  5.8989208 0.0010129

```

```

a2:b3-a2:b1 9.6 7.2010792 11.9989208 0.0000000
a2:b2-a1:b2 -2.0 -4.3989208 0.3989208 0.1522895
a1:b3-a1:b2 -0.6 -2.9989208 1.7989208 0.9754862
a2:b3-a1:b2 5.5 3.1010792 7.8989208 0.0000002
a1:b3-a2:b2 1.4 -0.9989208 3.7989208 0.5181172
a2:b3-a2:b2 7.5 5.1010792 9.8989208 0.0000000
a2:b3-a1:b3 6.1 3.7010792 8.4989208 0.0000000

```

\$`A:C`

```

              diff      lwr      upr      p adj
a2:c1-a1:c1 -1.6000000 -3.3564161 0.1564161 0.0859930
a1:c2-a1:c1 -1.4000000 -3.1564161 0.3564161 0.1609853
a2:c2-a1:c1 0.3333333 -1.4230828 2.0897495 0.9574880
a1:c2-a2:c1 0.2000000 -1.5564161 1.9564161 0.9902117
a2:c2-a2:c1 1.9333333 0.1769172 3.6897495 0.0257796
a2:c2-a1:c2 1.7333333 -0.0230828 3.4897495 0.0543105

```

\$`B:C`

```

              diff      lwr      upr      p adj
b2:c1-b1:c1 1.3 -1.0989208 3.6989208 0.5972275
b3:c1-b1:c1 4.8 2.4010792 7.1989208 0.0000045
b1:c2-b1:c1 0.5 -1.8989208 2.8989208 0.9891291
b2:c2-b1:c1 1.5 -0.8989208 3.8989208 0.4411593
b3:c2-b1:c1 4.9 2.5010792 7.2989208 0.0000029
b3:c1-b2:c1 3.5 1.1010792 5.8989208 0.0010129
b1:c2-b2:c1 -0.8 -3.1989208 1.5989208 0.9189969
b2:c2-b2:c1 0.2 -2.1989208 2.5989208 0.9998650
b3:c2-b2:c1 3.6 1.2010792 5.9989208 0.0006821
b1:c2-b3:c1 -4.3 -6.6989208 -1.9010792 0.0000382
b2:c2-b3:c1 -3.3 -5.6989208 -0.9010792 0.0021977
b3:c2-b3:c1 0.1 -2.2989208 2.4989208 0.9999957
b2:c2-b1:c2 1.0 -1.3989208 3.3989208 0.8164137
b3:c2-b1:c2 4.4 2.0010792 6.7989208 0.0000250
b3:c2-b2:c2 3.4 1.0010792 5.7989208 0.0014962

```

\$`A:B:C`

```

              diff      lwr      upr      p adj
a2:b1:c1-a1:b1:c1 -5.000000e+00 -8.92508183 -1.07491817 0.0034084
a1:b2:c1-a1:b1:c1 1.600000e+00 -2.32508183 5.52508183 0.9579810
a2:b2:c1-a1:b1:c1 -4.000000e+00 -7.92508183 -0.07491817 0.0422093
a1:b3:c1-a1:b1:c1 -6.000000e+00 -4.52508183 3.32508183 0.9999940
a2:b3:c1-a1:b1:c1 5.200000e+00 1.27491817 9.12508183 0.0019706
a1:b1:c2-a1:b1:c1 -6.000000e-01 -4.52508183 3.32508183 0.9999940
a2:b1:c2-a1:b1:c1 -3.400000e+00 -7.32508183 0.52508183 0.1482991
a1:b2:c2-a1:b1:c1 -1.800000e+00 -5.72508183 2.12508183 0.9097357
a2:b2:c2-a1:b1:c1 -2.000000e-01 -4.12508183 3.72508183 1.0000000
a1:b3:c2-a1:b1:c1 -8.000000e-01 -4.72508183 3.12508183 0.9998906
a2:b3:c2-a1:b1:c1 5.600000e+00 1.67491817 9.52508183 0.0006381
a1:b2:c1-a2:b1:c1 6.600000e+00 2.67491817 10.52508183 0.0000334
a2:b2:c1-a2:b1:c1 1.000000e+00 -2.92508183 4.92508183 0.9990898
a1:b3:c1-a2:b1:c1 4.400000e+00 0.47491817 8.32508183 0.0162497
a2:b3:c1-a2:b1:c1 1.020000e+01 6.27491817 14.12508183 0.0000000
a1:b1:c2-a2:b1:c1 4.400000e+00 0.47491817 8.32508183 0.0162497
a2:b1:c2-a2:b1:c1 1.600000e+00 -2.32508183 5.52508183 0.9579810
a1:b2:c2-a2:b1:c1 3.200000e+00 -0.72508183 7.12508183 0.2128271
a2:b2:c2-a2:b1:c1 4.800000e+00 0.87491817 8.72508183 0.0058219
a1:b3:c2-a2:b1:c1 4.200000e+00 0.27491817 8.12508183 0.0264537
a2:b3:c2-a2:b1:c1 1.060000e+01 6.67491817 14.52508183 0.0000000
a2:b2:c1-a1:b2:c1 -5.600000e+00 -9.52508183 -1.67491817 0.0006381
a1:b3:c1-a1:b2:c1 -2.200000e+00 -6.12508183 1.72508183 0.7384436
a2:b3:c1-a1:b2:c1 3.600000e+00 -0.32508183 7.52508183 0.1002066
a1:b1:c2-a1:b2:c1 -2.200000e+00 -6.12508183 1.72508183 0.7384436
a2:b1:c2-a1:b2:c1 -5.000000e+00 -8.92508183 -1.07491817 0.0034084
a1:b2:c2-a1:b2:c1 -3.400000e+00 -7.32508183 0.52508183 0.1482991
a2:b2:c2-a1:b2:c1 -1.800000e+00 -5.72508183 2.12508183 0.9097357
a1:b3:c2-a1:b2:c1 -2.400000e+00 -6.32508183 1.52508183 0.6254360
a2:b3:c2-a1:b2:c1 4.000000e+00 0.07491817 7.92508183 0.0422093
a1:b3:c1-a2:b2:c1 3.400000e+00 -0.52508183 7.32508183 0.1482991
a2:b3:c1-a2:b2:c1 9.200000e+00 5.27491817 13.12508183 0.0000000
a1:b1:c2-a2:b2:c1 3.400000e+00 -0.52508183 7.32508183 0.1482991
a2:b1:c2-a2:b2:c1 6.000000e-01 -3.32508183 4.52508183 0.9999940
a1:b2:c2-a2:b2:c1 2.200000e+00 -1.72508183 6.12508183 0.7384436

```

1	id	A	B	C	Y
2	1	a1	b1	c1	8
3	2	a1	b1	c1	6
4	3	a1	b1	c1	8
5	4	a1	b1	c1	9
6	5	a1	b1	c1	8
7	6	a1	b1	c2	9
8	7	a1	b1	c2	6
9	8	a1	b1	c2	8
10	9	a1	b1	c2	6
11	10	a1	b1	c2	7
12	11	a1	b2	c1	6
13	12	a1	b2	c1	9
14	13	a1	b2	c1	10
15	14	a1	b2	c1	11
16	15	a1	b2	c1	11
17	16	a1	b2	c2	10
18	17	a1	b2	c2	6
19	18	a1	b2	c2	5
20	19	a1	b2	c2	3
21	20	a1	b2	c2	6
22	21	a1	b3	c1	7
23	22	a1	b3	c1	8
24	23	a1	b3	c1	7
25	24	a1	b3	c1	8
26	25	a1	b3	c1	6
27	26	a1	b3	c2	7
28	27	a1	b3	c2	6
29	28	a1	b3	c2	8
30	29	a1	b3	c2	6
31	30	a1	b3	c2	8
32	31	a2	b1	c1	2
33	32	a2	b1	c1	4
34	33	a2	b1	c1	3
35	34	a2	b1	c1	2
36	35	a2	b1	c1	3
37	36	a2	b1	c2	4
38	37	a2	b1	c2	4
39	38	a2	b1	c2	3
40	39	a2	b1	c2	7
41	40	a2	b1	c2	4
42	41	a2	b2	c1	4
43	42	a2	b2	c1	3
44	43	a2	b2	c1	2
45	44	a2	b2	c1	2
46	45	a2	b2	c1	8
47	46	a2	b2	c2	6
48	47	a2	b2	c2	5
49	48	a2	b2	c2	6
50	49	a2	b2	c2	9
51	50	a2	b2	c2	12
52	51	a2	b3	c1	11
53	52	a2	b3	c1	15
54	53	a2	b3	c1	12
55	54	a2	b3	c1	12
56	55	a2	b3	c1	15
57	56	a2	b3	c2	15
58	57	a2	b3	c2	15
59	58	a2	b3	c2	13
60	59	a2	b3	c2	13
61	60	a2	b3	c2	11
62					

```

a2:b2:c2-a2:b2:c1 3.800000e+00 -0.12508183 7.72508183 0.0658543
a1:b3:c2-a2:b2:c1 3.200000e+00 -0.72508183 7.12508183 0.2128271
a2:b3:c2-a2:b2:c1 9.600000e+00 5.67491817 13.52508183 0.0000000
a2:b3:c1-a1:b3:c1 5.800000e+00 1.87491817 9.72508183 0.0003583
a1:b1:c2-a1:b3:c1 8.881784e-16 -3.92508183 3.92508183 1.0000000
a2:b1:c2-a1:b3:c1 -2.800000e+00 -6.72508183 1.12508183 0.3948439
a1:b2:c2-a1:b3:c1 -1.200000e+00 -5.12508183 2.72508183 0.9954470
a2:b2:c2-a1:b3:c1 4.000000e-01 -3.52508183 4.32508183 0.9999999
a1:b3:c2-a1:b3:c1 -2.000000e-01 -4.12508183 3.72508183 1.0000000
a2:b3:c2-a1:b3:c1 6.200000e+00 2.27491817 10.12508183 0.0001106
a1:b1:c2-a2:b3:c1 -5.800000e+00 -9.72508183 -1.87491817 0.0003583
a2:b1:c2-a2:b3:c1 -8.600000e+00 -12.52508183 -4.67491817 0.0000001
a1:b2:c2-a2:b3:c1 -7.000000e+00 -10.92508183 -3.07491817 0.0000099
a2:b2:c2-a2:b3:c1 -5.400000e+00 -9.32508183 -1.47491817 0.0011268
a1:b3:c2-a2:b3:c1 -6.000000e+00 -9.92508183 -2.07491817 0.0001997
a2:b3:c2-a2:b3:c1 4.000000e-01 -3.52508183 4.32508183 0.9999999
a2:b1:c2-a1:b1:c2 -2.800000e+00 -6.72508183 1.12508183 0.3948439
a1:b2:c2-a1:b1:c2 -1.200000e+00 -5.12508183 2.72508183 0.9954470
a2:b2:c2-a1:b1:c2 4.000000e-01 -3.52508183 4.32508183 0.9999999
a1:b3:c2-a1:b1:c2 -2.000000e-01 -4.12508183 3.72508183 1.0000000
a2:b3:c2-a1:b1:c2 6.200000e+00 2.27491817 10.12508183 0.0001106
a1:b2:c2-a2:b1:c2 1.600000e+00 -2.32508183 5.52508183 0.9579810
a2:b2:c2-a2:b1:c2 3.200000e+00 -0.72508183 7.12508183 0.2128271
a1:b3:c2-a2:b1:c2 2.600000e+00 -1.32508183 6.52508183 0.5074523
a2:b3:c2-a2:b1:c2 9.000000e+00 5.07491817 12.92508183 0.0000000
a2:b2:c2-a1:b2:c2 1.600000e+00 -2.32508183 5.52508183 0.9579810
a1:b3:c2-a1:b2:c2 1.000000e+00 -2.92508183 4.92508183 0.9990898
a2:b3:c2-a1:b2:c2 7.400000e+00 3.47491817 11.32508183 0.0000029
a1:b3:c2-a2:b2:c2 -6.000000e-01 -4.52508183 3.32508183 0.9999940
a2:b3:c2-a2:b2:c2 5.800000e+00 1.87491817 9.72508183 0.0003583
a2:b3:c2-a1:b3:c2 6.400000e+00 2.47491817 10.32508183 0.0000609

```

# > #シェッフェの方法

```
> library(agricolae)
```

```
> (mc.scheffe <- scheffe.test(result.aov, "A"))
```

```
$statistics
```

MSerror	Df	F	Mean	CV	Scheffe	CriticalDifference
3.266667	48	4.042652	7.466667	24.20615	2.010635	0.9382962

```
$parameters
```

test name	t	ntr	alpha
Scheffe	A	2	0.05

```
$means
```

	Y	std	r	Min	Max	Q25	Q50	Q75
a1	7.433333	1.794308	30	3	11	6.00	7.5	8
a2	7.500000	4.725208	30	2	15	3.25	6.0	12

```
$comparison
```

```
NULL
```

```
$groups
```

Y	groups
a2 7.500000	a
a1 7.433333	a

```
attr("class")
```

```
[1] "group"
```

```
> (mc.scheffe <- scheffe.test(result.aov, "B"))
```

```
$statistics
```

MSerror	Df	F	Mean	CV	Scheffe	CriticalDifference
3.266667	48	3.190727	7.466667	24.20615	2.526154	1.443817

```
$parameters
```

test name	t	ntr	alpha
Scheffe	B	3	0.05



```

$means
      Y      std  r Min Max  Q25 Q50  Q75
b1  5.55 2.372540 20   2   9 3.75 6.0  8.00
b2  6.70 3.130495 20   2  12 4.75 6.0  9.25
b3 10.15 3.391553 20   6  15 7.00 9.5 13.00

$comparison
NULL

$groups
      Y groups
b3 10.15      a
b2  6.70      b
b1  5.55      b

attr(,"class")
[1] "group"

> (mc.scheffe <- scheffe.test(result.aov, "C"))
$statistics
      MSerror Df      F      Mean      CV Scheffe CriticalDifference
      3.266667 48 4.042652 7.466667 24.20615 2.010635      0.9382962

$parameters
      test name.t ntr alpha
Scheffe      C    2  0.05

$means
      Y      std  r Min Max  Q25 Q50  Q75
c1 7.333333 3.781382 30   2  15   4 8.0 9.75
c2 7.600000 3.348700 30   3  15   6 6.5 9.00

$comparison
NULL

$groups
      Y groups
c2 7.600000      a
c1 7.333333      a

attr(,"class")
[1] "group"

> # group=FALSE を指定したシェッフェの方法
> (mc.scheffe <- scheffe.test(result.aov, "A", group=FALSE))
$statistics
      MSerror Df      F      Mean      CV Scheffe CriticalDifference
      3.266667 48 4.042652 7.466667 24.20615 2.010635      0.9382962

$parameters
      test name.t ntr alpha
Scheffe      A    2  0.05

$means
      Y      std  r Min Max  Q25 Q50  Q75
a1 7.433333 1.794308 30   3  11 6.00 7.5   8
a2 7.500000 4.725208 30   2  15 3.25 6.0  12

$comparison
      Difference pvalue sig      LCL      UCL
a1 - a2 -0.06666667 0.887 -0.7283853 0.595052

$groups
NULL

attr(,"class")
[1] "group"

```

```
> (mc.scheffe <- scheffe.test(result.aov, "B", group=FALSE))
$statistics
  MSerror Df      F      Mean      CV Scheffe CriticalDifference
3.266667 48 3.190727 7.466667 24.20615 2.526154      1.443817

$parameters
  test name.t ntr alpha
Scheffe      B   3  0.05

$means
      Y      std  r Min Max  Q25 Q50  Q75
b1  5.55 2.372540 20   2   9 3.75 6.0  8.00
b2  6.70 3.130495 20   2  12 4.75 6.0  9.25
b3 10.15 3.391553 20   6  15 7.00 9.5 13.00

$comparison
      Difference pvalue sig      LCL      UCL
b1 - b2      -1.15 0.1432      -2.434687  0.134687
b1 - b3      -4.60 0.0000 *** -5.884687 -3.315313
b2 - b3      -3.45 0.0000 *** -4.734687 -2.165313

$groups
NULL

attr(,"class")
[1] "group"
```

```
> (mc.scheffe <- scheffe.test(result.aov, "C", group=FALSE))
$statistics
  MSerror Df      F      Mean      CV Scheffe CriticalDifference
3.266667 48 4.042652 7.466667 24.20615 2.010635      0.9382962

$parameters
  test name.t ntr alpha
Scheffe      C   2  0.05

$means
      Y      std  r Min Max  Q25 Q50  Q75
c1  7.333333 3.781382 30   2  15  4 8.0 9.75
c2  7.600000 3.348700 30   3  15  6 6.5 9.00

$comparison
      Difference pvalue sig      LCL      UCL
c1 - c2 -0.2666667 0.5704      -0.9283853  0.395052

$groups
NULL

attr(,"class")
[1] "group"
>
>
>
>
```

> # 効果量

> #eta\_squared, omega\_squared関数を使う方法  
> library(effectsiz)

> eta\_squared(result.aov, alternative="two.sided")  
# Effect Size for ANOVA (Type I)

Parameter	Eta2 (partial)	95% CI
A	4.25e-04	[0.00, 0.06]
B	0.59	[0.40, 0.71]
C	6.76e-03	[0.00, 0.12]
A:B	0.64	[0.47, 0.75]
A:C	0.21	[0.04, 0.40]
B:C	2.76e-03	[0.00, 0.04]
A:B:C	0.16	[0.01, 0.34]

> omega\_squared(result.aov, alternative="two.sided")  
# Effect Size for ANOVA (Type I)

Parameter	Omega2 (partial)	95% CI
A	0.00	[0.00, 0.00]
B	0.53	[0.33, 0.66]
C	0.00	[0.00, 0.00]
A:B	0.58	[0.39, 0.70]
A:C	0.16	[0.02, 0.35]
B:C	0.00	[0.00, 0.00]
A:B:C	0.11	[0.00, 0.27]

> eta\_squared(result.anova, alternative="two.sided")  
# Effect Size for ANOVA (Type I)

Parameter	Eta2 (partial)	95% CI
A	4.25e-04	[0.00, 0.06]
B	0.59	[0.40, 0.71]
C	6.76e-03	[0.00, 0.12]
A:B	0.64	[0.47, 0.75]
A:C	0.21	[0.04, 0.40]
B:C	2.76e-03	[0.00, 0.04]
A:B:C	0.16	[0.01, 0.34]

> omega\_squared(result.anova, alternative="two.sided")  
# Effect Size for ANOVA (Type I)

Parameter	Omega2 (partial)	95% CI
A	0.00	[0.00, 0.00]
B	0.53	[0.33, 0.66]
C	0.00	[0.00, 0.00]
A:B	0.58	[0.39, 0.70]
A:C	0.16	[0.02, 0.35]
B:C	0.00	[0.00, 0.00]
A:B:C	0.11	[0.00, 0.27]

>

```
> eta_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type II)
```

Parameter	Eta2 (partial)	95% CI
A	4.25e-04	[0.00, 0.06]
B	0.59	[0.40, 0.71]
C	6.76e-03	[0.00, 0.12]
A:B	0.64	[0.47, 0.75]
A:C	0.21	[0.04, 0.40]
B:C	2.76e-03	[0.00, 0.04]
A:B:C	0.16	[0.01, 0.34]

```
> omega_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type II)
```

Parameter	Omega2 (partial)	95% CI
A	0.00	[0.00, 0.00]
B	0.53	[0.33, 0.66]
C	0.00	[0.00, 0.00]
A:B	0.58	[0.39, 0.70]
A:C	0.16	[0.02, 0.35]
B:C	0.00	[0.00, 0.00]
A:B:C	0.11	[0.00, 0.27]

```
> #EtaSq関数を使う方法
```

```
> library(DescTools)
```

```
> EtaSq(result.aov)
```

	eta.sq	eta.sq.part
A	8.997661e-05	0.0004249894
B	3.093846e-01	0.5938174596
C	1.439626e-03	0.0067567568
A:B	3.806460e-01	0.6426889480
A:C	5.623538e-02	0.2099428955
B:C	5.848479e-04	0.0027559890
A:B:C	3.999460e-02	0.1589486859

```
> #etaSquared関数を使う方法
```

```
> library(lsr)
```

```
> etaSquared(result.aov)
```

	eta.sq	eta.sq.part
A	8.997661e-05	0.0004249894
B	3.093846e-01	0.5938174596
C	1.439626e-03	0.0067567568
A:B	3.806460e-01	0.6426889480
A:C	5.623538e-02	0.2099428955
B:C	5.848479e-04	0.0027559890
A:B:C	3.999460e-02	0.1589486859

```
> # 単純効果
```

```
> library(phia)
```

```
>
```

```
> # B*Cの各水準におけるAの主効果
```

```
> testInteractions(mod.l, fixed=c("B", "C"), across="A")
```

```
F Test:
```

```
P-value adjustment method: holm
```

	Value	Df	Sum of Sq	F	Pr(>F)
b1 : c1	5.0	1	62.5	19.1327	0.0001961 ***
b2 : c1	5.6	1	78.4	24.0000	4.545e-05 ***
b3 : c1	-5.8	1	84.1	25.7449	3.132e-05 ***

```

b1 : c2      2.8  1      19.6  6.0000 0.0360117 *
b2 : c2     -1.6  1       6.4  1.9592 0.1680307
b3 : c2     -6.4  1     102.4 31.3469 6.131e-06 ***
Residuals      48      156.8

```

```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>

```

```
> # A*Cの各水準におけるBの2水準間の効果
```

```
> testInteractions(mod.1, fixed=c("A","C"), pairwise="B")
```

```
F Test:
```

```
P-value adjustment method: holm
```

	Value	Df	Sum of Sq	F	Pr(>F)
b1-b2 : a1 : c1	-1.6	1	6.4	1.9592	1.00000
b1-b3 : a1 : c1	0.6	1	0.9	0.2755	1.00000
b2-b3 : a1 : c1	2.2	1	12.1	3.7041	0.42153
b1-b2 : a2 : c1	-1.0	1	2.5	0.7653	1.00000
b1-b3 : a2 : c1	-10.2	1	260.1	79.6224	1.104e-10 ***
b2-b3 : a2 : c1	-9.2	1	211.6	64.7755	2.048e-09 ***
b1-b2 : a1 : c2	1.2	1	3.6	1.1020	1.00000
b1-b3 : a1 : c2	0.2	1	0.1	0.0306	1.00000
b2-b3 : a1 : c2	-1.0	1	2.5	0.7653	1.00000
b1-b2 : a2 : c2	-3.2	1	25.6	7.8367	0.05882 .
b1-b3 : a2 : c2	-9.0	1	202.5	61.9898	3.423e-09 ***
b2-b3 : a2 : c2	-5.8	1	84.1	25.7449	5.638e-05 ***
Residuals		48	156.8		

```

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>
>
>
>
>

```

```
> #平均値を比較する棒グラフの作成
```

```

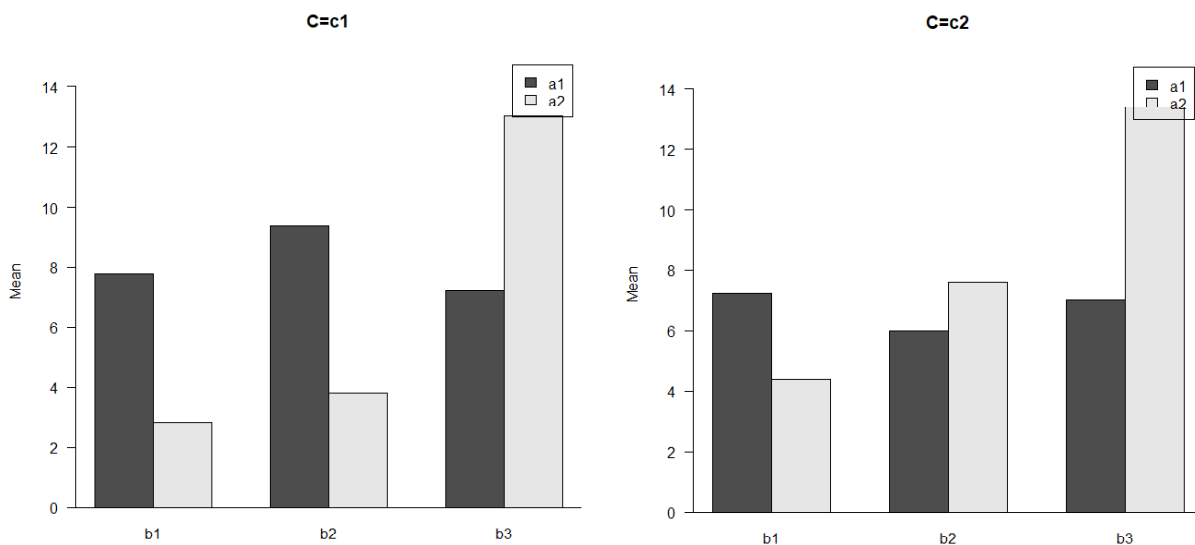
> d1t <- d1[d1$C=="c1",]
> m1t <- tapply(d1t[, "Y"], list(d1t[, "A"], d1t[, "B"]), mean)
> barplot(m1t, beside=TRUE, legend=TRUE, ylim=c(0, 20), ylab="Mean", main="C=c1")
> segments(0, 0, 10, 0)
>

```

```

> d1t <- d1[d1$C=="c2",]
> m1t <- tapply(d1t[, "Y"], list(d1t[, "A"], d1t[, "B"]), mean)
> barplot(m1t, beside=TRUE, legend=TRUE, ylim=c(0, 20), ylab="Mean", main="C=c2")
> segments(0, 0, 10, 0)
>
>
>
>

```



## 2つの被験者間要因と1つの被験者内要因(2B1W)

## stackデータの作成

被験者内要因の水準ごとにstackデータを作成して、最後に縦に繋げる

```
vn.x <- c("c1", "c2")
d3 <- NULL
for(i in vn.x) {
  d2 <- d1
  d2$C <- i
  d2$Y <- d1[,i]
  d3 <- rbind(d3, d2)
}
d4 <- d3[, c("id", "A", "B", "C", "Y")]
d4$id <- as.factor(d4$id)
d4$A <- as.factor(d4$A)
d4$B <- as.factor(d4$B)
d4$C <- as.factor(d4$C)
head(d4)
```

```
データフレーム名 : d4
被験者ID          : id
被験者間要因変数1 : A
被験者間要因変数2 : B
被験者内要因変数 : C
従属変数          : Y
```

## 【重要!!】

被験者idも含め、独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをしておく。

factor型にしていないと、間違った分散分析をした結果を出力してしまう。

間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

## 分散分析

## aovを使う方法

```
result.aov <- aov(Y ~ A*B*C + Error(id + id:A*B + id:A*B:C), data=d4)
summary(result.aov)
```

「球面性の仮定」が満たされることを前提としている。

## Anovaを使う方法

```
データフレーム 名 : d1
被験者ID        : id
被験者間要因変数1 : A
被験者間要因変数2 : B
被験者内要因名  : fac.c <- factor(c("c1", "c2"))
全被験者内要因  : youin
従属変数        : Y
```

```
library(car)
d1$A <- as.factor(d1$A)
d1$B <- as.factor(d1$B)
fac.c <- factor(c("c1", "c2"))
youin <- data.frame(fac.c)
mod.1 <- lm(cbind(c1, c2) ~ A*B, data=d1)
result.ANOVA <- Anova(mod.1, idata=youin, idesign=~fac.c)
summary(result.ANOVA, multivariate=FALSE)
```

「球面性の仮定」の検討をして、修正した結果も表示する。

あらかじめcarパッケージをインストールしておく必要がある。

multivariate=TRUE としておくと、多変量分散分析の結果も表示してくれる。

## 多重比較

## 被験者間要因

```
pairwise.t.test(d1$Y, d1$A, p.adjust.method="bonferroni")
```

## 被験者内要因

```
pairwise.t.test(d1$Y, d1$C, p.adjust.method="bonferroni", paired=TRUE)
```

p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"

## 効果量

effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```
library(effectsize)
```

```
eta_squared(分散分析オブジェクト, alternative="two.sided")
```

```
omega_squared(分散分析オブジェクト, alternative="two.sided")
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際, alternative="two.sided" を指定しておく, 下限と上限を推定する信頼区間となる。(指定しない場合, 上限は1.00となる。)

DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
```

```
EtaSq(aovオブジェクト, type=1)
```

各効果について,  $\eta^2$  乗, 偏  $\eta^2$  乗, 一般化  $\eta^2$  乗の値を計算する。

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。

typeオプションでtype=1と指定する。(デフォルトはtype=2)

```
> setwd("di:YY")
> d1 <- read.table("2B1W平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> head(d1)
```

```
  id  A  B c1 c2
1  1 a1 b1  8  9
2  2 a1 b1  6  6
3  3 a1 b1  8  8
4  4 a1 b1  9  6
5  5 a1 b1  8  7
6  6 a1 b2  6 10
```

```
>
>
```

```
> # stackデータの作成
```

```
> vn.x <- c("c1", "c2")
```

```
> d3 <- NULL
```

```
> for(i in vn.x){
```

```
+ d2 <- d1
```

```
+ d2$C <- i
```

```
+ d2$Y <- d1[,i]
```

```
+ d3 <- rbind(d3, d2)
```

```
+ }
```

```
> d4 <- d3[, c("id", "A", "B", "C", "Y")]
```

```
> d4$id <- as.factor(d4$id)
```

```
> d4$A <- as.factor(d4$A)
```

```
> d4$B <- as.factor(d4$B)
```

```
> d4$C <- as.factor(d4$C)
```

```
> head(d4)
```

```
  id  A  B  C  Y
1  1 a1 b1 c1  8
2  2 a1 b1 c1  6
3  3 a1 b1 c1  8
4  4 a1 b1 c1  9
5  5 a1 b1 c1  8
6  6 a1 b2 c1  6
```

```
>
>
>
```

1	id	A	B	c1	c2
2	1	a1	b1	8	9
3	2	a1	b1	6	6
4	3	a1	b1	8	8
5	4	a1	b1	9	6
6	5	a1	b1	8	7
7	6	a1	b2	6	10
8	7	a1	b2	9	6
9	8	a1	b2	10	5
10	9	a1	b2	11	3
11	10	a1	b2	11	6
12	11	a1	b3	7	7
13	12	a1	b3	8	6
14	13	a1	b3	7	8
15	14	a1	b3	8	6
16	15	a1	b3	6	8
17	16	a2	b1	2	4
18	17	a2	b1	4	4
19	18	a2	b1	3	3
20	19	a2	b1	2	7
21	20	a2	b1	3	4
22	21	a2	b2	4	6
23	22	a2	b2	3	5
24	23	a2	b2	2	6
25	24	a2	b2	2	9
26	25	a2	b2	8	12
27	26	a2	b3	11	15
28	27	a2	b3	15	15
29	28	a2	b3	12	13
30	29	a2	b3	12	13
31	30	a2	b3	15	11

```

>
> #各群の人数・平均値・標準偏差
> library(psych)
> (tABC <- describeBy(d4$Y, list(d4$A, d4$B, d4$C), mat=TRUE, digits=2))
  item group1 group2 group3 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1     a1     b1     c1   1  5  7.8 1.10      8    7.8 0.00    6   9    3 -0.62   -1.25 0.49
X12    2     a2     b1     c1   1  5  2.8 0.84      3    2.8 1.48    2   4    2  0.25   -1.82 0.37
X13    3     a1     b2     c1   1  5  9.4 2.07     10    9.4 1.48    6  11    5 -0.69   -1.41 0.93
X14    4     a2     b2     c1   1  5  3.8 2.49      3    3.8 1.48    2   8    6  0.80   -1.27 1.11
X15    5     a1     b3     c1   1  5  7.2 0.84      7    7.2 1.48    6   8    2 -0.25   -1.82 0.37
X16    6     a2     b3     c1   1  5 13.0 1.87     12   13.0 1.48   11  15    4  0.18   -2.18 0.84
X17    7     a1     b1     c2   1  5  7.2 1.30      7    7.2 1.48    6   9    3  0.26   -1.96 0.58
X18    8     a2     b1     c2   1  5  4.4 1.52      4    4.4 0.00    3   7    4  0.84   -1.12 0.68
X19    9     a1     b2     c2   1  5  6.0 2.55      6    6.0 1.48    3  10    7  0.43   -1.40 1.14
X110   10     a2     b2     c2   1  5  7.6 2.88      6    7.6 1.48    5  12    7  0.52   -1.73 1.29
X111   11     a1     b3     c2   1  5  7.0 1.00      7    7.0 1.48    6   8    2  0.00   -2.20 0.45
X112   12     a2     b3     c2   1  5 13.4 1.67     13   13.4 2.97   11  15    4 -0.25   -1.82 0.75
>

> (tAB <- describeBy(d4$Y, list(d4$A, d4$B), mat=TRUE, digits=2))
  item group1 group2 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1     a1     b1   1 10  7.5 1.18      8.0    7.50 1.48    6   9    3 -0.18   -1.67 0.37
X12    2     a2     b1   1 10  3.6 1.43      3.5    3.38 0.74    2   7    5  1.05    0.52 0.45
X13    3     a1     b2   1 10  7.7 2.83      7.5    7.88 3.71    3  11    8 -0.18   -1.66 0.90
X14    4     a2     b2   1 10  5.7 3.23      5.5    5.38 3.71    2  12   10  0.51   -1.03 1.02
X15    5     a1     b3   1 10  7.1 0.88      7.0    7.12 1.48    6   8    2 -0.16   -1.81 0.28
X16    6     a2     b3   1 10 13.2 1.69     13.0   13.25 2.97   11  15    4 -0.03   -1.85 0.53
>

> (tAC <- describeBy(d4$Y, list(d4$A, d4$C), mat=TRUE, digits=2))
  item group1 group2 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1     a1      c1   1 15  8.13 1.64      8    8.08 1.48    6  11    5  0.34   -1.04 0.42
X12    2     a2      c1   1 15  6.53 5.05      4    6.23 2.97    2  15   13  0.57   -1.50 1.31
X13    3     a1      c2   1 15  6.73 1.71      6    6.77 1.48    3  10    7 -0.09   -0.25 0.44
X14    4     a2      c2   1 15  8.47 4.32      7    8.38 4.45    3  15   12  0.25   -1.67 1.12
>

> (tBC <- describeBy(d4$Y, list(d4$B, d4$C), mat=TRUE, digits=2))
  item group1 group2 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1      b1      c1   1 10  5.3 2.79      5.0    5.25 4.45    2   9    7  0.05   -1.94 0.88
X12    2      b2      c1   1 10  6.6 3.66      7.0    6.62 5.19    2  11    9 -0.07   -1.87 1.16
X13    3      b3      c1   1 10 10.1 3.35      9.5   10.00 3.71    6  15    9  0.27   -1.66 1.06
X14    4      b1      c2   1 10  5.8 1.99      6.0    5.75 2.97    3   9    6  0.09   -1.56 0.63
X15    5      b2      c2   1 10  6.8 2.70      6.0    6.62 1.48    3  12    9  0.59   -0.95 0.85
X16    6      b3      c2   1 10 10.2 3.61      9.5   10.12 5.19    6  15    9  0.13   -1.85 1.14
>

> (tA <- describeBy(d4$Y, list(d4$A), mat=TRUE, digits=2))
  item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1     a1   1 30  7.43 1.79      7.5    7.38 2.22    3  11    8 0.05   -0.14 0.33
X12    2     a2   1 30  7.50 4.73      6.0    7.25 5.19    2  15   13 0.35   -1.52 0.86
>

> (tB <- describeBy(d4$Y, list(d4$B), mat=TRUE, digits=2))
  item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1      b1   1 20  5.55 2.37      6.0    5.56 2.97    2   9    7 -0.03   -1.57 0.53
X12    2      b2   1 20  6.70 3.13      6.0    6.69 4.45    2  12   10 0.11   -1.34 0.70
X13    3      b3   1 20 10.15 3.39      9.5   10.06 3.71    6  15    9  0.22   -1.62 0.76
>

> (tC <- describeBy(d4$Y, list(d4$C), mat=TRUE, digits=2))
  item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X11    1      c1   1 30  7.33 3.78      8.0    7.17 4.45    2  15   13 0.22   -0.88 0.69
X12    2      c2   1 30  7.60 3.35      6.5    7.29 2.22    3  15   12 0.77   -0.40 0.61
>

> (tall <- describe(d4$Y))
  vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
X1     1 60  7.47 3.54      7    7.25 2.97    2  15   13 0.44   -0.55 0.46
>

```



```
> # 分散分析
> #aovを使う方法
> result.aov <- aov(Y ~ A*B*C + Error(id + id:A*B + id:A:B:C), data=d4)
警告メッセージ:
aov(Y ~ A * B * C + Error(id:A*B + id:A:B:C), data = d4) で:
Error() モデルは特異です
```

```
> summary(result.aov)
```

```
Error: id
      Df Sum Sq Mean Sq F value    Pr(>F)
A       1   0.07    0.07   0.022    0.882
B       2 229.23   114.62   38.419 3.30e-08 ***
A:B     2 282.03   141.02   47.268 4.75e-09 ***
Residuals 24   71.60     2.98
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Error: id:A:B:C
      Df Sum Sq Mean Sq F value    Pr(>F)
C       1   1.07    1.07   0.300 0.58865
A:C     1  41.67   41.67  11.737 0.00221 **
B:C     2   0.43    0.22   0.061 0.94094
A:B:C   2  29.63   14.82   4.174 0.02783 *
Residuals 24  85.20     3.55
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>
```

```
> # Anovaを使う方法
> library(car)
> d1$A <- as.factor(d1$A)
> d1$B <- as.factor(d1$B)
> fac.c <- factor(c("c1", "c2"))
> youin <- data.frame(fac.c)
> mod.1 <- lm(cbind(c1, c2) ~ A*B, data=d1)
> result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.c)
> summary(result.Anova, multivariate=FALSE)
```

Univariate Type II Repeated-Measures ANOVA Assuming Sphericity

```
      SS num Df Error SS den Df      F    Pr(>F)
(Intercept) 3345.1      1    71.6    24 1121.2514 < 2.2e-16 ***
A           0.1      1    71.6    24   0.0223  0.882418
B          229.2      2    71.6    24  38.4190 3.304e-08 ***
A:B        282.0      2    71.6    24  47.2682 4.746e-09 ***
fac.c       1.1      1    85.2    24   0.3005  0.588651
A:fac.c     41.7      1    85.2    24  11.7371 0.002212 **
B:fac.c      0.4      2    85.2    24   0.0610 0.940938
A:B:fac.c   29.6      2    85.2    24   4.1737 0.027827 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>
>
>
>
```

```
> #多重比較
> #ボンフェロニの方法
> (mc.bon.A <- pairwise.t.test(d4$Y, d4$A, p.adjust.method="bonferroni"))
```

Pairwise comparisons using t tests with pooled SD

data: d4\$Y and d4\$A

```
a1
a2 0.94
```

```
P value adjustment method: bonferroni
```

```
> (mc.bon.B <- pairwise.t.test(d4$Y, d4$B, p.adjust.method="bonferroni"))
```

```
Pairwise comparisons using t tests with pooled SD
```

```
data: d4$Y and d4$B
```

```
      b1      b2
b2 0.6896 -
b3 2.9e-05 0.0018
```

```
P value adjustment method: bonferroni
```

```
> (mc.bon.C <- pairwise.t.test(d4$Y, d4$C, p.adjust.method="bonferroni", paired=TRUE))
```

```
Pairwise comparisons using paired t tests
```

```
data: d4$Y and d4$C
```

```
      c1
c2 0.66
```

```
P value adjustment method: bonferroni
```

```
>
```

```
>
```

```
>
```

```
> # 効果量
```

```
> #eta_squared, omega_squared関数を使う方法
```

```
> library(effectsiz)
```

```
>
```

```
> eta_squared(result.aov, alternative="two.sided")
```

```
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Eta2 (partial)	95% CI
id	A	9.30e-04	[0.00, 0.12]
id	B	0.76	[0.56, 0.85]
id	A:B	0.80	[0.62, 0.88]
id:A:B:C	C	0.01	[0.00, 0.20]
id:A:B:C	A:C	0.33	[0.06, 0.56]
id:A:B:C	B:C	5.06e-03	[0.00, 0.07]
id:A:B:C	A:B:C	0.26	[0.00, 0.50]

```
> omega_squared(result.aov, alternative="two.sided")
```

```
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Omega2 (partial)	95% CI
id	A	0.00	[0.00, 0.00]
id	B	0.73	[0.51, 0.84]
id	A:B	0.77	[0.58, 0.86]
id:A:B:C	C	0.00	[0.00, 0.00]
id:A:B:C	A:C	0.19	[0.00, 0.45]
id:A:B:C	B:C	0.00	[0.00, 0.00]
id:A:B:C	A:B:C	0.12	[0.00, 0.35]

```
>
```

```
> eta_squared(result.Anova, alternative="two.sided")
```

```
# Effect Size for ANOVA (Type II)
```

Parameter	Eta2 (partial)	95% CI
-----------	----------------	--------

A	9.30e-04	[0.00, 0.12]
B	0.76	[0.56, 0.85]
A:B	0.80	[0.62, 0.88]
fac.c	0.01	[0.00, 0.20]
A:fac.c	0.33	[0.06, 0.56]
B:fac.c	5.06e-03	[0.00, 0.07]
A:B:fac.c	0.26	[0.00, 0.50]

```
> omega_squared(result.Anova, alternative="two.sided")
# Effect Size for ANOVA (Type II)
```

Parameter	Omega2 (partial)	95% CI
A	0.00	[0.00, 0.00]
B	0.73	[0.51, 0.84]
A:B	0.77	[0.58, 0.86]
fac.c	0.00	[0.00, 0.00]
A:fac.c	0.19	[0.00, 0.45]
B:fac.c	0.00	[0.00, 0.00]
A:B:fac.c	0.12	[0.00, 0.35]

```
> #EtaSq関数を使う方法
```

```
> library(DescTools)
```

```
> EtaSq(result.aov, type=1)
```

	eta.sq	eta.sq.part	eta.sq.gen
A	8.997661e-05	0.0009302326	0.0004249894
B	3.093846e-01	0.7619944598	0.5938174596
A:B	3.806460e-01	0.7975303987	0.6426889480
C	1.439626e-03	0.0123647604	0.0067567568
A:C	5.623538e-02	0.3284287966	0.2099428955
B:C	5.848479e-04	0.0050603348	0.0027559890
A:B:C	3.999460e-02	0.2580551524	0.1589486859

```
> #平均値を比較する棒グラフの作成
```

```
> d4t <- d4[d4$C=="c1",]
```

```
> m4t <- tapply(d4t[, "Y"], list(d4t[, "A"], d4t[, "B"]), mean)
```

```
> barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean", main="C=c1")
```

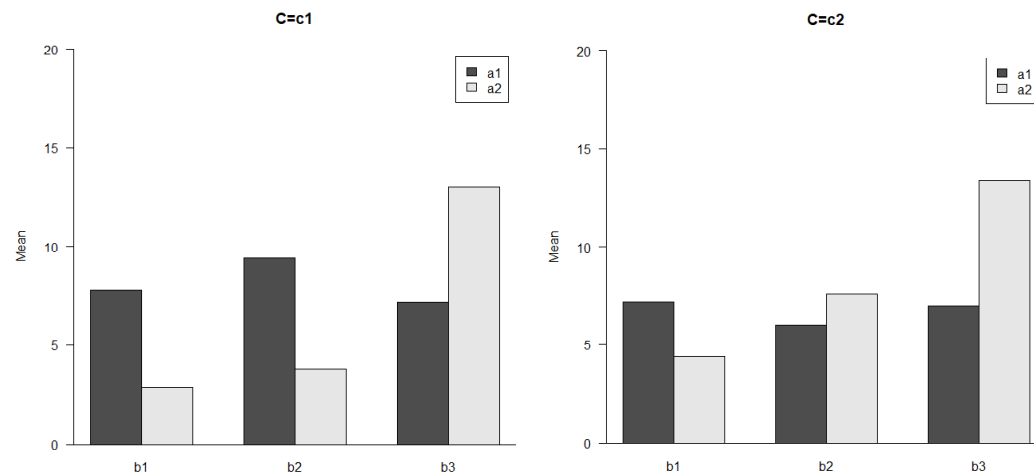
```
> segments(0, 0, 10, 0)
```

```
> d4t <- d4[d4$C=="c2",]
```

```
> m4t <- tapply(d4t[, "Y"], list(d4t[, "A"], d4t[, "B"]), mean)
```

```
> barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean", main="C=c2")
```

```
> segments(0, 0, 10, 0)
```



## 1つの被験者間要因と2つの被験者内要因(1B2W)

## stackデータの作成

被験者内要因の水準ごとにstackデータを作成して、最後に縦に繋げる

```
vn.x <- c("b1c1", "b1c2", "b2c1", "b2c2", "b3c1", "b3c2")
d3 <- NULL
for(i in vn.x) {
  d2 <- d1
  d2$X <- i
  d2$Y <- d1[,i]
  d3 <- rbind(d3, d2)
}
d3$B <- substr(d3$X, 1, 2)
d3$C <- substr(d3$X, 3, 4)
d4 <- d3[, c("id", "A", "B", "C", "Y")]
d4$id <- as.factor(d4$id)
d4$A <- as.factor(d4$A)
d4$B <- as.factor(d4$B)
d4$C <- as.factor(d4$C)
head(d4)
```

```
データフレーム名 : d4
被験者ID          : id
被験者間要因変数  : A
被験者内要因変数1 : B
被験者内要因変数2 : C
従属変数          : Y
```

## 【重要!!】

被験者idも含め、独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。

factor型にしていないと、間違った分散分析をした結果を出力してしまう。

間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

## 分散分析

## aovを使う方法

```
result.aov <- aov(Y ~ A*B*C + Error(id + id:A + id:A:B + id:A:C + id:A:B:C), data=d4)
summary(result.aov)
```

「球面性の仮定」が満たされることを前提としている。

## Anovaを使う方法

```
データフレーム名 : d1
被験者ID          : id
被験者間要因変数  : A
被験者内要因名1   : fac.b <- factor(c("b1", "b1", "b2", "b2", "b3", "b3"))
被験者内要因名2   : fac.c <- factor(c("c1", "c2", "c1", "c2", "c1", "c2"))
全被験者内要因    : youin
従属変数          : Y
```

## library(car)

```
d1$A <- as.factor(d1$A)
fac.b <- factor(c("b1", "b1", "b2", "b2", "b3", "b3"))
fac.c <- factor(c("c1", "c2", "c1", "c2", "c1", "c2"))
youin <- data.frame(fac.b, fac.c)
mod.1 <- lm(cbind(b1c1, b1c2, b2c1, b2c2, b3c1, b3c2) ~ A, data=d1)
result.Anova <- Anova(mod.1, idata=youin, idesign=~ fac.b*fac.c)
summary(result.Anova, multivariate=FALSE)
```

「球面性の仮定」の検討をして、修正した結果も表示する。

あらかじめcarパッケージをインストールしておく必要がある。

multivariate=TRUE としておくと、多変量分散分析の結果も表示してくれる。

## 多重比較

## 被験者間要因

```
pairwise.t.test(d1$Y, d1$A, p.adjust.method="bonferroni")
```

## 被験者内要因

```
pairwise.t.test(d1$Y, d1$C, p.adjust.method="bonferroni", paired=TRUE)
```

p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"

## 効果量

effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```
library(effectsize)
```

```
eta_squared(分散分析オブジェクト, alternative="two.sided")
```

```
omega_squared(分散分析オブジェクト, alternative="two.sided")
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際, alternative="two.sided" を指定しておく, 下限と上限を推定する信頼区間となる。(指定しない場合, 上限は1.00となる。)

## DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
```

```
EtaSq(aovオブジェクト, type=1)
```

各効果について,  $\eta^2$ 乗, 偏 $\eta^2$ 乗, 一般化 $\eta^2$ 乗の値を計算する。

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。

typeオプションでtype=1と指定する。(デフォルトはtype=2)

```
> setwd("d:¥¥")
> d1 <- read.table("1B2W平均値データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  id  A b1c1 b1c2 b2c1 b2c2 b3c1 b3c2
1  1 a1    8    9    6   10    7    7
2  2 a1    6    6    9    6    8    6
3  3 a1    8    8   10    5    7    8
4  4 a1    9    6   11    3    8    6
5  5 a1    8    7   11    6    6    8
6  6 a2    2    4    4    6   11   15
>
>
> # stackデータの作成
> vn.x <- c("b1c1", "b1c2", "b2c1", "b2c2", "b3c1", "b3c2")
> d3 <- NULL
> for(i in vn.x){
+   d2 <- d1
+   d2$X <- i
+   d2$Y <- d1[,i]
+   d3 <- rbind(d3, d2)
+ }
> d3$B <- substr(d3$X, 1, 2)
> d3$C <- substr(d3$X, 3, 4)
> d4<- d3[, c("id", "A", "B", "C", "Y")]
> d4$id <- as.factor(d4$id)
> d4$A <- as.factor(d4$A)
> d4$B <- as.factor(d4$B)
> d4$C <- as.factor(d4$C)
> head(d4)
  id  A  B  C  Y
1  1 a1  8  9  7
2  2 a1  6  6  8
3  3 a1  8  8  7
4  4 a1  9  6  8
5  5 a1  8  7  6
6  6 a2  2  4  6
7  7 a2  4  4  3
8  8 a2  3  3  2
9  9 a2  2  7  9
10 10 a2  3  4  8
11 11 a2  8  12 15
12 12 a2 15 11
```

```

1 1 a1 b1 c1 8
2 2 a1 b1 c1 6
3 3 a1 b1 c1 8
4 4 a1 b1 c1 9
5 5 a1 b1 c1 8
6 6 a2 b1 c1 2
>
>
>
>
> #各群の人数・平均値・標準偏差
> library(psych)
> (tABC <- describeBy(d4$Y, list(d4$A, d4$B, d4$C), mat=TRUE, digits=2))
      item group1 group2 group3 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      a1      b1      c1  1 5  7.8 1.10      8    7.8 0.00  6  9    3 -0.62   -1.25 0.49
X12      2      a2      b1      c1  1 5  2.8 0.84      3    2.8 1.48  2  4    2  0.25   -1.82 0.37
X13      3      a1      b2      c1  1 5  9.4 2.07     10    9.4 1.48  6 11    5 -0.69   -1.41 0.93
X14      4      a2      b2      c1  1 5  3.8 2.49      3    3.8 1.48  2  8    6  0.80   -1.27 1.11
X15      5      a1      b3      c1  1 5  7.2 0.84      7    7.2 1.48  6  8    2 -0.25   -1.82 0.37
X16      6      a2      b3      c1  1 5 13.0 1.87     12   13.0 1.48 11 15    4  0.18   -2.18 0.84
X17      7      a1      b1      c2  1 5  7.2 1.30      7    7.2 1.48  6  9    3  0.26   -1.96 0.58
X18      8      a2      b1      c2  1 5  4.4 1.52      4    4.4 0.00  3  7    4  0.84   -1.12 0.68
X19      9      a1      b2      c2  1 5  6.0 2.55      6    6.0 1.48  3 10    7  0.43   -1.40 1.14
X110     10      a2      b2      c2  1 5  7.6 2.88      6    7.6 1.48  5 12    7  0.52   -1.73 1.29
X111     11      a1      b3      c2  1 5  7.0 1.00      7    7.0 1.48  6  8    2  0.00   -2.20 0.45
X112     12      a2      b3      c2  1 5 13.4 1.67     13   13.4 2.97 11 15    4 -0.25   -1.82 0.75
>

> (tAB <- describeBy(d4$Y, list(d4$A, d4$B), mat=TRUE, digits=2))
      item group1 group2 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      a1      b1  1 10  7.5 1.18      8.0    7.50 1.48  6  9    3 -0.18   -1.67 0.37
X12      2      a2      b1  1 10  3.6 1.43      3.5    3.38 0.74  2  7    5  1.05    0.52 0.45
X13      3      a1      b2  1 10  7.7 2.83      7.5    7.88 3.71  3 11    8 -0.18   -1.66 0.90
X14      4      a2      b2  1 10  5.7 3.23      5.5    5.38 3.71  2 12   10  0.51   -1.03 1.02
X15      5      a1      b3  1 10  7.1 0.88      7.0    7.12 1.48  6  8    2 -0.16   -1.81 0.28
X16      6      a2      b3  1 10 13.2 1.69     13.0   13.25 2.97 11 15    4 -0.03   -1.85 0.53
>

> (tAC <- describeBy(d4$Y, list(d4$A, d4$C), mat=TRUE, digits=2))
      item group1 group2 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      a1      c1  1 15  8.13 1.64      8    8.08 1.48  6 11    5  0.34   -1.04 0.42
X12      2      a2      c1  1 15  6.53 5.05      4    6.23 2.97  2 15   13  0.57   -1.50 1.31
X13      3      a1      c2  1 15  6.73 1.71      6    6.77 1.48  3 10    7 -0.09   -0.25 0.44
X14      4      a2      c2  1 15  8.47 4.32      7    8.38 4.45  3 15   12  0.25   -1.67 1.12
>

> (tBC <- describeBy(d4$Y, list(d4$B, d4$C), mat=TRUE, digits=2))
      item group1 group2 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      b1      c1  1 10  5.3 2.79      5.0    5.25 4.45  2  9    7  0.05   -1.94 0.88
X12      2      b2      c1  1 10  6.6 3.66      7.0    6.62 5.19  2 11    9 -0.07   -1.87 1.16
X13      3      b3      c1  1 10 10.1 3.35      9.5   10.00 3.71  6 15    9  0.27   -1.66 1.06
X14      4      b1      c2  1 10  5.8 1.99      6.0    5.75 2.97  3  9    6  0.09   -1.56 0.63
X15      5      b2      c2  1 10  6.8 2.70      6.0    6.62 1.48  3 12    9  0.59   -0.95 0.85
X16      6      b3      c2  1 10 10.2 3.61      9.5   10.12 5.19  6 15    9  0.13   -1.85 1.14
>

> (tA <- describeBy(d4$Y, list(d4$A), mat=TRUE, digits=2))
      item group1 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      a1  1 30  7.43 1.79      7.5    7.38 2.22  3 11    8 0.05   -0.14 0.33
X12      2      a2  1 30  7.50 4.73      6.0    7.25 5.19  2 15   13 0.35   -1.52 0.86
>

> (tB <- describeBy(d4$Y, list(d4$B), mat=TRUE, digits=2))
      item group1 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      b1  1 20  5.55 2.37      6.0    5.56 2.97  2  9    7 -0.03   -1.57 0.53
X12      2      b2  1 20  6.70 3.13      6.0    6.69 4.45  2 12   10 0.11   -1.34 0.70
X13      3      b3  1 20 10.15 3.39      9.5   10.06 3.71  6 15    9 0.22   -1.62 0.76
>

> (tC <- describeBy(d4$Y, list(d4$C), mat=TRUE, digits=2))
      item group1 vars n mean sd median trimmed mad min max range skew kurtosis se
X11      1      c1  1 30  7.33 3.78      8.0    7.17 4.45  2 15   13 0.22   -0.88 0.69

```

```
X12      2      c2      1 30 7.60 3.35      6.5      7.29 2.22      3 15      12 0.77      -0.40 0.61
>
```

```
> (tall <- describe(d4$Y))
      vars  n mean      sd median trimmed  mad min max range skew kurtosis  se
X1       1 60 7.47 3.54      7      7.25 2.97   2 15     13 0.44     -0.55 0.46
```

```
> # 分散分析
```

```
> #aovを使う方法
```

```
> result.aov <- aov(Y ~ A*B*C + Error(id + id:A + id:A:B + id:A:C + id:A:B:C), data=d4)
```

```
警告メッセージ:
```

```
aov(Y ~ A * B * C + Error(id:A + id:A:B + id:A:C + id:A:B:C),  で:
Error() モデルは特異です
```

```
> summary(result.aov)
```

```
Error: id
      Df Sum Sq Mean Sq F value Pr(>F)
A       1  0.067   0.0667   0.024  0.882
Residuals 8 22.533   2.8167
```

```
Error: id:A:B
      Df Sum Sq Mean Sq F value Pr(>F)
B       2 229.23  114.62   37.38 9.34e-07 ***
A:B      2 282.03  141.02   45.98 2.33e-07 ***
Residuals 16  49.07    3.07
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Error: id:A:C
      Df Sum Sq Mean Sq F value Pr(>F)
C       1   1.07    1.07   0.199 0.6675
A:C      1 41.67   41.67   7.764 0.0237 *
Residuals 8 42.93    5.37
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Error: id:A:B:C
      Df Sum Sq Mean Sq F value Pr(>F)
B:C      2   0.43    0.217   0.082 0.9216
A:B:C     2 29.63   14.817   5.609 0.0143 *
Residuals 16 42.27    2.642
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>
```

```
> # Anovaを使う方法
```

```
> library(car)
```

```
> d1$A <- as.factor(d1$A)
```

```
> fac.b <- factor(c("b1", "b1", "b2", "b2", "b3", "b3"))
```

```
> fac.c <- factor(c("c1", "c2", "c1", "c2", "c1", "c2"))
```

```
> youin <- data.frame(fac.b, fac.c)
```

```
> mod.1 <- lm(cbind(b1c1, b1c2, b2c1, b2c2, b3c1, b3c2) ~ A, data=d1)
```

```
> result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.b*fac.c)
```

```
> summary(result.Anova, multivariate=FALSE)
```

Univariate Type II Repeated-Measures ANOVA Assuming Sphericity

	SS	num Df	Error SS	den Df	F	Pr(>F)
(Intercept)	3345.1	1	22.533	8	1187.5976	5.496e-10 ***
A	0.1	1	22.533	8	0.0237	0.88154
fac.b	229.2	2	49.067	16	37.3750	9.337e-07 ***
A:fac.b	282.0	2	49.067	16	45.9837	2.326e-07 ***
fac.c	1.1	1	42.933	8	0.1988	0.66754
A:fac.c	41.7	1	42.933	8	7.7640	0.02369 *
fac.b:fac.c	0.4	2	42.267	16	0.0820	0.92164

```
A:fac.b:fac.c 29.6 2 42.267 16 5.6088 0.01426 *
```

```
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

#### Mauchly Tests for Sphericity

	Test statistic	p-value
fac.b	0.52234	0.102998
A:fac.b	0.52234	0.102998
fac.b:fac.c	0.47128	0.071856
A:fac.b:fac.c	0.47128	0.071856

#### Greenhouse-Geisser and Huynh-Feldt Corrections for Departure from Sphericity

	GG eps	Pr(>F[GG])
fac.b	0.67674	3.718e-05 ***
A:fac.b	0.67674	1.439e-05 ***
fac.b:fac.c	0.65414	0.84339
A:fac.b:fac.c	0.65414	0.03164 *

```
---
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	HF eps	Pr(>F[HF])
fac.b	0.7659206	1.339476e-05
A:fac.b	0.7659206	4.589669e-06
fac.b:fac.c	0.7303445	8.662260e-01
A:fac.b:fac.c	0.7303445	2.650360e-02

```
> #多重比較
> #ボンフェロニの方法
> (mc.bon.A <- pairwise.t.test(d4$Y, d4$A, p.adjust.method="bonferroni"))
```

Pairwise comparisons using t tests with pooled SD

data: d4\$Y and d4\$A

```
  a1
a2 0.94
```

P value adjustment method: bonferroni

```
> (mc.bon.B <- pairwise.t.test(d4$Y, d4$B, p.adjust.method="bonferroni", paired=TRUE))
```

Pairwise comparisons using paired t tests

data: d4\$Y and d4\$B

```
  b1    b2
b2 0.2106 -
b3 0.0034 0.0258
```

P value adjustment method: bonferroni

```
> (mc.bon.C <- pairwise.t.test(d4$Y, d4$C, p.adjust.method="bonferroni", paired=TRUE))
```

Pairwise comparisons using paired t tests

data: d4\$Y and d4\$C

```
  c1
c2 0.66
```

P value adjustment method: bonferroni

```
>
```



&gt;

&gt; # 効果量

&gt; #eta\_squared, omega\_squared関数を使う方法

&gt; library(effects)

&gt;

&gt; eta\_squared(result.aov, alternative="two.sided")

# Effect Size for ANOVA (Type I)

Group	Parameter	Eta2 (partial)	95% CI
id	A	2.95e-03	[0.00, 0.26]
id:A:B	B	0.82	[0.61, 0.90]
id:A:B	A:B	0.85	[0.67, 0.92]
id:A:C	C	0.02	[0.00, 0.41]
id:A:C	A:C	0.49	[0.00, 0.77]
id:A:B:C	B:C	0.01	[0.00, 0.13]
id:A:B:C	A:B:C	0.41	[0.03, 0.65]

&gt; omega\_squared(result.aov, alternative="two.sided")

# Effect Size for ANOVA (Type I)

Group	Parameter	Omega2 (partial)	95% CI
id	A	0.00	[0.00, 0.00]
id:A:B	B	0.73	[0.43, 0.85]
id:A:B	A:B	0.77	[0.51, 0.87]
id:A:C	C	0.00	[0.00, 0.00]
id:A:C	A:C	0.33	[0.00, 0.68]
id:A:B:C	B:C	0.00	[0.00, 0.00]
id:A:B:C	A:B:C	0.25	[0.00, 0.53]

&gt;

&gt; eta\_squared(result.Anova, alternative="two.sided")

# Effect Size for ANOVA (Type II)

Parameter	Eta2 (partial)	95% CI
A	2.95e-03	[0.00, 0.28]
fac.b	0.82	[0.61, 0.90]
A:fac.b	0.85	[0.67, 0.92]
fac.c	0.02	[0.00, 0.41]
A:fac.c	0.49	[0.00, 0.77]
fac.b:fac.c	0.01	[0.00, 0.13]
A:fac.b:fac.c	0.41	[0.03, 0.65]

&gt; omega\_squared(result.Anova, alternative="two.sided")

# Effect Size for ANOVA (Type II)

Parameter	Omega2 (partial)	95% CI
A	0.00	[0.00, 0.00]
fac.b	0.73	[0.43, 0.85]
A:fac.b	0.77	[0.51, 0.87]
fac.c	0.00	[0.00, 0.00]
A:fac.c	0.33	[0.00, 0.68]
fac.b:fac.c	0.00	[0.00, 0.00]
A:fac.b:fac.c	0.25	[0.00, 0.53]

&gt;

&gt;

## &gt; #EtaSq関数を使う方法

&gt; library(DescTools)

&gt; EtaSq(result.aov, type=1)

	eta. sq	eta. sq. part	eta. sq. gen
A	8.997661e-05	0.002949853	0.0004249894
B	3.093846e-01	0.823691460	0.5938174596
A:B	3.806460e-01	0.851807108	0.6426889480
C	1.439626e-03	0.024242424	0.0067567568
A:C	5.623538e-02	0.492513790	0.2099428955
B:C	5.848479e-04	0.010148322	0.0027559890
A:B:C	3.999460e-02	0.412146500	0.1589486859

&gt;

## &gt; #平均値を比較する棒グラフの作成

&gt; d4t &lt;- d4[d4\$C=="c1",]

&gt; m4t &lt;- tapply(d4t[, "Y"], list(d4t[, "A"], d4t[, "B"]), mean)

&gt; barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean", main="C=c1")

&gt; segments(0, 0, 10, 0)

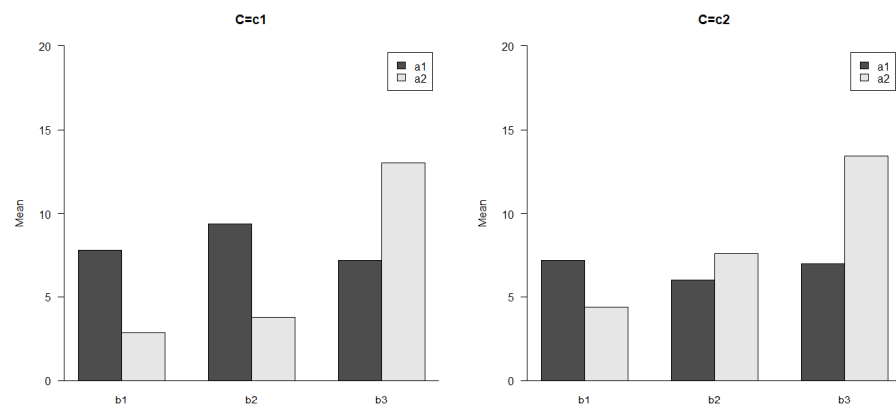
&gt;

&gt; d4t &lt;- d4[d4\$C=="c2",]

&gt; m4t &lt;- tapply(d4t[, "Y"], list(d4t[, "A"], d4t[, "B"]), mean)

&gt; barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean", main="C=c2")

&gt; segments(0, 0, 10, 0)



### 3つの被験者内要因(3W)

#### stackデータの作成

```
d3 <- stack(d2)
d3$A <- substr(d3$ind, 1, 2)
d3$B <- substr(d3$ind, 3, 4)
d3$C <- substr(d3$ind, 5, 6)
d3$id <- d1$id
d4 <- d3[, c("id", "A", "B", "C", "values")]
colnames(d4) <- c("id", "A", "B", "C", "Y")
d4$id <- as.factor(d4$id)
d4$A <- as.factor(d4$A)
d4$B <- as.factor(d4$B)
d4$C <- as.factor(d4$C)
head(d4)
```

```
データフレーム名 : d4
被験者ID           : id
被験者内要因変数1 : A
被験者内要因変数2 : B
被験者内要因変数3 : C
従属変数           : Y
```

#### 【重要！！】

被験者idも含め、独立変数をfactor型と言われる形式にしておかなければならない関数が多いので、まずこれをやっておく。

factor型にしていないと、間違った分散分析をした結果を出力してしまう。

間違っているかどうかは、独立変数の自由度Dfの値が「水準数-1」になっているか否かなどでわかる。

#### 分散分析

##### aovを使う方法

```
result.aov <- aov(Y ~ A*B*C + Error(id + id:A + id:B + id:C
+ id:A:B + id:A:C + id:B:C
+ id:A:B:C), data=d4)

summary(result.aov)
```

「球面性の仮定」が満たされることを前提としている。

#### Anovaを使う方法

データフレーム名 : d1

被験者ID : id

被験者内要因名1 : fac.a <- factor(c("a1", "a1", "a1", "a1", "a1", "a1", "a2", "a2", "a2", "a2", "a2", "a2"))

被験者内要因名2 : fac.b <- factor(c("b1", "b1", "b2", "b2", "b3", "b3", "b1", "b1", "b2", "b2", "b3", "b3"))

被験者内要因名3 : fac.c <- factor(c("c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2"))

全被験者内要因 : youin

従属変数 : Y

#### library(car)

fac.a <- factor(c("a1", "a1", "a1", "a1", "a1", "a1", "a2", "a2", "a2", "a2", "a2", "a2"))

fac.b <- factor(c("b1", "b1", "b2", "b2", "b3", "b3", "b1", "b1", "b2", "b2", "b3", "b3"))

fac.c <- factor(c("c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2"))

youin <- data.frame(fac.a, fac.b, fac.c)

mod.1 <- lm(cbind(a1b1c1, a1b1c2, a1b2c1, a1b2c2, a1b3c1, a1b3c2,
a2b1c1, a2b1c2, a2b2c1, a2b2c2, a2b3c1, a2b3c2) ~ 1, data=d1)

result.Anova <- Anova(mod.1, idata=youin, idesign=~ fac.a\*fac.b\*fac.c)

summary(result.Anova, multivariate=FALSE)

「球面性の仮定」の検討をして、修正した結果も表示する。

あらかじめcarパッケージをインストールしておく必要がある。

multivariate=TRUE としておく、多変量分散分析の結果も表示してくれる。

#### 多重比較

##### 被験者内要因

```
pairwise.t.test(d1$Y, d1$C, p.adjust.method="bonferroni", paired=TRUE)
```

```
p.adjusted.method: "holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr"
```

## 効果量

effectsizeパッケージのeta\_squared, omega\_squared関数を使う方法

```
library(effectsize)
```

```
eta_squared(分散分析オブジェクト, alternative="two.sided")
```

```
omega_squared(分散分析オブジェクト, alternative="two.sided")
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

aov, Anova関数で分散分析を実行した結果オブジェクトをeta\_squared, omega\_squaredに入れる。

効果量の信頼区間の推定する際, alternative="two.sided" を指定しておく, 下限と上限を推定する信頼区間となる。(指定しない場合, 上限は1.00となる。)

DescToolsパッケージのEtaSq関数を使う方法

```
library(DescTools)
```

```
EtaSq(aovオブジェクト, type=1)
```

各効果について,  $\eta^2$ 乗, 偏 $\eta^2$ 乗, 一般化 $\eta^2$ 乗の値を計算する。

あらかじめDescToolsパッケージをインストールしておく必要がある。

aov関数で分散分析を実行した結果オブジェクトをetaSqに入れる。

typeオプションでtype=1と指定する。(デフォルトはtype=2)

```
> setwd("d:¥¥")
> d1 <- read.table("3W平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d2 <- d1[-1]
> head(d2)
  a1b1c1 a1b1c2 a1b2c1 a1b2c2 a1b3c1 a1b3c2 a2b1c1 a2b1c2 a2b2c1 a2b2c2
1      8      9      6      10      7      7      2      4      4      6
2      6      6      9      6      8      6      4      4      3      5
3      8      8     10      5      7      8      3      3      2      6
4      9      6     11      3      8      6      2      7      2      9
5      8      7     11      6      6      8      3      4      8     12
  a2b3c1 a2b3c2
1     11     15
2     15     15
3     12     13
4     12     13
5     15     11
```

	id	a1b1c1	a1b1c2	a1b2c1	a1b2c2	a1b3c1	a1b3c2	a2b1c1	a2b1c2	a2b2c1	a2b2c2	a2b3c1	a2b3c2
1													
2	1	8	9	6	10	7	7	2	4	4	6	11	15
3	2	6	6	9	6	8	6	4	4	3	5	15	15
4	3	8	8	10	5	7	8	3	3	2	6	12	13
5	4	9	6	11	3	8	6	2	7	2	9	12	13
6	5	8	7	11	6	6	8	3	4	8	12	15	11
7													

> #stackデータの作成

```
> d3 <- stack(d2)
```

```
> d3$A <- substr(d3$ind, 1, 2)
```

```
> d3$B <- substr(d3$ind, 3, 4)
```

```
> d3$C <- substr(d3$ind, 5, 6)
```

```
> d3$id <- d1$id
```

```
> d4 <- d3[, c("id", "A", "B", "C", "values")]
```

```
> colnames(d4) <- c("id", "A", "B", "C", "Y")
```

```
> d4$id <- as.factor(d4$id)
```

```
> d4$A <- as.factor(d4$A)
```

```
> d4$B <- as.factor(d4$B)
```

```
> d4$C <- as.factor(d4$C)
```

```
> head(d4)
```

```
  id A B C Y
1  1 a1 b1 c1 8
2  2 a1 b1 c1 6
3  3 a1 b1 c1 8
4  4 a1 b1 c1 9
5  5 a1 b1 c1 8
6  1 a1 b1 c2 9
```

```
>
>
>
```

> #各群の人数・平均値・標準偏差

> library(psych)

> (tABC <- describeBy(d4\$Y, list(d4\$A, d4\$B, d4\$C), mat=TRUE, digits=2))

	item	group1	group2	group3	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	a1	b1	c1	1	5	7.8	1.10	8	7.8	0.00	6	9	3	-0.62	-1.25	0.49
X12	2	a2	b1	c1	1	5	2.8	0.84	3	2.8	1.48	2	4	2	0.25	-1.82	0.37
X13	3	a1	b2	c1	1	5	9.4	2.07	10	9.4	1.48	6	11	5	-0.69	-1.41	0.93
X14	4	a2	b2	c1	1	5	3.8	2.49	3	3.8	1.48	2	8	6	0.80	-1.27	1.11
X15	5	a1	b3	c1	1	5	7.2	0.84	7	7.2	1.48	6	8	2	-0.25	-1.82	0.37
X16	6	a2	b3	c1	1	5	13.0	1.87	12	13.0	1.48	11	15	4	0.18	-2.18	0.84
X17	7	a1	b1	c2	1	5	7.2	1.30	7	7.2	1.48	6	9	3	0.26	-1.96	0.58
X18	8	a2	b1	c2	1	5	4.4	1.52	4	4.4	0.00	3	7	4	0.84	-1.12	0.68
X19	9	a1	b2	c2	1	5	6.0	2.55	6	6.0	1.48	3	10	7	0.43	-1.40	1.14
X110	10	a2	b2	c2	1	5	7.6	2.88	6	7.6	1.48	5	12	7	0.52	-1.73	1.29
X111	11	a1	b3	c2	1	5	7.0	1.00	7	7.0	1.48	6	8	2	0.00	-2.20	0.45
X112	12	a2	b3	c2	1	5	13.4	1.67	13	13.4	2.97	11	15	4	-0.25	-1.82	0.75

>

> (tAB <- describeBy(d4\$Y, list(d4\$A, d4\$B), mat=TRUE, digits=2))

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	a1	b1	1	10	7.5	1.18	8.0	7.50	1.48	6	9	3	-0.18	-1.67	0.37
X12	2	a2	b1	1	10	3.6	1.43	3.5	3.38	0.74	2	7	5	1.05	0.52	0.45
X13	3	a1	b2	1	10	7.7	2.83	7.5	7.88	3.71	3	11	8	-0.18	-1.66	0.90
X14	4	a2	b2	1	10	5.7	3.23	5.5	5.38	3.71	2	12	10	0.51	-1.03	1.02
X15	5	a1	b3	1	10	7.1	0.88	7.0	7.12	1.48	6	8	2	-0.16	-1.81	0.28
X16	6	a2	b3	1	10	13.2	1.69	13.0	13.25	2.97	11	15	4	-0.03	-1.85	0.53

>

> (tAC <- describeBy(d4\$Y, list(d4\$A, d4\$C), mat=TRUE, digits=2))

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	a1	c1	1	15	8.13	1.64	8	8.08	1.48	6	11	5	0.34	-1.04	0.42
X12	2	a2	c1	1	15	6.53	5.05	4	6.23	2.97	2	15	13	0.57	-1.50	1.31
X13	3	a1	c2	1	15	6.73	1.71	6	6.77	1.48	3	10	7	-0.09	-0.25	0.44
X14	4	a2	c2	1	15	8.47	4.32	7	8.38	4.45	3	15	12	0.25	-1.67	1.12

>

> (tBC <- describeBy(d4\$Y, list(d4\$B, d4\$C), mat=TRUE, digits=2))

	item	group1	group2	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	b1	c1	1	10	5.3	2.79	5.0	5.25	4.45	2	9	7	0.05	-1.94	0.88
X12	2	b2	c1	1	10	6.6	3.66	7.0	6.62	5.19	2	11	9	-0.07	-1.87	1.16
X13	3	b3	c1	1	10	10.1	3.35	9.5	10.00	3.71	6	15	9	0.27	-1.66	1.06
X14	4	b1	c2	1	10	5.8	1.99	6.0	5.75	2.97	3	9	6	0.09	-1.56	0.63
X15	5	b2	c2	1	10	6.8	2.70	6.0	6.62	1.48	3	12	9	0.59	-0.95	0.85
X16	6	b3	c2	1	10	10.2	3.61	9.5	10.12	5.19	6	15	9	0.13	-1.85	1.14

>

> (tA <- describeBy(d4\$Y, list(d4\$A), mat=TRUE, digits=2))

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	a1	1	30	7.43	1.79	7.5	7.38	2.22	3	11	8	0.05	-0.14	0.33
X12	2	a2	1	30	7.50	4.73	6.0	7.25	5.19	2	15	13	0.35	-1.52	0.86

>

> (tB <- describeBy(d4\$Y, list(d4\$B), mat=TRUE, digits=2))

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	b1	1	20	5.55	2.37	6.0	5.56	2.97	2	9	7	-0.03	-1.57	0.53
X12	2	b2	1	20	6.70	3.13	6.0	6.69	4.45	2	12	10	0.11	-1.34	0.70
X13	3	b3	1	20	10.15	3.39	9.5	10.06	3.71	6	15	9	0.22	-1.62	0.76

>

> (tC <- describeBy(d4\$Y, list(d4\$C), mat=TRUE, digits=2))

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	c1	1	30	7.33	3.78	8.0	7.17	4.45	2	15	13	0.22	-0.88	0.69
X12	2	c2	1	30	7.60	3.35	6.5	7.29	2.22	3	15	12	0.77	-0.40	0.61

>

> (tall <- describe(d4\$Y))

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	60	7.47	3.54	7	7.25	2.97	2	15	13	0.44	-0.55	0.46

>

```

>
> # 分散分析
> #aovを使う方法
> result.aov <- aov(Y ~ A*B*C + Error(id + id:A + id:B + id:C
+                               + id:A:B + id:A:C + id:B:C
+                               + id:A:B:C), data=d4)

> summary(result.aov)

Error: id
      Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4  9.933    2.483

Error: id:A
      Df Sum Sq Mean Sq F value Pr(>F)
A         1  0.067   0.0667   0.021  0.891
Residuals  4 12.600   3.1500

Error: id:B
      Df Sum Sq Mean Sq F value    Pr(>F)
B         2 229.23  114.62    30.3 0.000185 ***
Residuals  8  30.27    3.78

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:C
      Df Sum Sq Mean Sq F value Pr(>F)
C         1  1.067   1.067   0.292  0.617
Residuals  4 14.600   3.650

Error: id:A:B
      Df Sum Sq Mean Sq F value    Pr(>F)
A:B         2 282.0  141.02   60.01 1.53e-05 ***
Residuals  8  18.8    2.35

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:A:C
      Df Sum Sq Mean Sq F value Pr(>F)
A:C         1 41.67   41.67   5.882 0.0723 .
Residuals  4 28.33    7.08

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:B:C
      Df Sum Sq Mean Sq F value Pr(>F)
B:C         2  0.433   0.2167   0.394  0.687
Residuals  8  4.400   0.5500

Error: id:A:B:C
      Df Sum Sq Mean Sq F value Pr(>F)
A:B:C         2 29.63  14.817   3.13  0.099 .
Residuals  8 37.87    4.733

---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
>

> # Anovaを使う方法
> library(car)
> fac.a <- factor(c("a1", "a1", "a1", "a1", "a1", "a1", "a2", "a2", "a2", "a2", "a2", "a2"))
> fac.b <- factor(c("b1", "b1", "b2", "b2", "b3", "b3", "b1", "b1", "b2", "b2", "b3", "b3"))
> fac.c <- factor(c("c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2", "c1", "c2"))
> youin <- data.frame(fac.a, fac.b, fac.c)
> mod.1 <- lm(cbind(a1b1c1, a1b1c2, a1b2c1, a1b2c2, a1b3c1, a1b3c2,
+                  a2b1c1, a2b1c2, a2b2c1, a2b2c2, a2b3c1, a2b3c2) ~ 1, data=d1)
> result.Anova <- Anova(mod.1, idata=youin, idesign=~fac.a*fac.b*fac.c)

```

Note: model has only an intercept; equivalent type-III tests substituted.

```
> summary(result.Anova, multivariate=FALSE)
```

Univariate Type III Repeated-Measures ANOVA Assuming Sphericity

	SS	num Df	Error SS	den Df	F	Pr(>F)
(Intercept)	3345.1	1	9.933	4	1347.0067	3.291e-06 ***
fac.a	0.1	1	12.600	4	0.0212	0.8913695
fac.b	229.2	2	30.267	8	30.2952	0.0001851 ***
fac.c	1.1	1	14.600	4	0.2922	0.6174859
fac.a:fac.b	282.0	2	18.800	8	60.0071	1.525e-05 ***
fac.a:fac.c	41.7	1	28.333	4	5.8824	0.0723430 .
fac.b:fac.c	0.4	2	4.400	8	0.3939	0.6867896
fac.a:fac.b:fac.c	29.6	2	37.867	8	3.1303	0.0990406 .

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Mauchly Tests for Sphericity

	Test statistic	p-value
fac.b	0.57164	0.43220
fac.a:fac.b	0.41931	0.27152
fac.b:fac.c	0.97452	0.96202
fac.a:fac.b:fac.c	0.31090	0.17336

Greenhouse-Geisser and Huynh-Feldt Corrections  
for Departure from Sphericity

	GG eps	Pr(>F[GG])
fac.b	0.70011	0.0013655 **
fac.a:fac.b	0.63263	0.0004371 ***
fac.b:fac.c	0.97515	0.6822543
fac.a:fac.b:fac.c	0.59203	0.1400839

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

	HF eps	Pr(>F[HF])
fac.b	0.9618198	0.0002382353
fac.a:fac.b	0.7909991	0.0001021096
fac.b:fac.c	1.8908912	0.6867896092
fac.a:fac.b:fac.c	0.6960966	0.1281291458

警告メッセージ:

summary.Anova.mlm(result.Anova, multivariate = FALSE) で:

HF eps > 1 treated as 1

```
>
```

```
> #多重比較
```

```
> #ボンフェロニの方法
```

```
> (mc.bon.A <- pairwise.t.test(d4$Y, d4$A, p.adjust.method="bonferroni", paired=TRUE))
```

Pairwise comparisons using paired t tests

data: d4\$Y and d4\$A

```

a1
a2 0.95
```

P value adjustment method: bonferroni

```
> (mc.bon.B <- pairwise.t.test(d4$Y, d4$B, p.adjust.method="bonferroni", paired=TRUE))
```

```
Pairwise comparisons using paired t tests
```

```
data: d4$Y and d4$B
```

```
      b1      b2
b2 0.2106 -
b3 0.0034 0.0258
```

```
P value adjustment method: bonferroni
```

```
> (mc.bon.C <- pairwise.t.test(d4$Y, d4$C, p.adjust.method="bonferroni", paired=TRUE))
```

```
Pairwise comparisons using paired t tests
```

```
data: d4$Y and d4$C
```

```
      c1
c2 0.66
```

```
P value adjustment method: bonferroni
```

```
> # 効果量
```

```
> #eta_squared, omega_squared関数を使う方法
```

```
> library(effectsiz)
```

```
>
```

```
> eta_squared(result.aov, alternative="two.sided")
```

```
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Eta2 (partial)	95% CI
id:A	A	5.26e-03	[0.00, 0.43]
id:B	B	0.88	[0.59, 0.95]
id:C	C	0.07	[0.00, 0.60]
id:A:B	A:B	0.94	[0.77, 0.97]
id:A:C	A:C	0.60	[0.00, 0.86]
id:B:C	B:C	0.09	[0.00, 0.44]
id:A:B:C	A:B:C	0.44	[0.00, 0.73]

```
> omega_squared(result.aov, alternative="two.sided")
```

```
# Effect Size for ANOVA (Type I)
```

Group	Parameter	Omega2 (partial)	95% CI
id:A	A	0.00	[0.00, 0.00]
id:B	B	0.82	[0.39, 0.92]
id:C	C	0.00	[0.00, 0.00]
id:A:B	A:B	0.89	[0.60, 0.95]
id:A:C	A:C	0.42	[0.00, 0.79]
id:B:C	B:C	0.00	[0.00, 0.00]
id:A:B:C	A:B:C	0.25	[0.00, 0.61]

```
>
```

```
> eta_squared(result.Anova, alternative="two.sided")
```

```
# Effect Size for ANOVA (Type III)
```

Parameter	Eta2 (partial)	95% CI
fac.a	5.26e-03	[0.00, 0.43]
fac.a:fac.c	0.60	[0.00, 0.86]
fac.c	0.07	[0.00, 0.60]
fac.b	0.88	[0.59, 0.95]
fac.a:fac.b	0.94	[0.77, 0.97]
fac.b:fac.c	0.09	[0.00, 0.44]
fac.a:fac.b:fac.c	0.44	[0.00, 0.73]



```
> omega_squared(result.aov, alternative="two.sided")
# Effect Size for ANOVA (Type III)
```

Parameter	Omega2 (partial)	95% CI
fac. a	0.00	[0.00, 0.00]
fac. a:fac. c	0.42	[0.00, 0.79]
fac. c	0.00	[0.00, 0.00]
fac. b	0.82	[0.39, 0.92]
fac. a:fac. b	0.89	[0.60, 0.95]
fac. b:fac. c	0.00	[0.00, 0.00]
fac. a:fac. b:fac. c	0.25	[0.00, 0.61]

```
>
>
```

```
> #EtaSq関数を使う方法
```

```
> library(DescTools)
```

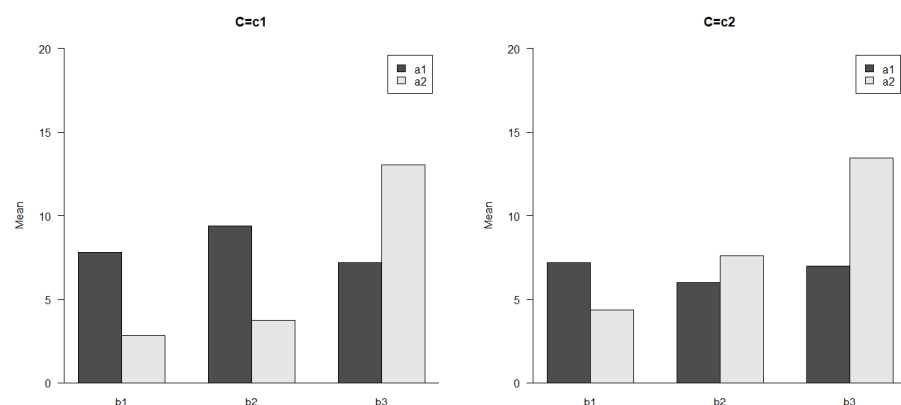
```
> EtaSq(result.aov, type=1)
```

	eta. sq	eta. sq. part	eta. sq. gen
A	8.997661e-05	0.005263158	0.0004249894
B	3.093846e-01	0.883365446	0.5938174596
C	1.439626e-03	0.068085106	0.0067567568
A:B	3.806460e-01	0.937506925	0.6426889480
A:C	5.623538e-02	0.595238095	0.2099428955
B:C	5.848479e-04	0.089655172	0.0027559890
A:B:C	3.999460e-02	0.439012346	0.1589486859

```
>
>
>
```

```
> #平均値を比較する棒グラフの作成
```

```
> d4t <- d4[d4$C=="c1",]
> m4t <- tapply(d4t[, "Y"], list(d4t[, "A"], d4t[, "B"]), mean)
> barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean", main="C=c1")
> segments(0, 0, 10, 0)
>
> d4t <- d4[d4$C=="c2",]
> m4t <- tapply(d4t[, "Y"], list(d4t[, "A"], d4t[, "B"]), mean)
> barplot(m4t, beside=TRUE, legend=TRUE, ylim=c(0, 20), las=1, ylab="Mean", main="C=c2")
> segments(0, 0, 10, 0)
>
```



## 10 分布の位置に関する推測 — ノンパラメトリック法 —

## 対応のある2群の分布の位置の比較 — ウィルコクソンの符号順位検定

`wilcox.test`(中央値を比較したい変数1, 中央値を比較したい変数2, `paired=TRUE`)

`paired=TRUE` を指定するのを忘れないこと。忘れると対応のない検定になってしまう。

```
> setwd("d:¥¥")
> d1 <- read.table("対応のある2群の分布位置データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> head(d1)
  番号 患者 家族
```

```
1    1    5    4
2    2    3    2
3    3    5    3
4    4    3    3
5    5    2    3
6    6    2    2
```

```
>
> dtmp <- d1[,c(-1)]
> d2 <- stack(dtmp)
>
>
```

```
> # 標本サイズ, 平均, SD, 相関係数
```

```
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD 患者 家族
患者 50 2.74 1.12 1.00 0.22
家族 50 3.18 1.02 0.22 1.00
>
>
```

```
> # 度数分布表
```

```
> t1 <- table(d1$患者)
> p1 <- prop.table(t1)*100
> rbind(t1, p1)
   1  2  3  4  5
t1  7 16 12 13  2
p1 14 32 24 26  4
> t1p <- t1
>
```

```
> t1 <- table(d1$家族)
> p1 <- prop.table(t1)*100
> rbind(t1, p1)
   1  2  3  4  5
t1  1 13 18 12  6
p1  2 26 36 24 12
> t1f <- t1
>
```

```
> # ウィルコクソンの符号順位和検定
```

```
> wilcox.test(d1$患者, d1$家族, paired=TRUE, exact=TRUE, conf.int=T)
```

Wilcoxon signed rank test with continuity correction

data: d1\$患者 and d1\$家族

V = 220.5, p-value = 0.02527

alternative hypothesis: true location shift is not equal to 0

95 percent confidence interval:

-1.000032e+00 -4.890761e-05

sample estimates:

(pseudo)median  
-0.5

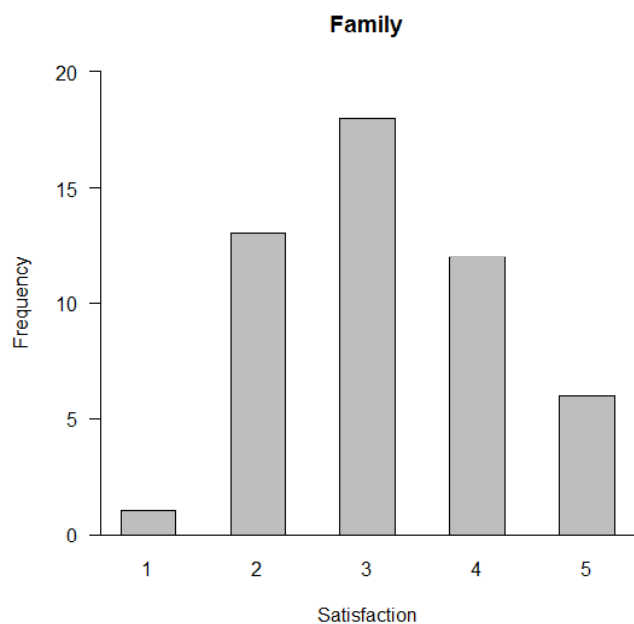
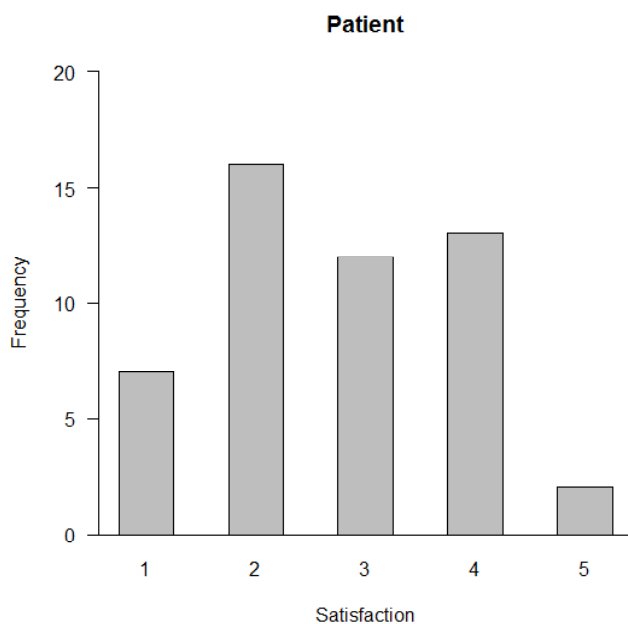
1	id	患者	家族
2	1	10	9
3	2	8	7
4	3	11	8
5	4	8	8
6	5	7	8
7	6	7	7
8	7	9	8
9	8	7	7
10	9	8	8
11	10	6	8
12	11	9	10
13	12	7	10
14	13	6	7
15	14	9	9
16	15	8	9
17	16	7	7
18	17	7	8
19	18	9	7
20	19	9	8
21	20	9	8

警告メッセージ:

```
1: In wilcox.test.default(d1$患者, d1$家族, paired = TRUE, exact = TRUE, :
  タイがあるため、正確な p 値を計算することができません
2: In wilcox.test.default(d1$患者, d1$家族, paired = TRUE, exact = TRUE, :
  タイがあるので、正確な信頼区間を計算することができません
3: In wilcox.test.default(d1$患者, d1$家族, paired = TRUE, exact = TRUE, :
  ゼロ値のため、正確な p 値を計算することができません
4: In wilcox.test.default(d1$患者, d1$家族, paired = TRUE, exact = TRUE, :
  ゼロ値のため、正確な信頼区間を計算することができません
>
>
```

> # 各群の分布

```
> barplot(t1p, ylim=c(0,20), width=0.7, space=1, xlab="Satisfaction",
+   ylab="Frequency", main="Patient", las=1)
> segments(0,0, 10,0)
>
> barplot(t1f, ylim=c(0,20), width=0.7, space=1, xlab="Satisfaction",
+   ylab="Frequency", main="Family", las=1)
> segments(0,0, 10,0)
>
```



>

## 対応のある多群の分布の位置の比較 — フリードマンの検定

```
friedman.test(as.matrix(データフレーム名))
```

as.matrix でデータ形式をmatrix形式にする必要がある。

```
> setwd("d:¥¥")
> d1 <- read.table("対応のある多群の分布位置データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  番号 お使い下さい 迷惑 ありがとう
1    1             5    5          4
2    2             4    4          3
3    3             4    5          4
4    4             4    4          4
5    5             4    3          3
6    6             5    4          3
>
> dtmp <- d1[,c(-1)]
> d2 <- stack(dtmp)
>
>
```

```
> # 標本サイズ, 平均, SD, 相関係数
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean   SD お使い下さい 迷惑 ありがとう
お使い下さい 129 5.03 1.14          1.00 0.24          0.31
迷惑          129 4.61 1.31          0.24 1.00          0.22
ありがとう    129 4.05 1.41          0.31 0.22          1.00
>
```

1	番号	お使い下さい	迷惑	ありがとう
2	1	5	5	4
3	2	4	4	3
4	3	4	5	4
5	4	4	4	4
6	5	4	3	3
7	6	5	4	3
8	7	6	5	5
9	8	7	7	5
10	9	5	5	2
11	10	5	6	4
12	11	5	6	4
13	12	3	2	6
14	13	6	6	4
15	14	6	5	3
16	15	5	5	2
17	16	5	4	5
18	17	7	4	6
19	18	5	6	5
20	19	6	5	4
21	20	5	5	4

```
> # 度数分布表
> t1 <- table(d1$お使い下さい)
> p1 <- prop.table(t1)*100
> round(rbind(t1,p1), 2)
      2    3    4    5    6    7
t1 1.00 12.0 25.00 49.00 28.00 14.00
p1 0.78  9.3 19.38 37.98 21.71 10.85
```

```
> t1c <- rbind(c(1:7),c(0, t1))
> colnames(t1c) <- c(1:7)
```

# カテゴリ 1 の人数が 0 人だが、グラフにこのカテゴリも  
# 含めるための応急処置

```
> t1 <- table(d1$迷惑)
> p1 <- prop.table(t1)*100
> round(rbind(t1,p1), 2)
      1    2    3    4    5    6    7
t1 1.00 5.00 21.00 32.00 36.00 25.00 9.00
p1 0.78 3.88 16.28 24.81 27.91 19.38 6.98
> t1i <- t1
```

```
> t1 <- table(d1$ありがとう)
> p1 <- prop.table(t1)*100
> round(rbind(t1,p1), 2)
      1    2    3    4    5    6    7
t1 1.00 15.00 35.00 34.00 20.0 17.00 7.00
p1 0.78 11.63 27.13 26.36 15.5 13.18 5.43
> t1t <- t1
>
```

```
> #フリードマンの検定
> friedman.test(as.matrix(dtmp), conf.int=T)
```

```
Friedman rank sum test
```

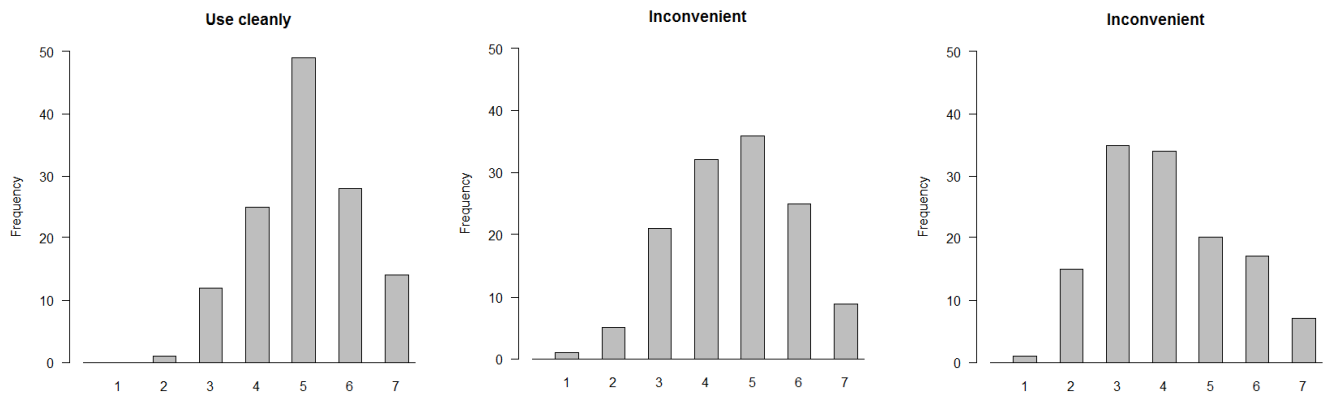
```
data: as.matrix(dtmp)
Friedman chi-squared = 49.14, df = 2, p-value = 2.135e-11
```

```
>
```

```
> #各群の分布
> barplot(t1c[2,], ylim=c(0,50), xlim=c(0,10), width=0.7, space=1, xlab="",
+   ylab="Frequency", main="Use cleanly", las=1)
> segments(0,0, 10,0)
>
> barplot(t1i, ylim=c(0,50), xlim=c(0,10), width=0.7, space=1, xlab="",
+   ylab="Frequency", main="Inconvenient", las=1)
> segments(0,0, 10,0)
>
> barplot(t1t, ylim=c(0,50), xlim=c(0,10), width=0.7, space=1, xlab="",
+   ylab="Frequency", main="Inconvenient", las=1)
> segments(0,0, 10,0)
```

```
>
```

```
>
```



## 対応のない2群の分布の位置の比較 — ウィルコクソンの順位和検定(マン・ホイットニーの検定)

`wilcox.test`(中央値を比較したい変数 ~ 群分け変数, データフレーム名) または  
`wilcox.test`(中央値を比較したい変数1, 中央値を比較したい変数2, `paired=FALSE`)

`paired=FALSE` は省略可.

```
> setwd("d:\\Y")
> d1 <- read.table("対応のない2群の分布位置データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
```

```
  番号  居住形態  バイト量
1     1     自宅         1
2     2     自宅         2
3     3     自宅         3
4     4     自宅         4
5     5     自宅         3
6     6     自宅         2
```

```
>
>
```

```
> # 度数分布表
> (t1 <- table(d1$居住形態, d1$バイト量))
```

```
      1  2  3  4  5
自宅   5 11 20 23 20
自宅外 4  3 14 25 20
```

```
>
```

```
> (p1 <- round(prop.table(t1, 1), 2))
```

```
      1  2  3  4  5
自宅  0.06 0.14 0.25 0.29 0.25
自宅外 0.06 0.05 0.21 0.38 0.30
```

```
>
>
>
```

```
> # 各群の記述統計量
> library(psych)
> describeBy(d1$バイト量, d1$居住形態)
group: 自宅
```

```
vars  n mean  sd median trimmed  mad min max range  skew kurtosis  se
1     1 79 3.53 1.2      4    3.62 1.48   1  5    4 -0.43   -0.77 0.13
```

```
-----
group: 自宅外
```

```
vars  n mean  sd median trimmed  mad min max range  skew kurtosis  se
1     1 66 3.82 1.11      4    3.96 1.48   1  5    4 -0.91    0.32 0.14
```

```
>
>
```

```
> #ウィルコクソンの順位和検定の実行
> wilcox.test(バイト量~居住形態, data=d1)
```

Wilcoxon rank sum test with continuity correction

```
data: バイト量 by 居住形態
W = 2241, p-value = 0.1321
alternative hypothesis: true location shift is not equal to 0
```

```
>
>
```

```
> #群ごとのデータフレームを作成
> d11 <- d1[d1$居住形態=="自宅",]
> d12 <- d1[d1$居住形態=="自宅外",]
>
```

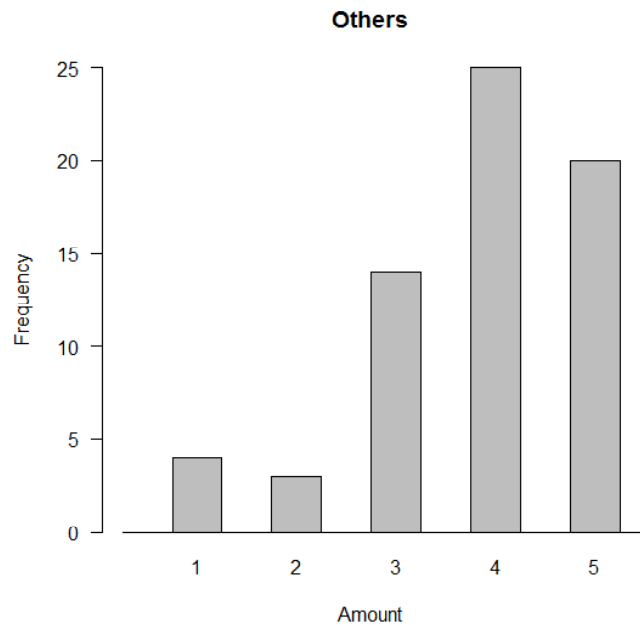
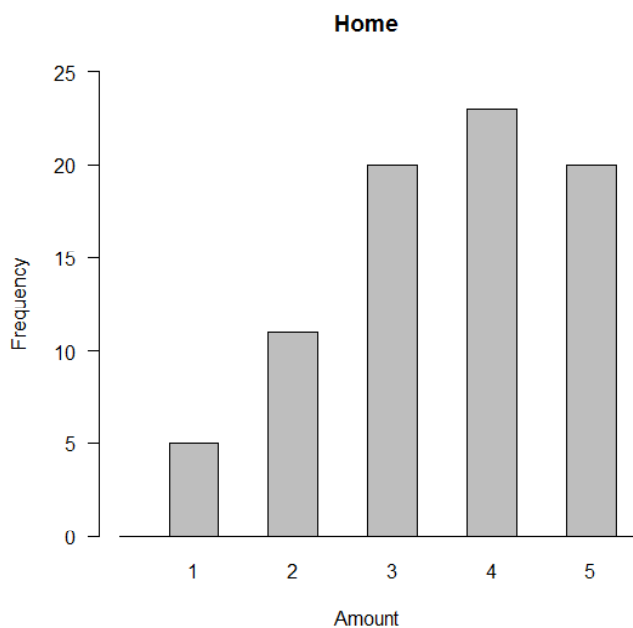
1	番号	居住形態	バイト量
2	1	自宅	5
3	2	自宅外	2
4	3	自宅外	3
5	4	自宅	3
6	5	自宅外	4
7	6	自宅外	5
8	7	自宅外	1
9	8	自宅	4
10	9	自宅	3
11	10	自宅	4
12	11	自宅	3
13	12	自宅	5
14	13	自宅	3
15	14	自宅外	4
16	15	自宅外	4
17	16	自宅外	3
18	17	自宅	5
19	18	自宅外	4
20	19	自宅外	5
21	20	自宅外	4

```
> #ウィルコクソンの順位和検定の実行
> wilcox.test(d11$バイト量, d12$バイト量, paired=FALSE)
```

Wilcoxon rank sum test with continuity correction

```
data: d11$バイト量 and d12$バイト量
W = 2241, p-value = 0.1321
alternative hypothesis: true location shift is not equal to 0
```

```
> # 各群の分布
> d1.v <- d1[d1$居住形態=="自宅",]
> d1.n <- d1[d1$居住形態=="自宅外",]
>
> t1h <- table(d1.v$バイト量)
> t1o <- table(d1.n$バイト量)
>
> barplot(t1h, ylim=c(0, 25), xlim=c(0, 7), width=0.7, space=1, xlab="Amount",
+   ylab="Frequency", main="Home", las=1)
> segments(0, 0, 10, 0)
>
> barplot(t1o, ylim=c(0, 25), xlim=c(0, 7), width=0.7, space=1, xlab="Amount",
+   ylab="Frequency", main="Others", las=1)
> segments(0, 0, 10, 0)
>
>
```



```
>
>
```



## 対応のない多群の分布の位置の比較 — クラスカル・ウォリスの検定

`kruskal.test`(中央値を比較したい変数 ~ 群分け変数, データフレーム名)      または  
`kruskal.test`(データフレーム名\$中央値を比較したい変数, データフレーム名\$群分け変数)

```
> setwd("d:¥¥")
> d1 <- read.table("対応のない多群の分布位置データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  番号 区分 嫌悪感
1     1 喫煙      1
2     2 禁煙      2
3     3 卒煙      5
4     4 喫煙      3
5     5 喫煙      1
6     6 喫煙      5
```

1	番号	区分	嫌悪感
2	1	喫煙	1
3	2	禁煙	2
4	3	卒煙	5
5	4	喫煙	3
6	5	喫煙	1
7	6	喫煙	5
8	7	禁煙	4
9	8	喫煙	2
10	9	禁煙	1
11	10	卒煙	1
12	11	卒煙	3
13	12	喫煙	2
14	13	卒煙	2
15	14	卒煙	2
16	15	喫煙	3
17	16	禁煙	2
18	17	禁煙	4
19	18	卒煙	4
20	19	卒煙	4
21	20	卒煙	4

```
> # 度数分布表
> (t1 <- table(d1$区分, d1$嫌悪感))
```

```
      1  2  3  4  5
喫煙  7 10  7  3  2
禁煙  6  6  4  5  7
卒煙  4  5  8  9  7
```

```
> (p1 <- round(prop.table(t1, 1), 2))
```

```
      1    2    3    4    5
喫煙 0.24 0.34 0.24 0.10 0.07
禁煙 0.21 0.21 0.14 0.18 0.25
卒煙 0.12 0.15 0.24 0.27 0.21
```

```
> # 各群の記述統計量
> library(psych)
> describeBy(d1$嫌悪感, d1$区分)
```

```
group: 喫煙
  vars  n mean   sd median trimmed  mad min max range skew kurtosis  se
1     1 29 2.41 1.18      2    2.32 1.48   1  5    4 0.58   -0.56 0.22
```

```
group: 禁煙
  vars  n mean   sd median trimmed  mad min max range skew kurtosis  se
1     1 28 3.04 1.53      3    3.04 1.48   1  5    4  0   -1.57 0.29
```

```
group: 卒煙
  vars  n mean   sd median trimmed  mad min max range skew kurtosis  se
1     1 33  3.3 1.31      3    3.37 1.48   1  5    4 -0.31  -1.08 0.23
```

```
> # クラスカル・ウォリスの検定
> kruskal.test(嫌悪感 ~ 区分, data=d1)
```

Kruskal-Wallis rank sum test

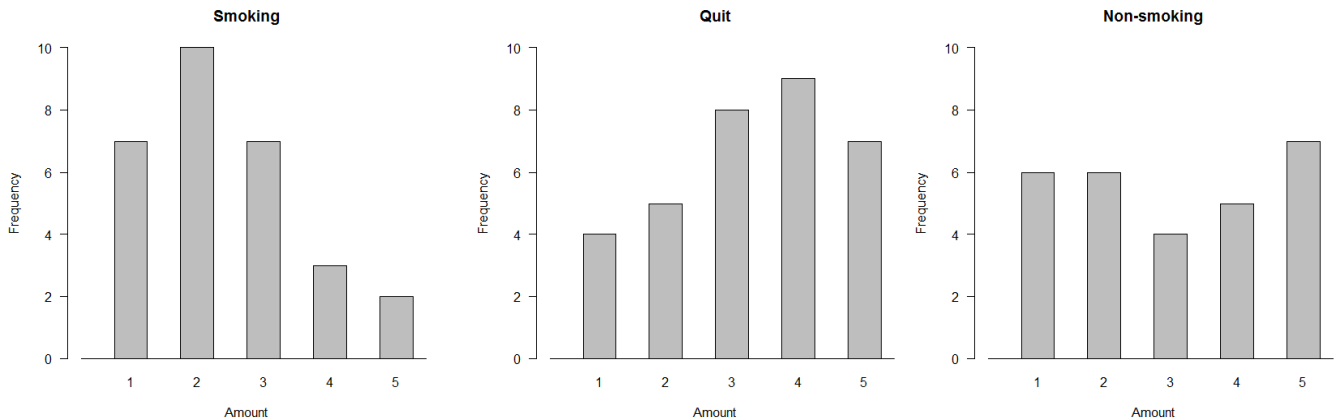
```
data: 嫌悪感 by 区分
Kruskal-Wallis chi-squared = 6.6299, df = 2, p-value = 0.03634
```

```
> # 各群の分布
> d1k <- d1[d1$区分=="喫煙",]
> d1s <- d1[d1$区分=="卒煙",]
> d1n <- d1[d1$区分=="禁煙",]
>
```

```

> t1s <- table(d1k$嫌悪感)
> t1q <- table(d1s$嫌悪感)
> t1n <- table(d1n$嫌悪感)
>
> barplot(t1s, ylim=c(0,10), xlim=c(0,7), width=0.7, space=1, xlab="Amount",
+   ylab="Frequency", main="Smoking", las=1)
> segments(0,0, 10,0)
>
> barplot(t1q, ylim=c(0,10), xlim=c(0,7), width=0.7, space=1, xlab="Amount",
+   ylab="Frequency", main="Quit", las=1)
> segments(0,0, 10,0)
>
> barplot(t1n, ylim=c(0,10), xlim=c(0,7), width=0.7, space=1, xlab="Amount",
+   ylab="Frequency", main="Non-smoking", las=1)
> segments(0,0, 10,0)
>

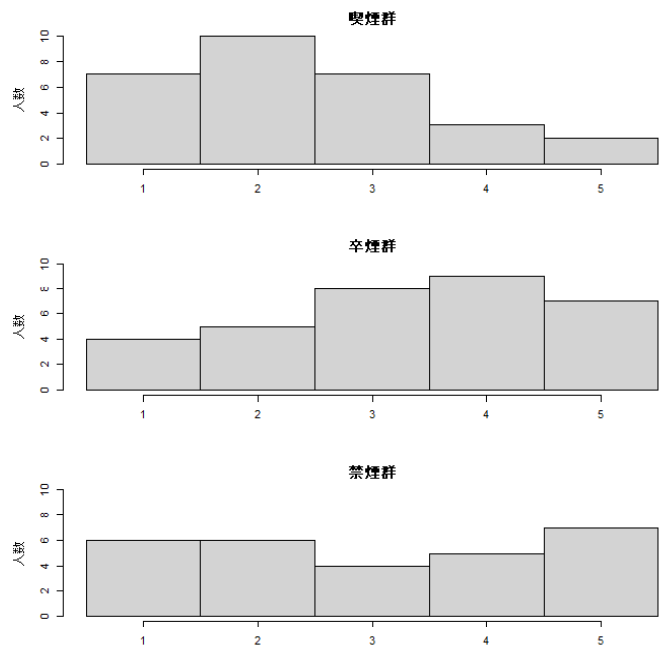
```



```

>
> #3つのグラフを縦横に並べて描く設定
> par(mar=c(5,4,2,1))
> par(mfrow=c(3,1))
> hist(d1k$嫌悪感, breaks=seq(0.5,5.5,1), ylim=c(0,10), ylab="人数", main="喫煙群", xlab="")
> hist(d1s$嫌悪感, breaks=seq(0.5,5.5,1), ylim=c(0,10), ylab="人数", main="卒煙群", xlab="")
> hist(d1n$嫌悪感, breaks=seq(0.5,5.5,1), ylim=c(0,10), ylab="人数", main="禁煙群", xlab="")
> par(mfrow=c(1,1))

```



## 11 相関係数に関する推測

## 相関係数の推測

```
cor.test(変数名1, 変数名2, method="算出方法")
```

```
method: "pearson" (default), "spearman", or "kendall"
```

```
> setwd("d:¥¥")
> d1 <- read.table("相関係数データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  番号 ストレス うつ傾向
1     1      20      18
2     2      23      21
3     3      30      29
4     4      25      29
5     5      26      22
6     6      21      19
```

```
>
>
> dtmp <- d1[,c(-1)]
>
>
> #記述統計量
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD ストレス うつ傾向
ストレス 154 23.39 5.14      1.00      0.62
うつ傾向 154 20.45 6.71      0.62      1.00
```

```
>
>
> #相関係数の検定 (H0:  $\rho = 0$ )
> cor.test(d1$ストレス, d1$うつ傾向, method="pearson")
```

Pearson's product-moment correlation

```
data: d1$ストレス and d1$うつ傾向
t = 9.6234, df = 152, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.5064286 0.7048699
sample estimates:
      cor
0.6153051
```

```
>
> #スピアマンの順位相関係数の検定
> cor.test(d1$ストレス, d1$うつ傾向, method="spearman")
```

Spearman's rank correlation rho

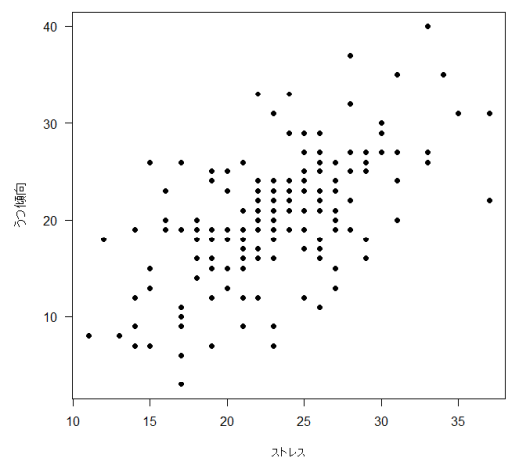
```
data: d1$ストレス and d1$うつ傾向
S = 239103.1, p-value < 2.2e-16
alternative hypothesis: true rho is not equal to 0
sample estimates:
      rho
0.6071809
```

警告メッセージ:

```
In cor.test.default(d1$ストレス, d1$うつ傾向, method = "spearman") :
  タイのため正確な p 値を計算することができません
```

```
>
>
> # 散布図
> plot(d1$ストレス, d1$うつ傾向, xlab="ストレス", ylab="うつ傾向", las=1, pch=19)
```

	A	B	C
1	番号	ストレス	うつ傾向
2	1	20	18
3	2	23	21
4	3	30	29
5	4	25	29
6	5	26	22
7	6	21	19
8	7	14	12
9	8	22	19
10	9	26	27
11	10	26	18
12	11	21	18
13	12	24	29
14	13	24	19
15	14	23	24
16	15	23	20
17	16	35	31
18	17	25	19
19	18	33	26
20	19	30	30
21	20	19	12



## 相関係数行列の各要素の検定

```
library(psych)
corr.test(データフレーム名, method="算出方法")

method: "pearson" (default), "spearman", or "kendall"
```

```
> setwd("d:¥¥")
> d1 <- read.table("データ例.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
```

```
  id x1 x2 x3 x4
1  1 23 28 23 17
2  2 18 22 23 14
3  3 15 12 15 13
4  4 12 16 22 25
5  5  5 24 13 28
6  6 14 16 15 16
```

	A	B	C	D	E
id	x1	x2	x3	x4	
1	23	28	23	17	
2	18	22	23	14	
3	15	12	15	13	
4	12	16	22	25	
5	5	24	13	28	
6	14	16	15	16	
7	17	29	8	22	
8	17	11	25	6	
9	24	11	28	8	
10	14	14	18	10	
11	21	14	22	13	
12	21	18	13	9	
13	16	11	19	13	
14	19	26	18	21	

```
> dtmp <- d1[,c(-1)]
```

## &gt; #記述統計量

```
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
  N Mean  SD  x1  x2  x3  x4
x1 245 15.22 5.37 1.00 -0.10 0.43 -0.26
x2 245 20.32 6.08 -0.10 1.00 -0.32 0.56
x3 245 18.52 5.12 0.43 -0.32 1.00 -0.46
x4 245 16.61 6.98 -0.26 0.56 -0.46 1.00
```

## &gt; # 相関係数行列の各要素の検定

```
> library(psych)
> (drl <- corr.test(dtmp))
Call:corr.test(x = dtmp)
Correlation matrix
      x1    x2    x3    x4
x1  1.00 -0.10  0.43 -0.26
x2 -0.10  1.00 -0.32  0.56
x3  0.43 -0.32  1.00 -0.46
x4 -0.26  0.56 -0.46  1.00
Sample Size
[1] 245
Probability values (Entries above the diagonal are adjusted for multiple tests.)
      x1    x2    x3    x4
x1 0.00 0.11  0  0
x2 0.11 0.00  0  0
x3 0.00 0.00  0  0
x4 0.00 0.00  0  0
```

To see confidence intervals of the correlations, print with the short=FALSE option

## &gt; # t値

```
> drl$t
      x1    x2    x3    x4
x1      Inf -1.620965  7.329962 -4.182839
x2 -1.620965      Inf -5.323804 10.559251
x3  7.329962 -5.323804      Inf -8.031193
x4 -4.182839 10.559251 -8.031193      Inf
```

## 2群の相関係数の差の検定

```
library(psych)
```

独立な2群における相関係数の差(r12 vs r34)

```
r.test(r12=.3, r34=.5, n=100, n2=150)
```

対応のある2つの相関係数の差(r12 vs r34)

```
r.test(r12=.3, r34=.5, r13=.1, r14=.1, r23=.1, r24=.1, n=150)
```

1つの変数を共有する, 対応のある2つの相関係数の差(r12 vs r13)

```
r.test(r12=.3, r13=.5, r23=.1, n=150)
```

あらかじめpsychパッケージをインストールしておく必要がある.

```
> # r.test を用いる方法
```

```
> library(psych)
```

```
>
```

```
> # 独立な2群における相関係数の差(r12 vs r34)
```

```
> r.test(r12=.3, r34=.5, n=100, n2=150)
```

```
Correlation tests
```

```
Call:r.test(n = 100, r12 = 0.3, r34 = 0.5, n2 = 150)
```

```
Test of difference between two independent correlations
```

```
z value 1.83 with probability 0.07>
```

```
>
```

```
> # 対応のある2つの相関係数の差(r12 vs r34)
```

```
> r.test(r12=.3, r34=.5, r13=.1, r14=.1, r23=.1, r24=.1, n=150)
```

```
Correlation tests
```

```
Call:r.test(n = 150, r12 = 0.3, r34 = 0.5, r23 = 0.1, r13 = 0.1, r14 = 0.1,
r24 = 0.1)
```

```
Test of difference between two dependent correlations
```

```
z value -2.07 with probability 0.04>
```

```
>
```

```
> # 1つの変数を共有する, 対応のある2つの相関係数の差(r12 vs r13)
```

```
> r.test(r12=.3, r13=.5, r23=.1, n=150)
```

```
Correlation tests
```

```
Call:r.test(n = 150, r12 = 0.3, r23 = 0.1, r13 = 0.5)
```

```
Test of difference between two correlated correlations
```

```
t value -2.09 with probability < 0.038>
```

```
>
```

```
>
```

## 2群の相関係数の差の信頼区間

## 自作関数を使う

cor2.dif(n=人数ベクトル, r=相関係数ベクトル)

```
# -----
cor2.dif <- function(n, r, conf=0.95) {
  z <- 1/2 * log((1+r)/(1-r))
  d.r <- r %>% c(1, -1)
  d.z <- z %>% c(1, -1)
  n3 <- n - 3
  inv.n3 <- 1/ n3
  se <- sqrt(sum(inv.n3))
  p.value <- round(2 * (1 - pnorm(abs(d.z/se))), 4)
  za <- qnorm(1-(1-confidence)/2)
  ci.l <- d.z - za * se
  ci.u <- d.z + za * se
  exp.2z.l <- exp(2*ci.l)
  exp.2z.u <- exp(2*ci.u)
  ci.l <- round((exp.2z.l-1)/(exp.2z.l+1), 4)
  ci.u <- round((exp.2z.u-1)/(exp.2z.u+1), 4)
  title <- "Difference of two correlation coefficients"
  stnames <- c("difference", "z", "p.value", paste("lower(", conf, ")", sep=""), paste("upper(", conf, ")", sep=""))
  statistic <- data.frame(d.r, round(d.z/se, 4), p.value, ci.l, ci.u)
  ts <- r/sqrt(1-r*r)*sqrt(n-2)
  ps <- 2*(1-pt(abs(ts), n-2))
  rhoTests <- round(data.frame(n, r, ts, ps), 4)
  colnames(rhoTests) <- c("N", "r", "t", "p")
  colnames(statistic) <- stnames
  out <- list(title, rhoTest=rhoTests, statistic=statistic)
  return(out)
}
# -----
```

各群における無相関検定と、相関係数の差の検定、信頼区間を出力する。  
conf= で信頼係数を調整できる。デフォルトは0.95.

```
> setwd("d:¥¥")
> d1 <- read.table("相関係数の差データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
```

```
  id shurou honnin kyouin
1  1      0     29     10
2  2      0     28     17
3  3      0     24     24
4  4      0     27     26
5  5      0     30     34
6  6      0     30     30
```

```
>
> d10 <- d1[d1$shurou==0, ]
> d11 <- d1[d1$shurou==1, ]
>
```

## &gt; # 記述統計量

```
> # shurou==0
> dtmp <- d10[, c(-1, -2)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD honnin kyouin
honnin 117 24.48 5.89   1.00   0.18
kyouin 117 23.38 5.76   0.18   1.00
```

1	id	shurou	honnin	kyouin
2	1	0	29	10
3	2	0	28	17
4	3	0	24	24
5	4	0	27	26
6	5	0	30	34
7	6	0	30	30
8	7	0	20	25
9	8	0	23	23
10	9	0	12	16
11	10	0	25	29
12	11	0	15	20
13	12	0	21	18
14	13	0	17	24
15	14	0	29	30
16	15	0	24	16
17	16	0	35	23
18	17	0	22	24
19	18	0	29	10
20	19	0	29	27
21	20	0	28	22

```

> #shurou==1
> dtmp <- d11[,c(-1,-2)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD honnin kyouin
honnin 153 30.24 5.59    1.0    0.3
kyouin 153 31.78 5.77    0.3    1.0
>
>
>
> # 相関係数の差の検定・信頼区間
> nv <- c(nrow(d10), nrow(d11))
> rv <- c(cor(d10$honnin, d10$kyouin), cor(d11$honnin, d11$kyouin))
> cor2.dif(n=nv, r=rv)

```

```

[[1]]
[1] "Difference of two correlation coefficients"

```

```

$rhoTest
      N      r      t      p
1 117 0.1779 1.9384 0.0550
2 153 0.3016 3.8873 0.0002

```

```

$statistic
      difference      z p.value lower(0.95) upper(0.95)
1 -0.123742 -1.0584  0.2899   -0.3584      0.1116

```

```

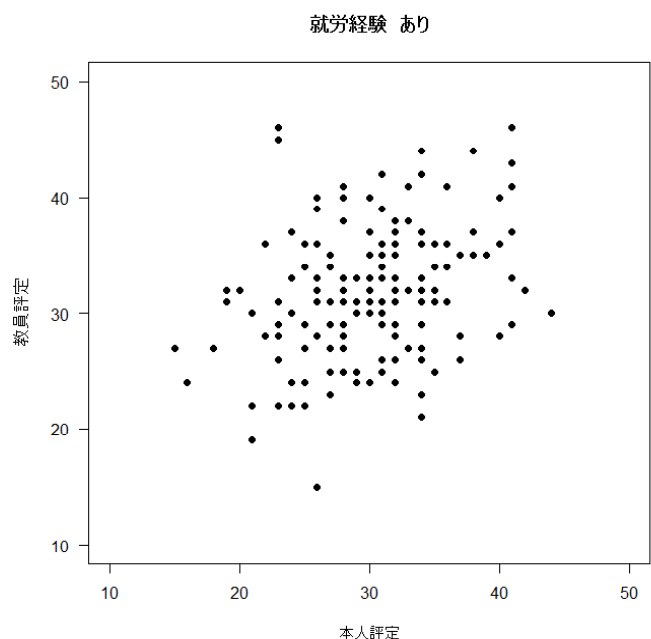
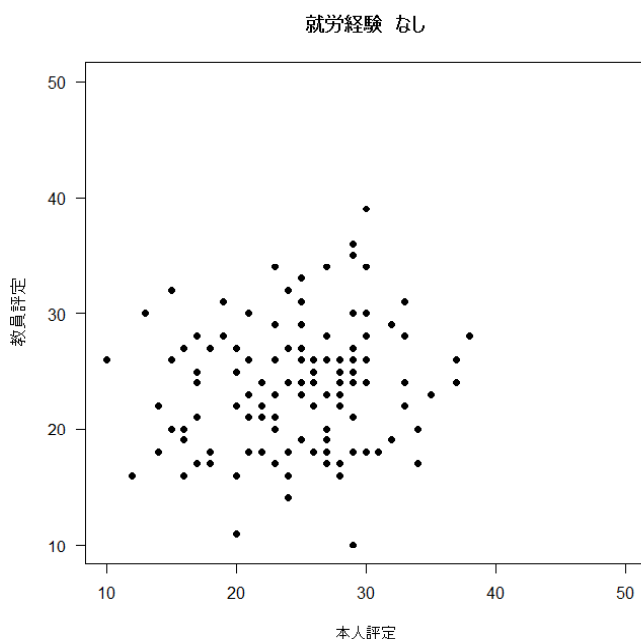
>

```

```

> # 散布図
> plot(d10$honnin, d10$kyouin, xlim=c(10,50), ylim=c(10,50),
+ main="就労経験 なし", xlab="本人評価", ylab="教員評価", las=1, pch=19)
>
> plot(d11$honnin, d11$kyouin, xlim=c(10,50), ylim=c(10,50),
+ main="就労経験 あり", xlab="本人評価", ylab="教員評価", las=1, pch=19)
>

```





## 多群の相関係数の等質性の推測

## 自作関数を使う

`cors.test`(`n`=人数ベクトル, `r`=相関係数ベクトル, `conf`=信頼係数)

```
# -----
cors.test<- function(n, r, conf=0.95) {
  z <- 1/2 * log((1+r)/(1-r))
  n3 <- n - 3
  df <- length(n) - 1
  X2 <- round(sum(z*z*n3) - sum(z*n3)^2 / sum(n3), 4)
  p.value <- round(1 - pchisq(X2, df), 5)
  title <- "Test : equality of correlation coefficients"
  statistic <- round(cbind(X2, df, p.value), 4)
  ts <- r/sqrt(1-r*r)*sqrt(n-2)
  ps <- 2*(1-pt(abs(ts), n-2))
  rhoTests <- round(data.frame(n, r, ts, ps), 4)
  colnames(rhoTests) <- c("N", "r", "t", "p")
  colnames(statistic) <- c("X2", "df", "p")
  rownames(statistic) <- ""
  inv.n3 <- 1/ n3
  npair <- length(n)*(length(n)-1)
  za <- qnorm(1-((1-conf)/(npair/2))/2)
  pairs <- matrix(c(0), npair, 7)
  k <- 0
  for(i in 1:length(n)) {
    for(j in c(1:length(n))[-i]) {
      d.r <- r[i] - r[j]
      d.z <- z[i] - z[j]
      se <- sqrt(sum(inv.n3[i]+inv.n3[j]))
      p.value <- round(2 * (1 - pnorm(abs(d.z/se))), 4)
      ci.l <- d.z - za * se
      ci.u <- d.z + za * se
      exp.2z.l <- exp(2*ci.l)
      exp.2z.u <- exp(2*ci.u)
      ci.l <- round((exp.2z.l-1)/(exp.2z.l+1), 4)
      ci.u <- round((exp.2z.u-1)/(exp.2z.u+1), 4)
      k <- k+1
      pairs[k,] <- round(c(r[i], r[j], d.r, round(d.z/se, 4), p.value, ci.l, ci.u), 4)
    }
  }
  rownames(pairs) <- c(1:npair)
  jconf <- round(100*(1-((1-conf)/(npair/2))), 1)
  colnames(pairs) <- c("r1", "r2", "diff", "z", "p", paste("lower(", jconf, ")", sep=""), paste("upper(",
jconf, ")", sep=""))
  out <- list(title, rhoTest=rhoTests, statistic= statistic, CI= pairs)
  return(out)
}
# -----
```

各群における無相関検定と、相関係数の等質性の検定、同時信頼区間を出力する。  
`conf=` で信頼係数を調整できる。デフォルトは0.95。

```
> setwd("d:¥¥¥")
> d1 <- read.table("相関係数の等質性データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id grade ruiji shinmitsu
1  1     L      5         4
2  2     L      5         4
3  3     L      3         2
4  4     L      2         3
5  5     L      3         3
6  6     L      5         6
>
```

```

> # 群別データの作成
> d10 <- d1[d1$grade=="L", ]
> d11 <- d1[d1$grade=="M", ]
> d12 <- d1[d1$grade=="H", ]
>
> #記述統計量
> #grade=="L"
> dtmp <- d10[, c(-1, -2)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD ruiji shinmitsu
ruiji  92 3.98 1.25  1.00      0.22
shinmitsu 92 3.96 1.14  0.22      1.00
>
> #grade=="M"
> dtmp <- d11[, c(-1, -2)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD ruiji shinmitsu
ruiji 108 3.60 1.27  1.00      0.34
shinmitsu 108 3.73 1.34  0.34      1.00
>
> #grade=="H"
> dtmp <- d12[, c(-1, -2)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD ruiji shinmitsu
ruiji 102 3.47 1.36  1.00      0.52
shinmitsu 102 3.50 1.33  0.52      1.00
>
> # 相関係数の等質性
> n1 <- nrow(d10)
> n2 <- nrow(d11)
> n3 <- nrow(d12)
> nv <- c(n1, n2, n3)
> r1 <- cor(d10$ruiji, d10$shinmitsu)
> r2 <- cor(d11$ruiji, d11$shinmitsu)
> r3 <- cor(d12$ruiji, d12$shinmitsu)
> rv <- c(r1, r2, r3)
>
> cors.test(n=nv, r=rv)
[[1]]
[1] "Test : equality of correlation coefficients"

$rhoTest
      N      r      t      p
1  92 0.2158 2.0970 0.0388
2 108 0.3425 3.7534 0.0003
3 102 0.5209 6.1017 0.0000

```

1	id	grade	ruiji	shinmitsu
2	1	L	5	4
3	2	L	5	4
4	3	L	3	2
5	4	L	2	3
6	5	L	3	3
7	6	L	5	6
8	7	L	3	3
9	8	L	5	6
10	9	L	3	4
11	10	L	3	4
12	11	L	4	6
13	12	L	4	2
14	13	L	4	2
15	14	L	5	3
16	15	L	3	5
17	16	L	5	3
18	17	L	5	3
19	18	L	4	3
20	19	L	3	4
21	20	L	3	4

```
$statistic
      X2 df      p
6.1903  2 0.0453
```

```
$CI
      r1      r2      diff      z      p lower(98.3) upper(98.3)
1 0.2158 0.3425 -0.1267 -0.9554 0.3394      -0.4483      0.2044
2 0.2158 0.5209 -0.3050 -2.4526 0.0142      -0.6094     -0.0086
3 0.3425 0.2158  0.1267  0.9554 0.3394      -0.2044      0.4483
4 0.3425 0.5209 -0.1784 -1.5747 0.1153      -0.5050      0.1143
5 0.5209 0.2158  0.3050  2.4526 0.0142       0.0086      0.6094
6 0.5209 0.3425  0.1784  1.5747 0.1153      -0.1143      0.5050
```

```
>
```

# 12 分割表に関する推測

## — $\chi^2$ 検定・フィッシャーの正確検定 —

## 2 × 2 表 — ファイ係数, カイ2乗検定, 尤度比検定, フィッシャーの正確検定

クロス表の出力とともに検定結果を表示

```
library(descr)
CrossTable(変数1, 変数2, chisq=T, fisher=T)
```

あらかじめdescrパッケージをインストールしておく必要がある。  
ピアソンのカイ2乗検定, イエーツの連続修正カイ2乗検定, フィッシャーの正確検定を行う。

ファイ係数

```
library(effectsize)
phi(テーブル名)
```

あらかじめeffectsizeパッケージをインストールしておく必要がある。

イエーツの連続修正を行ったカイ2乗検定

```
chisq.test(テーブル名)
```

尤度比検定・ピアソンのカイ2乗検定

```
library(vcd)
assocstats(テーブル名)
```

あらかじめvcdパッケージをインストールしておく必要がある。

フィッシャーの正確検定

```
fisher.test(テーブル名)
```

```
> setwd("d:¥¥¥")
> d1 <- read.table("ファイ係数データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  番号 通塾 睡眠不足
1     1    1         0
2     2    1         1
3     3    1         1
4     4    0         1
5     5    1         1
6     6    1         1
>
```

```
> # ファイ係数
> (t1 <- table(d1$通塾, d1$睡眠不足))
```

```
      0  1
0 30 10
1 20 40
```

```
> library(effectsize)
> phi(t1, alternative="two.sided")
```

```
Phi (adj.) |      95% CI
```

```
0.40      | [0.19, 0.60]
```

```
>
```

1	番号	通塾	睡眠不足
2	1	1	0
3	2	1	1
4	3	1	1
5	4	0	1
6	5	1	1
7	6	1	1
8	7	0	1
9	8	0	0
10	9	1	0
11	10	1	1
12	11	1	1
13	12	0	1
14	13	1	1
15	14	0	0
16	15	0	1
17	16	1	0
18	17	0	0
19	18	0	0
20	19	1	0
21	20	1	1

&gt; #クロス表と検定

```
> d1$通塾 <- factor(d1$通塾, levels=c(1,0), labels=c("yes", "no"))
> d1$睡眠不足 <- factor(d1$睡眠不足, levels=c(1,0), labels=c("yes", "no"))
```

```
> library(descr)
> (ct1 <-CrossTable(d1$通塾, d1$睡眠不足, expected=T, chisq=T, fisher=T))
```

Cell Contents

=====			
	N		
	Expected N		
Chi-square contribution			
N / Row Total			
N / Col Total			
N / Table Total			
=====			
=====			
d1\$通塾	d1\$睡眠不足		Total
	yes	no	
-----			
yes	40	20	60
	30.0	30.0	
	3.333	3.333	
	0.667	0.333	0.600
	0.800	0.400	
	0.400	0.200	
-----			
no	10	30	40
	20.0	20.0	
	5.000	5.000	
	0.250	0.750	0.400
	0.200	0.600	
	0.100	0.300	
-----			
Total	50	50	100
	0.500	0.500	
=====			

Statistics for All Table Factors

Pearson's Chi-squared test

```
Chi^2 = 16.66667      d.f. = 1      p = 4.455709e-05
```

Pearson's Chi-squared test with Yates' continuity correction

```
Chi^2 = 15.04167      d.f. = 1      p = 0.0001051636
```

Fisher's Exact Test for Count Data

Sample estimate odds ratio: 5.881258

Alternative hypothesis: true odds ratio is not equal to 1

p = 8.31e-05

95% confidence interval: 2.265787 16.39092

Alternative hypothesis: true odds ratio is less than 1

p = 1

95% confidence interval: % 0 14.03583

Alternative hypothesis: true odds ratio is greater than 1

p = 4.15e-05

95% confidence interval: % 2.58798 Inf

```
>
>
```

> #イエーツの連続修正を行ったカイ2乗検定

> chisq.test(t1)

Pearson's Chi-squared test with Yates' continuity correction

data: t1

X-squared = 15.042, df = 1, p-value = 0.0001052

> #尤度比検定・ピアソンのカイ2乗検定

> library(vcd)

> assocstats(t1)

	X <sup>2</sup>	df	P(> X <sup>2</sup> )
Likelihood Ratio	17.261	1	3.2582e-05
Pearson	16.667	1	4.4557e-05

Phi-Coefficient : 0.408

Contingency Coeff.: 0.378

Cramer's V : 0.408

> #フィッシャーの正確検定

> fisher.test(t1)

Fisher's Exact Test for Count Data

data: t1

p-value = 8.309e-05

alternative hypothesis: true odds ratio is not equal to 1

95 percent confidence interval:

2.265787 16.390918

sample estimates:

odds ratio

5.881258

**r × c 表 — クラメルの連関係数, カイ2乗検定, 尤度比検定**

クロス表の出力とともに検定結果を表示

```
library(descr)
CrossTable(変数1, 変数2, chisq=T, fisher=T)
```

あらかじめdescrパッケージをインストールしておく必要がある。  
ピアソンのカイ2乗検定, フィッシャーの正確検定を行う。

**クラメルの連関係数**

```
library(vcd)
assocstats(テーブル名)
```

あらかじめvcdパッケージをインストールしておく必要がある。

**尤度比検定・ピアソンのカイ2乗検定**

```
library(vcd)
assocstats(テーブル名)
```

あらかじめvcdパッケージをインストールしておく必要がある。

**フィッシャーの正確検定**

```
fisher.test(テーブル名)
```

```
> setwd("d:¥¥¥")
> d1 <- read.table("連関係数データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 モラトリアム 進路
1     1          低 就職
2     2          高 未定
3     3          高 就職
4     4          高 未定
5     5          高 進学
6     6          高 未定

> # 表を見やすくするため, カテゴリの順番を指定
> d1$モラトリアム <- factor(d1$モラトリアム, levels=c("高","低"), labels=c("高", "低"))
> d1$進路 <- factor(d1$進路, levels=c("進学","就職","未定"), labels=c("進学", "就職", "未定"))
```

1	番号	モラトリアム	進路
2	1	低	就職
3	2	高	未定
4	3	高	就職
5	4	高	未定
6	5	高	進学
7	6	高	未定
8	7	高	就職
9	8	高	未定
10	9	低	就職
11	10	低	就職
12	11	高	進学
13	12	低	未定
14	13	高	未定
15	14	高	進学
16	15	高	未定
17	16	低	就職
18	17	低	進学
19	18	高	進学
20	19	高	進学
21	20	高	未定



## &gt; #クロス表と検定

```
> d1$モラトリアム <- factor(d1$モラトリアム, levels=c("高", "低"), labels=c("高", "低"))
> d1$進路 <- factor(d1$進路, levels=c("就職", "進学", "未定"), labels=c("就職", "進学", "未定"))
```

```
> library(descr)
```

```
> (ctl <- CrossTable(d1$モラトリアム, d1$進路, expected=T, chisq=T, fisher=T))
```

```
Cell Contents
```

		N
	Expected N	
Chi-square	contribution	
	N / Row Total	
	N / Col Total	
	N / Table Total	

```
=====
```

d1\$モラトリアム	d1\$進路 就職	進学	未定	Total
高	62	29	36	127
	67.2	33.9	25.9	
	0.408	0.699	3.940	
	0.488	0.228	0.283	0.498
	0.459	0.426	0.692	
	0.243	0.114	0.141	
低	73	39	16	128
	67.8	34.1	26.1	
	0.404	0.694	3.910	
	0.570	0.305	0.125	0.502
	0.541	0.574	0.308	
	0.286	0.153	0.063	
Total	135	68	52	255
	0.529	0.267	0.204	

```
=====
```

```
Statistics for All Table Factors
```

```
Pearson's Chi-squared test
```

```
Chi^2 = 10.05543      d.f. = 2      p = 0.00655
```

```
Fisher's Exact Test for Count Data
```

```
Alternative hypothesis: two.sided
```

```
p = 0.00614
```

```
>
```

## &gt; #連関係数の検定

```
> (t1 <- table(d1$モラトリアム, d1$進路))
```

```
      就職 進学 未定
高    62   29   36
低    73   39   16
```

```
>
```

```
> # クラメルの連関係数・尤度比検定・ピアソンのカイ2乗検定
```

```
> library(vcd)
> assocstats(t1)
              X^2 df  P(> X^2)
Likelihood Ratio 10.263  2 0.0059063
Pearson          10.055  2 0.0065538
```

```
Phi-Coefficient   : 0.199
```

```
Contingency Coeff.: 0.195
```

```
Cramer's V       : 0.199
```

```
>
```

```
> # フィッシャーの正確検定
```

```
> fisher.test(t1)
```

```
Fisher's Exact Test for Count Data
```

```
data: t1
```

```
p-value = 0.006138
```

```
alternative hypothesis: two.sided
```

### Contingency Coefficientについて

連関の指標の1つとして、ピアソンの Contingency Coefficient というものがあります。これは、値が0になるとき連関無し（独立）であることを表しますが、完全な連関があっても値は1になりません（表のサイズに依存した、1未満の値になります）。計算式は、ピアソンのカイ乗値を、ピアソンのカイ2乗値に総度数を加えたもので割った値の、平方根 です。

ピアソンの Contingency Coefficient が使われることはまずありません。

Contingency Coefficient と、評定の一致度などを表すコーエンの一致係数（カッパ係数）とはまったく異なるものです。

## ピアソン残差・デビアンズ残差

標準化ピアソン残差（調整済み標準化残差）

Library(descr)

CrossTable(変数名 1, 変数名 2, asresid=T))

あらかじめdescrパッケージをインストールしておく必要がある。

## 対数線形モデルを用いる方法

フラットテーブル

t2 &lt;- ftable(d1[,c("変数名 1", "変数名 2")], row.vars=c("変数名 1", "変数名 2"))

t2 &lt;- as.data.frame(t2))

対数線形モデル

result.1 &lt;- glm(Freq ~ 変数名 1 + 変数名 2, data=フラットテーブル名, family=poisson)

ピアソン残差

(peares.1 &lt;- resid(対数線形モデル出力名, type="pearson"))

標準化ピアソン残差

xtabs(rstandard(対数線形モデル出力名, type="pearson")~フラットテーブル\$変数名 1+ フラットテーブル\$変数名 2)

デビアンズ残差

(devres.1 &lt;- resid(対数線形モデル出力名, type="deviance"))

標準化デビアンズ残差

xtabs(rstandard(対数線形モデル出力名, type="deviance")~フラットテーブル\$変数名 1+ フラットテーブル\$変数名 2)

```
> setwd("d:¥¥¥")
> d1 <- read.table("連関係数データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 モラトリアム 進路
1     1           低 就職
2     2           高 未定
3     3           高 就職
4     4           高 未定
5     5           高 進学
6     6           高 未定
>
```

&gt; # 表を見やすくするため、カテゴリの順番を指定

&gt; d1\$モラトリアム &lt;- factor(d1\$モラトリアム, levels=c("高", "低"), labels=c("高", "低"))

&gt; d1\$進路 &lt;- factor(d1\$進路, levels=c("就職", "進学", "未定"), labels=c("就職", "進学", "未定"))

&gt; library(descr)

&gt; (ctl &lt;- CrossTable(d1\$モラトリアム, d1\$進路, expected=T, asresid=T))

Cell Contents

	N
Expected N	
Chi-square contribution	
N / Row Total	
N / Col Total	
N / Table Total	
Adj Std Resid	

d1\$モラトリアム	d1\$進路 就職	進学	未定	Total
高	62	29	36	127
	67.2	33.9	25.9	
	0.408	0.699	3.940	
	0.488	0.228	0.283	0.498
	0.459	0.426	0.692	
	0.243	0.114	0.141	
	-1.314	-1.378	3.140	
低	73	39	16	128
	67.8	34.1	26.1	
	0.404	0.694	3.910	
	0.570	0.305	0.125	0.502
	0.541	0.574	0.308	
	0.286	0.153	0.063	
	1.314	1.378	-3.140	
Total	135	68	52	255
	0.529	0.267	0.204	

1	番号	モラトリアム	進路
2	1	低	就職
3	2	高	未定
4	3	高	就職
5	4	高	未定
6	5	高	進学
7	6	高	未定
8	7	高	就職
9	8	高	未定
10	9	低	就職
11	10	低	就職
12	11	高	進学
13	12	低	未定
14	13	高	未定
15	14	高	進学
16	15	高	未定
17	16	低	就職
18	17	低	進学
19	18	高	進学
20	19	高	進学
21	20	高	未定

>  
>  
>

> # 残差計算の準備

> # フラットテーブル

> t2 <- ftable(d1[,c("モラトリアム", "進路")], row.vars=c("モラトリアム", "進路"))

> (t2 <- as.data.frame(t2))

	モラトリアム	進路	Freq
1	高	就職	62
2	低	就職	73
3	高	進学	29
4	低	進学	39
5	高	未定	36
6	低	未定	16

> # 対数線形モデル

> result.1 <- glm(Freq ~ モラトリアム + 進路, data=t2, family=poisson)

> summary(result.1)

Call:

glm(formula = Freq ~ モラトリアム + 進路, family = poisson, data = t2)

Deviance Residuals:

1	2	3	4	5	6
-0.6470	0.6280	-0.8576	0.8143	1.8733	-2.1313

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	4.208198	0.106583	39.483	< 2e-16 ***
モラトリアム低	0.007843	0.125246	0.063	0.95
進路進学	-0.685767	0.148705	-4.612	4.00e-06 ***
進路未定	-0.954031	0.163212	-5.845	5.05e-09 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 53.722 on 5 degrees of freedom  
Residual deviance: 10.263 on 2 degrees of freedom  
AIC: 51.125

Number of Fisher Scoring iterations: 4

```
> # 標準化ピアソン残差（調整済み標準化残差）
```

```
> # ピアソン残差
```

```
> (peares.1 <- resid(result.1, type="pearson"))
      1      2      3      4      5      6
-0.6384728  0.6359739 -0.8362678  0.8329947  1.9850536 -1.9772843
>
```

```
> # 標準化ピアソン残差
```

```
> xtabs(rstandard(result.1, type="pearson")~t2$モラトリアム+ t2$進路)
      t2$進路
t2$モラトリアム 就職      進学      未定
高 -1.313671 -1.378350  3.140217
低  1.313672  1.378351 -3.140217
>
```

```
> # デビアンズ残差
```

```
> (devres.1 <- resid(result.1, type="deviance"))
      1      2      3      4      5      6
-0.6470397  0.6280379 -0.8576008  0.8142911  1.8733391 -2.1312969
>
```

```
> # 標準化デビアンズ残差
```

```
> xtabs(rstandard(result.1, type="deviance")~t2$モラトリアム+ t2$進路)
      t2$進路
t2$モラトリアム 就職      進学      未定
高 -1.331298 -1.413512  2.963492
低  1.297279  1.347402 -3.384812
>
>
>
```

## 2名の評定者の評定の一致度 — Cohenのカッパ係数

## vcdパッケージを使う方法

```
library(vcd)
Kappa(テーブル名)
```

あらかじめvcdパッケージをインストールしておく必要がある。

## psychパッケージを使う方法

```
library(psych)
cohen.kappa(テーブル名)
```

あらかじめpsychパッケージをインストールしておく必要がある。

## irr パッケージを使う方法

```
library(irr)
kappa2(データフレーム名)
```

あらかじめirrパッケージをインストールしておく必要がある。

**weight** オプションを指定することにより、重みカッパ係数を算出する。  
weight には"unweighted" (デフォルト), "equal", "squared"の水準がある。

```
> setwd("d:¥¥")
> d1 <- read.table("カッパ係数_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id rater1 rater2
1  1      1      1
2  2      1      1
3  3      1      1
4  4      1      1
5  5      1      1
6  6      1      1
>
> #評定データフレームの作成
> d2 <- d1[,c(-1)]
>
> #クロス表の作成
> (t1 <- table(d1$rater1, d1$rater2))

   1 2 3
1 6 1 1
2 1 7 0
3 2 1 6
>
> # vcdパッケージを使う方法
> library(vcd)
> Kappa(t1)

      value      ASE      z Pr(>|z|)
Unweighted 0.6411 0.1268 5.056 4.278e-07
Weighted    0.5946 0.1501 3.962 7.421e-05
>
>
```

id	rater1	rater2
1	1	1
2	1	1
3	1	1
4	1	1
5	1	1
6	1	1
7	1	2
8	1	3
9	2	2
10	2	2
11	2	2
12	2	2
13	2	2
14	2	2
15	2	2
16	2	1
17	3	3
18	3	3
19	3	3
20	3	3
21	3	3
22	3	3
23	3	1
24	3	2
25	3	1

```
> # pschパッケージを使う方法
> library(psych)
> cohen.kappa(t1)
```

```
Call: cohen.kappa1(x = x, w = w, n.obs = n.obs, alpha = alpha, levels = levels)
```

```
Cohen Kappa and Weighted Kappa correlation coefficients and confidence boundaries
```

	lower	estimate	upper
unweighted kappa	0.39	0.64	0.89
weighted kappa	0.18	0.55	0.92

```
Number of subjects = 25
```

```
警告メッセージ:
```

```
any(abs(bounds)) で: 'double' 型の引数を論理型に変換します
```

```
>
```

```
> # irr パッケージを使う方法
> library(irr)
> kappa2(d2)
```

```
Cohen's Kappa for 2 Raters (Weights: unweighted)
```

```
Subjects = 25
Raters = 2
Kappa = 0.641
```

```
z = 4.57
p-value = 4.78e-06
```

```
> kappa2(d2, weight="equal")
Cohen's Kappa for 2 Raters (Weights: equal)
```

```
Subjects = 25
Raters = 2
Kappa = 0.595
```

```
z = 3.81
p-value = 0.000137
```

```
> kappa2(d2, weight="squared")
Cohen's Kappa for 2 Raters (Weights: squared)
```

```
Subjects = 25
Raters = 2
Kappa = 0.548
```

```
z = 2.77
p-value = 0.00561
```

```
>
```

```
>
```

## 複数名の評定者の評定の一致度 — Fleissのカッパ係数

## irr パッケージを使う方法

```
library(irr)
kappam.fleiss(データフレーム名)
```

あらかじめirrパッケージをインストールしておく必要がある。

```
> setwd("d:¥¥")
> d1 <- read.table("カッパ係数Fleiss_データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  id rater1 rater2 rater3 rater4
1  1      1      1      1      1
2  2      1      1      1      1
3  3      1      1      1      1
4  4      1      1      1      2
5  5      1      1      2      3
6  6      1      1      2      3
>

> #評定データフレームの作成
> d2 <- d1[,c(-1)]
>
>
> # irr パッケージを使う方法
> library(irr)
> kappam.fleiss(d2)
Fleiss' Kappa for m Raters

Subjects = 25
Raters = 4
Kappa = 0.31

      z = 5.32
p-value = 1.01e-07
>
>
```

id	rater1	rater2	rater3	rater4
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1
4	1	1	1	2
5	1	1	2	3
6	1	1	2	3
7	1	2	2	2
8	1	3	2	2
9	2	2	3	2
10	2	2	2	2
11	2	2	3	2
12	2	2	2	3
13	2	2	3	2
14	2	2	2	2
15	2	2	2	2
16	2	1	2	3
17	3	3	2	2
18	3	3	2	1
19	3	3	3	1
20	3	3	3	3
21	3	3	3	1
22	3	3	3	2
23	3	1	3	3
24	3	2	2	3
25	3	1	2	3



## 13 比率に関する推測

## 1 群の比率の推測 — 二項検定

## 二項検定

```
binom.test(テーブル名, p=設定値)
```

## 近似検定

```
prop.test(テーブル名, p=設定値)
```

```
>
> setwd("d:¥¥")
> d1 <- read.table("1群の比率データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  番号  合否
1     1     1
2     2     1
3     3     0
4     4     1
5     5     1
6     6     1
```

```
> # 度数分布表
> d1$gouhi <- factor(d1$合否, levels=c(1,0), labels=c("success","failure"))
> (t1 <- table(d1$gouhi))
```

```
success failure
      19       6
```

```
>
> # 設定値
> p0 <- 0.6
```

```
> # 二項検定
> binom.test(t1, p=p0)
```

```
Exact binomial test
```

```
data:  t1
number of successes = 19, number of trials = 25, p-value = 0.1514
alternative hypothesis: true probability of success is not equal to 0.6
95 percent confidence interval:
 0.5487120 0.9064356
sample estimates:
probability of success
                0.76
```

```
> # 近似検定
> prop.test(t1, p=p0)
```

```
1-sample proportions test with continuity correction
```

```
data:  t1, null probability p0
X-squared = 2.0417, df = 1, p-value = 0.153
alternative hypothesis: true p is not equal to 0.6
95 percent confidence interval:
 0.5447916 0.8984194
sample estimates:
      p
0.76
```

```
>
>
>
```

1	番号	合否
2	1	1
3	2	1
4	3	0
5	4	1
6	5	1
7	6	1
8	7	1
9	8	1
10	9	1
11	10	1
12	11	0
13	12	1
14	13	1
15	14	0
16	15	0
17	16	1
18	17	1
19	18	1
20	19	0
21	20	0

## 対応のある2群の比率の比較 — マクネマーの検定

```
library(descr)
CrossTable(変数名 1, 変数名 2, mcnemar=T)
```

または

```
mcnemar.test(変数名 1, 変数名 2)
mcnemar.test(テーブル名)
```

対応のある比率の差の信頼区間を推定する関数  
dPP2(x=変数名 1, y=変数名 2)

conf=で信頼係数を設定できる。デフォルトは0.95.

```
#-----
dPP2 <- function(data, conf=0.95) {
  t1 <- table(data)
  a <- t1[1,1]
  b <- t1[1,2]
  c <- t1[2,1]
  d <- t1[2,2]
  n <- sum(t1)
  se <- sqrt(b+c-(b-c)^2/n)/n
  p1 <- mean(data[, 1])
  p2 <- mean(data[, 2])
  z0 <- qnorm((1-(1-conf)/2))
  ci.l <- p1-p2-z0*se
  ci.u <- p1-p2+z0*se
  title <- "Difference of paired proportions"
  stnames <- c("p1", "p2", "difference", paste("lower(", conf, ")", sep=""), paste("upper(", conf, ")",
sep=""))
  statistic <- round(data.frame(p1, p2, (p1-p2), ci.l, ci.u), 4)
  colnames(statistic) <- stnames
  out <- list(title, statistic=statistic)
  return(out)
}
#-----
```

```
> setwd("d:¥¥")
> d1 <- read.table("対応のある2群の比率データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 子ども 親
1     1      1  1
2     2      1  1
3     3      1  1
4     4      1  1
5     5      1  1
6     6      1  1
>
>
> # 各群における比率
> colMeans(d1[c("子ども", "親")])
  子ども      親
0.6133333 0.4400000
>
>
```

```
> #クロス表と検定
> library(descr)
> (t1 <- CrossTable(d1$子ども, d1$親, mcnemar=T))
Cell Contents
```

	N
Chi-square contribution	
N / Row Total	
N / Col Total	
N / Table Total	

```
=====
```

	d1\$親		
d1\$子ども	0	1	Total
0	21	8	29
	1.395	1.776	
	0.724	0.276	0.387
	0.500	0.242	
	0.280	0.107	
1	21	25	46
	0.880	1.119	
	0.457	0.543	0.613
	0.500	0.758	
	0.280	0.333	
Total	42	33	75
	0.560	0.440	

```
=====
```

1	番号	子ども	親
2	1	1	1
3	2	1	1
4	3	1	1
5	4	1	1
6	5	1	1
7	6	1	1
8	7	1	1
9	8	1	1
10	9	1	1
11	10	1	1
12	11	1	1
13	12	1	1
14	13	1	1
15	14	1	1
16	15	1	1
17	16	1	1
18	17	1	1
19	18	1	1
20	19	1	1
21	20	1	1

McNemar's Chi-squared test

```
Chi^2 = 5.827586      d.f. = 1      p = 0.01577676
```

McNemar's Chi-squared test with continuity correction

```
Chi^2 = 4.965517      d.f. = 1      p = 0.02585758
```

```
>
>
```

```
> #マクネマーの検定
```

```
> mcnemar.test(d1$子ども, d1$親)
```

McNemar's Chi-squared test with continuity correction

data: d1\$子ども and d1\$親

McNemar's chi-squared = 4.9655, df = 1, p-value = 0.02586

```
>
```

```
> #対応のある比率の差の信頼区間
```

```
> dPP2(data=d1[,c("子ども","親")])
```

```
[[1]]
```

```
[1] "Difference of paired proportions"
```

```
$statistic
```

```
      p1    p2 difference lower(0.95) upper(0.95)
1 0.6133 0.44      0.1733      0.0382      0.3085
```

## 対応のある多群の比率の比較 — コクランのQ検定

```
cochranq.test(データ名)
```

あらかじめCVSTパッケージをインストールしておく必要がある。  
データは、分析したい変数だけ残して、不要な変数は削除しておく。

## 同時信頼区間

```
dPPs(data=データ名)
```

```
#-----
dPPs <- function(data, conf=0.95) {
  nc <- ncol(data)
  npair <- (nc*(nc-1))/2
  pairs <- matrix(c(0), npair, 5)
  k <- 0
  for(i in 1:(nc-1)) {
    for(j in (i+1):nc) {
      d2 <- data[,c(i, j)]
      t1 <- table(d2)
      a <- t1[1,1]
      b <- t1[1,2]
      c <- t1[2,1]
      d <- t1[2,2]
      n <- sum(t1)
      se <- sqrt(b+c-(b-c)^2/n)/n
      p1 <- mean(data[, i])
      p2 <- mean(data[, j])
      z0 <- qnorm((1-((1-conf)/(npair/2))/2))
      ci.l <- p1-p2-z0*se
      ci.u <- p1-p2+z0*se
      k <- k+1
      pairs[k,] <- round(c(p1, p2, (p1-p2), ci.l, ci.u), 4)
    }
  }
  rownames(pairs) <- c(1:npair)
  jconf <- round(100*(1-((1-conf)/(npair/2))), 1)
  colnames(pairs) <- c("p1", "p2", "diff", paste("lower(", jconf, ")", sep=""), paste("upper(", jconf, ")", sep=""))
  title <- paste("Joint ", 100*conf, "% Confidence Intervals", sep="")
  out <- list(title, CI= pairs)
  return(out)
}
#-----
```

```
> setwd("d:¥¥")
> d1 <- read.table("対応のある多群の比率データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  番号 福祉系 教育系 幼保系
1     1      0      0      0
2     2      1      0      0
3     3      1      1      1
4     4      1      1      0
5     5      0      0      1
6     6      1      0      0
>
>
> # 各群の比率
> colMeans(d1[, c("福祉系", "教育系", "幼保系")])
  福祉系 教育系 幼保系
0.7232143 0.4196429 0.6071429
>
>
> # cochran.Qを使う方法
> library(CVST)
> cochranq.test(d1[, c("福祉系", "教育系", "幼保系")])
```

## Cochran's Q Test

```
data: d1[, c("福祉系", "教育系", "幼保系")]
Cochran's Q = 22.075, df = 2, p-value = 1.609e-05
```

&gt;

&gt;

&gt;

&gt; # 対応のある比率の差の信頼区間

&gt; dPPs(data=d1[, c("福祉系", "教育系", "幼保系")])

[1] "Joint 95% Confidence Intervals"

\$CI

	p1	p2	diff	lower(96.7)	upper(96.7)
1	0.7232	0.4196	0.3036	0.1670	0.4402
2	0.7232	0.6071	0.1161	-0.0003	0.2324
3	0.4196	0.6071	-0.1875	-0.3285	-0.0465

&gt;

&gt;

1	番号	福祉系	教育系	幼保系
2	1	0	0	0
3	2	1	0	0
4	3	1	1	1
5	4	1	1	0
6	5	0	0	1
7	6	1	0	0
8	7	1	0	1
9	8	1	1	1
10	9	0	1	1
11	10	0	1	0
12	11	1	1	1
13	12	0	1	1
14	13	1	1	1
15	14	1	1	1
16	15	1	1	1
17	16	0	0	0
18	17	1	0	1
19	18	1	0	1
20	19	0	0	0
21	20	1	1	1

## リスク差・リスク比・オッズ比の推測

## リスク差

prop.test(テーブル名)

## リスク比・オッズ比

```
library(epitools)
epitab(テーブル名, method=c("riskratio"))
epitab(テーブル名, method=c("oddsratio"))
```

データ値を、「0=なし, 1=あり」(「なし」のほうが小さい値)としておかなければならない  
 CrossTable関数による見やすい表と, table関数で作成して, リスク比・オッズ比を計算する元になる表で,  
 行および列の1,0の並び順が逆になることに注意する.

あらかじめepitoolsパッケージをインストールしておく必要がある.  
 method で "oddsratio" か "riskratio" を指定する. "rateratio" というオプションもある.  
 vcdパッケージにもoddsratioという関数がある. オッズ比の計算をしてくれる.

```
> setwd("d:¥¥")
>
```

## # 前向き研究

```
> d1 <- read.table("対応のない2群の比率データ前.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  番号 身内患者 医療関係
1    1         0         0
2    2         0         0
3    3         1         1
4    4         1         0
5    5         1         1
6    6         0         1
>
```

## # クロス表

```
> d1$身内患者rev <- factor(d1$身内患者, levels=c(1,0))
> d1$医療関係rev <- factor(d1$医療関係, levels=c(1,0))
> library(descr)
> CrossTable(d1$身内患者rev, d1$医療関係rev, prop.chisq=F, prop.c=F, prop.t=F)
```

Cell Contents	
	N
N / Row Total	

d1\$身内患者rev	d1\$医療関係rev		Total
	1	0	
1	30 0.400	45 0.600	75 0.500
0	15 0.200	60 0.800	75 0.500
Total	45	105	150

1	番号	身内患者	医療関係
2	1	0	0
3	2	0	0
4	3	1	1
5	4	1	0
6	5	1	1
7	6	0	1
8	7	0	0
9	8	1	0
10	9	1	0
11	10	1	0
12	11	0	0
13	12	0	0
14	13	0	0
15	14	0	0
16	15	0	0
17	16	1	0
18	17	0	0
19	18	1	0
20	19	0	0
21	20	1	1

```

> # リスク差
> (t1 <- table(d1$身内患者, d1$医療関係))

      0  1
0 60 15
1 45 30

> prop.test(t1)

      2-sample test for equality of proportions with continuity
      correction

data:  t1
X-squared = 6.2222, df = 1, p-value = 0.01262
alternative hypothesis: two.sided
95 percent confidence interval:
 0.04353114 0.35646886
sample estimates:
prop 1 prop 2
 0.8      0.6

>

> # リスク比
> library(epitools)
> epitab(t1, method=c("riskratio"))
$tab

      0 p0  1  p1 riskratio      lower      upper      p.value
0 60 0.8 15 0.2           1         NA         NA         NA
1 45 0.6 30 0.4           2 1.176315 3.400451 0.01215822

$measure
[1] "wald"

$conf.level
[1] 0.95

$pvalue
[1] "fisher.exact"

>

> # オッズ比
> library(epitools)
> epitab(t1, method=c("oddsratio"))
$tab

      0      p0  1      p1 oddsratio      lower      upper      p.value
0 60 0.5714286 15 0.3333333 1.000000         NA         NA         NA
1 45 0.4285714 30 0.6666667 2.666667 1.284531 5.535958 0.01215822

$measure
[1] "wald"

$conf.level
[1] 0.95

$pvalue
[1] "fisher.exact"

```



```
>
> # 後ろ向き研究
> d1 <- read.table("対応のない2群の比率データ後.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 医療関係 身内患者
1     1         0         0
2     2         0         0
3     3         1         0
4     4         1         0
5     5         0         0
6     6         0         0
>
```

```
> # クロス表
> d1$身内患者rev <- factor(d1$身内患者, levels=c(1,0))
> d1$医療関係rev <- factor(d1$医療関係, levels=c(1,0))
> library(descr)
> CrossTable(d1$身内患者rev, d1$医療関係rev, prop.chisq=F, prop.r=F, prop.t=F)
```

```
Cell Contents
-----
N
N / Col Total
-----
```

```
=====
d1$身内患者rev    d1$医療関係rev    Total
1                 25             12      37
                 0.333         0.158
-----
0                 50             64     114
                 0.667         0.842
-----
Total              75             76     151
                 0.497         0.503
=====
>
```

1	番号	医療関係	身内患者
2	1	0	0
3	2	0	0
4	3	1	0
5	4	1	0
6	5	0	0
7	6	0	0
8	7	1	0
9	8	0	0
10	9	0	0
11	10	1	1
12	11	1	0
13	12	1	0
14	13	0	0
15	14	0	0
16	15	0	1
17	16	0	1
18	17	0	0
19	18	1	0
20	19	0	0
21	20	0	0

```
> # オッズ比
> library(epitools)
> epitab(t1, method=c("oddsratio"))
$tab
      0      p0  1      p1 oddsratio    lower    upper    p.value
0 60 0.5714286 15 0.3333333 1.000000      NA      NA      NA
1 45 0.4285714 30 0.6666667 2.666667 1.284531 5.535958 0.01215822

$measure
[1] "wald"

$conf.level
[1] 0.95

$pvalue
[1] "fisher.exact"
```

```
>
>
```

```
>
>
```

```
> # 横断研究
> d1 <- read.table("対応のない2群の比率データ横.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 身内患者 医療関係
1     1         0         0
2     2         1         0
3     3         0         1
4     4         0         0
5     5         0         0
6     6         1         0
>
```

1	番号	身内患者	医療関係
2	1	0	0
3	2	1	0
4	3	0	1
5	4	0	0
6	5	0	0
7	6	1	0
8	7	0	0
9	8	0	0
10	9	0	0
11	10	0	0
12	11	0	0
13	12	0	1
14	13	0	0
15	14	0	0
16	15	0	1
17	16	0	0
18	17	1	0
19	18	0	1
20	19	0	0
21	20	0	0

```
> # クロス表
> d1$身内患者rev <- factor(d1$身内患者, levels=c(1,0))
> d1$医療関係rev <- factor(d1$医療関係, levels=c(1,0))
> library(descr)
> CrossTable(d1$身内患者rev, d1$医療関係rev, prop.chisq=F)
```

Cell Contents

	N
N / Row Total	
N / Col Total	
N / Table Total	

```
=====
d1$身内患者rev      d1$医療関係rev
                    1         0      Total
-----
1                   18        12        30
    0.600    0.400    0.200
    0.375    0.118
    0.120    0.080
-----
0                   30        90       120
    0.250    0.750    0.800
    0.625    0.882
    0.200    0.600
-----
Total                48       102       150
    0.320    0.680
=====
>
```

```
> # リスク差
> (t1 <- table(d1$身内患者, d1$医療関係))
```

```
      0  1
0 90 30
1 12 18
> prop.test(t1)
```

2-sample test for equality of proportions with continuity correction

```
data:  t1
X-squared = 11.9504, df = 1, p-value = 0.0005463
alternative hypothesis: two.sided
95 percent confidence interval:
 0.1375056 0.5624944
sample estimates:
prop 1 prop 2
 0.75   0.40
```

```
>
```

```
> # リスク比
> library(epitools)
> epitab(t1, method=c("riskratio"))
$tab
```

	0	p0	1	p1	riskratio	lower	upper	p.value
0	90	0.75	30	0.25	1.0	NA	NA	NA
1	12	0.40	18	0.60	2.4	1.567615	3.674372	0.0004215309

```
$measure
[1] "wald"
```

```
$conf.level
[1] 0.95
```

```
$pvalue
[1] "fisher.exact"
```

```
>
> # オッズ比
> library(epitools)
> epitab(t1, method=c("oddsratio"))
$tab
```

	0	p0	1	p1	oddsratio	lower	upper	p.value
0	90	0.8823529	30	0.625	1.0	NA	NA	NA
1	12	0.1176471	18	0.375	4.5	1.94424	10.41538	0.0004215309

```
$measure
[1] "wald"
```

```
$conf.level
[1] 0.95
```

```
$pvalue
[1] "fisher.exact"
```

## 対応のない多群の比率の等質性の推測

prop.test(テーブル名)

## 同時信頼区間

dNPs (x=群分け変数名, y=0/1変数名)

```
#-----
dNPs <- function(x, y, conf=0.95) {
  t1 <- table(x, y)
  nc <- length(levels(x))
  npair <- (nc*(nc-1))/2
  pairs <- matrix(c(0), npair, 5)
  k <- 0
  for(i in 1:(nc-1)) {
    for(j in (i+1) : nc) {
      t12 <- t1[c(i, j), ]
      tt <- prop.test(t12, conf.level=(1-((1-conf)/(npair/2))))
      p1 <- tt$estimate[1]
      p2 <- tt$estimate[2]
      ci.l <- tt$conf.int[1]
      ci.u <- tt$conf.int[2]
      k <- k+1
      pairs[k,] <- round(c(p1, p2, (p1-p2), ci.l, ci.u), 4)
    }
  }
  rownames(pairs) <- c(1:npair)
  jconf <- round(100*(1-((1-conf)/(npair/2))), 1)
  colnames(pairs) <- c("p1", "p2", "diff", paste("lower(", jconf, ")", sep=""), paste("upper(", jconf, ")", sep=""))
  title <- paste("Joint ", 100*conf, "% Confidence Intervals", sep="")
  out <- list(title, CI= pairs)
  return(out)
}
#-----
```

```
> setwd("d:¥¥")
> d1 <- read.table("対応のない多群の比率データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  番号 地域 罹患
1     1   A     1
2     2   C     0
3     3   B     0
4     4   B     0
5     5   C     0
6     6   A     1
>
>
```

番号	地域	罹患
1	A	1
2	C	0
3	B	0
4	B	0
5	C	0
6	A	1
7	A	1
8	C	1
9	B	0
10	B	0
11	C	0
12	B	0
13	C	0
14	A	0
15	A	0
16	C	1
17	A	0
18	C	1
19	B	0
20	C	0

```
> # クロス表
> d1$地域 <- factor(d1$地域, levels=c("A", "B", "C"))
> d1$罹患 <- factor(d1$罹患, levels=c(1, 0))

> library(descr)
> CrossTable(d1$地域, d1$罹患, prop.chisq=F, prop.c=F, prop.t=F)
```

Cell Contents

	N	
	N / Row	Total

```
=====
d1$地域  d1$罹患
          1      0  Total
-----
A          12     22     34
          0.353  0.647  0.239
-----
B           8     38     46
          0.174  0.826  0.324
-----
C          26     36     62
          0.419  0.581  0.437
-----
Total         46     96    142
=====
```

```
>
>
```

```
> # 対応のない多群の比率
> (t1 <- table(d1$地域, d1$罹患))
```

```
      1  0
A 12 22
B  8 38
C 26 36
```

```
> prop.test(t1)
```

3-sample test for equality of proportions without continuity correction

```
data:  t1
X-squared = 7.4355, df = 2, p-value = 0.02429
alternative hypothesis: two.sided
sample estimates:
  prop 1    prop 2    prop 3
0.3529412 0.1739130 0.4193548
```

```
>
> # 対応のない比率の差の信頼区間
> dNPs(x=d1$地域, y=d1$罹患)
```

[1] "Joint 95% Confidence Intervals"

```
$CI
      p1      p2      diff lower(96.7) upper(96.7)
1 0.3529 0.1739  0.1790    -0.0576     0.4157
2 0.3529 0.4194 -0.0664    -0.3087     0.1759
3 0.1739 0.4194 -0.2454    -0.4431    -0.0478
```

# 14 回帰分析

## 単回帰分析 — lm関数

オブジェクト名 <- lm(基準変数 ~ 説明変数, データフレーム名)  
summary(オブジェクト名)

lm() でモデルを指定し, 結果を「オブジェクト名」に保存する. その内容をsummary() で表示する.

```
> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id stress kyoufu support utsu work result
1  1     20    2.2      17   18     0      0
2  2     23    4.8      18   21     1      0
3  3     30    5.8      12   29     1      1
4  4     25    5.2      18   29     0      1
5  5     26    2.0       8   22     1      0
6  6     21    5.0      26   19     1      0
>
>
> # 記述統計量
> dtmp <- d1[,c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

	N	Mean	SD	stress	kyoufu	support	utsu	work	result
stress	245	22.94	5.25	1.00	0.40	-0.34	0.62	0.03	0.44
kyoufu	245	4.05	1.17	0.40	1.00	-0.03	0.31	0.10	0.20
support	245	18.42	4.96	-0.34	-0.03	1.00	-0.51	-0.03	-0.39
utsu	245	20.29	6.49	0.62	0.31	-0.51	1.00	0.02	0.76
work	245	0.50	0.50	0.03	0.10	-0.03	0.02	1.00	-0.08
result	245	0.26	0.44	0.44	0.20	-0.39	0.76	-0.08	1.00

```
>
>
> # 単回帰分析
> reg.1 <- lm(utsu ~ stress, data=d1)
> summary(reg.1)

Call:
lm(formula = utsu ~ stress, data = d1)

Residuals:
    Min       1Q   Median       3Q      Max
-11.6277  -3.2181  -0.0374   3.0771  15.6674

Coefficients:
(Intercept)  2.73611      1.46777      1.864      0.0635 .
stress       0.76506      0.06238     12.264    <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.114 on 243 degrees of freedom
Multiple R-squared:  0.3823,    Adjusted R-squared:  0.3798
F-statistic: 150.4 on 1 and 243 DF,  p-value: < 2.2e-16

>
>
> # 偏回帰係数の信頼区間
> confint(reg.1)
```

	2.5 %	97.5 %
(Intercept)	-0.1550688	5.6272812
stress	0.6421862	0.8879398

```
>
>
```

```

> #散布図と回帰直線を描く
> plot(d1$stress, d1$utsu, xlim=c(0,40), ylim=c(0,40),
+      pch=20, las=1, xlab="ストレス", ylab="うつ傾向")
>
> #回帰直線
> abline(reg.1, lwd=2)
> (a <- reg.1$coefficients["(Intercept)"])
(Intercept)
2.736106
> (b <- reg.1$coefficients["stress"])
stress
0.765063
> segments(-1, 38, 0.5, 38, lwd=2, lty=1)
> text(1, 38, paste("Y = ", round(a, 2), " + ", round(b, 2), " X", sep=""), pos=4)
>
>
> # データの標準化
> d2 <- as.data.frame(scale(d1[, c("stress", "kyoufu", "support", "utsu")]))
>
> #共分散行列の確認
> cov(d2)

```

```

      stress      kyoufu      support      utsu
stress  1.0000000  0.40369513 -0.33786517  0.6183281
kyoufu  0.4036951  1.00000000 -0.02670051  0.3124069
support -0.3378652 -0.02670051  1.00000000 -0.5064285
utsu     0.6183281  0.31240691 -0.50642854  1.0000000

```

```

> #標準偏回帰係数の推定
> reg.2 <- lm(utsu ~ stress, data=d2)
> summary(reg.2)

```

```

Call:
lm(formula = utsu ~ stress, data = d2)

```

```

Residuals:
    Min       1Q   Median       3Q      Max
-1.79071 -0.49560 -0.00575  0.47388  2.41283

```

```

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.448e-16  5.031e-02   0.00      1
stress       6.183e-01  5.042e-02  12.26 <2e-16 ***
---

```

```

Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

Residual standard error: 0.7875 on 243 degrees of freedom
Multiple R-squared:  0.3823,    Adjusted R-squared:  0.3798
F-statistic: 150.4 on 1 and 243 DF,  p-value: < 2.2e-16

```

```

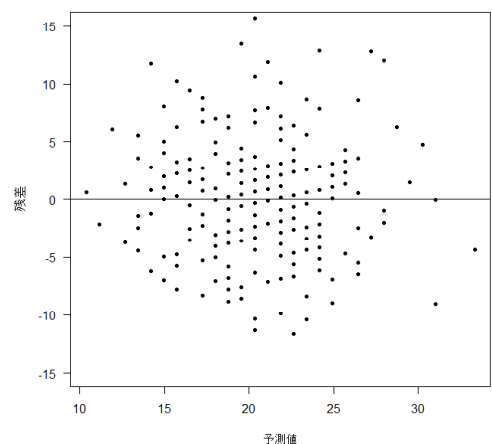
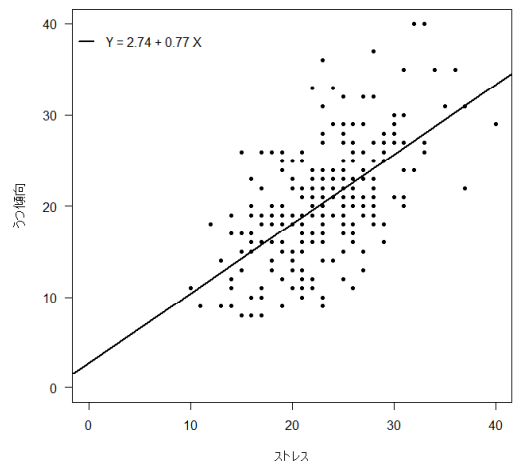
>
> # (標準) 偏回帰係数の信頼区間
> confint(reg.2)
      2.5 %      97.5 %
(Intercept) -0.09910683 0.09910683
stress       0.51901839 0.71763782

```

```

> # 残差プロット
> plot(reg.1$fitted.values, reg.1$residuals, pch=20, las=1,
+      ylim=c(-15, 15), xlab="予測値", ylab="残差", main="")
> segments(5, 0, 40, 0)
>
>

```





## 重回帰分析 — lm関数

## 偏回帰係数の推定と信頼区間

```
オブジェクト名1 <- lm(基準変数 ~ 説明変数1 + 説明変数2 + ..., データフレーム名1)
summary(オブジェクト名1)
confint(オブジェクト名1)
```

## 標準偏回帰係数の推定と信頼区間

```
データフレーム名2 <- as.data.frame(scale(データフレーム名1))
オブジェクト名2 <- lm(基準変数 ~ 説明変数1 + 説明変数2 + ..., データフレーム名2)
summary(オブジェクト名2)
confint(オブジェクト名2)
```

## 多重共線性の診断

```
VIF <- diag(solve(cor(説明変数間の相関係数行列)))
VIF      # VIF(variance inflaion factor)
1/VIF    # torelance
```

または

```
library(car)
vif(lmオブジェクト名)
```

あらかじめcarパッケージをインストールしておく必要がある。

## ステップワイズ分析

```
library(MASS)
オブジェクト名 <- stepAIC(lmオブジェクト名)
summary(オブジェクト名)
```

MASSパッケージは最初からインストールされている。

```
> setwd("d:¥¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  id stress kyoufu support utsu work result
1  1    20    2.2    17   18    0    0
2  2    23    4.8    18   21    1    0
3  3    30    5.8    12   29    1    1
4  4    25    5.2    18   29    0    1
5  5    26    2.0    8    22    1    0
6  6    21    5.0    26   19    1    0
```

```
>
> # 記述統計量
> dtmp <- d1[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD stress kyoufu support  utsu  work result
stress 245 22.94 5.25  1.00  0.40  -0.34  0.62  0.03  0.44
kyoufu 245  4.05 1.17  0.40  1.00  -0.03  0.31  0.10  0.20
support 245 18.42 4.96 -0.34 -0.03  1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49  0.62  0.31 -0.51  1.00  0.02  0.76
work    245  0.50 0.50  0.03  0.10 -0.03  0.02  1.00 -0.08
result  245  0.26 0.44  0.44  0.20 -0.39  0.76 -0.08  1.00
```

```
>
> #偏回帰係数の推定
> result.1 <- lm(utsu ~ stress + kyoufu, data=d1)
> summary(result.1)
```

1	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```

Call:
lm(formula = utsu ~ stress + kyoufu, data = d1)

Residuals:
    Min       1Q   Median       3Q      Max
-11.3092  -3.2097  -0.0432   3.1134  16.0238

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.90963    1.58558   1.204    0.230
stress       0.72759    0.06806  10.690 <2e-16 ***
kyoufu       0.41623    0.30522   1.364    0.174
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 5.105 on 242 degrees of freedom
Multiple R-squared:  0.387,    Adjusted R-squared:  0.382
F-statistic: 76.4 on 2 and 242 DF,  p-value: < 2.2e-16

> confint(result.1)
              2.5 %    97.5 %
(Intercept) -1.2136587 5.0329271
stress       0.5935196 0.8616657
kyoufu      -0.1850019 1.0174659
>
>
> #標準偏回帰係数の推定
> d2 <- as.data.frame(scale(d1[,c("stress", "kyoufu", "utsu")]))
> #共分散行列の確認
> round(cov(d2), 2)
      stress kyoufu utsu
stress  1.00   0.40 0.62
kyoufu  0.40   1.00 0.31
utsu    0.62   0.31 1.00

> result.2 <- lm(utsu ~ stress + kyoufu , data=d2)
> summary(result.2)

Call:
lm(formula = utsu ~ stress + kyoufu, data = d2)

Residuals:
    Min       1Q   Median       3Q      Max
-1.74165 -0.49430 -0.00665  0.47948  2.46771

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.235e-16  5.023e-02   0.000    1.000
stress      5.880e-01  5.501e-02  10.690 <2e-16 ***
kyoufu      7.502e-02  5.501e-02   1.364    0.174
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7861 on 242 degrees of freedom
Multiple R-squared:  0.387,    Adjusted R-squared:  0.382
F-statistic: 76.4 on 2 and 242 DF,  p-value: < 2.2e-16

> confint(result.2)
              2.5 %    97.5 %
(Intercept) -0.09893404 0.09893404
stress       0.47968584 0.69640299
kyoufu      -0.03334233 0.18337482
>
>
> #多重共線性の確認
> # carパッケージのvifを使う方法
> library(car)
> VIF <- vif(result.2)
> TOLERANCE <- 1/VIF

```

```
> round(data.frame(VIF, TOLERANCE), 3)
      VIF TOLERANCE
stress 1.195      0.837
kyoufu 1.195      0.837
>
```

```
> # 説明変数だけのデータから計算する方法
```

```
> d3 <- d1[,c("stress", "kyoufu")]
```

```
> VIF <- diag(solve(cor(d3)))
```

```
> tolerance <- 1/VIF
```

```
> data.frame(VIF, tolerance)
```

```
      VIF tolerance
```

```
stress 1.1947 0.8370302
```

```
kyoufu 1.1947 0.8370302
```

```
>
```

```
>
```

```
>
```

```
> #残差プロットなどを描く
```

```
> #図を4枚同時に並べる指定
```

```
> par(mfrow=c(2, 2))
```

```
> plot(result.2)
```

```
> par(mfrow=c(1, 1))
```

```
>
```

```
>
```

```
> #ステップワイズ分析
```

```
> #MASSパッケージの読み込み
```

```
> library(MASS)
```

```
> result.3 <- stepAIC(result.2)
```

```
Start:  AIC=-114.92
```

```
utsu ~ stress + kyoufu
```

```
      Df Sum of Sq  RSS      AIC
```

```
- kyoufu  1      1.149 150.71 -115.043
```

```
<none>                149.56 -114.919
```

```
- stress  1     70.624 220.19  -22.162
```

```
Step:  AIC=-115.04
```

```
utsu ~ stress
```

```
      Df Sum of Sq  RSS      AIC
```

```
<none>                150.71 -115.043
```

```
- stress  1     93.288 244.00    0.998
```

```
> summary(result.3)
```

```
Call:
```

```
lm(formula = utsu ~ stress, data = d2)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.79071 -0.49560 -0.00575  0.47388  2.41283
```

```
Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.448e-16  5.031e-02   0.00      1
stress      6.183e-01  5.042e-02  12.26 <2e-16 ***
```

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 0.7875 on 243 degrees of freedom
```

```
Multiple R-squared:  0.3823,    Adjusted R-squared:  0.3798
```

```
F-statistic: 150.4 on 1 and 243 DF,  p-value: < 2.2e-16
```

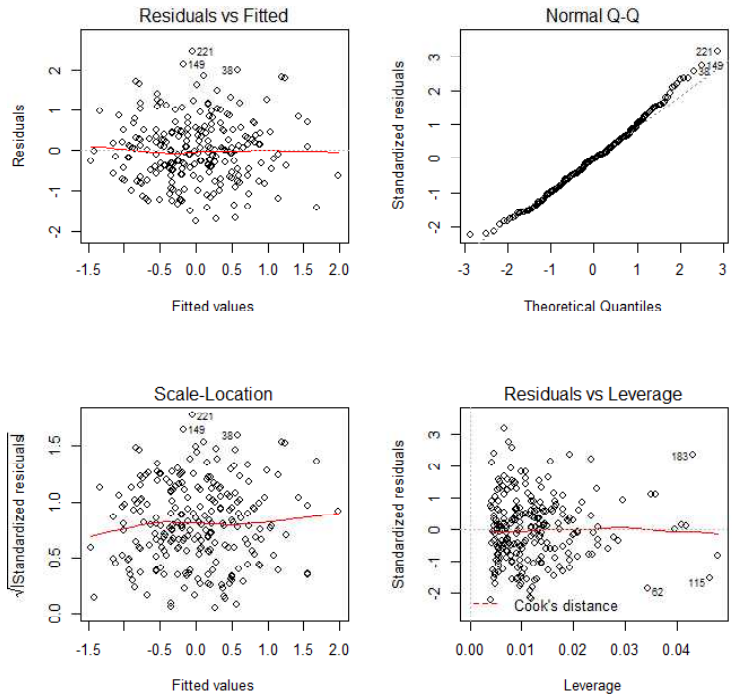
```
> # 信頼区間
```

```
> confint(result.3)
```

```
      2.5 %      97.5 %
```

```
(Intercept) -0.09910683 0.09910683
```

```
stress      0.51901839 0.71763782
```



## 重回帰分析 — glm関数

オブジェクト名 <- glm(基準変数 ~ 説明変数, family=gaussian, データフレーム名)  
summary(オブジェクト名)

glm() でモデルを指定し, 結果を「オブジェクト名」に保存する. その内容をsummary()で表示する.

```
> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  id stress kyoufu support utsu work result
1  1    20    2.2    17    18     0     0
2  2    23    4.8    18    21     1     0
3  3    30    5.8    12    29     1     1
4  4    25    5.2    18    29     0     1
5  5    26    2.0     8    22     1     0
6  6    21    5.0    26    19     1     0
```

1	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```
> # 記述統計量
> dtmp <- d1[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD stress kyoufu support  utsu  work result
stress 245 22.94 5.25  1.00  0.40 -0.34  0.62  0.03  0.44
kyoufu 245  4.05 1.17  0.40  1.00 -0.03  0.31  0.10  0.20
support 245 18.42 4.96 -0.34 -0.03  1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49  0.62  0.31 -0.51  1.00  0.02  0.76
work    245  0.50 0.50  0.03  0.10 -0.03  0.02  1.00 -0.08
result  245  0.26 0.44  0.44  0.20 -0.39  0.76 -0.08  1.00
```

```
> #偏回帰係数の推定
> result.1 <- glm(utsu ~ stress + kyoufu + support, family=gaussian, data=d1)
> summary(result.1)
```

```
Call:
glm(formula = utsu ~ stress + kyoufu + support, family = gaussian,
    data = d1)
```

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-11.0197   -3.3419    0.1646    3.0142   11.7647
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 13.22868    2.14218   6.175 2.78e-09 ***
stress       0.55796    0.06634   8.411 3.61e-15 ***
kyoufu       0.67134    0.28009   2.397  0.0173 *
support     -0.45934    0.06424  -7.150 1.03e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for gaussian family taken to be 21.58741)
```

```
Null deviance: 10288.0 on 244 degrees of freedom
Residual deviance: 5202.6 on 241 degrees of freedom
AIC: 1453.9
```

```
Number of Fisher Scoring iterations: 2
```

```

> confint(result.1)
Waiting for profiling to be done...
              2.5 %      97.5 %
(Intercept)  9.0300832 17.4272733
stress       0.4279414  0.6879808
kyoufu       0.1223773  1.2203088
support      -0.5852586 -0.3334245
>
>

> #標準偏回帰係数の推定
> d2 <- as.data.frame(scale(d1[,c("stress", "kyoufu", "support", "utsu")]))
> #共分散行列の確認
> round(cov(d2), 2)
      stress kyoufu support  utsu
stress    1.00   0.40   -0.34   0.62
kyoufu    0.40   1.00   -0.03   0.31
support   -0.34  -0.03   1.00  -0.51
utsu      0.62   0.31  -0.51   1.00

> result.2 <- glm(utsu ~ stress + kyoufu + support, family=gaussian, data=d2)
> summary(result.2)

Call:
glm(formula = utsu ~ stress + kyoufu + support, family = gaussian,
    data = d2)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.69706  -0.51466   0.02535   0.46419   1.81181

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.334e-17  4.571e-02   0.000   1.0000
stress       4.509e-01  5.361e-02   8.411 3.61e-15 ***
kyoufu       1.210e-01  5.048e-02   2.397  0.0173 *
support     -3.508e-01  4.907e-02  -7.150 1.03e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for gaussian family taken to be 0.5119875)

Null deviance: 244.00  on 244  degrees of freedom
Residual deviance: 123.39  on 241  degrees of freedom
AIC: 537.23

Number of Fisher Scoring iterations: 2

> confint(result.2)
Waiting for profiling to be done...
              2.5 %      97.5 %
(Intercept) -0.08959726  0.08959726
stress       0.34586463  0.55602983
kyoufu       0.02205570  0.21993258
support      -0.44701225 -0.25466489
>
>
>
>

```

## 重回帰分析 — sem関数

パッケージの読み込み  
library(sem)

## モデルの設定

## specifyEquations を使う方法

モデル名 &lt;- specifyEquations()

予測式

V(変数名.) = 分散, 誤差分散を表すパラメタ (または設定値)

C(変数名1, 変数名2) = 共分散を表すパラメタ (または設定値)

途中で行をあけると, モデル設定が終了したと判断されてしまう  
最終行に空白行を入れる。そうしないとモデル設定が終了したことにならない

## specifyModel を使う方法

モデル名 &lt;- specifyModel()

各パラメタ (または設定値) を設定する式

## パラメタ値の推定

semオブジェクト名 &lt;- sem(モデル名, S=共分散行列または相関係数行列, N=標本サイズ)

もしくは

semオブジェクト名 &lt;- sem(モデル名, data=データ行列, formula=~変数1+変数2+…)

## 【重要】 共分散行列や相関係数行列は, 分析に用いる観測変数だけのものにすること

余計な変数も含む共分散行列や相関係数行列を用いると, 解がおかしくなることがある

formula=~で, 共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ, 右辺の「~」の後ろに書く。

## 結果の出力

summary(semオブジェクト名)

## 標準化解の推定

stdCoef(semオブジェクト名)

あらかじめsemパッケージをインストールしておく必要がある。

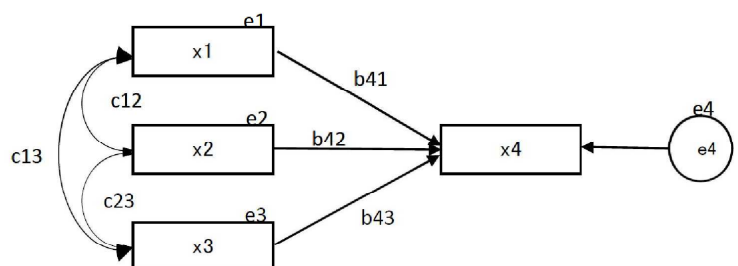
モデル部分のスキプトの例 (重回帰モデル  $x_4 = b_{41} * x_1 + b_{42} * x_2 + b_{43} * x_3 + e_4$ )

# specifyEquations を使う方法

```
seq.1 <- specifyEquations()
x4 = b41*x1 + b42*x2 + b43*x3      # 予測式
V(x4) = ev4                        # 内生変数の誤差分散
V(x1) = v1                          # 外生変数の分散
V(x2) = v2
V(x3) = v3
C(x1, x2) = c12                     # 外生変数の共分散
C(x1, x3) = c13
C(x2, x3) = c23
```

# specifyModel を使う方法

```
smd.1 <- specify.model()
x4 <- x1, b41, NA
x4 <- x2, b42, NA
x4 <- x3, b43, NA
x4 <-> x4, ev4, NA
x1 <-> x1, v1, NA
x2 <-> x2, v2, NA
```



```
x3 <-> x3, v3, NA
x1 <-> x2, c12, NA
x1 <-> x3, c13, NA
x2 <-> x3, c23, NA
```

```
# 1行目: x4に対するx1の偏回帰係数をb41とする (NAは、「初期値を特に指定しない」の意。以下同様)
# 2行目: x4に対するx2の偏回帰係数をb42とする
# 3行目: x4に対するx3の偏回帰係数をb43とする
# 4行目: x4の誤差分散をev4とする
# 5行目: x1の分散をv1とする
# 6行目: x2の分散をv2とする
# 7行目: x3の分散をv3とする
# 8行目: x1とx2の共分散をc12とする
# 9行目: x1とx3の共分散をc13とする
#10行目: x2とx3の共分散をc23とする
```

```
> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  id stress kyoufu support utsu work result
1  1     20    2.2      17   18      0      0
2  2     23    4.8      18   21      1      0
3  3     30    5.8      12   29      1      1
4  4     25    5.2      18   29      0      1
5  5     26    2.0       8   22      1      0
6  6     21    5.0      26   19      1      0
```

1	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```
> # 記述統計量
> dtmp <- d1[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD stress kyoufu support  utsu  work result
stress 245 22.94 5.25  1.00  0.40 -0.34  0.62  0.03  0.44
kyoufu 245  4.05 1.17  0.40  1.00 -0.03  0.31  0.10  0.20
support 245 18.42 4.96 -0.34 -0.03  1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49  0.62  0.31 -0.51  1.00  0.02  0.76
work    245  0.50 0.50  0.03  0.10 -0.03  0.02  1.00 -0.08
result  245  0.26 0.44  0.44  0.20 -0.39  0.76 -0.08  1.00
```

```
> #変数名の変更 (式を書くのにはこのほうが簡単)
> # x1: ストレス x2: 失敗恐怖 x3: ソーシャルサポート x4: うつ傾向
> colnames(d1) <- c("id", "x1", "x2", "x3", "x4", "work", "result")
> head(d1)
```

```
  id x1  x2 x3 x4 work result
1  1 20 2.2 17 18      0      0
2  2 23 4.8 18 21      1      0
3  3 30 5.8 12 29      1      1
4  4 25 5.2 18 29      0      1
5  5 26 2.0  8 22      1      0
6  6 21 5.0 26 19      1      0
```

```
> # 共分散行列・相関係数行列
> cov.d1 <- cov(d1[, c("x1", "x2", "x3", "x4")])
> cor.d1 <- cor(d1[, c("x1", "x2", "x3", "x4")])
>
>
```

```

> #sem パッケージの読み込み
> library(sem)
>
> # 適合度指標出力の指定
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RFI", "IFI", "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>
>
> # 構造方程式モデリング
> #  $x_4 = b_{41} * x_1 + b_{42} * x_2 + b_{43} * x_3 + e_4$ 
>
> # specifyEquations を使う方法
> # モデルの設定
> seq.1 <- specifyEquations()
1:  $x_4 = b_{41} * x_1 + b_{42} * x_2 + b_{43} * x_3$ 
2:  $V(x_4) = e_{v4}$ 
3:  $V(x_1) = v_1$ 
4:  $V(x_2) = v_2$ 
5:  $V(x_3) = v_3$ 
6:  $C(x_1, x_2) = c_{12}$ 
7:  $C(x_1, x_3) = c_{13}$ 
8:  $C(x_2, x_3) = c_{23}$ 
9:
Read 8 items

> # モデルの当てはめ（共分散行列）
> sem.seq.1 <- sem(seq.1, cov.d1, N=nrow(d1))
> summary(sem.seq.1)

Model Chi-square = 0 Df = 0 Pr(>ChiSq) = NA
Goodness-of-fit index = 1
AIC = 20
AICc = 0.9401709
BIC = 0
CAIC = 0

Normalized Residuals
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
-1.861e-15  0.000e+00  0.000e+00 -1.675e-17  0.000e+00  8.713e-16

R-square for Endogenous Variables
      x4
0.4943

Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
b41  0.5579611 0.06592870  8.4630984 2.603658e-17 x4 <--- x1
b42  0.6713431 0.27836253  2.4117582 1.587581e-02 x4 <--- x2
b43 -0.4593416 0.06384842 -7.1942506 6.280474e-13 x4 <--- x3
ev4  21.3219891 1.93040219 11.0453610 2.308379e-28 x4 <--> x4
v1   27.5413182 2.49347379 11.0453610 2.308379e-28 x1 <--> x1
v2   1.3695590 0.12399405 11.0453610 2.308379e-28 x2 <--> x2
v3   24.5971228 2.22691886 11.0453610 2.308379e-28 x3 <--> x3
c12  2.4793409 0.42400604  5.8474189 4.992591e-09 x2 <--> x1
c13 -8.7938274 1.75878193 -4.9999532 5.734423e-07 x3 <--> x1
c23 -0.1549716 0.37169984 -0.4169266 6.767321e-01 x3 <--> x2

Iterations = 0

> # 標準化解の表示
> (sc.sem.sew.1 <- stdCoef(sem.seq.1))
      Std. Estimate
b41 b41  0.45094723 x4 <--- x1
b42 b42  0.12099414 x4 <--- x2
b43 b43 -0.35083857 x4 <--- x3
ev4 ev4  0.50569259 x4 <--> x4
v1  v1  1.00000000 x1 <--> x1
v2  v2  1.00000000 x2 <--> x2
v3  v3  1.00000000 x3 <--> x3
c12 c12  0.40369513 x2 <--> x1
c13 c13 -0.33786517 x3 <--> x1
c23 c23 -0.02670051 x3 <--> x2

```



&gt;

&gt; # モデルの当てはめ (相関係数行列)

&gt; sem.seq.1 &lt;- sem(seq.1, cor.d1, N=nrow(d1))

&gt; summary(sem.seq.1)

Model Chisquare = -5.417888e-14 Df = 0 Pr(&gt;Chisq) = NA

Goodness-of-fit index = 1

AIC = 20

AICc = 0.9401709

BIC = -5.417888e-14

CAIC = -5.417888e-14

Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.475e-15	0.000e+00	0.000e+00	9.014e-18	0.000e+00	1.547e-15

R-square for Endogenous Variables

x4

0.4943

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )		
b41	0.45094723	0.05328394	8.4630984	2.603658e-17	x4	<--- x1
b42	0.12099414	0.05016844	2.4117582	1.587581e-02	x4	<--- x2
b43	-0.35083857	0.04876652	-7.1942506	6.280474e-13	x4	<--- x3
ev4	0.50569259	0.04578326	11.0453610	2.308379e-28	x4	<--> x4
v1	1.00000000	0.09053575	11.0453610	2.308379e-28	x1	<--> x1
v2	1.00000000	0.09053575	11.0453610	2.308379e-28	x2	<--> x2
v3	1.00000000	0.09053575	11.0453610	2.308379e-28	x3	<--> x3
c12	0.40369513	0.06903817	5.8474189	4.992591e-09	x2	<--> x1
c13	-0.33786517	0.06757367	-4.9999532	5.734423e-07	x3	<--> x1
c23	-0.02670051	0.06404126	-0.4169266	6.767321e-01	x3	<--> x2

Iterations = 0

&gt; (sc.sem.seq.1 &lt;- stdCoef(sem.seq.1))

		Std. Estimate		
b41	b41	0.45094723	x4	<--- x1
b42	b42	0.12099414	x4	<--- x2
b43	b43	-0.35083857	x4	<--- x3
ev4	ev4	0.50569259	x4	<--> x4
v1	v1	1.00000000	x1	<--> x1
v2	v2	1.00000000	x2	<--> x2
v3	v3	1.00000000	x3	<--> x3
c12	c12	0.40369513	x2	<--> x1
c13	c13	-0.33786517	x3	<--> x1
c23	c23	-0.02670051	x3	<--> x2

&gt; # specifyModel を使う方法

&gt; # モデルの設定

&gt; smd.1 &lt;- specifyModel()

```

1: x4 <- x1, b41, NA
2: x4 <- x2, b42, NA
3: x4 <- x3, b43, NA
4: x4 <-> x4, ev4, NA
5: x1 <-> x1, v1, NA
6: x2 <-> x2, v2, NA
7: x3 <-> x3, v3, NA
8: x1 <-> x2, c12, NA
9: x1 <-> x3, c13, NA
10: x2 <-> x3, c23, NA
11:

```

Read 10 records

NOTE: it is generally simpler to use specifyEquations() or cfa()  
 see ?specifyEquations

```

> # モデルの当てはめ（相関係数行列）
> sem.smd.1 <- sem(smd.1, cor.d1, N=nrow(d1))
> summary(sem.smd.1)

Model Chisquare = -5.417888e-14   Df =   0 Pr(>Chisq) = NA
Goodness-of-fit index =   1
AIC =   20
AICc =  0.9401709
BIC = -5.417888e-14
CAIC = -5.417888e-14

Normalized Residuals
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
-1.475e-15  0.000e+00  0.000e+00  9.014e-18  0.000e+00  1.547e-15

R-square for Endogenous Variables
      x4
0.4943

Parameter Estimates
      Estimate   Std Error   z value   Pr(>|z|)
b41  0.45094723  0.05328394   8.4630984 2.603658e-17 x4 <--- x1
b42  0.12099414  0.05016844   2.4117582 1.587581e-02 x4 <--- x2
b43 -0.35083857  0.04876652  -7.1942506 6.280474e-13 x4 <--- x3
ev4  0.50569259  0.04578326  11.0453610 2.308379e-28 x4 <--> x4
v1   1.00000000  0.09053575  11.0453610 2.308379e-28 x1 <--> x1
v2   1.00000000  0.09053575  11.0453610 2.308379e-28 x2 <--> x2
v3   1.00000000  0.09053575  11.0453610 2.308379e-28 x3 <--> x3
c12  0.40369513  0.06903817   5.8474189 4.992591e-09 x2 <--> x1
c13 -0.33786517  0.06757367  -4.9999532 5.734423e-07 x3 <--> x1
c23 -0.02670051  0.06404126  -0.4169266 6.767321e-01 x3 <--> x2

Iterations =   0

> (sc.sem.smd.1 <- stdCoef(sem.smd.1))
      Std. Estimate
b41 b41      0.45094723 x4 <--- x1
b42 b42      0.12099414 x4 <--- x2
b43 b43     -0.35083857 x4 <--- x3
ev4 ev4      0.50569259 x4 <--> x4
v1  v1      1.00000000 x1 <--> x1
v2  v2      1.00000000 x2 <--> x2
v3  v3      1.00000000 x3 <--> x3
c12 c12      0.40369513 x2 <--> x1
c13 c13     -0.33786517 x3 <--> x1
c23 c23     -0.02670051 x3 <--> x2

```

## 15 パス解析

## パス解析 — specifyEquations関数

パッケージの読み込み  
library(sem)

## 適合度指標の設定

```
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

## モデルの設定

```
モデル名 <- specifyEquations()
```

```
予測式
```

```
V(変数名.) = 分散, 誤差分散を表すパラメタ (または設定値)
```

```
C(変数名1, 変数名2) = 共分散表すパラメタ (または設定値)
```

## パラメタ値の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
```

```
もしくは
```

```
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+...)
```

## 【重要】 共分散行列は、分析に用いる観測変数だけのものにすること

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

## 結果の出力

```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

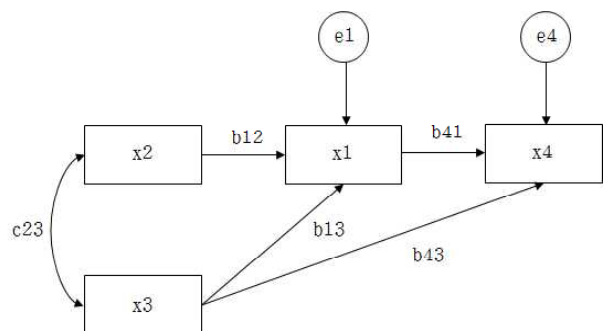
空白行があるところで、モデルの設定が終了したとみなされる。

(逆に、行をあけないと、モデル設定が終了したことになる)

## モデル部分のスキプトの例

```
パスモデル 1    x1 = b12 * x2 + b13 * x3 + e1      # x1についての回帰式
                  x4 = b41 * x1 + b43 * x3 + e4    # x4についての回帰式
```

```
seq.1 <- specifyEquations()
x1 = b12*x2 + b13*x3      # 予測式
x4 = b41*x1 + b43*x3
V(x1) = ev1               # 内生変数の誤差分散
V(x4) = ev4
V(x2) = v2                # 外生変数の分散
V(x3) = v3
C(x2, x3) = c23           # 外生変数の共分散
```



**パスモデル2** パスモデル1において、x2 と x3 の共分散を0と固定するモデル。

```
seq.1 <- specifyEquations()
x1 = b12*x2 + b13*x3          # 予測式
x4 = b41*x1 + b43*x3
V(x1) = ev1                   # 内生変数の誤差分散
V(x4) = ev4
V(x2) = v2                    # 外生変数の分散
V(x3) = v3
C(x2, x3) = 0                 # 外生変数の共分散
```

```
> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
```

```
  id stress kyoufu support utsu work result
1  1    20    2.2    17    18     0     0
2  2    23    4.8    18    21     1     0
3  3    30    5.8    12    29     1     1
4  4    25    5.2    18    29     0     1
5  5    26    2.0     8    22     1     0
6  6    21    5.0    26    19     1     0
```

1	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```
> # 記述統計量
> dtmp <- d1[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD stress kyoufu support  utsu  work result
stress 245 22.94 5.25  1.00  0.40  -0.34  0.62  0.03  0.44
kyoufu 245  4.05 1.17  0.40  1.00  -0.03  0.31  0.10  0.20
support 245 18.42 4.96 -0.34 -0.03  1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49  0.62  0.31 -0.51  1.00  0.02  0.76
work    245  0.50 0.50  0.03  0.10 -0.03  0.02  1.00 -0.08
result  245  0.26 0.44  0.44  0.20 -0.39  0.76 -0.08  1.00
```

```
> #変数名の変更（式を書くのにはこのほうが簡単）
> # x1:ストレス x2:失敗恐怖 x3:ソーシャルサポート x4:うつ傾向
> colnames(d1) <- c("id", "x1", "x2", "x3", "x4", "work", "result")
> head(d1)
```

```
  id x1  x2 x3 x4 work result
1  1 20 2.2 17 18     0     0
2  2 23 4.8 18 21     1     0
3  3 30 5.8 12 29     1     1
4  4 25 5.2 18 29     0     1
5  5 26 2.0  8 22     1     0
6  6 21 5.0 26 19     1     0
```

```
> # 共分散行列・相関係数行列
> cov.d1 <- cov(d1[, c("x1", "x2", "x3", "x4")])
> cor.d1 <- cor(d1[, c("x1", "x2", "x3", "x4")])
>
>
```

```
> #sem パッケージの読み込み
> library(sem)
```

```
> # 適合度指標出力の指定
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>
>
```

```
> # モデルの設定
> # specifyEquations を使う方法
> seq.1 <- specifyEquations()
1: x1 = b12*x2 + b13*x3
2: x4 = b41*x1 + b43*x3
3: V(x1) = ev1
4: V(x4) = ev4
5: V(x2) = v2
6: V(x3) = v3
7: C(x2, x3) = c23
8:
Read 7 items
```

```
> # 標準化解の推定
> sem.seq.1s <- sem(seq.1, S=cor.d1, N=nrow(d1))
> summary(sem.seq.1s)

Model Chisquare = 5.748331 Df = 1 Pr(>Chisq) = 0.01650433
Goodness-of-fit index = 0.9884923
Adjusted goodness-of-fit index = 0.8849227
RMSEA index = 0.1395004 90% CI: (0.0476635, 0.2587895)
Bentler-Bonett NFI = 0.9763773
Tucker-Lewis NNFI = 0.8799611
Bentler CFI = 0.9799935
Bentler RNI = 0.9799935
Bollen IFI = 0.9804063
SRMR = 0.03150648
AIC = 23.74833
AICc = 6.514288
BIC = 0.2470727
CAIC = -0.7529273
```

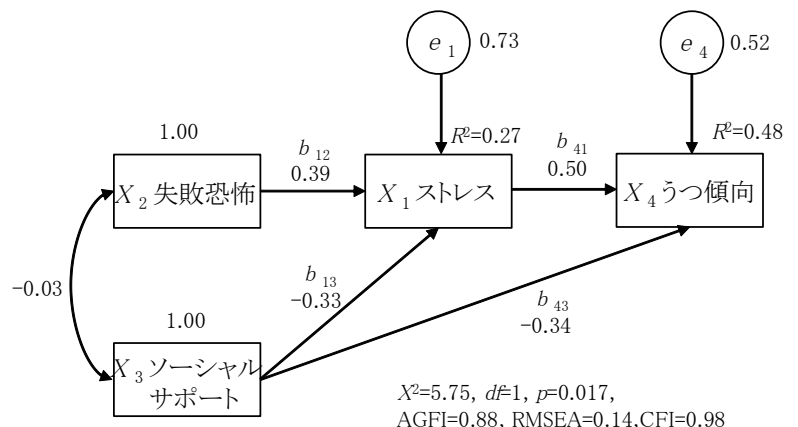
```
Normalized Residuals
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0000 0.0000  0.0000  0.1903  0.0000  1.5220
```

```
R-square for Endogenous Variables
      x1      x4
0.2700 0.4823
```

```
Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
b12  0.39495553 0.05471571  7.2183210 5.263335e-13 x1 <--- x2
b13 -0.32731966 0.05471571 -5.9821884 2.201594e-09 x1 <--- x3
b41  0.50485409 0.04894234 10.3152836 6.010359e-25 x4 <--- x1
b43 -0.33585592 0.04894234 -6.8622780 6.777099e-12 x4 <--- x3
ev1  0.72996847 0.06608824 11.0453610 2.308379e-28 x1 <--> x1
ev4  0.51774751 0.04687466 11.0453610 2.308379e-28 x4 <--> x4
v2   1.00000000 0.09053575 11.0453610 2.308379e-28 x2 <--> x2
v3   1.00000000 0.09053575 11.0453610 2.308379e-28 x3 <--> x3
c23 -0.02670051 0.06404126 -0.4169266 6.767321e-01 x3 <--> x2
```

```
Iterations = 0
```

```
>
>
> stdCoeef(sem.seq.1s)
      Std. Estimate
b12 b12  0.39495553 x1 <--- x2
b13 b13 -0.32731966 x1 <--- x3
b41 b41  0.50485409 x4 <--- x1
b43 b43 -0.33585592 x4 <--- x3
ev1 ev1  0.72996847 x1 <--> x1
ev4 ev4  0.51774751 x4 <--> x4
v2  v2   1.00000000 x2 <--> x2
v3  v3   1.00000000 x3 <--> x3
c23 c23 -0.02670051 x3 <--> x2
```



```

> # x1とx2の共分散を0に固定するモデル
> seq.2 <- specifyEquations()
1: x1 = b12*x2 + b13*x3
2: x4 = b41*x1 + b43*x3
3: V(x1) = ev1
4: V(x4) = ev4
5: V(x2) = v2
6: V(x3) = v3
7: C(x2, x3) = 0
8:
Read 7 items

> # 標準化解の推定
> sem.seq.2 <- sem(seq.2, S=cor.d1, N=nrow(d1))
> summary(sem.seq.2)

Model Chisquare = 5.922345 Df = 2 Pr(>Chisq) = 0.0517582
Goodness-of-fit index = 0.9881508
Adjusted goodness-of-fit index = 0.9407539
RMSEA index = 0.08965262 90% CI: (NA, 0.1772158)
Bentler-Bonett NFI = 0.9756622
Tucker-Lewis NNFI = 0.9504211
Bentler CFI = 0.9834737
Bentler RNI = 0.9834737
Bollen IFI = 0.9837476
SRMR = 0.03718342
AIC = 21.92234
AICc = 6.532514
BIC = -5.080172
CAIC = -7.080172

Normalized Residuals
      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-0.41710 -0.09514  0.02963  0.17210  0.10240  1.73600

R-square for Endogenous Variables
      x1      x4
0.2650 0.4795

Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
b12  0.3949555 0.05469620  7.220895 5.164635e-13 x1 <--- x2
b13 -0.3273197 0.05469620 -5.984322 2.172932e-09 x1 <--- x3
b41  0.5048541 0.04893927 10.315931 5.969982e-25 x4 <--- x1
b43 -0.3358559 0.04877005 -6.886520 5.717353e-12 x4 <--- x3
ev1  0.7299685 0.06608824 11.045361 2.308379e-28 x1 <--> x1
ev4  0.5177475 0.04687466 11.045361 2.308379e-28 x4 <--> x4
v2   1.0000000 0.09053575 11.045361 2.308379e-28 x2 <--> x2
v3   1.0000000 0.09053575 11.045361 2.308379e-28 x3 <--> x3

Iterations = 0
>
>
> stdCoef(sem.seq.2)
      Std. Estimate
b12 b12      0.3963259 x1 <--- x2
b13 b13     -0.3284554 x1 <--- x3
b41 b41      0.5044561 x4 <--- x1
b43 b43     -0.3367555 x4 <--- x3
ev1 ev1      0.7350428 x1 <--> x1
ev4 ev4      0.5205249 x4 <--> x4
v2  v2      1.0000000 x2 <--> x2
v3  v3      1.0000000 x3 <--> x3
      0.0000000 x3 <--> x2
>
>
>

```

## パス解析 — specifyModel関数

パッケージの読み込み  
library(sem)

## 適合度指標の設定

```
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

## モデルの設定

```
モデル名 <- specifyModel()
各パラメタ（または設定値）を設定する式
```

## パラメタ値の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
もしくは
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+...)
```

## 【重要】 共分散行列は、分析に用いる観測変数だけのものにすること

余計な変数も含む共分散行列や相関係数行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

## 結果の出力

```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

あらかじめsemパッケージをインストールしておく必要がある。  
 適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。  
 パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。  
 空白行があるところで、モデルの設定が終了したとみなされる。  
 （逆に、行をあげないと、モデル設定が終了したことにならない）

## モデル部分のスキプトの例

```
パスモデル 1    x1 = b12 * x2 + b13 * x3 + e1      # x1についての回帰式
                  x4 = b41 * x1 + b43 * x3 + e4      # x4についての回帰式
```

```
model.1 <- specifyModel()
```

```
x1 <- x2, b12, NA
x1 <- x3, b13, NA
x4 <- x1, b41, NA
x4 <- x3, b43, NA
x1 <-> x1, ev1, NA
x4 <-> x4, ev4, NA
x2 <-> x2, v2, NA
x3 <-> x3, v3, NA
x2 <-> x3, c23, NA
```

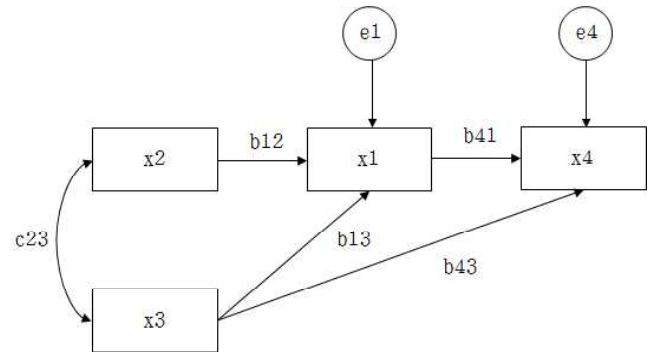
```
# 1行目：x1に対するx2の偏回帰係数をb12とする（NAは、初期値を特に指定しないの意。以下同様）
# 2行目：x1に対するx3の偏回帰係数をb13とする
# 3行目：x4に対するx1の偏回帰係数をb41とする
# 4行目：x4に対するx3の偏回帰係数をb43とする
# 5行目：x1の誤差分散をev1とする
# 6行目：x4の誤差分散をev4とする
# 7行目：x2の分散をv2とする
# 8行目：x3の分散をv3とする
# 9行目：x2とx3の共分散をc12とする
```



**パスモデル2** パスモデル1において、x2 と x3 の共分散を0と固定するモデル。

```
model.2 <- specifyModel()
x1 <- x2, b12, NA
x1 <- x3, b13, NA
x4 <- x1, b41, NA
x4 <- x3, b43, NA
x1 <-> x1, ev1, NA
x4 <-> x4, ev4, NA
x2 <-> x2, v2, NA
x3 <-> x3, v3, NA

# x2 <-> x3, c23, NA を削除
# x2 <-> x3, NA, 0 と指定しても同じ
```



```
> setwd("d:¥¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
```

```
  id stress kyoufu support utsu work result
1  1     20    2.2     17    18      0      0
2  2     23    4.8     18    21      1      0
3  3     30    5.8     12    29      1      1
4  4     25    5.2     18    29      0      1
5  5     26    2.0      8    22      1      0
6  6     21    5.0     26    19      1      0
```

```
>
>
> # 記述統計量
> dtmp <- d1[,c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean SD stress kyoufu support utsu work result
stress 245 22.94 5.25 1.00 0.40 -0.34 0.62 0.03 0.44
kyoufu 245 4.05 1.17 0.40 1.00 -0.03 0.31 0.10 0.20
support 245 18.42 4.96 -0.34 -0.03 1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49 0.62 0.31 -0.51 1.00 0.02 0.76
work    245 0.50 0.50 0.03 0.10 -0.03 0.02 1.00 -0.08
result  245 0.26 0.44 0.44 0.20 -0.39 0.76 -0.08 1.00
```

```
> #変数名の変更 (式を書くのにはこのほうが簡単)
> # x1:ストレス x2:失敗恐怖 x3:ソーシャルサポート x4:うつ傾向
> colnames(d1) <- c("id", "x1", "x2", "x3", "x4", "work", "result")
> head(d1)
```

```
  id x1 x2 x3 x4 work result
1  1 20 2.2 17 18      0      0
2  2 23 4.8 18 21      1      0
3  3 30 5.8 12 29      1      1
4  4 25 5.2 18 29      0      1
5  5 26 2.0 8 22      1      0
6  6 21 5.0 26 19      1      0
```

```
> # 共分散行列・相関係数行列
> cov.d1 <- cov(d1[,c("x1", "x2", "x3", "x4")])
> cor.d1 <- cor(d1[,c("x1", "x2", "x3", "x4")])
>
>
```

```
> #sem パッケージの読み込み
> library(sem)
```

```
> # 適合度指標出力の指定
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRM
```

```
R", "AIC", "AICc", "BIC", "CAIC"))
>
```

```
> # モデルの設定
> # specifyModel を使う方法
```

```
> smd.1 <- specifyModel()
1: x1 <- x2, b12, NA
2: x1 <- x3, b13, NA
3: x4 <- x1, b41, NA
4: x4 <- x3, b43, NA
5: x1 <-> x1, ev1, NA
6: x4 <-> x4, ev4, NA
7: x2 <-> x2, v2, NA
8: x3 <-> x3, v3, NA
9: x2 <-> x3, c23, NA
10:
Read 9 records
NOTE: it is generally simpler to use specifyEquations() or cfa()
      see ?specifyEquations
```

```
> # 標準化解の推定
> sem.smd.1 <- sem(smd.1, S=cor.d1, N=nrow(d1))
> summary(sem.smd.1)
```

```
Model Chisquare = 5.748331 Df = 1 Pr(>Chisq) = 0.01650433
Goodness-of-fit index = 0.9884923
Adjusted goodness-of-fit index = 0.8849227
RMSEA index = 0.1395004 90% CI: (0.0476635, 0.2587895)
Bentler-Bonett NFI = 0.9763773
Tucker-Lewis NNFI = 0.8799611
Bentler CFI = 0.9799935
Bentler RNI = 0.9799935
Bollen IFI = 0.9804063
SRMR = 0.03150648
AIC = 23.74833
AICc = 6.514288
BIC = 0.2470727
CAIC = -0.7529273
```

```
Normalized Residuals
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
0.0000 0.0000  0.0000  0.1903  0.0000  1.5220
```

```
R-square for Endogenous Variables
      x1      x4
0.2700 0.4823
```

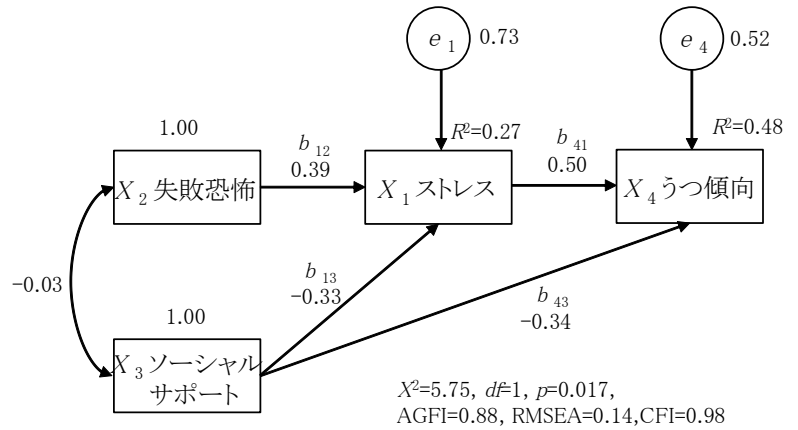
```
Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
b12  0.39495553 0.05471571  7.2183210 5.263335e-13 x1 <--- x2
b13 -0.32731966 0.05471571 -5.9821884 2.201594e-09 x1 <--- x3
b41  0.50485409 0.04894234 10.3152836 6.010359e-25 x4 <--- x1
b43 -0.33585592 0.04894234 -6.8622780 6.777099e-12 x4 <--- x3
ev1  0.72996847 0.06608824 11.0453610 2.308379e-28 x1 <--> x1
ev4  0.51774751 0.04687466 11.0453610 2.308379e-28 x4 <--> x4
v2   1.00000000 0.09053575 11.0453610 2.308379e-28 x2 <--> x2
v3   1.00000000 0.09053575 11.0453610 2.308379e-28 x3 <--> x3
c23 -0.02670051 0.06404126 -0.4169266 6.767321e-01 x3 <--> x2
```

```
Iterations = 0
>
> stdCoef(sem.seq.1)
```

```

Std. Estimate
1      0.4373010 x1 <--- f1
2      0.6293252 x2 <--- f1
3      0.3688269 x3 <--- f1
4      0.6154132 x7 <--- f3
5      0.3753166 x8 <--- f3
6      0.4116027 x9 <--- f3
7      0.5232285 f3 <--- f1
8      0.8087679 x1 <--> x1
9      0.6039498 x2 <--> x2
10     0.8639667 x3 <--> x3
11     0.6212666 x7 <--> x7
12     0.8591374 x8 <--> x8
13     0.8305832 x9 <--> x9
14     0.7262319 f3 <--> f3
15     1.0000000 f1 <--> f1
>
>

```



### > # x1とx2の共分散を0に固定するモデル

```
> smd.2 <- specifyModel()
```

```

1: x1 <- x2, b12, NA
2: x1 <- x3, b13, NA
3: x4 <- x1, b41, NA
4: x4 <- x3, b43, NA
5: x1 <-> x1, ev1, NA
6: x4 <-> x4, ev4, NA
7: x2 <-> x2, v2, NA
8: x3 <-> x3, v3, NA
9:

```

```
Read 8 records
```

```
NOTE: it is generally simpler to use specifyEquations() or cfa()
      see ?specifyEquations
```

### > # 標準化解の推定

```
> sem.smd.2 <- sem(smd.2, S=cor.d1, N=nrow(d1))
```

```
> summary(sem.smd.2)
```

```

Model Chisquare = 5.922345 Df = 2 Pr(>Chisq) = 0.0517582
Goodness-of-fit index = 0.9881508
Adjusted goodness-of-fit index = 0.9407539
RMSEA index = 0.08965262 90% CI: (NA, 0.1772158)
Bentler-Bonett NFI = 0.9756622
Tucker-Lewis NNFI = 0.9504211
Bentler CFI = 0.9834737
Bentler RNI = 0.9834737
Bollen IFI = 0.9837476
SRMR = 0.03718342
AIC = 21.92234
AICc = 6.532514
BIC = -5.080172
CAIC = -7.080172

```

#### Normalized Residuals

```

Min. 1st Qu. Median Mean 3rd Qu. Max.
-0.41710 -0.09514 0.02963 0.17210 0.10240 1.73600

```

#### R-square for Endogenous Variables

```

x1 x4
0.2650 0.4795

```

#### Parameter Estimates

```

Estimate Std Error z value Pr(>|z|)
b12 0.3949555 0.05469620 7.220895 5.164635e-13 x1 <--- x2
b13 -0.3273197 0.05469620 -5.984322 2.172932e-09 x1 <--- x3
b41 0.5048541 0.04893927 10.315931 5.969982e-25 x4 <--- x1
b43 -0.3358559 0.04877005 -6.886520 5.717353e-12 x4 <--- x3
ev1 0.7299685 0.06608824 11.045361 2.308379e-28 x1 <--> x1
ev4 0.5177475 0.04687466 11.045361 2.308379e-28 x4 <--> x4
v2 1.0000000 0.09053575 11.045361 2.308379e-28 x2 <--> x2

```

```
v3 1.0000000 0.09053575 11.045361 2.308379e-28 x3 <--> x3
```

```
Iterations = 0
```

```
> stdCoef(sem.smd.2)
      Std. Estimate
b12 b12 0.3963259 x1 <--- x2
b13 b13 -0.3284554 x1 <--- x3
b41 b41 0.5044561 x4 <--- x1
b43 b43 -0.3367555 x4 <--- x3
ev1 ev1 0.7350428 x1 <--> x1
ev4 ev4 0.5205249 x4 <--> x4
v2 v2 1.0000000 x2 <--> x2
v3 v3 1.0000000 x3 <--> x3
>
>
```

## 多母集団のパス解析

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

## 各群のモデルの設定 (specifyEquations関数)

```
モデル名1 <- specifyEquations()
  構造方程式
```

```
モデル名2 <- specifyEquations()
  構造方程式
...
```

## 各群のモデルの設定 (specifyModel関数)

```
モデル名1 <- specifyModel()
  構造方程式
```

```
モデル名2 <- specifyModel()
  構造方程式
...
```

各群のモデルの途中で行をあけると、モデル設定が終了したと判断されてしまう  
各群のモデルの最終行に空白行を入れる。そうしないとモデル設定が終了したことにならない

## 群分け変数のfactor化

```
群分け変数名 <- factor(群分け変数名)
```

## 分析モデルの設定

```
mgオブジェクト名 <- multigroupModel(モデル名1, モデル名2, ..., groups=levels(群分け変数名))
```

## パラメタ値の推定

```
semオブジェクト名 <- sem(モデル名, S=list(共分散行列1, 共分散行列2...),
                          N=c(標本サイズ1, 標本サイズ2, ...))
```

```
もしくは
semオブジェクト名 <- sem(mgオブジェクト名, data=データ行列, group="群分け変数",
                          formula=~変数1+変数2+...)
```

## 【重要】 共分散行列は、分析に用いる観測変数だけのものにする

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

## 結果の出力

```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

群分け変数名に「group」を用いると不具合が生じることがあるので他の名前を使う  
群分け変数のfactor化で、levels=c(水準1, 水準2, ...), labels=c(ラベル1, ラベル2, ...) を設定してもよい  
各群のモデルで、同じパラメタ名にすると等値制約になる  
群分け変数名はfactor関数を使ってfactor化しておかなければならない  
等値制約を加える場合は、例えば、b12.1 と b12.2 をともに b12というパラメタ名にすると、両群に共通な b12 というパラメタ値を推定する

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

```

> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  id stress kyoufu support utsu work result
1  1     20    2.2      17   18     0      0
2  2     23    4.8      18   21     1      0
3  3     30    5.8      12   29     1      1
4  4     25    5.2      18   29     0      1
5  5     26    2.0       8   22     1      0
6  6     21    5.0      26   19     1      0
>
>
> # 群別データ
> d11 <- d1[d1$work==1, colnames(d1) %in% c("work")==F]
> d10 <- d1[d1$work==0, colnames(d1) %in% c("work")==F]
>
>
> # 記述統計量
> dtmp <- d11[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD stress kyoufu support  utsu result
stress 122 23.10 5.08  1.00  0.35 -0.17  0.53  0.33
kyoufu 122  4.17 1.20  0.35  1.00  0.14  0.27  0.17
support 122 18.25 4.56 -0.17  0.14  1.00 -0.38 -0.31
utsu    122 20.44 5.49  0.53  0.27 -0.38  1.00  0.75
result  122  0.22 0.42  0.33  0.17 -0.31  0.75  1.00
>
> dtmp <- d10[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD stress kyoufu support  utsu result
stress 123 22.78 5.43  1.00  0.46 -0.47  0.69  0.54
kyoufu 123  3.93 1.13  0.46  1.00 -0.17  0.35  0.25
support 123 18.59 5.34 -0.47 -0.17  1.00 -0.59 -0.45
utsu    123 20.13 7.38  0.69  0.35 -0.59  1.00  0.79
result  123  0.29 0.46  0.54  0.25 -0.45  0.79  1.00
>
>
> #変数名の変更（式を書くのにはこのほうが簡単）
> # x1:ストレス x2:失敗恐怖 x3:ソーシャルサポート x4:うつ傾向
> colnames(d1) <- c("id", "x1", "x2", "x3", "x4", "work", "result")
> head(d1)
  id x1 x2 x3 x4 work result
1  1 20 2.2 17 18     0      0
2  2 23 4.8 18 21     1      0
3  3 30 5.8 12 29     1      1
4  4 25 5.2 18 29     0      1
5  5 26 2.0  8 22     1      0
6  6 21 5.0 26 19     1      0
>
>
>
> #sem パッケージの読み込み
> library(sem)
>
> # 適合度指標出力の指定
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>

```

```

>
> # モデルの設定
> # specifyEquations を使う方法
>
> seq.1 <- specifyEquations()
1: x1 = b12.1*x2 + b13.1*x3
2: x4 = b41.1*x1 + b43.1*x3
3: V(x1) = ev1.1
4: V(x4) = ev4.1
5: V(x2) = v2.1
6: V(x3) = v3.1
7: C(x2, x3) = c23.1
8:
Read 7 items

> seq.2 <- specifyEquations()
1: x1 = b12.2*x2 + b13.2*x3
2: x4 = b41.2*x1 + b43.2*x3
3: V(x1) = ev1.2
4: V(x4) = ev4.2
5: V(x2) = v2.2
6: V(x3) = v3.2
7: C(x2, x3) = c23.2
8:
Read 7 items

> # 多母集団分析
> d1$work <- factor(d1$work)
> mg.seq <- multigroupModel(seq.1, seq.2, groups=levels(d1$work))
> sem.mg.seq <- sem(mg.seq, data=d1, group="work", formula=~x1+x2+x3+x4)
> summary(sem.mg.seq)

```

```

Model Chisquare = 6.042957 Df = 2 Pr(>Chisq) = 0.04872913
Chisquare (null model) = 246.2831 Df = 12
Goodness-of-fit index = 0.9880118
Adjusted goodness-of-fit index = 0.9040942
RMSEA index = 0.1289871 90% CI: (0.008417286, 0.2528073)
Bentler-Bonett NFI = 0.9754634
Tucker-Lewis NNFI = 0.8964597
Bentler CFI = 0.9827433
Bentler RNI = 0.9827433
Bollen IFI = 0.9834497
SRMR = 0.0314776
AIC = 42.04296
AICc = 9.069505
BIC = -4.95956

```

```

Iterations: initial fits, 0 0 final fit, 0

```

```

work: 0

```

```

Model Chisquare = 0.9026323 Df = 1 Pr(>Chisq) = 0.3420769
Goodness-of-fit index = 0.9963279
Adjusted goodness-of-fit index = 0.9632787
RMSEA index = 0 90% CI: (NA, 0.23476)
Bentler-Bonett NFI = 0.9944273
Tucker-Lewis NNFI = 1.003746
Bentler CFI = 1
Bentler RNI = 1.000624
Bollen IFI = 1.000605
SRMR = 0.01590207
AIC = 18.90263
AICc = 2.495553
BIC = -3.909552
CAIC = -4.909552

```

```

Normalized Residuals

```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.0000	0.0000	0.0665	0.0000	0.5320

R-square for Endogenous Variables

x1	x4
0.3704	0.5634

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )			
b12.1	1.8791914	0.34991236	5.370463	7.853478e-08	x1	<---	x2
b13.1	-0.4114030	0.07410181	-5.551862	2.826424e-08	x1	<---	x3
b41.1	0.7164477	0.09212589	7.776834	7.436204e-15	x4	<---	x1
b43.1	-0.4709965	0.09366555	-5.028493	4.943508e-07	x4	<---	x3
ev1.1	18.5541031	2.37560947	7.810250	5.707481e-15	x1	<-->	x1
ev4.1	23.7533710	3.04130751	7.810250	5.707481e-15	x4	<-->	x4
v2.1	1.2784699	0.16369130	7.810250	5.707481e-15	x2	<-->	x2
v3.1	28.5069972	3.64994698	7.810250	5.707481e-15	x3	<-->	x3
c23.1	-1.0180328	0.55428088	-1.836673	6.625817e-02	x3	<-->	x2

work: 1

Model Chisquare = 5.140324 Df = 1 Pr(&gt;Chisq) = 0.02337628

Goodness-of-fit index = 0.9796275

Adjusted goodness-of-fit index = 0.7962752

RMSEA index = 0.1849799 90% CI: (0.05473528, 0.3556438)

Bentler-Bonett NFI = 0.9390293

Tucker-Lewis NNFI = 0.6827666

Bentler CFI = 0.9471278

Bentler RNI = 0.9471278

Bollen IFI = 0.9503011

SRMR = 0.04718079

AIC = 23.14032

AICc = 6.747467

BIC = 0.3363034

CAIC = -0.6636966

Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.0000	0.0000	0.0000	0.2036	0.0000	1.6286

R-square for Endogenous Variables

x1	x4
0.1685	0.3628

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )			
b12.2	1.5940497	0.35351596	4.509131	6.509380e-06	x1	<---	x2
b13.2	-0.2464606	0.09316002	-2.645562	8.155534e-03	x1	<---	x3
b41.2	0.5150020	0.07959184	6.470538	9.765471e-11	x4	<---	x1
b43.2	-0.3567843	0.08859546	-4.027118	5.646481e-05	x4	<---	x3
ev1.2	21.4329344	2.75552240	7.778175	7.357848e-15	x1	<-->	x1
ev4.2	19.1893399	2.46707498	7.778175	7.357848e-15	x4	<-->	x4
v2.2	1.4446416	0.18573016	7.778175	7.357848e-15	x2	<-->	x2
v3.2	20.8026690	2.67449242	7.778175	7.357848e-15	x3	<-->	x3
c23.2	0.7534345	0.50304910	1.497736	1.342020e-01	x3	<-->	x2

&gt; stdCoef(sem.mg.seq)

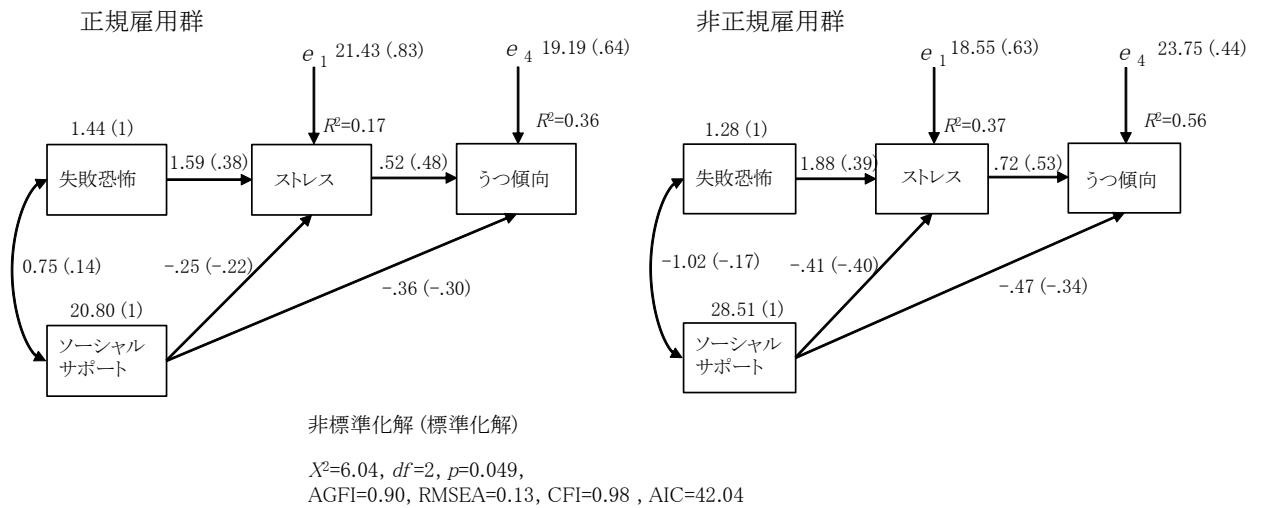
Group: 0

	Std. Estimate			
1 b12.1	0.3914194	x1	<---	x2
2 b13.1	-0.4046404	x1	<---	x3
3 b41.1	0.5272571	x4	<---	x1
4 b43.1	-0.3409239	x4	<---	x3
5 ev1.1	0.6296396	x1	<-->	x1
6 ev4.1	0.4365693	x4	<-->	x4
7 v2.1	1.0000000	x2	<-->	x2
8 v3.1	1.0000000	x3	<-->	x3
9 c23.1	-0.1686323	x3	<-->	x2



Group: 1  
 Std. Estimate

1	b12.2	0.3773805	x1	<---	x2
2	b13.2	-0.2214137	x1	<---	x3
3	b41.2	0.4764409	x4	<---	x1
4	b43.2	-0.2965261	x4	<---	x3
5	ev1.2	0.8315277	x1	<-->	x1
6	ev4.2	0.6371700	x4	<-->	x4
7	v2.2	1.0000000	x2	<-->	x2
8	v3.2	1.0000000	x3	<-->	x3
9	c23.2	0.1374377	x3	<-->	x2
>					
9	c23.2	-0.1686323	x3	<-->	x2



## 多母集団パス解析におけるパス係数の検定

## 制約モデル

b12 と b43 をモデル間で共通にする

```
seq.1 <- specifyEquations()
x1 = b12*x2 + b13.1*x3
x4 = b41.1*x1 + b43*x3
V(x1) = ev1.1
V(x4) = ev4.1
V(x2) = v2.1
V(x3) = v3.1
C(x2, x3) = c23.1

seq.2 <- specifyEquations()
x1 = b12*x2 + b13.2*x3
x4 = b41.2*x1 + b43*x3
V(x1) = ev1.2
V(x4) = ev4.2
V(x2) = v2.2
V(x3) = v3.2
C(x2, x3) = c23.2

d1$work <- factor(d1$work)
mg.seq <- multigroupModel(seq.1, seq.2, groups=levels(d1$work))
sem.mg.seq <- sem(mg.seq, data=d1, group="work", formula=~x1+x2+x3+x4)
```

## パス係数の検定

 $H_0: \beta_{13.1} = \beta_{13.2}$  など

# パス係数の差の値

```
(vb <- sem.mg.seq$coeff)
(vb1 <- vb[c("b12.1", "b13.1", "b41.1", "b43.1")])
(vb2 <- vb[c("b12.2", "b13.2", "b41.2", "b43.2")])
(vbd <- vb1-vb2)
```

# パス係数の差の標準誤差の推定

```
(vse <- sqrt(diag(sem.mg.seq$vcov)))
(vve <- diag(sem.mg.seq$vcov))
(vve1 <- vve[c("b12.1", "b13.1", "b41.1", "b43.1")])
(vve2 <- vve[c("b12.2", "b13.2", "b41.2", "b43.2")])
(vse12 <- sqrt(vve1 + vve2))
```

# 検定

```
(vz <- (vb1 - vb2) / vse12)
2*(1-pnorm(abs(vz)))
```

# 信頼区間の推定

```
(z0 <- qnorm(.975))
(vL <- (vb1 - vb2) - z0*vse12)
(vU <- (vb1 - vb2) + z0*vse12)
```

```
> setwd("d:¥¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id stress kyoufu support utsu work result
1  1     20    2.2      17   18      0      0
2  2     23    4.8      18   21      1      0
3  3     30    5.8      12   29      1      1
4  4     25    5.2      18   29      0      1
5  5     26    2.0       8   22      1      0
6  6     21    5.0      26   19      1      0
>
```

```

> # 群別データ
> d11 <- d1[d1$work==1, colnames(d1) %in% c("work")==F]
> d10 <- d1[d1$work==0, colnames(d1) %in% c("work")==F]
>
> # 記述統計量
> dtmp <- d11[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD stress kyoufu support  utsu result
stress 122 23.10 5.08  1.00  0.35 -0.17  0.53  0.33
kyoufu 122  4.17 1.20  0.35  1.00  0.14  0.27  0.17
support 122 18.25 4.56 -0.17  0.14  1.00 -0.38 -0.31
utsu    122 20.44 5.49  0.53  0.27 -0.38  1.00  0.75
result  122  0.22 0.42  0.33  0.17 -0.31  0.75  1.00
>
> dtmp <- d10[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean  SD stress kyoufu support  utsu result
stress 123 22.78 5.43  1.00  0.46 -0.47  0.69  0.54
kyoufu 123  3.93 1.13  0.46  1.00 -0.17  0.35  0.25
support 123 18.59 5.34 -0.47 -0.17  1.00 -0.59 -0.45
utsu    123 20.13 7.38  0.69  0.35 -0.59  1.00  0.79
result  123  0.29 0.46  0.54  0.25 -0.45  0.79  1.00
>
> #変数名の変更（式を書くのにはこのほうが簡単）
> # x1:ストレス x2:失敗恐怖 x3:ソーシャルサポート x4:うつ傾向
> colnames(d1) <- c("id", "x1", "x2", "x3", "x4", "work", "result")
> head(d1)
  id x1 x2 x3 x4 work result
1  1 20 2.2 17 18   0      0
2  2 23 4.8 18 21   1      0
3  3 30 5.8 12 29   1      1
4  4 25 5.2 18 29   0      1
5  5 26 2.0  8 22   1      0
6  6 21 5.0 26 19   1      0
>
>
> # 多母集団分析
> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>
>
> # 制約モデル
> # b12 と b43 をモデル間で共通にする
>
> seq.1 <- specifyEquations()
1: x1 = b12*x2 + b13.1*x3
2: x4 = b41.1*x1 + b43*x3
3: V(x1) = ev1.1
4: V(x4) = ev4.1
5: V(x2) = v2.1
6: V(x3) = v3.1
7: C(x2, x3) = c23.1
8:
Read 7 items

```

```

> seq.2 <- specifyEquations()
1: x1 = b12*x2 + b13.2*x3
2: x4 = b41.2*x1 + b43*x3
3: V(x1) = ev1.2
4: V(x4) = ev4.2
5: V(x2) = v2.2
6: V(x3) = v3.2
7: C(x2, x3) = c23.2
8:
Read 7 items

> d1$work <- factor(d1$work)
> mg.seq <- multigroupModel(seq.1, seq.2, groups=levels(d1$work))
> sem.mg.seq <- sem(mg.seq, data=d1, group="work", formula=~x1+x2+x3+x4)
> summary(sem.mg.seq)

```

```

Model Chisquare = 7.154842 Df = 4 Pr(>Chisq) = 0.1279283
Chisquare (null model) = 246.2831 Df = 12
Goodness-of-fit index = 0.9856524
Adjusted goodness-of-fit index = 0.9426097
RMSEA index = 0.0805695 90% CI: (NA, 0.1744588)
Bentler-Bonett NFI = 0.9709487
Tucker-Lewis NNFI = 0.9596022
Bentler CFI = 0.9865341
Bentler RNI = 0.9865341
Bollen IFI = 0.9869787
SRMR = 0.03869427
AIC = 39.15484
AICc = 9.540807
BIC = -14.85019

```

```
Iterations: initial fits, 0 0 final fit, 41
```

```
work: 0
```

```

Model Chisquare = 1.478764 Df = 1 Pr(>Chisq) = 0.223968
Goodness-of-fit index = 0.9941505
Adjusted goodness-of-fit index = 0.9415052
RMSEA index = 0.06264419 90% CI: (NA, 0.2589937)
Bentler-Bonett NFI = 0.9908704
Tucker-Lewis NNFI = 0.981583
Bentler CFI = 0.9969305
Bentler RNI = 0.9969305
Bollen IFI = 0.9970258
SRMR = 0.02701648
AIC = 19.47876
AICc = 3.071684
BIC = -3.33342
CAIC = -4.33342

```

```
Normalized Residuals
```

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.33145	0.00000	0.05265	0.11760	0.23191	0.68641	

```
R-square for Endogenous Variables
```

	x1	x4
0.3562	0.5501	

```
Parameter Estimates
```

	Estimate	Std Error	z value	Pr(> z )	
b12	1.7380842	0.24885864	6.984223	2.864368e-12	x1 <--- x2
b13.1	-0.4164422	0.07362779	-5.656047	1.548991e-08	x1 <--- x3
b41.1	0.7443567	0.08707909	8.548054	1.251811e-17	x4 <--- x1
b43	-0.4107062	0.06032937	-6.807732	9.914902e-12	x4 <--- x3
ev1.1	18.5789349	2.37878885	7.810250	5.707481e-15	x1 <--> x1
ev4.1	23.8340009	3.05163111	7.810250	5.707481e-15	x4 <--> x4
v2.1	1.2784699	0.16369130	7.810250	5.707481e-15	x2 <--> x2
v3.1	28.5069972	3.64994698	7.810250	5.707481e-15	x3 <--> x3
c23.1	-1.0180328	0.55428088	-1.836673	6.625817e-02	x3 <--> x2

```
work: 1
```

Model Chisquare = 5.676078 Df = 1 Pr(>Chisq) = 0.01719777  
 Goodness-of-fit index = 0.9770847  
 Adjusted goodness-of-fit index = 0.7708467  
 RMSEA index = 0.196584 90% CI: (0.06622647, 0.3661187)  
 Bentler-Bonett NFI = 0.9326746  
 Tucker-Lewis NNFI = 0.6417169  
 Bentler CFI = 0.9402862  
 Bentler RNI = 0.9402862  
 Bollen IFI = 0.9438701  
 SRMR = 0.05046779  
 AIC = 23.67608  
 AICc = 7.283221  
 BIC = 0.8720571  
 CAIC = -0.1279429

## Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-0.3399	-0.1436	0.0000	0.1575	0.1087	1.5255

## R-square for Endogenous Variables

x1	x4
0.1896	0.3830

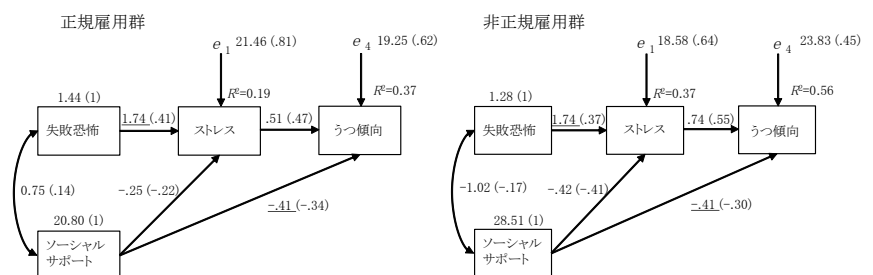
## Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
b12	1.7380842	0.24885864	6.984223	2.864368e-12	x1 <--- x2
b13.2	-0.2516773	0.08818425	-2.853993	4.317342e-03	x1 <--- x3
b41.2	0.5067888	0.09277849	5.462352	4.698659e-08	x4 <--- x1
b43	-0.4107062	0.06032937	-6.807732	9.914902e-12	x4 <--- x3
ev1.2	21.4625255	2.75932679	7.778175	7.357848e-15	x1 <--> x1
ev4.2	19.2480001	2.47461661	7.778175	7.357848e-15	x4 <--> x4
v2.2	1.4446416	0.18573016	7.778175	7.357848e-15	x2 <--> x2
v3.2	20.8026690	2.67449242	7.778175	7.357848e-15	x3 <--> x3
c23.2	0.7534345	0.50304910	1.497736	1.342020e-01	x3 <--> x2

> stdCoef(sem.mg.seq)

Group: 0

	Std. Estimate	
1 b12	0.3658290	x1 <--- x2
2 b13.1	-0.4138973	x1 <--- x3
3 b41.1	0.5494062	x4 <--- x1
4 b43	-0.3012877	x4 <--- x3
5 ev1.1	0.6437911	x1 <--> x1
6 ev4.1	0.4499310	x4 <--> x4
7 v2.1	1.0000000	x2 <--> x2
8 v3.1	1.0000000	x3 <--> x3
9 c23.1	-0.1686323	x3 <--> x2



非標準化解 (標準化解)

$\chi^2=7.15$ ,  $df=4$ ,  $p=0.128$ ,  
 AGFI=0.94, RMSEA=0.08, CFI=0.99, AIC=39.15

Group: 1

	Std. Estimate	
1 b12	0.4059282	x1 <--- x2
2 b13.2	-0.2230498	x1 <--- x3
3 b41.2	0.4669772	x4 <--- x1
4 b43	-0.3353959	x4 <--- x3
5 ev1.2	0.8103589	x1 <--> x1
6 ev4.2	0.6170487	x4 <--> x4
7 v2.2	1.0000000	x2 <--> x2
8 v3.2	1.0000000	x3 <--> x3
9 c23.2	0.1374377	x3 <--> x2

>

```

> # パス係数の検定
> (vb <- sem.mg.seq$coeff)
      b12      b13.1      b41.1      b43      ev1.1      ev4.1      v2.1
1. 7380842 -0.4164422  0.7443567 -0.4107062 18.5789349 23.8340009 1.2784699
      v3.1      c23.1      b13.2      b41.2      ev1.2      ev4.2      v2.2
28.5069972 -1.0180328 -0.2516773  0.5067888 21.4625255 19.2480001 1.4446416
      v3.2      c23.2
20.8026690  0.7534345
>

> # 関心のあるパス係数の取り出し
> (vb1 <- vb[c("b13.1", "b41.1")])
      b13.1      b41.1
-0.4164422  0.7443567
> (vb2 <- vb[c("b13.2", "b41.2")])
      b13.2      b41.2
-0.2516773  0.5067888
>

> # パス係数の差
> (vbd <- vb1-vb2)
      b13.1      b41.1
-0.1647649  0.2375679
>
>
> # パス係数の標準誤差
> (vse <- sqrt(diag(sem.mg.seq$vcov)))
      b12      b13.1      b41.1      b43      ev1.1      ev4.1      v2.1
0.24919533 0.07362920 0.08675249 0.06467567 2.38087824 3.05715764 0.16372405
      v3.1      c23.1      b13.2      b41.2      ev1.2      ev4.2      v2.2
3.65067675 0.55428201 0.09277965 0.07917452 2.76178042 2.47910233 0.18576895
      v3.2      c23.2
2.67505003 0.50304583
>
>
> # パス係数の標準誤差の2乗
> (vve <- diag(sem.mg.seq$vcov))
      b12      b13.1      b41.1      b43      ev1.1      ev4.1
0.062098310 0.005421260 0.007525994 0.004182942 5.668581192 9.346212836
      v2.1      v3.1      c23.1      b13.2      b41.2      ev1.2
0.026805563 13.327440737 0.307228541 0.008608063 0.006268604 7.627431081
      ev4.2      v2.2      v3.2      c23.2
6.145948342 0.034510101 7.155892678 0.253055110
>
>
> # 関心のあるパス係数の標準誤差の2乗の取り出し
> (vve1 <- vve[c("b13.1", "b41.1")])
      b13.1      b41.1
0.005421260 0.007525994
> (vve2 <- vve[c("b13.2", "b41.2")])
      b13.2      b41.2
0.008608063 0.006268604
>
>
> # パス係数の差の標準誤差
> vse12 <- sqrt(vve1 + vve2)
> names(vse12) <- c("b13", "b41")
> vse12
      b13      b41
0.1184454 0.1174504
>

> # 検定統計量
> vz <- (vb1 - vb2) / vse12
> names(vz) <- c("b13", "b41")
> vz
      b13      b41
-1.391062  2.022709
>

```

```

> # p値
> vp <- 2*(1-pnorm(abs(vz)))
> names(vp) <- c("b13", "b41")
> vp
      b13      b41
0.1642067 0.0431032
>
>
> # 信頼区間の推定
> (z0 <- qnorm(.975))
[1] 1.959964
> vL <- (vb1 - vb2) - z0*vse12
> vU <- (vb1 - vb2) + z0*vse12
> names(vL) <- c("b13L", "b41L")
> names(vU) <- c("b13U", "b41U")
> vL
      b13L      b41L
-0.396913743 0.007369375
> vU
      b13U      b41U
0.06738386 0.46776651
>
>
>

```

## 16 因子分析



## 確認的因子分析

### パッケージの読み込み

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

### モデルの設定

#### specifyEquations を使う方法

```
モデル名 <- specifyEquations()
予測式
V(変数名.) = 分散, 誤差分散を表すパラメタ (または設定値)
C(変数名1, 変数名2) = 共分散表すパラメタ (または設定値)
```

#### specifyModel を使う方法

```
モデル名 <- specifyModel()
各パラメタ (または設定値) を設定する式
```

### パラメタ値の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
```

#### もしくは

```
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+…)
```

#### 【重要】 共分散行列は、分析に用いる観測変数だけのものにする

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

### 結果の出力

```
summary(semオブジェクト名)
```

### 標準化解の推定

```
stdCoef(semオブジェクト名)
```

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

空白行があるところで、モデルの設定が終了したとみなされる。

(逆に、行をあけないと、モデル設定が終了したことになる)

### モデル部分のスキプトの例

```
確認的因子分析モデル  x1, x2, x3, x4, x5 の背後にf1を仮定
                        x5, x6, x7, x8, x10 の背後にf2を仮定
                        f1 と f2 の間に相関を仮定
```

```
seq.1 <- specifyEquations()
x1 = b1*f1          # 測定方程式
x2 = b2*f1
x3 = b3*f1
x4 = b4*f1
x5 = b5*f1
x6 = b6*f2
x7 = b7*f2
x8 = b8*f2
x9 = b9*f2
x10 = b10*f2
V(x1)= ev1          # 内生変数の誤差分散
V(x2)= ev2
V(x3)= ev3
V(x4)= ev4
V(x5)= ev5
V(x6)= ev6
V(x7)= ev7
```

```

V(x8)= ev8
V(x9)= ev9
V(x10)= ev10
V(f1) = 1      # 外生変数の分散
V(f2) = 1
C(f1, f2) =cf12 # 外生変数の共分散

```

```

> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計分析力尺度データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 xt yt
1    1  3  2  1  2  2  4  4  3  4  1  2  2  4  2  2  1  4  3  3  3  4 29 26
2    2  3  3  4  3  2  2  2  2  4  1  2  2  4  2  2  1  1  1  3  1 26 19
3    3  4  3  3  4  1  3  4  2  5  1  4  3  4  4  1  3  4  2  5  1 30 31
4    4  5  5  5  3  3  4  4  2  4  4  5  4  5  2  2  3  4  2  3  3 39 33
5    5  3  3  4  2  2  3  3  3  4  1  3  4  4  3  2  4  3  4  5  1 28 33
6    6  2  1  4  1  1  3  3  2  5  1  3  2  4  2  2  5  4  3  5  2 23 32
統計 数学 批判的思考力 国語 自己効力感
1    51  48          28  72      61
2    74  53          26  66      53
3    48  60          35  71      48
4    67  68          27  67      48
5    55  49          30  66      49
6    74  63          36  83      37
>
> vn.itemx <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")
>
>
> # 記述統計量
> library(psych)
> describe(d1[,vn.itemx])
  vars  n mean  sd median trimmed  mad min max range  skew kurtosis  se
x1    1 365 3.99 0.92      4   4.07 1.48  1  5  4 -0.56  -0.38 0.05
x2    2 365 3.09 0.96      3   3.07 1.48  1  5  4  0.06  -0.37 0.05
x3    3 365 4.06 0.85      4   4.13 1.48  1  5  4 -0.60  -0.09 0.04
x4    4 365 3.00 1.07      3   3.01 1.48  1  5  4 -0.03  -0.60 0.06
x5    5 365 2.19 0.93      2   2.11 1.48  1  5  4  0.44  -0.47 0.05
x6    6 365 3.04 1.01      3   3.03 1.48  1  5  4  0.01  -0.52 0.05
x7    7 365 3.12 0.99      3   3.12 1.48  1  5  4 -0.12  -0.54 0.05
x8    8 365 2.15 0.89      2   2.09 1.48  1  5  4  0.39  -0.50 0.05
x9    9 365 3.91 0.90      4   3.98 1.48  1  5  4 -0.50  -0.34 0.05
x10   10 365 2.13 0.95      2   2.05 1.48  1  5  4  0.45  -0.57 0.05
>
>
> # 共分散行列・相関係数行列
> (cov.d1 <- cov(d1[,vn.itemx]))
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10
x1  0.8378443 0.4101084 0.3604095 0.4780220 0.4136234 0.2695243 0.2894325 0.2017914 0.2053214 0.2318531
x2  0.4101084 0.9143760 0.3869712 0.5054945 0.4661147 0.2713985 0.2554795 0.2798811 0.2362110 0.1855186
x3  0.3604095 0.3869712 0.7240403 0.4093407 0.3700963 0.2642330 0.2123363 0.1965227 0.2088364 0.1647674
x4  0.4780220 0.5054945 0.4093407 1.1538462 0.5000000 0.2609890 0.3049451 0.3021978 0.3104396 0.2527473
x5  0.4136234 0.4661147 0.3700963 0.5000000 0.8662953 0.2845552 0.2362336 0.2438281 0.1949496 0.2144438
x6  0.2695243 0.2713985 0.2642330 0.2609890 0.2845552 1.0289478 0.4408475 0.3929700 0.3932410 0.4458528
x7  0.2894325 0.2554795 0.2123363 0.3049451 0.2362336 0.4408475 0.9833358 0.3942872 0.3345326 0.3251242
x8  0.2017914 0.2798811 0.1965227 0.3021978 0.2438281 0.3929700 0.3942872 0.7986602 0.2769833 0.3455141
x9  0.2053214 0.2362110 0.2088364 0.3104396 0.1949496 0.3932410 0.3345326 0.2769833 0.8164685 0.2588138
x10 0.2318531 0.1855186 0.1647674 0.2527473 0.2144438 0.4458528 0.3251242 0.3455141 0.2588138 0.9002409
>
>
> (cor.d1 <- cor(d1[,vn.itemx]))
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10
x1  1.0000000 0.4685485 0.4627358 0.4861746 0.4855013 0.2902817 0.3188707 0.2466834 0.2482463 0.2669635
x2  0.4685485 1.0000000 0.4755924 0.4921303 0.5237179 0.2798004 0.2694279 0.3275142 0.2733808 0.2044776
x3  0.4627358 0.4755924 1.0000000 0.4478469 0.4673050 0.3061322 0.2516471 0.2584343 0.2716158 0.2040848
x4  0.4861746 0.4921303 0.4478469 1.0000000 0.5001071 0.2395254 0.2862838 0.3148012 0.3198405 0.2479894
x5  0.4855013 0.5237179 0.4673050 0.5001071 1.0000000 0.3013955 0.2559516 0.2931364 0.2318033 0.2428293
x6  0.2902817 0.2798004 0.3061322 0.2395254 0.3013955 1.0000000 0.4382690 0.4334927 0.4290347 0.4632500

```

```
x7 0.3188707 0.2694279 0.2516471 0.2862838 0.2559516 0.4382690 1.0000000 0.4449188 0.3733514 0.3455564
x8 0.2466834 0.3275142 0.2584343 0.3148012 0.2931364 0.4334927 0.4449188 1.0000000 0.3430069 0.4074791
x9 0.2482463 0.2733808 0.2716158 0.3198405 0.2318033 0.4290347 0.3733514 0.3430069 1.0000000 0.3018828
x10 0.2669635 0.2044776 0.2040848 0.2479894 0.2428293 0.4632500 0.3455564 0.4074791 0.3018828 1.0000000
>
>
```

```
> # 確認の因子分析
```

```
> library(sem)
```

```
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

```
>
```

```
> seq.1 <- specifyEquations()
```

```
1: x1 = b1*f1
2: x2 = b2*f1
3: x3 = b3*f1
4: x4 = b4*f1
5: x5 = b5*f1
6: x6 = b6*f2
7: x7 = b7*f2
8: x8 = b8*f2
9: x9 = b9*f2
10: x10 = b10*f2
11: V(x1) = ev1
12: V(x2) = ev2
13: V(x3) = ev3
14: V(x4) = ev4
15: V(x5) = ev5
16: V(x6) = ev6
17: V(x7) = ev7
18: V(x8) = ev8
19: V(x9) = ev9
20: V(x10) = ev10
21: V(f1) = 1
22: V(f2) = 1
23: C(f1, f2) = cf12
24:
```

```
Read 23 items
```

```
> sem.seq.1 <- sem(seq.1, S=cor.d1, N=nrow(d1))
```

```
> summary(sem.seq.1)
```

```
Model Chisquare = 33.88776 Df = 34 Pr(>Chisq) = 0.4731506
Goodness-of-fit index = 0.982311
Adjusted goodness-of-fit index = 0.9713855
RMSEA index = 0 90% CI: (NA, 0.03787012)
Bentler-Bonett NFI = 0.9692328
Tucker-Lewis NNFI = 1.000141
Bentler CFI = 1
Bentler RNI = 1.000106
Bollen IFI = 1.000105
SRMR = 0.02521577
AIC = 75.88776
AICc = 36.58164
BIC = -166.7088
CAIC = -200.7088
```

```
Normalized Residuals
```

```
Min. 1st Qu. Median Mean 3rd Qu. Max.
-1.126000 -0.301100 -0.000002 0.012060 0.269200 1.400000
```

```
R-square for Endogenous Variables
```

```
x1 x2 x3 x4 x5 x6 x7 x8 x9 x10
0.4690 0.5038 0.4368 0.4860 0.5123 0.4989 0.4060 0.4250 0.3283 0.3516
```

```
Parameter Estimates
```

```
Estimate Std Error z value Pr(>|z|)
b1 0.6848289 0.04997713 13.702846 9.762113e-43 x1 <--- f1
b2 0.7098060 0.04944059 14.356748 9.665789e-47 x2 <--- f1
b3 0.6609198 0.05048056 13.092560 3.631562e-39 x3 <--- f1
b4 0.6971099 0.04971444 14.022283 1.138914e-44 x4 <--- f1
```

	A	B	C	D	E	F	G	H	I	J	K
1	番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
2	1	3	2	1	2	2	4	4	3	4	4
3	2	3	3	4	3	2	2	2	2	4	1
4	3	4	3	3	4	1	3	4	2	5	1
5	4	5	5	5	3	3	4	4	2	4	4
6	5	3	3	4	2	2	3	3	3	4	1
7	6	2	1	4	1	1	3	3	2	5	1
8	7	5	3	5	5	4	5	5	3	5	3
9	8	4	3	4	3	2	3	2	2	4	1
10	9	4	2	4	2	2	2	2	2	4	2
11	10	4	5	5	4	4	3	3	3	4	3
12	11	4	2	4	4	2	4	4	2	4	3
13	12	5	3	4	5	2	3	3	2	3	2
14	13	3	3	5	3	3	2	3	2	3	2
15	14	4	4	5	3	3	3	4	2	3	1
16	15	4	2	4	3	1	4	2	1	3	3
17	16	4	3	4	1	1	2	1	1	3	1
18	17	3	2	4	3	2	2	2	2	3	2
19	18	3	3	5	2	2	3	2	1	4	1
20	19	3	3	5	2	2	4	4	2	4	1
21	20	3	2	5	1	2	3	3	1	4	2

```

b5    0.7157307 0.04931213 14.514292 9.836799e-48 x5 <--- f1
b6    0.7063319 0.05138262 13.746513 5.344417e-43 x6 <--- f2
b7    0.6371831 0.05261420 12.110479 9.294967e-34 x7 <--- f2
b8    0.6519190 0.05235028 12.453017 1.346562e-35 x8 <--- f2
b9    0.5729784 0.05374247 10.661557 1.539969e-26 x9 <--- f2
b10   0.5929581 0.05339759 11.104585 1.191696e-28 x10 <--- f2
ev1   0.5310095 0.04785074 11.097207 1.294250e-28 x1 <--> x1
ev2   0.4961756 0.04622789 10.733253 7.105019e-27 x2 <--> x2
ev3   0.5631851 0.04943157 11.393227 4.519360e-30 x3 <--> x3
ev4   0.5140380 0.04704727 10.925989 8.659161e-28 x4 <--> x4
ev5   0.4877297 0.04585069 10.637346 1.997373e-26 x5 <--> x5
ev6   0.5010953 0.05033465 9.955276 2.391579e-23 x6 <--> x6
ev7   0.5939976 0.05365121 11.071467 1.725533e-28 x7 <--> x7
ev8   0.5750016 0.05289307 10.871018 1.584232e-27 x8 <--> x8
ev9   0.6716955 0.05706458 11.770796 5.520148e-32 x9 <--> x9
ev10  0.6484007 0.05599640 11.579328 5.245554e-31 x10 <--> x10
cf12  0.6115912 0.04689208 13.042526 7.009518e-39 f2 <--> f1

```

Iterations = 13

```

> stdCoeef(sem, seq.1)
      Std. Estimate
1    b1      0.6848288 x1 <--- f1
2    b2      0.7098060 x2 <--- f1
3    b3      0.6609198 x3 <--- f1
4    b4      0.6971098 x4 <--- f1
5    b5      0.7157307 x5 <--- f1
6    b6      0.7063318 x6 <--- f2
7    b7      0.6371832 x7 <--- f2
8    b8      0.6519190 x8 <--- f2
9    b9      0.5729785 x9 <--- f2
10   b10     0.5929581 x10 <--- f2
11   ev1     0.5310095 x1 <--> x1
12   ev2     0.4961755 x2 <--> x2
13   ev3     0.5631851 x3 <--> x3
14   ev4     0.5140379 x4 <--> x4
15   ev5     0.4877296 x5 <--> x5
16   ev6     0.5010953 x6 <--> x6
17   ev7     0.5939976 x7 <--> x7
18   ev8     0.5750016 x8 <--> x8
19   ev9     0.6716956 x9 <--> x9
20   ev10    0.6484007 x10 <--> x10
21           1.0000000 f1 <--> f1
22           1.0000000 f2 <--> f2
23   cf12    0.6115912 f2 <--> f1
>
>
>
>

```

> # 測定モデルの1つのパス係数を1にするモデル

```
> seq.2 <- specifyEquations()
```

```

1: x1 = 1*f1
2: x2 = b21*f1
3: x3 = b31*f1
4: x4 = b41*f1
5: x5 = b51*f1
6: x6 = 1*f2
7: x7 = b72*f2
8: x8 = b82*f2
9: x9 = b92*f2
10: x10 = b102*f2
11: V(x1)= ev1
12: V(x2)= ev2
13: V(x3)= ev3
14: V(x4)= ev4
15: V(x5)= ev5
16: V(x6)= ev6
17: V(x7)= ev7
18: V(x8)= ev8
19: V(x9)= ev9
20: V(x10)= ev10

```

```

21: V(f1) = vf1
22: V(f2) = vf2
23: C(f1, f2) = cf12
24:
Read 23 items

>
> sem.seq.2 <- sem(seq.2, data=d1, formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10)

> summary(sem.seq.2)

Model Chisquare = 33.88776 Df = 34 Pr(>Chisq) = 0.4731506
Goodness-of-fit index = 0.982311
Adjusted goodness-of-fit index = 0.9713855
RMSEA index = 0 90% CI: (NA, 0.03787012)
Bentler-Bonett NFI = 0.9692328
Tucker-Lewis NNFI = 1.000141
Bentler CFI = 1
Bentler RNI = 1.000106
Bollen IFI = 1.000105
SRMR = 0.02521578
AIC = 75.88776
AICc = 36.58164
BIC = -166.7088
CAIC = -200.7088

Normalized Residuals
      Min.      1st Qu.      Median      Mean      3rd Qu.      Max.
-1.1260000 -0.3011000 -0.0000011  0.0120600  0.2692000  1.4000000

R-square for Endogenous Variables
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10
0.4690 0.5038 0.4368 0.4860 0.5123 0.4989 0.4060 0.4250 0.3283 0.3516

Parameter Estimates
      Estimate Std Error z value Pr(>|z|)
b21  1.0364720 0.09060888 11.438968 2.670440e-30 x2 <--- f1
b31  0.9650876 0.08947584 10.786013 4.008021e-27 x3 <--- f1
b41  1.0179327 0.09028794 11.274293 1.757850e-29 x4 <--- f1
b51  1.0451237 0.09076681 11.514381 1.116579e-30 x5 <--- f1
b72  0.9021016 0.08930775 10.101045 5.465677e-24 x7 <--- f2
b82  0.9229635 0.08972783 10.286257 8.127490e-25 x8 <--- f2
b92  0.8112034 0.08782609 9.236473 2.547551e-20 x9 <--- f2
b102 0.8394897 0.08823871 9.513848 1.837376e-21 x10 <--- f2
ev1  0.5310093 0.04785071 11.097207 1.294251e-28 x1 <--> x1
ev2  0.4961754 0.04622787 10.733253 7.104984e-27 x2 <--> x2
ev3  0.5631850 0.04943156 11.393227 4.519349e-30 x3 <--> x3
ev4  0.5140379 0.04704726 10.925991 8.658991e-28 x4 <--> x4
ev5  0.4877294 0.04585067 10.637344 1.997424e-26 x5 <--> x5
ev6  0.5010953 0.05033465 9.955275 2.391597e-23 x6 <--> x6
ev7  0.5939979 0.05365123 11.071468 1.725514e-28 x7 <--> x7
ev8  0.5750022 0.05289310 10.871024 1.584124e-27 x8 <--> x8
ev9  0.6716954 0.05706459 11.770793 5.520301e-32 x9 <--> x9
ev10 0.6484003 0.05599638 11.579325 5.245723e-31 x10 <--> x10
vf1  0.4689906 0.06845154 6.851424 7.311836e-12 f1 <--> f1
vf2  0.4989046 0.07258636 6.873256 6.275275e-12 f2 <--> f2
cf12 0.2958367 0.04229842 6.994038 2.670857e-12 f2 <--> f1

Iterations = 26

> stdCoef(sem.seq.2)
      Std. Estimate
1      0.6848289 x1 <--- f1
2 b21  0.7098060 x2 <--- f1
3 b31  0.6609198 x3 <--- f1
4 b41  0.6971098 x4 <--- f1
5 b51  0.7157308 x5 <--- f1
6      0.7063318 x6 <--- f2
7 b72  0.6371830 x7 <--- f2
8 b82  0.6519185 x8 <--- f2
9 b92  0.5729787 x9 <--- f2
10 b102 0.5929583 x10 <--- f2

```

```
11 ev1      0.5310094    x1 <--> x1
12 ev2      0.4961755    x2 <--> x2
13 ev3      0.5631850    x3 <--> x3
14 ev4      0.5140380    x4 <--> x4
15 ev5      0.4877294    x5 <--> x5
16 ev6      0.5010953    x6 <--> x6
17 ev7      0.5939978    x7 <--> x7
18 ev8      0.5750022    x8 <--> x8
19 ev9      0.6716954    x9 <--> x9
20 ev10     0.6484004    x10 <--> x10
21 vf1      1.0000000    f1 <--> f1
22 vf2      1.0000000    f2 <--> f2
23 cf12     0.6115913    f2 <--> f1
>
>
>
```

## 2 次因子分析

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

```
モデル名 <- specifyEquations()
構造方程式
```

## 非標準化解の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
```

もしくは

```
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+…)
```

【重要】 共分散行列は、分析に用いる観測変数だけのものにすること

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

空白行があるところで、モデルの設定が終了したとみなされる。

(逆に、行をあげないと、モデル設定が終了したことにならない)

## 結果の出力

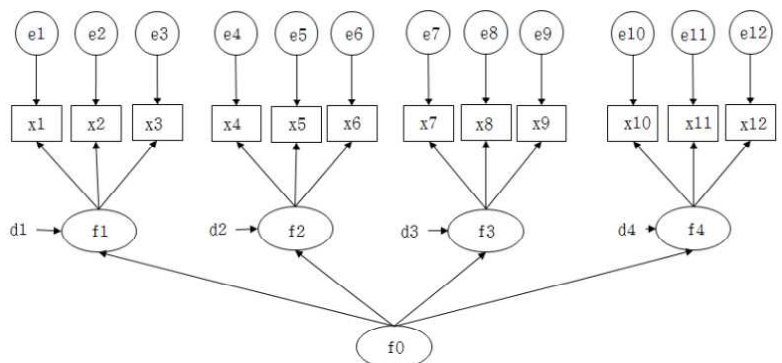
```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

## モデル部分のスキプトの例

```
seq.1 <- specifyEquations()
x1 = 1*f1          # 測定方程式
x2 = b2*f1          # f1の分散を1としても、f1の1つの測定方程式のパス係数を1としても同等の制約
x3 = b3*f1
x4 = 1*f2
x5 = b5*f2
x6 = b6*f2
x7 = 1*f3
x8 = b8*f3
x9 = b9*f3
x10 = 1*f4
x11 = b11*f4
x12 = b12*f4
f1 = a1*f0          # 潜在変数間の構造方程式
f2 = a2*f0
f3 = a3*f0
f4 = a4*f0
V(x1) = ev1         # 内生変数の誤差分散
V(x2) = ev2
V(x3) = ev3
V(x4) = ev4
V(x5) = ev5
V(x6) = ev6
V(x7) = ev7
V(x8) = ev8
V(x9) = ev9
V(x10) = ev10
V(x11) = ev11
V(x12) = ev12
V(f1) = evf1
V(f2) = evf2
V(f3) = evf3
V(f4) = evf4
V(f0) = 1           # 外生変数の分散
```



```
> setwd("d:¥¥")
> d1 <- read.table("拡張SEMデータ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12
1  4  3  2  3  4  4  2  4  2  4  4  3
2  2  3  2  5  3  4  5  3  3  4  4  4
3  1  4  2  2  3  3  4  3  2  3  4  4
4  1  5  2  2  3  2  3  2  2  5  4  3
5  2  3  1  4  2  1  3  3  2  1  1  1
6  2  4  3  3  5  4  4  3  4  2  4  1
>
>
```

```
> # 記述統計量
> library(psych)
> describe(d1)
  vars      n mean    sd median trimmed  mad min max range  skew kurtosis   se
x1      1 276 3.01 0.98      3    3.01 1.48    1  5     4  0.02   -0.36 0.06
x2      2 276 3.01 1.03      3    3.05 1.48    1  5     4 -0.17   -0.58 0.06
x3      3 276 3.00 0.99      3    3.01 1.48    1  5     4 -0.07   -0.44 0.06
x4      4 276 2.99 1.02      3    2.97 1.48    1  5     4  0.09   -0.50 0.06
x5      5 276 3.00 1.00      3    3.05 1.48    1  5     4 -0.23   -0.51 0.06
x6      6 276 3.03 1.04      3    3.02 1.48    1  5     4  0.01   -0.43 0.06
x7      7 276 3.03 0.99      3    3.03 1.48    1  5     4 -0.01   -0.45 0.06
x8      8 276 3.01 1.01      3    3.05 1.48    1  5     4 -0.18   -0.49 0.06
x9      9 276 3.00 1.01      3    3.01 1.48    1  5     4 -0.06   -0.66 0.06
x10    10 276 2.99 1.03      3    3.00 1.48    1  5     4 -0.03   -0.49 0.06
x11    11 276 2.99 1.00      3    2.99 1.48    1  5     4  0.00   -0.36 0.06
x12    12 276 2.99 1.01      3    3.00 1.48    1  5     4 -0.06   -0.42 0.06>
```

```
> # 共分散行列・相関係数行列
> cov.d1 <- cov(d1)
> round(cov.d1, 2)
  x1      x2      x3      x4      x5      x6      x7      x8      x9      x10     x11     x12
x1  0.96  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.07  0.22 -0.01  0.03  0.04  0.28  0.06  0.09  0.08  0.09  0.11
x3  0.19  0.22  0.99 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.05  0.20  0.24  0.03  0.04  0.02  0.09  0.08  0.09
x5 -0.01  0.03  0.03  0.20  1.01  0.22  0.05  0.23  0.02  0.08  0.08  0.11
x6  0.03  0.04  0.01  0.24  0.22  1.09  0.09  0.05  0.07  0.09  0.30  0.12
x7  0.11  0.28  0.12  0.03  0.05  0.09  0.98  0.21  0.23  0.07  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.05  0.21  1.03  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.03  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.07  0.09  0.06  1.05  0.17  0.23
x11 0.07  0.09  0.01  0.08  0.08  0.30  0.05  0.03  0.04  0.17  0.99  0.24
x12 0.07  0.11  0.07  0.09  0.11  0.12  0.07  0.11  0.22  0.23  0.24  1.01
```

```
> cor.d1 <- cor(d1)
> round(cor.d1, 2)
  x1      x2      x3      x4      x5      x6      x7      x8      x9      x10     x11     x12
x1  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09  0.08  0.09  0.10
x3  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02  0.09  0.08  0.08
x5 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02  0.08  0.08  0.10
x6  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07  0.09  0.29  0.11
x7  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23  0.06  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.04  0.21  1.00  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.00  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.06  0.09  0.06  1.00  0.17  0.22
x11 0.07  0.09  0.01  0.08  0.08  0.29  0.05  0.03  0.04  0.17  1.00  0.24
x12 0.07  0.10  0.07  0.08  0.10  0.11  0.07  0.11  0.22  0.22  0.24  1.00
```



```

>
> # 適合度指標の設定
> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR",
R", "AIC", "AICc", "BIC", "CAIC"))
>

```

```

> # 2次因子分析モデル
> # specifyEquations を使う方法

```

```

> seq.1 <- specifyEquations()

```

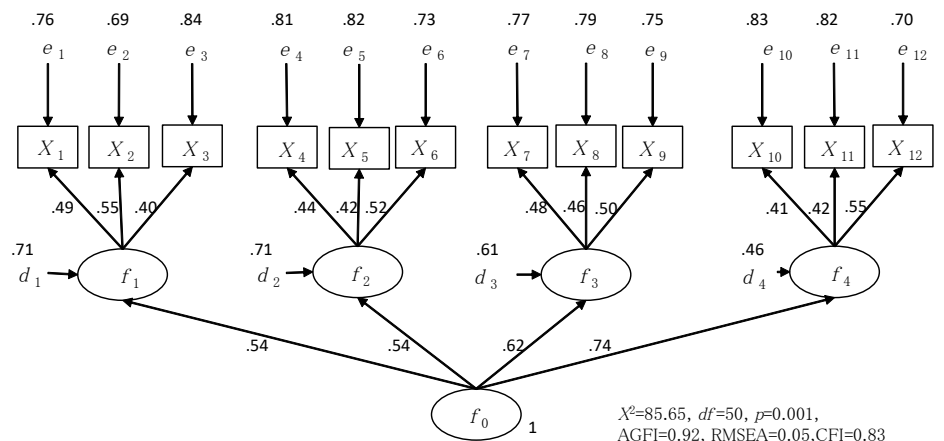
```

1: x1 = 1*f1
2: x2 = b2*f1
3: x3 = b3*f1
4: x4 = 1*f2
5: x5 = b5*f2
6: x6 = b6*f2
7: x7 = 1*f3
8: x8 = b8*f3
9: x9 = b9*f3
10: x10 = 1*f4
11: x11 = b11*f4
12: x12 = b12*f4
13: f1 = a1*f0
14: f2 = a2*f0
15: f3 = a3*f0
16: f4 = a4*f0
17: V(x1) = ev1
18: V(x2) = ev2
19: V(x3) = ev3
20: V(x4) = ev4
21: V(x5) = ev5
22: V(x6) = ev6
23: V(x7) = ev7
24: V(x8) = ev8
25: V(x9) = ev9
26: V(x10) = ev10
27: V(x11) = ev11
28: V(x12) = ev12
29: V(f1) = evf1
30: V(f2) = evf2
31: V(f3) = evf3
32: V(f4) = evf4
33: V(f0) = 1
34:

```

Read 33 items

1	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
2	4	3	2	3	4	4	2	4	2	4	4	3
3	2	3	2	5	3	4	5	3	3	4	4	4
4	1	4	2	2	3	3	4	3	2	3	4	4
5	1	5	2	2	3	2	3	2	2	5	4	3
6	2	3	1	4	2	1	3	3	2	1	1	1
7	2	4	3	3	5	4	4	3	4	2	4	1
8	4	4	4	4	4	3	2	1	2	2	2	1
9	3	4	3	3	3	4	3	4	3	4	4	3
10	3	4	3	2	3	5	4	4	3	2	5	4
11	3	5	2	2	4	4	4	4	5	4	1	3
12	4	2	3	4	4	5	3	4	2	5	3	2
13	4	4	2	3	4	2	2	4	4	3	3	4
14	3	4	3	2	3	3	4	4	4	4	2	5
15	4	3	5	4	2	3	3	2	2	4	3	3
16	1	4	3	2	3	2	4	1	3	2	3	2
17	4	4	3	2	3	4	4	4	3	3	3	3
18	3	2	2	4	1	4	4	4	4	3	2	2
19	4	4	4	4	4	3	3	4	2	2	1	3
20	3	3	3	3	3	4	2	3	4	4	3	3
21	4	2	5	2	2	2	4	2	4	4	3	5



```

> # 非標準化解の推定

```

```

> sem.seq.1 <- sem(seq.1, S=cor.d1, N=nrow(d1))
> summary(sem.seq.1)
> # もしくは
> sem.seq.2 <- sem(seq.1, data=d1, formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)
> summary(sem.seq.2)

```

```

Model Chisquare = 85.65067   Df = 50   Pr(>Chisq) = 0.001266141
Goodness-of-fit index = 0.9517556
Adjusted goodness-of-fit index = 0.9247387
RMSEA index = 0.05091931   90% CI: (0.03177626, 0.06890628)
Bentler-Bonett NFI = 0.6892108
Tucker-Lewis NNFI = 0.7754727
Bentler CFI = 0.8299035
Bentler RNI = 0.8299035
Bollen IFI = 0.8419676
SRMR = 0.05468587
AIC = 141.6507
AICc = 92.22557
BIC = -195.3694
CAIC = -245.3694

```

Normalized Residuals  
 Min. 1st Qu. Median Mean 3rd Qu. Max.  
 -1.323656 -0.502375 -0.070276 0.004355 0.079176 3.355377

R-square for Endogenous Variables  
 f1 x1 x2 x3 f2 x4 x5 x6 f3 x7 x8 x9 f4  
 0.2865 0.2423 0.3053 0.1560 0.2916 0.1916 0.1801 0.2728 0.3863 0.2287 0.2089 0.2466 0.5443  
 x10 x11 x12  
 0.1668 0.1788 0.3019

Parameter Estimates  
 Estimate Std Error z value Pr(>|z|)  
 b2 1.18444217 0.33646174 3.520288 4.310787e-04 x2 <--- f1  
 b3 0.81460137 0.23612783 3.449832 5.609354e-04 x3 <--- f1  
 b5 0.95102332 0.29149725 3.262546 1.104161e-03 x5 <--- f2  
 b6 1.21561522 0.37237855 3.264461 1.096726e-03 x6 <--- f2  
 b8 0.97710540 0.27181487 3.594746 3.247083e-04 x8 <--- f3  
 b9 1.06171791 0.29283572 3.625643 2.882428e-04 x9 <--- f3  
 b11 1.00440033 0.29641347 3.388511 7.027319e-04 x11 <--- f4  
 b12 1.31929411 0.37917611 3.479370 5.025935e-04 x12 <--- f4  
 a1 0.25814089 0.07747188 3.332059 8.620597e-04 f1 <--- f0  
 a2 0.24186184 0.07706108 3.138573 1.697726e-03 f2 <--- f0  
 a3 0.29438284 0.08161202 3.607101 3.096369e-04 f3 <--- f0  
 a4 0.30945693 0.08830870 3.504263 4.578726e-04 f4 <--- f0  
 ev1 0.72719838 0.09224684 7.883179 3.191558e-15 x1 <--> x1  
 ev2 0.74257763 0.11284740 6.580370 4.692786e-11 x2 <--> x2  
 ev3 0.83474938 0.08690155 9.605690 7.564965e-22 x3 <--> x3  
 ev4 0.84646398 0.09777197 8.657532 4.820883e-18 x4 <--> x4  
 ev5 0.82582943 0.09282243 8.896874 5.744182e-19 x5 <--> x5  
 ev6 0.79019912 0.11430942 6.912808 4.751515e-12 x6 <--> x6  
 ev7 0.75662545 0.09133527 8.284044 1.190611e-16 x7 <--> x7  
 ev8 0.81104951 0.09352396 8.672104 4.242064e-18 x8 <--> x8  
 ev9 0.77255764 0.09748155 7.925168 2.278378e-15 x9 <--> x9  
 ev10 0.87855180 0.09303965 9.442767 3.630620e-21 x10 <--> x10  
 ev11 0.81511595 0.08831059 9.230104 2.703688e-20 x11 <--> x11  
 ev12 0.70819501 0.10436513 6.785743 1.154903e-11 x12 <--> x12  
 evf1 0.16595404 0.07028706 2.361090 1.822133e-02 f1 <--> f1  
 evf2 0.14210093 0.06703471 2.119811 3.402197e-02 f2 <--> f2  
 evf3 0.13768833 0.06529029 2.108864 3.495635e-02 f3 <--> f3  
 evf4 0.08017732 0.05176428 1.548893 1.214075e-01 f4 <--> f4

Iterations = 51

```
> # 標準化解の推定
> stdCoef(sem, seq. 1)
> # もしくは
> stdCoef(sem, seq. 2)
```

	Std. Estimate	
1	0.4922756	x1 <--- f1
2	b2 0.5525170	x2 <--- f1
3	b3 0.3950238	x3 <--- f1
4	0.4377006	x4 <--- f2
5	b5 0.4244083	x5 <--- f2
6	b6 0.5222991	x6 <--- f2
7	0.4782265	x7 <--- f3
8	b8 0.4570781	x8 <--- f3
9	b9 0.4966078	x9 <--- f3
10	0.4084714	x10 <--- f4
11	b11 0.4228645	x11 <--- f4
12	b12 0.5494330	x12 <--- f4
13	a1 0.5352548	f1 <--- f0
14	a2 0.5400127	f2 <--- f0
15	a3 0.6215125	f3 <--- f0
16	a4 0.7377629	f4 <--- f0
17	ev1 0.7576647	x1 <--> x1
18	ev2 0.6947250	x2 <--> x2
19	ev3 0.8439562	x3 <--> x3
20	ev4 0.8084182	x4 <--> x4
21	ev5 0.8198776	x5 <--> x5
22	ev6 0.7272037	x6 <--> x6

```
23 ev7      0.7712994    x7 <--> x7
24 ev8      0.7910796    x8 <--> x8
25 ev9      0.7533807    x9 <--> x9
26 ev10     0.8331511    x10 <--> x10
27 ev11     0.8211856    x11 <--> x11
28 ev12     0.6981234    x12 <--> x12
29 evf1     0.7135023     f1 <--> f1
30 evf2     0.7083863     f2 <--> f2
31 evf3     0.6137222     f3 <--> f3
32 evf4     0.4557060     f4 <--> f4
33          1.0000000     f0 <--> f0
>
>
```

## 階層因子分析

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

```
モデル名 <- specifyEquations()
構造方程式
```

## 非標準化解の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
```

もしくは

```
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+…)
```

【重要】 共分散行列や相関係数行列は、分析に用いる観測変数だけのものにする  
余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

空白行があるところで、モデルの設定が終了したとみなされる。

(逆に、行をあげないと、モデル設定が終了したことにならない)

## 結果の出力

```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

## モデル部分のスキプトの例

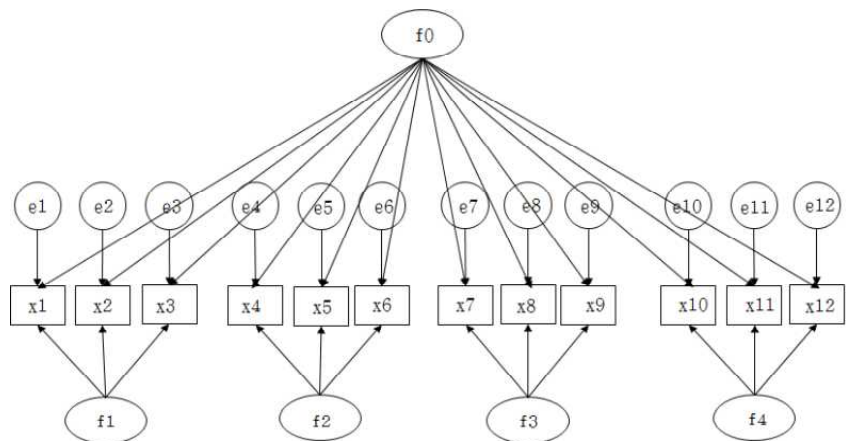
```
seq.1 <- specifyEquations()
x1 = 1*f0 # 測定方程式
```

```
x2 = g2*f0
x3 = g3*f0
x4 = g4*f0
x5 = g5*f0
x6 = g6*f0
x7 = g7*f0
x8 = g8*f0
x9 = g9*f0
x10 = g10*f0
x11 = g11*f0
x12 = g12*f0
```

```
x1 = 1*f1
x2 = b2*f1
x3 = b3*f1
x4 = 1*f2
x5 = b5*f2
x6 = b6*f2
x7 = 1*f3
x8 = b8*f3
x9 = b9*f3
x10 = 1*f4
x11 = b11*f4
x12 = b12*f4
```

```
# 内生変数の誤差分散
```

```
V(x1) = ev1
V(x2) = ev2
V(x3) = ev3
V(x4) = ev4
V(x5) = ev5
V(x6) = ev6
V(x7) = ev7
V(x8) = ev8
```



```

V(x9) = ev9
V(x10) = ev10
V(x11) = ev11
V(x12) = ev12
V(f0) = vf0          # 外生変数の分散
V(f1) = vf1
V(f2) = vf2
V(f3) = vf3
V(f4) = vf4

> setwd("d:¥¥")
> d1 <- read.table("拡張SEMデータ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12
1  4  3  2  3  4  4  2  4  2  4  4  3
2  2  3  2  5  3  4  5  3  3  4  4  4
3  1  4  2  2  3  3  4  3  2  3  4  4
4  1  5  2  2  3  2  3  2  2  5  4  3
5  2  3  1  4  2  1  3  3  2  1  1  1
6  2  4  3  3  5  4  4  3  4  2  4  1
>
>
> # 記述統計量
> library(psych)
> describe(d1)
  vars    n mean   sd median trimmed  mad min max range  skew kurtosis   se
x1     1 276 3.01 0.98      3    3.01 1.48    1  5    4  0.02   -0.36 0.06
x2     2 276 3.01 1.03      3    3.05 1.48    1  5    4 -0.17   -0.58 0.06
x3     3 276 3.00 0.99      3    3.01 1.48    1  5    4 -0.07   -0.44 0.06
x4     4 276 2.99 1.02      3    2.97 1.48    1  5    4  0.09   -0.50 0.06
x5     5 276 3.00 1.00      3    3.05 1.48    1  5    4 -0.23   -0.51 0.06
x6     6 276 3.03 1.04      3    3.02 1.48    1  5    4  0.01   -0.43 0.06
x7     7 276 3.03 0.99      3    3.03 1.48    1  5    4 -0.01   -0.45 0.06
x8     8 276 3.01 1.01      3    3.05 1.48    1  5    4 -0.18   -0.49 0.06
x9     9 276 3.00 1.01      3    3.01 1.48    1  5    4 -0.06   -0.66 0.06
x10    10 276 2.99 1.03      3    3.00 1.48    1  5    4 -0.03   -0.49 0.06
x11    11 276 2.99 1.00      3    2.99 1.48    1  5    4  0.00   -0.36 0.06
x12    12 276 2.99 1.01      3    3.00 1.48    1  5    4 -0.06   -0.42 0.06>
>

> # 共分散行列・相関係数行列
> cov.d1 <- cov(d1)
> round(cov.d1, 2)
  x1    x2    x3    x4    x5    x6    x7    x8    x9    x10   x11   x12
x1  0.96  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.07  0.22 -0.01  0.03  0.04  0.28  0.06  0.09  0.08  0.09  0.11
x3  0.19  0.22  0.99 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.05  0.20  0.24  0.03  0.04  0.02  0.09  0.08  0.09
x5 -0.01  0.03  0.03  0.20  1.01  0.22  0.05  0.23  0.02  0.08  0.08  0.11
x6  0.03  0.04  0.01  0.24  0.22  1.09  0.09  0.05  0.07  0.09  0.30  0.12
x7  0.11  0.28  0.12  0.03  0.05  0.09  0.98  0.21  0.23  0.07  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.05  0.21  1.03  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.03  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.07  0.09  0.06  1.05  0.17  0.23
x11 0.07  0.09  0.01  0.08  0.08  0.30  0.05  0.03  0.04  0.17  0.99  0.24
x12 0.07  0.11  0.07  0.09  0.11  0.12  0.07  0.11  0.22  0.23  0.24  1.01

> cor.d1 <- cor(d1)
> round(cor.d1, 2)
  x1    x2    x3    x4    x5    x6    x7    x8    x9    x10   x11   x12
x1  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09  0.08  0.09  0.10
x3  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02  0.09  0.08  0.08
x5 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02  0.08  0.08  0.10
x6  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07  0.09  0.29  0.11
x7  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23  0.06  0.05  0.07

```

```

x8  0.07 0.06 0.08 0.04 0.23 0.04 0.21 1.00 0.24 0.09 0.03 0.11
x9  0.08 0.09 0.09 0.02 0.02 0.07 0.23 0.24 1.00 0.06 0.04 0.22
x10 0.06 0.08 0.21 0.09 0.08 0.09 0.06 0.09 0.06 1.00 0.17 0.22
x11 0.07 0.09 0.01 0.08 0.08 0.29 0.05 0.03 0.04 0.17 1.00 0.24
x12 0.07 0.10 0.07 0.08 0.10 0.11 0.07 0.11 0.22 0.22 0.24 1.00

```

```

>
> # 適合度指標の設定
> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRM
R", "AIC", "AICc", "BIC", "CAIC"))

```

```

> # 階層因子分析
> # specifyEquations を使う方法

```

```

> seq.1 <- specifyEquations()

```

```

1: x1 = 1*f0
2: x2 = g2*f0
3: x3 = g3*f0
4: x4 = g4*f0
5: x5 = g5*f0
6: x6 = g6*f0
7: x7 = g7*f0
8: x8 = g8*f0
9: x9 = g9*f0
10: x10 = g10*f0
11: x11 = g11*f0
12: x12 = g12*f0
13: x1 = 1*f1
14: x2 = b2*f1
15: x3 = b3*f1
16: x4 = 1*f2
17: x5 = b5*f2
18: x6 = b6*f2
19: x7 = 1*f3
20: x8 = b8*f3
21: x9 = b9*f3
22: x10 = 1*f4
23: x11 = b11*f4
24: x12 = b12*f4
25: V(x1) = ev1
26: V(x2) = ev2
27: V(x3) = ev3
28: V(x4) = ev4
29: V(x5) = ev5
30: V(x6) = ev6
31: V(x7) = ev7
32: V(x8) = ev8
33: V(x9) = ev9
34: V(x10) = ev10
35: V(x11) = ev11
36: V(x12) = ev12
37: V(f0) = vf0
38: V(f1) = vf1
39: V(f2) = vf2
40: V(f3) = vf3
41: V(f4) = vf4
42:

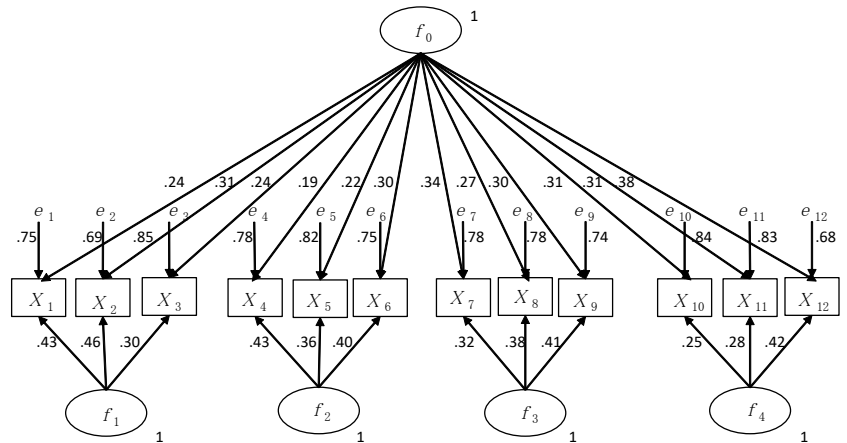
```

```

Read 41 items

```

1	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
2	4	3	2	3	4	4	2	4	2	4	4	3
3	2	3	2	5	3	4	5	3	3	4	4	4
4	1	4	2	2	3	3	4	3	2	3	4	4
5	1	5	2	2	3	2	3	2	2	5	4	3
6	2	3	1	4	2	1	3	3	2	1	1	1
7	2	4	3	3	5	4	4	3	4	2	4	1
8	4	4	4	4	4	3	2	1	2	2	2	1
9	3	4	3	3	3	4	3	4	3	4	4	3
10	3	4	3	2	3	5	4	4	3	2	5	4
11	3	5	2	2	4	4	4	4	5	4	1	3
12	4	2	3	4	4	5	3	4	2	5	3	2
13	4	4	2	3	4	2	2	4	4	3	3	4
14	3	4	3	2	3	3	4	4	4	4	2	5
15	4	3	5	4	2	3	3	4	2	2	4	3
16	1	4	3	2	3	2	4	1	3	2	3	2
17	4	4	3	2	3	4	4	4	3	3	3	3
18	3	2	2	4	1	4	4	4	4	3	2	2
19	4	4	4	4	4	3	3	4	2	2	1	3
20	3	3	3	3	3	4	2	3	4	4	3	3
21	4	2	5	2	2	2	4	2	4	4	3	5



$\chi^2=84.88$ ,  $df=42$ ,  $p=0.000$ ,  
 AGFI=0.91, RMSEA=0.06, CFI=0.80

```

> # 非標準化解の推定
> sem.seq.1 <- sem(seq.1, S=cov.d1, N=nrow(d1))
> summary(sem.seq.1)

```

```

Model Chisquare = 84.87574 Df = 42 Pr(>Chisq) = 0.0001000983
Goodness-of-fit index = 0.9520337
Adjusted goodness-of-fit index = 0.9109198
RMSEA index = 0.06092771 90% CI: (0.04201327, 0.07957644)
Bentler-Bonett NFI = 0.6920227
Tucker-Lewis NNFI = 0.6785349

```

Bentler CFI = 0.7954313  
 Bentler RNI = 0.7954313  
 Bollen IFI = 0.8164494  
 SRMR = 0.05440172  
 AIC = 156.8757  
 AICc = 96.02218  
 BIC = -151.1811  
 CAIC = -193.1811

## Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.162275	-0.514986	-0.000003	0.007099	0.114000	3.618964

## R-square for Endogenous Variables

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
0.2472	0.3110	0.1505	0.2199	0.1816	0.2499	0.2161	0.2190	0.2578	0.1605	0.1740	0.3222

## Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
g2	1.36278004	0.57087776	2.3871661	1.697882e-02	x2 <--- f0
g3	1.01201461	0.49912540	2.0275759	4.260356e-02	x3 <--- f0
g4	0.82244283	0.51096763	1.6095791	1.074898e-01	x4 <--- f0
g5	0.93502337	0.53372972	1.7518668	7.979672e-02	x5 <--- f0
g6	1.30716995	0.65017132	2.0105008	4.437821e-02	x6 <--- f0
g7	1.39988161	0.68252058	2.0510467	4.026240e-02	x7 <--- f0
g8	1.15031151	0.61174331	1.8803827	6.005594e-02	x8 <--- f0
g9	1.26438178	0.64580237	1.9578463	5.024805e-02	x9 <--- f0
g10	1.34069676	0.69421697	1.9312359	5.345388e-02	x10 <--- f0
g11	1.30864192	0.67601961	1.9358047	5.289163e-02	x11 <--- f0
g12	1.62137801	0.78110773	2.0757419	3.791783e-02	x12 <--- f0
b2	1.12133697	0.60936601	1.8401698	6.574331e-02	x2 <--- f1
b3	0.70910977	0.34633489	2.0474685	4.061211e-02	x3 <--- f1
b5	0.83365724	0.41277455	2.0196430	4.342043e-02	x5 <--- f2
b6	0.95383166	0.48029937	1.9859107	4.704324e-02	x6 <--- f2
b8	1.21773287	0.66497582	1.8312438	6.706416e-02	x8 <--- f3
b9	1.31240285	0.73944659	1.7748447	7.592353e-02	x9 <--- f3
b11	1.05986403	0.79144417	1.3391520	1.805212e-01	x11 <--- f4
b12	1.62522264	1.53274683	1.0603334	2.889930e-01	x12 <--- f4
ev1	0.72248994	0.11587596	6.2350286	4.516956e-10	x1 <--> x1
ev2	0.73647671	0.13782800	5.3434477	9.119516e-08	x2 <--> x2
ev3	0.84019349	0.08714634	9.6411797	5.356875e-22	x3 <--> x3
ev4	0.81681372	0.12325444	6.6270529	3.424550e-11	x4 <--> x4
ev5	0.82431208	0.09906363	8.3210368	8.719719e-17	x5 <--> x5
ev6	0.81512646	0.11566749	7.0471526	1.826160e-12	x6 <--> x6
ev7	0.76898385	0.09043266	8.5033866	1.841384e-17	x7 <--> x7
ev8	0.80067396	0.11295498	7.0884342	1.356378e-12	x8 <--> x8
ev9	0.76110989	0.12290135	6.1928522	5.908514e-10	x9 <--> x9
ev10	0.88525191	0.09837348	8.9988875	2.280165e-19	x10 <--> x10
ev11	0.81989028	0.09906338	8.2764215	1.269321e-16	x11 <--> x11
ev12	0.68760493	0.17629996	3.9001990	9.611367e-05	x12 <--> x12
vf0	0.05672909	0.04378159	1.2957293	1.950688e-01	f0 <--> f0
vf1	0.18056972	0.11596787	1.5570669	1.194546e-01	f1 <--> f1
vf2	0.19187615	0.11966139	1.6034926	1.088259e-01	f2 <--> f2
vf3	0.10082108	0.07966492	1.2655642	2.056691e-01	f3 <--> f3
vf4	0.06727200	0.08363181	0.8043829	4.211759e-01	f4 <--> f4

Iterations = 152

```
> sem.seq.2 <- sem(seq.1, data=d1, formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)
> summary(sem.seq.2)
```

Model Chisquare = 84.87574 Df = 42 Pr(>Chisq) = 0.0001000983  
 Goodness-of-fit index = 0.9520337  
 Adjusted goodness-of-fit index = 0.9109198  
 RMSEA index = 0.06092771 90% CI: (0.04201327, 0.07957644)  
 Bentler-Bonett NFI = 0.6920227  
 Tucker-Lewis NNFI = 0.6785349  
 Bentler CFI = 0.7954313  
 Bentler RNI = 0.7954313  
 Bollen IFI = 0.8164494  
 SRMR = 0.05440172

AIC = 156.8757  
 AICc = 96.02218  
 BIC = -151.1811  
 CAIC = -193.1811

## Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.162275	-0.514986	-0.000003	0.007099	0.114000	3.618964

## R-square for Endogenous Variables

x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
0.2472	0.3110	0.1505	0.2199	0.1816	0.2499	0.2161	0.2190	0.2578	0.1605	0.1740	0.3222

## Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
g2	1.36278004	0.57087776	2.3871661	1.697882e-02	x2 <--- f0
g3	1.01201461	0.49912540	2.0275759	4.260356e-02	x3 <--- f0
g4	0.82244283	0.51096763	1.6095791	1.074898e-01	x4 <--- f0
g5	0.93502337	0.53372972	1.7518668	7.979672e-02	x5 <--- f0
g6	1.30716995	0.65017132	2.0105008	4.437821e-02	x6 <--- f0
g7	1.39988161	0.68252058	2.0510467	4.026240e-02	x7 <--- f0
g8	1.15031151	0.61174331	1.8803827	6.005594e-02	x8 <--- f0
g9	1.26438178	0.64580237	1.9578463	5.024805e-02	x9 <--- f0
g10	1.34069676	0.69421697	1.9312359	5.345388e-02	x10 <--- f0
g11	1.30864192	0.67601961	1.9358047	5.289163e-02	x11 <--- f0
g12	1.62137801	0.78110773	2.0757419	3.791783e-02	x12 <--- f0
b2	1.12133697	0.60936601	1.8401698	6.574331e-02	x2 <--- f1
b3	0.70910977	0.34633489	2.0474685	4.061211e-02	x3 <--- f1
b5	0.83365724	0.41277455	2.0196430	4.342043e-02	x5 <--- f2
b6	0.95383166	0.48029937	1.9859107	4.704324e-02	x6 <--- f2
b8	1.21773287	0.66497582	1.8312438	6.706416e-02	x8 <--- f3
b9	1.31240285	0.73944659	1.7748447	7.592353e-02	x9 <--- f3
b11	1.05986403	0.79144417	1.3391520	1.805212e-01	x11 <--- f4
b12	1.62522264	1.53274683	1.0603334	2.889930e-01	x12 <--- f4
ev1	0.72248994	0.11587596	6.2350286	4.516956e-10	x1 <--> x1
ev2	0.73647671	0.13782800	5.3434477	9.119516e-08	x2 <--> x2
ev3	0.84019349	0.08714634	9.6411797	5.356875e-22	x3 <--> x3
ev4	0.81681372	0.12325444	6.6270529	3.424550e-11	x4 <--> x4
ev5	0.82431208	0.09906363	8.3210368	8.719719e-17	x5 <--> x5
ev6	0.81512646	0.11566749	7.0471526	1.826160e-12	x6 <--> x6
ev7	0.76898385	0.09043266	8.5033866	1.841384e-17	x7 <--> x7
ev8	0.80067396	0.11295498	7.0884342	1.356378e-12	x8 <--> x8
ev9	0.76110989	0.12290135	6.1928522	5.908514e-10	x9 <--> x9
ev10	0.88525191	0.09837348	8.9988875	2.280165e-19	x10 <--> x10
ev11	0.81989028	0.09906338	8.2764215	1.269321e-16	x11 <--> x11
ev12	0.68760493	0.17629996	3.9001990	9.611367e-05	x12 <--> x12
vf0	0.05672909	0.04378159	1.2957293	1.950688e-01	f0 <--> f0
vf1	0.18056972	0.11596787	1.5570669	1.194546e-01	f1 <--> f1
vf2	0.19187615	0.11966139	1.6034926	1.088259e-01	f2 <--> f2
vf3	0.10082108	0.07966492	1.2655642	2.056691e-01	f3 <--> f3
vf4	0.06727200	0.08363181	0.8043829	4.211759e-01	f4 <--> f4

Iterations = 152

```
>
>
> # 標準化解の推定
> stdCoef(sem, seq.1)
```

	Std. Estimate	
1	0.2431169	x1 <--- f0
2	g2 0.3139527	x2 <--- f0
3	g3 0.2423659	x3 <--- f0
4	g4 0.1914355	x4 <--- f0
5	g5 0.2218987	x5 <--- f0
6	g6 0.2986721	x6 <--- f0
7	g7 0.3366396	x7 <--- f0
8	g8 0.2705857	x8 <--- f0
9	g9 0.2973876	x9 <--- f0
10	g10 0.3109651	x10 <--- f0
11	g11 0.3128489	x11 <--- f0
12	g12 0.3834218	x12 <--- f0
13	0.4337452	x1 <--- f1
14	b2 0.4608867	x2 <--- f1



```

15  b3      0.3029827  x3 <--- f1
16      0.4280793  x4 <--- f2
17  b5      0.3638542  x5 <--- f2
18  b6      0.4008132  x6 <--- f2
19      0.3205876  x7 <--- f3
20  b8      0.3818688  x8 <--- f3
21  b9      0.4115140  x9 <--- f3
22      0.2525779  x10 <--- f4
23  b11     0.2759169  x11 <--- f4
24  b12     0.4185233  x12 <--- f4
25  ev1     0.7527593  x1 <--> x1
26  ev2     0.6890172  x2 <--> x2
27  ev3     0.8494603  x3 <--> x3
28  ev4     0.7801005  x4 <--> x4
29  ev5     0.8183711  x5 <--> x5
30  ev6     0.7501438  x6 <--> x6
31  ev7     0.7838974  x7 <--> x7
32  ev8     0.7809596  x8 <--> x8
33  ev9     0.7422169  x9 <--> x9
34  ev10    0.8395051  x10 <--> x10
35  ev11    0.8259954  x11 <--> x11
36  ev12    0.6778260  x12 <--> x12
37  vf0     1.0000000  f0 <--> f0
38  vf1     1.0000000  f1 <--> f1
39  vf2     1.0000000  f2 <--> f2
40  vf3     1.0000000  f3 <--> f3
41  vf4     1.0000000  f4 <--> f4
>

```

```
> stdCoef(sem, seq. 2)
```

```

      Std. Estimate
1      0.2431169  x1 <--- f0
2  g2      0.3139527  x2 <--- f0
3  g3      0.2423659  x3 <--- f0
4  g4      0.1914355  x4 <--- f0
5  g5      0.2218987  x5 <--- f0
6  g6      0.2986721  x6 <--- f0
7  g7      0.3366396  x7 <--- f0
8  g8      0.2705857  x8 <--- f0
9  g9      0.2973876  x9 <--- f0
10 g10     0.3109651  x10 <--- f0
11 g11     0.3128489  x11 <--- f0
12 g12     0.3834218  x12 <--- f0
13      0.4337452  x1 <--- f1
14  b2      0.4608867  x2 <--- f1
15  b3      0.3029827  x3 <--- f1
16      0.4280793  x4 <--- f2
17  b5      0.3638542  x5 <--- f2
18  b6      0.4008132  x6 <--- f2
19      0.3205876  x7 <--- f3
20  b8      0.3818688  x8 <--- f3
21  b9      0.4115140  x9 <--- f3
22      0.2525779  x10 <--- f4
23  b11     0.2759169  x11 <--- f4
24  b12     0.4185233  x12 <--- f4
25  ev1     0.7527593  x1 <--> x1
26  ev2     0.6890172  x2 <--> x2
27  ev3     0.8494603  x3 <--> x3
28  ev4     0.7801005  x4 <--> x4
29  ev5     0.8183711  x5 <--> x5
30  ev6     0.7501438  x6 <--> x6
31  ev7     0.7838974  x7 <--> x7
32  ev8     0.7809596  x8 <--> x8
33  ev9     0.7422169  x9 <--> x9
34  ev10    0.8395051  x10 <--> x10
35  ev11    0.8259954  x11 <--> x11
36  ev12    0.6778260  x12 <--> x12
37  vf0     1.0000000  f0 <--> f0
38  vf1     1.0000000  f1 <--> f1
39  vf2     1.0000000  f2 <--> f2
40  vf3     1.0000000  f3 <--> f3
41  vf4     1.0000000  f4 <--> f4

```

## 探索的因子分析

## スクリーンプロット

```
library(psych)
VSS.scree(データフレーム名, または, 相関係数行列名)
```

あらかじめpsychパッケージをインストールしておく必要がある。

## 探索的因子分析

```
library(psych)
library(GPArotation)
```

オブジェクト名 <- fa(データフレーム名, nfactors=因子数, rotate="回転法", fm="推定法")

または

オブジェクト名 <- fa(相関係数行列名, n.obs=データ数, nfactors=因子数, rotate="回転法", fm="推定法")

```
print(オブジェクト名, sort=TRUE)
```

あらかじめpsychパッケージをインストールしておく必要がある。

GPArotationパッケージもインストールしておくとうい。

共通性初期推定値はSMCになる。

## 回転方法 (rotate)

直交回転: "none", "varimax", "quartimax", "bentlerT", "geominT"

斜交回転: "promax", "oblimin", "simplimax", "bentlerQ", "geominQ", "cluster"

## 推定法 (fm)

重み無し最小2乗法: "minres", 重み付き最小2乗法: "wls", 一般化重み付き最小2乗法: "gls",  
主軸法 (反復主因子法): "pa", 最尤法: "ml"

## 【重要!!】

sort=TRUE により, 各項目は因子負荷が一番大きい因子にセットされ, 各因子においては, 因子負荷の大きい順に, セットされた項目が並べ替えられる。

独自性 (u2) = 1 - 共通性 (h2) である。

```
> setwd("d:¥¥¥")
> d1 <- read.table("統計分析力尺度データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 y1 y2 y3 y4 y5 y6 y7 y8 y9 y10 xt yt
1    1  1  3  2  1  2  2  4  4  3  4  4  3  2  1  2  1  4  3  3  3  4 29 26
2    2  2  3  3  4  3  2  2  2  2  4  1  2  2  4  2  2  1  1  1  3  1 26 19
3    3  3  4  3  3  4  1  3  4  2  5  1  4  3  4  4  1  3  4  2  5  1 30 31
4    4  4  5  5  5  3  3  4  4  2  4  4  5  4  5  2  2  3  4  2  3  3 39 33
5    5  5  3  3  4  2  2  3  3  3  4  1  3  4  4  3  2  4  3  4  5  1 28 33
6    6  6  2  1  4  1  1  3  3  2  5  1  3  2  4  2  2  5  4  3  5  2 23 32
  統計 数学 批判的思考力 国語 自己効力感
1    51   48           28   72         61
2    74   53           26   66         53
3    48   60           35   71         48
4    67   68           27   67         48
5    55   49           30   66         49
6    74   63           36   83         37
>
> vn.itemx <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10")
>
> d2 <- d1[, vn.itemx]
>
>
> # 記述統計量
> dtmp <- d2
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```

      N Mean   SD   x1    x2    x3    x4    x5    x6    x7    x8    x9    x10
x1  365 3.99 0.92 1.00 0.47 0.46 0.49 0.49 0.29 0.32 0.25 0.25 0.27
x2  365 3.09 0.96 0.47 1.00 0.48 0.49 0.52 0.28 0.27 0.33 0.27 0.20
x3  365 4.06 0.85 0.46 0.48 1.00 0.45 0.47 0.31 0.25 0.26 0.27 0.20
x4  365 3.00 1.07 0.49 0.49 0.45 1.00 0.50 0.24 0.29 0.31 0.32 0.25
x5  365 2.19 0.93 0.49 0.52 0.47 0.50 1.00 0.30 0.26 0.29 0.23 0.24
x6  365 3.04 1.01 0.29 0.28 0.31 0.24 0.30 1.00 0.44 0.43 0.43 0.46
x7  365 3.12 0.99 0.32 0.27 0.25 0.29 0.26 0.44 1.00 0.44 0.37 0.35
x8  365 2.15 0.89 0.25 0.33 0.26 0.31 0.29 0.43 0.44 1.00 0.34 0.41
x9  365 3.91 0.90 0.25 0.27 0.27 0.32 0.23 0.43 0.37 0.34 1.00 0.30
x10 365 2.13 0.95 0.27 0.20 0.20 0.25 0.24 0.46 0.35 0.41 0.30 1.00

```

```

>
>
> # 共分散行列・相関係数行列
> cov.d2 <- cov(d2)
> cor.d2 <- cor(d2)
>
> #スクリープロット
> library(psych)
> VSS.scree(cor.d2) # VSS.scree(d2) でも同じ
>
>
> #因子分析の実行
> library(psych)
>
> # データ行列から分析
> fac.1 <- fa(d2, nfactors=2, rotate="promax", fm="wls")
> print(fac.1, sort=TRUE)

```

```

Factor Analysis using method = wls
Call: fa(r = d2, nfactors = 2, rotate = "promax", fm = "wls")
Standardized loadings (pattern matrix) based upon correlation matrix

```

```

      item WLS1 WLS2 h2 u2 com
x5      5  0.73 -0.02 0.51 0.49 1
x2      2  0.72 -0.01 0.51 0.49 1
x4      4  0.68  0.02 0.49 0.51 1
x1      1  0.66  0.04 0.46 0.54 1
x3      3  0.65  0.02 0.43 0.57 1
x6      6 -0.05  0.75 0.52 0.48 1
x10     10 -0.05  0.63 0.36 0.64 1
x8      8  0.04  0.62 0.42 0.58 1
x7      7  0.03  0.61 0.40 0.60 1
x9      9  0.07  0.52 0.32 0.68 1

```

```

      WLS1 WLS2
SS loadings      2.41 2.02
Proportion Var    0.24 0.20
Cumulative Var    0.24 0.44
Proportion Explained 0.54 0.46
Cumulative Proportion 0.54 1.00

```

```

      With factor correlations of
      WLS1 WLS2
WLS1    1.0  0.6
WLS2    0.6  1.0

```

```

Mean item complexity = 1
Test of the hypothesis that 2 factors are sufficient.

```

```

The degrees of freedom for the null model are 45 and the objective function was 3.03 with Chi Square of 1088.82

```

```

The degrees of freedom for the model are 26 and the objective function was 0.08

```

```

The root mean square of the residuals (RMSR) is 0.02
The df corrected root mean square of the residuals is 0.03

```

```

The harmonic number of observations is 365 with the empirical chi square 18.62 with prob < 0.85

```

```

The total number of observations was 365 with Likelihood Chi Square = 30.21 with prob < 0.26

```

	A	B	C	D	E	F	G	H	I	J	K
1 番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	
2	1	3	2	1	2	2	4	4	3	4	4
3	2	3	3	4	3	2	2	2	2	4	1
4	3	4	3	3	4	1	3	4	2	5	1
5	4	5	5	5	3	3	4	4	2	4	4
6	5	3	3	4	2	2	3	3	3	4	1
7	6	2	1	4	1	1	3	3	2	5	1
8	7	5	3	5	5	4	5	5	3	5	3
9	8	4	3	4	3	2	3	2	2	4	1
10	9	4	2	4	2	2	2	2	2	4	2
11	10	4	5	5	4	4	3	3	3	4	3
12	11	4	2	4	4	2	4	4	2	4	3
13	12	5	3	4	5	2	3	3	2	3	2
14	13	3	3	5	3	3	2	3	2	3	2
15	14	4	4	5	3	3	3	4	2	3	1
16	15	4	2	4	3	1	4	2	1	3	3
17	16	4	3	4	1	1	2	1	1	3	1
18	17	3	2	4	3	2	2	2	2	3	2
19	18	3	3	5	2	2	3	2	1	4	1
20	19	3	3	5	2	2	4	4	2	4	1
21	20	3	2	5	1	2	3	3	1	4	2

Tucker Lewis Index of factoring reliability = 0.993  
 RMSEA index = 0.021 and the 90 % confidence intervals are 0 0.048  
 BIC = -123.18  
 Fit based upon off diagonal values = 1  
 Measures of factor score adequacy

	WLS1	WLS2
Correlation of (regression) scores with factors	0.91	0.89
Multiple R square of scores with factors	0.83	0.80
Minimum correlation of possible factor scores	0.67	0.59

## > # 相関行列から分析

```
> fac.2 <- fa(r=cor.d2, n.obs=nrow(d2), nfactors=2, rotate="promax", fm="wls")
> print(fac.2, sort=TRUE)
```

Factor Analysis using method = wls

Call: fa(r = cor.d2, nfactors = 2, n.obs = nrow(d2), rotate = "promax",  
 fm = "wls")

Standardized loadings (pattern matrix) based upon correlation matrix

	item	WLS1	WLS2	h2	u2	com
x5	5	0.73	-0.02	0.51	0.49	1
x2	2	0.72	-0.01	0.51	0.49	1
x4	4	0.68	0.02	0.49	0.51	1
x1	1	0.66	0.04	0.46	0.54	1
x3	3	0.65	0.02	0.43	0.57	1
x6	6	-0.05	0.75	0.52	0.48	1
x10	10	-0.05	0.63	0.36	0.64	1
x8	8	0.04	0.62	0.42	0.58	1
x7	7	0.03	0.61	0.40	0.60	1
x9	9	0.07	0.52	0.32	0.68	1

	WLS1	WLS2
SS loadings	2.41	2.02
Proportion Var	0.24	0.20
Cumulative Var	0.24	0.44
Proportion Explained	0.54	0.46
Cumulative Proportion	0.54	1.00

With factor correlations of

	WLS1	WLS2
WLS1	1.0	0.6
WLS2	0.6	1.0

Mean item complexity = 1

Test of the hypothesis that 2 factors are sufficient.

The degrees of freedom for the null model are 45 and the objective function was 3.03 with Chi Square of 1088.82

The degrees of freedom for the model are 26 and the objective function was 0.08

The root mean square of the residuals (RMSR) is 0.02

The df corrected root mean square of the residuals is 0.03

The harmonic number of observations is 365 with the empirical chi square 18.62 with prob < 0.85

The total number of observations was 365 with Likelihood Chi Square = 30.21 with prob < 0.26

Tucker Lewis Index of factoring reliability = 0.993

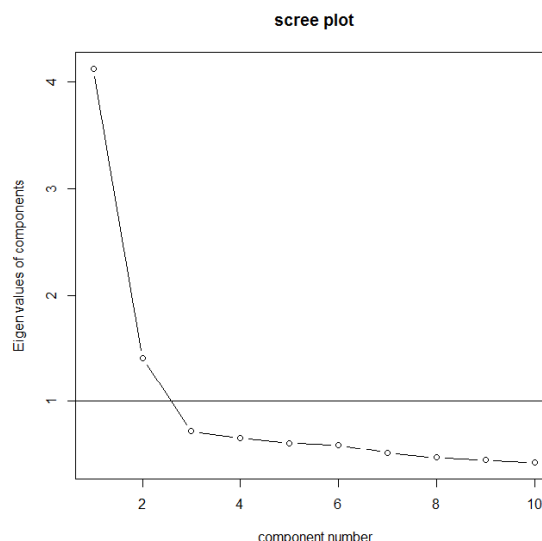
RMSEA index = 0.021 and the 90 % confidence intervals are 0 0.048

BIC = -123.18

Fit based upon off diagonal values = 1

Measures of factor score adequacy

	WLS1	WLS2
Correlation of (regression) scores with factors	0.91	0.89
Multiple R square of scores with factors	0.83	0.80
Minimum correlation of possible factor scores	0.67	0.59



# 17 潜在変数のある構造方程式 モデリング(共分散構造分析)

## 多重指標モデル

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

```
モデル名 <- specifyEquations()
構造方程式
```

## 非標準化解の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
```

もしくは

```
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+...)
```

**【重要】 共分散行列は、分析に用いる観測変数だけのものにする**

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

空白行があるところで、モデルの設定が終了したとみなされる。

(逆に、行をあげないと、モデル設定が終了したことにならない)

## 結果の出力

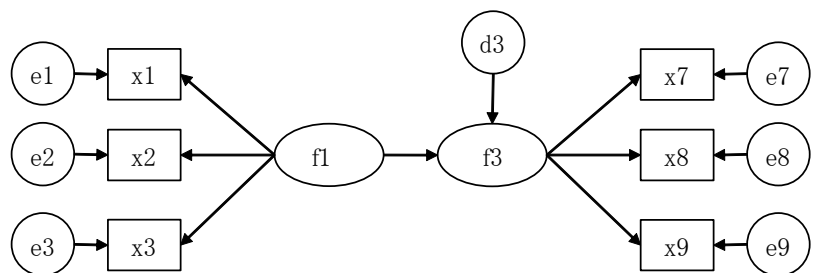
```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

## モデル部分のスキプトの例

```
seq.1 <- specifyEquations()
x1 = 1*f1
x2 = b2*f1
x3 = b3*f1
x7 = 1*f3
x8 = b8*f3
x9 = b9*f3
f3 = a31*f1
V(x1) = ev1
V(x2) = ev2
V(x3) = ev3
V(x7) = ev7
V(x8) = ev8
V(x9) = ev9
V(f3) = evf3
V(f1) = vf1
```



```
> setwd("d:¥¥¥")
> d1 <- read.table("拡張SEMデータ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12
1  4  3  2  3  4  4  2  4  2  4  4  3
2  2  3  2  5  3  4  5  3  3  4  4  4
3  1  4  2  2  3  3  4  3  2  3  4  4
4  1  5  2  2  3  2  3  2  2  5  4  3
```

```

5 2 3 1 4 2 1 3 3 2 1 1 1
6 2 4 3 3 5 4 4 3 4 2 4 1
>
>
> # 記述統計量
> library(psych)
> describe(d1)
  vars      n mean    sd median trimmed  mad min max range  skew kurtosis   se
x1     1 276 3.01 0.98      3    3.01 1.48    1  5     4  0.02   -0.36 0.06
x2     2 276 3.01 1.03      3    3.05 1.48    1  5     4 -0.17   -0.58 0.06
x3     3 276 3.00 0.99      3    3.01 1.48    1  5     4 -0.07   -0.44 0.06
x4     4 276 2.99 1.02      3    2.97 1.48    1  5     4  0.09   -0.50 0.06
x5     5 276 3.00 1.00      3    3.05 1.48    1  5     4 -0.23   -0.51 0.06
x6     6 276 3.03 1.04      3    3.02 1.48    1  5     4  0.01   -0.43 0.06
x7     7 276 3.03 0.99      3    3.03 1.48    1  5     4 -0.01   -0.45 0.06
x8     8 276 3.01 1.01      3    3.05 1.48    1  5     4 -0.18   -0.49 0.06
x9     9 276 3.00 1.01      3    3.01 1.48    1  5     4 -0.06   -0.66 0.06
x10    10 276 2.99 1.03      3    3.00 1.48    1  5     4 -0.03   -0.49 0.06
x11    11 276 2.99 1.00      3    2.99 1.48    1  5     4  0.00   -0.36 0.06
x12    12 276 2.99 1.01      3    3.00 1.48    1  5     4 -0.06   -0.42 0.06>
>

```

```
> # 共分散行列・相関係数行列
```

```

> cov.d1 <- cov(d1)
> round(cov.d1, 2)
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10     x11     x12
x1  0.96  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.07  0.22 -0.01  0.03  0.04  0.28  0.06  0.09  0.08  0.09  0.11
x3  0.19  0.22  0.99 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.05  0.20  0.24  0.03  0.04  0.02  0.09  0.08  0.09
x5 -0.01  0.03  0.03  0.20  1.01  0.22  0.05  0.23  0.02  0.08  0.08  0.11
x6  0.03  0.04  0.01  0.24  0.22  1.09  0.09  0.05  0.07  0.09  0.30  0.12
x7  0.11  0.28  0.12  0.03  0.05  0.09  0.98  0.21  0.23  0.07  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.05  0.21  1.03  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.03  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.07  0.09  0.06  1.05  0.17  0.23
x11 0.07  0.09  0.01  0.08  0.08  0.30  0.05  0.03  0.04  0.17  0.99  0.24
x12 0.07  0.11  0.07  0.09  0.11  0.12  0.07  0.11  0.22  0.23  0.24  1.01

```

```

> cor.d1 <- cor(d1)
> round(cor.d1, 2)
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10     x11     x12
x1  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09  0.08  0.09  0.10
x3  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02  0.09  0.08  0.08
x5 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02  0.08  0.08  0.10
x6  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07  0.09  0.29  0.11
x7  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23  0.06  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.04  0.21  1.00  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.00  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.06  0.09  0.06  1.00  0.17  0.22
x11 0.07  0.09  0.01  0.08  0.08  0.29  0.05  0.03  0.04  0.17  1.00  0.24
x12 0.07  0.10  0.07  0.08  0.10  0.11  0.07  0.11  0.22  0.22  0.24  1.00

```

```
> # 適合度指標の設定
```

```

> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR",
"R", "AIC", "AICc", "BIC", "CAIC"))

```

## &gt; # 多重指標モデル

seq.1 &lt;- specifyEquations()

```

1: x1 = 1*f1
2: x2 = b2*f1
3: x3 = b3*f1
4: x7 = 1*f3
5: x8 = b8*f3
6: x9 = b9*f3
7: f3 = a31*f1
8: V(x1) = ev1
9: V(x2) = ev2
10: V(x3) = ev3
11: V(x7) = ev7
12: V(x8) = ev8
13: V(x9) = ev9
14: V(f3) = evf3
15: V(f1) = vf1
16:

```

Read 15 items

	A	B	C	D	E	F	G	H	I	J	K
1	番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
2	1	3	2	1	2	2	4	4	3	4	4
3	2	3	3	4	3	2	2	2	2	4	1
4	3	4	3	3	4	1	3	4	2	5	1
5	4	5	5	5	3	3	4	4	2	4	4
6	5	3	3	4	2	2	3	3	3	4	1
7	6	2	1	4	1	1	3	3	2	5	1
8	7	5	3	5	5	4	5	5	3	5	3
9	8	4	3	4	3	2	3	2	2	4	1
10	9	4	2	4	2	2	2	2	2	4	2
11	10	4	5	5	4	4	3	3	3	4	3
12	11	4	2	4	4	2	4	4	2	4	3
13	12	5	3	4	5	2	3	3	2	3	2
14	13	3	3	5	3	3	2	3	2	3	2
15	14	4	4	5	3	3	3	4	2	3	1
16	15	4	2	4	3	1	4	2	1	3	3
17	16	4	3	4	1	1	2	1	1	3	1
18	17	3	2	4	3	2	2	2	2	3	2
19	18	3	3	5	2	2	3	2	1	4	1
20	19	3	3	5	2	2	4	4	2	4	1
21	20	3	2	5	1	2	3	3	1	4	2

## &gt; # 非標準化解の推定

sem.seq.1 &lt;- sem(seq.1, S=cov.d1, N=nrow(d1))

summary(sem.seq.1)

Model Chisquare = 9.187096 Df = 8 Pr(&gt;Chisq) = 0.3267596

Goodness-of-fit index = 0.9885242

Adjusted goodness-of-fit index = 0.969876

RMSEA index = 0.02322905 90% CI: (NA, 0.07683467)

Bentler-Bonett NFI = 0.9125166

Tucker-Lewis NNFI = 0.975273

Bentler CFI = 0.9868123

Bentler RNI = 0.9868123

Bollen IFI = 0.9877638

SRMR = 0.03268035

AIC = 35.1871

AICc = 10.57641

BIC = -35.77611

CAIC = -43.77611

## Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.122548	-0.332167	-0.005191	-0.031209	0.056294	1.362423

## R-square for Endogenous Variables

x1	x2	x3	f3	x7	x8	x9
0.1912	0.3961	0.1360	0.2738	0.3787	0.1409	0.1694

## Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
b2	1.5186975	0.43869639	3.461842	5.364926e-04	x2 <--- f1
b3	0.8561944	0.25021723	3.421804	6.220703e-04	x3 <--- f1
b8	0.6234700	0.19324808	3.226267	1.254161e-03	x8 <--- f3
b9	0.6838182	0.20630464	3.314604	9.177304e-04	x9 <--- f3
a31	0.7444214	0.23023345	3.233333	1.223549e-03	f3 <--- f1
ev1	0.7762467	0.08607366	9.018401	1.908553e-19	x1 <--> x1
ev2	0.6455500	0.13005700	4.963593	6.920086e-07	x2 <--> x2
ev3	0.8545419	0.08520639	10.029082	1.135676e-23	x3 <--> x3
ev7	0.6094471	0.11954549	5.098035	3.431974e-07	x7 <--> x7
ev8	0.8808253	0.08941441	9.851044	6.783564e-23	x8 <--> x8
ev9	0.8517256	0.09136838	9.321886	1.142907e-20	x9 <--> x9
evf3	0.2698155	0.11038347	2.444347	1.451145e-02	f3 <--> f3
vf1	0.1835426	0.07310119	2.510801	1.204575e-02	f1 <--> f1

Iterations = 31

&gt;

&gt; sem.seq.2 &lt;- sem(seq.1, data=d1, formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)

summary(sem.seq.2)

Model Chisquare = 9.187096 Df = 8 Pr(&gt;Chisq) = 0.3267596

Goodness-of-fit index = 0.9885242



Adjusted goodness-of-fit index = 0.969876  
 RMSEA index = 0.02322905 90% CI: (NA, 0.07683467)  
 Bentler-Bonett NFI = 0.9125166  
 Tucker-Lewis NNFI = 0.975273  
 Bentler CFI = 0.9868123  
 Bentler RNI = 0.9868123  
 Bollen IFI = 0.9877638  
 SRMR = 0.03268035  
 AIC = 35.1871  
 AICc = 10.57641  
 BIC = -35.77611  
 CAIC = -43.77611

## Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-1.122548	-0.332167	-0.005191	-0.031209	0.056294	1.362423

## R-square for Endogenous Variables

x1	x2	x3	f3	x7	x8	x9
0.1912	0.3961	0.1360	0.2738	0.3787	0.1409	0.1694

## Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
b2	1.5186975	0.43869639	3.461842	5.364926e-04	x2 <--- f1
b3	0.8561944	0.25021723	3.421804	6.220703e-04	x3 <--- f1
b8	0.6234700	0.19324808	3.226267	1.254161e-03	x8 <--- f3
b9	0.6838182	0.20630464	3.314604	9.177304e-04	x9 <--- f3
a31	0.7444214	0.23023345	3.233333	1.223549e-03	f3 <--- f1
ev1	0.7762467	0.08607366	9.018401	1.908553e-19	x1 <--> x1
ev2	0.6455500	0.13005700	4.963593	6.920086e-07	x2 <--> x2
ev3	0.8545419	0.08520639	10.029082	1.135676e-23	x3 <--> x3
ev7	0.6094471	0.11954549	5.098035	3.431974e-07	x7 <--> x7
ev8	0.8808253	0.08941441	9.851044	6.783564e-23	x8 <--> x8
ev9	0.8517256	0.09136838	9.321886	1.142907e-20	x9 <--> x9
evf3	0.2698155	0.11038347	2.444347	1.451145e-02	f3 <--> f3
vf1	0.1835426	0.07310119	2.510801	1.204575e-02	f1 <--> f1

Iterations = 31

&gt;

&gt;

&gt; # 標準化解の推定

&gt; stdCoef(sem, seq. 1)

	Std. Estimate	
1	0.4373010	x1 <--- f1
2	b2 0.6293252	x2 <--- f1
3	b3 0.3688269	x3 <--- f1
4	0.6154132	x7 <--- f3
5	b8 0.3753166	x8 <--- f3
6	b9 0.4116027	x9 <--- f3
7	a31 0.5232285	f3 <--- f1
8	ev1 0.8087679	x1 <--> x1
9	ev2 0.6039498	x2 <--> x2
10	ev3 0.8639667	x3 <--> x3
11	ev7 0.6212666	x7 <--> x7
12	ev8 0.8591374	x8 <--> x8
13	ev9 0.8305832	x9 <--> x9
14	evf3 0.7262319	f3 <--> f3
15	vf1 1.0000000	f1 <--> f1

&gt;

&gt; stdCoef(sem, seq. 2)

	Std. Estimate	
1	0.4373010	x1 <--- f1
2	b2 0.6293252	x2 <--- f1
3	b3 0.3688269	x3 <--- f1
4	0.6154132	x7 <--- f3
5	b8 0.3753166	x8 <--- f3
6	b9 0.4116027	x9 <--- f3
7	a31 0.5232285	f3 <--- f1
8	ev1 0.8087679	x1 <--> x1
9	ev2 0.6039498	x2 <--> x2
10	ev3 0.8639667	x3 <--> x3

```

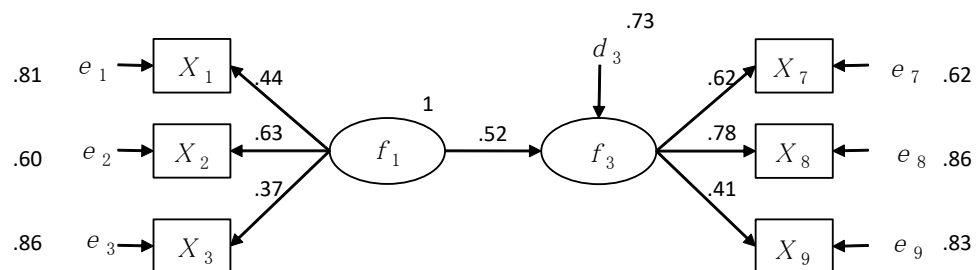
11 ev7      0.6212666 x7 <--> x7
12 ev8      0.8591374 x8 <--> x8
13 ev9      0.8305832 x9 <--> x9
14 evf3     0.7262319 f3 <--> f3
15 vfl      1.0000000 f1 <--> f1

```

```

>
>
>
>
>
>

```



$\chi^2=9.19$ ,  $df=8$ ,  $p=0.327$ ,  
 AGFI=0.97, RMSEA=0.02, CFI=0.99

## 潜在変数の構造方程式モデリング — semパッケージ

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                               "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

```
モデル名 <- specifyEquations()
構造方程式
```

## 非標準化解の推定

```
semオブジェクト名 <- sem(モデル名, S=共分散行列, N=標本サイズ)
```

もしくは

```
semオブジェクト名 <- sem(モデル名, data=データ行列, formula=~変数1+変数2+…)
```

【重要】 共分散行列は、分析に用いる観測変数だけのものにすること

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

空白行があるところで、モデルの設定が終了したとみなされる。

(逆に、行をあげないと、モデル設定が終了したことにならない)

## 結果の出力

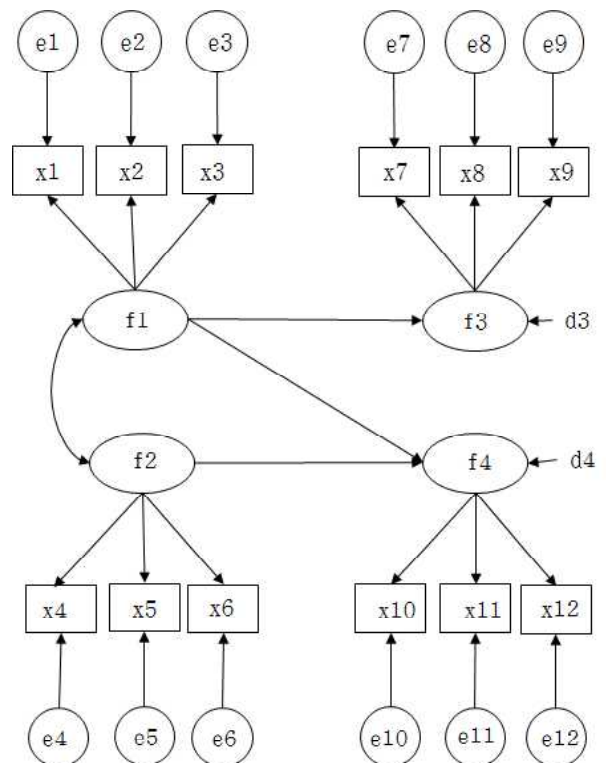
```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

## モデル部分のスキプトの例

```
seq.1 <- specifyEquations()
x1 = 1*f1                                # 測定方程式
x2 = b2*f1
x3 = b3*f1
x4 = 1*f2
x5 = b5*f2
x6 = b6*f2
x7 = 1*f3
x8 = b8*f3
x9 = b9*f3
x10 = 1*f4
x11 = b11*f4
x12 = b12*f4
f3 = a31*f1                                # 潜在変数間の予測式
f4 = a41*f1 + a42*f2
V(x1) = ev1                                # 内生変数の誤差分散
V(x2) = ev2
V(x3) = ev3
V(x4) = ev4
V(x5) = ev5
V(x6) = ev6
V(x7) = ev7
V(x8) = ev8
V(x9) = ev9
V(x10) = ev10
V(x11) = ev11
V(x12) = ev12
V(f3) = evf3
V(f4) = evf4
V(f1) = vf1                                # 外生変数の分散
V(f2) = vf2
C(f1, f2) = cf12                           # 外生変数の共分散
```



```

> setwd("d:¥¥")
> d1 <- read.table("拡張SEMデータ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12
1  4  3  2  3  4  4  2  4  2  4  4  3
2  2  3  2  5  3  4  5  3  3  4  4  4
3  1  4  2  2  3  3  4  3  2  3  4  4
4  1  5  2  2  3  2  3  2  2  5  4  3
5  2  3  1  4  2  1  3  3  2  1  1  1
6  2  4  3  3  5  4  4  3  4  2  4  1
>
>
> # 記述統計量
> library(psych)
> describe(d1)
  vars      n mean      sd median trimmed  mad min max range  skew kurtosis   se
x1    1 276 3.01 0.98      3   3.01 1.48    1  5    4  0.02   -0.36 0.06
x2    2 276 3.01 1.03      3   3.05 1.48    1  5    4 -0.17   -0.58 0.06
x3    3 276 3.00 0.99      3   3.01 1.48    1  5    4 -0.07   -0.44 0.06
x4    4 276 2.99 1.02      3   2.97 1.48    1  5    4  0.09   -0.50 0.06
x5    5 276 3.00 1.00      3   3.05 1.48    1  5    4 -0.23   -0.51 0.06
x6    6 276 3.03 1.04      3   3.02 1.48    1  5    4  0.01   -0.43 0.06
x7    7 276 3.03 0.99      3   3.03 1.48    1  5    4 -0.01   -0.45 0.06
x8    8 276 3.01 1.01      3   3.05 1.48    1  5    4 -0.18   -0.49 0.06
x9    9 276 3.00 1.01      3   3.01 1.48    1  5    4 -0.06   -0.66 0.06
x10  10 276 2.99 1.03      3   3.00 1.48    1  5    4 -0.03   -0.49 0.06
x11  11 276 2.99 1.00      3   2.99 1.48    1  5    4  0.00   -0.36 0.06
x12  12 276 2.99 1.01      3   3.00 1.48    1  5    4 -0.06   -0.42 0.06>
>

> # 共分散行列・相関係数行列
> cov.d1 <- cov(d1)
> round(cov.d1, 2)
  x1    x2    x3    x4    x5    x6    x7    x8    x9    x10   x11   x12
x1  0.96  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.07  0.22 -0.01  0.03  0.04  0.28  0.06  0.09  0.08  0.09  0.11
x3  0.19  0.22  0.99 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.05  0.20  0.24  0.03  0.04  0.02  0.09  0.08  0.09
x5 -0.01  0.03  0.03  0.20  1.01  0.22  0.05  0.23  0.02  0.08  0.08  0.11
x6  0.03  0.04  0.01  0.24  0.22  1.09  0.09  0.05  0.07  0.09  0.30  0.12
x7  0.11  0.28  0.12  0.03  0.05  0.09  0.98  0.21  0.23  0.07  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.05  0.21  1.03  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.03  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.07  0.09  0.06  1.05  0.17  0.23
x11 0.07  0.09  0.01  0.08  0.08  0.30  0.05  0.03  0.04  0.17  0.99  0.24
x12 0.07  0.11  0.07  0.09  0.11  0.12  0.07  0.11  0.22  0.23  0.24  1.01

> cor.d1 <- cor(d1)
> round(cor.d1, 2)
  x1    x2    x3    x4    x5    x6    x7    x8    x9    x10   x11   x12
x1  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09  0.08  0.09  0.10
x3  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02  0.09  0.08  0.08
x5 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02  0.08  0.08  0.10
x6  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07  0.09  0.29  0.11
x7  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23  0.06  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.04  0.21  1.00  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.00  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.06  0.09  0.06  1.00  0.17  0.22
x11 0.07  0.09  0.01  0.08  0.08  0.29  0.05  0.03  0.04  0.17  1.00  0.24
x12 0.07  0.10  0.07  0.08  0.10  0.11  0.07  0.11  0.22  0.22  0.24  1.00

>
>
> # 適合度指標の設定
> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI", "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>

```

## &gt; # モデルの設定

&gt; seq.1 &lt;- specifyEquations()

```

1: x1 = 1*f1
2: x2 = b2*f1
3: x3 = b3*f1
4: x4 = 1*f2
5: x5 = b5*f2
6: x6 = b6*f2
7: x7 = 1*f3
8: x8 = b8*f3
9: x9 = b9*f3
10: x10 = 1*f4
11: x11 = b11*f4
12: x12 = b12*f4
13: f3 = a31*f1
14: f4 = a41*f1 + a42*f2
15: V(x1) = ev1
16: V(x2) = ev2
17: V(x3) = ev3
18: V(x4) = ev4
19: V(x5) = ev5
20: V(x6) = ev6
21: V(x7) = ev7
22: V(x8) = ev8
23: V(x9) = ev9
24: V(x10) = ev10
25: V(x11) = ev11
26: V(x12) = ev12
27: V(f3) = evf3
28: V(f4) = evf4
29: V(f1) = vf1
30: V(f2) = vf2
31: C(f1, f2) = cf12
32:

```

Read 31 items

	A	B	C	D	E	F	G	H	I	J	K
1	番号	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10
2	1	3	2	1	2	2	4	4	3	4	4
3	2	3	3	4	3	2	2	2	2	4	1
4	3	4	3	3	4	1	3	4	2	5	1
5	4	5	5	5	3	3	4	4	2	4	4
6	5	3	3	4	2	2	3	3	3	4	1
7	6	2	1	4	1	1	3	3	2	5	1
8	7	5	3	5	5	4	5	5	3	5	3
9	8	4	3	4	3	2	3	2	2	4	1
10	9	4	2	4	2	2	2	2	2	4	2
11	10	4	5	5	4	4	3	3	3	4	3
12	11	4	2	4	4	2	4	4	2	4	3
13	12	5	3	4	5	2	3	3	2	3	2
14	13	3	3	5	3	3	2	3	2	3	2
15	14	4	4	5	3	3	3	4	2	3	1
16	15	4	2	4	3	1	4	2	1	3	3
17	16	4	3	4	1	1	2	1	1	3	1
18	17	3	2	4	3	2	2	2	2	3	2
19	18	3	3	5	2	2	3	2	1	4	1
20	19	3	3	5	2	2	4	4	2	4	1
21	20	3	2	5	1	2	3	3	1	4	2

## &gt; # 非標準化解の推定

&gt; sem.seq.1 &lt;- sem(seq.1, S=cov.d1, N=nrow(d1))

&gt; summary(sem.seq.1)

```

Model Chisquare = 83.02649   Df = 50 Pr(>Chisq) = 0.002305156
Goodness-of-fit index = 0.953118
Adjusted goodness-of-fit index = 0.9268641
RMSEA index = 0.04900945   90% CI: (0.02933125, 0.06721064)
Bentler-Bonett NFI = 0.6987328
Tucker-Lewis NNFI = 0.7919997
Bentler CFI = 0.842424
Bentler RNI = 0.842424
Bollen IFI = 0.8536001
SRMR = 0.05428157
AIC = 139.0265
AICc = 89.60139
BIC = -197.9936
CAIC = -247.9936

```

## Normalized Residuals

```

Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
-1.1979 -0.4154 -0.1110  0.1063  0.2682  3.7707

```

## R-square for Endogenous Variables

```

x1    x2    x3    x4    x5    x6    f3    x7    x8    x9    f4    x10    x11
0.2005 0.3366 0.1483 0.1676 0.1510 0.3358 0.3151 0.3396 0.1550 0.1889 0.3986 0.1587 0.2398
x12
0.2356

```

## Parameter Estimates

```

Estimate Std Error z value Pr(>|z|)
b2 1.36735501 0.35959534 3.802483 1.432533e-04 x2 <--- f1
b3 0.87319115 0.24829770 3.516711 4.369299e-04 x3 <--- f1
b5 0.93095463 0.28755691 3.237462 1.205979e-03 x5 <--- f2
b6 1.44205072 0.44080666 3.271391 1.070200e-03 x6 <--- f2

```

```

b8 0.69064973 0.20222874 3.415191 6.373734e-04 x8 <--- f3
b9 0.76264446 0.21695812 3.515169 4.394741e-04 x9 <--- f3
b11 1.19289364 0.34403278 3.467384 5.255512e-04 x11 <--- f4
b12 1.19518272 0.34506435 3.463652 5.328960e-04 x12 <--- f4
a31 0.73855824 0.22437994 3.291552 9.963618e-04 f3 <--- f1
a41 0.27972608 0.14094606 1.984632 4.718541e-02 f4 <--- f1
a42 0.48591374 0.19143208 2.538309 1.113896e-02 f4 <--- f2
ev1 0.76736460 0.08448196 9.083177 1.054506e-19 x1 <--> x1
ev2 0.70911152 0.10882447 6.516104 7.215706e-11 x2 <--> x2
ev3 0.84237443 0.08478715 9.935166 2.926887e-23 x3 <--> x3
ev4 0.87161274 0.09408803 9.263801 1.972820e-20 x4 <--> x4
ev5 0.85520193 0.08927338 9.579585 9.743548e-22 x5 <--> x5
ev6 0.72177968 0.12479743 5.783610 7.311424e-09 x6 <--> x6
ev7 0.64784113 0.10759778 6.020952 1.733937e-09 x7 <--> x7
ev8 0.86634009 0.08962579 9.666192 4.197044e-22 x8 <--> x8
ev9 0.83169514 0.09190842 9.049173 1.440558e-19 x9 <--> x9
ev10 0.88718878 0.09237533 9.604174 7.677062e-22 x10 <--> x10
ev11 0.75453604 0.09302534 8.111080 5.017166e-16 x11 <--> x11
ev12 0.77543984 0.09461868 8.195421 2.497185e-16 x12 <--> x12
evf3 0.22817241 0.09387725 2.430540 1.507635e-02 f3 <--> f3
evf4 0.10061013 0.05046156 1.993797 4.617421e-02 f4 <--> f4
vf1 0.19242458 0.07260671 2.650231 8.043671e-03 f1 <--> f1
vf2 0.17544899 0.07599748 2.308616 2.096491e-02 f2 <--> f2
cf12 0.03756293 0.02467677 1.522198 1.279595e-01 f2 <--> f1

```

```

Iterations = 53
>

```

```

> sem.seq.2 <- sem(seq.1, data=d1, formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)
> summary(sem.seq.2)

```

```

Model Chisquare = 83.02649 Df = 50 Pr(>Chisq) = 0.002305156
Goodness-of-fit index = 0.953118
Adjusted goodness-of-fit index = 0.9268641
RMSEA index = 0.04900945 90% CI: (0.02933125, 0.06721064)
Bentler-Bonett NFI = 0.6987328
Tucker-Lewis NNFI = 0.7919997
Bentler CFI = 0.842424
Bentler RNI = 0.842424
Bollen IFI = 0.8536001
SRMR = 0.05428157
AIC = 139.0265
AICc = 89.60139
BIC = -197.9936
CAIC = -247.9936

```

```

Normalized Residuals
  Min. 1st Qu. Median Mean 3rd Qu. Max.
-1.1979 -0.4154 -0.1110 0.1063 0.2682 3.7707

```

```

R-square for Endogenous Variables
  x1    x2    x3    x4    x5    x6    f3    x7    x8    x9    f4    x10    x11
0.2005 0.3366 0.1483 0.1676 0.1510 0.3358 0.3151 0.3396 0.1550 0.1889 0.3986 0.1587 0.2398
  x12
0.2356

```

```

Parameter Estimates
  Estimate Std Error z value Pr(>|z|)
b2 1.36735501 0.35959534 3.802483 1.432533e-04 x2 <--- f1
b3 0.87319115 0.24829770 3.516711 4.369299e-04 x3 <--- f1
b5 0.93095463 0.28755691 3.237462 1.205979e-03 x5 <--- f2
b6 1.44205072 0.44080666 3.271391 1.070200e-03 x6 <--- f2
b8 0.69064973 0.20222874 3.415191 6.373734e-04 x8 <--- f3
b9 0.76264446 0.21695812 3.515169 4.394741e-04 x9 <--- f3
b11 1.19289364 0.34403278 3.467384 5.255512e-04 x11 <--- f4
b12 1.19518272 0.34506435 3.463652 5.328960e-04 x12 <--- f4
a31 0.73855824 0.22437994 3.291552 9.963618e-04 f3 <--- f1
a41 0.27972608 0.14094606 1.984632 4.718541e-02 f4 <--- f1
a42 0.48591374 0.19143208 2.538309 1.113896e-02 f4 <--- f2
ev1 0.76736460 0.08448196 9.083177 1.054506e-19 x1 <--> x1
ev2 0.70911152 0.10882447 6.516104 7.215706e-11 x2 <--> x2
ev3 0.84237443 0.08478715 9.935166 2.926887e-23 x3 <--> x3

```

```

ev4 0.87161274 0.09408803 9.263801 1.972820e-20 x4 <--> x4
ev5 0.85520193 0.08927338 9.579585 9.743548e-22 x5 <--> x5
ev6 0.72177968 0.12479743 5.783610 7.311424e-09 x6 <--> x6
ev7 0.64784113 0.10759778 6.020952 1.733937e-09 x7 <--> x7
ev8 0.86634009 0.08962579 9.666192 4.197044e-22 x8 <--> x8
ev9 0.83169514 0.09190842 9.049173 1.440558e-19 x9 <--> x9
ev10 0.88718878 0.09237533 9.604174 7.677062e-22 x10 <--> x10
ev11 0.75453604 0.09302534 8.111080 5.017166e-16 x11 <--> x11
ev12 0.77543984 0.09461868 8.195421 2.497185e-16 x12 <--> x12
evf3 0.22817241 0.09387725 2.430540 1.507635e-02 f3 <--> f3
evf4 0.10061013 0.05046156 1.993797 4.617421e-02 f4 <--> f4
vfl 0.19242458 0.07260671 2.650231 8.043671e-03 f1 <--> f1
vfl2 0.17544899 0.07599748 2.308616 2.096491e-02 f2 <--> f2
cfl2 0.03756293 0.02467677 1.522198 1.279595e-01 f2 <--> f1

```

Iterations = 53

>  
>

### > # 標準化解の推定

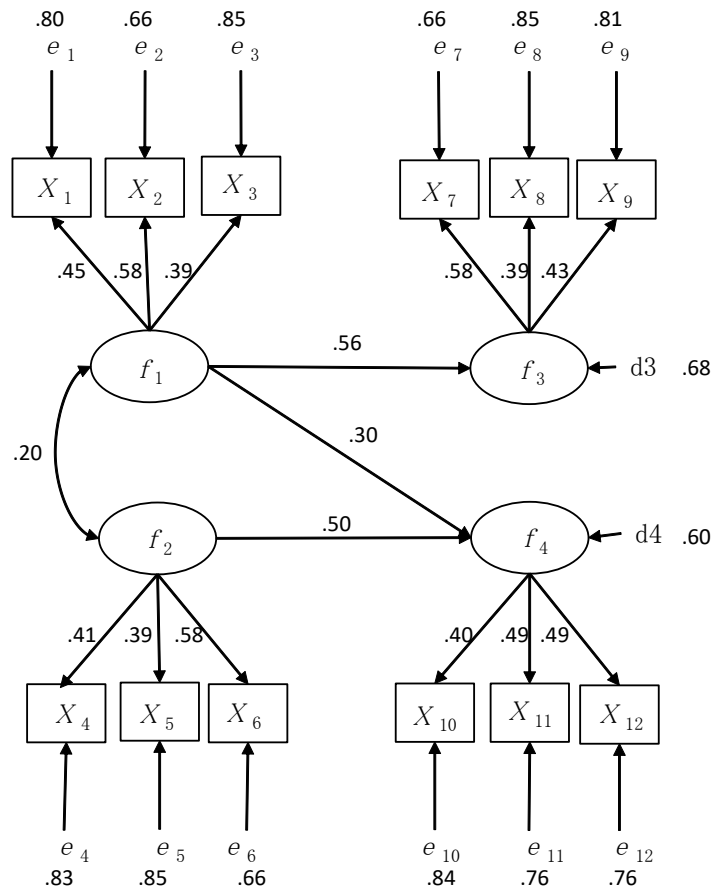
> stdCoef(sem. seq. 1)

	Std. Estimate	
1	0.4477570	x1 <--- f1
2	0.5801591	x2 <--- f1
3	0.3851425	x3 <--- f1
4	0.4093448	x4 <--- f2
5	0.3885378	x5 <--- f2
6	0.5794495	x6 <--- f2
7	0.5827475	x7 <--- f3
8	0.3936893	x8 <--- f3
9	0.4346837	x9 <--- f3
10	0.3983190	x10 <--- f4
11	0.4897399	x11 <--- f4
12	0.4853742	x12 <--- f4
13	0.5613136	f3 <--- f1
14	0.2999926	f4 <--- f1
15	0.4976017	f4 <--- f2
16	0.7995137	x1 <--> x1
17	0.6634155	x2 <--> x2
18	0.8516652	x3 <--> x3
19	0.8324368	x4 <--> x4
20	0.8490384	x5 <--> x5
21	0.6642383	x6 <--> x6
22	0.6604053	x7 <--> x7
23	0.8450087	x8 <--> x8
24	0.8110501	x9 <--> x9
25	0.8413420	x10 <--> x10
26	0.7601548	x11 <--> x11
27	0.7644119	x12 <--> x12
28	0.6849270	f3 <--> f3
29	0.6013624	f4 <--> f4
30	1.0000000	f1 <--> f1
31	1.0000000	f2 <--> f2
32	0.2044343	f2 <--> f1

>

> stdCoef(sem. seq. 2)

	Std. Estimate	
1	0.4477570	x1 <--- f1
2	0.5801591	x2 <--- f1
3	0.3851425	x3 <--- f1
4	0.4093448	x4 <--- f2
5	0.3885378	x5 <--- f2
6	0.5794495	x6 <--- f2
7	0.5827475	x7 <--- f3
8	0.3936893	x8 <--- f3
9	0.4346837	x9 <--- f3
10	0.3983190	x10 <--- f4
11	0.4897399	x11 <--- f4
12	0.4853742	x12 <--- f4
13	0.5613136	f3 <--- f1
14	0.2999926	f4 <--- f1
15	0.4976017	f4 <--- f2



$\chi^2=83.03, df=50, p=0.002,$   
AGFI=0.93, RMSEA=0.05, CFI=0.84

```
16 ev1      0.7995137    x1 <--> x1
17 ev2      0.6634155    x2 <--> x2
18 ev3      0.8516652    x3 <--> x3
19 ev4      0.8324368    x4 <--> x4
20 ev5      0.8490384    x5 <--> x5
21 ev6      0.6642383    x6 <--> x6
22 ev7      0.6604053    x7 <--> x7
23 ev8      0.8450087    x8 <--> x8
24 ev9      0.8110501    x9 <--> x9
25 ev10     0.8413420    x10 <--> x10
26 ev11     0.7601548    x11 <--> x11
27 ev12     0.7644119    x12 <--> x12
28 evf3     0.6849270     f3 <--> f3
29 evf4     0.6013624     f4 <--> f4
30 vf1      1.0000000     f1 <--> f1
31 vf2      1.0000000     f2 <--> f2
32 cf12     0.2044343     f2 <--> f1
>
>
```



## 潜在変数の構造方程式モデリング — lavaanパッケージ

パッケージの読み込み  
library(lavaan)

あらかじめlavaanパッケージをインストールしておく必要がある。

## モデルの設定

```
モデル名 <- '
# 測定変数の指定 (= ~ を使う)
潜在変数名1 ~ 観測変数1 + 観測変数2 + ...
潜在変数名2 ~ 観測変数1 + 観測変数2 + ...

# 回帰式 (~ を使う)
従属変数名1 ~ 説明変数1 + 説明変数2 + ...
従属変数名2 ~ 説明変数1 + 説明変数2 + ...

# 分散, 共分散 (~ を使う)
変数名i ~ 変数名i      # 分散
, 変数名j ~ 変数名k    # 共分散
```

モデル部分はシングルカンマ (') でくくる。

すべての変数の分散を推定する場合は分散の指定は省略できる。

## パラメタ値の推定

```
lavaanオブジェクト名 <- sem(モデル名, data=データ名)      または
```

```
lavaanオブジェクト名 <- lavaan(モデル名, data=データ名, model.type="sem",
                                auto.var=TRUE, auto.fix.first=TRUE, auto.cov.lv.x=FALSE, auto.cov.y=FALSE)
```

auto.var=TRUE : すべての変数の分散または残差分散を推定する

auto.fix.first=TRUE : 非標準化解において測定変数のうち最初の1つのパス係数を1に固定する

auto.cov.lv.x=FALSE : 潜在変数間の共分散を自動的に推定しない

auto.cov.y=FALSE : 従属変数の残差間の共分散を自動的に推定しない

```
summary(lavaanオブジェクト名, fit.measures=TRUE, standardized=TRUE)
```

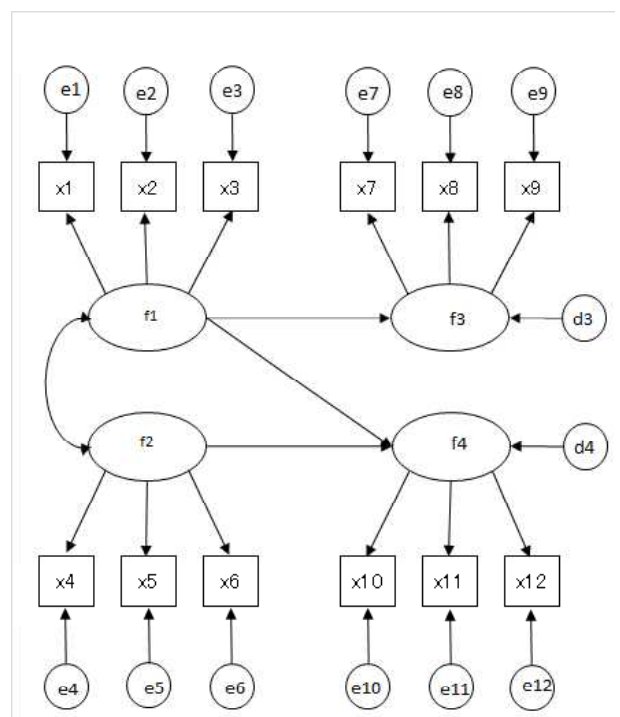
## モデル部分のスキプトの例

```
model.1 <- '
# latent variable definitions
f1 ~ x1 + x2 + x3
f2 ~ x4 + x5 + x6
f3 ~ x7 + x8 + x9
f4 ~ x10 + x11 + x12

# regression
f3 ~ f1
f4 ~ f1 + f2

# variances and covariances
f1 ~ f2
```

# すべての分散を推定するので、分散の式は省略している



```

> setwd("d:¥¥")
> d1 <- read.table("共分散構造分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12
1  4  3  2  3  4  4  2  4  2  4  4  3
2  2  3  2  5  3  4  5  3  3  4  4  4
3  1  4  2  2  3  3  4  3  2  3  4  4
4  1  5  2  2  3  2  3  2  2  5  4  3
5  2  3  1  4  2  1  3  3  2  1  1  1
6  2  4  3  3  5  4  4  3  4  2  4  1
>

```

```

> #データフレームの行数（標本の大きさ）、平均、標準偏差、共分散、相関
> n.d1 <- nrow(d1)
> mean.d1 <- apply(d1, 2, mean)
> sd.d1 <- apply(d1, 2, sd)
> cor.d1 <- cor(d1)
> round(data.frame(n.d1, mean.d1, sd.d1, cor.d1), 2)

```

```

  n.d1 mean.d1 sd.d1  x1  x2  x3  x4  x5  x6  x7  x8  x9
x1  276    3.01  0.98  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08
x2  276    3.01  1.03  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09
x3  276    3.00  0.99  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09
x4  276    2.99  1.02  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02
x5  276    3.00  1.00 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02
x6  276    3.03  1.04  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07
x7  276    3.03  0.99  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23
x8  276    3.01  1.01  0.07  0.06  0.08  0.04  0.23  0.04  0.21  1.00  0.24
x9  276    3.00  1.01  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.00
x10 276    2.99  1.03  0.06  0.08  0.21  0.09  0.08  0.09  0.06  0.09  0.06
x11 276    2.99  1.00  0.07  0.09  0.01  0.08  0.08  0.29  0.05  0.03  0.04
x12 276    2.99  1.01  0.07  0.10  0.07  0.08  0.10  0.11  0.07  0.11  0.22
  x10 x11 x12
x1  0.06  0.07  0.07
x2  0.08  0.09  0.10
x3  0.21  0.01  0.07
x4  0.09  0.08  0.08
x5  0.08  0.08  0.10
x6  0.09  0.29  0.11
x7  0.06  0.05  0.07
x8  0.09  0.03  0.11
x9  0.06  0.04  0.22
x10 1.00  0.17  0.22
x11 0.17  1.00  0.24
x12 0.22  0.24  1.00
>
>

```

```

> #共分散構造分析の実行
> #lavaanパッケージの読み込み
> library(lavaan)
>
> # モデルの設定
> model.1 <- '
+   # latent variable definitions
+   f1 =~ x1 + x2 + x3
+   f2 =~ x4 + x5 + x6
+   f3 =~ x7 + x8 + x9
+   f4 =~ x10 + x11 + x12
+
+   # regression
+   f3 =~ f1
+   f4 =~ f1 + f2
+
+   # variances and covariances
+   f1 =~ f2
+
>

```

	A	B	C	D	E	F	G	H	I	J	K	L
1	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12
2	4	3	2	3	4	4	2	4	2	4	4	3
3	2	3	2	5	3	4	5	3	3	4	4	4
4	1	4	2	2	3	3	4	3	2	3	4	4
5	1	5	2	2	3	2	3	2	2	5	4	3
6	2	3	1	4	2	1	3	3	2	1	1	1
7	2	4	3	3	5	4	4	3	4	2	4	1
8	4	4	4	4	4	3	2	1	2	2	2	1
9	3	4	3	3	3	4	3	4	3	4	4	3
10	3	4	3	2	3	5	4	4	3	2	5	4
11	3	5	2	2	4	4	4	4	5	4	1	3
12	4	2	3	4	4	5	3	4	2	5	3	2
13	4	4	2	3	4	2	2	4	4	3	3	4
14	3	4	3	2	3	3	4	4	4	4	2	5
15	4	3	5	4	2	3	3	2	2	4	3	3
16	1	4	3	2	3	2	4	1	3	2	3	2
17	4	4	3	2	3	4	4	4	3	3	3	3
18	3	2	2	4	1	4	4	4	4	3	2	2
19	4	4	4	4	4	3	3	4	2	2	1	3
20	3	3	3	3	3	4	2	3	4	4	3	3
21	4	2	5	2	2	2	4	2	4	4	3	5

```
> # lavaan関数を使って計算
> lav.model.1 <- lavaan(model.1, data=d1, model.type="sem",
+                        fixed.x=F, meanstructure=F,
+                        auto.var=TRUE, auto.fix.first=TRUE)
> summary(lav.model.1, fit.measures=TRUE, standardized=TRUE)
```

lavaan 0.6-12 ended normally after 58 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	28
Number of observations	276

Model Test User Model:

Test statistic	83.328
Degrees of freedom	50
P-value (Chi-square)	0.002

Model Test Baseline Model:

Test statistic	276.593
Degrees of freedom	66
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	0.842
Tucker-Lewis Index (TLI)	0.791

Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-4630.239
Loglikelihood unrestricted model (H1)	-4588.575
Akaike (AIC)	9316.478
Bayesian (BIC)	9417.849
Sample-size adjusted Bayesian (BIC)	9329.066

Root Mean Square Error of Approximation:

RMSEA	0.049
90 Percent confidence interval - lower	0.030
90 Percent confidence interval - upper	0.067
P-value RMSEA <= 0.05	0.508

Standardized Root Mean Square Residual:

SRMR	0.054
------	-------

Parameter Estimates:

Standard errors	Standard
Information	Expected
Information saturated (h1) model	Structured

Latent Variables:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 =~						
x1	1.000				0.438	0.448
x2	1.367	0.359	3.809	0.000	0.599	0.580
x3	0.873	0.248	3.523	0.000	0.382	0.385
f2 =~						
x4	1.000				0.418	0.409
x5	0.931	0.287	3.243	0.001	0.389	0.389
x6	1.442	0.440	3.277	0.001	0.603	0.579
f3 =~						
x7	1.000				0.576	0.583
x8	0.691	0.202	3.421	0.001	0.398	0.394

x9	0.763	0.217	3.522	0.000	0.439	0.435
f4 ~						
x10	1.000				0.408	0.398
x11	1.193	0.343	3.474	0.001	0.487	0.490
x12	1.195	0.344	3.470	0.001	0.488	0.485
Regressions:						
	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f3 ~						
f1	0.739	0.224	3.298	0.001	0.561	0.561
f4 ~						
f1	0.280	0.141	1.988	0.047	0.300	0.300
f2	0.486	0.191	2.543	0.011	0.498	0.498
Covariances:						
	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 ~						
f2	0.037	0.025	1.525	0.127	0.204	0.204
Variances:						
	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
.x1	0.765	0.084	9.100	0.000	0.765	0.800
.x2	0.707	0.108	6.528	0.000	0.707	0.663
.x3	0.839	0.084	9.953	0.000	0.839	0.852
.x4	0.868	0.094	9.281	0.000	0.868	0.832
.x5	0.852	0.089	9.597	0.000	0.852	0.849
.x6	0.719	0.124	5.794	0.000	0.719	0.664
.x7	0.645	0.107	6.032	0.000	0.645	0.660
.x8	0.863	0.089	9.684	0.000	0.863	0.845
.x9	0.829	0.091	9.066	0.000	0.829	0.811
.x10	0.884	0.092	9.622	0.000	0.884	0.841
.x11	0.752	0.093	8.126	0.000	0.752	0.760
.x12	0.773	0.094	8.210	0.000	0.773	0.764
f1	0.192	0.072	2.655	0.008	1.000	1.000
f2	0.175	0.076	2.313	0.021	1.000	1.000
.f3	0.227	0.093	2.435	0.015	0.685	0.685
.f4	0.100	0.050	1.997	0.046	0.601	0.601

## 成長曲線モデル — lavaanパッケージ

パッケージの読み込み  
library(lavaan)

あらかじめlavaanパッケージをインストールしておく必要がある。

## モデルの設定

```
モデル名 <- '
# 測定変数の指定 (= ~ を使う)
切片潜在変数名 =~ 1*観測変数1 + 1*観測変数2 + 1*観測変数3 + ...
傾き潜在変数名 =~ 0*観測変数1 + 1*観測変数2 + 2*観測変数3 + ...

# 回帰式 (~ を使う)
切片潜在変数名 ~ 説明変数1 + 説明変数2 + ...
傾き潜在変数名 ~ 説明変数1 + 説明変数2 + ...
観測変数名1 ~ 説明変数1 + 説明変数2 + ...

# 分散, 共分散 (~~ を使う)
変数名i ~~ 変数名i      # 分散
変数名j ~~ 変数名k      # 共分散
,
```

モデル部分はシングルカンマ (') でくくる。

すべての変数の分散を推定する場合は分散の指定は省略できる。

## パラメタ値の推定

lavaanオブジェクト名 <- growth(モデル名, data=データ名)    または

lavaanオブジェクト名 <- lavaan(モデル名, data=データ名, model.type="growth",  
fixed.x=F, int.lv.free=TRUE, auto.var=TRUE)

fixed.x=FALSE    : 外生変数となる観測変数の分散, 共分散, 平均を標本平均で固定する  
int.lv.free = TRUE : 潜在変数の切片を推定する  
auto.var=TRUE    : すべての変数の分散または残差分散を推定する

summary(lavaanオブジェクト名, fit.measures=TRUE)

結果の解釈には非標準化解を用いるので standardized=TRUE はつけていない。  
切片変数と傾き変数の相関を求めるために標準化解を求めることはある。

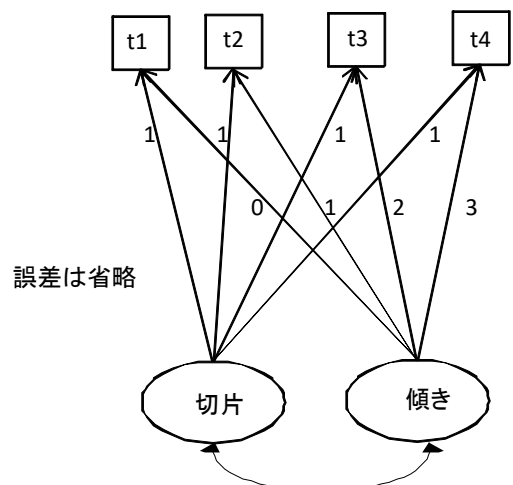
## モデル部分のスキプトの例

```
model.1 <- '
# latent variable definitions
f.inter =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
f.slope =~ 0*t1 + 1*t2 + 2*t3 + 3*t4

# regression

# variances and covariances
f.inter ~~ f.slope
,
```

# すべての分散を推定するので, 分散の式は省略している



```
> setwd("d:¥¥")
> d1 <- read.table("成長曲線モデル_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id t1 t2 t3 t4
1  1  3  6  8 12
2  2  8 12 17 22
3  3  8 11 13 16
4  4  7 13 20 25
5  5  7 13 20 25
6  6  7 14 20 26
>
```

```
> # id 列の削除
> d1 <- d1[,c(-1)]
>
```

```
> #データフレームの行数（標本の大きさ）、平均、標準偏差、共分散、相関
```

```
> n.d1 <- nrow(d1)
> mean.d1 <- apply(d1, 2, mean)
> sd.d1 <- apply(d1, 2, sd)
> cor.d1 <- cor(d1)
> round(data.frame(n.d1, mean.d1, sd.d1, cor.d1), 2)
  n.d1 mean.d1 sd.d1  t1  t2  t3  t4
t1   60    5.62  1.53 1.00 0.74 0.57 0.50
t2   60   10.10  2.20 0.74 1.00 0.93 0.91
t3   60   14.73  3.44 0.57 0.93 1.00 0.98
t4   60   19.15  4.68 0.50 0.91 0.98 1.00
```

	A	B	C	D	E
1	id	t1	t2	t3	t4
2	1	3	6	8	12
3	2	8	12	17	22
4	3	8	11	13	16
5	4	7	13	20	25
6	5	7	13	20	25
7	6	7	14	20	26
8	7	7	13	18	22
9	8	8	12	16	20
10	9	6	13	20	28
11	10	7	13	21	27
12	11	7	12	19	25
13	12	7	13	18	25
14	13	6	11	17	22
15	14	6	11	16	22
16	15	6	10	14	19
17	16	6	11	14	19
18	17	6	11	14	19
19	18	6	9	13	15
20	19	7	10	13	16
21	20	6	8	11	12

```
> #lavaanパッケージの読み込み
> library(lavaan)
>
```

```
> # モデルの設定
```

```
> model.1 <- '
+ # latent variable definitions
+ f.inter =~ 1*t1 + 1*t2 + 1*t3 + 1*t4
+ f.slope =~ 0*t1 + 1*t2 + 2*t3 + 3*t4
+
+ # regression
+
+ # variances and covariances
+ , f.inter =~ f.slope
+ '
```

```
> # growth関数を使って計算
```

```
> gro.model.1 <- growth(model.1, data=d1)
>
> summary(gro.model.1, fit.measures=TRUE)
```

lavaan 0.6-12 ended normally after 43 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	9
Number of observations	60

Model Test User Model:

Test statistic	2.609
Degrees of freedom	5
P-value (Chi-square)	0.760

Model Test Baseline Model:

Test statistic	390.102
Degrees of freedom	6
P-value	0.000

User Model versus Baseline Model:

Comparative Fit Index (CFI)	1.000
Tucker-Lewis Index (TLI)	1.007

## Loglikelihood and Information Criteria:

Loglikelihood user model (H0)	-384.343
Loglikelihood unrestricted model (H1)	-383.039
Akaike (AIC)	786.687
Bayesian (BIC)	805.536
Sample-size adjusted Bayesian (BIC)	777.228

## Root Mean Square Error of Approximation:

RMSEA	0.000
90 Percent confidence interval - lower	0.000
90 Percent confidence interval - upper	0.124
P-value RMSEA $\leq$ 0.05	0.813

## Standardized Root Mean Square Residual:

SRMR	0.017
------	-------

## Parameter Estimates:

Standard errors	Standard
Information	Expected
Information saturated (h1) model	Structured

## Latent Variables:

	Estimate	Std. Err	z-value	P(> z )
f. inter = ~				
t1	1.000			
t2	1.000			
t3	1.000			
t4	1.000			
f. slope = ~				
t1	0.000			
t2	1.000			
t3	2.000			
t4	3.000			

## Covariances:

	Estimate	Std. Err	z-value	P(> z )
f. inter ~~				
f. slope	0.482	0.261	1.843	0.065

## Intercepts:

	Estimate	Std. Err	z-value	P(> z )
.t1	0.000			
.t2	0.000			
.t3	0.000			
.t4	0.000			
f. inter	5.610	0.185	30.268	0.000
f. slope	4.515	0.177	25.470	0.000

## Variances:

	Estimate	Std. Err	z-value	P(> z )
.t1	0.355	0.141	2.512	0.012
.t2	0.164	0.060	2.707	0.007
.t3	0.360	0.096	3.743	0.000
.t4	0.044	0.187	0.236	0.813
f. inter	1.886	0.378	4.992	0.000
f. slope	1.858	0.346	5.374	0.000

## # lavaan関数を使って計算. 同一の結果

```
lav.model.1 <- lavaan(model.1, data=d1, model.type="growth",
                      fixed.x=F, int.lv.free=TRUE, auto.var=TRUE)
summary(lav.model.1, fit.measures=TRUE)
```

## 多母集団同時分析 — semパッケージ

```
library(sem)
opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
                                "SRMR", "AIC", "AICc", "BIC", "CAIC"))
```

## 各集団のモデルの設定

```
モデル名1 <- specifyEquations()
  構造方程式
```

```
モデル名2 <- specifyEquations()
  構造方程式
```

```
...
```

## 群分け変数

```
群分け変数名 <- factor(群分け変数名)
```

## 多母集団モデルの設定

```
mgオブジェクト名 <- multigroupModel(モデル名1, モデル名2, ..., groups=levels(群分け変数名))
```

## 非標準化解の推定

```
semオブジェクト名 <- sem(mgオブジェクト名, S=list(共分散行列1, 共分散行列2, ...),
                           N=c(標本サイズ1, 標本サイズ2, ...))
```

もしくは

```
semオブジェクト名 <- sem(mgオブジェクト名, data=データ行列, group="群分け変数",
                           formula=~変数1 + 変数2 + ...)
```

## 結果の出力

```
summary(semオブジェクト名)
```

## 標準化解の推定

```
stdCoef(semオブジェクト名)
```

群分け変数名に「group」を用いると不具合が生じることがあるので他の名前を使う

群分け変数のfactor化で、levels=c(水準1, 水準2, ...), labels=c(ラベル1, ラベル2, ...)を設定してもよい  
各群のモデルで、同じパラメタ名にすると等値制約になる

群分け変数名はfactor関数を使ってfactor化しておかなければならない

等値制約を加える場合は、例えば、b12.1 と b12.2 をともに b12というパラメタ名にすると、両群に共通な b12 というパラメタ値を推定する

## 【重要】 共分散行列は、分析に用いる観測変数だけのものにすること

余計な変数も含む共分散行列を用いると、解がおかしくなることがある

formula=~で、共分散行列を求める変数を指定する。必要なすべての変数を「+」でつなぎ、右辺の「~」の後ろに書く。

あらかじめsemパッケージをインストールしておく必要がある。

適合度指標は、デフォルトではAICくらいしか出力されない。必要なものを出力するように指定する。

パス係数の頭文字は b にしておくのが無難。他の文字を用いると、それだけで不具合になる場合がある。

空白行があると、モデルの設定が終了したとみなされる。

(逆に、行をあげないと、モデル設定が終了したことにならない)

## モデル部分のスキプトの例

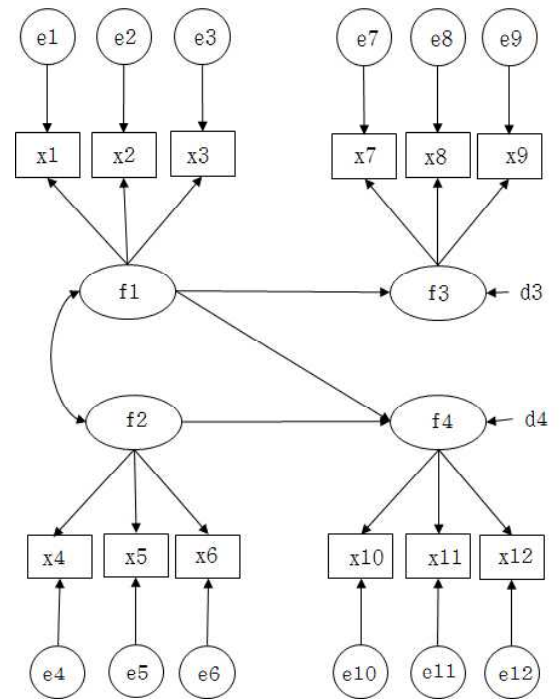
```
seq.1 <- specifyEquations()
x1 = 1*f1
x2 = b2.1*f1
x3 = b3.1*f1
x4 = 1*f2
x5 = b5.1*f2
x6 = b6.1*f2
x7 = 1*f3
x8 = b8.1*f3
```



```

x9 = b9.1*f3
x10 = 1*f4
x11 = b11.1*f4
x12 = b12.1*f4
f3 = a31.1*f1
f4 = a41.1*f1 + a42.1*f2
V(x1) = ev1.1
V(x2) = ev2.1
V(x3) = ev3.1
V(x4) = ev4.1
V(x5) = ev5.1
V(x6) = ev6.1
V(x7) = ev7.1
V(x8) = ev8.1
V(x9) = ev9.1
V(x10) = ev10.1
V(x11) = ev11.1
V(x12) = ev12.1
V(f3) = evf3.1
V(f4) = evf4.1
V(f1) = vf1.1
V(f2) = vf2.1
C(f1, f2) = cf12.1

```



```

seq.2 <- specifyEquations()
x1 = 1*f1
x2 = b2.2*f1
x3 = b3.2*f1
x4 = 1*f2
x5 = b5.2*f2
x6 = b6.2*f2
x7 = 1*f3
x8 = b8.2*f3
x9 = b9.2*f3
x10 = 1*f4
x11 = b11.2*f4
x12 = b12.2*f4
f3 = a31.2*f1
f4 = a41.2*f1 + a42.2*f2
V(x1) = ev1.2
V(x2) = ev2.2
V(x3) = ev3.2
V(x4) = ev4.2
V(x5) = ev5.2
V(x6) = ev6.2
V(x7) = ev7.2
V(x8) = ev8.2
V(x9) = ev9.2
V(x10) = ev10.2
V(x11) = ev11.2
V(x12) = ev12.2
V(f3) = evf3.2
V(f4) = evf4.2
V(f1) = vf1.2
V(f2) = vf2.2
C(f1, f2) = cf12.2

```

```

> setwd("d:¥¥")
> d0 <- read.table("多母集団分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d0)
  id x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 era
1  1  4  3  2  3  4  4  2  4  2  4  4  3 2010
2  2  2  3  2  5  3  4  5  3  3  4  4  4 2010
3  3  1  4  2  2  3  3  4  3  2  3  4  4 2000
4  4  1  5  2  2  3  2  3  2  2  5  4  3 2000
5  5  2  3  1  4  2  1  3  3  2  1  1  1 2000
6  6  2  4  3  3  5  4  4  3  4  2  4  1 2010
>

>
> # 変数名
> vn.items <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10", "x11", "x12")
>
>
> # 共分散行列を計算するデータ
> d1 <- d0[, vn.items]
>

> # 標本サイズ, 平均, 標準偏差, 共分散, 相関
> n.d1 <- nrow(d1)
> mean.d1 <- apply(d1, 2, mean)
> sd.d1 <- apply(d1, 2, sd)
> cov.d1 <- cov(d1)
> cor.d1 <- cor(d1)
> round(data.frame(n.d1, mean.d1, sd.d1, cov.d1, cor.d1), 2)
)
  n.d1 mean.d1 sd.d1  x1  x2  x3  x4  x5  x6  x7  x8  x9  x10  x11  x12
x1  276    3.01  0.98  0.96  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  276    3.01  1.03  0.28  1.07  0.22 -0.01  0.03  0.04  0.28  0.06  0.09  0.08  0.09  0.11
x3  276    3.00  0.99  0.19  0.22  0.99 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  276    2.99  1.02  0.27 -0.01 -0.01  1.05  0.20  0.24  0.03  0.04  0.02  0.09  0.08  0.09
x5  276    3.00  1.00 -0.01  0.03  0.03  0.20  1.01  0.22  0.05  0.23  0.02  0.08  0.08  0.11
x6  276    3.03  1.04  0.03  0.04  0.01  0.24  0.22  1.09  0.09  0.05  0.07  0.09  0.30  0.12
x7  276    3.03  0.99  0.11  0.28  0.12  0.03  0.05  0.09  0.98  0.21  0.23  0.07  0.05  0.07
x8  276    3.01  1.01  0.07  0.06  0.08  0.04  0.23  0.05  0.21  1.03  0.24  0.09  0.03  0.11
x9  276    3.00  1.01  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.03  0.06  0.04  0.22
x10 276    2.99  1.03  0.06  0.08  0.21  0.09  0.08  0.09  0.07  0.09  0.06  1.05  0.17  0.23
x11 276    2.99  1.00  0.07  0.09  0.01  0.08  0.08  0.30  0.05  0.03  0.04  0.17  0.99  0.24
x12 276    2.99  1.01  0.07  0.11  0.07  0.09  0.11  0.12  0.07  0.11  0.22  0.23  0.24  1.01

  x1.1 x2.1 x3.1 x4.1 x5.1 x6.1 x7.1 x8.1 x9.1 x10.1 x11.1 x12.1
x1  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09  0.08  0.09  0.10
x3  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02  0.09  0.08  0.08
x5 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02  0.08  0.08  0.10
x6  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07  0.09  0.29  0.11
x7  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23  0.06  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.04  0.21  1.00  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.00  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.06  0.09  0.06  1.00  0.17  0.22
x11 0.07  0.09  0.01  0.08  0.08  0.29  0.05  0.03  0.04  0.17  1.00  0.24
x12 0.07  0.10  0.07  0.08  0.10  0.11  0.07  0.11  0.22  0.22  0.24  1.00
>
>
> # 群ごとのデータフレーム作成と記述統計量
> ds1 <- d1[d0$era==2000, vn.items]
> n.ds1 <- nrow(ds1)
> mean.ds1 <- apply(ds1, 2, mean)
> sd.ds1 <- apply(ds1, 2, sd)
> cov.ds1 <- cov(ds1)
> cor.ds1 <- cor(ds1)
> round(data.frame(n.ds1, mean.ds1, sd.ds1, cov.ds1, cor.ds1), 2)
  n.ds1 mean.ds1 sd.ds1  x1  x2  x3  x4  x5  x6  x7  x8  x9  x10  x11
x1    140    3.02  0.97  0.94  0.34  0.22  0.23 -0.01  0.08  0.12  0.06  0.16  0.01  0.15
x2    140    2.99  1.12  0.34  1.26  0.30 -0.01  0.10  0.02  0.26  0.05  0.20  0.02  0.24
x3    140    2.96  1.01  0.22  0.30  1.02 -0.08  0.10  0.22  0.16  0.16  0.16  0.15  0.14

```

```

x4    140    3.04    1.02    0.23   -0.01   -0.08    1.04    0.28    0.26    0.05    0.08    0.10    0.03   -0.01
x5    140    2.91    1.03   -0.01    0.10    0.10    0.28    1.06    0.36    0.11    0.26    0.08    0.11    0.08
x6    140    3.00    1.04    0.08    0.02    0.22    0.26    0.36    1.08    0.01    0.09    0.04    0.21    0.32
x7    140    2.98    0.93    0.12    0.26    0.16    0.05    0.11    0.01    0.86    0.27    0.27   -0.02    0.10
x8    140    2.94    1.01    0.06    0.05    0.16    0.08    0.26    0.09    0.27    1.02    0.26    0.06    0.12
x9    140    2.98    1.03    0.16    0.20    0.16    0.10    0.08    0.04    0.27    0.26    1.06    0.06    0.14
x10   140    3.05    1.08    0.01    0.02    0.15    0.03    0.11    0.21   -0.02    0.06    0.06    1.16    0.17
x11   140    3.01    1.02    0.15    0.24    0.14   -0.01    0.08    0.32    0.10    0.12    0.14    0.17    1.04
x12   140    2.96    1.00    0.04    0.13    0.13    0.09    0.21    0.17    0.06    0.15    0.21    0.25    0.33

```

```

      x12  x1.1  x2.1  x3.1  x4.1  x5.1  x6.1  x7.1  x8.1  x9.1  x10.1  x11.1  x12.1
x1  0.04  1.00  0.31  0.23  0.23 -0.01  0.08  0.14  0.06  0.16  0.01  0.15  0.04
x2  0.13  0.31  1.00  0.27 -0.01  0.09  0.02  0.25  0.04  0.17  0.02  0.21  0.12
x3  0.13  0.23  0.27  1.00 -0.08  0.10  0.21  0.18  0.15  0.16  0.14  0.13  0.13
x4  0.09  0.23 -0.01 -0.08  1.00  0.26  0.24  0.05  0.08  0.10  0.03 -0.01  0.09
x5  0.21 -0.01  0.09  0.10  0.26  1.00  0.34  0.12  0.25  0.08  0.10  0.08  0.21
x6  0.17  0.08  0.02  0.21  0.24  0.34  1.00  0.01  0.09  0.03  0.19  0.31  0.17
x7  0.06  0.14  0.25  0.18  0.05  0.12  0.01  1.00  0.29  0.29 -0.02  0.11  0.06
x8  0.15  0.06  0.04  0.15  0.08  0.25  0.09  0.29  1.00  0.25  0.06  0.12  0.15
x9  0.21  0.16  0.17  0.16  0.10  0.08  0.03  0.29  0.25  1.00  0.05  0.13  0.20
x10 0.25  0.01  0.02  0.14  0.03  0.10  0.19 -0.02  0.06  0.05  1.00  0.15  0.24
x11 0.33  0.15  0.21  0.13 -0.01  0.08  0.31  0.11  0.12  0.13  0.15  1.00  0.32
x12 1.00  0.04  0.12  0.13  0.09  0.21  0.17  0.06  0.15  0.20  0.24  0.32  1.00
>

```

```

> ds2 <- d1[d0$era==2010,vn.items]
> n.ds2 <- nrow(ds2)
> mean.ds2 <- apply(ds2, 2, mean)
> sd.ds2 <- apply(ds2, 2, sd)
> cov.ds2 <- cov(ds2)
> cor.ds2 <- cor(ds2)
> round(data.frame(n.ds2, mean.ds2, sd.ds2, cov.ds2, cor.ds2), 2)
      n.ds2 mean.ds2 sd.ds2
x1    136    3.01  0.99
x2    136    3.04  0.94
x3    136    3.04  0.98
x4    136    2.93  1.03
x5    136    3.09  0.97
x6    136    3.05  1.05
x7    136    3.08  1.05
x8    136    3.09  1.01
x9    136    3.02  1.00
x10   136    2.93  0.98
x11   136    2.97  0.97
x12   136    3.01  1.02
      x1  x2  x3  x4  x5  x6  x7  x8  x9  x10  x11
x1  0.99 0.22 0.15 0.30 -0.02 -0.02 0.09 0.07 0.00 0.10 -0.01
x2  0.22 0.88 0.14 0.00 -0.06 0.06 0.31 0.06 -0.02 0.14 -0.06
x3  0.15 0.14 0.96 0.07 -0.05 -0.19 0.06 -0.01 0.01 0.28 -0.12
x4  0.30 0.00 0.07 1.05 0.14 0.23 0.01 0.01 -0.06 0.14 0.17
x5 -0.02 -0.06 -0.05 0.14 0.94 0.08 -0.03 0.19 -0.05 0.06 0.08
x6 -0.02 0.06 -0.19 0.23 0.08 1.10 0.17 0.00 0.10 -0.02 0.28
x7 0.06 0.06 -0.06 0.01 -0.03 0.17 1.11 0.15 0.19 0.16 -0.01
x8 0.07 0.06 -0.01 0.01 0.19 0.00 0.15 1.03 0.23 0.14 -0.06
x9 0.00 -0.02 0.01 -0.06 -0.05 0.10 0.19 0.23 1.00 0.06 -0.05
x10 0.10 0.14 0.28 0.14 0.06 -0.02 0.16 0.14 0.06 0.95 0.18
x11 -0.01 -0.06 -0.12 0.17 0.08 0.28 -0.01 -0.06 -0.05 0.18 0.95
x12 0.09 0.01 0.08 -0.01 0.06 0.09 0.07 0.24 0.22 0.14

```

```

      x12  x1.1  x2.1  x3.1  x4.1  x5.1  x6.1  x7.1  x8.1  x9.1  x10.1  x11.1  x12.1
x1  0.10  1.00  0.24  0.15  0.30 -0.02 -0.02  0.09  0.07  0.00  0.11 -0.01  0.10
x2  0.09  0.24  1.00  0.15 -0.01 -0.06  0.06  0.31  0.07 -0.02  0.16 -0.06  0.09
x3  0.01  0.15  0.15  1.00  0.07 -0.05 -0.19  0.06 -0.01  0.01  0.30 -0.13  0.01
x4  0.08  0.30 -0.01  0.07  1.00  0.14  0.22  0.00  0.01 -0.06  0.14  0.17  0.08
x5 -0.01 -0.02 -0.06 -0.05  0.14  1.00  0.08 -0.03  0.20 -0.05  0.06  0.09 -0.01
x6  0.06 -0.02  0.06 -0.19  0.22  0.08  1.00  0.16  0.00  0.10 -0.02  0.27  0.05
x7  0.09  0.09  0.31  0.06  0.00 -0.03  0.16  1.00  0.14  0.18  0.16  0.00  0.08
x8  0.07  0.07  0.07 -0.01  0.01  0.20  0.00  0.14  1.00  0.22  0.14 -0.06  0.07
x9  0.24  0.00 -0.02  0.01 -0.06 -0.05  0.10  0.18  0.22  1.00  0.06 -0.05  0.23
x10 0.22  0.11  0.16  0.30  0.14  0.06 -0.02  0.16  0.14  0.06  1.00  0.19  0.22
x11 0.14 -0.01 -0.06 -0.13  0.17  0.09  0.27  0.00 -0.06 -0.05  0.19  1.00  0.14
x12 1.04  0.10  0.09  0.01  0.08 -0.01  0.05  0.08  0.07  0.23  0.22  0.14  1.00
>
>

```

```

> # 適合度指標の設定
> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
+                                "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>

```

```

> # 多母集団分析モデル
> # specifyEquations を使う方法

```

```

> seq.1 <- specifyEquations()
1: x1 = 1*f1
2: x2 = b2.1*f1
3: x3 = b3.1*f1
4: x4 = 1*f2
5: x5 = b5.1*f2
6: x6 = b6.1*f2
7: x7 = 1*f3
8: x8 = b8.1*f3
9: x9 = b9.1*f3
10: x10 = 1*f4
11: x11 = b11.1*f4
12: x12 = b12.1*f4
13: f3 = a31.1*f1
14: f4 = a41.1*f1 + a42.1*f2
15: V(x1) = ev1.1
16: V(x2) = ev2.1
17: V(x3) = ev3.1
18: V(x4) = ev4.1
19: V(x5) = ev5.1
20: V(x6) = ev6.1
21: V(x7) = ev7.1
22: V(x8) = ev8.1
23: V(x9) = ev9.1
24: V(x10) = ev10.1
25: V(x11) = ev11.1
26: V(x12) = ev12.1
27: V(f3) = evf3.1
28: V(f4) = evf4.1
29: V(f1) = vf1.1
30: V(f2) = vf2.1
31: C(f1, f2) = cf12.1
32:
Read 31 items

```

```

> seq.2 <- specifyEquations()
1: x1 = 1*f1
2: x2 = b2.2*f1
3: x3 = b3.2*f1
4: x4 = 1*f2
5: x5 = b5.2*f2
6: x6 = b6.2*f2
7: x7 = 1*f3
8: x8 = b8.2*f3
9: x9 = b9.2*f3
10: x10 = 1*f4
11: x11 = b11.2*f4
12: x12 = b12.2*f4
13: f3 = a31.2*f1
14: f4 = a41.2*f1 + a42.2*f2
15: V(x1) = ev1.2
16: V(x2) = ev2.2
17: V(x3) = ev3.2
18: V(x4) = ev4.2
19: V(x5) = ev5.2
20: V(x6) = ev6.2
21: V(x7) = ev7.2
22: V(x8) = ev8.2
23: V(x9) = ev9.2
24: V(x10) = ev10.2
25: V(x11) = ev11.2
26: V(x12) = ev12.2
27: V(f3) = evf3.2
28: V(f4) = evf4.2
29: V(f1) = vf1.2
30: V(f2) = vf2.2
31: C(f1, f2) = cf12.2
32:
Read 31 items
>

```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	era
2	1	4	3	2	3	4	4	2	4	2	4	4	3	2010
3	2	2	3	2	5	3	4	5	3	3	4	4	4	2010
4	3	1	4	2	2	3	3	4	3	2	3	4	4	2000
5	4	1	5	2	2	3	2	3	2	2	5	4	3	2000
6	5	2	3	1	4	2	1	3	3	2	1	1	1	2000
7	6	2	4	3	3	5	4	4	3	4	2	4	1	2010
8	7	4	4	4	4	4	3	2	1	2	2	2	1	2000
9	8	3	4	3	3	3	4	3	4	3	4	4	3	2000
10	9	3	4	3	2	3	5	4	4	3	2	5	4	2010
11	10	3	5	2	2	4	4	4	4	5	4	1	3	2010
12	11	4	2	3	4	4	5	3	4	2	5	3	2	2000
13	12	4	4	2	3	4	2	2	4	4	3	3	4	2010
14	13	3	4	3	2	3	3	4	4	4	4	2	5	2010
15	14	4	3	5	4	2	3	3	2	2	4	3	3	2010
16	15	1	4	3	2	3	2	4	1	3	2	3	2	2010
17	16	4	4	3	2	3	4	4	4	3	3	3	3	2000
18	17	3	2	2	4	1	4	4	4	4	3	2	2	2000
19	18	4	4	4	4	4	3	3	4	2	2	1	3	2010
20	19	3	3	3	3	3	4	2	3	4	4	3	3	2010
21	20	4	2	5	2	2	2	4	2	4	4	3	5	2010

```

> # 群分け変数
> d0$era <- factor(d0$era)

> # 多母集団モデルの設定
> mg.seq <- multigroupModel(seq.1, seq.2, groups=levels(d0$era))

> # 非標準化解の推定
> sem.mg.seq.1 <- sem(mg.seq, S=list(cov.ds1, cov.ds2), N=c(nrow(ds1), nrow(ds2)))
> summary(sem.mg.seq.1)
>
> sem.mg.seq.2 <- sem(mg.seq, data=d0, group="era",
+                      formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)
> summary(sem.mg.seq.2)

```

```

Model Chisquare = 139.2735 Df = 100 Pr(>Chisq) = 0.005784765
Chisquare (null model) = 351.4344 Df = 132
Goodness-of-fit index = 0.9238904
Adjusted goodness-of-fit index = 0.8904022
RMSEA index = 0.05354138 90% CI: (0.0297816, 0.07371039)
Bentler-Bonett NFI = 0.6036998
Tucker-Lewis NNFI = 0.7637515
Bentler CFI = 0.8210239
Bentler RNI = 0.8210239
Bollen IFI = 0.8438021
SRMR = 0.07013975
AIC = 251.2735
AICc = 168.4242
BIC = -422.7666

```

Iterations: initial fits, 55 60 final fit, 0

era: 2000

```

Model Chisquare = 55.62624 Df = 50 Pr(>Chisq) = 0.271284
Goodness-of-fit index = 0.9396586
Adjusted goodness-of-fit index = 0.9058675
RMSEA index = 0.02845226 90% CI: (NA, 0.06355037)
Bentler-Bonett NFI = 0.7238251
Tucker-Lewis NNFI = 0.9451572
Bentler CFI = 0.9584524
Bentler RNI = 0.9584524
Bollen IFI = 0.9628427
SRMR = 0.05942191
AIC = 111.6262
AICc = 70.25687
BIC = -191.4559
CAIC = -241.4559

```

Normalized Residuals

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-1.576431	-0.339524	0.003613	0.046905	0.524147	2.489451

R-square for Endogenous Variables

	x1	x2	x3	x4	x5	x6	f3	x7	x8	x9	f4	x10	x11	x12
	0.2201	0.3168	0.2498	0.1351	0.2667	0.4718	0.3518	0.3523	0.2093	0.2718	0.4097	0.1134	0.3386	0.3100

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
b2.1	1.3872113	0.43174005	3.213071	1.313239e-03	x2 <--- f1
b3.1	1.1086159	0.35482899	3.124367	1.781881e-03	x3 <--- f1
b5.1	1.4190272	0.49898398	2.843833	4.457439e-03	x5 <--- f2
b6.1	1.9016015	0.68950021	2.757942	5.816655e-03	x6 <--- f2
b8.1	0.8407922	0.27068295	3.106188	1.895159e-03	x8 <--- f3
b9.1	0.9763751	0.30276095	3.224904	1.260147e-03	x9 <--- f3
b11.1	1.6412882	0.62840894	2.611816	9.006283e-03	x11 <--- f4
b12.1	1.5367848	0.58957726	2.606588	9.144942e-03	x12 <--- f4
a31.1	0.7152017	0.26434392	2.705573	6.818672e-03	f3 <--- f1
a41.1	0.2758235	0.16536167	1.668002	9.531542e-02	f4 <--- f1

```

a42.1 0.4334890 0.23600565 1.836774 6.624332e-02 f4 <--- f2
ev1.1 0.7346959 0.11009121 6.673520 2.497397e-11 x1 <--> x1
ev2.1 0.8600468 0.15384179 5.590463 2.264655e-08 x2 <--> x2
ev3.1 0.7649704 0.12022942 6.362589 1.983808e-10 x3 <--> x3
ev4.1 0.9010753 0.12073763 7.463086 8.451929e-14 x4 <--> x4
ev5.1 0.7797266 0.12933962 6.028521 1.654667e-09 x5 <--> x5
ev6.1 0.5699717 0.16769611 3.398837 6.767306e-04 x6 <--> x6
ev7.1 0.5542339 0.11191377 4.952330 7.333020e-07 x7 <--> x7
ev8.1 0.8052118 0.12007488 6.705914 2.001501e-11 x8 <--> x8
ev9.1 0.7697465 0.12864153 5.983655 2.181852e-09 x9 <--> x9
ev10.1 1.0246569 0.13516758 7.580641 3.438515e-14 x10 <--> x10
ev11.1 0.6899581 0.13409140 5.145432 2.669065e-07 x11 <--> x11
ev12.1 0.6890995 0.12477337 5.522809 3.336229e-08 x12 <--> x12
evf3.1 0.1953882 0.09465758 2.064159 3.900268e-02 f3 <--> f3
evf4.1 0.0773900 0.05460812 1.417188 1.564278e-01 f4 <--> f4
vfl.1 0.2072876 0.09671749 2.143227 3.209485e-02 f1 <--> f1
vf2.1 0.1408055 0.08265240 1.703586 8.845847e-02 f2 <--> f2
cfl2.1 0.0480028 0.03075306 1.560911 1.185447e-01 f2 <--> f1

```

era: 2010

```

Model Chisquare = 83.64727 Df = 50 Pr(>Chisq) = 0.002004207
Goodness-of-fit index = 0.9076585
Adjusted goodness-of-fit index = 0.8559472
RMSEA index = 0.07060298 90% CI: (0.04271215, 0.09650504)
Bentler-Bonett NFI = 0.4424171
Tucker-Lewis NNFI = 0.4713682
Bentler CFI = 0.5995214
Bentler RNI = 0.5995214
Bollen IFI = 0.6635867
SRMR = 0.08117283
AIC = 139.6473
AICc = 98.82484
BIC = -161.9855
CAIC = -211.9855

```

#### Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-2.417689	-0.622212	-0.015302	0.005705	0.347314	2.970949

#### R-square for Endogenous Variables

x1	x2	x3	x4	x5	x6	f3	x7	x8	x9	f4	x10	x11
0.1830	0.2500	0.1217	0.8147	0.0245	0.0576	0.3041	0.3805	0.0861	0.0885	0.2979	0.5350	0.0562
x12												
0.1042												

#### Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )
b2.2	1.10458250	0.44416964	2.4868483	1.288804e-02
b3.2	0.80544968	0.36631935	2.1987637	2.789473e-02
b5.2	0.16365829	0.21023534	0.7784528	4.363021e-01
b6.2	0.27162739	0.32803269	0.8280498	4.076423e-01
b8.2	0.45773220	0.27596081	1.6586855	9.717918e-02
b9.2	0.45725176	0.27424816	1.6672920	9.545636e-02
b11.2	0.32345609	0.20141897	1.6058869	1.082988e-01
b12.2	0.46072051	0.25194333	1.8286672	6.744948e-02
a31.2	0.84473328	0.38813133	2.1764110	2.952454e-02
a41.2	0.83749152	0.38897437	2.1530764	3.131268e-02
a42.2	0.10568853	0.16340612	0.6467844	5.177715e-01
ev1.2	0.80481061	0.12513131	6.4317285	1.261610e-10
ev2.2	0.66011094	0.11874083	5.5592582	2.709237e-08
ev3.2	0.84402006	0.11842218	7.1272128	1.024217e-12
ev4.2	0.19545490	0.99937157	0.1955778	8.449406e-01
ev5.2	0.91728688	0.11512440	7.9677886	1.615389e-15
ev6.2	1.03762725	0.14654947	7.0803892	1.437502e-12
ev7.2	0.68878863	0.25168301	2.7367308	6.205305e-03
ev8.2	0.94053817	0.12905876	7.2876740	3.153519e-13
ev9.2	0.91104001	0.12557288	7.2550699	4.014546e-13
ev10.2	0.44229097	0.26328133	1.6799177	9.297333e-02
ev11.2	0.89403858	0.11448992	7.8088849	5.769616e-15
ev12.2	0.92880835	0.12787023	7.2636795	3.767001e-13
evf3.2	0.29446801	0.23557310	1.2500070	2.112970e-01

```

evf4.2 0.35724476 0.26130566 1.3671528 1.715774e-01 f4 <--> f4
vf1.2 0.18031993 0.10413285 1.7316334 8.333885e-02 f1 <--> f1
vf2.2 0.85939278 1.00702339 0.8533990 3.934381e-01 f2 <--> f2
cf12.2 0.08774834 0.06012634 1.4593993 1.444552e-01 f2 <--> f1
>
>
>
> # 標準化解の推定
> stdCoef(sem.mg.seq.1)
>
> stdCoef(sem.mg.seq.2)

```

```

Group: 2000
      Std. Estimate
1          0.4690995 x1 <--- f1
2    b2.1    0.5628938 x2 <--- f1
3    b3.1    0.4998325 x3 <--- f1
4          0.3676214 x4 <--- f2
5    b5.1    0.5163940 x5 <--- f2
6    b6.1    0.6868961 x6 <--- f2
7          0.5935213 x7 <--- f3
8    b8.1    0.4574431 x8 <--- f3
9    b9.1    0.5213693 x9 <--- f3
10         0.3367950 x10 <--- f4
11  b11.1    0.5818586 x11 <--- f4
12  b12.1    0.5567891 x12 <--- f4
13  a31.1    0.5931030 f3 <--- f1
14  a41.1    0.3468321 f4 <--- f1
15  a42.1    0.4492512 f4 <--- f2
16  ev1.1    0.7799457 x1 <--> x1
17  ev2.1    0.6831506 x2 <--> x2
18  ev3.1    0.7501674 x3 <--> x3
19  ev4.1    0.8648545 x4 <--> x4
20  ev5.1    0.7333373 x5 <--> x5
21  ev6.1    0.5281738 x6 <--> x6
22  ev7.1    0.6477324 x7 <--> x7
23  ev8.1    0.7907458 x8 <--> x8
24  ev9.1    0.7281740 x9 <--> x9
25  ev10.1   0.8865691 x10 <--> x10
26  ev11.1   0.6614405 x11 <--> x11
27  ev12.1   0.6899858 x12 <--> x12
28  evf3.1   0.6482289 f3 <--> f3
29  evf4.1   0.5903202 f4 <--> f4
30  vf1.1    1.0000000 f1 <--> f1
31  vf2.1    1.0000000 f2 <--> f2
32  cf12.1   0.2809766 f2 <--> f1

```

```

Group: 2010
      Std. Estimate
1          0.4278337 x1 <--- f1
2    b2.2    0.4999760 x2 <--- f1
3    b3.2    0.3488976 x3 <--- f1
4          0.9026117 x4 <--- f2
5    b5.2    0.1564586 x5 <--- f2
6    b6.2    0.2399766 x6 <--- f2
7          0.6168840 x7 <--- f3
8    b8.2    0.2934977 x8 <--- f3
9    b9.2    0.2975115 x9 <--- f3
10         0.7314304 x10 <--- f4
11  b11.2    0.2370681 x11 <--- f4
12  b12.2    0.3227618 x12 <--- f4
13  a31.2    0.5514417 f3 <--- f1
14  a41.2    0.4985470 f4 <--- f1
15  a42.2    0.1373496 f4 <--- f2
16  ev1.2    0.8169583 x1 <--> x1
17  ev2.2    0.7500240 x2 <--> x2
18  ev3.2    0.8782704 x3 <--> x3
19  ev4.2    0.1852921 x4 <--> x4
20  ev5.2    0.9755207 x5 <--> x5
21  ev6.2    0.9424112 x6 <--> x6

```

```
22 ev7.2      0.6194542    x7 <--> x7
23 ev8.2      0.9138591    x8 <--> x8
24 ev9.2      0.9114869    x9 <--> x9
25 ev10.2     0.4650095    x10 <--> x10
26 ev11.2     0.9437987    x11 <--> x11
27 ev12.2     0.8958248    x12 <--> x12
28 evf3.2     0.6959121     f3 <--> f3
29 evf4.2     0.7020589     f4 <--> f4
30 vf1.2      1.0000000     f1 <--> f1
31 vf2.2      1.0000000     f2 <--> f2
32 cf12.2     0.2229057     f2 <--> f1
>
```



## 多母集団同時分析 — lavaanパッケージ

パッケージの読み込み  
library(lavaan)

あらかじめlavaanパッケージをインストールしておく必要がある。  
lavaanパッケージをインストールするために、mnormtパッケージをインストールしておく。

## モデルの設定

```
モデル名 <- '
# 測定変数の指定 (= ~ を使う)
潜在変数名1 ~ 観測変数1 + 観測変数2 + ...
潜在変数名2 ~ 観測変数1 + 観測変数2 + ...

# 回帰式 (~ を使う)
従属変数名1 ~ 説明変数1 + 説明変数2 + ...
従属変数名2 ~ 説明変数1 + 説明変数2 + ...

# 分散, 共分散 (~ ~ を使う)
変数名i ~ ~ 変数名i      # 分散
変数名j ~ ~ 変数名k      # 共分散
,
```

モデル部分はシングルカンマ (') でくくる。  
すべての変数の分散を推定する場合は分散の指定は省略できる。

## パラメタ値の推定

```
lavaanオブジェクト名 <- lavaan(モデル名, data=データ名, model.type="sem", group="群分け変数名",
fixed.x=FALSE, meanstructure=FALSE, auto.var=TRUE, auto.fix.first=TRUE)
```

group="群分け変数名" : 母集団を分ける変数の名前を指定する  
fixed.x=FALSE : 外生変数となる観測変数の分散, 共分散, 平均を標本平均で固定する  
meanstructure=False : 平均構造の検討をしない  
auto.var=TRUE : すべての変数の分散または残差分散を推定する  
auto.fix.first=TRUE : 非標準化解において測定変数のうち最初の1つのパス係数を1に固定する

```
summary(lavaanオブジェクト名, fit.measures=TRUE, standardized=TRUE)
```

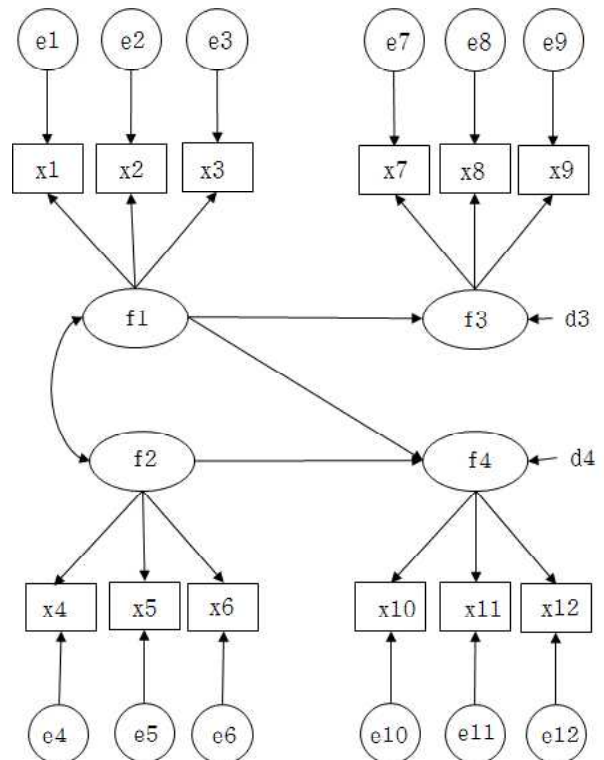
## モデル部分のスキプトの例

```
model.1 <- '
# latent variable definitions
f1 =~ x1 + x2 + x3
f2 =~ x4 + x5 + x6
f3 =~ x7 + x8 + x9
f4 =~ x10 + x11 + x12

# regression
f3 ~ f1
f4 ~ f1 + f2

# variances and covariances
f1 ~ ~ f2
,
```

# すべての分散を推定するので, 分散の式は省略している



```

> setwd("d:¥¥¥")
> d1 <- read.table("多母集団分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> # id 列の削除
> d1 <- d1[,c(-1)]
>
> #データフレームの行数（標本の大きさ）、平均、標準偏差、共分散、相関
> n.d1 <- nrow(d1)
> mean.d1 <- apply(d1, 2, mean)
> sd.d1 <- apply(d1, 2, sd)
> cov.d1 <- cov(d1)
> cor.d1 <- cor(d1)
> round(data.frame(n.d1, mean.d1, sd.d1, cov.d1, cor.d1),2)
>
> # 群ごとのデータフレーム作成と記述統計量
> ds1 <- d1[d1$era==2000,]
> n.ds1 <- nrow(ds1)
> mean.ds1 <- apply(ds1, 2, mean)
> sd.ds1 <- apply(ds1, 2, sd)
> cov.ds1 <- cov(ds1)
> cor.ds1 <- cor(ds1)
>
> ds2 <- d1[d1$era==2010,]
> n.ds2 <- nrow(ds2)
> mean.ds2 <- apply(ds2, 2, mean)
> sd.ds2 <- apply(ds2, 2, sd)
> cov.ds2 <- cov(ds2)
> cor.ds2 <- cor(ds2)
>
> #lavaanパッケージの読み込み
> library(lavaan)
>
> # モデルの設定
> model.1 <-
+ # latent variable definitions
+ f1 =~ x1 + x2 + x3
+ f2 =~ x4 + x5 + x6
+ f3 =~ x7 + x8 + x9
+ f4 =~ x10 + x11 + x12
+
+ # regression
+ f3 =~ f1
+ f4 =~ f1 + f2
+
+ # variances and covariances
+ , f1 =~ f2
+
>
> # lavaan関数を使って計算
> fit.model.1 <- lavaan(model.1, data=d1, model.type="sem", group="era",
+ fixed.x=F, meanstructure=F,
+ auto.var=TRUE, auto.fix.first=TRUE)
>
> summary(fit.model.1, fit.measures=TRUE, standardized=TRUE)

```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	era
2	1	4	3	2	3	4	4	2	4	2	4	4	3	2010
3	2	2	3	2	5	3	4	5	3	3	4	4	4	2010
4	3	1	4	2	2	3	3	4	3	2	3	4	4	2000
5	4	1	5	2	2	3	2	3	2	2	5	4	3	2000
6	5	2	3	1	4	2	1	3	3	2	1	1	1	2000
7	6	2	4	3	3	5	4	4	3	4	2	4	1	2010
8	7	4	4	4	4	4	3	2	1	2	2	2	1	2000
9	8	3	4	3	3	3	4	3	4	3	4	4	3	2000
10	9	3	4	3	2	3	5	4	4	3	2	5	4	2010
11	10	3	5	2	2	4	4	4	4	5	4	1	3	2010
12	11	4	2	3	4	4	5	3	4	2	5	3	2	2000
13	12	4	4	2	3	4	2	2	4	4	3	3	4	2010
14	13	3	4	3	2	3	3	4	4	4	4	2	5	2010
15	14	4	3	5	4	2	3	3	2	2	4	3	3	2010
16	15	1	4	3	2	3	2	4	1	3	2	3	2	2010
17	16	4	4	3	2	3	4	4	4	3	3	3	3	2000
18	17	3	2	2	4	1	4	4	4	4	3	2	2	2000
19	18	4	4	4	4	4	3	3	4	2	2	1	3	2010
20	19	3	3	3	3	3	4	2	3	4	4	3	3	2010
21	20	4	2	5	2	2	2	4	2	4	4	3	5	2010

lavaan 0.6-12 ended normally after 146 iterations

Estimator	ML
Optimization method	NLMINB
Number of model parameters	56
Number of observations per group:	
2010	136
2000	140

Model Test User Model:

Test statistic	140.293
Degrees of freedom	100

P-value (Chi-square) 0.005  
 Test statistic for each group:  
 2010 84.267  
 2000 56.026

## Model Test Baseline Model:

Test statistic 353.995  
 Degrees of freedom 132  
 P-value 0.000

## User Model versus Baseline Model:

Comparative Fit Index (CFI) 0.818  
 Tucker-Lewis Index (TLI) 0.760

## Loglikelihood and Information Criteria:

Loglikelihood user model (H0) -4611.761  
 Loglikelihood unrestricted model (H1) -4541.615  
 Akaike (AIC) 9335.523  
 Bayesian (BIC) 9538.265  
 Sample-size adjusted Bayesian (BIC) 9360.699

## Root Mean Square Error of Approximation:

RMSEA 0.054  
 90 Percent confidence interval - lower 0.031  
 90 Percent confidence interval - upper 0.074  
 P-value RMSEA  $\leq$  0.05 0.362

## Standardized Root Mean Square Residual:

SRMR 0.070

## Parameter Estimates:

Standard errors Standard  
 Information Expected  
 Information saturated (h1) model Structured

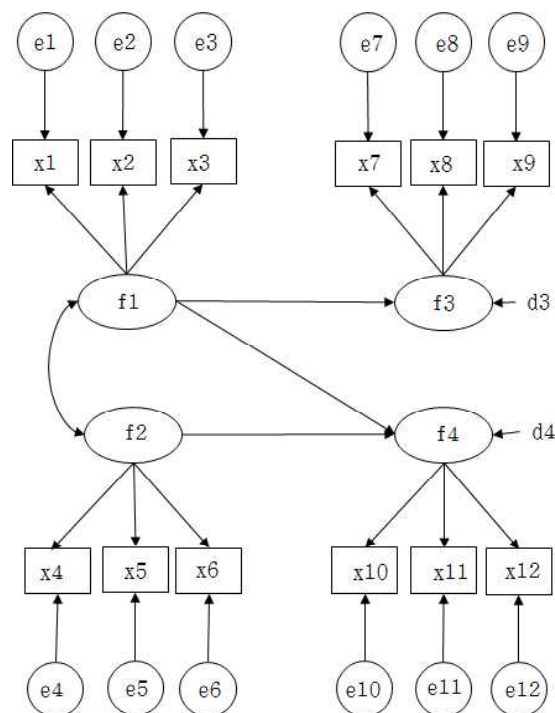
## Group 1 [2010]:

## Latent Variables:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 =						
x1	1.000				0.423	0.428
x2	1.105	0.443	2.496	0.013	0.467	0.500
x3	0.805	0.365	2.207	0.027	0.341	0.349
f2 =						
x4	1.000				0.924	0.903
x5	0.164	0.209	0.781	0.435	0.151	0.156
x6	0.272	0.327	0.831	0.406	0.251	0.240
f3 =						
x7	1.000				0.648	0.617
x8	0.458	0.275	1.665	0.096	0.297	0.293
x9	0.457	0.273	1.673	0.094	0.296	0.298
f4 =						
x10	1.000				0.711	0.731
x11	0.323	0.201	1.612	0.107	0.230	0.237
x12	0.461	0.251	1.835	0.066	0.327	0.323

## Regressions:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f3 ~						
f1	0.845	0.387	2.184	0.029	0.551	0.551
f4 ~						
f1	0.837	0.388	2.161	0.031	0.499	0.499
f2	0.106	0.163	0.649	0.516	0.137	0.137



## Covariances:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 ~~ f2	0.087	0.059	1.465	0.143	0.223	0.223

## Variances:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
.x1	0.799	0.124	6.455	0.000	0.799	0.817
.x2	0.655	0.117	5.580	0.000	0.655	0.750
.x3	0.838	0.117	7.154	0.000	0.838	0.878
.x4	0.194	0.988	0.196	0.844	0.194	0.185
.x5	0.911	0.114	7.997	0.000	0.911	0.976
.x6	1.030	0.145	7.107	0.000	1.030	0.942
.x7	0.684	0.249	2.747	0.006	0.684	0.619
.x8	0.934	0.128	7.315	0.000	0.934	0.914
.x9	0.904	0.124	7.282	0.000	0.904	0.911
.x10	0.439	0.260	1.686	0.092	0.439	0.465
.x11	0.887	0.113	7.838	0.000	0.887	0.944
.x12	0.922	0.126	7.291	0.000	0.922	0.896
f1	0.179	0.103	1.738	0.082	1.000	1.000
f2	0.853	0.996	0.857	0.392	1.000	1.000
.f3	0.292	0.233	1.255	0.210	0.696	0.696
.f4	0.355	0.258	1.372	0.170	0.702	0.702

## Group 2 [2000]:

## Latent Variables:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 =~ x1	1.000				0.454	0.469
x2	1.387	0.430	3.225	0.001	0.629	0.563
x3	1.109	0.354	3.136	0.002	0.503	0.500
f2 =~ x4	1.000				0.374	0.368
x5	1.419	0.497	2.854	0.004	0.531	0.516
x6	1.902	0.687	2.768	0.006	0.711	0.687
f3 =~ x7	1.000				0.547	0.594
x8	0.841	0.270	3.117	0.002	0.460	0.457
x9	0.976	0.302	3.236	0.001	0.534	0.521
f4 =~ x10	1.000				0.361	0.337
x11	1.641	0.626	2.621	0.009	0.592	0.582
x12	1.537	0.587	2.616	0.009	0.554	0.557

## Regressions:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f3 ~ f1	0.715	0.263	2.715	0.007	0.593	0.593
f4 ~ f1	0.276	0.165	1.674	0.094	0.347	0.347
f2	0.433	0.235	1.843	0.065	0.449	0.449

## Covariances:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 ~~ f2	0.048	0.030	1.566	0.117	0.281	0.281

## Variances:

	Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
.x1	0.729	0.109	6.698	0.000	0.729	0.780
.x2	0.854	0.152	5.610	0.000	0.854	0.683
.x3	0.760	0.119	6.386	0.000	0.760	0.750
.x4	0.895	0.119	7.490	0.000	0.895	0.865
.x5	0.774	0.128	6.050	0.000	0.774	0.733
.x6	0.566	0.166	3.411	0.001	0.566	0.528
.x7	0.550	0.111	4.970	0.000	0.550	0.648
.x8	0.799	0.119	6.730	0.000	0.799	0.791
.x9	0.764	0.127	6.005	0.000	0.764	0.728
.x10	1.017	0.134	7.608	0.000	1.017	0.887
.x11	0.685	0.133	5.164	0.000	0.685	0.661

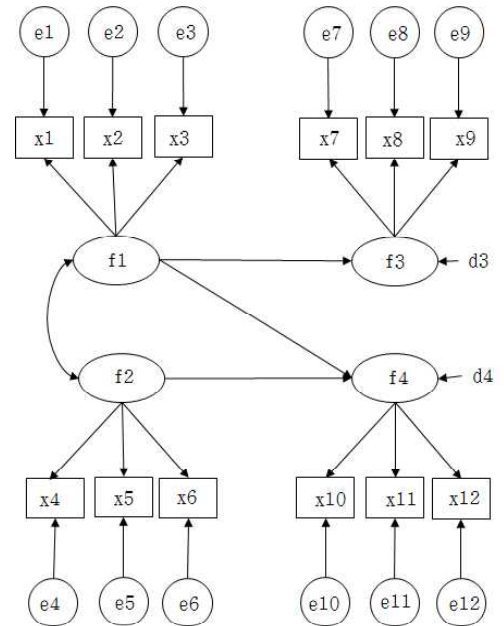
.x12	0.684	0.123	5.543	0.000	0.684	0.690
f1	0.206	0.096	2.151	0.031	1.000	1.000
f2	0.140	0.082	1.710	0.087	1.000	1.000
.f3	0.194	0.094	2.072	0.038	0.648	0.648
.f4	0.077	0.054	1.422	0.155	0.590	0.590

## 等値制約 — semパッケージ

```
seq.1 <- specifyEquations()
x1 = 1*f1
x2 = b2.1*f1
x3 = b3.1*f1
x4 = 1*f2
x5 = b5.1*f2
x6 = b6.1*f2
x7 = 1*f3
x8 = b8.1*f3
x9 = b9.1*f3
x10 = 1*f4
x11 = b11.1*f4
x12 = b12*f4
f3 = a31.1*f1
f4 = a41.1*f1 + a42.1*f2
V(x1) = ev1.1
V(x2) = ev2.1
V(x3) = ev3.1
V(x4) = ev4.1
V(x5) = ev5.1
V(x6) = ev6.1
V(x7) = ev7.1
V(x8) = ev8.1
V(x9) = ev9.1
V(x10) = ev10.1
V(x11) = ev11.1
V(x12) = ev12.1
V(f3) = evf3.1
V(f4) = evf4.1
V(f1) = vf1
V(f2) = vf2.1
C(f1, f2) = cf12.1
```

#b12の値は2つの母集団で共通

#f1の分散の値は2つの母集団で共通



```
seq.2 <- specifyEquations()
x1 = 1*f1
x2 = b2.2*f1
x3 = b3.2*f1
x4 = 1*f2
x5 = b5.2*f2
x6 = b6.2*f2
x7 = 1*f3
x8 = b8.2*f3
x9 = b9.2*f3
x10 = 1*f4
x11 = b11.2*f4
x12 = b12*f4
f3 = a31.2*f1
f4 = a41.2*f1 + a42.2*f2
V(x1) = ev1.2
V(x2) = ev2.2
V(x3) = ev3.2
V(x4) = ev4.2
V(x5) = ev5.2
V(x6) = ev6.2
V(x7) = ev7.2
V(x8) = ev8.2
V(x9) = ev9.2
V(x10) = ev10.2
V(x11) = ev11.2
V(x12) = ev12.2
V(f3) = evf3.2
V(f4) = evf4.2
V(f1) = vf1
V(f2) = vf2.2
C(f1, f2) = cf12.2
```

#b12の値は2つの母集団で共通

#f1の分散の値は2つの母集団で共通

```

> setwd("d:¥¥")
> d0 <- read.table("多母集団分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d0)
  id x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 era
1  1  4  3  2  3  4  4  2  4  2  4  4  3 2010
2  2  2  3  2  5  3  4  5  3  3  4  4  4 2010
3  3  1  4  2  2  3  3  4  3  2  3  4  4 2000
4  4  1  5  2  2  3  2  3  2  2  5  4  3 2000
5  5  2  3  1  4  2  1  3  3  2  1  1  1 2000
6  6  2  4  3  3  5  4  4  3  4  2  4  1 2010
>

>
> # 変数名
> vn.items <- c("x1", "x2", "x3", "x4", "x5", "x6", "x7", "x8", "x9", "x10", "x11", "x12")
>
>
> # 共分散行列を計算するデータ
> d1 <- d0[, vn.items]
>

> # 標本サイズ, 平均, 標準偏差, 共分散, 相関
> n.d1 <- nrow(d1)
> mean.d1 <- apply(d1, 2, mean)
> sd.d1 <- apply(d1, 2, sd)
> cov.d1 <- cov(d1)
> cor.d1 <- cor(d1)
> round(data.frame(n.d1, mean.d1, sd.d1, cov.d1, cor.d1), 2)
)
  n.d1 mean.d1 sd.d1  x1  x2  x3  x4  x5  x6  x7  x8  x9  x10  x11  x12
x1  276    3.01  0.98  0.96  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  276    3.01  1.03  0.28  1.07  0.22 -0.01  0.03  0.04  0.28  0.06  0.09  0.08  0.09  0.11
x3  276    3.00  0.99  0.19  0.22  0.99 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  276    2.99  1.02  0.27 -0.01 -0.01  1.05  0.20  0.24  0.03  0.04  0.02  0.09  0.08  0.09
x5  276    3.00  1.00 -0.01  0.03  0.03  0.20  1.01  0.22  0.05  0.23  0.02  0.08  0.08  0.11
x6  276    3.03  1.04  0.03  0.04  0.01  0.24  0.22  1.09  0.09  0.05  0.07  0.09  0.30  0.12
x7  276    3.03  0.99  0.11  0.28  0.12  0.03  0.05  0.09  0.98  0.21  0.23  0.07  0.05  0.07
x8  276    3.01  1.01  0.07  0.06  0.08  0.04  0.23  0.05  0.21  1.03  0.24  0.09  0.03  0.11
x9  276    3.00  1.01  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.03  0.06  0.04  0.22
x10 276    2.99  1.03  0.06  0.08  0.21  0.09  0.08  0.09  0.07  0.09  0.06  1.05  0.17  0.23
x11 276    2.99  1.00  0.07  0.09  0.01  0.08  0.08  0.30  0.05  0.03  0.04  0.17  0.99  0.24
x12 276    2.99  1.01  0.07  0.11  0.07  0.09  0.11  0.12  0.07  0.11  0.22  0.23  0.24  1.01

  x1.1 x2.1 x3.1 x4.1 x5.1 x6.1 x7.1 x8.1 x9.1 x10.1 x11.1 x12.1
x1  1.00  0.28  0.19  0.27 -0.01  0.03  0.11  0.07  0.08  0.06  0.07  0.07
x2  0.28  1.00  0.22 -0.01  0.02  0.04  0.28  0.06  0.09  0.08  0.09  0.10
x3  0.19  0.22  1.00 -0.01  0.03  0.01  0.12  0.08  0.09  0.21  0.01  0.07
x4  0.27 -0.01 -0.01  1.00  0.20  0.23  0.03  0.04  0.02  0.09  0.08  0.08
x5 -0.01  0.02  0.03  0.20  1.00  0.21  0.05  0.23  0.02  0.08  0.08  0.10
x6  0.03  0.04  0.01  0.23  0.21  1.00  0.09  0.04  0.07  0.09  0.29  0.11
x7  0.11  0.28  0.12  0.03  0.05  0.09  1.00  0.21  0.23  0.06  0.05  0.07
x8  0.07  0.06  0.08  0.04  0.23  0.04  0.21  1.00  0.24  0.09  0.03  0.11
x9  0.08  0.09  0.09  0.02  0.02  0.07  0.23  0.24  1.00  0.06  0.04  0.22
x10 0.06  0.08  0.21  0.09  0.08  0.09  0.06  0.09  0.06  1.00  0.17  0.22
x11 0.07  0.09  0.01  0.08  0.08  0.29  0.05  0.03  0.04  0.17  1.00  0.24
x12 0.07  0.10  0.07  0.08  0.10  0.11  0.07  0.11  0.22  0.22  0.24  1.00
>
>
> # 群ごとのデータフレーム作成と記述統計量
> ds1 <- d1[d0$era==2000, vn.items]
> n.ds1 <- nrow(ds1)
> mean.ds1 <- apply(ds1, 2, mean)
> sd.ds1 <- apply(ds1, 2, sd)
> cov.ds1 <- cov(ds1)
> cor.ds1 <- cor(ds1)
> round(data.frame(n.ds1, mean.ds1, sd.ds1, cov.ds1, cor.ds1), 2)
  n.ds1 mean.ds1 sd.ds1  x1  x2  x3  x4  x5  x6  x7  x8  x9  x10  x11
x1    140    3.02  0.97  0.94  0.34  0.22  0.23 -0.01  0.08  0.12  0.06  0.16  0.01  0.15
x2    140    2.99  1.12  0.34  1.26  0.30 -0.01  0.10  0.02  0.26  0.05  0.20  0.02  0.24
x3    140    2.96  1.01  0.22  0.30  1.02 -0.08  0.10  0.22  0.16  0.16  0.16  0.15  0.14

```

```

x4    140    3.04    1.02    0.23   -0.01   -0.08    1.04    0.28    0.26    0.05    0.08    0.10    0.03   -0.01
x5    140    2.91    1.03   -0.01    0.10    0.10    0.28    1.06    0.36    0.11    0.26    0.08    0.11    0.08
x6    140    3.00    1.04    0.08    0.02    0.22    0.26    0.36    1.08    0.01    0.09    0.04    0.21    0.32
x7    140    2.98    0.93    0.12    0.26    0.16    0.05    0.11    0.01    0.86    0.27    0.27   -0.02    0.10
x8    140    2.94    1.01    0.06    0.05    0.16    0.08    0.26    0.09    0.27    1.02    0.26    0.06    0.12
x9    140    2.98    1.03    0.16    0.20    0.16    0.10    0.08    0.04    0.27    0.26    1.06    0.06    0.14
x10   140    3.05    1.08    0.01    0.02    0.15    0.03    0.11    0.21   -0.02    0.06    0.06    1.16    0.17
x11   140    3.01    1.02    0.15    0.24    0.14   -0.01    0.08    0.32    0.10    0.12    0.14    0.17    1.04
x12   140    2.96    1.00    0.04    0.13    0.13    0.09    0.21    0.17    0.06    0.15    0.21    0.25    0.33

```

```

      x12  x1.1  x2.1  x3.1  x4.1  x5.1  x6.1  x7.1  x8.1  x9.1  x10.1  x11.1  x12.1
x1  0.04  1.00  0.31  0.23  0.23 -0.01  0.08  0.14  0.06  0.16  0.01  0.15  0.04
x2  0.13  0.31  1.00  0.27 -0.01  0.09  0.02  0.25  0.04  0.17  0.02  0.21  0.12
x3  0.13  0.23  0.27  1.00 -0.08  0.10  0.21  0.18  0.15  0.16  0.14  0.13  0.13
x4  0.09  0.23 -0.01 -0.08  1.00  0.26  0.24  0.05  0.08  0.10  0.03 -0.01  0.09
x5  0.21 -0.01  0.09  0.10  0.26  1.00  0.34  0.12  0.25  0.08  0.10  0.08  0.21
x6  0.17  0.08  0.02  0.21  0.24  0.34  1.00  0.01  0.09  0.03  0.19  0.31  0.17
x7  0.06  0.14  0.25  0.18  0.05  0.12  0.01  1.00  0.29  0.29 -0.02  0.11  0.06
x8  0.15  0.06  0.04  0.15  0.08  0.25  0.09  0.29  1.00  0.25  0.06  0.12  0.15
x9  0.21  0.16  0.17  0.16  0.10  0.08  0.03  0.29  0.25  1.00  0.05  0.13  0.20
x10 0.25  0.01  0.02  0.14  0.03  0.10  0.19 -0.02  0.06  0.05  1.00  0.15  0.24
x11 0.33  0.15  0.21  0.13 -0.01  0.08  0.31  0.11  0.12  0.13  0.15  1.00  0.32
x12 1.00  0.04  0.12  0.13  0.09  0.21  0.17  0.06  0.15  0.20  0.24  0.32  1.00
>

```

```

> ds2 <- d1[d0$era==2010,vn.items]
> n.ds2 <- nrow(ds2)
> mean.ds2 <- apply(ds2, 2, mean)
> sd.ds2 <- apply(ds2, 2, sd)
> cov.ds2 <- cov(ds2)
> cor.ds2 <- cor(ds2)
> round(data.frame(n.ds2, mean.ds2, sd.ds2, cov.ds2, cor.ds2), 2)
      n.ds2 mean.ds2 sd.ds2
x1    136    3.01  0.99
x2    136    3.04  0.94
x3    136    3.04  0.98
x4    136    2.93  1.03
x5    136    3.09  0.97
x6    136    3.05  1.05
x7    136    3.08  1.05
x8    136    3.09  1.01
x9    136    3.02  1.00
x10   136    2.93  0.98
x11   136    2.97  0.97
x12   136    3.01  1.02
      x1  x2  x3  x4  x5  x6  x7  x8  x9  x10  x11
x1  0.99 0.22 0.15 0.30 -0.02 -0.02 0.09 0.07 0.00 0.11 -0.01
x2  0.22 0.88 0.14 0.00 -0.06 0.06 0.31 0.06 -0.02 0.14 -0.06
x3  0.15 0.14 0.96 0.07 -0.05 -0.19 0.06 -0.01 0.01 0.28 -0.12
x4  0.30 0.00 0.07 1.05 0.14 0.23 0.01 0.01 -0.06 0.14 0.17
x5 -0.02 -0.06 -0.05 0.14 0.94 0.08 -0.03 0.19 -0.05 0.06 0.08
x6 -0.02 0.06 -0.19 0.23 0.08 1.10 0.17 0.00 0.10 -0.02 0.28
x7 0.10 0.31 0.06 0.01 -0.03 0.17 1.11 0.15 0.19 0.16 -0.01
x8 0.07 0.06 -0.01 0.01 0.19 0.00 0.15 1.03 0.23 0.14 -0.06
x9 0.00 -0.02 0.01 -0.06 -0.05 0.10 0.19 0.23 1.00 0.06 -0.05
x10 0.10 0.14 0.28 0.14 0.06 -0.02 0.16 0.14 0.06 0.95 0.18
x11 -0.01 -0.06 -0.12 0.17 0.08 0.28 -0.01 -0.06 -0.05 0.18 0.95
x12 0.10 0.09 0.01 0.08 -0.01 0.06 0.09 0.07 0.24 0.22 0.14

```

```

      x12  x1.1  x2.1  x3.1  x4.1  x5.1  x6.1  x7.1  x8.1  x9.1  x10.1  x11.1  x12.1
x1  0.10  1.00  0.24  0.15  0.30 -0.02 -0.02  0.09  0.07  0.00  0.11 -0.01  0.10
x2  0.09  0.24  1.00  0.15 -0.01 -0.06  0.06  0.31  0.07 -0.02  0.16 -0.06  0.09
x3  0.01  0.15  0.15  1.00  0.07 -0.05 -0.19  0.06 -0.01  0.01  0.30 -0.13  0.01
x4  0.08  0.30 -0.01  0.07  1.00  0.14  0.22  0.00  0.01 -0.06  0.14  0.17  0.08
x5 -0.01 -0.02 -0.06 -0.05  0.14  1.00  0.08 -0.03  0.20 -0.05  0.06  0.09 -0.01
x6  0.06 -0.02  0.06 -0.19  0.22  0.08  1.00  0.16  0.00  0.10 -0.02  0.27  0.05
x7  0.09  0.09  0.31  0.06  0.00 -0.03  0.16  1.00  0.14  0.18  0.16  0.00  0.08
x8  0.07  0.07  0.07 -0.01  0.01  0.20  0.00  0.14  1.00  0.22  0.14 -0.06  0.07
x9  0.24  0.00 -0.02  0.01 -0.06 -0.05  0.10  0.18  0.22  1.00  0.06 -0.05  0.23
x10 0.22  0.11  0.16  0.30  0.14  0.06 -0.02  0.16  0.14  0.06  1.00  0.19  0.22
x11 0.14 -0.01 -0.06 -0.13  0.17  0.09  0.27  0.00 -0.06 -0.05  0.19  1.00  0.14
x12 1.04  0.10  0.09  0.01  0.08 -0.01  0.05  0.08  0.07  0.23  0.22  0.14  1.00
>
>
>
>
> # 適合度指標の設定
> library(sem)
> opt <- options(fit.indices = c("GFI", "AGFI", "RMSEA", "NFI", "NNFI", "CFI", "RNI", "IFI",
+ "SRMR", "AIC", "AICc", "BIC", "CAIC"))
>

```



> # 等値制約モデルの設定

> seq.1 <- specifyEquations()

1: x1 = 1\*f1

2: x2 = b2.1\*f1

3: x3 = b3.1\*f1

4: x4 = 1\*f2

5: x5 = b5.1\*f2

6: x6 = b6.1\*f2

7: x7 = 1\*f3

8: x8 = b8.1\*f3

9: x9 = b9.1\*f3

10: x10 = 1\*f4

11: x11 = b11.1\*f4

12: x12 = b12\*f4 # b12は共通

13: f3 = a31.1\*f1

14: f4 = a41.1\*f1 + a42.1\*f2

15: V(x1) = ev1.1

16: V(x2) = ev2.1

17: V(x3) = ev3.1

18: V(x4) = ev4.1

19: V(x5) = ev5.1

20: V(x6) = ev6.1

21: V(x7) = ev7.1

22: V(x8) = ev8.1

23: V(x9) = ev9.1

24: V(x10) = ev10.1

25: V(x11) = ev11.1

26: V(x12) = ev12.1

27: V(f3) = evf3.1

28: V(f4) = evf4.1

29: V(f1) = vf1 # vf1は共通

30: V(f2) = vf2.1

31: C(f1, f2) = cf12.1

32:

Read 31 items

> seq.2 <- specifyEquations()

1: x1 = 1\*f1

2: x2 = b2.2\*f1

3: x3 = b3.2\*f1

4: x4 = 1\*f2

5: x5 = b5.2\*f2

6: x6 = b6.2\*f2

7: x7 = 1\*f3

8: x8 = b8.2\*f3

9: x9 = b9.2\*f3

10: x10 = 1\*f4

11: x11 = b11.2\*f4

12: x12 = b12\*f4 # b12は共通

13: f3 = a31.2\*f1

14: f4 = a41.2\*f1 + a42.2\*f2

15: V(x1) = ev1.2

16: V(x2) = ev2.2

17: V(x3) = ev3.2

18: V(x4) = ev4.2

19: V(x5) = ev5.2

20: V(x6) = ev6.2

21: V(x7) = ev7.2

22: V(x8) = ev8.2

23: V(x9) = ev9.2

24: V(x10) = ev10.2

25: V(x11) = ev11.2

26: V(x12) = ev12.2

27: V(f3) = evf3.2

28: V(f4) = evf4.2

29: V(f1) = vf1 # vf1は共通

30: V(f2) = vf2.2

31: C(f1, f2) = cf12.2

32:

Read 31 items

>

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	era
2	1	4	3	2	3	4	4	2	4	2	4	4	3	2010
3	2	2	3	2	5	3	4	5	3	3	4	4	4	2010
4	3	1	4	2	2	3	3	4	3	2	3	4	4	2000
5	4	1	5	2	2	3	2	3	2	2	5	4	3	2000
6	5	2	3	1	4	2	1	3	3	2	1	1	1	2000
7	6	2	4	3	3	5	4	4	3	4	2	4	1	2010
8	7	4	4	4	4	4	3	2	1	2	2	2	1	2000
9	8	3	4	3	3	3	4	3	4	3	4	4	3	2000
10	9	3	4	3	2	3	5	4	4	3	2	5	4	2010
11	10	3	5	2	2	4	4	4	4	5	4	1	3	2010
12	11	4	2	3	4	4	5	3	4	2	5	3	2	2000
13	12	4	4	2	3	4	2	2	4	4	3	3	4	2010
14	13	3	4	3	2	3	3	4	4	4	4	2	5	2010
15	14	4	3	5	4	2	3	3	2	2	4	3	3	2010
16	15	1	4	3	2	3	2	4	1	3	2	3	2	2010
17	16	4	4	3	2	3	4	4	4	3	3	3	3	2000
18	17	3	2	2	4	1	4	4	4	4	3	2	2	2000
19	18	4	4	4	4	4	3	3	4	2	2	1	3	2010
20	19	3	3	3	3	3	4	2	3	4	4	3	3	2010
21	20	4	2	5	2	2	2	4	2	4	4	3	5	2010

```

> # 群分け変数
> d0$era <- factor(d0$era)

> # 多母集団モデルの設定
> mg.seq <- multigroupModel(seq.1, seq.2, groups=levels(d0$era))

> # 非標準化解の推定
> sem.mg.seq.1 <- sem(mg.seq, S=list(cov.ds1, cov.ds2), N=c(nrow(ds1), nrow(ds2)))
> summary(sem.mg.seq.1)
>
> sem.mg.seq.2 <- sem(mg.seq, data=d0, group="era",
+                      formula=~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12)
> summary(sem.mg.seq.2)

```

```

Model Chisquare = 142.1825 Df = 102 Pr(>Chisq) = 0.005291511
Chisquare (null model) = 351.4344 Df = 132
Goodness-of-fit index = 0.923128
Adjusted goodness-of-fit index = 0.8914749
RMSEA index = 0.05362383 90% CI: (0.03018218, 0.07358887)
Bentler-Bonett NFI = 0.5954225
Tucker-Lewis NNFI = 0.7630234
Bentler CFI = 0.8168817
Bentler RNI = 0.8168817
Bollen IFI = 0.8389057
SRMR = 0.07158386
AIC = 250.1825
AICc = 169.0603
BIC = -431.0984

```

Iterations: initial fits, 55 60 final fit, 139

era: 2000

```

Model Chisquare = 56.45178 Df = 50 Pr(>Chisq) = 0.2465492
Goodness-of-fit index = 0.9391849
Adjusted goodness-of-fit index = 0.9051285
RMSEA index = 0.03046825 90% CI: (NA, 0.06473732)
Bentler-Bonett NFI = 0.7197265
Tucker-Lewis NNFI = 0.93711
Bentler CFI = 0.9523561
Bentler RNI = 0.9523561
Bollen IFI = 0.9573905
SRMR = 0.06075693
AIC = 112.4518
AICc = 71.08242
BIC = -190.6303
CAIC = -240.6303

```

Normalized Residuals

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
	-1.56083	-0.25398	0.02940	0.05207	0.43291	2.50861

R-square for Endogenous Variables

	x1	x2	x3	x4	x5	x6	f3	x7	x8	x9	f4	x10	x11
	0.2035	0.3143	0.2536	0.1291	0.2539	0.4952	0.3492	0.3526	0.2094	0.2704	0.4270	0.1615	0.3376
x12													
	0.2497												

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
b2.1	1.44281177	0.38926081	3.706543	2.101079e-04	x2 <--- f1
b3.1	1.16664530	0.32821629	3.554502	3.786963e-04	x3 <--- f1
b5.1	1.41635442	0.50340099	2.813571	4.899457e-03	x5 <--- f2
b6.1	1.99275202	0.73428175	2.713879	6.650040e-03	x6 <--- f2
b8.1	0.84069210	0.27159117	3.095432	1.965262e-03	x8 <--- f3
b9.1	0.97336255	0.30332029	3.209026	1.331857e-03	x9 <--- f3
b11.1	1.35187355	0.40102623	3.371035	7.488628e-04	x11 <--- f4
b12	1.12205994	0.20653943	5.432667	5.551798e-08	x12 <--- f4
a31.1	0.74499427	0.25274994	2.947555	3.202981e-03	f3 <--- f1

```

a41.1 0.35272750 0.18436477 1.913205 5.572188e-02 f4 <--- f1
a42.1 0.55411081 0.26372698 2.101077 3.563417e-02 f4 <--- f2
ev1.1 0.74276258 0.10499522 7.074251 1.502578e-12 x1 <--> x1
ev2.1 0.86175104 0.15439532 5.581458 2.385101e-08 x2 <--> x2
ev3.1 0.76004083 0.12095401 6.283718 3.305707e-10 x3 <--> x3
ev4.1 0.90727147 0.12057573 7.524495 5.292456e-14 x4 <--> x4
ev5.1 0.79322332 0.12803738 6.195248 5.819332e-10 x5 <--> x5
ev6.1 0.54459531 0.17377645 3.133884 1.725088e-03 x6 <--> x6
ev7.1 0.55355130 0.11227618 4.930265 8.211829e-07 x7 <--> x7
ev8.1 0.80477965 0.12019513 6.695609 2.147749e-11 x8 <--> x8
ev9.1 0.77086937 0.12874386 5.987621 2.129330e-09 x9 <--> x9
ev10.1 1.00017156 0.13531666 7.391341 1.453552e-13 x10 <--> x10
ev11.1 0.69061094 0.13451646 5.134026 2.836094e-07 x11 <--> x11
ev12.1 0.72869908 0.11188128 6.513146 7.359298e-11 x12 <--> x12
evf3.1 0.19620436 0.09525853 2.059704 3.942687e-02 f3 <--> f3
evf4.1 0.11037855 0.05673097 1.945649 5.169693e-02 f4 <--> f4
vf1 0.18972488 0.06652922 2.851753 4.347889e-03 f1 <--> f1
vf2.1 0.13455050 0.08042451 1.673004 9.432659e-02 f2 <--> f2
cf12.1 0.04434851 0.02792561 1.588095 1.122650e-01 f2 <--> f1

```

era: 2010

```

Model Chisquare = 85.73067 Df = 50 Pr(>Chisq) = 0.001242832
Goodness-of-fit index = 0.9065989
Adjusted goodness-of-fit index = 0.8542942
RMSEA index = 0.07275598 90% CI: (0.04545513, 0.09841906)
Bentler-Bonett NFI = 0.4285295
Tucker-Lewis NNFI = 0.438636
Bentler CFI = 0.5747243
Bentler RNI = 0.5747243
Bollen IFI = 0.6427564
SRMR = 0.08272922
AIC = 141.7307
AICc = 100.9082
BIC = -159.9021
CAIC = -209.9021

```

Normalized Residuals

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
-2.433016	-0.621893	-0.072683	-0.007113	0.332538	2.836605

R-square for Endogenous Variables

x1	x2	x3	x4	x5	x6	f3	x7	x8	x9	f4	x10	x11
0.1906	0.3249	0.0736	0.4507	0.0404	0.1081	0.3338	0.4514	0.0669	0.0745	0.3129	0.2209	0.0980
x12												
0.2405												

Parameter Estimates

	Estimate	Std Error	z value	Pr(> z )	
b2.2	1.22882736	0.44681848	2.7501713	5.956412e-03	x2 <--- f1
b3.2	0.61084721	0.10857287	5.6261494	1.842768e-08	x3 <--- f1
b5.2	0.28253734	0.49889024	0.5663317	5.711683e-01	x5 <--- f2
b6.2	0.50039605	0.31958163	1.5657848	1.173990e-01	x6 <--- f2
b8.2	0.37012869	0.23522290	1.5735232	1.155977e-01	x8 <--- f3
b9.2	0.38504562	0.36194886	1.0638122	2.874138e-01	x9 <--- f3
b11.2	0.67450823	0.22922686	2.9425357	3.255362e-03	x11 <--- f4
b12	1.12205994	0.20653943	5.4326669	5.551798e-08	x12 <--- f4
a31.2	0.94016351	0.31814536	2.9551382	3.125290e-03	f3 <--- f1
a41.2	0.38605522	0.39713333	0.9721048	3.309984e-01	f4 <--- f1
a42.2	0.22414503	0.24955984	0.8981615	3.690995e-01	f4 <--- f2
ev1.2	0.80592002	0.19924005	4.0449700	5.232982e-05	x1 <--> x1
ev2.2	0.59530461	0.12785409	4.6561250	3.222163e-06	x2 <--> x2
ev3.2	0.89062250	0.13437664	6.6278074	3.407095e-11	x3 <--> x3
ev4.2	0.57953099	0.11708934	4.9494769	7.441322e-07	x4 <--> x4
ev5.2	0.90236193	0.34394060	2.6235982	8.700636e-03	x5 <--> x5
ev6.2	0.98201767	0.11552221	8.5006825	1.884792e-17	x6 <--> x6
ev7.2	0.61049579	0.14855990	4.1094253	3.966450e-05	x7 <--> x7
ev8.2	0.96050019	0.27682076	3.4697549	5.209335e-04	x8 <--> x8
ev9.2	0.92516749	0.12630031	7.3251404	2.386487e-13	x9 <--> x9
ev10.2	0.71972704	0.12176773	5.9106548	3.407504e-09	x10 <--> x10
ev11.2	0.85456365	0.11967505	7.1407002	9.285670e-13	x11 <--> x11
ev12.2	0.81140946	0.11845010	6.8502216	7.373568e-12	x12 <--> x12

```

evf3.2 0.33471136 0.15612118 2.1439203 3.203928e-02 f3 <--> f3
evf4.2 0.14018911 0.26910582 0.5209442 6.024057e-01 f4 <--> f4
vf1    0.18972488 0.06652922 2.8517528 4.347889e-03 f1 <--> f1
vf2.2  0.47545662 0.35331673 1.3456952 1.784008e-01 f2 <--> f2
cf12.2 0.06742791 0.05589901 1.2062451 2.277230e-01 f2 <--> f1
>

```

# > # 標準化解の推定

```
> stdCoef(sem.mg.seq.1)
```

```
>
```

```
> stdCoef(sem.mg.seq.2)
```

```
Group: 2000
```

	Std.	Estimate	
1		0.4510666	x1 <--- f1
2	b2.1	0.5606032	x2 <--- f1
3	b3.1	0.5035817	x3 <--- f1
4		0.3593734	x4 <--- f2
5	b5.1	0.5038711	x5 <--- f2
6	b6.1	0.7037276	x6 <--- f2
7		0.5938131	x7 <--- f3
8	b8.1	0.4575488	x8 <--- f3
9	b9.1	0.5199742	x9 <--- f3
10		0.4018642	x10 <--- f4
11	b11.1	0.5810708	x11 <--- f4
12	b12	0.4997119	x12 <--- f4
13	a31.1	0.5909734	f3 <--- f1
14	a41.1	0.3500562	f4 <--- f1
15	a42.1	0.4631008	f4 <--- f2
16	ev1.1	0.7965389	x1 <--> x1
17	ev2.1	0.6857240	x2 <--> x2
18	ev3.1	0.7464055	x3 <--> x3
19	ev4.1	0.8708508	x4 <--> x4
20	ev5.1	0.7461139	x5 <--> x5
21	ev6.1	0.5047674	x6 <--> x6
22	ev7.1	0.6473860	x7 <--> x7
23	ev8.1	0.7906491	x8 <--> x8
24	ev9.1	0.7296269	x9 <--> x9
25	ev10.1	0.8385052	x10 <--> x10
26	ev11.1	0.6623567	x11 <--> x11
27	ev12.1	0.7502880	x12 <--> x12
28	evf3.1	0.6507504	f3 <--> f3
29	evf4.1	0.5730035	f4 <--> f4
30	vf1	1.0000000	f1 <--> f1
31	vf2.1	1.0000000	f2 <--> f2
32	cf12.1	0.2775711	f2 <--> f1

```
Group: 2010
```

	Std.	Estimate	
1		0.4365258	x1 <--- f1
2	b2.2	0.5699934	x2 <--- f1
3	b3.2	0.2713559	x3 <--- f1
4		0.6713234	x4 <--- f2
5	b5.2	0.2009066	x5 <--- f2
6	b6.2	0.3288227	x6 <--- f2
7		0.6718929	x7 <--- f3
8	b8.2	0.2585861	x8 <--- f3
9	b9.2	0.2729710	x9 <--- f3
10		0.4699609	x10 <--- f4
11	b11.2	0.3130131	x11 <--- f4
12	b12	0.4903575	x12 <--- f4
13	a31.2	0.5777449	f3 <--- f1
14	a41.2	0.3722827	f4 <--- f1
15	a42.2	0.3421730	f4 <--- f2
16	ev1.2	0.8094452	x1 <--> x1
17	ev2.2	0.6751075	x2 <--> x2
18	ev3.2	0.9263660	x3 <--> x3
19	ev4.2	0.5493249	x4 <--> x4
20	ev5.2	0.9596365	x5 <--> x5
21	ev6.2	0.8918756	x6 <--> x6

```
22 ev7.2      0.5485599    x7 <--> x7
23 ev8.2      0.9331332    x8 <--> x8
24 ev9.2      0.9254868    x9 <--> x9
25 ev10.2     0.7791368    x10 <--> x10
26 ev11.2     0.9020228    x11 <--> x11
27 ev12.2     0.7595495    x12 <--> x12
28 evf3.2     0.6662108    f3 <--> f3
29 evf4.2     0.6871266    f4 <--> f4
30   vf1      1.0000000    f1 <--> f1
31   vf2.2    1.0000000    f2 <--> f2
32 cf12.2     0.2245030    f2 <--> f1
>
```

## 等値制約 — lavaanパッケージ

## lavaanパッケージを使う方法の例（2群の多母集団分析）

```
model.1 <- '
# latent variable definitions
f1 =~ x1 + c("b12", "")*x2 + c("b131", "b132")*x3
f2 =~ x4 + equal(c("b12", "")*x5 + c(NA, 0)*x6
f3 =~ x7 + x8 + equal(c("b131", "b132"))*x9
f4 =~ x10 + x11 + c("b412", "b412")*x12
```

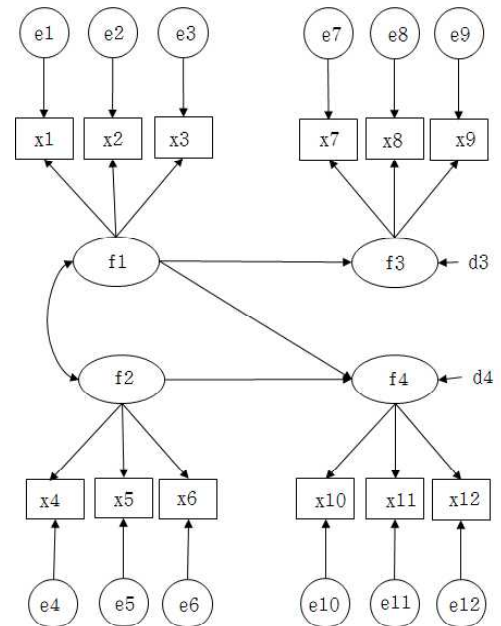
```
# regression
f3 ~ f1
f4 ~ f1 + f2
```

```
# variances and covariances
f1 ~~ c("vf1", "vf1")*f1
f1 ~~ f2
```

b12 : 第1群において,  $x2 \leftarrow f1$  と  $x5 \leftarrow f2$  は同じ値  
 b131 : 第1群において,  $x3 \leftarrow f1$  と  $x9 \leftarrow f3$  は同じ値  
 b132 : 第2群において,  $x3 \leftarrow f1$  と  $x9 \leftarrow f3$  は同じ値  
 b412 : 第1群と第2群の  $x12 \leftarrow f4$  は同じ値  
 c(NA, 0) :  $x6 \leftarrow f2$  のパス係数を, 第1群は推定するが,  
 第2群は0に固定  
 vf1 : 第1群と第2群のf1の分散は同じ値

equal()を使うように説明されているが, c()に同じ系数名を  
 書くだけでも同じ値で推定されるようだ.

semパッケージを使うときも, 係数に同じ名前をつければ,  
 等値制約となる.



```
> setwd("d:¥¥¥")
> d1 <- read.table("多母集団分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> # id 列の削除
> d1 <- d1[,c(-1)]

> #データフレームの行数（標本の大きさ）、平均、標準偏差、共分散、相関
> n.d1 <- nrow(d1)
> mean.d1 <- apply(d1, 2, mean)
> sd.d1 <- apply(d1, 2, sd)
> cov.d1 <- cov(d1)
> cor.d1 <- cor(d1)
> round(data.frame(n.d1, mean.d1, sd.d1, cov.d1, cor.d1), 2)
>

> # 群ごとのデータフレーム作成と記述統計量
> ds1 <- d1[d1$era==2000,]
> n.ds1 <- nrow(ds1)
> mean.ds1 <- apply(ds1, 2, mean)
> sd.ds1 <- apply(ds1, 2, sd)
> cov.ds1 <- cov(ds1)
> cor.ds1 <- cor(ds1)
>

> ds2 <- d1[d1$era==2010,]
> n.ds2 <- nrow(ds2)
> mean.ds2 <- apply(ds2, 2, mean)
> sd.ds2 <- apply(ds2, 2, sd)
> cov.ds2 <- cov(ds2)
> cor.ds2 <- cor(ds2)
```

```

>
> #lavaanパッケージの読み込み
> library(lavaan)

> model.1 <- '
+ # latent variable definitions
+ f1 =~ x1 + c("b12","")*x2 + c("b131","b132")*x3
+ f2 =~ x4 + equal(c("b12","")*x5 + c(NA,0)*x6
+ f3 =~ x7 + x8 + equal(c("b131","b132"))*x9
+ f4 =~ x10 + x11 + c("b412","b412")*x12
+
+ # regression
+ f3 =~ f1
+ f4 =~ f1 + f2
+
+ # variances and covariancess
+ f1 =~ c("vf1","vf1")*f1
+ , f1 =~ f2
+
>
> # lavaan関数を使って計算
> lav.model.1 <- lavaan(model.1, data=d1, model.type="sem", group="era",
+ fixed.x=F, meanstructure=F,
+ auto.var=TRUE, auto.fix.first=TRUE)

> summary(lav.model.1, fit.measures=TRUE, standardized=TRUE)

```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	id	x1	x2	x3	x4	x5	x6	x7	x8	x9	x10	x11	x12	era
2	1	4	3	2	3	4	4	2	4	2	4	4	3	2010
3	2	2	3	2	5	3	4	5	3	3	4	4	4	2010
4	3	1	4	2	2	3	3	4	3	2	3	4	4	2000
5	4	1	5	2	2	3	2	3	2	2	5	4	3	2000
6	5	2	3	1	4	2	1	3	3	2	1	1	1	2000
7	6	2	4	3	3	5	4	4	3	4	2	4	1	2010
8	7	4	4	4	4	4	3	2	1	2	2	2	1	2000
9	8	3	4	3	3	3	4	3	4	3	4	4	3	2000
10	9	3	4	3	2	3	5	4	4	3	2	5	4	2010
11	10	3	5	2	2	4	4	4	4	5	4	1	3	2010
12	11	4	2	3	4	4	5	3	4	2	5	3	2	2000
13	12	4	4	2	3	4	2	2	4	4	3	3	4	2010
14	13	3	4	3	2	3	3	4	4	4	4	2	5	2010
15	14	4	3	5	4	2	3	3	2	2	4	3	3	2010
16	15	1	4	3	2	3	2	4	1	3	2	3	2	2010
17	16	4	4	3	2	3	4	4	4	3	3	3	3	2000
18	17	3	2	2	4	1	4	4	4	4	3	2	2	2000
19	18	4	4	4	4	4	3	3	4	2	2	1	3	2010
20	19	3	3	3	3	3	4	2	3	4	4	3	3	2010
21	20	4	2	5	2	2	2	4	2	4	4	3	5	2010

lavaan 0.6-12 ended normally after 99 iterations

```

Estimator                      ML
Optimization method            NLMINB
Number of model parameters      55
Number of equality constraints    5

```

```

Number of observations per group:
  2010                      136
  2000                      140

```

Model Test User Model:

```

Test statistic                  175.730
Degrees of freedom              106
P-value (Chi-square)           0.000
Test statistic for each group:
  2010                      90.874
  2000                      84.855

```

Model Test Baseline Model:

```

Test statistic                  353.995
Degrees of freedom              132
P-value                        0.000

```

User Model versus Baseline Model:

```

Comparative Fit Index (CFI)    0.686
Tucker-Lewis Index (TLI)       0.609

```

Loglikelihood and Information Criteria:

```

Loglikelihood user model (H0)   -4629.480
Loglikelihood unrestricted model (H1) -4541.615

Akaike (AIC)                   9358.960
Bayesian (BIC)                 9539.980
Sample-size adjusted Bayesian (BIC) 9381.438

```

Root Mean Square Error of Approximation:

```

RMSEA                          0.069
90 Percent confidence interval - lower 0.050

```

90 Percent confidence interval - upper 0.087  
P-value RMSEA <= 0.05 0.047

Standardized Root Mean Square Residual:

SRMR 0.087

Parameter Estimates:

Standard errors  
Information  
Information saturated (h1) model Standard  
Expected  
Structured

Group 1 [2010]:

Latent Variables:

		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 = ~							
x1		1.000				0.478	0.479
x2	(b12)	<b>0.848</b>	<b>0.261</b>	<b>3.253</b>	<b>0.001</b>	<b>0.405</b>	<b>0.441</b>
x3	(b131)	<b>0.552</b>	<b>0.205</b>	<b>2.688</b>	<b>0.007</b>	<b>0.264</b>	<b>0.270</b>
f2 = ~							
x4		1.000				0.423	0.420
x5	(b12)	<b>0.848</b>	<b>0.261</b>	<b>3.253</b>	<b>0.001</b>	<b>0.359</b>	<b>0.365</b>
x6		0.935	0.514	1.818	0.069	0.396	0.378
f3 = ~							
x7		1.000				0.600	0.571
x8		0.537	0.283	1.899	0.058	0.322	0.318
x9	(b131)	<b>0.552</b>	<b>0.205</b>	<b>2.688</b>	<b>0.007</b>	<b>0.331</b>	<b>0.332</b>
f4 = ~							
x10		1.000				0.420	0.440
x11		0.734	0.353	2.081	0.037	0.308	0.318
x12	(b412)	<b>1.241</b>	<b>0.355</b>	<b>3.499</b>	<b>0.000</b>	<b>0.522</b>	<b>0.506</b>

Regressions:

		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f3 ~							
f1		0.720	0.286	2.515	0.012	0.574	0.574
f4 ~							
f1		0.353	0.207	1.705	0.088	0.401	0.401
f2		0.386	0.282	1.372	0.170	0.389	0.389

Covariances:

		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 ~ ~							
f2		0.044	0.046	0.954	0.340	0.217	0.217

Variances:

		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1	(vf1)	<b>0.228</b>	<b>0.069</b>	<b>3.322</b>	<b>0.001</b>	<b>1.000</b>	<b>1.000</b>
.x1		0.766	0.120	6.383	0.000	0.766	0.770
.x2		0.681	0.110	6.181	0.000	0.681	0.806
.x3		0.882	0.115	7.662	0.000	0.882	0.927
.x4		0.835	0.139	6.015	0.000	0.835	0.823
.x5		0.838	0.124	6.771	0.000	0.838	0.867
.x6		0.936	0.156	5.997	0.000	0.936	0.857
.x7		0.742	0.188	3.941	0.000	0.742	0.674
.x8		0.918	0.129	7.099	0.000	0.918	0.899
.x9		0.884	0.121	7.289	0.000	0.884	0.890
.x10		0.736	0.115	6.416	0.000	0.736	0.807
.x11		0.845	0.118	7.181	0.000	0.845	0.899
.x12		0.790	0.144	5.473	0.000	0.790	0.744
f2		0.179	0.105	1.700	0.089	1.000	1.000
.f3		0.241	0.168	1.438	0.150	0.670	0.670
.f4		0.109	0.072	1.527	0.127	0.620	0.620

Group 2 [2000]:

Latent Variables:

		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
--	--	----------	----------	---------	---------	---------	----------



f1 =~							
x1		1.000				0.478	0.490
x2		1.343	0.335	4.009	0.000	0.642	0.573
<b>x3</b>	<b>(b132)</b>	<b>1.014</b>	<b>0.208</b>	<b>4.878</b>	<b>0.000</b>	<b>0.485</b>	<b>0.482</b>
f2 =~							
x4		1.000				0.331	0.325
x5		2.514	2.450	1.026	0.305	0.831	0.809
x6		0.000				0.000	0.000
f3 =~							
x7		1.000				0.538	0.585
x8		0.856	0.255	3.356	0.001	0.461	0.458
<b>x9</b>	<b>(b132)</b>	<b>1.014</b>	<b>0.208</b>	<b>4.878</b>	<b>0.000</b>	<b>0.546</b>	<b>0.532</b>
f4 =~							
x10		1.000				0.448	0.411
x11		1.199	0.408	2.940	0.003	0.538	0.528
<b>x12</b>	<b>(b412)</b>	<b>1.241</b>	<b>0.355</b>	<b>3.499</b>	<b>0.000</b>	<b>0.556</b>	<b>0.566</b>
Regressions:							
		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f3 ~							
f1		0.682	0.193	3.545	0.000	0.606	0.606
f4 ~							
f1		0.378	0.182	2.083	0.037	0.404	0.404
f2		0.311	0.230	1.353	0.176	0.229	0.229
Covariances:							
		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
f1 ~~							
f2		0.036	0.040	0.907	0.364	0.229	0.229
Variances:							
		Estimate	Std. Err	z-value	P(> z )	Std. lv	Std. all
<b>f1</b>	<b>(vf1)</b>	<b>0.228</b>	<b>0.069</b>	<b>3.322</b>	<b>0.001</b>	<b>1.000</b>	<b>1.000</b>
.x1		0.725	0.105	6.916	0.000	0.725	0.760
.x2		0.842	0.152	5.544	0.000	0.842	0.672
.x3		0.775	0.114	6.817	0.000	0.775	0.767
.x4		0.925	0.152	6.076	0.000	0.925	0.894
.x5		0.365	0.663	0.550	0.582	0.365	0.345
.x6		1.071	0.128	8.367	0.000	1.071	1.000
.x7		0.558	0.100	5.582	0.000	0.558	0.658
.x8		0.800	0.118	6.749	0.000	0.800	0.790
.x9		0.757	0.120	6.311	0.000	0.757	0.717
.x10		0.987	0.138	7.130	0.000	0.987	0.831
.x11		0.747	0.134	5.587	0.000	0.747	0.721
.x12		0.656	0.126	5.225	0.000	0.656	0.679
f2		0.109	0.118	0.923	0.356	1.000	1.000
.f3		0.183	0.080	2.302	0.021	0.633	0.633
.f4		0.149	0.078	1.915	0.055	0.742	0.742

## 18 さまざまな多変量データ解析

## クラスター分析

データの標準化 (測定単位に意味がなければデータを標準化しておくことがおすすめされる)  
 (新)データフレーム名 <- `scale(データフレーム名)`  
 特に指定しなければ, 平均=0, 標準偏差=1に標準化する.

### 距離行列の作成

距離行列名 <- `dist((新)データフレーム名, method="距離定義名")`  
`method="euclidean", "maximum", "manhattan", "canberra", "binary" or "minkowski"`

### 階層的クラスター分析

出力名 <- `hclust(距離行列名^2, method="方法名")`  
`method = "ward.D", "centroid" の場合`

出力名 <- `hclust(距離行列名, method="方法名")`  
`method = "ward.D2", "single", "complete", "average", "mcquitty", "median" の場合`

### デンドログラムの表示

縦向き  
`plot(出力名)`

横向き  
`plot(as.dendrogram(出力名), horiz=TRUE)`

あるクラスタ数におけるクラスタリング状況の出力  
`cutree(出力名)`

### 非階層的クラスター分析

`kmeans(データフレーム名, クラスタ数)`

```
> setwd("d:¥¥¥")
> d1 <- read.table("クラスター分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

	item	p.na	b.na	p.correct	b.correct
1	国01	0.028	0.395	0.462	0.337
2	国07	0.019	0.377	0.944	0.251
3	国08	0.143	0.479	0.509	0.305
4	国09	0.090	0.525	0.767	0.301
5	国13	0.206	0.585	0.717	0.275
6	国14	0.237	0.565	0.846	0.283

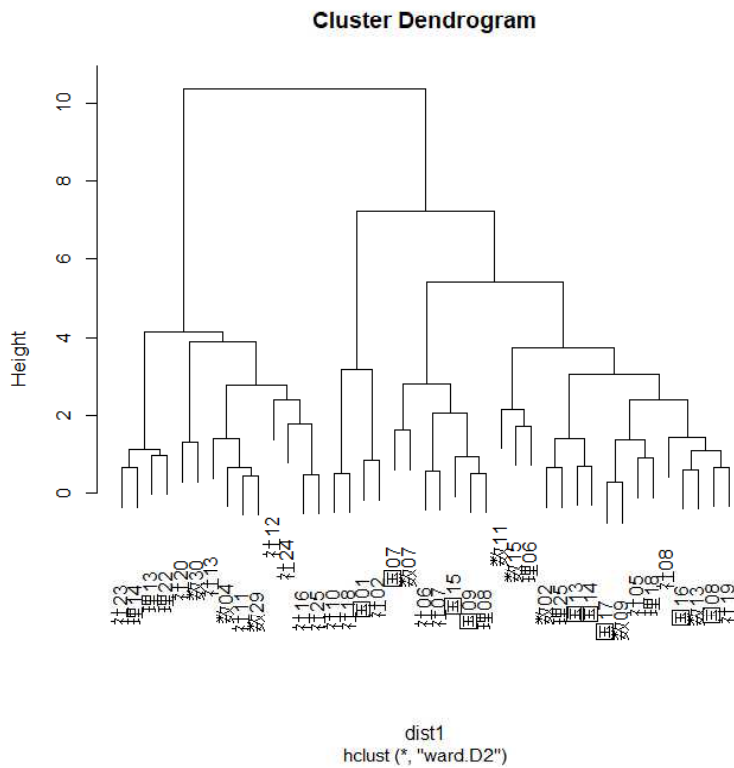
```
>
> # データの標準化
> d2 <- d1[, c("p.na", "b.na", "p.correct", "b.correct")]
> d2 <- scale(d2)
>
> # 距離行列の作成
> rownames(d2) <- d1$item
> dist1 <- dist(d2, method="euclidean")
>
```

	A	B	C	D	E
1	item	p.na	b.na	p.correct	b.correct
2	国01	0.028	0.395	0.462	0.337
3	国07	0.019	0.377	0.944	0.251
4	国08	0.143	0.479	0.509	0.305
5	国09	0.090	0.525	0.767	0.301
6	国13	0.206	0.585	0.717	0.275
7	国14	0.237	0.565	0.846	0.283
8	国15	0.062	0.534	0.805	0.256
9	国16	0.135	0.553	0.544	0.377
10	国17	0.147	0.428	0.701	0.293
11	社02	0.054	0.303	0.454	0.344
12	社05	0.241	0.49	0.658	0.292
13	社06	0.067	0.469	0.554	0.252
14	社07	0.045	0.478	0.611	0.21
15	社08	0.081	0.498	0.384	0.299
16	社10	0.028	0.143	0.475	0.436
17	社11	0.32	0.588	0.469	0.293
18	社12	0.393	0.576	0.549	0.103
19	社13	0.326	0.596	0.632	0.215
20	社16	0.271	0.582	0.416	0.221
21	社18	0.022	0.151	0.49	0.481

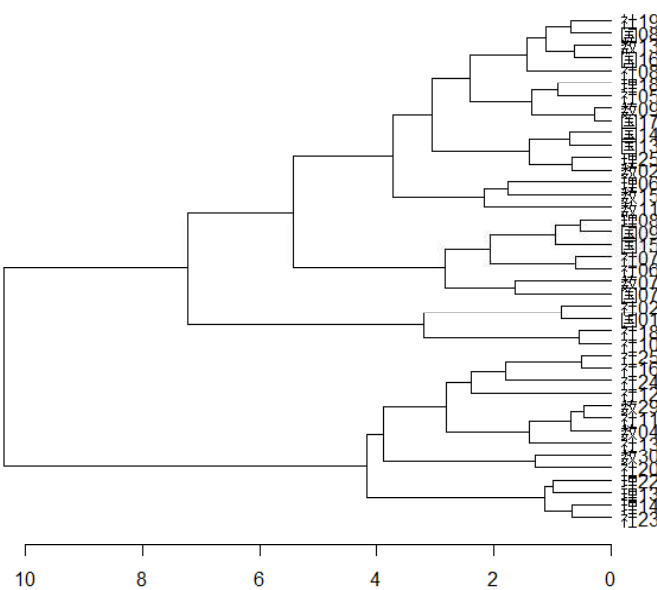
```
> # 階層的クラスター分析
> hclust1 <- hclust(dist1, method="ward.D2")
```

```
# 平方ユークリッド距離・Ward法
```

```
> # 縦向きのデンドログラム
> plot(hclust1)
```



```
> # 横向きのデンドログラム
> plot(as.dendrogram(hclust1), horiz=TRUE)
```



> #あるクラスター数における分類状況

> **cutree**(hclust1, k=8)

国01	国07	国08	国09	国13	
	1	2	3	2	3
国14	国15	国16	国17	社02	
	3	2	3	3	1
社05	社06	社07	社08	社10	
	3	2	2	3	4
社11	社12	社13	社16	社18	
	5	5	5	5	4
社19	社20	社23	社24	社25	
	3	6	7	5	5
数02	数04	数07	数09	数11	
	3	5	2	3	8
数13	数15	数29	数30	理06	
	3	8	5	6	8
理08	理13	理14	理18	理22	
	2	7	7	3	7
理25					
	3				

```
> # 非階層的クラスター分析
```

```
> kmeans1 <- kmeans(d2, 8)
```

```
> kmeans1
```

```
K-means clustering with 8 clusters of sizes 8, 2, 11, 5, 5, 4, 4, 2
```

```
Cluster means:
```

	p.na	b.na	p.correct	b.correct
1	-0.9359166	-0.4090408	1.10945829	-0.52525861
2	-1.3595650	-3.1765849	-0.27831316	1.75893008
3	0.1795248	0.2579431	0.62075262	0.59585997
4	0.1252978	0.5125858	-1.25963056	0.59123084
5	-0.9640269	-0.6739169	-0.32720618	0.01027442
6	1.1384655	0.8465117	-0.07088825	-0.39940401
7	0.7357503	0.8376307	-0.79934479	-1.53064882
8	2.4642359	0.4291043	-1.86610150	-0.57878298

```
Clustering vector:
```

国01	国07	国08	国09		
	5	1	5		1
国13					
	3	3	1	国16	3
国17			3	社06	
	1	5			5
社07			3	社11	
	1	5	2		6
社12				社18	
	7	6	7		2
社19				社24	
	3	8	4		7
社25				数07	
	7	3	6		1
数09				数15	
	1	3	3		3
数29				理08	
	6	8	4		1
理13				理22	
	4	4	3		4
理25					
	3				

```
Within cluster sum of squares by cluster:
```

```
[1] 8.6698755 0.1420172 13.6068424 3.0583587 3.6257138 1.3138106 4.6073177
[8] 0.8415862
(between_SS / total_SS = 77.6 %)
```

```
Available components:
```

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
>
```

```
# 階層的な方法と非階層的な方法で計算方法が異なるので、
# 同じクラス多数でも、必ずしも分類状況は一致しない
```

## 主成分分析

```
library(psych)
```

```
オブジェクト名 <- principal(データ名, nfactors=因子数, rotate="回転方法")
```

```
print(オブジェクト名, sort=TRUE)
```

## スクリープロット

```
VSS.scree(オブジェクト名)
```

あらかじめpsychパッケージをインストールしておく必要がある。

データには、素データ、相関係数行列、共分散行列のいずれかを指定する。

## 回転方法 (rotate)

直交回転: "none", "varimax", "quartimax"

斜交回転: "promax", "oblimin", "simplimax", "cluster"

```
> setwd("d:¥¥")
> d1 <- read.table("因子分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  x1 x2 x3 x4 x5 x6 x7 x8
1  3  2  4  4  5  2  3  4
2  3  2  3  3  3  2  3  2
3  1  3  1  3  2  2  2  2
4  3  4  1  3  4  3  4  5
5  1  3  3  3  4  4  3  4
6  2  1  3  4  2  2  4  3
>
>
```

## # 記述統計量

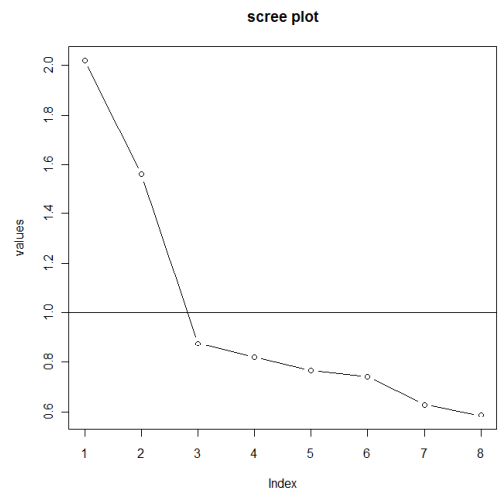
```
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

	N	Mean	SD	x1	x2	x3	x4	x5	x6	x7	x8
x1	346	2.97	1.01	1.00	0.30	0.38	0.27	0.09	0.06	0.08	0.09
x2	346	3.00	1.02	0.30	1.00	0.24	0.17	0.04	0.06	0.01	0.06
x3	346	3.03	1.02	0.38	0.24	1.00	0.19	0.08	0.05	0.03	0.06
x4	346	3.02	1.00	0.27	0.17	0.19	1.00	0.07	0.02	0.05	0.03
x5	346	3.00	1.03	0.09	0.04	0.08	0.07	1.00	0.24	0.32	0.36
x6	346	3.04	1.03	0.06	0.06	0.05	0.02	0.24	1.00	0.19	0.20
x7	346	3.01	1.02	0.08	0.01	0.03	0.05	0.32	0.19	1.00	0.25
x8	346	3.01	1.02	0.09	0.06	0.06	0.03	0.36	0.20	0.25	1.00

```
>
>
```

```
> # psychパッケージの読み込み
> library(psych)
> # スクリープロット
> VSS.scree(d1)
```

	A	B	C	D	E	F	G	H
1	x1	x2	x3	x4	x5	x6	x7	x8
2	3	2	4	4	5	2	3	4
3	3	2	3	3	3	2	3	2
4	1	3	1	3	2	2	2	2
5	3	4	1	3	4	3	4	5
6	1	3	3	3	4	4	3	4
7	2	1	3	4	2	2	4	3
8	2	3	2	3	1	3	2	3
9	5	3	4	5	3	4	2	3
10	3	3	3	4	4	3	3	2
11	4	4	4	3	3	3	3	4
12	4	4	3	2	1	2	3	3
13	3	3	2	3	1	3	1	3
14	2	3	2	3	3	2	2	1
15	3	3	2	5	4	5	4	4
16	3	3	4	2	3	3	3	5
17	4	3	2	4	4	4	3	1
18	3	3	4	4	5	3	5	4
19	3	3	4	3	3	4	1	5
20	5	4	5	3	2	3	3	3
21	3	4	3	3	4	4	4	4



> # 回転を行わない主成分分析

```
> prin.1 <- principal(d1, nfactors=2, rotate="none")
> print(prin.1, sort=TRUE)
```

Principal Components Analysis

Call: principal(r = d1, nfactors = 2, rotate = "none")

Standardized loadings (pattern matrix) based upon correlation matrix

	item	PC1	PC2	h2	u2	com
utsu	5	0.92	-0.06	0.84	0.16	1.0
result	7	0.81	-0.20	0.69	0.31	1.1
stress	2	0.78	0.15	0.63	0.37	1.1
support	4	-0.62	0.28	0.46	0.54	1.4
work	6	0.03	0.64	0.42	0.58	1.0
kyoufu	3	0.46	0.63	0.61	0.39	1.8
id	1	-0.09	0.41	0.18	0.82	1.1

	PC1	PC2
SS loadings	2.70	1.13
Proportion Var	0.39	0.16
Cumulative Var	0.39	0.55
Proportion Explained	0.71	0.29
Cumulative Proportion	0.71	1.00

Mean item complexity = 1.2

Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is 0.13  
with the empirical chi square 182.64 with prob < 2.9e-35

Fit based upon off diagonal values = 0.81

>

> # varimax回転を行う主成分分析

```
> prin.2 <- principal(d1, nfactors=2, rotate="varimax")
> print(prin.2, sort=TRUE)
```

Principal Components Analysis

Call: principal(r = d1, nfactors = 2, rotate = "varimax")

Standardized loadings (pattern matrix) based upon correlation matrix

	item	RC1	RC2	h2	u2	com
utsu	5	0.92	0.04	0.84	0.16	1.0
result	7	0.82	-0.11	0.69	0.31	1.0
stress	2	0.76	0.23	0.63	0.37	1.2
support	4	-0.65	0.21	0.46	0.54	1.2
kyoufu	3	0.39	0.68	0.61	0.39	1.6
work	6	-0.04	0.64	0.42	0.58	1.0
id	1	-0.13	0.40	0.18	0.82	1.2

	RC1	RC2
SS loadings	2.68	1.15
Proportion Var	0.38	0.16
Cumulative Var	0.38	0.55
Proportion Explained	0.70	0.30
Cumulative Proportion	0.70	1.00

Mean item complexity = 1.2

Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is 0.13  
with the empirical chi square 182.64 with prob < 2.9e-35

Fit based upon off diagonal values = 0.81



## 主成分得点を用いた重回帰分析

重回帰分析において、説明変数間の相関が高いときに、無相関ないくつかの主成分を説明変数にして、重回帰分析を行う。説明変数が無相関になるという利点がある。

この方法と、「主成分回帰分析」と言われるものとは、解釈するものが異なる。主成分回帰分析では、いくつかの主成分を採用したとき、結局もとの変数がどう基準変数に影響するかを解釈する。つまり、もとの説明変数に対する回帰係数を解釈する。これに対し、主成分得点を用いた回帰分析では、もとの説明変数ではなく、主成分得点がどう基準変数に影響するかに関心があり、主成分得点に対する回帰係数を解釈する。

具体的な手続きには、説明変数を主成分分析して主成分得点を求め、それを説明変数にして重回帰分析を行う。

## 回帰分析

lm, glm, semのいずれを用いてもよい

## 主成分分析

```
library(psych)
```

```
オブジェクト名 <- principal(データ名, nfactors=主成分数, rotate="none")
```

```
print(オブジェクト名)
```

```
スコア名 <- オブジェクト名$scores
```

あらかじめpsychパッケージをインストールしておく必要がある。  
主成分数は変数数以下にする。

```
> setwd("d:¥¥¥")
> d1 <- read.table("サポートデータ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  support_s support_p kenkou
1         17         13     23
2         16         19     22
3         22         22     29
4         14         17     19
5         18         17     26
6         14         15     22
>
>
> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

	A	B	C
1	support_s	support_p	kenkou
2	17	13	23
3	16	19	22
4	22	22	29
5	14	17	19
6	18	17	26
7	14	15	22
8	15	19	17
9	14	13	20
10	14	14	20
11	17	12	27
12	18	19	22
13	18	13	24
14	9	14	6
15	15	16	23
16	19	18	24

```

      N Mean  SD support_s support_p kenkou
support_s 250 14.99 3.04      1.00      0.38  0.66
support_p 250 14.99 3.03      0.38      1.00  0.02
kenkou     250 20.00 5.01      0.66      0.02  1.00
>
>
> #標準偏回帰係数の推定
> d2 <- as.data.frame(scale(d1[,c("support_s", "support_p", "kenkou")]))
> #共分散行列の確認
> round(cov(d2), 2)
      support_s support_p kenkou
support_s      1.00      0.38  0.66
support_p      0.38      1.00  0.02
kenkou         0.66      0.02  1.00
> result.2 <- lm(kenkou ~ support_s + support_p, data=d2)
> summary(result.2)
```

Call:

```
lm(formula = kenkou ~ support_s + support_p, data = d2)
```

```

Residuals:
    Min       1Q   Median       3Q      Max
-1.87478 -0.43616  0.05371  0.50240  1.79567

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.444e-17  4.467e-02   0.000      1
support_s    7.675e-01  4.835e-02  15.872 < 2e-16 ***
support_p   -2.718e-01  4.835e-02  -5.621 5.11e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7063 on 247 degrees of freedom
Multiple R-squared:  0.5051,    Adjusted R-squared:  0.5011
F-statistic: 126.1 on 2 and 247 DF,  p-value: < 2.2e-16

> confint(result.2)
              2.5 %      97.5 %
(Intercept) -0.08798634  0.08798634
support_s    0.67226113  0.86274015
support_p   -0.36705158 -0.17657255
>
>
>
>
> # 説明変数の抽出
> d3 <- d2[,c("support_s", "support_p")]
>
>
> #主成分分析の実行
> # psychパッケージの読み込み
> library(psych)
> prin.1 <- principal(d3, nfactors=2, rotate="none", scores=TRUE)
> print(prin.1)

Principal Components Analysis
Call: principal(r = d3, nfactors = 2, rotate = "none", scores = TRUE)
Standardized loadings (pattern matrix) based upon correlation matrix
      PC1   PC2 h2    u2 com
support_s 0.83 -0.56 1 2.2e-16 1.7
support_p 0.83  0.56 1 2.2e-16 1.7

      PC1   PC2
SS loadings    1.38 0.62
Proportion Var    0.69 0.31
Cumulative Var    0.69 1.00
Proportion Explained 0.69 0.31
Cumulative Proportion 0.69 1.00

Mean item complexity = 1.7
Test of the hypothesis that 2 components are sufficient.

The root mean square of the residuals (RMSR) is 0
with the empirical chi square 0 with prob < NA

Fit based upon off diagonal values = 1

```

```

#> 主成分得点
> dPC <- prin.1$scores
> head(dPC)
      PC1      PC2
[1,] 0.002829912 -1.182324397
[2,] 0.996054277  0.887391725
[3,] 2.782506917  0.001807549
[4,] 0.201959837  0.886875872
[5,] 0.995707811 -0.294934736
[6,] -0.195260616  0.295454715
>

```

```

> # 主成分得点を説明変数にしたデータの作成
> d4 <- data.frame(dPC, d2[, "kenkou"])
> colnames(d4) <- c("pc1", "pc2", "kenkou")
>
>
> # 記述統計量
> dtmp <- d4
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
      N Mean SD  pc1  pc2 kenkou
pc1  250   0  1 1.00  0.00  0.41
pc2  250   0  1 0.00  1.00 -0.58
kenkou 250   0  1 0.41 -0.58  1.00
>
>
# 主成分得点間の相関は0になっている。

> # 主成分得点を用いた回帰分析
> res3 <- lm(kenkou~pc1+pc2, data=d4)
> summary(res3)

Call:
lm(formula = kenkou ~ pc1 + pc2, data = d4)

Residuals:
    Min       1Q   Median       3Q      Max
-1.87478 -0.43616  0.05371  0.50240  1.79567

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  1.176e-18  4.467e-02   0.000      1
pc1          4.115e-01  4.476e-02   9.193 <2e-16 ***
pc2         -5.795e-01  4.476e-02 -12.946 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7063 on 247 degrees of freedom
Multiple R-squared:  0.5051,    Adjusted R-squared:  0.5011
F-statistic: 126.1 on 2 and 247 DF,  p-value: < 2.2e-16

```

## 主成分回帰分析・PLS回帰分析

重回帰分析において、説明変数の数が多いときに、無相関ないくつかの主成分（またはそのようなもの）に説明変数を集約して重回帰分析を行ったとしたときの、もとの説明変数の影響度を推定する方法。

説明変数が非常に多きときに、重要な説明変数を見つけるのに適した分析法。

説明変数を集約するとき、説明変数の主成分を用いる場合を「主成分回帰分析」、基準変数との共分散を最大にするような成分を合成する場合を「PLS回帰分析」と言う。

主成分回帰分析が、説明変数内だけで、より少数の変数に情報をまとめるのに対し、PLS回帰分析では、集約された変数は基準変数と関連が強いのが望ましいという制約を課している点が異なる。

主成分回帰分析は、主成分得点がどう基準変数に影響するかに関心がある「主成分得点を用いた回帰分析」とは、解釈するものが異なる。

## 主成分回帰分析

```
library(pls)
```

```
オブジェクト名 <- pcr(基準変数名 ~ 変数1, 変数2, ..., data=データ名)
```

```
summary(オブジェクト名)
```

```
オブジェクト名$coefficient
```

あらかじめplsパッケージをインストールしておく必要がある。

主成分数の上限を設定するときは、ncomp=最大主成分数 をオプションに指定する。

## PLS回帰分析

```
library(pls)
```

```
オブジェクト名 <- pls(基準変数名 ~ 変数1, 変数2, ..., data=データ名)
```

```
summary(オブジェクト名)
```

```
オブジェクト名$coefficient
```

あらかじめplsパッケージをインストールしておく必要がある。

主成分数の上限を設定するときは、ncomp=最大主成分数 をオプションに指定する。

```
> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id stress kyoufu support utsu work result
1  1     20    2.2      17    18      0      0
2  2     23    4.8      18    21      1      0
3  3     30    5.8      12    29      1      1
4  4     25    5.2      18    29      0      1
5  5     26    2.0       8    22      1      0
6  6     21    5.0      26    19      1      0
>
>
> # 記述統計量
> dtmp <- d1[,c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```

      N Mean  SD stress kyoufu support  utsu  work result
stress 245 22.94 5.25  1.00  0.40 -0.34  0.62  0.03  0.44
kyoufu 245  4.05 1.17  0.40  1.00 -0.03  0.31  0.10  0.20
support 245 18.42 4.96 -0.34 -0.03  1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49  0.62  0.31 -0.51  1.00  0.02  0.76
work    245  0.50 0.50  0.03  0.10 -0.03  0.02  1.00 -0.08
result  245  0.26 0.44  0.44  0.20 -0.39  0.76 -0.08  1.00
>
```

1	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```

>
>
> #標準偏回帰係数の推定
> d2 <- as.data.frame(scale(d1[,c("stress", "kyoufu", "utsu", "work")]))
> result.2 <- lm(utsu ~ stress + kyoufu + support, data=d2)
> summary(result.2)

Call:
lm(formula = utsu ~ stress + kyoufu + support, data = d2)

Residuals:
    Min       1Q   Median       3Q      Max
-1.69706 -0.51466  0.02535  0.46419  1.81181

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.497e-17  4.571e-02   0.000  1.0000
stress       4.509e-01  5.361e-02   8.411 3.61e-15 ***
kyoufu       1.210e-01  5.048e-02   2.397  0.0173 *
support     -3.508e-01  4.907e-02  -7.150 1.03e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.7155 on 241 degrees of freedom
Multiple R-squared:  0.4943,    Adjusted R-squared:  0.488
F-statistic: 78.52 on 3 and 241 DF,  p-value: < 2.2e-16

> confint(result.2)
                2.5 %      97.5 %
(Intercept) -0.09004947  0.09004947
stress       0.34533426  0.55656020
kyoufu       0.02155634  0.22043194
support     -0.44749766 -0.25417948

> # 主成分回帰分析
> library(pls)
> res3 <- pcr(utsu ~ stress + kyoufu + support, data=d2)
> summary(res3)
Data:   X dimension: 245 3
        Y dimension: 245 1
Fit method: svdpc
Number of components considered: 3
TRAINING: % variance explained
      1 comps  2 comps  3 comps      # その主成分までを用いて何%の分散が説明されるか
X       51.32   83.78   100.00
utsu    45.50   48.92   49.43

> res3$coefficient
, , 1 comps
      utsu
stress  0.3795492
kyoufu  0.2963544
support -0.2522451
, , 2 comps
      utsu
stress  0.3778659
kyoufu  0.1761000
support -0.3960610
, , 3 comps
      utsu
stress  0.4509472
kyoufu  0.1209941
support -0.3508386

```

# 第1主成分のみを採用したときの、もとの変数にかかる係数

# 第2主成分まで採用したときの、もとの変数にかかる係数

# 第3主成分まで採用したときの、もとの変数にかかる係数  
# もともと3変数なので、重回帰分析の結果と一致する

```

>
>
> # PLS 回帰分析
> library(pls)
> res4 <- plsr(utsu ~ stress + kyoufu + support, data=d2)
> summary(res4)
Data:    X dimension: 245 3
        Y dimension: 245 1
Fit method: kernelpls
Number of components considered: 3
TRAINING: % variance explained
      1 comps  2 comps  3 comps
X       50.74   81.91  100.00
utsu    48.75   49.38   49.43
> res4$coefficient
, , 1 comps

            utsu
stress    0.4093243
kyoufu    0.2068089
support  -0.3352484

, , 2 comps

            utsu
stress    0.4283727
kyoufu    0.1324056
support  -0.3703227

, , 3 comps

            utsu
stress    0.4509472
kyoufu    0.1209941
support  -0.3508386

```

# その成分までを用いて何%の分散が説明されるか

# 第1成分のみを採用したときの、もとの変数にかかる係数

# 第2成分まで採用したときの、もとの変数にかかる係数

# 第3成分まで採用したときの、もとの変数にかかる係数  
# もともと3変数なので、重回帰分析の結果と一致する

## 線形判別分析

独立変数：量的変数  
従属変数：質的変数

```
従属変数 <- factor(従属変数)
```

```
library(MASS)
```

```
ldaオブジェクト名 <- lda(従属変数 ~ 独立変数1 + 独立変数2 + ..., data=データ名)
```

```
print(ldaオブジェクト名)
```

あらかじめ従属変数をfactor化しておく必要がある。

MASSパッケージのlda関数 (linear discriminant analysis) を用いる。MASSパッケージはプレインストールされているので、インストールする必要はない。

各独立変数について、線形判別関数の係数を推定する。

## 判別結果の確認

```
predictオブジェクト名 <- predict(ldaオブジェクト名)
```

```
tableオブジェクト名 <- table(従属変数, predictオブジェクト名$class, dnn=c("Pre", "Post"))
```

predictオブジェクトの中に、classという変数名で予測されたカテゴリが保存されているので、従属変数の値と予測値のクロス集計表を書く。

## 一致率

```
sum(diag(tableオブジェクト名)) / sum(tableオブジェクト名)
```

対角要素の合計／全要素の合計

```
> rm(list=ls())
>
> setwd("d:¥¥¥")
>
> d1 <- read.table("DSA_SampleData.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(d1)
  unitID year      area future grade practice club motiveL motiveE stress fear support
1      1 20X1      Science  Advance      3      Few      0      17      18      20      22      9
2      2 20X2 Humanities Employment      2     Many      1      16      14      23      48      9
3      3 20X1 Interdisciplinary Advance      2     Many      1      16      18      30      58      6
4      4 20X1 Humanities Employment      1      Few      0      12      14      15      40     12
5      5 20X1      Science  Advance      3      Few      0      16      15      25      52      9
6      6 20X2 Interdisciplinary Employment      2     Many      1      18      18      26      20      4
  depression adaptation anxiety efficacy item1 item2 item3 x1 x2 x3 x4_1 x4_2 x5_1 x5_2
1          18          10      26      49      3      2      1  2  2  4      4      3      4      4
2          21          12      24      57      3      3      4  3  2  2      2      2      1      4
3          29          12      29      42      4      3      3  4  1  3      4      2      1      5
4           8          12      23      41      3      5      5  4  3  4      3      2      4      5
5          29          13      25      41      5      5      5  3  3  4      4      2      4      4
6          22           9      23      47      3      3      4  2  2  3      3      3      1      4
```

## &gt; # 記述統計量

```
> library(psych)
```

```
> describeBy(d1$fear, list(d1$area), mat=TRUE, digits=2)
```

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	Humanities	1	85	41.11	10.51	40	40.96	11.86	20	68	48	0.16	-0.31	1.14
X12	2	Interdisciplinary	1	64	39.56	12.26	40	38.88	11.86	20	76	56	0.53	0.19	1.53
X13	3	Science	1	88	41.16	12.44	42	40.44	11.86	22	78	56	0.46	-0.07	1.33

```
> describeBy(d1$depression, list(d1$area), mat=TRUE, digits=2)
```

	item	group1	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X11	1	Humanities	1	85	19.62	6.42	19	19.38	5.93	8	37	29	0.33	-0.26	0.70
X12	2	Interdisciplinary	1	64	19.39	6.48	19	19.13	5.93	8	40	32	0.45	0.34	0.81
X13	3	Science	1	88	21.66	6.44	22	21.51	7.41	8	40	32	0.24	-0.31	0.69

```
> describeBy(d1$efficacy, list(d1$area), mat=TRUE, digits=2)
      item      group1 vars  n mean  sd median trimmed  mad min max range  skew kurtosis  se
X11    1      Humanities  1 85 48.72 9.82    50  48.83 10.38  26 73   47 -0.11   -0.33 1.06
X12    2 Interdisciplinary  1 64 52.05 8.90    52  52.08  9.64  32 73   41  0.01   -0.45 1.11
X13    3         Science   1 88 48.91 9.33    47  48.46  8.90  33 77   44  0.56   -0.13 0.99
```

```
> # 線形判別分析
```

```
> d2 <- d1
```

```
> d2$area <- factor(d2$area)
```

```
> library(MASS)
```

```
> res1 <- lda(area ~ fear + depression + efficacy, data=d2)
```

```
> print(res1)
```

Call:

```
lda(area ~ fear + depression + efficacy, data = d2)
```

Prior probabilities of groups:

Humanities	Interdisciplinary	Science
0.3586498	0.2700422	0.3713080

Group means:

	fear	depression	efficacy
Humanities	41.10588	19.62353	48.71765
Interdisciplinary	39.56250	19.39062	52.04688
Science	41.15909	21.65909	48.90909

Coefficients of linear discriminants:

	LD1	LD2
fear	0.01205321	0.03883839
depression	0.10248864	-0.12507076
efficacy	-0.07316804	-0.06875929

Proportion of trace:

	LD1	LD2
0.6945	0.3055	

```
> # 予測カテゴリー等
```

```
> pres1 <- predict(res1)
```

```
> # 判別結果
```

```
> (t1 <- table(d2$area, pres1$class, dnn=c("Pre", "Post")))
```

		Post		
Pre		Humanities	Interdisciplinary	Science
	Humanities	41	13	31
	Interdisciplinary	22	15	27
	Science	31	9	48

```
> # 一致率
```

```
> sum(diag(t1)) / sum(t1)
```

```
[1] 0.4388186
```

unitID	year	area	future	grade	practice	club	motiveL	motiveE	stress	fear	support	depression	adaptation	anxiety	efficacy	it
1	20X1	Science	Advance	3	Few	0	17	18	20	22	9	18	10	26	49	
2	20X2	Humanitie	Employe	2	Many	1	16	14	23	48	9	21	12	24	57	
3	20X1	Interdiscipli	Advance	2	Many	1	16	18	30	58	6	29	12	29	42	
4	20X1	Humanitie	Employe	1	Few	0	12	14	15	40	12	8	12	23	41	
5	20X1	Science	Advance	3	Few	0	16	15	25	52	9	29	13	25	41	
6	20X2	Interdiscipli	Employe	2	Many	1	18	18	26	20	4	22	9	23	47	
7	20X1	Humanitie	Employe	3	Few	1	21	20	21	50	13	19	12	37	41	
8	20X3	Science	Advance	2	Few	0	17	16	14	22	12	12	19	26	43	
9	20X2	Interdiscipli	Advance	3	Few	1	16	17	22	44	9	19	14	20	54	
10	20X4	Science	Advance	3	Many	1	19	16	37	38	8	31	12	36	55	



## 数量化 I 類

独立変数：質的変数  
従属変数：量的変数

独立変数 <- factor(独立変数)

lmオブジェクト名 <- lm(従属変数 ~ 独立変数1 + 独立変数2 + ..., data=データ名)  
summary(lmオブジェクト名)

あらかじめ、各独立変数をfactor化しておく必要がある。  
各独立変数の第1カテゴリを基準として、第2カテゴリ以下の係数を推定する。

## 多重共線性の診断

library(car)  
vif(lmオブジェクト名)

carパッケージのvif関数を用いる。あらかじめcarパッケージをインストールしておく必要がある。

## ステップワイズ分析

stepオブジェクト名 <- step(lmオブジェクト名, direction="both")  
summary(stepオブジェクト名)

directionのオプションとして、"both", "backward", "forward"がある。

```
> rm(list=ls())
>
> setwd("d:¥¥¥")
>
> d1 <- read.table("DSA_SampleData.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  unitID year      area future grade practice club motiveL motiveE stress fear support
1      1 20X1    Science  Advance      3     Few      0      17      18      20      22       9
2      2 20X2 Humanities Employment      2    Many      1      16      14      23      48       9
3      3 20X1 Interdisciplinary Advance      2    Many      1      16      18      30      58       6
4      4 20X1 Humanities Employment      1     Few      0      12      14      15      40      12
5      5 20X1    Science  Advance      3     Few      0      16      15      25      52       9
6      6 20X2 Interdisciplinary Employment      2    Many      1      18      18      26      20       4
  depression adaptation anxiety efficacy item1 item2 item3 x1 x2 x3 x4_1 x4_2 x5_1 x5_2
1          18          10      26      49      3      2      1  2  2  4      4      3      4      4
2          21          12      24      57      3      3      4  3  2  2      2      2      1      4
3          29          12      29      42      4      3      3  4  1  3      4      2      1      5
4           8          12      23      41      3      5      5  4  3  4      3      2      4      5
5          29          13      25      41      5      5      5  3  3  4      4      2      4      4
6          22           9      23      47      3      3      4  2  2  3      3      3      1      4
```

```
> # 記述統計量
> library(psych)
>
> describeBy(d1$depression, list(d1$area), digits=2, mat=TRUE)
  item group1 vars  n mean  sd median trimmed mad min max range skew kurtosis  se
X11   1 Humanities  1 85 19.62 6.42   19  19.38 5.93   8 37  29 0.33  -0.26 0.70
X12   2 Interdisciplinary  1 64 19.39 6.48   19  19.13 5.93   8 40  32 0.45   0.34 0.81
X13   3 Science      1 88 21.66 6.44   22  21.51 7.41   8 40  32 0.24  -0.31 0.69

> describeBy(d1$depression, list(d1$future), digits=2, mat=TRUE)
  item group1 vars  n mean  sd median trimmed mad min max range skew kurtosis  se
X11   1 Advance    1 116 21.73 6.89   21.5  21.49 7.41   8 40  32 0.30  -0.38 0.64
X12   2 Employment  1 121 18.96 5.82   19.0  18.92 5.93   8 37  29 0.12  -0.26 0.53

> describeBy(d1$depression, list(d1$club), digits=2, mat=TRUE)
  item group1 vars  n mean  sd median trimmed mad min max range skew kurtosis  se
X11   1      0    1 119 20.08 7.39   20  19.77 7.41   8 40  32 0.34  -0.41 0.68
X12   2      1    1 118 20.56 5.48   20  20.41 5.93   9 37  28 0.35  -0.06 0.50
```

```

> # 数量化1類
> d2 <- d1
> d2$area <- factor(d2$area)
> d2$practice <- factor(d2$future)
> d2$club <- factor(d2$club)

> q1 <- lm(depression ~ area + future + club, data=d2)
> summary(q1)

Call:
lm(formula = depression ~ area + future + club, data = d2)

Residuals:
    Min       1Q   Median       3Q      Max
-13.8339  -4.4383   0.1498   4.1661  19.4564

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)      21.4290     1.1443  18.727  <2e-16 ***
areaInterdisciplinary -0.8854     1.0928  -0.810   0.4187
areaScience         0.4049     1.1602   0.349   0.7274
futureEmployment    -2.5788     0.9939  -2.595   0.0101 *
club1                0.5881     0.8326   0.706   0.4807
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.377 on 232 degrees of freedom
Multiple R-squared:  0.05422, Adjusted R-squared:  0.03791
F-statistic: 3.325 on 4 and 232 DF, p-value: 0.01134

```

```

> # 多重共線性
> library(car)
> vif(q1)
          GVIF Df GVIF^(1/(2*Df))
area    1.442318 2      1.095886
future  1.438531 1      1.199388
club    1.009948 1      1.004961

```

```

> # ステップワイズ分析
> stepq1 <- step(q1, direction="both")

```

```

Start: AIC=883.12
depression ~ area + future + club

      Df Sum of Sq  RSS   AIC
- area    2    59.204 9493.6 880.61
- club    1    20.291 9454.7 881.63
<none>                 9434.4 883.12
- future  1   273.793 9708.2 887.90

```

```

Step: AIC=880.61
depression ~ future + club

      Df Sum of Sq  RSS   AIC
- club    1    25.88 9519.5 879.25
<none>                 9493.6 880.61
+ area    2    59.20 9434.4 883.12
- future  1   467.77 9961.4 890.00

```

```

Step: AIC=879.25
depression ~ future

      Df Sum of Sq  RSS   AIC
<none>                 9519.5 879.25
+ club    1    25.88 9493.6 880.61
+ area    2    64.79 9454.7 881.63
- future  1   455.76 9975.3 888.33

```

```
> summary(stepq1)
```

```
Call:
```

```
lm(formula = depression ~ future, data = d2)
```

```
Residuals:
```

```
      Min       1Q   Median       3Q      Max
-13.7328  -3.9587   0.0413   4.2672  18.2672
```

```
Coefficients:
```

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    21.7328     0.5909  36.777 < 2e-16 ***
futureEmployment -2.7741     0.8270  -3.354 0.000928 ***
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 6.365 on 235 degrees of freedom
```

```
Multiple R-squared:  0.04569,    Adjusted R-squared:  0.04163
```

```
F-statistic: 11.25 on 1 and 235 DF,  p-value: 0.0009277
```

unitID	year	area	future	grade	practice	club	motiveL	motiveE	stress	fear	support	depression	adaptation	anxiety	efficacy	it
1	20X1	Science	Advance	3	Few		0	17	18	20	22	9	18	10	26	49
2	20X2	Humanitie	Employme	2	Many		1	16	14	23	48	9	21	12	24	57
3	20X1	Interdiscipli	Advance	2	Many		1	16	18	30	58	6	29	12	29	42
4	20X1	Humanitie	Employme	1	Few		0	12	14	15	40	12	8	12	23	41
5	20X1	Science	Advance	3	Few		0	16	15	25	52	9	29	13	25	41
6	20X2	Interdiscipli	Employme	2	Many		1	18	18	26	20	4	22	9	23	47
7	20X1	Humanitie	Employme	3	Few		1	21	20	21	50	13	19	12	37	41
8	20X3	Science	Advance	2	Few		0	17	16	14	22	12	12	19	26	43
9	20X2	Interdiscipli	Advance	3	Few		1	16	17	22	44	9	19	14	20	54
10	20X4	Science	Advance	3	Many		1	19	16	37	38	8	31	12	36	55

## 数量化Ⅱ類

独立変数：質的変数  
従属変数：質的変数

```
独立変数 <- factor(独立変数)
従属変数 <- factor(従属変数)
```

```
library(MASS)
ldaオブジェクト名 <- lda(従属変数 ~ 独立変数1 + 独立変数2 + ..., data=データ名)
print(ldaオブジェクト名)
```

あらかじめ、各独立変数、従属変数をfactor化しておく必要がある。

MASSパッケージのlda関数 (linear discriminant analysis) を用いる。MASSパッケージはプレインストールされているので、インストールする必要はない。

各独立変数の第1カテゴリを基準として、第2カテゴリ以下のカテゴリスコア、また、各個体のサンプルスコアを推定する。

## サンプルスコアのプロット

```
predictオブジェクト名 <- predict(ldaオブジェクト名)
plot(従属変数, predictオブジェクト名$x)
```

predictオブジェクトの中に、xという変数名でサンプルスコアが保存されているので、各独立変数の水準ごとに、サンプルスコアの分布を比較する図を描く。

```
> rm(list=ls())
>
> setwd("d:¥¥¥")
>
> d1 <- read.table("DSA_SampleData.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  unitID year      area future grade practice club motiveL motiveE stress fear support
1      1 20X1    Science  Advance     3     Few     0     17     18     20    22      9
2      2 20X2 Humanities Employment  2    Many     1     16     14     23    48      9
3      3 20X1 Interdisciplinary Advance  2    Many     1     16     18     30    58      6
4      4 20X1 Humanities Employment  1     Few     0     12     14     15    40     12
5      5 20X1      Science  Advance  3     Few     0     16     15     25    52      9
6      6 20X2 Interdisciplinary Employment  2    Many     1     18     18     26    20      4

  depression adaptation anxiety efficacy item1 item2 item3 x1 x2 x3 x4_1 x4_2 x5_1 x5_2
1          18          10      26      49      3      2      1  2  2  4      4      3      4      4
2          21          12      24      57      3      3      4  3  2  2      2      2      1      4
3          29          12      29      42      4      3      3  4  1  3      4      2      1      5
4           8          12      23      41      3      5      5  4  3  4      3      2      4      5
5          29          13      25      41      5      5      5  3  3  4      4      2      4      4
6          22           9      23      47      3      3      4  2  2  3      3      3      1      4
```

```
> # 数量化2類
> d2 <- d1
> d2$future <- factor(d2$future)
> d2$year <- factor(d2$year)
> d2$area <- factor(d2$area)
> d2$practice <- factor(d2$practice)
>
> library(MASS)
> q2 <- lda(future ~ year + area + practice, data=d2)
> print(q2)
```

Call:

```
lda(future ~ year + area + practice, data = d2)
```

Prior probabilities of groups:

```
Advance Employment
0.4894515 0.5105485
```

Group means:

	year20X2	year20X3	year20X4	areaInterdisciplinary	areaScience	practiceMany
Advance	0.2758621	0.2672414	0.3448276	0.2500000	0.6206897	0.4741379
Employment	0.2727273	0.2727273	0.1487603	0.2892562	0.1322314	0.5123967

## # カテゴリスコア

Coefficients of linear discriminants:

	LD1
year20X2	-0.6360216
year20X3	-0.6806659
year20X4	-1.5091438
areaInterdisciplinary	-1.1934599
areaScience	-2.5984586
practiceMany	0.3226241

## > # サンプルスコアのプロット

```
> pq2 <- predict(q2)
```

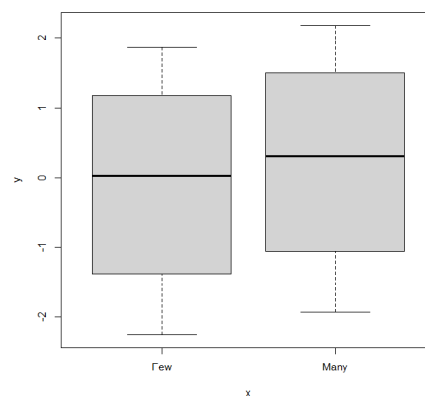
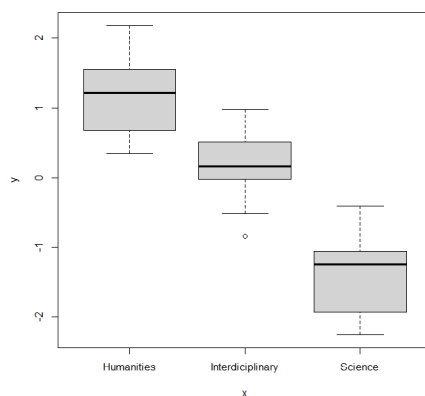
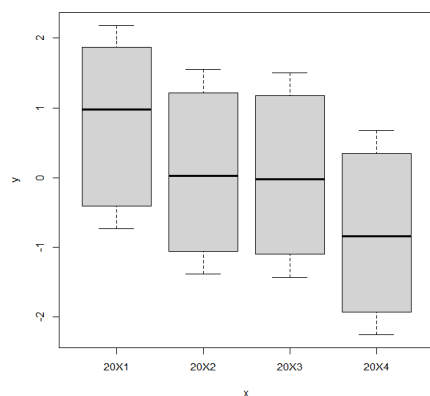
```
> plot(d2$year, pq2$x)
```

```
>
```

```
> plot(d2$area, pq2$x)
```

```
>
```

```
> plot(d2$practice, pq2$x)
```



unitID	year	area	future	grade	practice	club	motiveL	motiveE	stress	fear	support	depression	adaptation	anxiety	efficacy	it
1	20X1	Science	Advance	3	Few	0	17	18	20	22	9	18	10	26	49	
2	20X2	Humanitie	Employme	2	Many	1	16	14	23	48	9	21	12	24	57	
3	20X1	Interdiscipli	Advance	2	Many	1	16	18	30	58	6	29	12	29	42	
4	20X1	Humanitie	Employme	1	Few	0	12	14	15	40	12	8	12	23	41	
5	20X1	Science	Advance	3	Few	0	16	15	25	52	9	29	13	25	41	
6	20X2	Interdiscipli	Employme	2	Many	1	18	18	26	20	4	22	9	23	47	
7	20X1	Humanitie	Employme	3	Few	1	21	20	21	50	13	19	12	37	41	
8	20X3	Science	Advance	2	Few	0	17	16	14	22	12	12	19	26	43	
9	20X2	Interdiscipli	Advance	3	Few	1	16	17	22	44	9	19	14	20	54	
10	20X4	Science	Advance	3	Many	1	19	16	37	38	8	31	12	36	55	

## 数量化Ⅲ類

質的変数の各カテゴリをダミー変数化した対応分析に相当。

## ダミー変数データの作成

ダミー変数名 <- ifelse(データ名\$質的変数名=="カテゴリ名", 1, 0)

質的変数ごとに、第2カテゴリ以下の各カテゴリへの応答をダミー変数化する。応答が第1カテゴリであれば、各ダミー変数の値はすべて0、応答が第2カテゴリ以下であれば、当該ダミー変数の値が1、その他のカテゴリのダミー変数の値が0になる。

## データがすべて0の行、列を削除

データ名 <- データ名[rowSums(データ名)>0, colSums(データ名)>0]

## 【重要！】

すべてのダミー変数の値が0の個体（すべての質的変数の回答が第1カテゴリ）、または、すべての値が0のダミー変数（だれもそのカテゴリを選択していない）があると計算できないので、そのような行や列を削除する。

## library(MASS)

オブジェクト名 <- corresp(オブジェクト名, nf=次元数)

MASSパッケージのcorresp関数を用いる。MASSパッケージはプレインストールされているので、インストールする必要はない。

各独立変数の第1カテゴリを基準として、第2カテゴリ以下のカテゴリスコア（列スコア）、また、各個体のサンプルスコア（行スコア）を推定する。

```
> rm(list=ls())
>
> setwd("d:¥¥¥")
>
> d1 <- read.table("DSA_SampleData.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

unitID	year	area	future	grade	practice	club	motiveL	motiveE	stress	fear	support
1	20X1	Science	Advance	3	Few	0	17	18	20	22	9
2	20X2	Humanities	Employment	2	Many	1	16	14	23	48	9
3	20X1	Interdisciplinary	Advance	2	Many	1	16	18	30	58	6
4	20X1	Humanities	Employment	1	Few	0	12	14	15	40	12
5	20X1	Science	Advance	3	Few	0	16	15	25	52	9
6	20X2	Interdisciplinary	Employment	2	Many	1	18	18	26	20	4

```

depression adaptation anxiety efficacy item1 item2 item3 x1 x2 x3 x4_1 x4_2 x5_1 x5_2
1      18      10      26      49      3      2      1  2  2  4      4      3      4      4
2      21      12      24      57      3      3      4  3  2  2      2      2      1      4
3      29      12      29      42      4      3      3  4  1  3      4      2      1      5
4       8      12      23      41      3      5      5  4  3  4      3      2      4      5
5      29      13      25      41      5      5      5  3  3  4      4      2      4      4
6      22       9      23      47      3      3      4  2  2  3      3      3      1      4

```

```
> # 度数分布表
> table(d1$year)
```

```
20X1 20X2 20X3 20X4
  50   65   64   58
```

```
> table(d1$area)
```

```
Humanities Interdisciplinary Science
      85           64           88
```

```
> table(d1$future)
```

```
Advance Employment
    116         121
```

unitID	year	area	future	grade	practice	club	motiveL	motiveE	stress	fear	support	depression	adaptation	anxiety	efficacy	it
1	20X1	Science	Advance	3	Few	0	17	18	20	22	9	18	10	26	49	
2	20X2	Humanitie	Employe	2	Many	1	16	14	23	48	9	21	12	24	57	
3	20X1	Interdiscipli	Advance	2	Many	1	16	18	30	58	6	29	12	29	42	
4	20X1	Humanitie	Employe	1	Few	0	12	14	15	40	12	8	12	23	41	
5	20X1	Science	Advance	3	Few	0	16	15	25	52	9	29	13	25	41	
6	20X2	Interdiscipli	Employe	2	Many	1	18	18	26	20	4	22	9	23	47	
7	20X1	Humanitie	Employe	3	Few	1	21	20	21	50	13	19	12	37	41	
8	20X3	Science	Advance	2	Few	0	17	16	14	22	12	12	19	26	43	
9	20X2	Interdiscipli	Advance	3	Few	1	16	17	22	44	9	19	14	20	54	
10	20X4	Science	Advance	3	Many	1	19	16	37	38	8	31	12	36	55	

```

> # ダミー変数データの作成
> d2 <- NULL
>
> d2$year20X2 <- ifelse(d1$year=="20X2", 1, 0)
> d2$year20X3 <- ifelse(d1$year=="20X3", 1, 0)
> d2$year20X4 <- ifelse(d1$year=="20X4", 1, 0)
>
> d2$areaScience <- ifelse(d1$area=="Science", 1, 0)
> d2$areaInterdisciplinary <- ifelse(d1$area=="Interdisciplinary", 1, 0)
>
> d2$futureEmployment <- ifelse(d1$future=="Employment", 1, 0)
>
> d2 <- data.frame(d2)

```

```

> # 数量化3類
> library(MASS)

> # データがすべて0の行, 列を削除
> d3 <- d2[rowSums(d2)>0, colSums(d2)>0]

> (q3 <- corresp(d3, nf=2))

```

First canonical correlation(s): 0.7636169 0.7057800

Row scores:

	[, 1]	[, 2]
1	-1.58614017	-1.66682273
2	0.69749634	-0.39196293
3	1.24484470	1.42570092
...		
235	-1.94594360	0.36265267
236	-0.39985396	-1.32931676
237	-0.06371583	-0.81691619

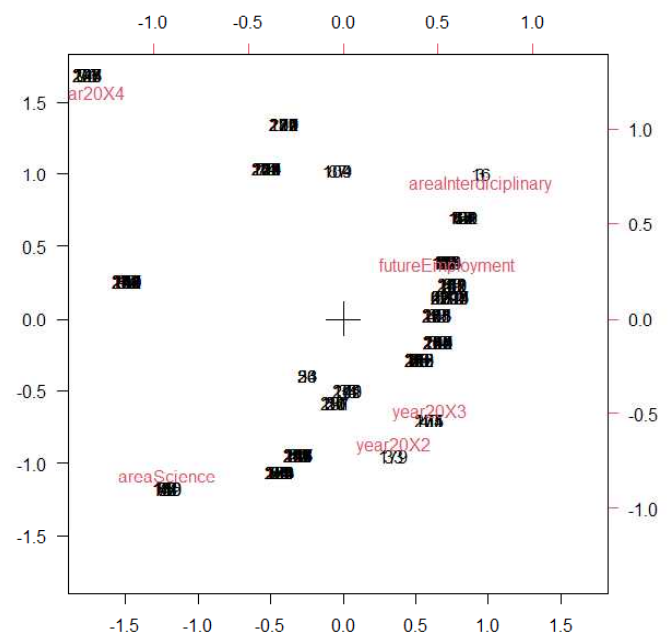
Column scores:

	[, 1]	[, 2]
year20X2	0.3474754	-0.9455314
year20X3	0.6005329	-0.7000003
year20X4	-1.7607073	1.6883163
areaScience	-1.2112034	-1.1764102
areaInterdisciplinary	0.9505844	1.0062313
futureEmployment	0.7177646	0.3922522

```

> # カテゴリ値の図示
> biplot(q3, las=1)

```



## 対応分析

データフレーム化したクロス表を用いて、行カテゴリと、列カテゴリの関連性を分析し、図示する。

## クロス表のデータフレーム化

```
テーブル名 <- table(行変数名, 列変数名)
データフレーム名 <- as.data.frame(matrix(テーブル名, 行数, 列数))
rownames(データフレーム名) <- c("行カテゴリ1", "行カテゴリ2", ...)
colnames(データフレーム名) <- c("列カテゴリ1", "列カテゴリ2", ...)
```

## MASS パッケージを使う方法

```
library(MASS)
オブジェクト名 <- corresp(データフレーム名, nf=2)
plot(オブジェクト名, las=1)
```

MASSパッケージは、Rインストール時に、自動的にインストールされてる。  
テーブルをデータフレーム化したものを読み込む。

## CA パッケージを用いる方法

```
library(ca)
オブジェクト名 <- ca(データフレーム名)
plot(オブジェクト名)
```

あらかじめ ca パッケージをインストールしておく必要がある。  
テーブルをデータフレーム化したものを読み込む。

## FactoMineR パッケージを使う方法

```
library(FactoMineR)
オブジェクト名 <- CA(データフレーム名)
plot(オブジェクト名)
```

あらかじめ FactoMineR パッケージをインストールしておく必要がある。  
テーブルをデータフレーム化したものを読み込む。

```
> setwd("i:¥¥¥¥documents¥¥¥¥scripts¥¥¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 入学年度  学科 性別 モラトリアム 自己効力感 学習意欲 進路
1    1      20Y1 看護学   女         高          49        23 就職
2    2      20Y2 心理学   男         低          57        29 就職
3    3      20Y1  医学   女         高          42        23 進学
4    4      20Y1 看護学   女         高          41        23 就職
5    5      20Y2  医学   男         低          41        22 就職
6    6      20Y1 心理学   女         低          47        24 就職
>
```

## # クロス表

```
> (t1 <- table(d1$進路, d1$学科))

      医学 看護学 心理学
就職    64     80     75
進学    10     13     15
不明     2      7      4
>
```

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職



```
> # データフレーム化したクロス表の作成
> t2 <- as.data.frame(matrix(t1,3,3))
> rownames(t2) <- c("Work", "Study", "Unkown")
> colnames(t2) <- c("Med", "Nur", "Psy")
> t2
```

	Med	Nur	Psy
Work	64	80	75
Study	10	13	15
Unkown	2	7	4

```
>
```

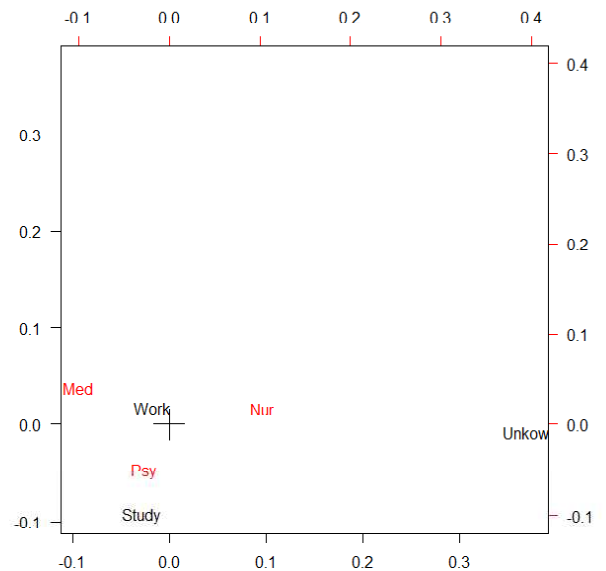
```
> # 対応分析
```

```
> # MASS パッケージを使う方法
> library(MASS)
>
> cr1 <- corresp(t2, nf=2)
> cr1
First canonical correlation(s): 0.08388651 0.03824868
```

```
Row scores:
      [,1]      [,2]
Work -0.2042625  0.4372111
Study -0.3421564 -2.4470783
Unkown  4.4411877 -0.2123270
```

```
Column scores:
      [,1]      [,2]
Med -1.1939676  1.0616369
Nur  1.2277632  0.4388595
Psy -0.3407955 -1.3252165
>
```

```
> # プロット表示
> plot(cr1, las=1)
>
>
>
```



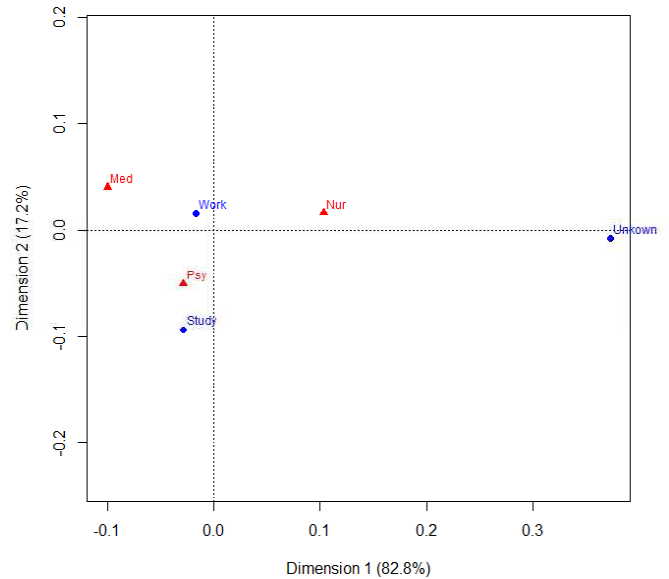
```
> # CA パッケージを用いる方法
> library(ca)
>
> cr2 <- ca(t2)
> cr2
```

```
Principal inertias (eigenvalues):
      1      2
Value  0.007037 0.001463
Percentage 82.79% 17.21%
```

```
Rows:
      Work      Study      Unkown
Mass    0.811111  0.140741  0.048148
ChiDist 0.023943  0.097900  0.372644
Inertia 0.000465  0.001349  0.006686
Dim. 1  -0.204263 -0.342156  4.441188
Dim. 2   0.437211 -2.447078 -0.212327
```

```
Columns:
      Med      Nur      Psy
Mass    0.281481 0.370370 0.348148
ChiDist 0.108076 0.104352 0.058194
Inertia 0.003288 0.004033 0.001179
Dim. 1  -1.193968 1.227763 -0.340796
Dim. 2   1.061637 0.438859 -1.325217
>
```

```
> # プロット表示
> plot(cr2)
>
>
>
>
```



```
> # FactoMineR パッケージを使う方法
> library(FactoMineR)
>
> cr3 <- CA(t2)
> cr3
```

**\*\*Results of the Correspondence Analysis (CA)\*\***

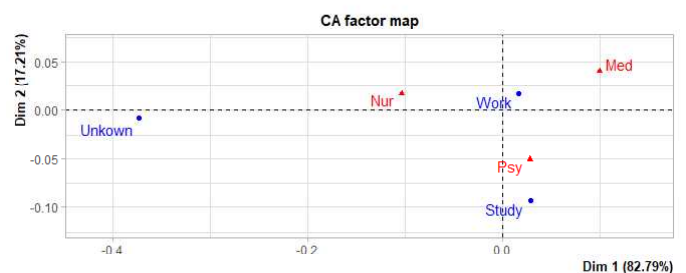
The row variable has 3 categories; the column variable has 3 categories

The chi square of independence between the two variables is equal to 2.294975 (p-value = 0.681684).

\*The results are available in the following objects:

name	description
1 "\$eig"	"eigenvalues"
2 "\$col"	"results for the columns"
3 "\$col\$coord"	"coord. for the columns"
4 "\$col\$cos2"	"cos2 for the columns"
5 "\$col\$contrib"	"contributions of the columns"
6 "\$row"	"results for the rows"
7 "\$row\$coord"	"coord. for the rows"
8 "\$row\$cos2"	"cos2 for the rows"
9 "\$row\$contrib"	"contributions of the rows"
10 "\$call"	"summary called parameters"
11 "\$call\$marge.col"	"weights of the columns"
12 "\$call\$marge.row"	"weights of the rows"

```
> # プロット表示
> plot(cr3)
>
>
>
>
>
>
>
```



## 多重対応分析

## MASS パッケージを使う方法

```
library(MASS)
オブジェクト名 <- mca(データフレーム名, nf=2)
```

```
biplot(オブジェクト名$rs, オブジェクト名$cs, var.axes=F)
```

## カテゴリのみのプロット

```
plot(オブジェクト名$cs, las=1)
text(オブジェクト名$cs, rownames(オブジェクト名$cs))
```

MASSパッケージは、Rインストール時に、自動的にインストールされてる。

## CA パッケージを用いる方法

```
library(ca)
オブジェクト名 <- mjca(データフレーム名)
plot(オブジェクト名)
```

あらかじめ ca パッケージをインストールしておく必要がある。

## FactoMineR パッケージを使う方法

```
library(FactoMineR)
オブジェクト名 <- MCA(データフレーム名)
plot(オブジェクト名)
```

あらかじめ FactoMineR パッケージをインストールしておく必要がある。

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("統計図表データ.csv", header=T, sep=";", fileEncoding="shift-jis")
> head(d1)
```

	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
1	1	20Y1	看護学	女	高	49	23	就職
2	2	20Y2	心理学	男	低	57	29	就職
3	3	20Y1	医学	女	高	42	23	進学
4	4	20Y1	看護学	女	高	41	23	就職
5	5	20Y2	医学	男	低	41	22	就職
6	6	20Y1	心理学	女	低	47	24	就職

	A	B	C	D	E	F	G	H
1	番号	入学年度	学科	性別	モラトリアム	自己効力感	学習意欲	進路
2	1	20Y1	看護学	女	高	49	23	就職
3	2	20Y2	心理学	男	低	57	29	就職
4	3	20Y1	医学	女	高	42	23	進学
5	4	20Y1	看護学	女	高	41	23	就職
6	5	20Y2	医学	男	低	41	22	就職
7	6	20Y1	心理学	女	低	47	24	就職
8	7	20Y3	看護学	女	高	41	26	就職
9	8	20Y2	医学	男	低	43	29	就職
10	9	20Y2	看護学	女	低	54	21	就職
11	10	20Y3	医学	女	高	55	30	進学
12	11	20Y1	心理学	男	低	66	25	就職
13	12	20Y1	心理学	女	低	38	30	就職
14	13	20Y1	心理学	女	低	45	29	就職
15	14	20Y1	医学	男	高	53	23	就職
16	15	20Y1	心理学	男	高	47	25	就職

```
> # 必要な変数の取り出し
> d2 <- d1[,c("入学年度", "学科", "進路")]
```

```
> # 変数のfactor化
> for(i in colnames(d2)){
+   d2[,i] <- factor(d2[,i])
+ }
```

```
> # 多重対応分析
```

```
> # MASS パッケージを使う方法
```

```
> library(MASS)
```

```
>
```

```
> mcr1 <- mca(d2, nf=2)
```

```
> mcr1
```

```
Call:
```

```
mca(df = d2, nf = 2)
```

Multiple correspondence analysis of 270 cases of 3 factors

Correlations 0.659 0.606 cumulative % explained 32.96 63.24

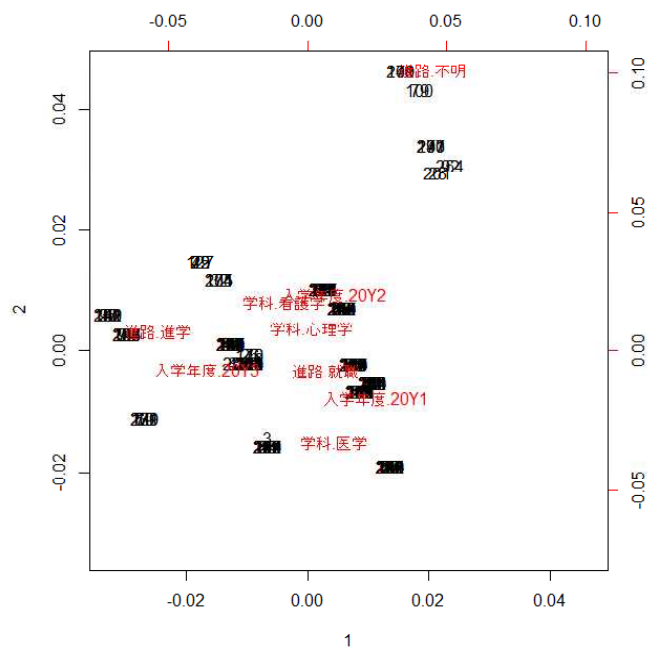
```
>
```

```
> # プロット表示
```

```
> biplot(mcr1$rs, mcr1$cs, var.axes=F)
```

```
>
```

```
# 変数カテゴリとオブザベーションの
# 両方が表示される。
```

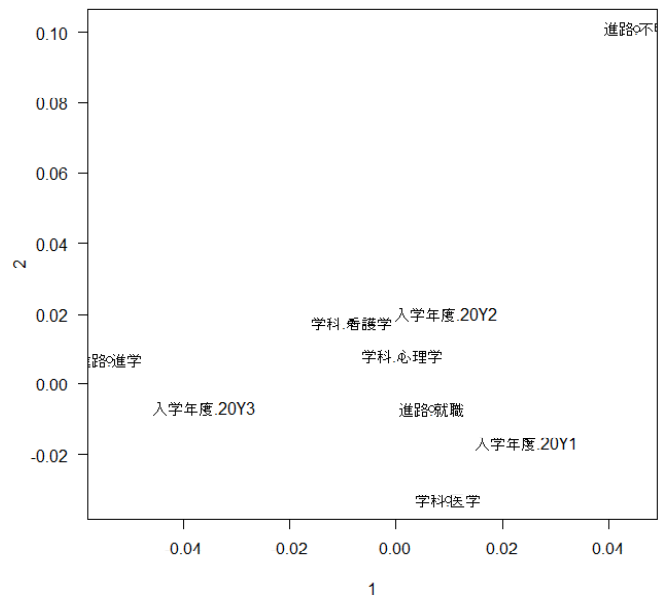


```
> # カテゴリのみ
```

```
> plot(mcr1$cs, las=1)
```

```
> text(mcr1$cs, rownames(mcr1$cs))
```

```
>
```



```
>
> # CA パッケージを用いる方法
> library(ca)
>
> mcr2 <- mjca(d2)
> mcr2
```

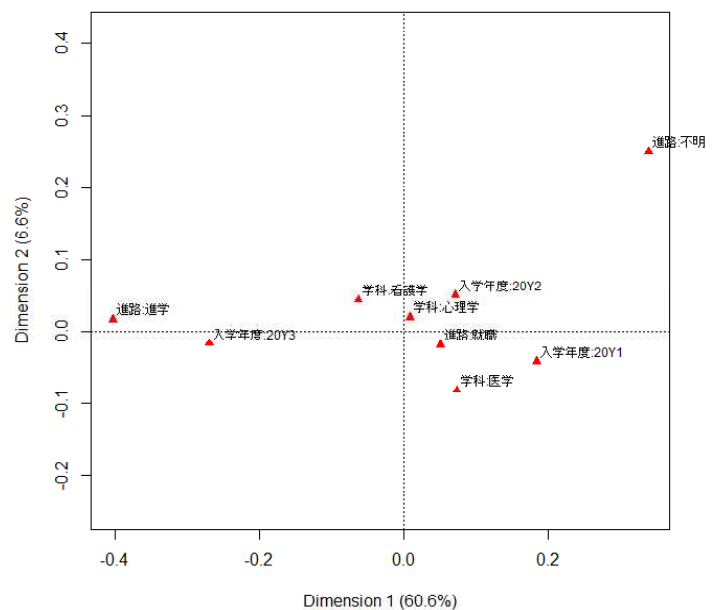
```
Eigenvalues:
      1          2          3
Value  0.023097 0.002512 0.000753
Percentage 60.59%   6.59%   1.98%
```

```
Columns:
  入学年度:20Y1  入学年度:20Y2  入学年度:20Y3  学科:医学  学科:看護学
Mass          0.107407    0.120988    0.104938  0.093827  0.123457
ChiDist       0.856142    0.770249    0.891904  0.928060  0.761002
Inertia       0.078727    0.071780    0.083478  0.080813  0.071497
Dim. 1        1.209453    0.469473   -1.779185  0.480274 -0.416871
Dim. 2       -0.818424    1.005151   -0.321199 -1.622477  0.862505

  学科:心理学  進路:就職  進路:進学  進路:不明
Mass          0.116049  0.270370  0.046914  0.016049
ChiDist       0.792695  0.282452  1.476909  2.605635
Inertia       0.072921  0.021570  0.102331  0.108965
Dim. 1        0.055173  0.329027 -2.658654  2.228612
Dim. 2        0.394231 -0.354900  0.340376  4.983760
>
```

```
> # プロット表示
> plot(mcr2)
>
>
>
```

# 変数カテゴリのみが表示される。



```
>
> # FactoMineR パッケージを使う方法
> library(FactoMineR)
>
> mcr3 <- MCA(d2)
> mcr3
```

```
**Results of the Multiple Correspondence Analysis (MCA)**
The analysis was performed on 270 individuals, described by 3 variables
*The results are available in the following objects:
```

	name	description
1	"\$eig"	"eigenvalues"
2	"\$var"	"results for the variables"
3	"\$var\$coord"	"coord. of the categories"

```

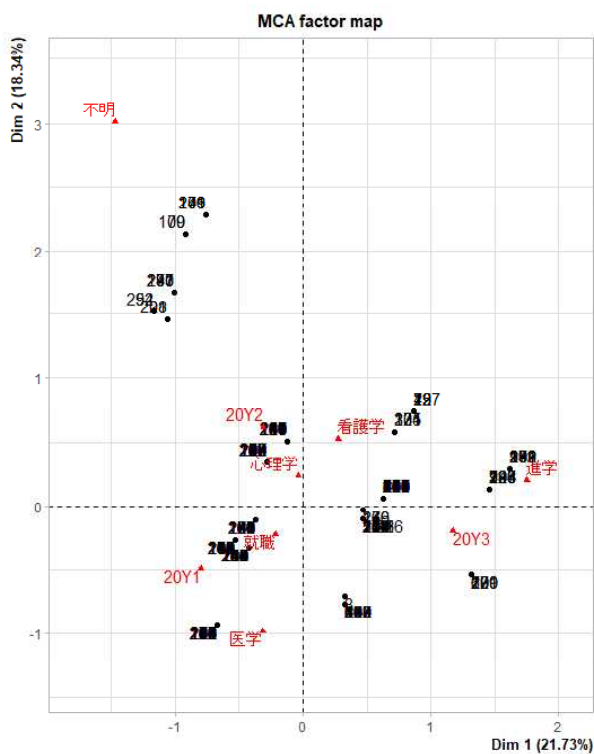
4 "$var$cos2"      "cos2 for the categories"
5 "$var$contrib"   "contributions of the categories"
6 "$var$v.test"    "v-test for the categories"
7 "$ind"           "results for the individuals"
8 "$ind$coord"     "coord. for the individuals"
9 "$ind$cos2"      "cos2 for the individuals"
10 "$ind$contrib"   "contributions of the individuals"
11 "$call"          "intermediate results"
12 "$call$marge.col" "weights of columns"
13 "$call$marge.li"  "weights of rows"
>

```

```

> # プロット表示
> plot(mcr3)
>
>
# 変数カテゴリとオブザベーションの
# 両方が表示される。

```



## ロジスティック回帰分析 — 素データ

オブジェクト名 <- glm(基準変数 ~ 説明変数, family=binomial, データフレーム名)  
summary(オブジェクト名)

glm() でモデルを指定し、結果を「オブジェクト名」に保存する。その内容をsummary()で表示する。

## 疑似R2乗の推定

library(DescTools)

PseudoR2(オブジェクト名, which="all")

あらかじめ DescTools パッケージをインストールしておく必要がある。  
whichで "McFadden", "McFaddenAdj", "CoxSnell", "Nagelkerke", "AldrichNelson", "VeallZimmermann", "Efron", "McKelveyZavoina", "Tjur", "all" のいずれかを指定する。すべて出力する場合は"all"とする。  
which を指定しない場合は "McFadden" の疑似R2乗が出力される。

```
> setwd("d:¥¥¥")
> d1 <- read.table("ロジスティック回帰分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  kyoufu support stress result
1     29      16     28      0
2     27      19     17      1
3     27      16     21      1
4     26      23     22      0
5     26      22     23      1
6     26      20     25      1
>
```

## &gt; # 記述統計量

```
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD kyoufu support stress result
kyoufu 245 15.22 5.37  1.00  -0.10  0.43  -0.21
support 245 20.32 6.08 -0.10  1.00  -0.32  0.48
stress  245 18.52 5.12  0.43 -0.32  1.00  -0.39
result  245  0.55 0.50 -0.21  0.48 -0.39  1.00
```

>

## &gt; # 説明変数の標準化

```
> d2 <- as.data.frame(scale(d1[, c("kyoufu", "support", "stress")]))
> d2 <- data.frame(d2, d1$result)
> colnames(d2) <- c("kyoufu", "support", "stress", "result")
```

## &gt; # 共分散行列の確認

```
> cov(d2)
      kyoufu support stress result
kyoufu 1.0000000 0.1034273 0.4255222 -0.1032508
support -0.1034273 1.0000000 -0.3231937 0.2385116
stress  0.4255222 -0.3231937 1.0000000 -0.1966482
result -0.1032508 0.2385116 -0.1966482 0.2484108
```

## &gt; # 標準偏回帰係数の推定

```
> result.2 <- glm(result ~ kyoufu + support + stress, family=binomial, d2)
> summary(result.2)
```

Call:

```
glm(formula = result ~ kyoufu + support + stress, family = binomial,
    data = d2)
```

Deviance Residuals:

```
      Min       1Q   Median       3Q      Max
-2.5189  -0.8689   0.3166   0.7980   2.0672
```

	A	B	C	D
1	kyoufu	support	stress	result
2	29	16	28	0
3	27	19	17	1
4	27	16	21	1
5	26	23	22	0
6	26	22	23	1
7	26	20	25	1
8	26	20	21	0
9	26	19	25	0
10	26	16	26	0
11	25	32	24	1
12	25	15	21	0
13	25	13	29	0
14	25	12	26	0
15	24	21	20	1
16	24	11	28	0
17	23	31	26	1
18	23	29	20	1
19	23	28	23	1
20	23	21	24	0
21	23	14	29	0

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.3065	0.1568	1.954	0.050708 .
kyoufu	-0.1587	0.1715	-0.926	0.354625
support	1.1552	0.1954	5.912	3.38e-09 ***
stress	-0.7613	0.1980	-3.845	0.000121 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 337.09 on 244 degrees of freedom  
 Residual deviance: 249.58 on 241 degrees of freedom  
 AIC: 257.58

Number of Fisher Scoring iterations: 5

&gt; # 疑似R2乗の推定

&gt; library(DescTools)

&gt; PseudoR2(result.2, which="all")

	McFadden	McFaddenAdj	CoxSnell	Nagelkerke
	0.2596013	0.2358685	0.3003507	0.4018721
	AldrichNelson	VeallZimmermann	Efron	McKelveyZavoina
	0.2631759	0.4544564	0.3294545	0.4449739
	Tjur	AIC	BIC	logLik
	0.3216345	257.5785243	271.5835572	-124.7892622
	logLik0	G2		
	-168.5433263	87.5081282		

&gt; # (標準) 偏回帰係数の信頼区間

&gt; confint(result.2)

Waiting for profiling to be done...

	2.5 %	97.5 %
(Intercept)	0.002021354	0.6187654
kyoufu	-0.498291957	0.1768211
support	0.789789998	1.5587577
stress	-1.164559209	-0.3850987

&gt; #ステップワイズ分析

&gt; #MASSパッケージの読み込み

&gt; library(MASS)

&gt; result.3 &lt;- stepAIC(result.2)

Start: AIC=257.58

result ~ kyoufu + support + stress

	Df	Deviance	AIC
- kyoufu	1	250.44	256.44
<none>		249.58	257.58
- stress	1	266.10	272.10
- support	1	295.24	301.24

Step: AIC=256.44

result ~ support + stress

	Df	Deviance	AIC
<none>		250.44	256.44
- stress	1	273.83	277.83
- support	1	295.74	299.74

&gt; summary(result.3)

Call:

glm(formula = result ~ support + stress, family = binomial, data = d2)



## Deviance Residuals:

Min	1Q	Median	3Q	Max
-2.4713	-0.8605	0.3192	0.8171	2.1162

## Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.3070	0.1565	1.961	0.0498 *
support	1.1502	0.1954	5.886	3.95e-09 ***
stress	-0.8292	0.1850	-4.482	7.39e-06 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 337.09 on 244 degrees of freedom  
 Residual deviance: 250.44 on 242 degrees of freedom  
 AIC: 256.44

Number of Fisher Scoring iterations: 5

## ロジスティック回帰分析 ― 集計データ

オブジェクト名 <- glm(割合変数 ~ 独立変数群, data=データフレーム名, weights=ウエイト変数,  
family=binomial(link="リンク関数名"))

または

オブジェクト名 <- glm(二値変数 ~ 独立変数群, data=データフレーム名,  
family=binomial(link="リンク関数名"))

summary(オブジェクト名)

confint(オブジェクト名)

データが集計されている場合には、独立変数の各水準における事象発生割合を基準変数とし、ウエイト変数（独立変数の各水準のサイズ）を指定する。

データが集計されていない（0,1のまま）場合には、二値変数を基準変数に用いる。

ランダム成分は二項分布とする。

リンク関数 identity: 恒等リンク      線形回帰  
logit:      ロジット関数      ロジスティック回帰  
probit:      プロビット関数      プロビット回帰

> # Data from Norton P. G. & Dunn, E. V. (1985). Br. Med. J., 291, 630-632.  
> # Agresti, A. (2007). An introduction to categorical data analysis, 2nd ed. Wiley, p69.

> setwd("i:¥¥Rdocuments¥¥programs¥¥")

> d1 <- read.table("snoringData.csv", header=TRUE, sep="," , fileEncoding="shift-jis")

> d1\$snr <- c(0, 2, 4, 5)

> d1\$subtotal <- d1\$yes + d1\$no

> d1\$pyes <- d1\$yes / d1\$subtotal

> d1

	snoring	yes	no	snr	subtotal	pyes
1	never	24	1355	0	1379	0.01740392
2	occasional	35	603	2	638	0.05485893
3	nearly	21	192	4	213	0.09859155
4	every	30	224	5	254	0.11811024

	A	B	C
1	snoring	yes	no
2	never	24	1355
3	occasional	35	603
4	nearly	21	192
5	every	30	224
n			

> # GLM

> #Binomial(identity) linear regression

> result.1 <- glm(pyes ~ snr, data=d1, weights=subtotal,  
+ family=binomial(link="identity"))

> summary(result.1)

Call:

glm(formula = pyes ~ snr, family = binomial(link = "identity"),  
data = d1, weights = subtotal)

Deviance Residuals:

1	2	3	4
0.04478	-0.21322	0.11010	0.09798

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	0.017247	0.003451	4.998	5.80e-07 ***
snr	0.019778	0.002805	7.051	1.77e-12 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 65.904481 on 3 degrees of freedom  
Residual deviance: 0.069191 on 2 degrees of freedom  
AIC: 24.322

Number of Fisher Scoring iterations: 3

```
> confint(result.1)
Waiting for profiling to be done...
                2.5 %      97.5 %
(Intercept) 0.01132939 0.02483298
snr          0.01451898 0.02550672
```

```
> #Binomial(logit) logistic regression
> result.2 <- glm(pyes ~ snr, data=d1, weights=subtotal,
+               family=binomial(link="logit"))
```

```
> summary(result.2)
```

Call:  
glm(formula = pyes ~ snr, family = binomial(link = "logit"),  
data = d1, weights = subtotal)

Deviance Residuals:

1	2	3	4
-0.8346	1.2521	0.2758	-0.6845

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-3.86625	0.16621	-23.261	< 2e-16 ***
snr	0.39734	0.05001	7.945	1.94e-15 ***

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 65.9045 on 3 degrees of freedom  
Residual deviance: 2.8089 on 2 degrees of freedom  
AIC: 27.061

Number of Fisher Scoring iterations: 4

```
> confint(result.2)
Waiting for profiling to be done...
                2.5 %      97.5 %
(Intercept) -4.2072190 -3.5544117
snr          0.2999362  0.4963887
```

```
> #Binomial(probit) probit regression
> result.3 <- glm(pyes ~ snr, data=d1, weights=subtotal,
+               family=binomial(link="probit"))
```

```
> summary(result.3)
```

Call:  
glm(formula = pyes ~ snr, family = binomial(link = "probit"),  
data = d1, weights = subtotal)

Deviance Residuals:

1	2	3	4
-0.6188	1.0388	0.1684	-0.6175

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-2.06055	0.07017	-29.367	< 2e-16 ***
snr	0.18777	0.02348	7.997	1.28e-15 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

Null deviance: 65.9045 on 3 degrees of freedom  
 Residual deviance: 1.8716 on 2 degrees of freedom  
 AIC: 26.124

Number of Fisher Scoring iterations: 4

> confint(result.3)

Waiting for profiling to be done...

	2.5 %	97.5 %
(Intercept)	-2.2026985	-1.9262643
snr	0.1416397	0.2343393

## 多項ロジスティック回帰分析

```
library(VGAM)
```

```
オブジェクト名 <- vglm(基準変数 ~ 説明変数, family=multinomial, data=d1)
```

```
summary(オブジェクト名)
```

あらかじめVGAMパッケージをインストールしておく必要がある。  
vglm は glm を含む、より一般化された分析モデルである。

```
> setwd("d:¥¥¥")
> d1 <- read.table("多項ロジスティック回帰分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  kyoufu support stress result
1      29      16      28      1
2      27      19      17      2
3      27      16      21      2
4      26      23      22      2
5      26      22      23      3
6      26      20      25      2
```

	A	B	C	D
1	kyoufu	support	stress	result
2	29	16	28	1
3	27	19	17	2
4	27	16	21	2
5	26	23	22	2
6	26	22	23	3
7	26	20	25	2
8	26	20	21	1
9	26	19	25	1
10	26	16	26	1
11	25	32	24	3
12	25	15	21	1
13	25	13	29	1
14	25	12	26	1
15	24	21	20	2
16	24	11	28	1
17	23	31	26	3
18	23	29	20	3
19	23	28	23	2
20	23	21	24	1
21	23	14	29	1

```
> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD kyoufu support stress result
kyoufu 245 15.22 5.37  1.00  -0.10  0.43  -0.21
support 245 20.32 6.08 -0.10   1.00 -0.32  0.48
stress  245 18.52 5.12  0.43  -0.32  1.00 -0.39
result  245  0.55 0.50 -0.21   0.48 -0.39  1.00
```

```
> # 偏回帰係数の推定と分析結果の表示
> # VGAMパッケージの読み込み
> library(VGAM)
> result.1 <- vglm(result ~ kyoufu + support + stress,
+                  family=multinomial, data=d1)
```

```
> summary(result.1)
```

Call:

```
vglm(formula = result ~ kyoufu + support + stress, family = multinomial,
     data = d1)
```

Coefficients:

```
      Estimate Std. Error z value Pr(>|z|)
(Intercept):1  0.641363   1.204523  0.532  0.5944
(Intercept):2  1.812117   1.021766  1.774  0.0761 .
kyoufu:1       0.052284   0.040671  1.286  0.1986
kyoufu:2      -0.009506   0.035416 -0.268  0.7884
support:1     -0.243084   0.040026 -6.073 1.25e-09 ***
support:2     -0.131794   0.032755 -4.024 5.73e-05 ***
stress:1       0.189729   0.048025  3.951 7.80e-05 ***
stress:2       0.080441   0.039218  2.051  0.0403 *
---
```

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Residual deviance: 440.6041 on 482 degrees of freedom

Log-likelihood: -220.302 on 482 degrees of freedom

Number of Fisher scoring iterations: 5

No Hauck-Donner effect found in any of the estimates

Reference group is level 3 of the response

> #標準偏回帰係数の推定

> d2 <- as.data.frame(scale(d1[,c("kyoufu", "support", "stress")]))

> d2 <- data.frame(d2, d1\$result)

> colnames(d2) <- c("kyoufu", "support", "stress", "result")

> head(d2)

	kyoufu	support	stress	result
1	2.568375	-0.70988932	1.8521250	1
2	2.195706	-0.21672425	-0.2975205	2
3	2.195706	-0.70988932	0.4841688	2
4	2.009371	0.44082919	0.6795911	2
5	2.009371	0.27644083	0.8750134	3
6	2.009371	-0.05233589	1.2658581	2

> #共分散行列の確認

> cov(d2)

	kyoufu	support	stress	result
kyoufu	1.0000000	-0.1034273	0.4255222	-0.1850753
support	-0.1034273	1.0000000	-0.3231937	0.3927501
stress	0.4255222	-0.3231937	1.0000000	-0.3303219
result	-0.1850753	0.3927501	-0.3303219	0.6432921

> result.2 <- vglm(result ~ kyoufu + support + stress,  
+ family=multinomial, data=d2)

> summary(result.2)

Call:

vglm(formula = result ~ kyoufu + support + stress, family = multinomial,  
data = d2)

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept):1	0.01211	0.21474	0.056	0.9550
(Intercept):2	0.47959	0.18733	2.560	0.0105 *
kyoufu:1	0.28059	0.21827	1.286	0.1986
kyoufu:2	-0.05102	0.19007	-0.268	0.7884
support:1	-1.47872	0.24348	-6.073	1.25e-09 ***
support:2	-0.80172	0.19925	-4.024	5.73e-05 ***
stress:1	0.97087	0.24575	3.951	7.80e-05 ***
stress:2	0.41163	0.20068	2.051	0.0403 *

---  
Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Residual deviance: 440.6041 on 482 degrees of freedom

Log-likelihood: -220.302 on 482 degrees of freedom

Number of Fisher scoring iterations: 5

No Hauck-Donner effect found in any of the estimates

Reference group is level 3 of the response

>

## 対数線形モデル — glm関数

## データ形式（一覧データ（フラットなクロス表））

テーブル名 <- ftable(データフレーム名[, c("変数名1", "変数名2", ...)], row.vars=c("変数名1", "変数名2", ...))

各カテゴリ変数の水準の組み合わせに対する度数の一覧表「sex=F, method=1, grade=A」の度数 = 7 など

sex	method	grade	Freq
F	1	A	7
F	1	B	11
F	1	C	2
F	2	A	10
F	2	B	4
F	2	C	4
M	1	A	8
M	1	B	8
M	1	C	7
M	2	A	13
M	2	B	7
M	2	C	16

## モデルの指定法

glmオブジェクト名 <- glm(度数変数 ~ 説明変数1 + 説明変数2 + ..., family=poisson, データフレーム名)

## 交互作用の指定法

説明変数名1 : 説明変数名2      # : 区切り ... 交互作用のみ  
説明変数名1 \* 説明変数名2      # \* 区切り ... 主効果も含んだ交互作用

## パラメタ推定値の表示

summary(glmオブジェクト名)

## ステップワイズ分析

library(MASS)

AICオブジェクト名 <- stepAIC(glmオブジェクト名)

summary(AICオブジェクト名)

MASSパッケージは最初からインストールされている

## 要因の主効果，交互作用の検討

library(car)

Anova(glmオブジェクト名)

あらかじめ car パッケージをインストールしておく必要がある。  
Anova は，anova とは異なる関数であることに注意。

## 残差分析

オブジェクト名 <- glm(度数変数~見たい効果以下の効果変数名の指定, family=poisson, データフレーム名)

## モデルによる予測値

xtabs(fitted.values(オブジェクト名) ~ 変数名の指定)

## 残差・標準化残差（ピアソン）

xtabs(residuals(オブジェクト名, type="pearson") ~ 変数名の指定)

xtabs(rstandard(オブジェクト名, type="pearson") ~ 変数名の指定)

標準化残差の絶対値が大きいセルが，見たい効果が影響しているセルとなる。  
デビアンズ残差を得るには，type="deviance" とするか，typeの指定を省略する。



```
> setwd("d:¥¥")
> d1 <- read.table("対数線形モデル_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id sex method grade
1  1  F      1      B
2  2  M      1      B
3  3  F      2      A
4  4  M      2      C
5  5  M      2      C
6  6  M      3      A
>
```

### > #クロス表

```
> (t1 <- table(d1[, c("method", "grade", "sex")], dnn=list("method", "grade", "sex")))
, , sex = F
```

```
      grade
method A  B  C
1      7 11  2
2     10  4  4
3      3  3  2
```

```
, , sex = M
```

```
      grade
method A  B  C
1      8  8  7
2     13  7 16
3      6  5  6
```

### > #一覧データ

```
> ft1 <- fttable(d1[, c("sex", "method", "grade")], row.vars=c("sex", "method", "grade"))
> (d2 <- as.data.frame(ft1))
```

```
  sex method grade Freq
1  F      1      A     7
2  M      1      A     8
3  F      2      A    10
4  M      2      A    13
5  F      3      A     3
6  M      3      A     6
7  F      1      B    11
8  M      1      B     8
9  F      2      B     4
10 M      2      B     7
11 F      3      B     3
12 M      3      B     5
13 F      1      C     2
14 M      1      C     7
15 F      2      C     4
16 M      2      C    16
17 F      3      C     2
18 M      3      C     6
>
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	M	2	C
6	5	M	2	C
7	6	M	3	A
8	7	M	2	C
9	8	M	2	A
10	9	M	2	A
11	10	M	2	A
12	11	F	2	A
13	12	F	2	B
14	13	M	2	B
15	14	M	1	A
16	15	M	2	C
17	16	M	3	A
18	17	M	1	B
19	18	M	3	C
20	19	F	2	A
21	20	M	2	C

### > #飽和モデル

```
> result.full <- glm(Freq ~ method + grade + sex
+                      + sex:method + sex:grade + method:grade
+                      + sex:method:grade, family=poisson, d2)
```

```
> summary(result.full)
```

Call:

```
glm(formula = Freq ~ method + grade + sex + sex:method + sex:grade +
  method:grade + sex:method:grade, family = poisson, data = d2)
```

Deviance Residuals:

[1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	1.945910	0.377964	5.148	2.63e-07 ***
method2	0.356675	0.492805	0.724	0.4692
method3	-0.847298	0.690066	-1.228	0.2195
gradeB	0.451985	0.483494	0.935	0.3499
gradeC	-1.252763	0.801784	-1.562	0.1182
sexM	0.133531	0.517549	0.258	0.7964
method2:sexM	0.128833	0.666918	0.193	0.8468
method3:sexM	0.559616	0.876275	0.639	0.5231
gradeB:sexM	-0.451985	0.695533	-0.650	0.5158
gradeC:sexM	1.119232	0.954313	1.173	0.2409
method2:gradeB	-1.368276	0.764046	-1.791	0.0733 .
method3:gradeB	-0.451985	0.948911	-0.476	0.6338
method2:gradeC	0.336472	0.996422	0.338	0.7356
method3:gradeC	0.847298	1.214985	0.697	0.4856
method2:gradeB:sexM	0.749237	1.026424	0.730	0.4654
method3:gradeB:sexM	0.269664	1.231706	0.219	0.8267
method2:gradeC:sexM	0.004699	1.183274	0.004	0.9968
method3:gradeC:sexM	-0.713766	1.441312	-0.495	0.6204

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 3.5563e+01 on 17 degrees of freedom  
 Residual deviance: -2.2013e-25 on 0 degrees of freedom  
 AIC: 101.28

Number of Fisher Scoring iterations: 3

```
> # 要因の主効果, 交互作用の検討
> library(car)
> Anova(result.full)
```

Analysis of Deviance Table (Type II tests)

Response: Freq

	LR	Chisq	Df	Pr(>Chisq)
method	11.0975	2	0.003892	**
grade	1.4586	2	0.482244	
sex	7.4533	1	0.006332	**
method:sex	1.2316	2	0.540205	
grade:sex	5.4028	2	0.067110	.
method:grade	6.0188	4	0.197751	
method:grade:sex	0.9904	4	0.911255	

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

&gt;

```
> #ステップワイズ分析
> #MASSパッケージの読み込み
> library(MASS)
> result.aic <- stepAIC(result.full)
```

Start: AIC=101.28

```
Freq ~ method + grade + sex + sex:method + sex:grade + method:grade +
      sex:method:grade
```

	Df	Deviance	AIC
- method:grade:sex	4	0.99035	94.272
<none>		0.00000	101.282

Step: AIC=94.27

```
Freq ~ method + grade + sex + method:sex + grade:sex + method:grade
```

```

      Df Deviance   AIC
- method:sex    2  2.2220 91.504
- method:grade  4   7.0091 92.291
<none>          0  0.9904 94.272
- grade:sex     2   6.3932 95.675

```

```

Step:  AIC=91.5
Freq ~ method + grade + sex + grade:sex + method:grade

```

```

      Df Deviance   AIC
- method:grade  4   9.1956 90.477
<none>          2  2.2220 91.504
- grade:sex     2   8.5797 93.861

```

```

Step:  AIC=90.48
Freq ~ method + grade + sex + grade:sex

```

```

      Df Deviance   AIC
<none>          2   9.1956 90.477
- grade:sex     2  15.5533 92.835
- method       2  20.2931 97.575

```

```
> summary(result.aic)
```

```

Call:
glm(formula = Freq ~ method + grade + sex + grade:sex, family = poisson,
     data = d2)

```

```

Deviance Residuals:
      Min       1Q   Median       3Q      Max
-1.5563  -0.5128   0.1097   0.3377   1.6722

```

```

Coefficients:
      Estimate Std. Error z value Pr(>|z|)
(Intercept)   1.9529    0.2551   7.656 1.91e-14 ***
method2        0.2278    0.2044   1.114  0.2651
method3       -0.5423    0.2515  -2.156  0.0311 *
gradeB        -0.1054    0.3249  -0.324  0.7457
gradeC        -0.9163    0.4183  -2.190  0.0285 *
sexM           0.3001    0.2950   1.017  0.3090
gradeB:sexM   -0.1947    0.4389  -0.444  0.6572
gradeC:sexM    0.9877    0.4965   1.989  0.0467 *
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)
```

```

Null deviance: 35.5627 on 17 degrees of freedom
Residual deviance: 9.1956 on 10 degrees of freedom
AIC: 90.477

```

```
Number of Fisher Scoring iterations: 4
```

```
> # ステップワイズ分析後の要因の主効果, 交互作用の検討
```

```
> Anova(result.aic)
Analysis of Deviance Table (Type II tests)
```

```

Response: Freq
      LR Chisq Df Pr(>Chisq)
method    11.0975  2  0.003892 **
grade     1.4586  2  0.482244
sex       7.4533  1  0.006332 **
grade:sex  6.3577  2  0.041634 *
---

```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
> # 主効果モデル
```

```
> result.1 <- glm(Freq ~ method + grade + sex,
+                 family=poisson, d2)
> summary(result.1)
```

```
Call:
```

```
glm(formula = Freq ~ method + grade + sex, family = poisson,
    data = d2)
```

```
Deviance Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.49529  -0.69830  -0.05981   0.23877   2.28630
```

```
Coefficients:
```

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.8319     0.2233   8.203 2.35e-16 ***
method2         0.2278     0.2044   1.114 0.26507
method3        -0.5423     0.2515  -2.156 0.03106 *
gradeB         -0.2126     0.2182  -0.974 0.32988
gradeC         -0.2392     0.2198  -1.088 0.27637
sexM            0.5021     0.1868   2.688 0.00719 **
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for poisson family taken to be 1)
```

```
Null deviance: 35.563 on 17 degrees of freedom
Residual deviance: 15.553 on 12 degrees of freedom
AIC: 92.835
```

```
Number of Fisher Scoring iterations: 4
```

```
> Anova(result.1)
```

```
Analysis of Deviance Table (Type II tests)
```

```
Response: Freq
```

```
      LR Chisq Df Pr(>Chisq)
method 11.0975  2  0.003892 **
grade   1.4586  2  0.482244
sex      7.4533  1  0.006332 **
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
>
```

```
> # sex:grade の交互作用を見るための残差分析
```

```
> #主効果とsex:method, method:gradeの交互作用までを仮定したモデル
```

```
> result.2 <- glm(Freq ~ sex + method + grade + sex:method + method:grade, family=poisson, d2)
> summary(result.2)
```

```
Call:
```

```
glm(formula = Freq ~ sex + method + grade + sex:method + method:grade,
    family = poisson, data = d2)
```

```
Deviance Residuals:
```

```
      Min       1Q   Median       3Q      Max
-1.19065  -0.32089   0.00029   0.25977   0.93241
```

```
Coefficients:
```

```
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.9426     0.3056   6.356 2.07e-10 ***
sexM            0.1398     0.3057   0.457 0.6476
method2         0.0943     0.4170   0.226 0.8211
method3        -0.8848     0.5381  -1.644 0.1001
gradeB          0.2364     0.3454   0.684 0.4937
gradeC         -0.5108     0.4216  -1.212 0.2257
sexM:method2     0.5534     0.4205   1.316 0.1882
sexM:method3     0.6140     0.5266   1.166 0.2436
```

```
method2:gradeB -0.9740      0.5037 -1.934  0.0531 .
method3:gradeB -0.3542      0.5962 -0.594  0.5525
method2:gradeC  0.3711      0.5208  0.712  0.4762
method3:gradeC  0.3930      0.6433  0.611  0.5412
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

(Dispersion parameter for poisson family taken to be 1)

```
Null deviance: 35.5627 on 17 degrees of freedom
Residual deviance: 6.3932 on 6 degrees of freedom
AIC: 95.675
```

Number of Fisher Scoring iterations: 4

> # 予測値

```
> xtabs(fitted.values(result.2) ~ sex + grade + method, data=d2)
, , method = 1
```

```
      grade
sex      A      B      C
F  6.976744  8.837209  4.186047
M  8.023256 10.162791  4.813953
```

```
, , method = 2
```

```
      grade
sex      A      B      C
F  7.666667  3.666667  6.666667
M 15.333333  7.333333 13.333333
```

```
, , method = 3
```

```
      grade
sex      A      B      C
F  2.880000  2.560000  2.560000
M  6.120000  5.440000  5.440000
```

> # 残差 (ピアソン)

```
> xtabs(residuals(result.2, type="pearson") ~ sex + grade + method, data=d2)
, , method = 1
```

```
      grade
sex      A      B      C
F  0.008804509  0.727540078 -1.068457799
M -0.008210247 -0.678434621  0.996342033
```

```
, , method = 2
```

```
      grade
sex      A      B      C
F  0.842700972  0.174077656 -1.032795559
M -0.595879572 -0.123091491  0.730296743
```

```
, , method = 3
```

```
      grade
sex      A      B      C
F  0.070710678  0.275000000 -0.350000000
M -0.048507125 -0.188648444  0.240098019
```

> # 標準化残差 (ピアソン)

```
> xtabs(rstandard(result.2, type="pearson") ~ sex + grade + method, data=d2)
, , method = 1
```

```
      grade
sex      A      B      C
F  0.01491867  1.33154217 -1.64294963
M -0.01491865 -1.33154032  1.64294735
```

```
, , method = 2

      grade
sex      A      B      C
F  1.36218225  0.23891939 -1.59410745
M -1.36218123 -0.23891921  1.59410626

, , method = 3

      grade
sex      A      B      C
F  0.10718662  0.40441176 -0.51470588
M -0.10718662 -0.40441176  0.51470588

> xtabs(rstandard(result.2, type="pearson") ~ sex + grade, data=d2)
      grade
sex      A      B      C
F  1.484288  1.974873 -3.751763
M -1.484286 -1.974871  3.751759
>
> xtabs(residuals(result.2, type="pearson") ~ sex + method, data=d2)
      method
sex      1      2      3
F -0.332113212 -0.016016931 -0.004289322
M  0.309697165  0.011325681  0.002942451
>
> xtabs(residuals(result.2, type="pearson") ~ grade + method, data=d2)
      method
grade      1      2      3
A  0.000594262  0.246821400  0.022203553
B  0.049105457  0.050986165  0.086351556
C -0.072115766 -0.302498816 -0.109901981

> # 標準化残差（デビアンズ）
> xtabs(rstandard(result.2, type="deviance") ~ sex + grade + method, data=d2)
, , method = 1

      grade
sex      A      B      C
F  0.01491039  1.28211793 -1.83084729
M -0.01492587 -1.38352865  1.53752401

, , method = 2

      grade
sex      A      B      C
F  1.30060957  0.23543000 -1.72341213
M -1.39912435 -0.24076438  1.54496236

, , method = 3

      grade
sex      A      B      C
F  0.10645490  0.39358552 -0.53542276
M -0.10753979 -0.41005579  0.50623366

> xtabs(rstandard(result.2) ~ sex + grade, data=d2)
      grade
sex      A      B      C
F  1.421975  1.911133 -4.089682
M -1.521590 -2.034349  3.588720
```

## 対数線形モデル — loglm関数

データ形式（テーブルデータ または 一覧データ）

テーブルデータ（クロス表）

テーブル名 &lt;- table(データフレーム名[, c("変数名1", "変数名2", ...)], dnn=c("変数名1", "変数名2", ...))

一覧データ（フラットなクロス表）

テーブル名 &lt;- ftable(データフレーム名[, c("変数名1", "変数名2", ...)], row.vars=c("変数名1", "変数名2", ...))

各カテゴリ変数の水準の組み合わせに対する度数の一覧表「sex=F, method=1, grade=A」の度数 = 7 など

sex	method	grade	Freq
F	1	A	7
F	1	B	11
F	1	C	2
F	2	A	10
F	2	B	4
F	2	C	4
M	1	A	8
M	1	B	8
M	1	C	7
M	2	A	13
M	2	B	7
M	2	C	16

## モデルの指定法

テーブルデータを使う場合

library(MASS)

オブジェクト名 &lt;- loglm( ~ 変数1 + 変数2 + 変数1:変数2 + ..., テーブル名)

一覧データを使う場合

library(MASS)

オブジェクト名 &lt;- loglm(度数変数 ~ 説明変数1 + 説明変数2 + 変数1:変数2 + ..., データフレーム名)

MASSパッケージは最初からインストールされている

## 交互作用の指定法

説明変数名1 : 説明変数名2 # : 区切り ... 交互作用のみ

説明変数名1 \* 説明変数名2 # \* 区切り ... 主効果も含んだ交互作用

## パラメタ推定値の表示

coef(オブジェクト名)

## ステップワイズ分析

library(MASS)

AICオブジェクト名 &lt;- stepAIC(loglmオブジェクト名)

coef(AICオブジェクト名)

## 尤度比検定

anova(loglmオブジェクト名1, loglmオブジェクト名2)

AICオブジェクト名も設定可能

一覧データを用いた場合は、モデルの記述に不具合が生じるようだが、計算は正しくしてくれる

## 残差分析

テーブルデータを使う場合

オブジェクト名 &lt;- loglm(~変数名の指定, テーブル名)

一覧データを使う場合

オブジェクト名 &lt;- loglm(度数変数~変数名の指定, データフレーム名)

## 観測値のモデル予測からのデビアンズ残差の表

apply(residuals(オブジェクト名), c(配列の次元を指定), sum)

```
> setwd("d:¥¥")
> d1 <- read.table("対数線形モデル_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id sex method grade
1  1  F      1      B
2  2  M      1      B
3  3  F      2      A
4  4  M      2      C
5  5  M      2      C
6  6  M      3      A
>
```

```
> #クロス表
```

```
> (t1 <- table(d1[,c("method", "grade", "sex")], dnn=list("method", "grade", "sex")))
, , sex = F
```

```
      grade
method A  B  C
1      7 11  2
2     10  4  4
3      3  3  2
```

```
, , sex = M
```

```
      grade
method A  B  C
1      8  8  7
2     13  7 16
3      6  5  6
```

```
>
> #一覧データ
> ft1 <- ftable(d1[,c("sex", "method", "grade")], row.vars=c("sex", "method", "grade"))
> (d2 <- as.data.frame(ft1))
```

```
  sex method grade Freq
1   F      1      A     7
2   M      1      A     8
3   F      2      A    10
4   M      2      A    13
5   F      3      A     3
6   M      3      A     6
7   F      1      B    11
8   M      1      B     8
9   F      2      B     4
10  M      2      B     7
11  F      3      B     3
12  M      3      B     5
13  F      1      C     2
14  M      1      C     7
15  F      2      C     4
16  M      2      C    16
17  F      3      C     2
18  M      3      C     6
```

	A	B	C	D
1	id	sex	method	grade
2	1	F	1	B
3	2	M	1	B
4	3	F	2	A
5	4	M	2	C
6	5	M	2	C
7	6	M	3	A
8	7	M	2	C
9	8	M	2	A
10	9	M	2	A
11	10	M	2	A
12	11	F	2	A
13	12	F	2	B
14	13	M	2	B
15	14	M	1	A
16	15	M	2	C
17	16	M	3	A
18	17	M	1	B
19	18	M	3	C
20	19	F	2	A
21	20	M	2	C

```
> #MASSパッケージの読み込み
```

```
> library(MASS)
```

```
> # 飽和モデル
```

```
> result.full <- loglm(~ method + grade + sex
+                      + sex:method + sex:grade + method:grade
+                      + sex:method:grade, t1)
```

```
> # または
```

```
> # result.full <- loglm(Freq~ method + grade + sex
+                      + sex:method + sex:grade + method:grade
+                      + sex:method:grade, d2)
```

```
> coef(result.full)
$(Intercept)
[1] 1.75465
```



```

$method
      1      2      3
0.1023078 0.3051205 -0.4074284

$grade
      A      B      C
0.209226512 -0.001717884 -0.207508628

$sex
      F      M
-0.3099278 0.3099278

$method.grade
      grade
method      A      B      C
  1 -0.05350831 0.38342865 -0.3299203
  2  0.16477037 -0.39195020 0.2271798
  3 -0.11126207 0.00852155 0.1027405

$method.sex
      sex
method      F      M
  1  0.13195435 -0.13195435
  2 -0.05811795 0.05811795
  3 -0.07383640 0.07383640

$grade.sex
      sex
grade      F      M
  A  0.1284206 -0.1284206
  B  0.1845965 -0.1845965
  C -0.3130172 0.3130172

$method.grade.sex
, , sex = F

      grade
method      A      B      C
  1 -0.01721290 0.15260380 -0.13539089
  2  0.10844296 -0.09635866 -0.01208429
  3 -0.09123005 -0.05624513 0.14747519

, , sex = M

      grade
method      A      B      C
  1  0.01721290 -0.15260380 0.13539089
  2 -0.10844296 0.09635866 0.01208429
  3  0.09123005 0.05624513 -0.14747519

> # 飽和モデルの残差
> residuals(result.full)
Re-fitting to get frequencies and fitted values
, , sex = F

      grade
method A B C
  1 0 0 0
  2 0 0 0
  3 0 0 0

, , sex = M

      grade
method A B C
  1 0 0 0
  2 0 0 0
  3 0 0 0
>

```

```

>
> # ステップワイズ分析
> (result.aic <- stepAIC(result.full))

Start:  AIC=36
~method + grade + sex + sex:method + sex:grade + method:grade +
  sex:method:grade

              Df    AIC
- method:grade:sex  4 28.99
<none>              36.00

Step:  AIC=28.99
~method + grade + sex + method:sex + grade:sex + method:grade

              Df    AIC
- method:sex      2 26.222
- method:grade    4 27.009
<none>            28.990
- grade:sex       2 30.393

Step:  AIC=26.22
~method + grade + sex + grade:sex + method:grade

              Df    AIC
- method:grade    4 25.196
<none>            26.222
- grade:sex       2 28.580

Step:  AIC=25.2
~method + grade + sex + grade:sex

              Df    AIC
<none>          25.196
- grade:sex     2 27.553
- method        2 32.293
Call:
loglm(formula = ~method + grade + sex + grade:sex, data = tl,
      evaluate = FALSE)

Statistics:
              X^2 df  P(> X^2)
Likelihood Ratio 9.195595 10 0.5136471
Pearson          9.224254 10 0.5109616

> coef(result.aic)

$(Intercept)`
[1] 1.789734

$method
      1      2      3
0.1048468 0.3326307 -0.4374775

$grade
      A      B      C
0.208382813 0.005650259 -0.214033071

$sex
      F      M
-0.2822199 0.2822199

$grade.sex
      sex
grade
  A 0.1321676 -0.1321676
  B 0.2295396 -0.2295396
  C -0.3617072 0.3617072

```

```

>
> # 主効果モデル
> result.1 <- loglm(~ method + grade + sex, t1)
> # または
> #result.1 <- loglm(Freq ~ method + grade + sex, d2)
>
>
> # sex:grade の交互作用効果を見るための残差分析
> #sex:method, method:gradeの交互作用までを仮定したモデル
> result.2 <- loglm(~ method + grade + sex
+                   + sex:method + method:grade, t1)
>
> # 尤度比検定
> anova(result.1, result.2)
LR tests for hierarchical log-linear models

Model 1:
~method + grade + sex
Model 2:
Freq ~ grade + method + sex + grade:sex

      Deviance df Delta(Dev) Delta(df) P(> Delta(Dev)
Model 1   15.553284 12
Model 2    6.393183  6   9.160101         6      0.16477
Saturated  0.000000  0   6.393183         6      0.38062
>
>
> # sex:grade の交互作用を見るための残差分析
> #主効果とsex:method, method:gradeの交互作用までを仮定したモデル
>
> result.2 <- loglm(~ sex + method + grade + sex:method + method:grade, t1)
> # または
> #result.2 <- loglm(Freq~ method + grade + sex + sex:method + method:grade, d2)
>
> coef(result.2)

$(Intercept)`
[1] 1.78938

$method
      1      2      3
0.1316042 0.3016217 -0.4332260

$grade
      A      B      C
0.15415153 -0.05217953 -0.10197200

$sex
      F      M
-0.2644468 0.2644468

$method.grade
      grade
method      A      B      C
1 -0.06267259 0.38004726 -0.31737467
2  0.13830209 -0.39296578 0.25466369
3 -0.07562951 0.01291852 0.06271099

$method.sex
      sex
method      F      M
1  0.19456585 -0.19456585
2 -0.08212677 0.08212677
3 -0.11243908 0.11243908
>

```

```

> #予測値
> fitted.values(result.2)
Re-fitting to get fitted values
, , sex = F

      grade
method  A      B      C
  1 6.976744 8.837209 4.186047
  2 7.666667 3.666667 6.666667
  3 2.880000 2.560000 2.560000

, , sex = M

      grade
method  A      B      C
  1 8.023256 10.162791 4.813953
  2 15.333333 7.333333 13.333333
  3 6.120000 5.440000 5.440000

>
>
> #ピアソン残差
> residuals(result.2, type="pearson")

Re-fitting to get frequencies and fitted values
, , sex = F

      grade
method  A      B      C
  1 0.008804509 0.727540078 -1.068457799
  2 0.842700972 0.174077656 -1.032795559
  3 0.070710678 0.275000000 -0.350000000

, , sex = M

      grade
method  A      B      C
  1 -0.008210247 -0.678434621 0.996342035
  2 -0.595879572 -0.123091491 0.730296743
  3 -0.048507125 -0.188648444 0.240098019

>
> apply(residuals(result.2, type="pearson"), c("sex", "grade"), sum)

Re-fitting to get frequencies and fitted values
      grade
sex      A      B      C
F 0.9222162 1.1766177 -2.451253
M -0.6525969 -0.9901746 1.966737
>
>
>
>
> #デビアンズ残差
> residuals(result.2)
Re-fitting to get frequencies and fitted values
, , sex = F

      grade
method  A      B      C
  1 0.008799624 0.700535218 -1.190653100
  2 0.804609625 0.171535271 -1.116569901
  3 0.070227965 0.267638153 -0.364087473

, , sex = M

      grade
method  A      B      C

```

```

1 -0.008214218 -0.704923252 0.932409555
2 -0.612040160 -0.124042121 0.707782794
3 -0.048666954 -0.191281248 0.236145930

```

```
> apply(residuals(result.2), c("sex", "grade"), sum)
```

Re-fitting to get frequencies and fitted values

```

      grade
sex      A      B      C
F  0.8836372  1.139709 -2.671310
M -0.6689213 -1.020247  1.876338

```

## カウントデータの分析

## ポアソンモデル

```
オブジェクト名 <- glm(計数変数 ~ 独立変数 + offset(log(全体数変数)), data=データフレーム名,
  family=poisson(link="log"))
```

## 負の二項分布モデル

```
library(MASS)
```

```
オブジェクト名 <- glm.nb(計数変数 ~ 独立変数 + offset(log(全体数変数)), data=データフレーム名,
  link=log)
```

```
summary(オブジェクト名)
```

```
confint(オブジェクト名)
```

```
> # Data from British department of transport.
```

```
> # Agresti, A. (2007). An introduction to categorical data analysis, 2nd ed. Wiley, p83.
```

```
> setwd("i:¥¥Rdocuments¥¥programs¥¥")
```

```
>
> d1 <- read.table("collisionsData.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
```

```
> d1$x <- d1$year - 1975
```

```
> head(d1)
```

```
  year trainKm collTT collTR  x
1 2003    518      0      3 28
2 2002    516      1      3 27
3 2001    508      0      4 26
4 2000    503      1      3 25
5 1999    505      1      2 24
6 1998    487      0      4 23
```

```
>
```

## &gt; # 記述統計量

```
> dtmp <- d1
```

```
> ntmp <- nrow(dtmp)
```

```
> mtmp <- colMeans(dtmp)
```

```
> stmp <- apply(dtmp, 2, sd)
```

```
> ctmp <- cor(dtmp)
```

```
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
```

```
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
```

```
> ktmp
```

```
      N      Mean      SD  year trainKm collTT collTR      x
year   29 1989.00  8.51  1.00   0.72 -0.56 -0.34  1.00
trainKm 29 440.00 38.68  0.72   1.00 -0.44 -0.26  0.72
collTT  29   1.66  1.32 -0.56 -0.44  1.00 -0.03 -0.56
collTR  29   4.21  2.81 -0.34 -0.26 -0.03  1.00 -0.34
x       29  14.00  8.51  1.00   0.72 -0.56 -0.34  1.00
```

```
>
```

## &gt; # GLM

```
>
```

## &gt; #Poisson loglinear

```
> result.1 <- glm(collTR ~ x + offset(log(trainKm)), data=d1,
+   family=poisson(link="log"))
```

```
> summary(result.1)
```

```
Call:
```

```
glm(formula = collTR ~ x + offset(log(trainKm)), family = poisson(link = "log"),
  data = d1)
```

```
Deviance Residuals:
```

```
      Min       1Q   Median       3Q      Max
-2.0580  -0.7825  -0.0826   0.3775   3.3873
```

```
Coefficients:
```

```
      Estimate Std. Error z value Pr(>|z|)
(Intercept) -4.21142    0.15892  -26.50  < 2e-16 ***
x            -0.03292    0.01076   -3.06  0.00222 **
```

```
---
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

	A	B	C	D
1	year	trainKm	collTT	collTR
2	2003	518	0	3
3	2002	516	1	3
4	2001	508	0	4
5	2000	503	1	3
6	1999	505	1	2
7	1998	487	0	4
8	1997	463	1	1
9	1996	437	2	2
10	1995	423	1	2
11	1994	415	2	4
12	1993	425	0	4
13	1992	430	1	4
14	1991	439	2	6
15	1990	431	1	2
16	1989	436	4	4
17	1988	443	2	4
18	1987	397	1	6
19	1986	414	2	13
20	1985	418	0	5
21	1984	389	5	3
22	1983	401	2	7
23	1982	372	2	3
24	1981	417	2	2
25	1980	430	2	2
26	1979	426	3	3
27	1978	430	2	4
28	1977	425	1	8
29	1976	426	2	12
30	1975	436	5	2

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 47.376 on 28 degrees of freedom  
Residual deviance: 37.853 on 27 degrees of freedom  
AIC: 133.52

Number of Fisher Scoring iterations: 5

```
> confint(result.1)
Waiting for profiling to be done...
              2.5 %      97.5 %
(Intercept) -4.5342219 -3.9105409
x            -0.0542105 -0.0119652
>
>
```

```
> # Negative binomial loglinear
> library(MASS)
> result.2 <- glm.nb(collTR ~ x + offset(log(trainKm)), data=d1,
+                   link=log)
> summary(result.2)
```

Call:  
glm.nb(formula = collTR ~ x + offset(log(trainKm)), data = d1,  
link = log, init.theta = 10.11828724)

Deviance Residuals:

	Min	1Q	Median	3Q	Max
	-1.72370	-0.65461	-0.05868	0.32984	2.64065

Coefficients:

	Estimate	Std. Error	z value	Pr(> z )
(Intercept)	-4.19999	0.19584	-21.446	< 2e-16 ***
x	-0.03367	0.01288	-2.615	0.00893 **

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for Negative Binomial(10.1183) family taken to be 1)

Null deviance: 32.045 on 28 degrees of freedom  
Residual deviance: 25.264 on 27 degrees of freedom  
AIC: 132.69

Number of Fisher Scoring iterations: 1

Theta: 10.12  
Std. Err.: 8.00

2 x log-likelihood: -126.69

```
> confint(result.2)
Waiting for profiling to be done...
              2.5 %      97.5 %
(Intercept) -4.593077 -3.815438383
x            -0.059499 -0.008290411
>
```

## 19 階層線形モデル (マルチレベルモデル)



## 階層線形モデルの例

## ランダム切片モデル

レベル1の回帰式（基準変数の回帰式）

$$y = b_0 + b_1 * support + e$$

# 説明変数は、群ごとに平均偏差化しておくことが多い

レベル2の回帰式（回帰係数の回帰式）

$$b_0 = g_{00} + g_{01} * rooms + g_{02} * futoukou + u_0$$

# ランダム切片

回帰式の統合

$$\begin{aligned} y &= (g_{00} + g_{01} * rooms + g_{02} * futoukou + u_0) + b_1 * support + e \\ &= (g_{00} + g_{01} * rooms + g_{02} * futoukou + b_1 * support) + (u_0 + e) \end{aligned}$$

固定効果

g00 ... (Intercept)  
g01 ... rooms  
g02 ... futoukou  
b1 ... support

変量効果

u0 ... (Intercept)  
e ... (Residual)

## ランダム切片・ランダムスロープモデル 1

レベル1の回帰式（基準変数の回帰式）

$$y = b_0 + b_1 * support + e$$

レベル2の回帰式（回帰係数の回帰式）

$$\begin{aligned} b_0 &= g_{00} + g_{01} * rooms + g_{02} * futoukou + u_0 \\ b_1 &= g_{10} + u_1 \end{aligned}$$

# ランダム切片

# ランダムスロープ

回帰式の統合

$$\begin{aligned} y &= (g_{00} + g_{01} * rooms + g_{02} * futoukou + u_0) + \\ &\quad (g_{10} + u_1) * support + e \\ &= (g_{00} + g_{01} * rooms + g_{02} * futoukou + g_{10} * support) + (u_0 + u_1 * support + e) \end{aligned}$$

固定効果

g00 ... (Intercept)  
g01 ... rooms  
g02 ... futoukou  
g10 ... support

変量効果

u0 ... (Intercept)  
u1 ... support  
e ... (Residual)

## ランダム切片・ランダムスロープモデル 2

レベル1の回帰式（基準変数の回帰式）

$$y = b_0 + b_1 * support + e$$

レベル2の回帰式（回帰係数の回帰式）

$$\begin{aligned} b_0 &= g_{00} + g_{01} * rooms + g_{02} * futoukou + u_0 \\ b_1 &= g_{10} + g_{11} * rooms + g_{12} * futoukou + u_1 \end{aligned}$$

# ランダム切片

# ランダムスロープ

回帰式の統合

$$\begin{aligned} y &= (g_{00} + g_{01} * rooms + g_{02} * futoukou + u_0) + \\ &\quad (g_{10} + g_{11} * rooms + g_{12} * futoukou + u_1) * support + e \\ &= (g_{00} + g_{01} * rooms + g_{02} * futoukou + g_{10} * support + \\ &\quad g_{11} * rooms * support + g_{12} * futoukou * support) + (u_0 + u_1 * support + e) \end{aligned}$$

固定効果

g00 ... (Intercept)  
g01 ... rooms  
g02 ... futoukou  
g10 ... support  
g11 ... rooms \* support  
g12 ... futoukou \* support

変量効果

u0 ... (Intercept)  
u1 ... support  
e ... (Residual)

## Rで階層線形モデル分析をするためのデータ構造

レベル1とレベル2のデータを統合した1つのデータが必要である。

## レベル1データ

	A	B	C	D
1	id	school	support	kenkou
2	1	e	10	11
3	2	g	12	21
4	3	h	7	19
5	4	a	14	7
6	5	a	14	7

## レベル2データ

	A	B	C
1	school	rooms	futoukou
2	a	3	0
3	b	1	1
4	c	2	2
5	d	5	2
6	e	4	3
7	f	5	4
8	g	7	5
9	h	3	6

## レベル2の情報をレベル1に統合したデータ

	A	B	C	D	E	F
1	id	school	support	kenkou	rooms	futoukou
2	1	e	10	11	4	3
3	2	g	12	21	7	5
4	3	h	7	19	3	6
5	4	a	14	7	3	0
6	5	a	14	7	3	0

## 説明変数の平均偏差データ化

階層線形モデルでは，群ごとに，低階層の説明変数を平均偏差データ化して分析することが多い。

## 所属する群における説明変数の値が入った変数の生成

群平均変数名 <- ave(説明変数名, 群分け変数名)

平均偏差化説明変数名 <- 説明変数名 - 群平均変数名

## 階層線形モデル — lmer関数

lme4パッケージの読み込み  
library(lme4)

## ランダム切片モデル

```
オブジェクト名1 <- lmer(基準変数 ~ 固定効果説明変数
+ (1 | 群分け変数), data=データフレーム名)
summary(オブジェクト名1)
```

## ランダム切片・ランダムスロープモデル

```
オブジェクト名1 <- lmer(基準変数 ~ 固定効果説明変数
+ (1+レベル1の説明変数 | 群分け変数), data=データフレーム名)
summary(オブジェクト名2)
```

## ランダム切片・ランダムスロープモデル, 相関の推定なし

```
オブジェクト名3 <- lmer(基準変数 ~ 固定効果説明変数
+ (1 | 群分け変数) + (0+レベル1の説明変数 | 群分け変数),
data=データフレーム名)
summary(オブジェクト名3)
```

## モデルの適合度の比較

```
anova(オブジェクト名1, オブジェクト名2, ...)
```

あらかじめlme4パッケージをインストールしておく必要がある。  
 「(1 | 群分け変数)」で、ランダム切片の変量効果変数を表す。  
 「(1+レベル1の説明変数 | 群分け変数)」で、ランダム切片・ランダムスロープの変量効果変数を表す。  
 切片とスロープの相関係数を推定する場合は「1+」を省略してもよい。  
 「(0+レベル1の説明変数 | 群分け変数)」で、ランダムスロープの変量効果変数を表す。  
 「0+」とすると、切片とスロープの相関係数は推定されない。

【注意】: lmerを用いた場合は、切片とスロープの相関の推定がうまくいかないことがある。  
 固定効果の異なるモデル間でも、適合度の比較ができる。

```
> setwd("d:¥¥")
>
> # レベル1のデータの読み込み
> d1 <- read.table("階層線形モデル_データ1.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)

  id school support kenkou
1  1      a        5      8
2  2      a       10      7
3  3      a       14      8
4  4      a       11      8
5  5      a       14      7
6  6      a       10      6
>
> # 記述統計量
> library(psych)
> describeBy(d1[,c("support", "kenkou")], d1[,c("school")], mat=TRUE, digits=2)
      item group1 vars  n mean  sd median trimmed  mad min max range skew kurtosis  se
support1      1    a   1 40  9.70 3.23  10.0    9.84 4.45   1 14   13 -0.40  -0.63 0.51
support2      2    b   1 40  9.28 3.34   9.0    9.28 2.97   2 16   14  0.00  -0.67 0.53
support3      3    c   1 40 11.15 2.63  11.0   11.41 2.97   4 15   11 -0.74   0.06 0.42
support4      4    d   1 40 10.68 2.84  10.5   10.66 3.71   5 17   12  0.08  -0.88 0.45
support5      5    e   1 40 10.28 2.63  10.0   10.22 2.97   5 17   12  0.28  -0.31 0.42
support6      6    f   1 40 10.50 3.35  10.5   10.38 2.97   4 18   14  0.39  -0.22 0.53
```

```

support7      7      g      1 40  9.80 2.36  10.0    9.97 2.97   3 14    11 -0.68    0.16 0.37
support8      8      h      1 40  9.20 2.15   9.0    9.31 1.48   3 13    10 -0.57    0.36 0.34
kenkou1       9      a      2 40  9.38 3.07   9.0    9.19 2.97   3 18    15  0.56    0.06 0.49
kenkou2      10      b      2 40 11.43 3.13  12.0    11.38 2.97   5 19    14  0.05   -0.35 0.49
kenkou3      11      c      2 40 13.05 3.01  12.5    12.94 2.97   7 20    13  0.34   -0.41 0.48
kenkou4      12      d      2 40 13.22 3.32  13.0    13.28 2.97   7 20    13 -0.18   -0.60 0.52
kenkou5      13      e      2 40 13.75 3.43  14.0    14.03 2.97   4 21    17 -0.71    0.74 0.54
kenkou6      14      f      2 40 15.43 3.03  15.0    15.50 2.97   8 21    13 -0.26   -0.45 0.48
kenkou7      15      g      2 40 15.75 3.05  16.0    15.72 2.97  10 22    12  0.02   -0.93 0.48
kenkou8      16      h      2 40 16.38 2.39  16.0    16.25 2.97  12 22    10  0.43   -0.38 0.38
>

```

```

> # 各群の相関係数行列
> by(d1[,c("support", "kenkou")], d1$school, cor)

```

```

d1$school: a
      support      kenkou
support  1.0000000 -0.1151671
kenkou -0.1151671  1.0000000
-----

```

```

d1$school: b
      support      kenkou
support  1.0000000 -0.1072993
kenkou -0.1072993  1.0000000
-----

```

```

d1$school: c
      support      kenkou
support  1.0000000 -0.1370703
kenkou -0.1370703  1.0000000
-----

```

```

d1$school: d
      support      kenkou
support  1.0000000  0.4814743
kenkou  0.4814743  1.0000000
-----

```

```

d1$school: e
      support      kenkou
support  1.0000000  0.2888883
kenkou  0.2888883  1.0000000
-----

```

```

d1$school: f
      support      kenkou
support  1.0000000  0.2690342
kenkou  0.2690342  1.0000000
-----

```

```

d1$school: g
      support      kenkou
support  1.0000000  0.3672454
kenkou  0.3672454  1.0000000
-----

```

```

d1$school: h
      support      kenkou
support  1.0000000  0.4632841
kenkou  0.4632841  1.0000000
>

```

```

> # 群ごとに、説明変数を平均偏差データ化
> d1$support.gmean <- ave(d1$support, d1$school)
> d1$support.gcen <- d1$support - d1$support.gmean
> head(d1)

```

```

  id school support kenkou support.gmean support.gcen
1  1     a       5       8          9.7        -4.7
2  2     a      10       7          9.7         0.3
3  3     a      14       8          9.7         4.3
4  4     a      11       8          9.7         1.3
5  5     a      14       7          9.7         4.3
6  6     a      10       6          9.7         0.3
>

```

	A	B	C	D
1	id	school	support	kenkou
2	1	a	5	8
3	2	a	10	7
4	3	a	14	8
5	4	a	11	8
6	5	a	14	7
7	6	a	10	6
8	7	a	5	9
9	8	a	9	8
10	9	a	6	18
11	10	a	6	11
12	11	a	10	9
13	12	a	1	12
14	13	a	14	13
15	14	a	13	14
16	15	a	13	9
17	16	a	13	7
18	17	a	13	13
19	18	a	12	14
20	19	a	10	9
21	20	a	7	15

```

# レベル1のデータに群平均の列を追加
# 個人データから群平均を引いた変数の作成

```

> # レベル2のデータの読み込み

> d2 <- read.table("階層線形モデル\_データ2.csv", header=TRUE, sep=";", fileEncoding="shift-jis")

> d2

```

school rooms futoukou
1      a      3        0
2      b      1        1
3      c      2        2
4      d      5        2
5      e      4        3
6      f      5        4
7      g      7        5
8      h      3        6

```

	A	B	C
1	school	rooms	futoukou
2	a	3	0
3	b	1	1
4	c	2	2
5	d	5	2
6	e	4	3
7	f	5	4
8	g	7	5
9	h	3	6

> # レベル2データの情報をレベル1データに反映したデータの作成

> d3 <- merge(d1, d2, by="school") # 「by」で指定した変数をキーとして、レベル2のデータを追加

> d3 <- d3[order(d3\$id),]

> head(d3)

```

school id support kenkou support.gmean support.gcen rooms futoukou
1      a  1      5      8          9.7        -4.7      3        0
2      a  2     10      7          9.7         0.3      3        0
3      a  3     14      8          9.7         4.3      3        0
4      a  4     11      8          9.7         1.3      3        0
5      a  5     14      7          9.7         4.3      3        0
6      a  6     10      6          9.7         0.3      3        0

```

>

>

>

> # lmer関数を使う方法

> #lme4パッケージの読み込み

> library(lme4)

>

```

> # ランダム切片モデル
> result.lmer0 <- lmer(kenkou ~ support.gcen + rooms + futoukou
+                       + (1 | school),
+                       data=d3)
> summary(result.lmer0)

Linear mixed model fit by REML
Formula: kenkou ~ support.gcen + rooms + futoukou + (1 | school)
Data: d3
    AIC   BIC logLik deviance REMLdev
 1640 1663 -814.1    1619    1628
Random effects:
Groups   Name             Variance Std.Dev.
school   (Intercept)  0.23613   0.48593
Residual                    9.18282   3.03032
Number of obs: 320, groups: school, 8

# ランダム切片 (u0) の分散
# ランダム誤差 (e) の分散

Fixed effects:
              Estimate Std. Error t value
(Intercept)  10.14209    0.57510  17.635
support.gcen   0.17821    0.06026   2.957
rooms          0.07732    0.15674   0.493
futoukou       1.08342    0.14729   7.356

Correlation of Fixed Effects:
              (Intr) spprt. rooms
support.gcn   0.000
rooms        -0.649  0.000
futoukou     -0.218  0.000 -0.507
>
>

```

# 固定効果

```

> # ランダム切片・ランダムスロープモデル 1  関連の推定なし
> result.lmer1 <- lmer(kenkou ~ support.gcen + rooms + futoukou
+                        + (1 | school) + (0+support.gcen | school),
+                        data=d3)
> summary(result.lmer1)

Linear mixed model fit by REML
Formula: kenkou ~ support.gcen + rooms + futoukou + (1 | school) + (0 + support.gcen | school)
Data: d3
    AIC   BIC logLik deviance REMLdev
 1635 1662 -810.6    1613    1621
Random effects:
Groups   Name             Variance Std.Dev.
school   (Intercept)    0.246992 0.49698
school   support.gcen  0.062756 0.25051
Residual                    8.748394 2.95777
Number of obs: 320, groups: school, 8

Fixed effects:
              Estimate Std. Error t value
(Intercept)  10.14209    0.57517  17.633
support.gcen  0.20971    0.10736   1.953
rooms        0.07732    0.15676   0.493
futoukou     1.08342    0.14731   7.355

Correlation of Fixed Effects:
              (Intr) spprt. rooms
support.gcn  0.000
rooms       -0.649  0.000
futoukou    -0.218  0.000 -0.507
>

```

# ランダム切片 (u0) の分散  
# ランダムスロープ (u1) の分散  
# ランダム誤差 (e) の分散

# 固定効果

```

> # ランダム切片・ランダムスロープモデル 1  関連の推定あり
> result.lmer2 <- lmer(kenkou ~ support.gcen + rooms + futoukou
+                        + (1+support.gcen | school),
+                        data=d3)
> summary(result.lmer2)

Linear mixed model fit by REML
Formula: kenkou ~ support.gcen + rooms + futoukou + (1 + support.gcen | school)
Data: d3
    AIC   BIC logLik deviance REMLdev
 1637 1667 -810.6    1613    1621
Random effects:
Groups   Name             Variance Std.Dev. Corr
school   (Intercept)    0.259881 0.50979
          support.gcen  0.060644 0.24626  0.366
Residual                    8.752990 2.95855
Number of obs: 320, groups: school, 8

Fixed effects:
              Estimate Std. Error t value
(Intercept)  10.38923    0.57129  18.186
support.gcen  0.20797    0.10611   1.960
rooms        0.03605    0.15519   0.232
futoukou     1.05130    0.14594   7.204

Correlation of Fixed Effects:
              (Intr) spprt. rooms
support.gcn  0.104
rooms       -0.645  0.003
futoukou    -0.217 -0.016 -0.508
>
>

```

# ランダム切片 (u0) の分散  
# ランダムスロープ (u1) の分散  
# ランダム誤差 (e) の分散

# 固定効果

> # ランダム切片・ランダムスロープモデル 2 相関の推定なし

```
> result.lmer3 <- lmer(kenkou ~ support.gcen + rooms + futoukou
+ (support.gcen * rooms) + (support.gcen * futoukou)
+ (1 | school) + (0+support.gcen | school),
+ data=d3)
> summary(result.lmer3)
```

Linear mixed model fit by REML

Formula: kenkou ~ support.gcen + rooms + futoukou + (support.gcen \* rooms) + (support.gcen \* futoukou) + (1 | school) + (0 + support.gcen | school)

Data: d3

AIC	BIC	logLik	deviance	REMLdev
1641	1674	-811.3	1605	1623

Random effects:

Groups	Name	Variance	Std. Dev.
school	(Intercept)	0.246920	0.49691
school	support.gcen	0.011578	0.10760
Residual		8.751276	2.95826

# ランダム切片 (u0) の分散  
# ランダムスロープ (u1) の分散  
# ランダム誤差 (e) の分散

Number of obs: 320, groups: school, 8

Fixed effects:

# 固定効果

	Estimate	Std. Error	t value
(Intercept)	10.14209	0.57517	17.633
support.gcen	-0.27189	0.16320	-1.666
rooms	0.07732	0.15676	0.493
futoukou	1.08342	0.14731	7.355
support.gcen:rooms	0.08166	0.04855	1.682
support.gcen:futoukou	0.06393	0.04687	1.364

Correlation of Fixed Effects:

	(Intr)	spprt. rooms	futouk	spprt. gcen:r
support.gcen	0.000			
rooms	-0.649	0.000		
futoukou	-0.218	0.000	-0.507	
spprt.gcen:r	0.000	-0.661	0.000	0.000
spprt.gcen:f	0.000	-0.130	0.000	0.000

>  
>

> # ランダム切片・ランダムスロープモデル 2 相関の推定あり

```
> result.lmer4 <- lmer(kenkou ~ support.gcen + rooms + futoukou
+ (support.gcen * rooms) + (support.gcen * futoukou)
+ (1+support.gcen | school),
+ data=d3)
> summary(result.lmer4)
```

Linear mixed model fit by REML

Formula: kenkou ~ support.gcen + rooms + futoukou + (support.gcen \* rooms) + (support.gcen \* futoukou) + (1 + support.gcen | school)

Data: d3

AIC	BIC	logLik	deviance	REMLdev
1643	1680	-811.3	1605	1623

Random effects:

Groups	Name	Variance	Std. Dev.	Corr
school	(Intercept)	0.246910	0.49690	
	support.gcen	0.011382	0.10668	0.203
Residual		8.751965	2.95837	

# ランダム切片 (u0) の分散  
# ランダムスロープ (u1) の分散  
# ランダム誤差 (e) の分散

Number of obs: 320, groups: school, 8

Fixed effects:

# 固定効果

	Estimate	Std. Error	t value
(Intercept)	10.14209	0.57517	17.633
support.gcen	-0.27001	0.16277	-1.659
rooms	0.07732	0.15676	0.493
futoukou	1.08342	0.14731	7.355
support.gcen:rooms	0.08211	0.04843	1.695
support.gcen:futoukou	0.06234	0.04676	1.333



Correlation of Fixed Effects:

```
(Intr) spprt. rooms futouk spprt.gcn:r
support.gcn 0.082
rooms -0.649 -0.053
futoukou -0.218 -0.018 -0.507
spprt.gcn:r -0.049 -0.661 0.075 -0.038
spprt.gcn:f -0.016 -0.130 -0.037 0.073 -0.567
>
>
```

> # モデルの適合度の比較

> anova(result.lmer0, result.lmer1, result.lmer2, result.lmer3, result.lmer4)

refitting model(s) with ML (instead of REML)

Data: d3

Models:

```
result.lmer0: kenkou ~ support.gcen + rooms + futoukou + (1 | school)
result.lmer1: kenkou ~ support.gcen + rooms + futoukou + (1 | school) + (0 + support.gcen | school)
result.lmer2: kenkou ~ support.gcen + rooms + futoukou + (1 + support.gcen | school)
result.lmer3: kenkou ~ support.gcen + rooms + futoukou + (support.gcen * rooms) + (support.gcen * futoukou) + (1 | school) + (0 + support.gcen | school)
result.lmer4: kenkou ~ support.gcen + rooms + futoukou + (support.gcen * rooms) + (support.gcen * futoukou) + (1 + support.gcen | school)
```

	npars	AIC	BIC	logLik	deviance	Chisq	Df	Pr(>Chisq)
result.lmer0	6	1630.6	1653.2	-809.28	1618.6			
result.lmer1	7	1626.7	1653.1	-806.34	1612.7	5.8875	1	0.015249 *
result.lmer2	8	1628.5	1658.6	-806.23	1612.5	0.2165	1	0.641720
result.lmer3	9	1621.6	1655.5	-801.81	1603.6	8.8495	1	0.002932 **
result.lmer4	10	1623.6	1661.2	-801.78	1603.6	0.0425	1	0.836646

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

>  
>

## 階層線形モデル — lme関数

nlmeパッケージの読み込み  
library(nlme)

## ランダム切片モデル

オブジェクト名1 <- lme(基準変数 ~ 固定効果説明変数,  
random = ~ 1 | 群分け変数, data=データフレーム名)  
summary(オブジェクト名1)

## ランダム切片・ランダムスロープモデル

オブジェクト名2 <- lme(基準変数 ~ 固定効果説明変数,  
random = ~ 1+レベル1の説明変数 | 群分け変数, data=データフレーム名)  
summary(オブジェクト名2)

あらかじめnlmeパッケージをインストールしておく必要がある。

「1 | 群分け変数」で、ランダム切片の変量効果変数を表す。

「1+レベル1の説明変数 | 群分け変数」で、ランダム切片・ランダムスロープの変量効果変数を表す。  
ランダム切片・ランダムスロープモデルの「1+」は省略してもよい。

【注意】：lmeを用いた場合は、固定効果の異なるモデル間の適合度の比較ができない。  
切片とスロープの相関の推定はlmerよりはうまくいく。

```
> setwd("d:¥¥")
>
```

## &gt; # レベル1のデータの読み込み

```
> d1 <- read.table("階層線形モデル_データ1.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  id school support kenkou
1  1      a         5      8
2  2      a        10      7
3  3      a        14      8
4  4      a        11      8
5  5      a        14      7
6  6      a        10      6
>
```

## &gt; # 記述統計量

```
> library(psych)
```

```
> describeBy(d1[,c("support", "kenkou")], d1[,c("school")], mat=TRUE, digits=2)
```

	item	group	l	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
support1	1	a	1	40	9.70	3.23	10.0	9.84	4.45	1	14	13	-0.40	-0.63	0.51	
support2	2	b	1	40	9.28	3.34	9.0	9.28	2.97	2	16	14	0.00	-0.67	0.53	
support3	3	c	1	40	11.15	2.63	11.0	11.41	2.97	4	15	11	-0.74	0.06	0.42	
support4	4	d	1	40	10.68	2.84	10.5	10.66	3.71	5	17	12	0.08	-0.88	0.45	
support5	5	e	1	40	10.28	2.63	10.0	10.22	2.97	5	17	12	0.28	-0.31	0.42	
support6	6	f	1	40	10.50	3.35	10.5	10.38	2.97	4	18	14	0.39	-0.22	0.53	
support7	7	g	1	40	9.80	2.36	10.0	9.97	2.97	3	14	11	-0.68	0.16	0.37	
support8	8	h	1	40	9.20	2.15	9.0	9.31	1.48	3	13	10	-0.57	0.36	0.34	
kenkou1	9	a	2	40	9.38	3.07	9.0	9.19	2.97	3	18	15	0.56	0.06	0.49	
kenkou2	10	b	2	40	11.43	3.13	12.0	11.38	2.97	5	19	14	0.05	-0.35	0.49	
kenkou3	11	c	2	40	13.05	3.01	12.5	12.94	2.97	7	20	13	0.34	-0.41	0.48	
kenkou4	12	d	2	40	13.22	3.32	13.0	13.28	2.97	7	20	13	-0.18	-0.60	0.52	
kenkou5	13	e	2	40	13.75	3.43	14.0	14.03	2.97	4	21	17	-0.71	0.74	0.54	
kenkou6	14	f	2	40	15.43	3.03	15.0	15.50	2.97	8	21	13	-0.26	-0.45	0.48	
kenkou7	15	g	2	40	15.75	3.05	16.0	15.72	2.97	10	22	12	0.02	-0.93	0.48	
kenkou8	16	h	2	40	16.38	2.39	16.0	16.25	2.97	12	22	10	0.43	-0.38	0.38	

```
>
```

```
> # 各群の相関係数行列
> by(d1[,c("support", "kenkou")], d1$school, cor)
```

```
d1$school: a
      support    kenkou
support 1.0000000 -0.1151671
kenkou -0.1151671  1.0000000
```

```
d1$school: b
      support    kenkou
support 1.0000000 -0.1072993
kenkou -0.1072993  1.0000000
```

```
d1$school: c
      support    kenkou
support 1.0000000 -0.1370703
kenkou -0.1370703  1.0000000
```

```
d1$school: d
      support    kenkou
support 1.0000000  0.4814743
kenkou  0.4814743  1.0000000
```

```
d1$school: e
      support    kenkou
support 1.0000000  0.2888883
kenkou  0.2888883  1.0000000
```

```
d1$school: f
      support    kenkou
support 1.0000000  0.2690342
kenkou  0.2690342  1.0000000
```

```
d1$school: g
      support    kenkou
support 1.0000000  0.3672454
kenkou  0.3672454  1.0000000
```

```
d1$school: h
      support    kenkou
support 1.0000000  0.4632841
kenkou  0.4632841  1.0000000
```

```
>
```

```
> # 群ごとに、説明変数を平均偏差データ化
```

```
> d1$support.gmean <- ave(d1$support, d1$school)
> d1$support.gcen <- d1$support - d1$support.gmean
> head(d1)
```

```
  id school support kenkou support.gmean support.gcen
1  1     a      5      8         9.7         -4.7
2  2     a     10      7         9.7          0.3
3  3     a     14      8         9.7          4.3
4  4     a     11      8         9.7          1.3
5  5     a     14      7         9.7          4.3
6  6     a     10      6         9.7          0.3
```

```
>
```

	A	B	C	D
1	id	school	support	kenkou
2	1	a	5	8
3	2	a	10	7
4	3	a	14	8
5	4	a	11	8
6	5	a	14	7
7	6	a	10	6
8	7	a	5	9
9	8	a	9	8
10	9	a	6	18
11	10	a	6	11
12	11	a	10	9
13	12	a	1	12
14	13	a	14	13
15	14	a	13	14
16	15	a	13	9
17	16	a	13	7
18	17	a	13	13
19	18	a	12	14
20	19	a	10	9
21	20	a	7	15

```
# レベル1のデータに群平均の列を追加
```

```
# 個人データから群平均を引いた変数の作成
```

```
> # レベル2のデータの読み込み
```

```
> d2 <- read.table("階層線形モデル_データ2.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> d2
```

```
  school rooms futoukou
1      a      3         0
2      b      1         1
3      c      2         2
4      d      5         2
5      e      4         3
6      f      5         4
7      g      7         5
8      h      3         6
```

	A	B	C
1	school	rooms	futoukou
2	a	3	0
3	b	1	1
4	c	2	2
5	d	5	2
6	e	4	3
7	f	5	4
8	g	7	5
9	h	3	6

```
> # レベル2データの情報をレベル1データに反映したデータの作成
```

```
> d3 <- merge(d1, d2, by="school") # 「by」で指定した変数をキーとして、レベル2のデータを追加
```

```
> d3 <- d3[order(d3$id),]
```

```
> head(d3)
```

```
  school id support kenkou support.gmean support.gcen rooms futoukou
1      a  1         5       8           9.7         -4.7      3         0
2      a  2        10       7           9.7          0.3      3         0
3      a  3        14       8           9.7          4.3      3         0
4      a  4        11       8           9.7          1.3      3         0
5      a  5        14       7           9.7          4.3      3         0
6      a  6        10       6           9.7          0.3      3         0
```

```
> # lme関数を使う方法
```

```
> #nlmeパッケージの読み込み
```

```
> library(nlme)
```

```
>
```

```
> # ランダム切片モデル
> result.lme0 <- lme(kenkou ~ support.gcen + rooms + futoukou,
+                   random = ~ 1 | school,
+                   data=d3)
> summary(result.lme0)
```

Linear mixed-effects model fit by REML

Data: d3

AIC	BIC	logLik
1640.27	1662.804	-814.135

Random effects:

Formula: ~1 | school

(Intercept) Residual

StdDev: 0.4859375 3.030316

# ランダム切片 (u0) の標準偏差

Fixed effects: kenkou ~ support.gcen + rooms + futoukou

# 固定効果

	Value	Std. Error	DF	t-value	p-value
(Intercept)	10.142085	0.5751724	311	17.633121	0.0000
support.gcen	0.178211	0.0602575	311	2.957489	0.0033
rooms	0.077324	0.1567559	5	0.493277	0.6427
futoukou	1.083417	0.1473102	5	7.354662	0.0007

Correlation:

(Intr) spprt. rooms

support.gcen	0.000
rooms	-0.649 0.000
futoukou	-0.218 0.000 -0.507

Standardized Within-Group Residuals:

Min	Q1	Med	Q3	Max
-2.981065892	-0.689897574	0.001932531	0.732537139	2.947451891

Number of Observations: 320

Number of Groups: 8

## &gt; # ランダム切片・ランダムスロープモデル 1

```
> result.lme1 <- lme(kenkou ~ support.gcen + rooms + futoukou,
+                    random = ~1+support.gcen | school,
+                    data=d3)
> summary(result.lme1)
```

Linear mixed-effects model fit by REML

Data: d3  
 AIC BIC logLik  
 1637.177 1667.223 -810.5884

Random effects:

Formula: ~1 + support.gcen | school  
 Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr	
(Intercept)	0.5097907	(Intr)	# ランダム切片 (u0) の標準偏差
support.gcen	0.2462596	0.366	# ランダムスロープ (u1) の標準偏差
Residual	2.9585450		# ランダム誤差 (e) の標準偏差

Fixed effects: kenkou ~ support.gcen + rooms + futoukou # 固定効果

	Value	Std.Error	DF	t-value	p-value
(Intercept)	10.389482	0.5712641	311	18.186827	0.0000
support.gcen	0.207970	0.1061179	311	1.959798	0.0509
rooms	0.036003	0.1551850	5	0.231998	0.8257
futoukou	1.051264	0.1459317	5	7.203805	0.0008

Correlation:

	(Intr)	spprt.	rooms
support.gcen	0.104		
rooms	-0.645	0.003	
futoukou	-0.217	-0.016	-0.508

Standardized Within-Group Residuals:

	Min	Q1	Med	Q3	Max
	-3.14590341	-0.65637835	0.01818610	0.67811975	2.69669315

Number of Observations: 320

Number of Groups: 8

>

## &gt; # ランダム切片・ランダムスロープモデル 2

```
> result.lme2 <- lme(kenkou ~ support.gcen + rooms + futoukou
+                      + (support.gcen * rooms) + (support.gcen * futoukou),
+                      random = ~1+support.gcen | school,
+                      data=d3)
> summary(result.lme2)
```

Linear mixed-effects model fit by REML

Data: d3  
 AIC BIC logLik  
 1642.548 1680.042 -811.2742

Random effects:

Formula: ~1 + support.gcen | school  
 Structure: General positive-definite, Log-Cholesky parametrization

	StdDev	Corr	
(Intercept)	0.4968951	(Intr)	# ランダム切片 (u0) の標準偏差
support.gcen	0.1066797	0.203	# ランダムスロープ (u1) の標準偏差
Residual	2.9583731		# ランダム誤差 (e) の標準偏差

Fixed effects: kenkou ~ support.gcen + rooms + futoukou + (support.gcen \* rooms) + (support.gcen \* futoukou) # 固定効果

	Value	Std.Error	DF	t-value	p-value
(Intercept)	10.142085	0.5751714	309	17.633152	0.0000
support.gcen	-0.270004	0.1627717	309	-1.658792	0.0982
rooms	0.077324	0.1567556	5	0.493278	0.6427
futoukou	1.083417	0.1473100	5	7.354675	0.0007
support.gcen:rooms	0.082114	0.0484339	309	1.695392	0.0910
support.gcen:futoukou	0.062341	0.0467619	309	1.333158	0.1835

```
Correlation:
              (Intr) spprt. rooms futouk spprt.gcn:r
support.gcen      0.082
rooms            -0.649 -0.053
futoukou         -0.218 -0.018 -0.507
support.gcen:rooms -0.049 -0.661  0.075 -0.038
support.gcen:futoukou -0.016 -0.130 -0.037  0.073 -0.567
```

```
Standardized Within-Group Residuals:
      Min      Q1      Med      Q3      Max
-3.11835666 -0.65749189  0.02274071  0.69469014  2.67482674
```

```
Number of Observations: 320
```

```
Number of Groups: 8
```

```
>
```

```
> # モデルの適合度の比較
```

```
> anova(result.lme0, result.lme1)
```

	Model	df	AIC	BIC	logLik	Test	L.Ratio	p-value
result.lme0	1	6	1640.270	1662.804	-814.1350			
result.lme1	2	8	1637.177	1667.223	-810.5884	1 vs 2	7.093129	0.0288

```
>
```

```
>
```

## 20 検定力分析



## 検定力分析 — 2 群の平均値

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

## 検定力の計算

対応のある2群の平均値

pwr.t.test(d=効果量, n=標本サイズ, type="paired", sig.level=有意水準, alternative="two.sided")

対応のない2群の平均値

pwr.t2n.test(d=効果量, n1=標本サイズ1, n2=標本サイズ2, sig.level=有意水準, alternative="two.sided")

## 標本サイズの推定

対応のある2群の平均値 (1群あたりの標本サイズ)

pwr.t.test(d=効果量, type="paired", sig.level=有意水準, power=検定力, alternative="two.sided")

対応のない2群の平均値 (1群あたりの標本サイズ)

pwr.t.test(d=効果量, type="two.sample", sig.level=有意水準, power=検定力, alternative="two.sided")

## 標本サイズと検定力の図

```
ptt <- pwr.t.test(d=効果量, type="two.sample", sig.level=有意水準, power=検定力, alternative="two.sided")
plot(ptt)
```

## 参考目安

効果量 (d)    小 : 0.2    中 : 0.5    大 : 0.8

```
> rm(list=ls())
>
> # pwr パッケージの読み込み
> library("pwr")
>
>
> # 検定力の計算
>
> # 対応のある2群の平均値
> pwr.t.test(d=0.24, n=100, type="paired", sig.level=0.05, alternative="two.sided")
```

Paired t test power calculation

```
      n = 100
      d = 0.24
sig.level = 0.05
  power = 0.6615591
alternative = two.sided
```

NOTE: n is number of \*pairs\*

```
> # 対応のない2群の平均値
> pwr.t2n.test(d=0.24, n1=100, n2=80, sig.level=0.05, alternative="two.sided")
```

```
t test power calculation
```

```
      n1 = 100
      n2 = 80
      d = 0.24
sig.level = 0.05
  power = 0.3564065
alternative = two.sided
```

```
> # 標本サイズの推定
```

```
>
> # 対応のある2群の平均値 (1群あたりの標本サイズ)
> pwr.t.test(d=0.5, type="paired", sig.level=0.05, power=0.8, alternative="two.sided")
```

```
Paired t test power calculation
```

```
      n = 33.36713          # n=34
      d = 0.5
sig.level = 0.05
  power = 0.8
alternative = two.sided
```

NOTE: n is number of \*pairs\*

```
> # 対応のない2群の平均値 (1群あたりの標本サイズ)
> pwr.t.test(d=0.5, type="two.sample", sig.level=0.05, power=0.8, alternative="two.sided")
```

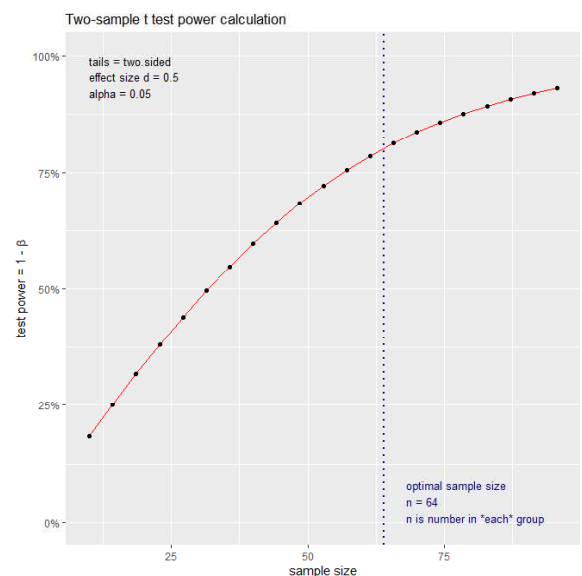
```
Two-sample t test power calculation
```

```
      n = 63.76561          # 1群あたり n=64
      d = 0.5
sig.level = 0.05
  power = 0.8
alternative = two.sided
```

NOTE: n is number in \*each\* group

```
> # 標本サイズと検定力の図
```

```
> ptt <- pwr.t.test(d=0.5, type="two.sample", sig.level=0.05, power=0.8, alternative="two.sided")
> plot(ptt)
```



## 検定力分析 — 相関係数

**pwr パッケージの読み込み**  
`library("pwr")`

あらかじめ pwr パッケージをインストールしておく必要がある。

**検定力の計算**

`pwr.r.test(r=相関係数, n=標本サイズ, sig.level=有意水準, alternative="two.sided")`

**標本サイズの推定**

`pwr.r.test(r=相関係数, sig.level=有意水準, power=検定力, alternative="two.sided")`

**標本サイズと検定力の図**

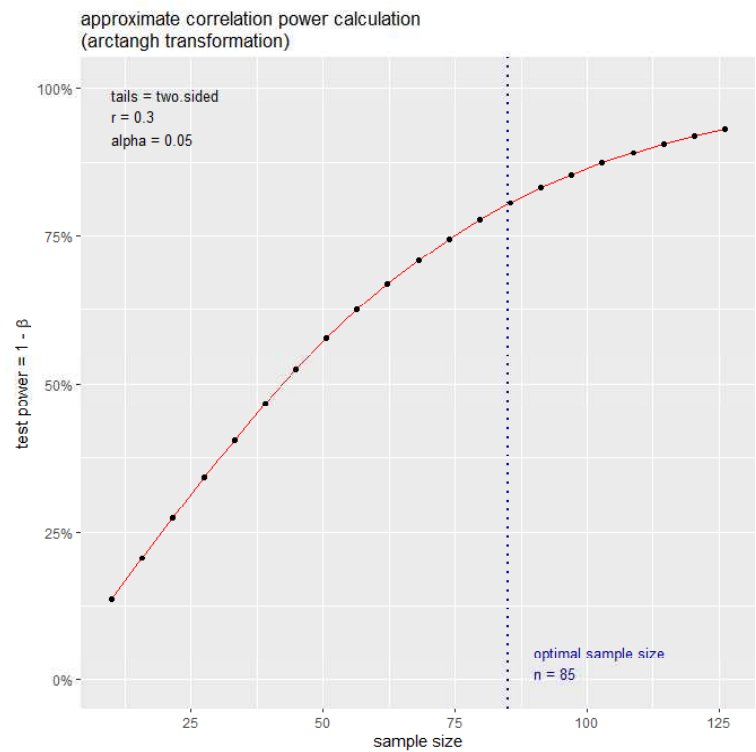
`pwt <- pwr.r.test(r=相関係数, sig.level=有意水準, power=検定力, alternative="two.sided")`  
`plot(pwt)`

**参考目安**

効果量 (r)    小 : 0.1    中 : 0.3    大 : 0.5

```
> rm(list=ls())
>
> # pwr パッケージの読み込み
> library("pwr")
>
>
> # 検定力の計算
> pwr.r.test(r=0.15, n=200, sig.level=0.05, alternative="two.sided")
      approximate correlation power calculation (arctangh transformation)
               n = 200
               r = 0.15
      sig.level = 0.05
       power = 0.5659427
  alternative = two.sided
>
>
> # 標本サイズの推定
> pwr.r.test(r=0.3, sig.level=0.05, power=0.80, alternative="two.sided")
      approximate correlation power calculation (arctangh transformation)
               n = 84.07364                # n=85
               r = 0.3
      sig.level = 0.05
       power = 0.8
  alternative = two.sided
>
>
>
```

```
> # 標本サイズと検定力の図  
> prt <- pwr.r.test(r=0.3, sig.level=0.05,  
+               power=0.80, alternative="two.sided")  
> plot(prt)  
>  
>  
>  
>  
>
```



## 検定力分析 — クロス集計表

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

## 効果量

(効果量 <- ES.w2(割合のクロス表))

## 検定力の計算

pwr.chisq.test(w=効果量, df=クロス表の自由度, N=標本サイズ, sig.level=有意水準)

## 標本サイズの推定

pwr.chisq.test(w=効果量, df=クロス表の自由度, sig.level=有意水準, power=検定力)

## 標本サイズと検定力の図

pct <- pwr.chisq.test(w=効果量, df=クロス表の自由度, sig.level=有意水準, power=検定力)  
plot(pct)

## 参考目安

効果量 (連関係数) 小 : 0.1 中 : 0.3 大 : 0.5

```
> rm(list=ls())
```

## &gt; # クロス表

```
> (t1 <- matrix(c(10, 20, 30, 40, 50, 60), nrow=2, byrow=TRUE))
```

```
      [,1] [,2] [,3]
[1,]   10   20   30
[2,]   40   50   60
```

## &gt; # 割合

```
> (p1 <- prop.table(t1))
```

```
      [,1]      [,2]      [,3]
[1,] 0.04761905 0.0952381 0.1428571
[2,] 0.19047619 0.2380952 0.2857143
```

```
>
```

```
>
```

## &gt; # クラメル連関係数

```
> library(vcd)
```

```
> assocstats(t1)
```

```
              X^2 df P(> X^2)
Likelihood Ratio 2.9027  2  0.23425
Pearson          2.8000  2  0.24660
```

```
Phi-Coefficient   : NA
```

```
Contingency Coeff.: 0.115
```

```
Cramer's V       : 0.115
```

```
>
```

```
>
```

```
>
```

## &gt; # pwr パッケージの読み込み

```
> library("pwr")
```

```
>
```

```

>
> # 効果量
> (esw <- ES.w2(p1))
[1] 0.1154701
>
>
> # 標本サイズ
> sum(t1)
[1] 210

> # 検定力の計算
> pwr.chisq.test(w=esw, df=2, N=210, sig.level=0.05)

```

Chi squared power calculation

```

      w = 0.1154701
      N = 210
      df = 2
sig.level = 0.05
power = 0.3023521

```

NOTE: N is the number of observations

```

> # 標本サイズの推定
> pwr.chisq.test(w=0.3, df=2, sig.level=0.05, power=0.8)

```

Chi squared power calculation

```

      w = 0.3
      N = 107.0521
      df = 2
sig.level = 0.05
power = 0.8

```

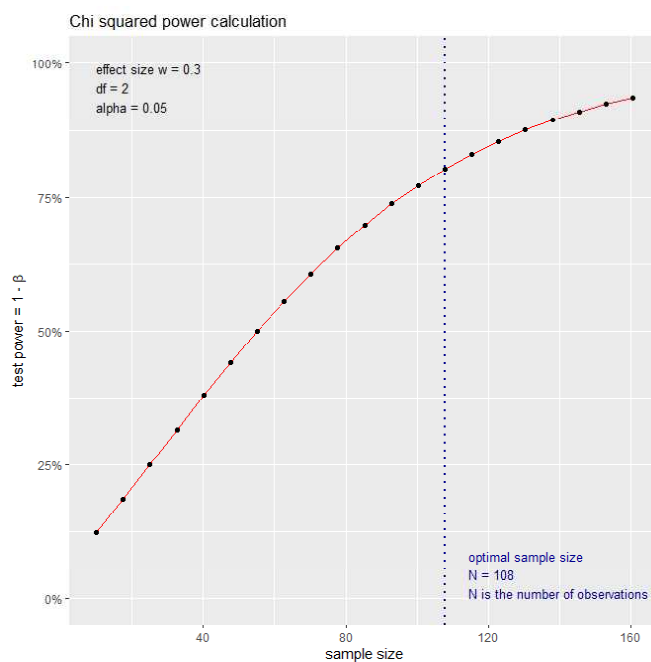
# n=108

NOTE: N is the number of observations

```

>
>
>
>
> # 標本サイズと検定力の図
> pct <- pwr.chisq.test(w=0.3, df=2, sig.level=0.05, power=0.8)
> plot(pct)
>
>
>
>

```



## 検定力分析 — 2 群の比率

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

比率

```
p1 <- 比率1
p2 <- 比率2
```

効果量

```
効果量 <- ES.h(比率1, 比率2)
2*(asin(sqrt(比率1))-asin(sqrt(比率2)))
```

## 検定力の計算

対応のある2群の比率

```
pwr.p.test(h=効果量, n=標本サイズ, sig.level=有意水準, alternative="two.sided")
```

対応のない2群の比率 (標本サイズ異なる)

```
pwr.2p2n.test(h=効果量, n1=標本サイズ1, n2=標本サイズ2, sig.level=有意水準, alternative="two.sided")
```

## 標本サイズの推定

```
p1 <- 比率1
p2 <- 比率2
効果量 <- ES.h(比率1, 比率2)
```

対応のある2群の比率 (1群あたりの標本サイズ)

```
pwr.p.test(h=効果量, sig.level=有意水準, power=検定力, alternative="two.sided")
```

対応のない2群の比率 (1群あたりの標本サイズ)

```
pwr.2p.test(h=効果量, sig.level=有意水準, power=検定力, alternative="two.sided")
```

## 標本サイズと検定力の図

```
ppt <- pwr.2p.test(h=効果量, sig.level=有意水準, power=検定力, alternative="two.sided")
plot(ppt)
```

## 参考目安

効果量 (h)    小 : 0.2    中 : 0.5    大 : 0.8

```
> rm(list=ls())
```

```
> # pwr パッケージの読み込み
> library("pwr")
```

```
>
```

```
>
```

```
>
```

```
>
```

```
> # 比率
```

```
> p1 <- 0.5
```

```
> p2 <- 0.4
```

```
>
```

```
> # 効果量
> (esh <- ES.h(p1, p2))
[1] 0.2013579
>
> 2*(asin(sqrt(p1))-asin(sqrt(p2)))
[1] 0.2013579
```

```
> # 検定力の計算
```

```
>
> # 対応のある2群の比率
> pwr.p.test(h=esh, n=60, sig.level=0.05, alternative="two.sided")

proportion power calculation for binomial distribution (arcsine transformation)

      h = 0.2013579
      n = 60
sig.level = 0.05
  power = 0.3447014
alternative = two.sided
```

```
> # 対応のない2群の比率 (標本サイズ異なる)
```

```
> pwr.2p2n.test(h=esh, n1=60, n2=80, sig.level=0.05, alternative="two.sided")

difference of proportion power calculation for binomial distribution (arcsine transformation)

      h = 0.2013579
     n1 = 60
     n2 = 80
sig.level = 0.05
  power = 0.2182687
alternative = two.sided
```

NOTE: different sample sizes

```
> # 標本サイズの推定
```

```
> p1 <- 0.6
> p2 <- 0.35
> esh <- ES.h(p1, p2)
>
```

```
> # 対応のある2群の比率 (1群あたりの標本サイズ)
```

```
> pwr.p.test(h=esh, sig.level=0.05, power=0.8, alternative="two.sided")

proportion power calculation for binomial distribution (arcsine transformation)

      h = 0.5060506
      n = 30.64917          # n=31
sig.level = 0.05
  power = 0.8
alternative = two.sided
```

```
> # 対応のない2群の比率 (1群あたりの標本サイズ)
```

```
> pwr.2p.test(h=esh, sig.level=0.05, power=0.8, alternative="two.sided")

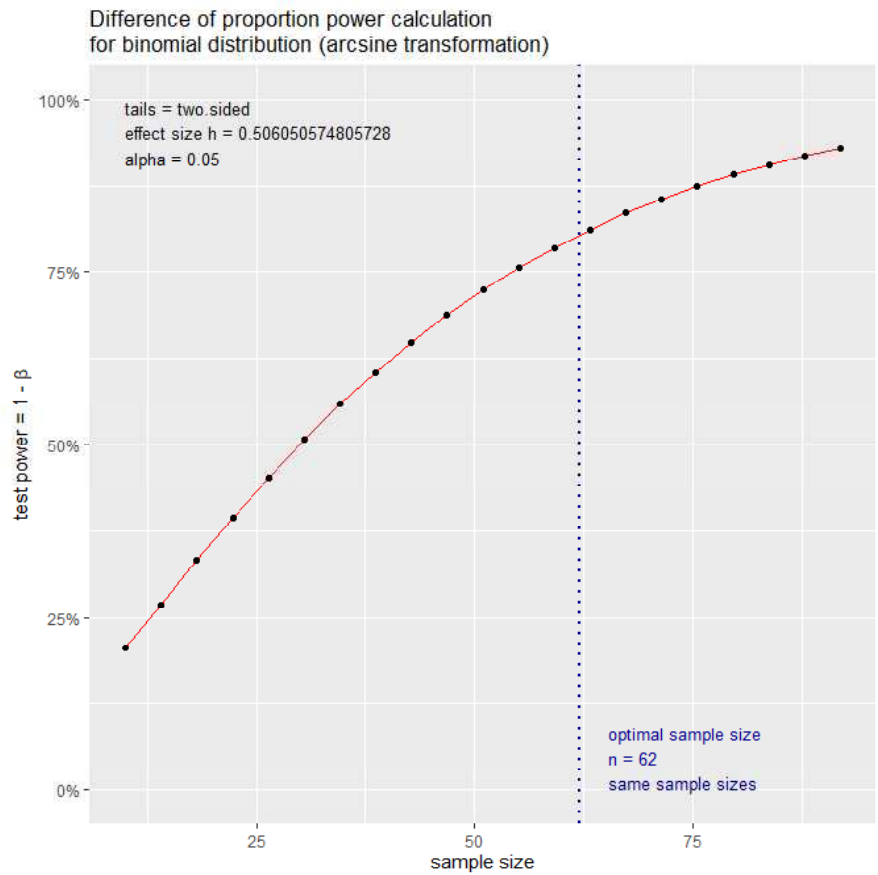
Difference of proportion power calculation for binomial distribution (arcsine transformation)

      h = 0.5060506
      n = 61.29835          # 1群あたり, n=62
sig.level = 0.05
  power = 0.8
alternative = two.sided
```

NOTE: same sample sizes



```
> # 標本サイズと検定力の図  
> ppt <- pwr.2p.test(h=esh, sig.level=0.05, power=0.8, alternative="two.sided")  
> plot(ppt)
```



## 検定力分析 — 1つの被験者間要因 (1 B)

**pwr パッケージの読み込み**  
`library("pwr")`

あらかじめ pwr パッケージをインストールしておく必要がある。

**設定**

分散分析表と効果量の推定を行って、以下の入力エリアの値（自由度、偏  $\eta^2$  乗、水準数）を得る。  
 それらを入力エリアで設定する。

**検定力の計算**

入力エリア Begin -----

**自由度**

`dfu <- 被験者間要因の自由度`  
`dfv <- 残差の自由度`

**偏イータ2乗(partial eta square)**

`peta2 <- 偏  $\eta^2$  乗`

入力エリア End -----

**f2 統計量**

`(ef2 <- peta2/(1-peta2))`

**検定力**

`pwr.f2.test(u=dfu, v=dfv, f2=ef2, sig.level=有意水準)`

**標本サイズの推定**

入力エリア Begin -----

**水準数**

`nu <- 被験者間要因の水準数`

**偏イータ2乗(partial eta square)**

`peta2 <- 偏  $\eta^2$  乗`

入力エリア End -----

**自由度**

`(dfu <- (nu-1))`

**f2 統計量**

`(ef2 <- peta2/(1-peta2))`

**標本サイズ (1群あたりの標本サイズ)**

`(pf2t <- pwr.f2.test(u=dfu, f2=ef2, sig.level=有意水準, power=検定力))`  
`(n <- ceiling((pf2t$v + nu)/nu))`

**参考目安**

効果量 ( $\eta^2$  乗)    小 : 0.01    中 : 0.06    大 : 0.14

```

> rm(list=ls())
>
> setwd("c:\\home\\Rdocuments\\scripts\\")
>
>
> # パッケージの読み込み
> library(DescTools)
> library("pwr")
>
>
> # 1B要因
>
> d1 <- read.table("1B平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 group utsu
1    1    40   29
2    2    40   32
3    3    40   22
4    4    40   28
5    5    40   27
6    6    40   20
>
> # 分散分析
> d1$group <- as.factor(d1$group)
> result.aov <- aov(utsu ~ group, data=d1)
> summary(result.aov)
              Df Sum Sq Mean Sq F value Pr(>F)
group          2     291   145.67   3.392 0.0353 *
Residuals    235  10093    42.95
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # 効果量
> (eta2 <- EtaSq(result.aov))
              eta.sq eta.sq.part
group 0.02805788 0.02805788
>
>
> # 検定力の計算
>
> # 入力エリア Begin -----
>
> # 自由度
> dfu <- 2
> dfv <- 235
>
> # 偏イータ2乗(partial eta square)
> peta2 <- 0.02805788
>
> # 入力エリア End -----
>
>
> # f2 統計量
> (ef2 <- peta2/(1-peta2))
[1] 0.02886785
>
> # 検定力
> pwr.f2.test(u=dfu, v=dfv, f2=ef2, sig.level=0.05)

Multiple regression power calculation

      u = 2
      v = 235
      f2 = 0.02886785
sig.level = 0.05
power = 0.6407604

```

```

> # 標本サイズの推定
>
> # 入力エリア Begin -----
>
> # 水準数
> nu <- 3
>
> # 偏イータ2乗 (partial eta square)
> peta2 <- 0.06
>
> # 入力エリア End -----
>
>
> # 自由度
> (dfu <- (nu-1))
[1] 2
>
> # f2 統計量
> (ef2 <- peta2/(1-peta2))
[1] 0.25
>
> # 標本サイズ (1群あたりの標本サイズ)
> (pf2t <- pwr.f2.test(u=dfu, f2=ef2, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 2
      v = 150.9791
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling((pf2t$v + nu)/nu))
[1] 52
>

```

# 1群あたり, n=52

## 検定力分析 — 1つの被験者内要因 (1W)

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

## 設定

分散分析表と効果量の推定を行って、以下の入力エリアの値（自由度、偏  $\eta^2$  乗、水準数）を得る。  
それらを入力エリアで設定する。

## 検定力の計算

入力エリア Begin -----

## 自由度

dfu <- 被験者内要因の自由度  
dfv <- 残差の自由度

## 偏イータ2乗(partial eta square)

peta2 <- 偏  $\eta^2$  乗

入力エリア End -----

## f2 統計量

(ef2 <- peta2/(1-peta2))

## 検定力

pwr.f2.test(u=dfu, v=dfv, f2=ef2, sig.level=有意水準)

## 標本サイズの推定

入力エリア Begin -----

## 水準数

nu <- 被験者内要因の水準数

## 偏イータ2乗(partial eta square)

peta2 <- 偏  $\eta^2$  乗

入力エリア End -----

## 自由度

(dfu <- (nu-1))

## f2 統計量

(ef2 <- peta2/(1-peta2))

## 標本サイズ (1群あたりの標本サイズ)

(pf2t <- pwr.f2.test(u=dfu, f2=ef2, sig.level=有意水準, power=検定力))  
(n <- ceiling(pf2t\$v/dfu + 1))

## 参考目安

効果量 ( $\eta^2$  乗)    小 : 0.01    中 : 0.06    大 : 0.14

```

> rm(list=ls())
>
> setwd("c:\\home\\Rdocuments\\scripts\\")
>
>
> # パッケージの読み込み
> library(DescTools)
> library("pwr")
>
>
>
> # 1W要因
>
> d1 <- read.table("1W平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  Team Rank Cooperate
1   13   15         14
2   15   15         10
3   11   13         13
4   14   14         18
5   15   10         14
6   11   11         14
>
> #stackデータの作成
> dtmp <- d1
> d3 <- stack(dtmp)
> did <- c(1:nrow(dtmp))
> d4 <- data.frame(d3, did)
> colnames(d4) <- c("y", "x", "id")
> d4$x <- as.factor(d4$x)
> d4$id <- as.factor(d4$id)
> head(d4)
   y    x id
1 13 Team  1
2 15 Team  2
3 11 Team  3
4 14 Team  4
5 15 Team  5
6 11 Team  6
>
> #分散分析
> result.aov <- aov(y ~ x + id, data=d4)
> summary(result.aov)
          Df Sum Sq Mean Sq F value    Pr(>F)
x              2   124.1    62.05  30.229 9.05e-13 ***
id             163   595.1     3.65   1.778 6.35e-06 ***
Residuals    326   669.2     2.05
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # 効果量
> (eta2 <- EtaSq(result.aov))
      eta.sq eta.sq.part
x  0.08939073  0.1564409
id 0.42859736  0.4706710
>
>
> # 検定力の計算
>
> # 入力エリア Begin -----
>
> # 自由度
> dfu <- 2
> dfv <- 326
>
> # 偏イータ2乗(partial eta square)
> peta2 <- 0.1564409
>
> # 入力エリア End -----

```

```

> # f2 統計量
> (ef2 <- peta2/(1-peta2))
[1] 0.1854534
>
> # 検定力
> pwr.f2.test(u=dfu, v=dfv, f2=ef2, sig.level=0.05)

Multiple regression power calculation

      u = 2
      v = 326
      f2 = 0.1854534
sig.level = 0.05
power = 1

>
>
>
>
> # 標本サイズの推定
>
> # 入力エリア Begin -----
>
> # 水準数
> nu <- 3
>
> # 偏イータ2乗(partial eta square)
> peta2 <- 0.06
>
> # 入力エリア End -----
>
>
> # 自由度
> (dfu <- (nu-1))
[1] 2
>
> # f2 統計量
> (ef2 <- peta2/(1-peta2))
[1] 0.25
>
> # 標本サイズ (1群あたりの標本サイズ)
> (pf2t <- pwr.f2.test(u=dfu, f2=ef2, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 2
      v = 150.9791
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling(pf2t$v/dfu + 1))
[1] 77                                     # n=77
>
>

```

## 検定力分析 — 2つの被験者間要因 (2B)

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

## 設定

分散分析表と効果量の推定を行って、以下の入力エリアの値（自由度、偏  $\eta^2$  乗、水準数）を得る。  
それらを入力エリアで設定する。

## 検定力の計算

入力エリア Begin -----

## 自由度

```
dfuB1 <- 被験者間要因1の自由度
dfuB2 <- 被験者間要因2の自由度
dfuB1B2 <- 交互作用の自由度
dfv <- 残差の自由度
```

## 偏イータ2乗(partial eta square)

```
peta2B1 <- 被験者間要因1の偏  $\eta^2$  乗
peta2B2 <- 被験者間要因2の偏  $\eta^2$  乗
peta2B1B2 <- 交互作用の偏  $\eta^2$  乗
```

入力エリア End -----

## f2 統計量

```
(ef2B1 <- peta2B1/(1-peta2B1))
(ef2B2 <- peta2B2/(1-peta2B2))
(ef2B1B2 <- peta2B1B2/(1-peta2B1B2))
```

## 検定力

```
pwr.f2.test(u=dfuB1, v=dfv, f2=ef2B1, sig.level=有意水準)
pwr.f2.test(u=dfuB2, v=dfv, f2=ef2B2, sig.level=有意水準)
pwr.f2.test(u=dfuB1B2, v=dfv, f2=ef2B1B2, sig.level=有意水準)
```

## 標本サイズの推定

入力エリア Begin -----

## 水準数

```
nuB1 <- 被験者間要因1の水準数
nuB2 <- 被験者間要因2の水準数
```

## 偏イータ2乗(partial eta square)

```
peta2B1 <- 被験者間要因1の偏  $\eta^2$  乗
peta2B2 <- 被験者間要因2の偏  $\eta^2$  乗
peta2B1B2 <- 交互作用の偏  $\eta^2$  乗
```

入力エリア End -----

## 自由度

```
(dfuB1 <- (nuB1-1))
(dfuB2 <- (nuB2-1))
(dfuB1B2 <- dfuB1 * dfuB2)
```

## f2 統計量

```
(ef2B1 <- peta2B1/(1-peta2B1))
(ef2B2 <- peta2B2/(1-peta2B2))
(ef2B1B2 <- peta2B1B2/(1-peta2B1B2))
```



## 標本サイズ (1群あたりの標本サイズ)

```
(pf2t <- pwr.f2.test(u=dfuB1, f2=ef2B1, sig.level=有意水準, power=検定力))
(n <- ceiling((pf2t$v + nuB1*nuB2)/(nuB1*nuB2)))
```

```
(pf2t <- pwr.f2.test(u=dfuB2, f2=ef2B2, sig.level=有意水準, power=検定力))
(n <- ceiling((pf2t$v + nuB1*nuB2)/(nuB1*nuB2)))
```

```
(pf2t <- pwr.f2.test(u=dfuB1B2, f2=ef2B1B2, sig.level=有意水準, power=検定力))
(n <- ceiling((pf2t$v + nuB1*nuB2)/(nuB1*nuB2)))
```

## 参考目安

効果量 ( $\eta^2$ 乗)    小 : 0.01    中 : 0.06    大 : 0.14

```
> rm(list=ls())
>
> setwd("c:\\home\\Rdocuments\\scripts\\")
>
>
> # パッケージの読み込み
> library(DescTools)
> library("pwr")
>
>
> # 2B要因
>
> d1 <- read.table("2B平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> # 独立変数をfactor型に変換
> d1$id <- factor(d1$id)
> d1$work <- factor(d1$work, levels=c("many", "few"), labels=c("Many", "Few"))
> d1$future <- factor(d1$future, levels=c("shingaku", "shushoku", "mitei"),
+                       labels=c("Shingaku", "Shushoku", "Mitei"))
> head(d1)
  id work future tekiou
1  1  Few Shingaku    17
2  2  Few Shingaku    13
3  3  Few Shingaku    12
4  4  Few Shingaku    14
5  5  Few Shushoku    10
6  6  Few Shushoku    12
>
> # 分散分析
> mod.1 <- lm(tekiou~work*future, data=d1)
> result.aov <- aov(mod.1)
> summary(result.aov)
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
work	1	1.0	1.02	0.269	0.60448
future	2	213.6	106.81	28.294	1.16e-11 ***
work:future	2	38.1	19.04	5.043	0.00722 **
Residuals	220	830.5	3.78		

```
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
```

```

> # 効果量
> (eta2 <- EtaSq(result.aov))
      eta.sq  eta.sq.part
work      0.0003938638 0.0005134451
future    0.1972081460 0.2045909323
work:future 0.0351479090 0.0438332936
>
>
>
> # 検定力の計算
>
> # 入力エリア Begin -----
>
> # 自由度
> dfuB1 <- 1
> dfuB2 <- 2
> dfuB1B2 <- 2
> dfv <- 220
>
> # 偏イータ2乗(partial eta square)
> peta2B1 <- 0.0005134451
> peta2B2 <- 0.2045909323
> peta2B1B2 <- 0.0438332936
>
> # 入力エリア End -----
>
>
> # f2 統計量
> (ef2B1 <- peta2B1/(1-peta2B1))
[1] 0.0005137089
> (ef2B2 <- peta2B2/(1-peta2B2))
[1] 0.2572147
> (ef2B1B2 <- peta2B1B2/(1-peta2B1B2))
[1] 0.04584273
>
>
> # 検定力
> pwr.f2.test(u=dfuB1, v=dfv, f2=ef2B1, sig.level=0.05)

Multiple regression power calculation

      u = 1
      v = 220
      f2 = 0.0005137089
sig.level = 0.05
power = 0.0630488

> pwr.f2.test(u=dfuB2, v=dfv, f2=ef2B2, sig.level=0.05)

Multiple regression power calculation

      u = 2
      v = 220
      f2 = 0.2572147
sig.level = 0.05
power = 0.9999999

> pwr.f2.test(u=dfuB1B2, v=dfv, f2=ef2B1B2, sig.level=0.05)

Multiple regression power calculation

      u = 2
      v = 220
      f2 = 0.04584273
sig.level = 0.05
power = 0.8188361

```

```

> # 標本サイズの推定
>
> # 入力エリア Begin -----
>
> # 水準数
> nuB1 <- 2
> nuB2 <- 3
>
> # 偏イータ2乗(partial eta square)
> peta2B1 <- 0.06
> peta2B2 <- 0.06
> peta2B1B2 <- 0.06
>
> # 入力エリア End -----
>
> # 自由度
> (dfuB1 <- (nuB1-1))
[1] 1
> (dfuB2 <- (nuB2-1))
[1] 2
> (dfuB1B2 <- dfuB1 * dfuB2)
[1] 2
>
> # f2 統計量
> (ef2B1 <- peta2B1/(1-peta2B1))
[1] 0.06382979
> (ef2B2 <- peta2B2/(1-peta2B2))
[1] 0.06382979
> (ef2B1B2 <- peta2B1B2/(1-peta2B1B2))
[1] 0.06382979

> # 標本サイズ (1群あたりの標本サイズ)
> (pf2t <- pwr.f2.test(u=dfuB1, f2=ef2B1, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 1
      v = 122.9155
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling((pf2t$v + nuB1*nuB2)/(nuB1*nuB2)))
[1] 22
>
> (pf2t <- pwr.f2.test(u=dfuB2, f2=ef2B2, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 2
      v = 150.9791
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling((pf2t$v + nuB1*nuB2)/(nuB1*nuB2)))
[1] 27
>
> (pf2t <- pwr.f2.test(u=dfuB1B2, f2=ef2B1B2, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 2
      v = 150.9791
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling((pf2t$v + nuB1*nuB2)/(nuB1*nuB2)))
[1] 27

```

# 1群あたり, n=27

## 検定力分析 — 1つの被験者間要因と1つの被験者内要因 (1 B 1 W)

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

## 設定

分散分析表と効果量の推定を行って、以下の入力エリアの値（自由度、偏  $\eta^2$  乗、水準数）を得る。  
それらを入力エリアで設定する。

## 検定力の計算

入力エリア Begin -----

## 自由度

dfuB <- 被験者間要因の自由度  
dfvB <- 被験者間要因の残差の自由度

dfuW <- 被験者内要因の自由度  
dfuBW <- 交互作用の自由度  
dfvW <- 被験者内要因の残差の自由度

## 偏イータ2乗(partial eta square)

peta2B <- 被験者間要因の偏  $\eta^2$  乗  
peta2W <- 被験者内要因の偏  $\eta^2$  乗  
peta2BW <- 交互作用の偏  $\eta^2$  乗

入力エリア End -----

## f2 統計量

(ef2B <- peta2B/(1-peta2B))  
(ef2W <- peta2W/(1-peta2W))  
(ef2BW <- peta2BW/(1-peta2BW))

## 検定力

pwr.f2.test(u=dfuB, v=dfvB, f2=ef2B, sig.level=有意水準)  
pwr.f2.test(u=dfuW, v=dfvW, f2=ef2W, sig.level=有意水準)  
pwr.f2.test(u=dfuBW, v=dfvW, f2=ef2BW, sig.level=有意水準)

## 標本サイズの推定

入力エリア Begin -----

## 水準数

nuB <- 被験者間要因の水準数  
nuW <- 被験者内要因の水準数

## 偏イータ2乗(partial eta square)

peta2B <- 被験者間要因の偏  $\eta^2$  乗  
peta2W <- 被験者内要因の偏  $\eta^2$  乗  
peta2BW <- 交互作用の偏  $\eta^2$  乗

入力エリア End -----

**自由度**

```
(dfuB <- (nuB-1))
(dfuW <- (nuW-1))
(dfuBW <- dfuB * dfuW)
```

**f2 統計量**

```
(ef2B <- peta2B/(1-peta2B))
(ef2W <- peta2W/(1-peta2W))
(ef2BW <- peta2BW/(1-peta2BW))
```

**標本サイズ (1群あたりの標本サイズ)**

```
(pf2t <- pwr.f2.test(u=dfuB, f2=ef2B, sig.level=有意水準, power=検定力))
(n <- ceiling((pf2t$v + nuB)/nuB))
```

```
(pf2t <- pwr.f2.test(u=dfuW, f2=ef2W, sig.level=有意水準, power=検定力))
(n <- ceiling((pf2t$v/dfuW + nuB)/nuB))
```

```
(pf2t <- pwr.f2.test(u=dfuBW, f2=ef2BW, sig.level=有意水準, power=検定力))
(n <- ceiling((pf2t$v/dfuW + nuB)/nuB))
```

**参考目安**

効果量 ( $\eta^2$ 乗)    小 : 0.01    中 : 0.06    大 : 0.14

```
> rm(list=ls())
>
> setwd("c:\\home\\Rdocuments\\scripts\\")
>
>
> # パッケージの読み込み
> library(DescTools)
> library("pwr")
>
>
>
> # 1B1W要因
>
> d1 <- read.table("1B1W平均値データ_3時点.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
> # stackデータの作成
> d2 <- d1[,c("mh1", "mh2", "mh3")]
> d3 <- stack(d2)
> d4 <- data.frame(d1$id, d1$group, d3)
> colnames(d4) <- c("id", "group", "y", "x")
> head(d4)
  id group y x
1  1 experiment 11 mh1
2  2 experiment 18 mh1
3  3 experiment  9 mh1
4  4 experiment 17 mh1
5  5 experiment  9 mh1
6  6 experiment 12 mh1
>
```

```

> # 独立変数をfactor型に変換
> d4$id <- factor(d4$id)
> d4$group <- factor(d4$group)
> d4$x <- factor(d4$x)
>
>
>
>
> # 分散分析
> result.aov <- aov(y ~ group * x + Error(id + id:group + id:group:x), data=d4)

警告メッセージ:
aov(y ~ group * x + Error(id + id:group + id:group:x), data = d4) で:
  Error() モデルは特異です

> summary(result.aov)

Error: id
      Df Sum Sq Mean Sq F value    Pr(>F)
group    1  144.7   144.74   25.82 1.87e-06 ***
Residuals 95   532.6     5.61
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:group:x
      Df Sum Sq Mean Sq F value    Pr(>F)
x         2  307.9   153.93   36.83 3.03e-14 ***
group:x    2  123.4    61.70   14.76 1.10e-06 ***
Residuals 190   794.1     4.18
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
>
> # 効果量
> (eta2 <- EtaSq(result.aov, type=1))
      eta.sq eta.sq.part eta.sq.gen
group  0.07607493   0.2137043 0.09837276
x      0.16180455   0.2793801 0.18835006
group:x 0.06486216   0.1345093 0.08510749
>
>
>
> # 検定力の計算
>
> # 入力エリア Begin -----
>
> # 自由度
> dfuB <- 1
> dfvB <- 95
>
> dfuW <- 2
> dfuBW <- 2
> dfvW <- 190
>
>
> # 偏イータ2乗(partial eta square)
> peta2B <- 0.2137043
> peta2W <- 0.2793801
> peta2BW <- 0.1345093
>
> # 入力エリア End -----
>
>
> # f2 統計量
> (ef2B <- peta2B/(1-peta2B))
[1] 0.2717862
> (ef2W <- peta2W/(1-peta2W))
[1] 0.3876941
> (ef2BW <- peta2BW/(1-peta2BW))
[1] 0.1554139
>
>

```

```

> # 検定力
> pwr.f2.test(u=dfuB, v=dfvB, f2=ef2B, sig.level=0.05)

Multiple regression power calculation

      u = 1
      v = 95
      f2 = 0.2717862
sig.level = 0.05
power = 0.999102

> pwr.f2.test(u=dfuW, v=dfvW, f2=ef2W, sig.level=0.05)

Multiple regression power calculation

      u = 2
      v = 190
      f2 = 0.3876941
sig.level = 0.05
power = 1

> pwr.f2.test(u=dfuBW, v=dfvW, f2=ef2BW, sig.level=0.05)

Multiple regression power calculation

      u = 2
      v = 190
      f2 = 0.1554139
sig.level = 0.05
power = 0.9990991

>
>
>
>
>
>
> # 標本サイズの推定
>
> # 入力エリア Begin -----
>
> # 水準数
> nuB <- 2
> nuW <- 3
>
> # 偏イータ2乗(partial eta square)
> peta2B <- 0.06
> peta2W <- 0.06
> peta2BW <- 0.06
>
> # 入力エリア End -----
>
> # 自由度
> (dfuB <- (nuB-1))
[1] 1
> (dfuW <- (nuW-1))
[1] 2
> (dfuBW <- dfuB * dfuW)
[1] 2
>
> # f2 統計量
> (ef2B <- peta2B/(1-peta2B))
[1] 0.06382979
> (ef2W <- peta2W/(1-peta2W))
[1] 0.06382979
> (ef2BW <- peta2BW/(1-peta2BW))
[1] 0.06382979
>
>
>

```

```
> # 標本サイズ (1群あたりの標本サイズ)
> (pf2t <- pwr.f2.test(u=dfuB, f2=ef2B, sig.level=0.05, power=0.8))
```

Multiple regression power calculation

```
      u = 1
      v = 122.9155
      f2 = 0.06382979
sig.level = 0.05
power = 0.8
```

```
> (n <- ceiling((pf2t$v + nuB)/nuB))
[1] 63
```

```
>
> (pf2t <- pwr.f2.test(u=dfuW, f2=ef2W, sig.level=0.05, power=0.8))
```

Multiple regression power calculation

```
      u = 2
      v = 150.9791
      f2 = 0.06382979
sig.level = 0.05
power = 0.8
```

```
> (n <- ceiling((pf2t$v/dfuW + nuB)/nuB))
[1] 39
```

```
>
> (pf2t <- pwr.f2.test(u=dfuBW, f2=ef2BW, sig.level=0.05, power=0.8))
```

Multiple regression power calculation

```
      u = 2
      v = 150.9791
      f2 = 0.06382979
sig.level = 0.05
power = 0.8
```

```
> (n <- ceiling((pf2t$v/dfuW + nuB)/nuB))
[1] 39
```

# 1群あたり, n=39



## 検定力分析 — 2つの被験者内要因 (2W)

pwr パッケージの読み込み  
library("pwr")

あらかじめ pwr パッケージをインストールしておく必要がある。

## 設定

分散分析表と効果量の推定を行って、以下の入力エリアの値（自由度、偏  $\eta^2$  乗、水準数）を得る。  
それらを入力エリアで設定する。

## 検定力の計算

入力エリア Begin -----

## 自由度

dfuW1 <- 被験者内要因1の自由度  
dfuW2 <- 被験者内要因2の自由度  
dfuW1W2 <- 交互作用の自由度

dfvW1 <- 被験者内要因1の残差の自由度  
dfvW2 <- 被験者内要因2の残差の自由度  
dfvW1W2 <- 交互作用の残差の自由度

## 偏イータ2乗(partial eta square)

peta2W1 <- 被験者内要因1の偏  $\eta^2$  乗  
peta2W2 <- 被験者内要因2の偏  $\eta^2$  乗  
peta2W1W2 <- 交互作用の偏  $\eta^2$  乗

入力エリア End -----

## f2 統計量

(ef2W1 <- peta2W1/(1-peta2W1))  
(ef2W2 <- peta2W2/(1-peta2W2))  
(ef2W1W2 <- peta2W1W2/(1-peta2W1W2))

## 検定力

pwr.f2.test(u=dfuW1, v=dfv, f2=ef2W1, sig.level=有意水準)  
pwr.f2.test(u=dfuW2, v=dfv, f2=ef2W2, sig.level=有意水準)  
pwr.f2.test(u=dfuW1W2, v=dfv, f2=ef2W1W2, sig.level=有意水準)

## 標本サイズの推定

入力エリア Begin -----

## 水準数

nuW1 <- 被験者内要因1の水準数  
nuW2 <- 被験者内要因2の水準数

## 偏イータ2乗(partial eta square)

peta2W1 <- 被験者内要因1の偏  $\eta^2$  乗  
peta2W2 <- 被験者内要因2の偏  $\eta^2$  乗  
peta2W1W2 <- 交互作用の偏  $\eta^2$  乗

入力エリア End -----

**自由度**

```
(dfuW1 <- (nuW1-1))
(dfW2 <- (nuW2-1))
(dfW1W2 <- dfuW1 * dfuW2)
```

**f2 統計量**

```
(ef2W1 <- peta2W1/(1-peta2W1))
(ef2W2 <- peta2W2/(1-peta2W2))
(ef2W1W2 <- peta2W1W2/(1-peta2W1W2))
```

**標本サイズ (1群あたりの標本サイズ)**

```
(pf2t <- pwr.f2.test(u=dfuW1, f2=ef2W1, sig.level=有意水準, power=検定力))
(n <- ceiling(pf2t$v/dfuW1 + 1))
```

```
(pf2t <- pwr.f2.test(u=dfuW2, f2=ef2W2, sig.level=有意水準, power=検定力))
(n <- ceiling(pf2t$v/dfuW2 + 1))
```

```
(pf2t <- pwr.f2.test(u=dfuW1W2, f2=ef2W1W2, sig.level=有意水準, power=検定力))
(n <- ceiling(pf2t$v/(dfuW1*dfuW2) + 1))
```

**参考目安**

効果量 ( $\eta^2$ 乗)    小 : 0.01    中 : 0.06    大 : 0.14

```
> rm(list=ls())
>
> setwd("c:\\home\\Rdocuments\\scripts\\")
>
>
> # パッケージの読み込み
> library(DescTools)
> library("pwr")
>
>
>
> # 2W要因
>
> d1 <- read.table("2W平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d2 <- d1[-1]
> head(d2)
  cs cd ns nd
1 34 22 36 22
2 36 21 33 17
3 34 25 36 23
4 36 29 32 21
5 36 25 34 21
6 37 23 33 24
>
> #stackデータの作成
> d3 <- stack(d2)
> d3$x1 <- substr(d3$ind, 1, 1)
> d3$x2 <- substr(d3$ind, 2, 2)
> d3$id <- d1$id
> d4 <- d3[, c("id", "values", "x1", "x2")]
> colnames(d4) <- c("id", "y", "x1", "x2")
>
> # 独立変数をfactor型に変換
> d4$id <- factor(d4$id)
> d4$x1 <- factor(d4$x1)
> d4$x2 <- factor(d4$x2)
```

```

> head(d4)
  id y x1 x2
1  1 34  c  s
2  2 36  c  s
3  3 34  c  s
4  4 36  c  s
5  5 36  c  s
6  6 37  c  s
>
>
> # 分散分析
> result.aov <- aov(y ~ x1 * x2 + Error(id + id:x1 + id:x2 + id:x1:x2), data=d4)
> summary(result.aov)

Error: id
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals 137   1461    10.66

Error: id:x1
          Df Sum Sq Mean Sq F value Pr(>F)
x1          1    552   552.0   98.21 <2e-16 ***
Residuals 137    770     5.6
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:x2
          Df Sum Sq Mean Sq F value Pr(>F)
x2          1 17387  17387   3184 <2e-16 ***
Residuals 137   748     5
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: id:x1:x2
          Df Sum Sq Mean Sq F value Pr(>F)
x1:x2       1     0.1   0.065   0.014 0.905
Residuals 137  618.9   4.518
>
> # 効果量
> EtaSq(result.aov, type=1)
          eta.sq eta.sq.part eta.sq.gen
x1    2.563036e-02 0.4175491679 1.330144e-01
x2    8.073083e-01 0.9587517931 8.285468e-01
x1:x2 3.028162e-06 0.0001053593 1.812605e-05

> # 検定力の計算
>
> # 入力エリア Begin -----
>
> # 自由度
> dfuW1 <- 1
> dfuW2 <- 1
> dfuW1W2 <- 1
>
> dfvW1 <- 137
> dfvW2 <- 137
> dfvW1W2 <- 137
>
> # 偏イータ2乗(partial eta square)
> peta2W1 <- 0.4175491679
> peta2W2 <- 0.9587517931
> peta2W1W2 <- 0.0001053593
>
> # 入力エリア End -----
>
>

```

```

> # f2 統計量
> (ef2W1 <- peta2W1/(1-peta2W1))
[1] 0.7168831
> (ef2W2 <- peta2W2/(1-peta2W2))
[1] 23.24348
> (ef2W1W2 <- peta2W1W2/(1-peta2W1W2))
[1] 0.0001053704
>
>
> # 検定力
> pwr.f2.test(u=dfuW1, v=dfvW1, f2=ef2W1, sig.level=0.05)

Multiple regression power calculation

      u = 1
      v = 137
      f2 = 0.7168831
sig.level = 0.05
power = 1

> pwr.f2.test(u=dfuW2, v=dfvW2, f2=ef2W2, sig.level=0.05)

Multiple regression power calculation

      u = 1
      v = 137
      f2 = 23.24348
sig.level = 0.05
power = 1

> pwr.f2.test(u=dfuW1W2, v=dfvW1W2, f2=ef2W1W2, sig.level=0.05)

Multiple regression power calculation

      u = 1
      v = 137
      f2 = 0.0001053704
sig.level = 0.05
power = 0.05165605

> # 標本サイズの推定
>
> # 入力エリア Begin -----
>
> # 水準数
> nuW1 <- 2
> nuW2 <- 2
>
> # 偏イータ2乗(partial eta square)
> peta2W1 <- 0.06
> peta2W2 <- 0.06
> peta2W1W2 <- 0.06
>
> # 入力エリア End -----
>
> # 自由度
> (dfuW1 <- (nuW1-1))
[1] 1
> (dfuW2 <- (nuW2-1))
[1] 1
> (dfuW1W2 <- dfuW1 * dfuW2)
[1] 1
>
> # f2 統計量
> (ef2W1 <- peta2W1/(1-peta2W1))
[1] 0.06382979
> (ef2W2 <- peta2W2/(1-peta2W2))
[1] 0.06382979
> (ef2W1W2 <- peta2W1W2/(1-peta2W1W2))
[1] 0.06382979

```

```

> # 標本サイズ (1群あたりの標本サイズ)
> (pf2t <- pwr.f2.test(u=dfuW1, f2=ef2W1, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 1
      v = 122.9155
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling(pf2t$v/dfuW1 + 1))
[1] 124
>
> (pf2t <- pwr.f2.test(u=dfuW2, f2=ef2W2, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 1
      v = 122.9155
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling(pf2t$v/dfuW2 + 1))
[1] 124
>
> (pf2t <- pwr.f2.test(u=dfuW1W2, f2=ef2W1W2, sig.level=0.05, power=0.8))

Multiple regression power calculation

      u = 1
      v = 122.9155
      f2 = 0.06382979
sig.level = 0.05
power = 0.8

> (n <- ceiling(pf2t$v/(dfuW1*dfuW2) + 1))
[1] 124

```

# 1群あたり, n=124

## 21 信頼区間に基づく標本サイズの の推定

## 信頼区間に基づく標本サイズの推定 — 1 群の平均値

信頼区間に基づいて、1 群の平均値について推測する場合に、最低限必要な標本サイズを推定する。  
以下の自作関数を用いる。

```
# -----
# 標本サイズの推定 1 群の平均値

sampleN1m <- function(h=0.5, conf.level=0.95) {
  a2 <- 1 - (1-conf.level)/2
  n <- 2
  repeat{
    df <- n-1
    t1 <- qt(a2, df)
    tn <- t1*t1/(h*h)
    if (tn > n) n <- n+1
    if (tn <= n) break
  }
  nout <- matrix(c(h, conf.level, n), 1, 3)
  colnames(nout) <- c("h", "conf.level", "N")
  return(data.frame(nout))
}
# -----

# 標本サイズの推定 1 群の平均値
# h : 平均値の信頼区間の半幅 (デフォルトは0.5)
# conf.level : 信頼係数 (デフォルトは0.95)

> sampleN1m(h=0.5, conf.level=0.95)      # 信頼係数が0.95で良ければ,
      h conf.level  N                    # conf.level=0.95は省略できる
1 0.5      0.95  18

> sampleN1m(h=0.5, conf.level=0.90)
      h conf.level  N
1 0.5      0.9  13
```

## 信頼区間に基づく標本サイズの推定 — 2 群の平均値差

信頼区間に基づいて、2 群の平均値の差について推測する場合に、最低限必要な標本サイズを推定する。  
 対応のない場合と、対応のある場合がある  
 以下の自作関数を用いる。

```
# -----
# 標本サイズ 対応のない 2 群の平均値

sampleN2m <- function(h=0.5, conf.level=0.95) {
  a2 <- 1 - (1-conf.level)/2
  n <- 2
  repeat{
    df <- 2*(n-1)
    t0 <- qt(a2, df)
    tn <- 2*t0*t0/(h*h)
    if (tn > n) n <- n+1
    if (tn <= n) break
  }
  nout <- matrix(c(h, conf.level, n),1,3)
  colnames(nout) <- c("h", "conf.level", "N.per.group")
  return(data.frame(nout))
}
# -----

# 標本サイズ 対応のない 2 群の平均値
# h : 平均値の差の信頼区間の半幅 (デフォルトは0.5)
# conf.level : 信頼係数 (デフォルトは0.95)

> sampleN2m(h=0.5, conf.level=0.95)
      h conf.level N.per.group
1 0.5      0.95      32
```

```
# -----
# 標本サイズ 対応のある 2 群の平均値

sampleN2ma <- function(h=0.5, r=0, conf.level=0.95) {
  a2 <- 1 - (1-conf.level)/2
  n <- 2
  repeat{
    df <- n-1
    t0 <- qt(a2, df)
    tn <- 2*t0*t0*(1-r)/(h*h)
    if (tn > n) n <- n+1
    if (tn <= n) break
  }
  nout <- matrix(c(h, conf.level, n),1,3)
  colnames(nout) <- c("h", "conf.level", "N")
  return(data.frame(nout))
}
# -----

# 標本サイズ 対応のある 2 群の平均値
# h : 平均値の差の信頼区間の半幅 (デフォルトは0.5)
# r : 2 群のデータの相関係数 (デフォルトは0)
# conf.level : 信頼係数 (デフォルトは0.95)

> sampleN2ma(h=0.5, r=0.3, conf.level=0.95)
      h conf.level N
1 0.5      0.95 24
```



## 信頼区間に基づく標本サイズの推定 — 相関係数

信頼区間に基づいて、相関係数、および、対応のない2群の相関係数の差について推測する場合に、最低必要な標本サイズを推定する。

以下の自作関数を用いる。

```
# -----
# 標本サイズ 相関係数

sampleN1r <- function(r=0, h=0.1, conf.level=0.95) {
a2 <- 1 - (1-conf.level)/2
n <- 4
repeat{
  t0 <- qnorm(a2)
  se <- 1/sqrt(n-3)
  lz <- atanh(r)-t0*se
  uz <- atanh(r)+t0*se
  elz <- tanh(lz)
  euz <- tanh(uz)
  hci <- (euz-elz)/2
  if (hci > h) n <- n+1
  if (hci <= h) break
}
nout <- matrix(c(r, h, conf.level, n),1,4)
colnames(nout) <- c("r", "h", "conf.level", "N")
return(data.frame(nout))
}
# -----

# 標本サイズ 相関係数
# r : 相関係数の推定値 (デフォルトは0)
# h : 相関係数の信頼区間の半幅 (デフォルトは0.1)
# conf.level : 信頼係数 (デフォルトは0.95)

> sampleN1r(r=0.4, h=0.1, conf.level=0.95)
   r   h conf.level   N
1 0.4 0.1      0.95 273

# -----
# 標本サイズ 対応のない2群の相関係数

sampleN2r <- function(dr=0, h=0.1, conf.level=0.95) {
a2 <- 1 - (1-conf.level)/2
n <- 4
repeat{
  t0 <- qnorm(a2)
  se <- sqrt(2/(n-3))
  lz <- atanh(dr)-t0*se
  uz <- atanh(dr)+t0*se
  elz <- tanh(lz)
  euz <- tanh(uz)
  hci <- (euz-elz)/2
  if (hci > h) n <- n+1
  if (hci <= h) break
}
nout <- matrix(c(dr, h, conf.level, n),1,4)
colnames(nout) <- c("dr", "h", "conf.level", "N.per.group")
return(data.frame(nout))
}
# -----

# 標本サイズ 対応のない2群の相関係数
# dr : 相関係数の差の推定値 (デフォルトは0)
# h : 相関係数の差の信頼区間の半幅 (デフォルトは0.1)
# conf.level : 信頼係数 (デフォルトは0.95)

> sampleN2r(dr=0.2, h=0.1, conf.level=0.95)
   dr   h conf.level N.per.group
1 0.2 0.1      0.95          707
```

## 信頼区間に基づく標本サイズの推定 — 比率

信頼区間に基づいて、1 群の比率、および、対応のない 2 群の比率の差について推測する場合に、最低限必要な標本サイズを推定する。

以下の自作関数を用いる。

```
# -----
# 標本サイズ 1 群の比率

sampleN1p <- function(p=0.5, h=0.1, conf.level=0.95) {
  a2 <- 1 - (1-conf.level)/2
  n <- 2
  repeat{
    t0 <- qnorm(a2)
    tn <- t0*t0*p*(1-p)/(h*h)
    if (tn > n) n <- n+1
    if (tn <= n) break
  }
  nout <- matrix(c(p, h, conf.level, n), 1, 4)
  colnames(nout) <- c("p", "h", "conf.level", "N")
  return(data.frame(nout))
}
# -----
```

```
# 標本サイズ 1 群の比率
# p : 比率の推定値 (デフォルトは0.5)
# h : 比率の信頼区間の半幅 (デフォルトは0.1)
# conf.level : 信頼係数 (デフォルトは0.95)
```

```
> sampleN1p(p=0.4, h=0.1, conf.level=0.95)
      p      h conf.level      N
1 0.4 0.1      0.95 93
```

```
# -----
# 標本サイズ 対応のない 2 群の比率

sampleN2p <- function(p1=0.5, p2=0.5, h=0.1, conf.level=0.95) {
  a2 <- 1 - (1-conf.level)/2
  n <- 2
  repeat{
    t0 <- qnorm(a2)
    tn <- t0*t0*(p1*(1-p1)+p2*(1-p2))/(h*h)
    if (tn > n) n <- n+1
    if (tn <= n) break
  }
  nout <- matrix(c(p1, p2, h, conf.level, n), 1, 5)
  colnames(nout) <- c("p1", "p2", "h", "conf.level", "N.per.group")
  return(data.frame(nout))
}
# -----
```

```
# 標本サイズ 対応のない 2 群の比率
# p1 : 第 1 群の比率の推定値 (デフォルトは0.5)
# p2 : 第 2 群の比率の推定値 (デフォルトは0.5)
# h : 比率の差の信頼区間の半幅 (デフォルトは0.1)
# conf.level : 信頼係数 (デフォルトは0.95)
```

```
> sampleN2p(h=0.1, p1=0.1, conf.level=0.95)
      p1      p2      h conf.level N.per.group
1 0.1 0.5 0.1      0.95      131
```

```
# p2の設定を省略しているので、
# デフォルト値0.5が用いられる
```

## 22 項目分析

## 項目分析 — 多肢選択式解答データ

## 自作スクリプトを使って、多肢選択項目の項目分析を実行する

トレースライン作成にあたっては、群サイズが均等になるように群を構成している。同じ得点でも入る群が異なることがある。

3群の場合は、Kelley=TRUEとすると、下位27%、中位46%、上位27%と受験者を群わけする。

## 複数選択に対応

1つの項目において、選択枝を正しく選択できたら正答、それ以外は誤答とする。正答選択枝は複数あってもよく、すべて正しく選択できた場合のみ正答とする。解答の順序は逆順になっていてもよい。

## 解答データファイル

1行に一人の受験者、1列に1つの変数が入力されているcsvファイルとする。

無解答は空白にする。

解答データは、1, 2, 3のような数値でも、A, B, Cのような記号でも、どちらでも良い。

1行目は変数名とする。

2行目以降、各受験者のデータを入力する。

1列目は必ず受験番号 (ID) とする。

2列目から属性変数を設定する (省略可)。

属性変数の後にテスト項目を設定する。最終テスト項目より右には何も入力しない。

複数の選択枝を選択している場合は、同一セル内に、「,」（半角カンマ）で区切って入力する。

## 解答データの例

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	NO.	School	Grade	Type	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
2	1	1	1	1	A,B	A,C	A,D	B	A	D	C	C	B	D
3	2	1	2	2	A,B	A,C	A,D	B	A	D	C	C	B	D
4	3	1	2	3	A,B	A,C	A,D	B	A	D	C	A	B	D
5	4	1	3	2	A,B	A,C	A,D	B	A	D	C	B	A	D
6	5	2	1	1	A,B	A,C	A,D	B	A	D	C	D	B	D
7	6	2	2	3	A,B	A,C	A,D	B	A	D	C	C	B	A
8	7	2	2	1	A,B	A,C	A,D	B	A	D	C	C	B	
9	8	1	1	1	A,B	A,C	A,D	B	A	D	C	A	C	D
10	9	1	1	1	B,A	C,A	D,A	B	A	B	C	B	B	D
11	10	1	1	1	B,A	C,A	D,A	B	A	B	C	D	B	D

## 正答キーファイル

1行目に項目名、2行目に正答記号の入ったcsvファイルとする。

1列名は、1行目に「科目」、2行目に科目名を入れておくといよい。

項名は、解答データと同一でなければならない。

正答枝が複数ある場合は、同一セル内に、「,」（半角カンマ）で区切って入力する。

## 正答キーデータの例

	A	B	C	D	E	F	G	H	I	J	K
1	TEST	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
2	Sample	A,B	A,C	D,A	B	A	D	C	C	B	D

## スクリプトで設定すべきもの

# Data File Name (解答データ)

ansFileName &lt;- "解答データファイル名(.csv)"

# Key File name (正答キーデータ)

keyFileName &lt;- "正答キーデータファイル名(.csv)"

# Column Number of First Test Item: (テスト項目が始まる列の番号(数字))

firstItemCol &lt;- 5

# Option Categories (選択枝に用いた記号)

OPTIONS &lt;- c("A", "B", "C", "D")

OPTIONS &lt;- c("0", "1")

**【重要】** 0/1データを分析するときなど、数字データも、数値ではなく文字として扱う。

# Number of groups in trace line (トレースラインを描くときの群の数)

NGROUP &lt;- 3

# Following Kelly's percentages (Kelleyの基準を使うか否か)

KELLEY &lt;- TRUE

# Score output file name (0/1採点したデータを保存するファイル名)

scoFileName &lt;- "ScoreData.csv"

# Total Score Summary output file name (テスト得点の記述統計量を保存するファイル名)

totFileName &lt;- "TotalSummary.csv"

# Item Analysis output file name (全項目の項目分析の結果を保存するファイル名)

itaFileName &lt;- "ItemAnalysis.csv"

# Result of Each Item output file name (各項目の詳細な分析結果を保存するファイル名)

reiFileName &lt;- "ResultEachItem.csv"

# Trace line data output file name (トレースラインの座標データを保存するファイル名)

tldFileName &lt;- "TraceLineData.csv"

# Trace line output file name (トレースラインの図を保存するPDFファイル名)

tlgFileName &lt;- "TraceLine.pdf"

## 出力ファイル

以下の6個のファイルを出力

0/1採点した得点データ

テスト得点の記述統計量

全項目の項目分析の結果

各項目の項目分析の詳細な結果

トレースラインの座標データ

トレースラインの図

## 得点データ

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	NO.	School	Grade	Type	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Total	Prop
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	1
3	2	1	2	2	1	1	1	1	1	1	1	1	1	1	10	1
4	3	1	2	3	1	1	1	1	1	1	1	0	1	1	9	0.9
5	4	1	3	2	1	1	1	1	1	1	1	0	0	1	8	0.8
6	5	2	1	1	1	1	1	1	1	1	1	0	1	1	9	0.9
7	6	2	2	3	1	1	1	1	1	1	1	1	1	0	9	0.9
8	7	2	2	1	1	1	1	1	1	1	1	1	1	0	9	0.9
9	8	1	1	1	1	1	1	1	1	1	1	0	0	1	8	0.8
10	9	1	1	1	1	1	1	1	1	0	1	0	1	1	8	0.8
11	10	1	1	1	1	1	1	1	1	0	1	0	1	1	8	0.8

各項目：各項目の0/1得点

Total：合計得点

Prop：合計得点の得点率

## テスト合計得点および得点率の記述統計量

	A	B	C	D	E	F	G	H	I
1	Score	Items	N	Mean	SD	Min	Med	Max	Alpha
2	Total	10	100	4.76	2.39	1	4	10	0.73
3	Prop	10	100	0.48	0.24	0.1	0.4	1	0.73

Items：項目数

N：受験者数

Mean：平均値

SD：標準偏差

Min：最小値

Med：中央値

Max：最大値

Alpha：アルファ係数

## 全項目の項目分析の結果

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Item	N	P	D	Corr	Alpha	Dalpha	Key	A	B	C	D	NA
2	X1	100	0.21	0.78	0.73	0.73	0.66	A,B	0.48	0.53	0.43	0.48	0.08
3	X2	100	0.23	0.85	0.74	0.73	0.65	A,C	0.48	0.41	0.55	0.48	0.08
4	X3	100	0.19	0.7	0.7	0.73	0.66	D,A	0.48	0.45	0.48	0.51	0.08
5	X4	100	0.18	0.67	0.68	0.73	0.67	B	0.54	0.18	0.18	0.05	0.05
6	X5	100	0.71	0.89	0.52	0.73	0.69	A	0.71	0.14	0.1	0.04	0.01
7	X6	100	0.5	0.44	0.16	0.73	0.75	D	0.1	0.18	0.19	0.5	0.03
8	X7	100	0.79	0.7	0.42	0.73	0.7	C	0.06	0.07	0.79	0.07	0.01
9	X8	100	0.64	0	-0.14	0.73	0.79	C	0.09	0.1	0.64	0.13	0.04
10	X9	100	0.58	0.52	0.26	0.73	0.73	B	0.13	0.58	0.13	0.12	0.04
11	X10	100	0.73	0.41	0.16	0.73	0.74	D	0.07	0.05	0.09	0.73	0.06

Item：項目名

N：受験者数

P：正答率

D：D値（上位群の正答率－下位群の正答率）

Corr：I-T相関係数（項目得点と、合計点から当該項目の得点を引いた得点との相関係数）

Alpha：アルファ係数（テスト全体のアルファ係数なので、全項目に同じ値が入っている）

Dalpha：当該項目を削除したときのアルファ係数

Key：正答キー

各選択枝：当該選択枝の選択率

NA：無答率

## 各項目の項目分析の詳細な結果

	A	B	C	D	E	F	G	H
1	Item	Key	N	P	D	Corr	Alpha	DAlpha
2	X1	A,B	100	0.21	0.78	0.73	0.73	0.66
3	Prop	A	B	C	D	NA		
4	All	0.48	0.53	0.43	0.48	0.08		
5	U	0.85	0.93	0.22	0	0		
6	M	0.52	0.43	0.5	0.54	0		
7	L	0.04	0.3	0.52	0.85	0.3		
8	D	0.81	0.63	-0.3	-0.85	-0.3		
9	Corr	0.64	0.42	-0.24	-0.62	-0.25		
10								

Item: 項目名

Key: 正答キー

N: 受験者数

P: 正答率

D: D値 (上位群の正答率 - 下位群の正答率)

Corr: I-T相関係数 (項目得点と、合計点から当該項目の得点を引いた得点との相関係数)

Alpha: アルファ係数 (テスト全体のアルファ係数なので、全項目に同じ値が入っている)

Dalpha: 当該項目を削除したときのアルファ係数

選択率

All: 受験者全体

U: 上位群

M: 中位群

L: 下位群

Missing: 無答率

識別力

D: 当該選択枝のD値 (上位群の選択率 - 下位群の選択率)

Corr: 当該選択枝のI-T相関係数

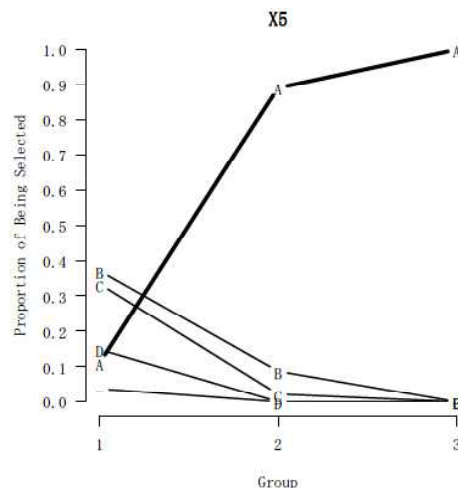
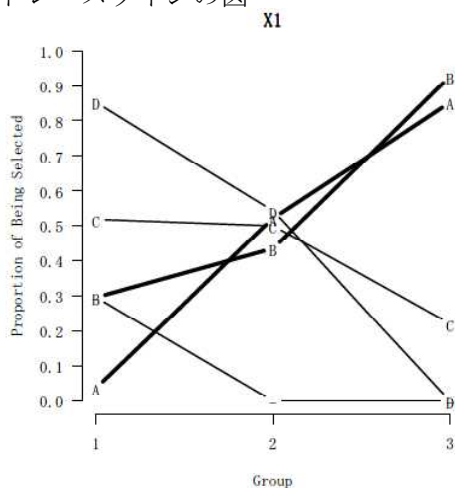
(当該選択枝の選択の有無と、合計点から当該項目の得点を引いた得点との相関係数)

## トレースラインの座標データ

Item	Option	1	2	3
X1	A	0.037037	0.521739	0.851852
X1	B	0.296296	0.434783	0.925926
X1	C	0.518519	0.5	0.222222
X1	D	0.851852	0.543478	0
X1	NA	0.296296	0	0
X2	A	0.037037	0.521739	0.851852

X5	A	0.111111	0.891304	1
X5	B	0.37037	0.086957	0
X5	C	0.333333	0.021739	0
X5	D	0.148148	0	0
X5	NA	0.037037	0	0

## トレースラインの図



太線: 正答枝

細線: 誤答枝

-: 無回答

**関数**

設定すべき値を指定して、以下のスクリプト全体を実行する。

以下のサイトからダウンロード可能

[https://www.educa.nagoya-u.ac.jp/~ishii-h/test\\_system.html](https://www.educa.nagoya-u.ac.jp/~ishii-h/test_system.html)

```
# Item Analysis

# Memory clear
rm(list=ls())

# Set work directory
setwd("e:¥¥")

# Data File Name
ansFileName <- "SampleResponseData2(MC).csv"

# Key File name
keyFileName <- "SampleAnswerKey(MC).csv"

# Column Number of First Test Item:
firstItemCol <- 5

# Option Categories
OPTIONS <- c("A", "B", "C", "D")

# OPTIONS <- c("0", "1")

# Number of groups in trace line
NGROUP <- 3

# Following Kelly's percentages
KELLEY <- TRUE

# Score output file name
scoFileName <- "ScoreData.csv"

# Total Score Summary output file name
totFileName <- "TotalSummary.csv"

# Item Analysis output file name
itaFileName <- "ItemAnalysis.csv"

# Result of Each Item output file name
reiFileName <- "ResultEachItem.csv"

# Trace line data output file name
tldFileName <- "TraceLineData.csv"

# Trace line graph output file name
tlgFileName <- "TraceLineGraph.pdf"

#-----
# Basically, no change is necessary in the following.
#-----

# Digits
rdgts <- 2
```



```

# -----

# Data import
dans <- read.table(ansFileName, header=T, sep="," , fileEncoding="shift-jis")
dkey <- read.table(keyFileName, header=T, sep="," , fileEncoding="shift-jis")

# Number of Items
nItem <- ncol(dans) - firstItemCol + 1

# Number of OPTIONS
nOption <- length(OPTIONS)

# Sample size
nSample <- nrow(dans)

# Item names
itemNames <- colnames(dans)[firstItemCol : ncol(dans)]

# ID variable name
idName <- colnames(dans)[1]

# Score matrix
score <- as.data.frame(matrix(c(0), nrow=nSample, ncol=nItem ))
colnames(score) <- c(itemNames)

# Scoring
for(i in itemNames){
  dkeytmp <- rep(0, nOption)
  names(dkeytmp) <- OPTIONS
  for(j in OPTIONS){
    if (grepl(j,dkey[,i])) dkeytmp[j] <- 1
  }
  dansitmp <- matrix(c(0), nrow=nSample, ncol=nOption)
  colnames(dansitmp) <- OPTIONS
  for(j in OPTIONS){
    dansitmp[,j] <- ifelse(grepl(j, dans[,i]), 1, 0)
  }
  score[,i] <- ifelse((apply(dansitmp, 1, paste, collapse="")==paste(dkeytmp, collapse="")), 1, 0)
}

# Total scores
score$Total <- rowSums(score[,itemNames])
score$Prop <- floor(100 * (rowMeans(score[,itemNames]) +0.005)) / 100

# Score Data
dscore <- data.frame(dans[,c(1:(firstItemCol-1))], score)

# Output Score data
write.table(dscore, scoFileName, row.names=F, sep="," )

# Total Scores Summary
tot <- prop <- NULL
tot$Score <- "Total"
tot$Items <- length(itemNames)
tot$N <- nSample
tot$Mean <- floor(10^rdgts * (mean(score[, "Total"])+0.005))/10^rdgts
tot$SD <- floor(10^rdgts * ( sd(score[, "Total"])+0.005))/10^rdgts
tot$Min <- min(score[, "Total"])
tot$Med <- median(score[, "Total"])
tot$Max <- max(score[, "Total"])
tot$Alpha <- floor(10^rdgts * (length(itemNames)/(length(itemNames)-1) *
                        (1-sum(apply(score[, itemNames], 2, var))/var(score[, "Total"])+0.005)) / 10
^rdgts
prop$Score <- "Prop"
prop$Items <- tot$Items
prop$N <- tot$N
prop$Mean <- floor(10^rdgts * (mean(score[, "Total"])/prop$Items+0.005))/10^rdgts

```

```

prop$SD <- floor(10^rdgts * ( sd(score[, "Total"])/prop$Items+0.005))/10^rdgts
prop$Min <- min(score[, "Prop"])
prop$Med <- median(score[, "Prop"])
prop$Max <- max(score[, "Prop"])
prop$Alpha <- tot$Alpha

totout <- as.data.frame(tot)
propout <- as.data.frame(prop)
tpout <- rbind(totout, propout)

# Output Total score summary
write.table(tpout, totFileName, row.names=F, sep=",")

#remove tpout, totout, propout, prop
rm(tpout, totout, propout, prop)

# Item analysis
optionNames <- paste("o", OPTIONS, sep="")
itaNames <- c("Item", "N", "P", "D", "Corr", "Alpha", "Dalpha", "Key", optionNames, "oNA")
dita <- matrix(c(0), nrow=length(itemNames), ncol=length(itaNames))
Corr <- rep(0, length(itemNames))
names(Corr) <- itemNames
rownames(dita) <- itemNames
colnames(dita) <- itaNames

dita[, "Item"] <- itemNames
dita[, "N"] <- tot$N
dita[, "P"] <- floor(10^rdgts * (colMeans(score[, itemNames])+0.005))/ 10^rdgts
dita[, "Alpha"] <- tot$Alpha

for(i in itemNames){
  Corr[i] <- cor(score[, i], (score[, "Total"]-score[, i]))
  dita[i, "Corr"] <- floor(10^rdgts * (Corr[i]+0.005))/ 10^rdgts
  scoreDi <- score[, itemNames[(itemNames %in% i)==F]]
  dita[i, "Dalpha"] <- floor(10^rdgts * (ncol(scoreDi)/(ncol(scoreDi)-1) *
    (1-sum(apply(scoreDi, 2, var))/var(rowSums(scoreDi)))+0.005))/ 10^rdg
ts
}
dita[, "Key"] <- as.character(t(dkey[, itemNames]))

# Data Sorting
vCorr <- Corr[Corr>0]
sNames <- names(vCorr)[order(vCorr, decreasing=T)]
sNames <- paste("dscore$", sNames, sep="", collapse=" ")
eval(parse(text=paste("dsort <- dscore[order(dscore$Total", sNames, ", dscore[, 1]),]", sep="") ))
dsort$sortID <- c(1:nrow(dsort))

# remove score, scoreDi, dscore, Corr
rm(score, scoreDi, dscore, Corr)

# Subgroup size
nSub <- ceiling(nSample * 0.27)

# Subgroup label
dsort$LMU <- ifelse(dsort[, "sortID"]<=nSub, "L",
  ifelse((dsort[, "sortID"]>nSub)&(dsort[, "sortID"]<=(nSample-nSub)), "M", "U"))

# D index
dsortL <- dsort[dsort$LMU=="L", ]
dsortU <- dsort[dsort$LMU=="U", ]
dita[, "D"] <- floor(10^rdgts * (colMeans(dsortU[, itemNames]) -
  colMeans(dsortL[, itemNames])+0.005))/ 10^rdgts

# remove dsortU, dsortL
rm(dsortU)

```

```

rm(dsorL)

# Each Item
drei <- NULL
v1 <- c("Item", "Key", "N", "P", "D", "Corr", "Alpha", "DAlpha")

for(i in itemNames){
  # subjects*options matrix per item
  dkeytmp <- rep(0, nOption)
  names(dkeytmp) <- OPTIONS
  for(j in OPTIONS){
    if (grepl(j, dkey[,i])) dkeytmp[j] <- 1
  }
  dansitmp <- matrix(c(0), nrow=nSample, ncol=(nOption+1))
  colnames(dansitmp) <- c(OPTIONS, "NA")
  for(j in OPTIONS){
    dansitmp[,j] <- ifelse(grepl(j, dans[,i]), 1, 0)
  }
  dansitmp[, "NA"] <- ifelse((sum(dkeytmp) - rowSums(dansitmp))>=0, (sum(dkeytmp) - rowSums(dansitmp)), 0)

  # choice ratio of each option per item
  dita[i, c(optionNames, "oNA")] <- floor(10^rdgts * (colMeans(dansitmp)+0.005)) / 10^rdgts

  # each item analysis
  v2 <- c(i, as.character(dkey[1,i]), dita[i, "N"], dita[i, "P"], dita[i, "D"], dita[i, "Corr"], dita[i, "Alpha"], dita[i, "Dalpha"])
  v3 <- c("Prop", OPTIONS, "NA")
  v4 <- c("All", dita[i, c(optionNames, "oNA")])

  # UML based
  dsortitmp <- dsort[order(dsorL[, idName]), c(idName, i, "Total", "LMU")]
  dansitmp <- cbind(dsortitmp, dansitmp)
  dansitmpL <- dansitmp[dansitmp[, "LMU"]=="L", ]
  dansitmpM <- dansitmp[dansitmp[, "LMU"]=="M", ]
  dansitmpU <- dansitmp[dansitmp[, "LMU"]=="U", ]
  v5 <- c("U", (floor(10^rdgts * (colMeans(dansitmpU[, c(OPTIONS, "NA")]) + 0.005)) / 10^rdgts))
  v6 <- c("M", (floor(10^rdgts * (colMeans(dansitmpM[, c(OPTIONS, "NA")]) + 0.005)) / 10^rdgts))
  v7 <- c("L", (floor(10^rdgts * (colMeans(dansitmpL[, c(OPTIONS, "NA")]) + 0.005)) / 10^rdgts))
  v8 <- c("D", (floor(10^rdgts * (colMeans(dansitmpU[, c(OPTIONS, "NA")]) - colMeans(dansitmpL[, c(OPTIONS, "NA")]) + 0.005)) / 10^rdgts))
  v9 <- c("Corr", (floor(10^rdgts * (cor(dansitmp[, c(OPTIONS, "NA")], (dansitmp[, "Total"] - dansitmp[, i]), use = "pairwise.complete.obs") + 0.005)) / 10^rdgts))

  v10 <- rep("", length(v5))

  nemp <- length(v3) - length(v1)
  if (nemp >= 0) {
    v1 <- c(v1, rep("", nemp))
    v2 <- c(v2, rep("", nemp))
  } else {
    nemp <- abs(nemp)
    v3 <- c(v3, rep("", nemp))
    v4 <- c(v4, rep("", nemp))
    v5 <- c(v5, rep("", nemp))
    v6 <- c(v6, rep("", nemp))
    v7 <- c(v7, rep("", nemp))
    v8 <- c(v8, rep("", nemp))
    v9 <- c(v9, rep("", nemp))
    v10 <- c(v10, rep("", nemp))
  }
  temp <- rbind(v1, v2, v3, v4, v5, v6, v7, v8, v9, v10)
  drei <- rbind(drei, temp)
}

#remove dansitmpL, dansitmpM, dansitmpU
rm(dansitmpL, dansitmpM, dansitmpU)

#remove v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
rm(v1, v2, v3, v4, v5, v6, v7, v8, v9, v10)

```

```

# Output Item Analysis Results
colnames(dita) <- c("Item", "N", "P", "D", "Corr", "Alpha", "Dalpha", "Key", OPTIONS, "NA")
write.table(dita, itaFileName, row.names=F, sep=",")
write.table(drei, reiFileName, row.names=F, col.names=F, sep=",")

#remove dita, drei
rm(dita, drei)

# Trace lines

pdf(tlgFileName, paper="a4", width=7, height=14, family="Japan1", onefile=T)
  layout(matrix(c(1,2,3,4,5,6), 3,2, byrow=T))
par(mar=c(6,5,2,1))

tlData <- NULL

for(i in itemNames){
  # subjects*options matrix per item
  dkeytmp <- rep(0, nOption)
  names(dkeytmp) <- OPTIONS
  for(j in OPTIONS){
    if (grepl(j, dkey[,i])) dkeytmp[j] <- 1
  }
  dansitmp <- matrix(c(0), nrow=nSample, ncol=(nOption+1))
  colnames(dansitmp) <- c(OPTIONS, "NA")
  for(j in OPTIONS){
    dansitmp[,j] <- ifelse(grepl(j, dans[,i]), 1, 0)
  }
  dansitmp[, "NA"] <- ifelse((sum(dkeytmp) - rowSums(dansitmp))>=0, (sum(dkeytmp) - rowSums(dansitm
p)), 0)

  dsortitmp <- dsort[order(dsort[,idName]),c(idName,"LMU","sortID")]
  dansitmp <- cbind(dsortitmp, dansitmp)
  dansitmp <- dansitmp[order(dansitmp[, "sortID"]),]

  dansitmp$group <- NGROUP
  if ((NGROUP==3) && (KELLEY==TRUE)) {
    dansitmp$group[1:nSub] <- 1
    dansitmp$group[(nSub+1):(nSample-nSub)] <- 2
    dansitmp$group[(nSample-nSub+1):nSample] <- 3
  } else {
    nEach <- floor(nSample/NGROUP +0.5)
    for(igp in 1:(NGROUP-1)){
      hb <- (igp-1)*nEach+1
      he <- igp*nEach
      dansitmp$group[hb:he] <- igp
    }
  }

  ioprop <- matrix(c(0), nrow=(nOption+1), ncol=NGROUP)
  rownames(ioprop) <- c(OPTIONS, "NA")
  colnames(ioprop) <- c(1:NGROUP)

  lwdv <- rep(1, (nOption+1))
  lwdv <- 1.5*c(dkeytmp, 0) + lwdv
  names(lwdv) <- c(OPTIONS, "NA")

  for(j in 1:NGROUP){
    dansitmpj <- dansitmp[dansitmp$group==j,]
    ioprop[,j] <- colMeans(dansitmpj[,c(OPTIONS, "NA")])
  }

  plot(c(1:NGROUP), ioprop["NA",], xlim=c(1, NGROUP), ylim =c(0,1), las=1, axes=F,
        xlab="Group", ylab="Proportion of Being Selected",
        pch="-", type="b", lty=1, lwd=lwdv["NA"], main=i)
  axis(side=1, c(1:NGROUP))
  axis(side=2, seq(0,1, 0.1), las=1)

```

```

for (j in OPTIONS) {
  par(new=T)
  plot(c(1:NGROUP), ioprop[j,], xlim=c(1, NGROUP), ylim=c(0,1), las=1, axes=F,
       pch=j, type="b", lty=1, lwd=lwdv[j], xlab="", ylab="", xaxt="n", yaxt="n")
}

iopropi <- data.frame(i, rownames(ioprop), ioprop)
tlData <- rbind(tlData, iopropi)
}

dev.off()

# Output Trace Line Data
colnames(tlData) <- c("Item", "Option", c(1:NGROUP))

write.table(tlData, tldFileName, row.names=F, sep=",")

```

## 項目分析 — 解答類型データ

自作スクリプトを使って、解答類型に従って評定したデータの項目分析を実行する

多肢選択式択一問題、準正答のある多肢選択問題の項目分析に利用することもできる。

トレースライン作成にあたっては、群サイズが均等になるように群を構成している。同じ得点でも入る群が異なることがある。

3群の場合は、Kelley=TRUEとすると、下位27%、中位46%、上位27%と受験者を群わけする。

## 項目別の解答類型に対応

項目において、解答類型と評点を設定する。解答類型及び評点は項目ごとに異なってもよいが、評定記号はなるべく共通にしたほうが良い。

## 解答データファイル

1行に一人の受験者、1列に1つの変数が入力されているcsvファイルとする。

無解答は空白にする。

解答類型データは、1, 2, 3のような数値でも、A, B, Cのような記号でも、どちらでも良い。

1行目は変数名とする。

2行目以降、各受験者のデータを入力する。

1列目は必ず受験番号 (ID) とする。

2列目から属性変数を設定する (省略可)。

属性変数の後にテスト項目を設定する。最終テスト項目より右には何も入力しない。

## 解答データの例

1	NO.	School	Grade	Type	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10
2	1	1	1	1	1	1	1	1	1	1	1	1	1	1
3	2	1	2	2	2	1	1	1	9	1	1	1	1	9
4	3	1	2	3	1	1	1	1	1	1	1	1	1	1
5	4	1	3	2	1	1	2	3	1	1	1	1	1	1
6	5	2	1	1	1	1	1	1	1	1	1	1	1	1
7	6	2	2	3	3	9	9	3	9	9	2	3	1	4

## 類型キーファイル

1行目は変数名とする。項目名変数はItem、類型データか得点データかの別を表す変数はCSとする。

1列目に項目名を入力する。項目ごとに類型と評点の2行を使うので、項目名は2回繰り返して次に進む。

項目名は、解答データと同一でなければならない。

2列目に類型 (Category) か、得点 (Score) かの別をC, Sで入力する。

3列目以降に、各項目のC行に類型、S行に評点を入力する。当該項目に類型がない場合は空欄とする。

項目ごとに選択枝と評点を設定すれば、択一式または準正答のある多肢選択式項目の分析にも使える。

## 正答キーデータの例

Item	CS	CS1	CS2	CS3	CS4	CS9	CS0
X1	C	1	2	3	4	9	0
X1	S	1	0.5	0	0	0	0
X2	C	1				9	0
X2	S	1				0	0
X3	C	1	2			9	0
X3	S	1	0.5			0	0
X4	C	1	2	3		9	0
X4	S	1	0.5	0		0	0
X5	C	1	2	3	4	9	0
X5	S	1	0.5	0	0	0	0
X6	C	1	2	3	4	9	0
X6	S	1	0.5	0	0	0	0
X7	C	1	2	3	4	9	0
X7	S	1	0.5	0.5	0	0	0
X8	C	1	2	3	4	9	0
X8	S	1	0.5	0.5	0.3	0	0
X9	C	1	2	3	4	9	0
X9	S	1	0.7	0.5	0.3	0	0
X10	C	1	2	3	4	9	0
X10	S	1	1	0.5	0.5	0	0

## スクリプトで設定すべきもの

# Data File Name (解答データ)

ansFileName &lt;- "解答データファイル名(.csv)"

# Key File name (正答キーデータ)

keyFileName &lt;- "正答キーデータファイル名(.csv)"

# Column Number of First Test Item: (テスト項目が始まる列の番号(数字))

firstItemCol &lt;- 5

# Option Categories (解答類型に用いた記号)

OPTIONS &lt;- c("1", "2", "3", "4", "9", "0")

OPTIONS &lt;- c("A", "B", "C", "D", "E", "F")

**【重要】** 0/1データを分析するときなど、数字データも、数値ではなく文字として扱う。

# Number of groups in trace line (トレースラインを描くときの群の数)

NGROUP &lt;- 4

# Following Kelly's percentages (Kelleyの基準を使うか否か)

KELLEY &lt;- TRUE

# Score output file name (0/1採点したデータを保存するファイル名)

scoFileName &lt;- "ScoreData.csv"

# Total Score Summary output file name (テスト得点の記述統計量を保存するファイル名)

totFileName &lt;- "TotalSummary.csv"

# Item Analysis output file name (全項目の項目分析の結果を保存するファイル名)

itaFileName &lt;- "ItemAnalysis.csv"

# Result of Each Item output file name (各項目の詳細な分析結果を保存するファイル名)

reiFileName &lt;- "ResultEachItem.csv"

# Trace line data output file name (トレースラインの座標データを保存するファイル名)

tldFileName &lt;- "TraceLineData.csv"

# Trace line output file name (トレースラインの図を保存するPDFファイル名)

tlgFileName &lt;- "TraceLine.pdf"

## 出力ファイル

以下の6個のファイルを出力

- 採点した得点データ
- テスト得点の記述統計量
- 全項目の項目分析の結果
- 各項目の項目分析の詳細な結果
- トレースラインの座標データ
- トレースラインの図

## 得点データ

NO.	School	Grade	Type	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	Total	Prop
1	1	1	1	1	1	1	1	1	1	1	1	1	1	10	1
2	1	2	2	2	1	1	1	0	1	1	1	1	0	8	0.8
3	1	2	3	1	1	1	1	1	1	1	1	1	0	9	0.9
4	1	3	2	1	1	0.5	0	1	1	1	1	1	0	7.5	0.75
5	2	1	1	1	1	1	1	1	1	1	1	1	0	9	0.9
6	2	2	3	0	0	0	0	0	0	0	0.5	0.5	1	2.5	0.25

各項目：各項目の得点

Total：合計得点

Prop：合計得点の得点率

## テスト合計得点および得点率の記述統計量

Score	Items	N	Mean	SD	Min	Med	Max	Alpha
Total	10	150	6.35	2.21	0	6.3	10	0.7
Prop	10	150	0.64	0.22	0	0.63	1	0.7

Items：項目数

N：受験者数

Mean：平均値

SD：標準偏差

Min：最小値

Med：中央値

Max：最大値

Alpha：アルファ係数

## 全項目の項目分析の結果

Item	N	P	D	Corr	Alpha	Dalpha	Categories	1	2	3	4	9	0	NA
X1	150	0.59	0.59	0.31	0.7	0.69	123490	0.54	0.11	0.2	0.08	0.03	0.04	0
X2	150	0.57	0.68	0.39	0.7	0.67	190	0.57				0.41	0.02	0
X3	150	0.71	0.46	0.38	0.7	0.68	1290	0.62	0.19			0.09	0.02	0
X4	150	0.62	0.62	0.39	0.7	0.67	12390	0.58	0.07	0.15		0.17	0.02	0
X5	150	0.73	0.59	0.41	0.7	0.67	123490	0.73	0	0	0	0.2	0.07	0
X6	150	0.7	0.67	0.45	0.7	0.66	123490	0.64	0.11	0	0	0.18	0.07	0
X7	150	0.7	0.54	0.42	0.7	0.67	123490	0.56	0.11	0.17	0	0.09	0.07	0
X8	150	0.67	0.55	0.43	0.7	0.67	123490	0.5	0.17	0.13	0.08	0	0.12	0
X9	150	0.69	0.4	0.4	0.7	0.68	123490	0.45	0.14	0.19	0.16	0.07	0	0
X10	150	0.37	0.27	0.09	0.7	0.73	123490	0.24	0.06	0.06	0.08	0.04	0.08	0.44

Item：項目名

N：受験者数

P：正答率

D：D値（上位群の正答率－下位群の正答率）

Corr：I-T相関係数（項目得点と、合計点から当該項目の得点を引いた得点との相関係数）

Alpha：アルファ係数（テスト全体のアルファ係数なので、全項目に同じ値が入っている）

Dalpha：当該項目を削除したときのアルファ係数

Categories：使用解答類型

各類型：当該類型の分類率

NA：無答率



## 各項目の項目分析の詳細な結果

Item	Categories	N	P	D	Corr	Alpha	DAlpha	
X3		1290	150	0.71	0.46	0.38	0.7	0.68
Prop		1	2	3	4	9	0	NA
All		0.62	0.19			0.09	0.02	0
U		0.95	0.02			0	0	0
M		0.57	0.24			0.06	0	0
L		0.37	0.27			0.22	0.07	0
D		0.59	-0.24			-0.22	-0.07	0
Corr		0.38	-0.15			-0.23	-0.4	NA

Item: 項目名

Category: 使用解答類型

N: 受験者数

P: 正答率

D: D値 (上位群の正答率 - 下位群の正答率)

Corr: I-T相関係数 (項目得点と、合計点から当該項目の得点を引いた得点との相関係数)

Alpha: アルファ係数 (テスト全体のアルファ係数なので、全項目に同じ値が入っている)

Dalpha: 当該項目を削除したときのアルファ係数

選択率

All: 受験者全体

U: 上位群

M: 中位群

L: 下位群

NA: 無答率

識別力

D: 当該選択枝のD値 (上位群の選択率 - 下位群の選択率)

Corr: 当該選択枝のI-T相関係数

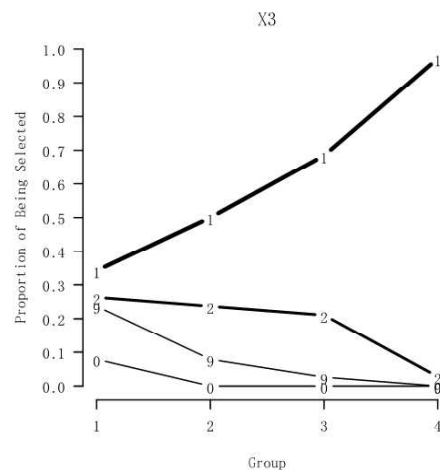
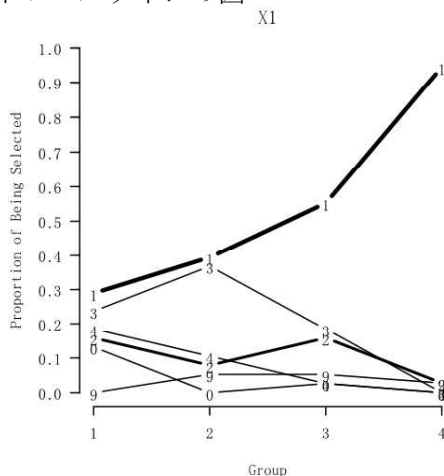
(当該選択枝の選択の有無と、合計点から当該項目の得点を引いた得点との相関係数)

## トレースラインの座標データ

Item	Category	1	2	3	4
X1	1	0.289474	0.394737	0.552632	0.944444
X1	2	0.157895	0.078947	0.157895	0.027778
X1	3	0.236842	0.368421	0.184211	0
X1	4	0.184211	0.105263	0.026316	0
X1	9	0	0.052632	0.052632	0.027778
X1	0	0.131579	0	0.026316	0
X1	NA	0	0	0	0

X3	1	0.342105	0.5	0.684211	0.972222
X3	2	0.263158	0.236842	0.210526	0.027778
X3	3	0.078947	0.184211	0.078947	0
X3	4	0	0	0	0
X3	9	0.236842	0.078947	0.026316	0
X3	0	0.078947	0	0	0
X3	NA	0	0	0	0

## トレースラインの図



太線: 得点のある解答類型

細線: 得点のない解答類型

**関数**

設定すべき値を指定して、以下のスクリプト全体を実行する。

以下のサイトからダウンロード可能

[https://www.educa.nagoya-u.ac.jp/~ishii-h/test\\_system.html](https://www.educa.nagoya-u.ac.jp/~ishii-h/test_system.html)

```
# Item Analysis

# Memory clear
rm(list=ls())

# Set work directory
setwd("e:¥¥Ruikei¥¥")

# Data File Name
ansFileName <- "Ruikei Data.csv"

# Key File name
keyFileName <- "Ruikei Key.csv"

# Column Number of First Test Item:
firstItemCol <- 5

# Option Categories
OPTIONS <- c("1","2","3","4","9","0")

# OPTIONS <- c("A","B","C","D","E","F")

# Number of groups in trace line
NGROUP <- 4

# Following Kelly's percentages
KELLEY <- TRUE

# Score output file name
scoFileName <- "ScoreData.csv"

# Total Score Summary output file name
totFileName <- "TotalSummary.csv"

# Item Analysis output file name
itaFileName <- "ItemAnalysis.csv"

# Result of Each Item output file name
reiFileName <- "ResultEachItem.csv"

# Trace line data output file name
tldFileName <- "TraceLineData.csv"

# Trace line graph output file name
tlgFileName <- "TraceLineGraph.pdf"

#-----
# Basically, no change is necessary in the following.
#-----

# Digits
rdgts <- 2
```

```

# -----

# Data import
dans <- read.table(ansFileName, header=T, sep=";", fileEncoding="shift-jis")
dkey <- read.table(keyFileName, header=T, sep=";", fileEncoding="shift-jis", na.strings=c("", NA))

# Number of Items
nItem <- ncol(dans) - firstItemCol + 1

# Number of OPTIONS
nOption <- length(OPTIONS)

# Sample size
nSample <- nrow(dans)

# Item names
itemNames <- colnames(dans)[firstItemCol : ncol(dans)]

# ID variable name
idName <- colnames(dans)[1]

# Key Category
dkeyc <- dkey[grepl("C", dkey$CS),]
colnames(dkeyc) <- c("Item", "CS", OPTIONS)

# Key Score
dkeys <- as.matrix(dkey[grepl("S", dkey$CS),])
colnames(dkeys) <- c("Item", "CS", OPTIONS)
dkeys1 <- dkeys[, c("Item", "CS")]
dkeys2 <- dkeys[, c(OPTIONS)]
for(j in OPTIONS){
  dkeys2[, j] <- ifelse(is.na(dkeys2[, j]), 0, dkeys2[, j])
}
storage.mode(dkeys2) <- "double"
dkeys <- data.frame(dkeys1, dkeys2)
colnames(dkeys) <- c("Item", "CS", OPTIONS)

# Score matrix
score <- as.data.frame(matrix(c(0), nrow=nSample, ncol=nItem))
colnames(score) <- c(itemNames)

# Scoring
for(i in itemNames){
  dansitmp <- matrix(c(0), nrow=nSample, ncol=nOption)
  colnames(dansitmp) <- OPTIONS
  for(j in OPTIONS){
    dansitmp[, j] <- ifelse(is.na(dans[, i]), 0,
                           ifelse(grepl(j, dans[, i]), dkeys[dkeys$Item==i, j], 0))
  }
  score[, i] <- rowSums(dansitmp)
}

# Total scores
score$Total <- rowSums(score[, itemNames])
score$Prop <- floor(100 * (rowMeans(score[, itemNames]) + 0.005)) / 100

# Score Data
dscore <- data.frame(dans[, c(1:(firstItemCol-1))], score)

# Output Score data
write.table(dscore, scoFileName, row.names=F, sep=";")

# Total Scores Summary
tot <- prop <- NULL
tot$Score <- "Total"

```

```

tot$Items <- length(itemNames)
tot$N <- nSample
tot$Mean <- floor(10^rdgts * (mean(score[, "Total"])+0.005))/10^rdgts
tot$SD <- floor(10^rdgts * (sd(score[, "Total"])+0.005))/10^rdgts
tot$Min <- min(score[, "Total"])
tot$Med <- median(score[, "Total"])
tot$Max <- max(score[, "Total"])
tot$Alpha <- floor(10^rdgts * (length(itemNames)/(length(itemNames)-1) *
                                (1-sum(apply(score[, itemNames], 2, var))/var(score[, "Total"])+0.005)) / 10
^rdgts
prop$Score <- "Prop"
prop$Items <- tot$Items
prop$N <- tot$N
prop$Mean <- floor(10^rdgts * (mean(score[, "Total"])/prop$Items+0.005))/10^rdgts
prop$SD <- floor(10^rdgts * (sd(score[, "Total"])/prop$Items+0.005))/10^rdgts
prop$Min <- min(score[, "Prop"])
prop$Med <- median(score[, "Prop"])
prop$Max <- max(score[, "Prop"])
prop$Alpha <- tot$Alpha

totout <- as.data.frame(tot)
propout <- as.data.frame(prop)
tpout <- rbind(totout, propout)

# Output Total score summary
write.table(tpout, totFileName, row.names=F, sep=",")

#remove tpout, totout, propout, prop
rm(tpout, totout, propout, prop)

# Item analysis
optionNames <- paste("o", OPTIONS, sep="")
itaNames <- c("Item", "N", "P", "D", "Corr", "Alpha", "Dalpna", "Categories", optionNames, "oNA")
dita <- matrix(c(0), nrow=length(itemNames), ncol=length(itaNames))
Corr <- rep(0, length(itemNames))
names(Corr) <- itemNames
rownames(dita) <- itemNames
colnames(dita) <- itaNames

dita[, "Item"] <- itemNames
dita[, "N"] <- tot$N
dita[, "P"] <- floor(10^rdgts * (colMeans(score[, itemNames])+0.005))/ 10^rdgts
dita[, "Alpha"] <- tot$Alpha

for(i in itemNames){
  Corr[i] <- cor(score[, i], (score[, "Total"]-score[, i]))
  dita[i, "Corr"] <- floor(10^rdgts * (Corr[i]+0.005))/ 10^rdgts
  scoreDi <- score[, itemNames[(itemNames %in% i)==F]]
  dita[i, "Dalpna"] <- floor(10^rdgts * (ncol(scoreDi)/(ncol(scoreDi)-1) *
                                (1-sum(apply(scoreDi, 2, var))/var(rowSums(scoreDi)))+0.005))/ 10^rdg
ts
}

dkeycs <- dkeyc[, OPTIONS]
for(j in OPTIONS){
  dkeycs[, j] <- ifelse(is.na(dkeycs[, j]), "", dkeycs[, j])
}
rownames(dkeycs) <- itemNames

vCategories <- apply(dkeycs[, OPTIONS], 1, paste, collapse="")
names(vCategories) <- itemNames
dita[, "Categories"] <- vCategories

# Data Sorting
vCorr <- Corr[Corr>0]
sNames <- names(vCorr)[order(vCorr, decreasing=T)]

```

```

sNames <- paste("dscore$", sNames, sep="", collapse=" ")
eval(parse(text=paste("dsort <- dscore[order(dscore$Total", sNames, ", dscore[, 1]),]", sep="") ))
dsort$sortID <- c(1:nrow(dsort))

# remove score, scoreDi, dscore, Corr
rm(score, scoreDi, dscore, Corr)

# Subgroup size
nSub <- ceiling(nSample * 0.27)

# Subgroup label
dsort$LMU <- ifelse(dsort[, "sortID"]<=nSub, "L",
  ifelse((dsort[, "sortID"]>nSub)&(dsort[, "sortID"]<=(nSample-nSub)), "M", "U"))

# D index
dsortL <- dsort[dsort$LMU=="L", ]
dsortU <- dsort[dsort$LMU=="U", ]
dita[, "D"] <- floor(10^rdgts * (colMeans(dsortU[, itemNames]) -
  colMeans(dsortL[, itemNames])+0.005))/ 10^rdgts

# remove dsortU, dsortL
rm(dsortU)
rm(dsortL)

# Each Item
drei <- NULL
v1 <- c("Item", "Categories", "N", "P", "D", "Corr", "Alpha", "DAlpha")

for(i in itemNames){
  # subjects*options matrix per item
  dansitmp <- matrix(c(0), nrow=nSample, ncol=(nOption+1))
  colnames(dansitmp) <- c(OPTIONS, "NA")
  for(j in OPTIONS){
    dansitmp[, j] <- ifelse(is.na(dans[, i]), 0,
      ifelse(grepl(j, dans[, i]), 1, 0))
  }
  dansitmp[, "NA"] <- ifelse((rowSums(dansitmp)==0), 1, 0)

  # choice ratio of each option per item
  dita[i, c(optionNames, "oNA")] <- floor(10^rdgts * (colMeans(dansitmp)+0.005))/ 10^rdgts

  for(j in OPTIONS){
    if (dkeycs[i, j] == "") {
      dita[i, paste("o", j, collapse="", sep="")] <- ""
    }
  }
}

# each item analysis
v2 <- c(i, vCategories[i], dita[i, "N"], dita[i, "P"], dita[i, "D"], dita[i, "Corr"], dita[i, "Alpha"],
  dita[i, "Dalpha"])
v3 <- c("Prop", OPTIONS, "NA")
v4 <- c("All", dita[i, c(optionNames, "oNA")])

# UML based
dsortitmp <- dsort[order(dsort[, idName]), c(idName, i, "Total", "LMU")]
dansitmp <- cbind(dsortitmp, dansitmp)
dansitmpL <- dansitmp[dansitmp[, "LMU"]=="L", ]
dansitmpM <- dansitmp[dansitmp[, "LMU"]=="M", ]
dansitmpU <- dansitmp[dansitmp[, "LMU"]=="U", ]
v5 <- c("U", (floor(10^rdgts * (colMeans(dansitmpU[, c(OPTIONS, "NA")])+0.005))/ 10^rdgts))
v6 <- c("M", (floor(10^rdgts * (colMeans(dansitmpM[, c(OPTIONS, "NA")])+0.005))/ 10^rdgts))
v7 <- c("L", (floor(10^rdgts * (colMeans(dansitmpL[, c(OPTIONS, "NA")])+0.005))/ 10^rdgts))
v8 <- c("D", (floor(10^rdgts * (colMeans(dansitmpU[, c(OPTIONS, "NA")]) - colMeans(dansitmpL[, c(OPTIONS, "NA")])+0.005))/ 10^rdgts))
v9 <- c("Corr", (floor(10^rdgts * (cor(dansitmp[, c(OPTIONS, "NA")], (dansitmp[, "Total"] - dansitmp[, i]),
```

```

v10 <- rep("", length(v5))

nemp <- length(v3) - length(v1)
if (nemp >= 0) {
  v1 <- c(v1, rep("", nemp))
  v2 <- c(v2, rep("", nemp))
} else {
  nemp <- abs(nemp)
  v3 <- c(v3, rep("", nemp))
  v4 <- c(v4, rep("", nemp))
  v5 <- c(v5, rep("", nemp))
  v6 <- c(v6, rep("", nemp))
  v7 <- c(v7, rep("", nemp))
  v8 <- c(v8, rep("", nemp))
  v9 <- c(v9, rep("", nemp))
  v10 <- c(v10, rep("", nemp))
}
temp1 <- rbind(v1, v2, v3, v4)
temp2 <- rbind(v5, v6, v7, v8, v9, v10)

for(j in OPTIONS){
  if (dkeycs[i,j] == "") {
    temp2[c("v5", "v6", "v7", "v8", "v9"), j] <- ""
  }
}
drei <- rbind(drei, temp1, temp2)
}

#remove dansitmpL, dansitmpM, dansitmpU
rm(dansitmpL, dansitmpM, dansitmpU)

#remove v1, v2, v3, v4, v5, v6, v7, v8, v9, v10
rm(v1, v2, v3, v4, v5, v6, v7, v8, v9, v10)

# Output Item Analysis Results
colnames(dita) <- c("Item", "N", "P", "D", "Corr", "Alpha", "Dalpha", "Categories", OPTIONS, "NA")

write.table(dita, itaFileName, row.names=F, sep=",")
write.table(drei, reiFileName, row.names=F, col.names=F, sep=",")

#remove dita, drei
rm(dita, drei)

# Trace lines

pdf(tlgFileName, paper="a4", width=7, height=14, family="Japan1", onefile=T)
layout(matrix(c(1, 2, 3, 4, 5, 6), 3, 2, byrow=T))
par(mar=c(6, 5, 2, 1))

tlData <- NULL

for(i in itemNames){
  # subjects*options matrix per item
  dansitmp <- matrix(c(0), nrow=nSample, ncol=(nOption+1))
  colnames(dansitmp) <- c(OPTIONS, "NA")
  for(j in OPTIONS){
    dansitmp[,j] <- ifelse(is.na(dans[,i]), 0,
                          ifelse(grepl(j, dans[,i]), 1, 0))
  }
  dansitmp[, "NA"] <- ifelse((rowSums(dansitmp)==0), 1, 0)

  dsortitmp <- dsort[order(dsort[,idName]), c(idName, "LMU", "sortID")]
  dansitmp <- cbind(dsortitmp, dansitmp)
  dansitmp <- dansitmp[order(dansitmp[, "sortID"]), ]
}

```

```

dansitmp$group <- NGROUP
if ((NGROUP==3) && (KELLEY==TRUE)) {
  dansitmp$group[1:nSub] <- 1
  dansitmp$group[(nSub+1):(nSample-nSub)] <- 2
  dansitmp$group[(nSample-nSub+1):nSample] <- 3
} else {
  nEach <- floor(nSample/NGROUP +0.5)
  for(igp in 1:(NGROUP-1)){
    hb <- (igp-1)*nEach+1
    he <- igp*nEach
    dansitmp$group[hb:he] <- igp
  }
}

ioprop <- matrix(c(0), nrow=(nOption+1), ncol=NGROUP)
rownames(ioprop) <- c(OPTIONS, "NA")
colnames(ioprop) <- c(1:NGROUP)

lwdv <- rep(1, (nOption+1))
lwdv <- 2*c(as.matrix(dkeys[dkeys$Item==i, OPTIONS]), 0) + lwdv
names(lwdv) <- c(OPTIONS, "NA")

for(j in 1:NGROUP){
  dansitmpj <- dansitmp[dansitmp$group==j,]
  ioprop[, j] <- colMeans(dansitmpj[, c(OPTIONS, "NA")])
}

plot("¥n", xlim=c(1, NGROUP), ylim=c(0,1), las=1, axes=F,
      xlab="Group", ylab="Proportion of Being Selected",
      pch="-", type="b", lty=1, lwd=lwdv["NA"], main=i)
axis(side=1, c(1:NGROUP))
axis(side=2, seq(0,1, 0.1), las=1)

for (j in dkeycs[i,]){
  if (j != "") {
    j <- as.character(j)
    par(new=T)
    plot(c(1:NGROUP), ioprop[j,], xlim=c(1, NGROUP), ylim=c(0,1), las=1, axes=F,
         pch=j, type="b", lty=1, lwd=lwdv[j], xlab="", ylab="", xaxt="n", yaxt="n")
  }
}

iopropi <- data.frame(i, rownames(ioprop), ioprop)
tlData <- rbind(tlData, iopropi)
}

dev.off()

# Output Trace Line Data
colnames(tlData) <- c("Item", "Category", c(1:NGROUP))

write.table(tlData, tldFileName, row.names=F, sep=",")

```

## 項目分析 — itemanalysisパッケージ

多枝選択項目の項目分析を行う。

```
library(itemanalysis)
オブジェクト名 <- itemanalysis1(data=回答データ,
                                key=正答キー,
                                options=選択式号ベクトル,
                                ngroup=群数,
                                correction=TRUE)
```

オブジェクト名\$plots[項目番号]

あらかじめitemanalysisパッケージをインストールしておく必要がある。  
パッケージがうまく展開されないときは、最新版のRをダウンロードして、パッケージのインストールからやり直す。  
トレースライン作成にあたっては、得点可能範囲を均等幅に分割して群を構成している。

```
> rm(list=ls())
> setwd("d:¥¥")

> # Answer file name
> ansFileName <- "IWS3_ANS.csv"
>
> # Key file name
> keyFileName <- "IWS3_KEY.csv"
```

	A	B	C	D	E	F	G	H	I	J	K
1	subject	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
2	IWS3	5	1	2	2	2	4	2	2	2	4
3											

```
> dans0 <- read.table(ansFileName, header=T, sep="," , fileEncoding="shift-jis")
> dkey0 <- read.table(keyFileName, header=T, sep="," , fileEncoding="shift-jis")
```

## &gt; 不要な列の削除

```
> dans <- dans0[, -1]
> dkey <- t(dkey0[, -1])
```

# 正答キーを縦ベクトルに変換しておく

```
> head(dkey)
      [, 1]
Item1    5
Item2    1
Item3    2
Item4    2
Item5    2
Item6    4>
```

	A	B	C	D	E	F	G	H	I	J	K
2	StudentID	Item1	Item2	Item3	Item4	Item5	Item6	Item7	Item8	Item9	Item10
3		5	1	2	2	2	4	2	2	2	4
4		5	5	2	2	3	2	4	2	3	2
5		5	1	2	2	4	5	2	2	3	2
6		5	1	2	2	4	1	2	4	4	2
7		5	1	2	2	2	4	2	4	3	2
8		5	5	2	2	2	2	4	2	3	2
9		5	1	2	2	3	2	3	2	2	4
10		5	1	2	2	2	1	4	2	2	2
11		5	5	2	1	1	2	3	4	2	1
12		3	1	2	2	4	3	2	4	4	4
13		5	1	3	2	4	4	3	4	3	4
14		5	5	2	2	2	5	5	4	2	2
15		5	5	2	1	4	3	2	4	2	2
16		1	1	2	2	4	2	5	2	3	2
17		5	1	2	2	4	4	2	4	3	5
18		5	1	2	2	1	2	2	4	4	5
19		5	1	2	2	2	4	2	4	2	2
20		5	1	2	2	2	4	2	4	2	2
21		5	1	4	2	4	4	3	4	2	4

```
> # 項目分析
> library(itemanalysis)
>
> item.analysis <- itemanalysis1(data=dans,
+                               key=dkey,
+                               options=c("1", "2", "3", "4", "5"),
+                               ngroup=4,
+                               correction=TRUE)
```

## &gt; # 実行結果

```
*****
itemanalysis: An R package for Classical Test Theory Item Analysis
```

Cengiz Zopluoglu

University of Miami  
Department of Educational and Psychological Studies  
Research, Measurement, and Evaluation Program

c.zopluoglu@miami.edu

Please report any programming bug or problem you experience to improve the code.

\*\*\*\*\*



Processing Date: Tue Jan 29 16:28:39 2019

## ITEM STATISTICS

	Item Difficulty	Item Threshold	Point-Biserial	Biserial
Item 1	0.784	-0.786	0.225	0.317
Item 2	0.727	-0.604	0.243	0.325
Item 3	0.937	-1.530	0.116	0.228
Item 4	0.842	-1.003	0.173	0.262
Item 5	0.480	0.050	0.417	0.523
Item 6	0.382	0.300	0.248	0.316
Item 7	0.747	-0.665	0.073	0.099
Item 8	0.307	0.504	0.053	0.070
Item 9	0.289	0.556	0.020	0.027
Item 10	0.279	0.586	0.151	0.202
Item 11	0.273	0.604	0.054	0.072
Item 12	0.639	-0.356	0.223	0.285
Item 13	0.598	-0.248	0.044	0.056
Item 14	0.465	0.088	0.391	0.491
Item 15	0.383	0.298	0.289	0.368
Item 16	0.443	0.143	0.174	0.219
Item 17	0.412	0.222	0.415	0.524
Item 18	0.631	-0.335	0.353	0.452
Item 19	0.585	-0.215	0.283	0.358
Item 20	0.208	0.813	0.093	0.131
Item 21	0.558	-0.146	0.254	0.319
Item 22	0.395	0.266	0.185	0.235
Item 23	0.371	0.329	0.344	0.440
Item 24	0.548	-0.121	0.403	0.507
Item 25	0.522	-0.055	0.185	0.232
Item 26	0.532	-0.080	0.261	0.328
Item 27	0.327	0.448	0.172	0.224
Item 28	0.580	-0.202	0.058	0.073
Item 29	0.298	0.530	0.344	0.454
Item 30	0.468	0.080	0.265	0.333

## DISTRACTOR SELECTION PROPORTIONS

	1	2	3	4	5
Item 1	0.070	0.074	0.046	0.026	0.784
Item 2	0.727	0.028	0.015	0.029	0.201
Item 3	0.011	0.937	0.022	0.015	0.015
Item 4	0.100	0.842	0.025	0.018	0.015
Item 5	0.097	0.480	0.085	0.316	0.022
Item 6	0.043	0.351	0.104	0.382	0.120
Item 7	0.042	0.747	0.062	0.056	0.093
Item 8	0.026	0.307	0.049	0.604	0.014
Item 9	0.031	0.289	0.495	0.168	0.017
Item 10	0.016	0.452	0.120	0.279	0.133
Item 11	0.273	0.242	0.199	0.197	0.089
Item 12	0.058	0.145	0.639	0.103	0.055
Item 13	0.046	0.191	0.598	0.117	0.048
Item 14	0.201	0.114	0.135	0.085	0.465
Item 15	0.032	0.095	0.383	0.343	0.147
Item 16	0.025	0.089	0.334	0.443	0.109
Item 17	0.412	0.121	0.234	0.090	0.143
Item 18	0.094	0.154	0.631	0.060	0.061
Item 19	0.116	0.175	0.585	0.084	0.040
Item 20	0.044	0.130	0.490	0.128	0.208
Item 21	0.040	0.121	0.558	0.192	0.089
Item 22	0.073	0.395	0.177	0.077	0.278
Item 23	0.074	0.194	0.209	0.152	0.371
Item 24	0.049	0.548	0.122	0.125	0.156
Item 25	0.060	0.184	0.522	0.137	0.097
Item 26	0.042	0.136	0.201	0.089	0.532
Item 27	0.066	0.160	0.397	0.327	0.050
Item 28	0.061	0.580	0.102	0.052	0.205
Item 29	0.094	0.279	0.191	0.138	0.298
Item 30	0.099	0.158	0.213	0.468	0.062

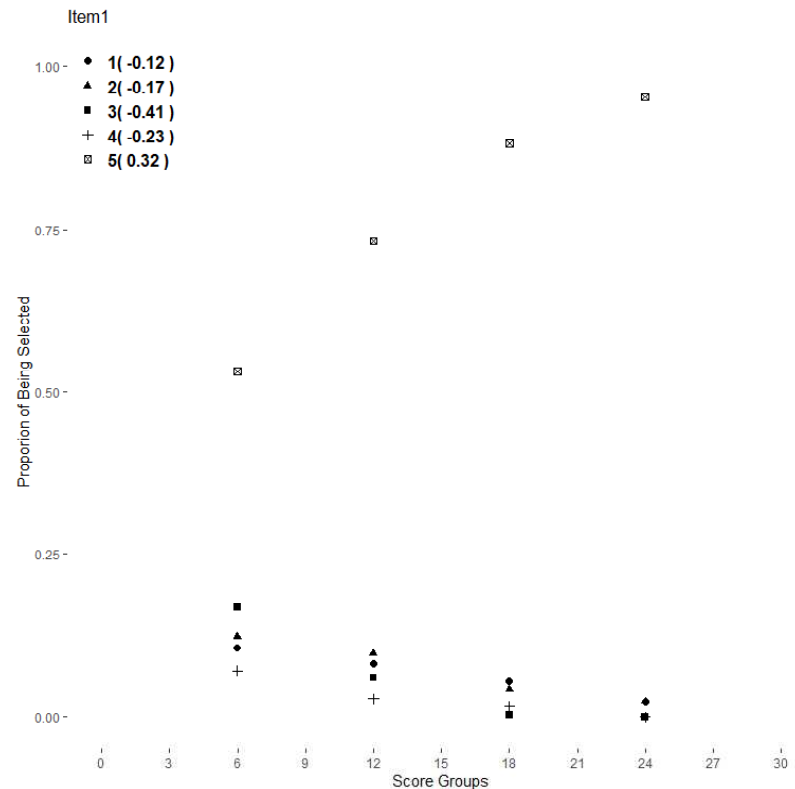
## DISTRATOR Point-Biserial

	1	2	3	4	5
Item 1	-0.062	-0.089	-0.190	-0.087	0.225
Item 2	0.243	-0.126	-0.096	-0.085	-0.153
Item 3	-0.078	0.116	-0.078	-0.058	-0.013
Item 4	-0.127	0.173	-0.087	-0.069	-0.020
Item 5	-0.053	0.417	-0.148	-0.286	-0.124
Item 6	0.031	-0.048	-0.139	0.248	-0.189
Item 7	-0.078	0.073	-0.111	0.030	0.014
Item 8	-0.039	0.053	-0.085	-0.003	0.012
Item 9	-0.084	0.020	0.164	-0.180	-0.071
Item 10	-0.053	0.069	-0.225	0.151	-0.066
Item 11	0.054	-0.034	-0.003	-0.005	-0.021
Item 12	-0.048	-0.174	0.223	-0.052	-0.081
Item 13	0.030	-0.050	0.044	-0.007	-0.030
Item 14	-0.074	-0.224	-0.154	-0.150	0.391
Item 15	-0.144	-0.225	0.289	-0.160	0.076
Item 16	-0.101	-0.151	-0.060	0.174	0.003
Item 17	0.415	-0.208	-0.241	-0.185	0.053
Item 18	-0.117	-0.232	0.353	-0.146	-0.073
Item 19	-0.110	-0.166	0.283	-0.103	-0.066
Item 20	-0.091	-0.171	0.161	-0.126	0.093
Item 21	-0.145	-0.227	0.254	-0.090	0.042
Item 22	-0.083	0.185	-0.174	-0.134	0.074
Item 23	-0.054	-0.200	-0.112	-0.075	0.344
Item 24	-0.093	0.403	-0.200	-0.168	-0.165
Item 25	-0.104	-0.138	0.185	-0.063	0.024
Item 26	-0.069	-0.132	-0.106	-0.101	0.261
Item 27	0.031	-0.118	-0.083	0.172	-0.020
Item 28	0.052	0.058	-0.006	0.007	-0.101
Item 29	-0.067	-0.186	-0.126	-0.014	0.344
Item 30	-0.046	-0.166	-0.123	0.265	-0.033

## DISTRATOR Biserial

	1	2	3	4	5
Item 1	-0.118	-0.166	-0.412	-0.229	0.317
Item 2	0.325	-0.324	-0.308	-0.214	-0.219
Item 3	-0.280	0.228	-0.219	-0.186	-0.043
Item 4	-0.216	0.262	-0.232	-0.207	-0.064
Item 5	-0.092	0.523	-0.266	-0.374	-0.347
Item 6	0.069	-0.062	-0.235	0.316	-0.306
Item 7	-0.175	0.099	-0.219	0.061	0.024
Item 8	-0.103	0.070	-0.180	-0.004	0.039
Item 9	-0.209	0.027	0.205	-0.268	-0.216
Item 10	-0.165	0.087	-0.366	0.202	-0.104
Item 11	0.072	-0.047	-0.005	-0.008	-0.038
Item 12	-0.097	-0.269	0.285	-0.088	-0.166
Item 13	0.066	-0.072	0.056	-0.012	-0.063
Item 14	-0.105	-0.370	-0.243	-0.268	0.491
Item 15	-0.352	-0.391	0.368	-0.207	0.117
Item 16	-0.269	-0.267	-0.078	0.219	0.005
Item 17	0.524	-0.336	-0.333	-0.327	0.082
Item 18	-0.204	-0.353	0.452	-0.292	-0.145
Item 19	-0.180	-0.245	0.358	-0.184	-0.149
Item 20	-0.202	-0.272	0.202	-0.201	0.131
Item 21	-0.330	-0.368	0.319	-0.130	0.075
Item 22	-0.157	0.235	-0.255	-0.247	0.099
Item 23	-0.101	-0.288	-0.159	-0.115	0.440
Item 24	-0.198	0.507	-0.324	-0.269	-0.250
Item 25	-0.206	-0.201	0.232	-0.099	0.041
Item 26	-0.154	-0.207	-0.151	-0.179	0.328
Item 27	0.059	-0.178	-0.105	0.224	-0.043
Item 28	0.103	0.073	-0.010	0.014	-0.144
Item 29	-0.116	-0.248	-0.183	-0.023	0.454
Item 30	-0.079	-0.251	-0.173	0.333	-0.064

```
> # トレースラインの表示
> item.analysis$plots[[1]]
[[1]]
```



```
> # 群平均の値の取り出し
> item.analysis$plots[[1]][1]
$data
      sg      p      opt
(2, 98, 9]  5.99 0.106194690 1( -0.12 )
(9, 15]    12.00 0.081585082 1( -0.12 )
(15, 21]   18.00 0.056300268 1( -0.12 )
(21, 27]   24.00 0.023529412 1( -0.12 )
(2, 98, 9]1  5.99 0.123893805 2( -0.17 )
(9, 15]1    12.00 0.097902098 2( -0.17 )
(15, 21]1   18.00 0.042895442 2( -0.17 )
(21, 27]1   24.00 0.023529412 2( -0.17 )
(2, 98, 9]2  5.99 0.168141593 3( -0.41 )
(9, 15]2    12.00 0.060606061 3( -0.41 )
(15, 21]2   18.00 0.002680965 3( -0.41 )
(21, 27]2   24.00 0.000000000 3( -0.41 )
(2, 98, 9]3  5.99 0.070796460 4( -0.23 )
(9, 15]3    12.00 0.027972028 4( -0.23 )
(15, 21]3   18.00 0.016085791 4( -0.23 )
(21, 27]3   24.00 0.000000000 4( -0.23 )
(2, 98, 9]4  5.99 0.530973451 5( 0.32 )
(9, 15]4    12.00 0.731934732 5( 0.32 )
(15, 21]4   18.00 0.882037534 5( 0.32 )
(21, 27]4   24.00 0.952941176 5( 0.32 )
```

```
> # すべての項目のトレースラインをPDFファイルに出力
> pdf("TraceLineFigures.pdf", paper="a4", width=7, height=7, family="Japan1")
> item.analysis$plots
> layout(1)
> dev.off()
windows
2
```

## 23 項目応答理論

## 項目パラメタの推定 — 1PL, 2PL, 3PLモデル

## 四分相関係数の計算

```
library(polycor)
library(psych)
オブジェクト名 <- polychoric(データフレーム名)
相関行列名 <- オブジェクト名$rho
```

あらかじめpolycorパッケージとpsychパッケージをインストールしておく必要がある。  
polycorの出力には四分相関係数以外のものも含まれるので、\$rhoで四分相関係数だけを切り取る。

## 一次元性の確認

```
VSS.scree(相関行列名)
```

## fa.parallel関数を使う方法

```
library(psych)
fa.parallel(データ名, fa = "pc")
```

あらかじめpsychパッケージをインストールしておく必要がある。  
スクリーンプロットを表示する。

## 項目パラメタの推定

```
library(ltm)
library(irtoys)
オブジェクト名 <- est(resp=データフレーム名, model="モデル名", engine="ltm")
```

あらかじめltmパッケージとirtoysパッケージをインストールしておく必要がある。  
model には "1PL", "2PL", "3PL" を指定できる。  
engine には "ic1", "bilog", "ltm" を指定できる。"ltm" にしておくとRだけで完結するが、"bilog"など  
とするとBILOGソフトが別途必要になる。

【重要】項目識別力パラメタajは、 $D=1.7$ を掛けた値として推定される。 $aj = D \cdot aj^*$  の $aj^*$ を求めるときは、 $aj/D$  を計算する必要がある。

```
> setwd("d:¥¥")
> d1 <- read.table("irt_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> head(d1)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1      1  1  1  1  1  0  1  1  1  1  1
2      1  1  1  0  1  0  0  1  0  0  0
3      1  1  1  1  1  1  1  1  1  1  1
4      1  1  1  0  1  0  0  0  0  0  0
5      1  1  1  0  1  0  0  1  1  1  0
6      1  1  0  1  0  0  0  0  0  1  0
>
```

```
> # 人数
> nrow(d1)
[1] 778
>
```

```
> # 項目正答率
> (pv <- colMeans(d1))
```

```
      s03      s04      s05      s06      s07      s08      s09
0.71208226 0.82519280 0.73778920 0.73007712 0.88688946 0.06298201 0.56683805
      s12      s14      s15      s16
0.53727506 0.55398458 0.58097686 0.44344473 >
```

	A	B	C	D	E	F	G	H	I	J	K
1	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	1	1	1	1	1	0	1	1	1	1	1
3	1	1	1	0	1	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0	0	0	0
6	1	1	1	0	1	0	0	1	1	1	0
7	1	1	0	1	0	0	0	0	0	1	0
8	1	0	1	1	1	0	1	0	1	0	0
9	0	1	1	1	1	0	0	1	1	0	1
10	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1	1	1	1
12	1	1	1	1	1	0	0	1	0	0	0
13	1	1	1	1	1	0	1	1	1	0	0
14	1	1	1	1	1	0	1	1	1	1	1
15	0	1	1	1	0	0	0	1	0	1	1
16	1	0	1	1	1	0	1	1	0	0	0
17	1	1	1	0	0	0	0	1	1	0	1
18	1	1	1	1	1	0	0	0	1	1	1
19	1	1	1	1	1	0	0	1	0	1	1
20	1	1	1	0	1	0	0	0	0	1	0
21	0	1	1	1	1	1	1	0	1	1	1

```
> # 四分相関係数の計算
> library(polycor)
> library(psych)
> t.cor <- polychoric(d1)
> t.cor <- t.cor$rho
> round(t.cor, 2)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
s03 1.00 0.06 0.33 0.34 0.21 0.10 0.25 0.30 0.12 0.15 0.27
s04 0.06 1.00 0.31 0.33 0.24 0.16 0.10 0.15 0.17 0.16 0.21
s05 0.33 0.31 1.00 0.52 0.29 0.35 0.32 0.40 0.26 0.24 0.29
s06 0.34 0.33 0.52 1.00 0.48 0.14 0.38 0.38 0.30 0.25 0.40
s07 0.21 0.24 0.29 0.48 1.00 0.36 0.48 0.42 0.48 0.37 0.41
s08 0.10 0.16 0.35 0.14 0.36 1.00 0.38 0.29 0.13 0.27 0.24
s09 0.25 0.10 0.32 0.38 0.48 0.38 1.00 0.27 0.27 0.33 0.35
s12 0.30 0.15 0.40 0.38 0.42 0.29 0.27 1.00 0.40 0.28 0.53
s14 0.12 0.17 0.26 0.30 0.48 0.13 0.27 0.40 1.00 0.31 0.35
s15 0.15 0.16 0.24 0.25 0.37 0.27 0.33 0.28 0.31 1.00 0.35
s16 0.27 0.21 0.29 0.40 0.41 0.24 0.35 0.53 0.35 0.35 1.00
```

```
> # 一次元性の確認
```

```
> # VSS.screen を使う方法
```

```
> VSS.screen(t.cor)
```

```
>
```

```
>
```

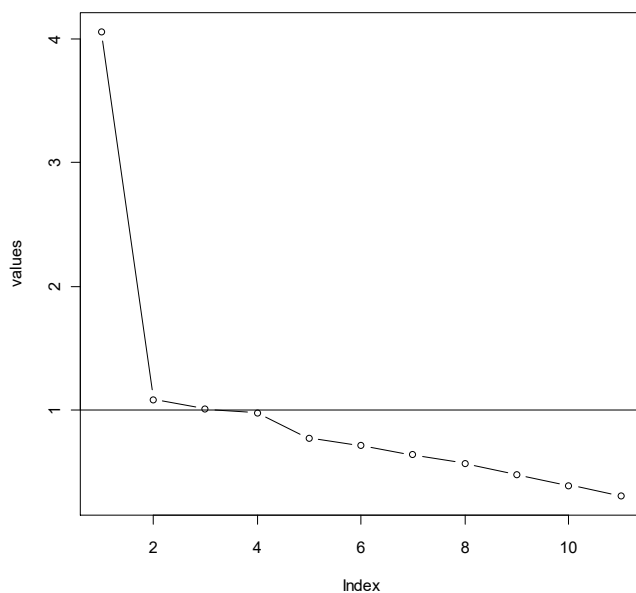
```
> # fa.parallel関数を使う方法
```

```
> library(psych)
```

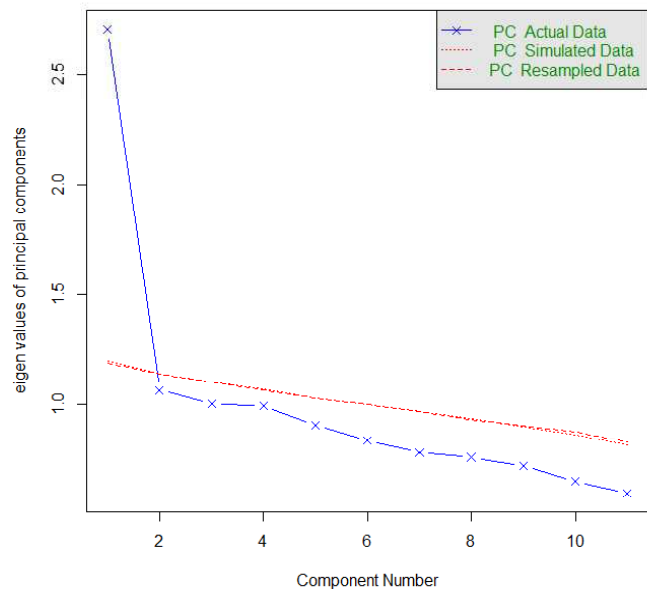
```
> fa.parallel(d1, fa = "pc")
```

Parallel analysis suggests that the number of factors = NA and the number of components = 1

screen plot



Parallel Analysis Scree Plots



```
> # 項目母数の推定
```

```
> library(ltm)
```

```
> library(irtoys)
```

```
> p.all <- est(resp=d1, model="2PL", engine="ltm")
```

```
> colnames(p.all$est) <- c("aj", "bj", "cj")
```

```
> colnames(p.all$sse) <- c("aj", "bj", "cj")
```

```
> rownames(p.all$sse) <- rownames(p.all$est)
```

```
> p.all
```

```
$est
      aj      bj c.j
s03 0.7519081 -1.3479054 0
s04 0.6336668 -2.6447011 0
s05 1.2141256 -1.0870115 0
s06 1.4940753 -0.9263137 0
s07 1.7233574 -1.7290162 0
s08 0.9815908 3.1497698 0
s09 1.0866067 -0.3036178 0
s12 1.4498429 -0.1380151 0
s14 1.0159834 -0.2558882 0
s15 0.9046749 -0.4216555 0
s16 1.4752955 0.2216158 0
```

# 項目母数の推定値

```
$se
      aj      bj c.j
s03 0.1155375 0.20441485 0
s04 0.1250574 0.47668423 0
s05 0.1526624 0.12175749 0
s06 0.1800035 0.09653309 0
s07 0.2457579 0.15253089 0
s08 0.2276118 0.58227949 0
s09 0.1333925 0.08662509 0
s12 0.1680297 0.06968662 0
s14 0.1275844 0.08936450 0
s15 0.1194251 0.10329955 0
s16 0.1764927 0.06974481 0
```

# 標準誤差

```
$vcm
$vcm[[1]]
      [,1]      [,2]
[1,] 0.01334892 0.01938195
[2,] 0.01938195 0.04178543
```

# 分散共分散行列

```
$vcm[[2]]
      [,1]      [,2]
[1,] 0.01563935 0.05625316
[2,] 0.05625316 0.22722786
```

```
$vcm[[3]]
      [,1]      [,2]
[1,] 0.0233058 0.01345200
[2,] 0.0134520 0.01482489
```

```
$vcm[[4]]
      [,1]      [,2]
[1,] 0.03240126 0.011221411
[2,] 0.01122141 0.009318638
```

```
$vcm[[5]]
      [,1]      [,2]
[1,] 0.06039696 0.03077978
[2,] 0.03077978 0.02326567
```

```
$vcm[[6]]
      [,1]      [,2]
[1,] 0.05180715 -0.1273928
[2,] -0.12739281 0.3390494
```

```
$vcm[[7]]
      [,1]      [,2]
[1,] 0.017793546 0.003502177
[2,] 0.003502177 0.007503907
```

```
$vcm[[8]]
      [,1]      [,2]
[1,] 0.028233972 0.001676149
[2,] 0.001676149 0.004856224
```

```
$vcm[[9]]
      [,1]      [,2]
```

```
[1,] 0.016277785 0.003006243
[2,] 0.003006243 0.007986014
```

```
$vcm[[10]]
      [,1]      [,2]
[1,] 0.014262349 0.005055913
[2,] 0.005055913 0.010670797
```

```
$vcm[[11]]
      [,1]      [,2]
[1,] 0.031149668 -0.002364131
[2,] -0.002364131 0.004864339
```

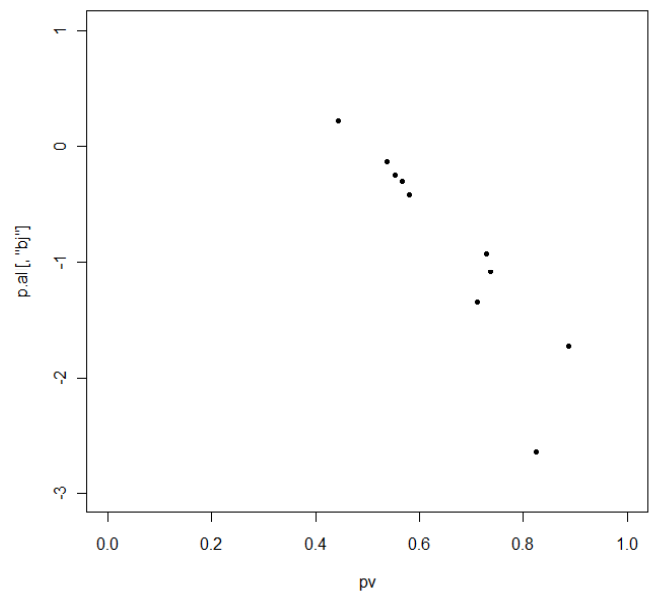
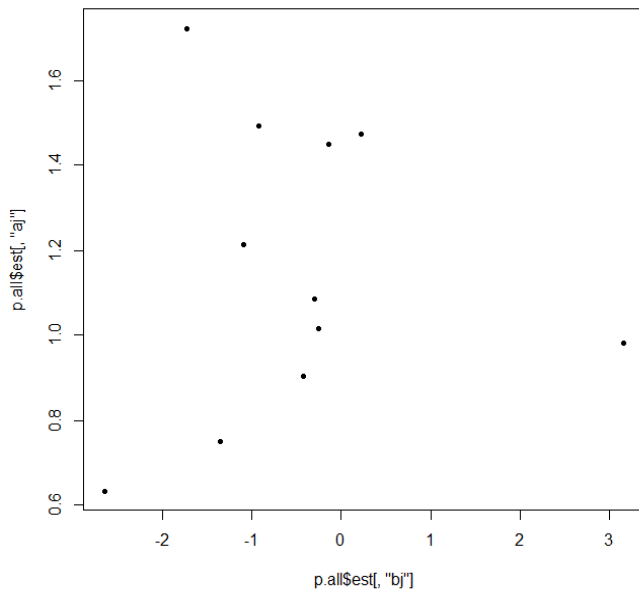
```
# 項目困難度と項目識別力の散布図
```

```
plot(p.all$est[, "bj"], p.all$est[, "aj"], pch=20)
```

```
>
```

```
> # 項目正答率と困難度の散布図
```

```
> plot(pv, p.all$est[, "bj"], xlim=c(0, 1), ylim=c(-3, 1), pch=20)
```





## 特性関数・情報関数の図示

```
library(irtoys)
```

あらかじめltmパッケージとirtoysパッケージをインストールしておく必要がある。

## 項目特性曲線

```
plot(irf(項目パラメタデータ行列))
```

## 項目情報曲線

```
plot(iif(項目パラメタデータ行列))
```

## テスト特性曲線

```
plot(trf(項目パラメタデータ行列))
```

## テスト情報曲線

```
plot(tif(項目パラメタデータ行列))
```

項目パラメタデータ行列を1項目の項目パラメタだけにすれば、その項目だけの特性曲線を描く。  
項目パラメタデータ行列に複数の項目パラメタを入れれば、それらすべての特性曲線を描く。

【重要】項目識別力パラメタ $a_j$ は、 $D=1.7$ を掛けた値である。 $a_j = D \cdot a_j^*$  の $a_j^*$ を求めるときは、 $a_j/D$  を計算する必要がある。

```
> setwd("d:¥¥¥")
> d1 <- read.table("irt_データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
>
>
> # 項目母数の推定
> library(ltm)
> library(irtoys)
> p.all <- est(resp=d1, model="2PL", engine="ltm")
> colnames(p.all$est) <- c("aj", "bj", "cj")
> colnames(p.all$se) <- c("aj", "bj", "cj")
> rownames(p.all$se) <- rownames(p.all$est)
> p.all
```

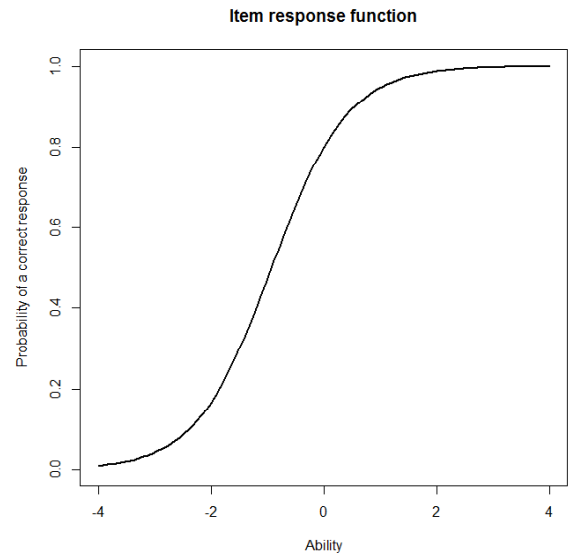
```
$est                                     # 項目母数の推定値
      aj      bj      cj
s03 0.7519081 -1.3479054  0
s04 0.6336668 -2.6447011  0
s05 1.2141256 -1.0870115  0
s06 1.4940753 -0.9263137  0
s07 1.7233574 -1.7290162  0
s08 0.9815908  3.1497698  0
s09 1.0866067 -0.3036178  0
s12 1.4498429 -0.1380151  0
s14 1.0159834 -0.2558882  0
s15 0.9046749 -0.4216555  0
s16 1.4752955  0.2216158  0
```

```
$se                                     # 標準誤差
      aj      bj      cj
s03 0.1155375 0.20441485  0
s04 0.1250574 0.47668423  0
s05 0.1526624 0.12175749  0
s06 0.1800035 0.09653309  0
s07 0.2457579 0.15253089  0
s08 0.2276118 0.58227949  0
s09 0.1333925 0.08662509  0
s12 0.1680297 0.06968662  0
s14 0.1275844 0.08936450  0
s15 0.1194251 0.10329955  0
s16 0.1764927 0.06974481  0
```

	A	B	C	D	E	F	G	H	I	J	K
1	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	1	1	1	1	1	0	1	1	1	1	1
3	1	1	1	0	1	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0	0	0	0
6	1	1	1	0	1	0	0	1	1	1	0
7	1	1	0	1	0	0	0	0	0	1	0
8	1	0	1	1	1	0	1	0	1	0	0
9	0	1	1	1	1	0	0	1	1	0	1
10	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1	1	1	1
12	1	1	1	1	1	0	0	1	0	0	0
13	1	1	1	1	1	0	1	1	1	0	0
14	1	1	1	1	1	0	1	1	1	1	1
15	0	1	1	1	0	0	0	1	0	1	1
16	1	0	1	1	1	0	1	1	0	0	0
17	1	1	1	0	0	0	0	1	1	0	1
18	1	1	1	1	1	0	0	0	1	1	1
19	1	1	1	1	1	0	0	1	0	1	1
20	1	1	1	0	1	0	0	0	0	1	0
21	0	1	1	1	1	1	1	0	1	1	1

```
$vcm
# 省略
```

```
>
> # 特性関数, 情報関数の図示
>
> # 項目特性曲線
> plot(irf(p.all$est["s06",]))
>
>
```

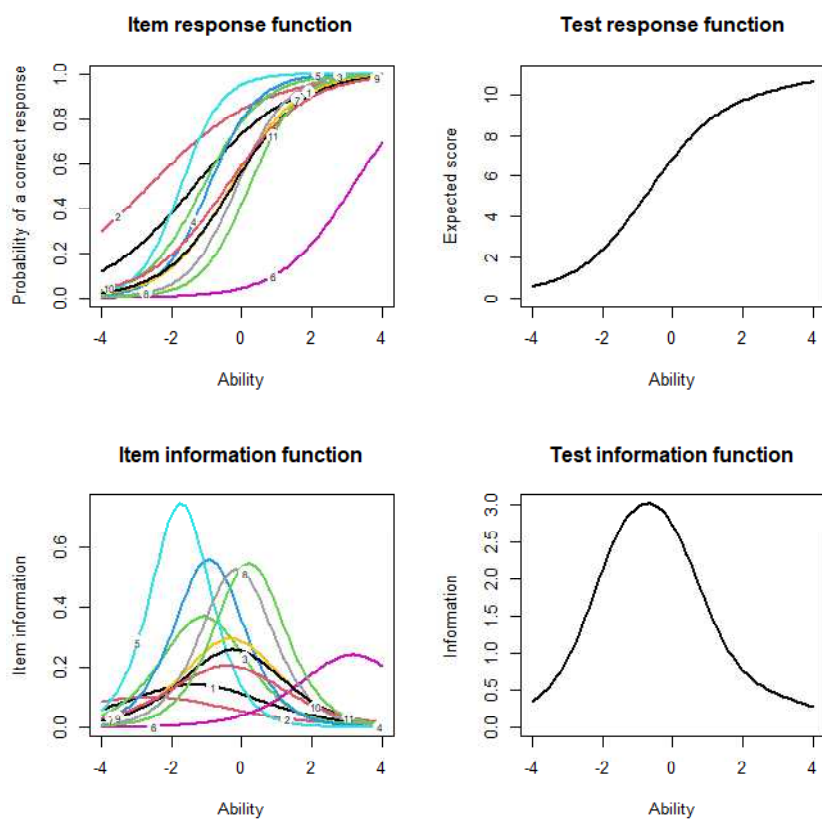


```
> # 図出力画面を2*2に分割
> par(mfrow=c(2, 2))
>
> # 項目特性曲線
> plot(irf(p.all$est), co=NA, label=TRUE)
>
> # テスト特性曲線
> plot(trf(p.all$est))
>
> # 項目情報曲線
> plot(iif(p.all$est), co=NA, label=TRUE)
>
> # テスト情報曲線
> plot(tif(p.all$est))
```

```

>
>
> # 図出力画面を1*1に戻す
> par(mfrow=c(1, 1))
>

```



## 特性値の推定 — 項目パラメタを推定する場合

```
library(irtoys)
```

あらかじめirtoysパッケージをインストールしておく必要がある。

## 最尤推定値・ベイズモーダル推定値

```
mlebm(resp=データ行列, ip=項目パラメタ行列, method="推定方法")
```

method を"ML"または省略すると最尤推定値, "BM"とするとベイズモーダル推定値を計算する.  
"BM"とした場合は mu= と sigma= で事前分布 (正規分布) を指定する. デフォルトは標準正規分布.

## 期待事後推定値

```
eap(resp=データ行列, ip=項目パラメタ行列, qu=normal.qu())
```

qu は求積点と重みの指定をする. 標準正規分布の場合 qu=normal.qu() とする.

【重要】項目識別力パラメタ $a_j$ は,  $D=1.7$ を掛けた値として推定される。 $a_j = D \cdot a_j^*$  の $a_j^*$ を求めるときは,  $a_j/D$  を計算する必要がある。

```
> rm(list=ls())
> setwd("d:¥¥")

> d1 <- read.table("itemScores.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  studentID s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1      A001   1   1   1   1   1   0   1   1   1   1   1
2      A002   1   1   1   0   1   0   0   1   0   0   0
3      A003   1   1   1   1   1   1   1   1   1   1   1
4      A004   1   1   1   0   1   0   0   0   0   0   0
5      A005   1   1   1   0   1   0   0   1   1   1   0
6      A006   1   1   0   1   0   0   0   0   0   1   0
>
>
> # 0/1採点データを取り出して, matrix型で保存
> itemScores <- as.matrix(d1[,c("s03","s04","s05","s06","s07",
+                               "s08","s09","s12","s14","s15","s16")])
> rownames(itemScores) <- d1[,c("studentID")]
> head(itemScores)
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
A001   1   1   1   1   1   0   1   1   1   1   1
A002   1   1   1   0   1   0   0   1   0   0   0
A003   1   1   1   1   1   1   1   1   1   1   1
A004   1   1   1   0   1   0   0   0   0   0   0
A005   1   1   1   0   1   0   0   1   1   1   0
A006   1   1   0   1   0   0   0   0   0   1   0
>
>
> # 項目母数の推定
> library(ltm)
> library(irtoys)
> p.all <- est(resp=itemScores, model="2PL", engine="ltm")
> colnames(p.all$est) <- c("aj", "bj", "cj")
> colnames(p.all$se) <- c("aj", "bj", "cj")
> rownames(p.all$se) <- rownames(p.all$est)
> p.all
$est
      aj      bj  cj
s03 0.7519081 -1.3479054 0
s04 0.6336668 -2.6447011 0
s05 1.2141256 -1.0870115 0
s06 1.4940753 -0.9263137 0
s07 1.7233574 -1.7290162 0
s08 0.9815908  3.1497698 0
```

```
s09 1.0866067 -0.3036178 0
s12 1.4498429 -0.1380151 0
s14 1.0159834 -0.2558882 0
s15 0.9046749 -0.4216555 0
s16 1.4752955 0.2216158 0
```

```
$se
      aj      bj      cj
s03 0.1155375 0.20441485 0
s04 0.1250574 0.47668423 0
s05 0.1526624 0.12175749 0
s06 0.1800035 0.09653309 0
s07 0.2457579 0.15253089 0
s08 0.2276118 0.58227949 0
s09 0.1333925 0.08662509 0
s12 0.1680297 0.06968662 0
s14 0.1275844 0.08936450 0
s15 0.1194251 0.10329955 0
s16 0.1764927 0.06974481 0
>
$vcml
$vcml[[1]]
# 省略
```

	A	B	C	D	E	F	G	H	I	J	K	L
1	studentID	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	A001		1	1	1	1	1	0	1	1	1	1
3	A002		1	1	1	0	1	0	0	1	0	0
4	A003		1	1	1	1	1	1	1	1	1	1
5	A004		1	1	1	0	1	0	0	0	0	0
6	A005		1	1	1	0	1	0	0	1	1	1
7	A006		1	1	0	1	0	0	0	0	0	1
8	A007		1	0	1	1	1	0	1	0	1	0
9	A008		0	1	1	1	1	0	0	1	1	0
10	A009		0	1	0	0	0	0	0	0	0	0
11	A010		1	1	1	1	1	0	1	1	1	1

```
> # 特性値の推定
```

```
> # 最尤推定値
```

```
> th.mle <- mlebme(resp=itemScores, ip=p.all$est)
```

```
> head(th.mle)
```

```
      est      sem      n
[1,]  2.4947583 1.3271304 11
[2,] -0.7873886 0.5763237 11
[3,]  3.9999379 1.9051382 11
[4,] -1.2792655 0.5933037 11
[5,] -0.1386948 0.5942225 11
[6,] -1.4734953 0.6085549 11
```

```
> # 期待事後推定値
```

```
> th.eap <- eap(resp=itemScores, ip=p.all$est, qu=normal.qu())
```

```
> head(th.eap)
```

```
      est      sem      n
[1,]  1.26809690 0.6508009 11
[2,] -0.58653249 0.5089365 11
[3,]  1.71697773 0.7015662 11
[4,] -0.96204634 0.5107556 11
[5,] -0.07748696 0.5242392 11
[6,] -1.10349864 0.5143460 11
```

```
>
```

```
>
```

```
> # 特性値のヒストグラム
```

```
> est.mle <- th.mle[,c("est")]
```

```
> est.eap <- th.eap[,c("est")]
```

```
> est.raw <- rowSums(itemScores)
```

```
>
```

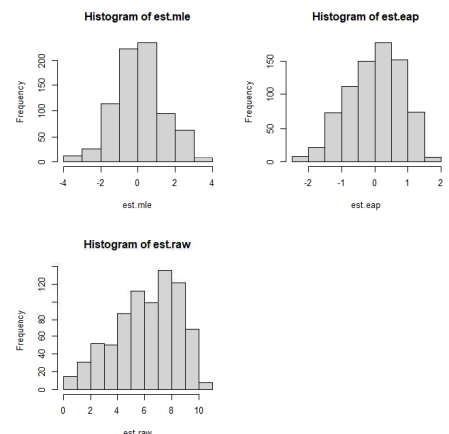
```
> par(mfrow=c(2, 2))
```

```
> hist(est.mle)
```

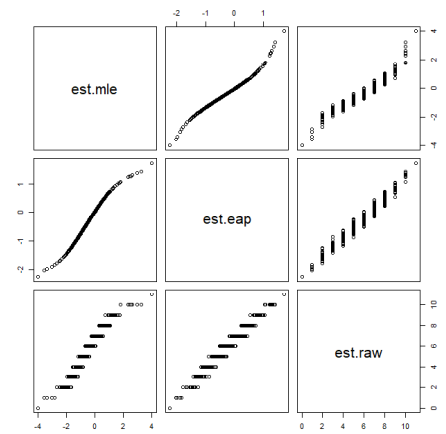
```
> hist(est.eap)
```

```
> hist(est.raw)
```

```
> par(mfrow=c(1, 1))
```



```
> # 特性値の散布図  
> ests <- cbind(est.mle, est.eap, est.raw)  
> pairs(ests)  
>
```



## 特性値の推定 — 項目パラメタが既知の場合

```
library(irtoys)
```

あらかじめirtoysパッケージをインストールしておく必要がある。

## 最尤推定値・ベイズモーダル推定値

```
mlebm(resp=データ行列, ip=項目パラメタ行列, method="推定方法")
```

method を"ML"または省略すると最尤推定値, "BM"とするとベイズモーダル推定値を計算する.  
"BM"とした場合は mu= と sigma= で事前分布 (正規分布) を指定する. デフォルトは標準正規分布.

## 期待事後推定値

```
eap(resp=データ行列, ip=項目パラメタ行列, qu=normal.qu())
```

qu は求積点と重みの指定をする. 標準正規分布の場合 qu=normal.qu() とする.

【重要】項目識別力パラメタ $a_j$ は,  $D=1.7$ を掛けた値である。 $a_j = D \cdot a_j^*$  の $a_j^*$ を使うときは,  $a_j^* \cdot 1.7$ 倍してから特性値の推定を行う必要がある。

```
> rm(list=ls())
> setwd("d:¥¥¥")

> d1 <- read.table("itemScores.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  studentID s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1      A001   1   1   1   1   1   0   1   1   1   1   1
2      A002   1   1   1   0   1   0   0   1   0   0   0
3      A003   1   1   1   1   1   1   1   1   1   1   1
4      A004   1   1   1   0   1   0   0   0   0   0   0
5      A005   1   1   1   0   1   0   0   1   1   1   0
6      A006   1   1   0   1   0   0   0   0   0   1   0
>
>
> # 0/1採点データを取り出して, matrix型で保存
> itemScores <- as.matrix(d1[,c("s03","s04","s05","s06","s07",
+                               "s08","s09","s12","s14","s15","s16")])
> rownames(itemScores) <- d1[,c("studentID")]
> head(itemScores)
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
A001   1   1   1   1   1   0   1   1   1   1   1
A002   1   1   1   0   1   0   0   1   0   0   0
A003   1   1   1   1   1   1   1   1   1   1   1
A004   1   1   1   0   1   0   0   0   0   0   0
A005   1   1   1   0   1   0   0   1   1   1   0
A006   1   1   0   1   0   0   0   0   0   1   0
>
```

	A	B	C	D	E	F	G	H	I	J	K	L	
1	studentID	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16	
2	A001		1	1	1	1	1	0	1	1	1	1	1
3	A002		1	1	1	0	1	0	0	1	0	0	0
4	A003		1	1	1	1	1	1	1	1	1	1	1
5	A004		1	1	1	0	1	0	0	0	0	0	0
6	A005		1	1	1	0	1	0	0	1	1	1	0
7	A006		1	1	0	1	0	0	0	0	0	1	0
8	A007		1	0	1	1	1	0	1	0	1	0	0
9	A008		0	1	1	1	1	0	0	1	1	0	1
10	A009		0	1	0	0	0	0	0	0	0	0	0
11	A010		1	1	1	1	1	0	1	1	1	1	1

```
> # 項目パラメータデータファイルの読み込み
> # 1PLロジスティックモデルを想定
```

```
> p1 <- read.table("ItemParameters.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
> head(p1)
```

```
  Item      aj      bj  cj
1 s03 1.157375 -1.3479054 0
2 s04 1.157375 -2.6447011 0
3 s05 1.157375 -1.0870115 0
4 s06 1.157375 -0.9263137 0
5 s07 1.157375 -1.7290162 0
6 s08 1.157375  3.1497698 0
```

	A	B	C	D	E
1	Item	aj	bj	cj	
2	s03	1.157375	-1.34791	0	
3	s04	1.157375	-2.6447	0	
4	s05	1.157375	-1.08701	0	
5	s06	1.157375	-0.92631	0	
6	s07	1.157375	-1.72902	0	
7	s08	1.157375	3.14977	0	
8	s09	1.157375	-0.30362	0	
9	s12	1.157375	-0.13802	0	
10	s14	1.157375	-0.25589	0	
11	s15	1.157375	-0.42166	0	
12	s16	1.157375	0.221616	0	
13					

```
> # 項目パラメータの取り出し, matrix型として保存
> itemParameters <- as.matrix(p1[,c("aj", "bj", "cj")])
> rownames(itemParameters) <- p1[,c("Item")]
> head(itemParameters)
      aj      bj  cj
s03 1.157375 -1.3479054 0
s04 1.157375 -2.6447011 0
s05 1.157375 -1.0870115 0
s06 1.157375 -0.9263137 0
s07 1.157375 -1.7290162 0
s08 1.157375  3.1497698 0
```

```
> # 項目識別力aj*を1.7倍する必要がある場合
```

```
> # 項目パラメータデータファイルの読み込み
> p2 <- read.table("ItemParameters_original.csv", header=TRUE, sep=";", fileEncoding="shift-jis")
```

```
> head(p2)
```

```
  Item      aj      bj  cj
1 s03 0.680809 -1.3479054 0
2 s04 0.680809 -2.6447011 0
3 s05 0.680809 -1.0870115 0
4 s06 0.680809 -0.9263137 0
5 s07 0.680809 -1.7290162 0
6 s08 0.680809  3.1497698 0
```

	A	B	C	D	E
1	Item	aj	bj	cj	
2	s03	0.680809	-1.34791	0	
3	s04	0.680809	-2.6447	0	
4	s05	0.680809	-1.08701	0	
5	s06	0.680809	-0.92631	0	
6	s07	0.680809	-1.72902	0	
7	s08	0.680809	3.14977	0	
8	s09	0.680809	-0.30362	0	
9	s12	0.680809	-0.13802	0	
10	s14	0.680809	-0.25589	0	
11	s15	0.680809	-0.42166	0	
12	s16	0.680809	0.221616	0	
13					

```
> # 項目パラメータの取り出し, matrix型として保存
> itemParameters2 <- as.matrix(p2[,c("aj", "bj", "cj")])
> head(p2)
```

```
  Item      aj      bj  cj
1 s03 0.680809 -1.3479054 0
2 s04 0.680809 -2.6447011 0
3 s05 0.680809 -1.0870115 0
4 s06 0.680809 -0.9263137 0
5 s07 0.680809 -1.7290162 0
6 s08 0.680809  3.1497698 0
```

```
> itemParameters2[,c("aj")] <- 1.7 * itemParameters2[,c("aj")]
> rownames(itemParameters2) <- p2[,c("Item")]
> head(itemParameters2)
```

```
      aj      bj  cj
s03 1.157375 -1.3479054 0
s04 1.157375 -2.6447011 0
s05 1.157375 -1.0870115 0
s06 1.157375 -0.9263137 0
s07 1.157375 -1.7290162 0
s08 1.157375  3.1497698 0
```



```
> # 特性値の推定
> # 項目識別力 $a_j^*$ を1.7倍する必要がある場合は, ip= itemParameters2を用いる
>
> library(ltm)
> library(irtos)
```

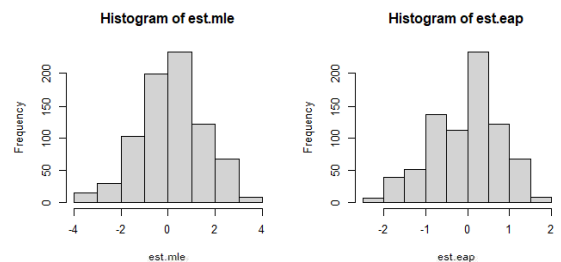
```
> th.mle <- mlebmme(resp=itemScores, ip=itemParameters)
> head(th.mle)
```

```
      est      sem  n
[1,]  2.42451143 1.2231941 11
[2,] -0.83050226 0.5960153 11
[3,]  3.99994747 1.7339911 11
[4,] -1.25108334 0.6132899 11
[5,]  0.01471514 0.6290329 11
[6,] -1.25108334 0.6132899 11
```

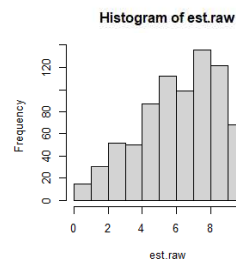
```
> th.eap <- eap(resp=itemScores, ip=itemParameters, qu=normal.qu())
> head(th.eap)
```

```
      est      sem  n
[1,]  1.25652038 0.6530764 11
[2,] -0.60910294 0.5220986 11
[3,]  1.79053933 0.7047696 11
[4,] -0.92445247 0.5231309 11
[5,]  0.04200225 0.5442996 11
[6,] -0.92445247 0.5231309 11
```

```
>
>
>
>
```

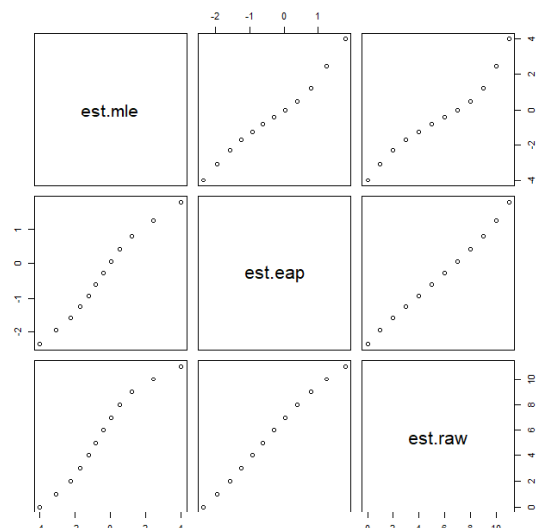


```
> # 特性値のヒストグラム
> est.mle <- th.mle[,c("est")]
> est.eap <- th.eap[,c("est")]
> est.raw <- rowSums(itemScores)
>
> par(mfrow=c(2, 2))
> hist(est.mle)
> hist(est.eap)
> hist(est.raw)
> par(mfrow=c(1, 1))
```



```
> # 特性値の散布図
> ests <- cbind(est.mle, est.eap, est.raw)
> pairs(ests)
>
```

```
# 1PLロジスティックモデルなので,
# 素点合計と特性値 $\theta$ が一対一対応する
```



## 局所独立性の確認

## 正答確率の推定

```
正答確率推定データ <- irf(ip= 項目パラメタ, x = 特性値)
```

## 実際の正誤と正答確率推定値の残差

```
残差データ <- 正誤データ - 正答確率推定データ
```

## Q3統計量

```
cor(残差データ)
```

絶対値0.2以上なら局所独立の仮定に問題ありと考える

## 項目適合度

```
itf(resp=データ, item=項目番号, ip=項目パラメタ, theta=特性値)
```

適合度検定。帰無仮説が「適合している」であることに注意

```
> setwd("d:¥¥¥")
> d1 <- read.table("irt_データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> head(d1)
  s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1    1    1    1    1    1    0    1    1    1    1    1
2    1    1    1    0    1    0    0    1    0    0    0
3    1    1    1    1    1    1    1    1    1    1    1
4    1    1    1    0    1    0    0    0    0    0    0
5    1    1    1    0    1    0    0    1    1    1    0
6    1    1    0    1    0    0    0    0    0    1    0
>
>
```

## # 項目母数の推定

```
> library(ltm)
> library(irtoys)
> p.all <- est(resp=d1, model="2PL", engine="ltm")
> colnames(p.all$est) <- c("aj", "bj", "cj")
> colnames(p.all$sse) <- c("aj", "bj", "cj")
> rownames(p.all$sse) <- rownames(p.all$est)
> p.all
$est
```

```
      aj      bj  cj
s03 0.7519081 -1.3479054 0
s04 0.6336668 -2.6447011 0
s05 1.2141256 -1.0870115 0
s06 1.4940753 -0.9263137 0
s07 1.7233574 -1.7290162 0
s08 0.9815908  3.1497698 0
s09 1.0866067 -0.3036178 0
s12 1.4498429 -0.1380151 0
s14 1.0159834 -0.2558882 0
s15 0.9046749 -0.4216555 0
s16 1.4752955  0.2216158 0
```

```
$sse
      aj      bj  cj
s03 0.1155375 0.20441485 0
s04 0.1250574 0.47668423 0
s05 0.1526624 0.12175749 0
s06 0.1800035 0.09653309 0
s07 0.2457579 0.15253089 0
s08 0.2276118 0.58227949 0
s09 0.1333925 0.08662509 0
s12 0.1680297 0.06968662 0
s14 0.1275844 0.08936450 0
s15 0.1194251 0.10329955 0
s16 0.1764927 0.06974481 0
```

	A	B	C	D	E	F	G	H	I	J	K
1	s03	s04	s05	s06	s07	s08	s09	s12	s14	s15	s16
2	1	1	1	1	1	0	1	1	1	1	1
3	1	1	1	0	1	0	0	1	0	0	0
4	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0	0	0	0
6	1	1	1	0	1	0	0	1	1	1	0
7	1	1	0	1	0	0	0	0	0	1	0
8	1	0	1	1	1	0	1	0	1	0	0
9	0	1	1	1	1	0	0	1	1	0	1
10	0	1	0	0	0	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1	1	1	1
12	1	1	1	1	1	0	0	1	0	0	0
13	1	1	1	1	1	0	1	1	1	0	0
14	1	1	1	1	1	0	1	1	1	1	1
15	0	1	1	1	0	0	0	1	0	1	1
16	1	0	1	1	1	0	1	1	0	0	0
17	1	1	1	0	0	0	0	1	1	0	1
18	1	1	1	1	1	0	0	0	1	1	1
19	1	1	1	1	1	0	0	1	0	1	1
20	1	1	1	0	1	0	0	0	0	1	0
21	0	1	1	1	1	1	1	0	1	1	1

```
$vcm
# 省略
```

```
> # 特性値の推定
```

```
> th.bme <- mlebme(resp=d1, ip=p.all$est)
> head(th.bme)
```

```
      est      sem  n
[1,]  2.4947583 1.3271304 11
[2,] -0.7873886 0.5763237 11
[3,]  3.9999379 1.9051382 11
[4,] -1.2792655 0.5933037 11
[5,] -0.1386948 0.5942225 11
[6,] -1.4734953 0.6085549 11
```

```
> th.eap <- eap(resp=d1, ip=p.all$est, qu=normal.qu())
> head(th.eap)
```

```
      est      sem  n
[1,]  1.26809690 0.6508009 11
[2,] -0.58653249 0.5089365 11
[3,]  1.71697773 0.7015662 11
[4,] -0.96204634 0.5107556 11
[5,] -0.07748696 0.5242392 11
[6,] -1.10349864 0.5143460 11
```

```
> # 正答確率の推定
```

```
> cr.est <- irf(ip= p.all$est, x = th.bme[, 1])
> round(head(cr.est$f), 2)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
[1,]  0.95 0.96 0.99 0.99 1.00 0.34 0.95 0.98 0.94 0.93 0.97
[2,]  0.60 0.76 0.59 0.55 0.84 0.02 0.37 0.28 0.37 0.42 0.18
[3,]  0.98 0.99 1.00 1.00 1.00 0.70 0.99 1.00 0.99 0.98 1.00
[4,]  0.51 0.70 0.44 0.37 0.68 0.01 0.26 0.16 0.26 0.32 0.10
[5,]  0.71 0.83 0.76 0.76 0.94 0.04 0.54 0.50 0.53 0.56 0.37
[6,]  0.48 0.68 0.38 0.31 0.61 0.01 0.22 0.13 0.22 0.28 0.08
```

```
>
```

```
> # 実際の正誤と正答確率推定値の残差
```

```
> cr.rsd <- d1 - cr.est$f
```

```
> round(head(cr.rsd), 2)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
1  0.05 0.04 0.01 0.01 0.00 -0.34 0.05 0.02 0.06 0.07 0.03
2  0.40 0.24 0.41 -0.55 0.16 -0.02 -0.37 0.72 -0.37 -0.42 -0.18
3  0.02 0.01 0.00 0.00 0.00 0.30 0.01 0.00 0.01 0.02 0.00
4  0.49 0.30 0.56 -0.37 0.32 -0.01 -0.26 -0.16 -0.26 -0.32 -0.10
5  0.29 0.17 0.24 -0.76 0.06 -0.04 -0.54 0.50 0.47 0.44 -0.37
6  0.52 0.32 -0.38 0.69 -0.61 -0.01 -0.22 -0.13 -0.22 0.72 -0.08
```

```
>
```

```
> # Q3統計量。絶対値0.2以上なら局所独立の仮定に検討の余地あり
```

```
> round(cor(cr.rsd, use="pairwise.complete.obs"), 2)
```

```
      s03 s04 s05 s06 s07 s08 s09 s12 s14 s15 s16
s03  1.00 -0.09 0.00 -0.03 -0.13 -0.06 -0.05 -0.06 -0.14 -0.10 -0.10
s04 -0.09 1.00 0.03 0.02 -0.07 -0.01 -0.11 -0.11 -0.05 -0.04 -0.07
s05  0.00 0.03 1.00 0.00 -0.22 0.00 -0.12 -0.12 -0.15 -0.13 -0.23
s06 -0.03 0.02 0.00 1.00 -0.14 -0.09 -0.12 -0.23 -0.17 -0.17 -0.20
s07 -0.13 -0.07 -0.22 -0.14 1.00 0.00 -0.04 -0.12 -0.02 -0.08 -0.12
s08 -0.06 -0.01 0.00 -0.09 0.00 1.00 0.01 -0.06 -0.09 -0.02 -0.11
s09 -0.05 -0.11 -0.12 -0.12 -0.04 0.01 1.00 -0.24 -0.12 -0.03 -0.16
s12 -0.06 -0.11 -0.12 -0.23 -0.12 -0.06 -0.24 1.00 -0.08 -0.16 -0.11
s14 -0.14 -0.05 -0.15 -0.17 -0.02 -0.09 -0.12 -0.08 1.00 -0.04 -0.13
s15 -0.10 -0.04 -0.13 -0.17 -0.08 -0.02 -0.03 -0.16 -0.04 1.00 -0.10
s16 -0.10 -0.07 -0.23 -0.20 -0.12 -0.11 -0.16 -0.11 -0.13 -0.10 1.00
```

## &gt; # 項目適合度

```
> itf(resp=d1, item=1, ip=p.all$est, theta=th.bme[, 1])
Statistic      DF      P-value
4.741022e+01  7.000000e+00  4.642388e-08
```

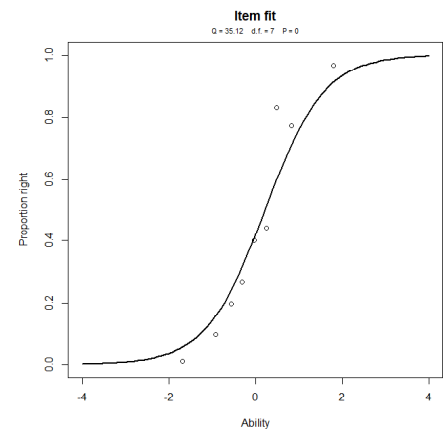
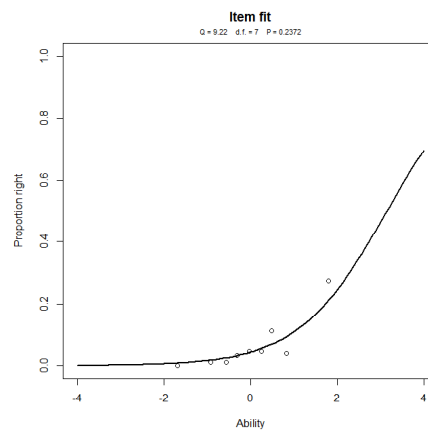
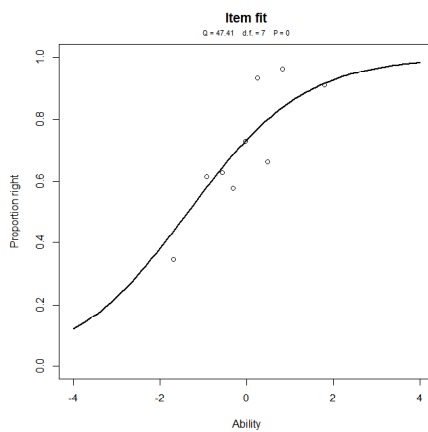
```
> itf(resp=d1, item=6, ip=p.all$est, theta=th.bme[, 1])
Statistic      DF      P-value
9.2211003  7.0000000  0.2371694
```

```
> itf(resp=d1, item=11, ip=p.all$est, theta=th.bme[, 1])
Statistic      DF      P-value
3.512343e+01  7.000000e+00  1.060259e-05
```

## 警告メッセージ:

```
itf(resp = d1, item = 11, ip = p.all$est, theta = th.bme[, 1]) で:
item fit statistic computed for a test of less than 20 items
```

## # 項目数が少ないと、警告メッセージが表示されることがある



&gt;

## 等化 — 共通項目法

```
library(irtoys)
```

あらかじめirtoysパッケージをインストールしておく必要がある。

```
sca(old.ip = 基準とする冊子の項目パラメタ行列,
    old.items = 基準冊子における共通項目の名前または列番号,
    new.ip = 基準に等化したい冊子の項目パラメタ行列,
    new.items = 基準に等化したい冊子における共通項目の名前または列番号,
    method = "等化法")
```

method には "MM" (Mean/Mean), "MS" (Mean/Sigma), "SL" (Stocking-Lord), "HB" (Haebara) を指定できる。デフォルトは "MS" である。  
method に "SL" または "HB" を用いた場合は, old.qu=norma.qu(), new.qu=norma.qu(), bec=FALSE などの指定も必要になる。

```
> setwd("d:¥¥¥")
>
> # 冊子Sのデータファイルの読み込み
> ds <- read.table("irt_データ1.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(ds)
  s03 s04 s05 s06 s07 s08 s09 s12
1    1    1    1    1    1    0    1    1
2    1    1    1    0    1    0    0    1
3    1    1    1    1    1    1    1    1
4    1    1    1    0    1    0    0    0
5    1    1    1    0    1    0    0    1
6    1    1    0    1    0    0    0    0

> # 人数
> nrow(ds)
[1] 380
>
```

冊子s データ

	A	B	C	D	E	F	G	H
1	s03	s04	s05	s06	s07	s08	s09	s12
2	1	1	1	1	1	0	1	1
3	1	1	1	0	1	0	0	1
4	1	1	1	1	1	1	1	1
5	1	1	1	0	1	0	0	0
6	1	1	1	0	1	0	0	1
7	1	1	0	1	0	0	0	0
8	1	0	1	1	1	0	1	0
9	0	1	1	1	1	0	0	1
10	0	1	0	0	0	0	0	0
11	1	1	1	1	1	0	1	1
12	1	1	1	1	1	0	0	1
13	1	1	1	1	1	0	1	1
14	1	1	1	1	1	0	1	1
15	0	1	1	1	0	0	0	1
16	1	0	1	1	1	0	1	1
17	1	1	1	0	0	0	0	1
18	1	1	1	1	1	0	0	0
19	1	1	1	1	1	0	0	1
20	1	1	1	0	1	0	0	0
21	0	1	1	1	1	1	1	0

```
> # 冊子tのデータファイルの読み込み
> dt <- read.table("irt_データ2.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(dt)
  t03 t04 t05 t06 t12 t14 t15 t16
1    1    1    0    0    0    1    1    0
2    0    1    1    1    1    1    0    0
3    0    1    1    1    0    1    0    0
4    1    1    1    1    1    1    1    1
5    1    0    1    1    1    1    1    0
6    0    0    0    1    0    0    0    0

> # 人数
> nrow(dt)
[1] 398
>
```

冊子t データ

	A	B	C	D	E	F	G	H
1	t03	t04	t05	t06	t12	t14	t15	t16
2	1	1	0	0	0	1	1	0
3	0	1	1	1	1	1	0	0
4	0	1	1	1	0	1	0	0
5	1	1	1	1	1	1	1	1
6	1	0	1	1	1	1	1	0
7	0	0	0	1	0	0	0	0
8	1	0	1	1	0	0	1	0
9	0	0	0	0	1	0	0	1
10	1	1	0	1	1	1	1	1
11	1	1	1	0	0	1	1	0
12	0	1	1	1	1	1	1	1
13	0	1	1	0	0	1	1	1
14	1	1	1	1	0	1	1	1
15	0	0	0	0	0	1	0	0
16	1	1	0	0	0	0	1	0
17	1	1	1	1	1	1	0	0
18	0	1	0	0	0	0	0	0
19	0	1	1	0	0	0	0	0
20	1	0	1	1	0	0	0	0
21	0	1	1	1	0	1	1	0

```
> # 四分相関係数の計算
> library(polycor)
> library(psych)
> t.cor.s <- polychoric(ds)
> t.cor.s <- t.cor.s$rho
> round(t.cor.s, 2)
```

```

      s03  s04  s05  s06  s07  s08  s09  s12
s03  1.00 -0.02 0.36 0.28 0.24 0.06 0.15 0.26
s04 -0.02 1.00 0.35 0.42 0.19 0.03 -0.03 0.21
s05 0.36 0.35 1.00 0.46 0.35 0.31 0.26 0.45
s06 0.28 0.42 0.46 1.00 0.41 0.09 0.32 0.33
s07 0.24 0.19 0.35 0.41 1.00 0.92 0.40 0.41
s08 0.06 0.03 0.31 0.09 0.92 1.00 0.36 0.22
s09 0.15 -0.03 0.26 0.32 0.40 0.36 1.00 0.23
s12 0.26 0.21 0.45 0.33 0.41 0.22 0.23 1.00
>
> t.cor.t <- polychoric(dt)
> t.cor.t <- t.cor.t$rho
> round(t.cor.t, 2)
      t03  t04  t05  t06  t12  t14  t15  t16
t03  1.00 0.14 0.29 0.40 0.34 0.09 0.23 0.28
t04 0.14 1.00 0.26 0.22 0.08 0.07 0.12 0.35
t05 0.29 0.26 1.00 0.58 0.35 0.24 0.24 0.30
t06 0.40 0.22 0.58 1.00 0.43 0.18 0.24 0.44
t12 0.34 0.08 0.35 0.43 1.00 0.34 0.27 0.55
t14 0.09 0.07 0.24 0.18 0.34 1.00 0.31 0.29
t15 0.23 0.12 0.24 0.24 0.27 0.31 1.00 0.39
t16 0.28 0.35 0.30 0.44 0.55 0.29 0.39 1.00
>
>

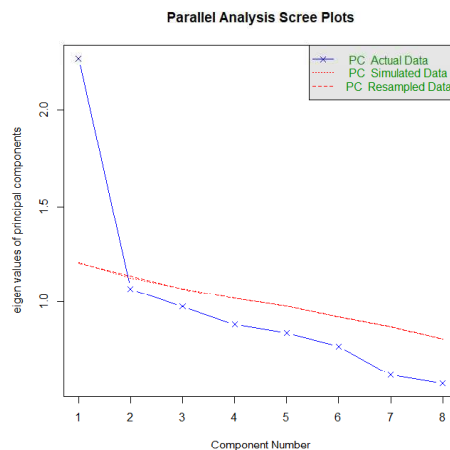
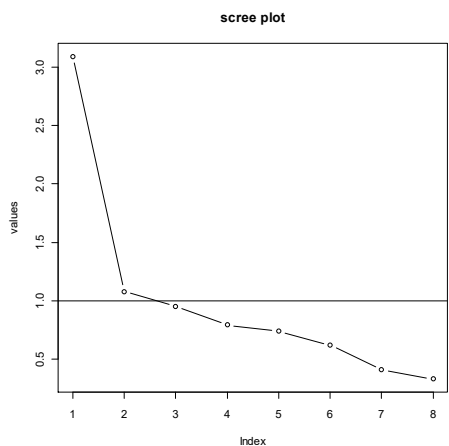
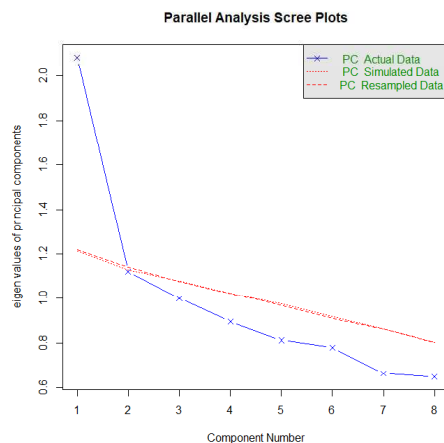
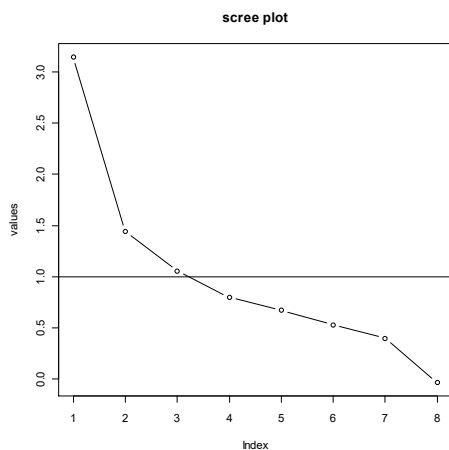
```

```
> # 一次元性の確認
```

```

> # VSS.scree を使う方法
> VSS.scree(t.cor.s)
> VSS.scree(t.cor.t)
>
> # fa.parallel関数を使う方法
> library(psych)
> fa.parallel(ds, fa = "pc")
> fa.parallel(dt, fa = "pc")

```



```

> # 項目母数の推定
> library(ltm)
> library(irtoys)

> # 冊子s
> (p.s <- est(resp=ds, model="2PL", engine="ltm"))
$est
      [,1]      [,2] [,3]
s03 0.7705687 -1.35005808 0
s04 0.7313849 -2.17329785 0
s05 1.7760631 -0.81834536 0
s06 1.5016486 -0.77167362 0
s07 1.4794851 -1.74290303 0
s08 0.7880403 3.53510139 0
s09 0.7771965 -0.41847844 0
s12 1.2080520 -0.06943932 0

$se
      [,1]      [,2] [,3]
[1,] 0.1835226 0.3046659 0
[2,] 0.2008428 0.5276427 0
[3,] 0.3662551 0.1233839 0
[4,] 0.2969631 0.1309896 0
[5,] 0.3409735 0.2637599 0
[6,] 0.3097197 1.1783263 0
[7,] 0.1764709 0.1684350 0
[8,] 0.2323375 0.1099113 0

$vcm
# 省略

> # 冊子t
> (p.t <- est(resp=dt, model="2PL", engine="ltm"))
$est
      [,1]      [,2] [,3]
t03 0.9003816 -1.1510996 0
t04 0.6039492 -2.9640459 0
t05 1.2714928 -1.1421325 0
t06 1.6534735 -1.0351572 0
t12 1.5431110 -0.2142970 0
t14 0.7095130 -0.3159867 0
t15 0.8331013 -0.4619254 0
t16 1.7019090 0.1444933 0

$se
      [,1]      [,2] [,3]
[1,] 0.1830130 0.22550849 0
[2,] 0.1880158 0.84133720 0
[3,] 0.2415768 0.17788000 0
[4,] 0.3139183 0.14187396 0
[5,] 0.2693993 0.09572488 0
[6,] 0.1569843 0.16815224 0
[7,] 0.1688423 0.15787306 0
[8,] 0.3138695 0.08975633 0

$vcm
# 省略

> # 冊子 t の項目母数を冊子sの尺度へ等化
> # 共通項目の番号 03, 04, 05, 06, 12

> # irtoys パッケージの読み込み
> library(irtoys)
> (p.tsa <- sca(old.ip=p.s$est, old.items=c(1:4,8),
+               new.ip=p.t$est, new.items=c(1:5),
+               method="MS"))

```

```
$slope
[1] 0.7753178
```

```
$intercept
[1] -0.02760581
```

```
$scaled.ip
      [,1]      [,2] [,3]
t03 1.1613065 -0.92007377 0
t04 0.7789699 -2.32568328 0
t05 1.6399634 -0.91312149 0
t06 2.1326397 -0.83018161 0
t12 1.9902949 -0.19375408 0
t14 0.9151254 -0.27259594 0
t15 1.0745288 -0.38574476 0
t16 2.1951115 0.08442242 0
```

```
> # 項目パラメタだけの取り出し
```

```
> (p.ts <- p.tsa$scaled.ip)
      [,1]      [,2] [,3]
t03 1.1613065 -0.92007377 0
t04 0.7789699 -2.32568328 0
t05 1.6399634 -0.91312149 0
t06 2.1326397 -0.83018161 0
t12 1.9902949 -0.19375408 0
t14 0.9151254 -0.27259594 0
t15 1.0745288 -0.38574476 0
t16 2.1951115 0.08442242 0
>
```

```
> # 共通項目のパラメタ推定値の統合
```

```
> # 各テストにおける共通項目のパラメタ
```

```
> (p.sc <- p.s$est[c(1:4, 8),])
      [,1]      [,2] [,3]
s03 0.7705687 -1.35005808 0
s04 0.7313849 -2.17329785 0
s05 1.7760631 -0.81834536 0
s06 1.5016486 -0.77167362 0
s12 1.2080520 -0.06943932 0
```

```
> (p.tsc <- p.ts[c(1:5),])
      [,1]      [,2] [,3]
t03 1.1613065 -0.9200738 0
t04 0.7789699 -2.3256833 0
t05 1.6399634 -0.9131215 0
t06 2.1326397 -0.8301816 0
t12 1.9902949 -0.1937541 0
```

```
> # 困難度パラメタの統合（幾何平均）
```

```
> p.c <- p.sc
> p.c[,1] <- sqrt(p.sc[,1] * p.tsc[,1])
```

```
> # 識別力パラメタの統合（算術平均）
```

```
> p.c[,2] <- (p.sc[,2] + p.tsc[,2])/2
>
```

```
> p.c
      [,1]      [,2] [,3]
s03 0.9459738 -1.1350659 0
s04 0.7548025 -2.2494906 0
s05 1.7066571 -0.8657334 0
s06 1.7895462 -0.8009276 0
s12 1.5506062 -0.1315967 0
>
```



## &gt; # 共通でない項目のパラメタ推定値

```
> (p.sr <- p.s$est[c(-1:-4, -8),])
      [, 1]      [, 2]      [, 3]
s07 1.4794851 -1.7429030      0
s08 0.7880403  3.5351014      0
s09 0.7771965 -0.4184784      0
```

```
> (p.tsr <- p.ts[c(-1:-5),])
      [, 1]      [, 2]      [, 3]
t14 0.9151254 -0.27259594      0
t15 1.0745288 -0.38574476      0
t16 2.1951115  0.08442242      0
>
```

## &gt; # 等化による全項目のパラメタ推定値

```
> p.all2 <- rbind(p.c, p.sr, p.tsr)
> p.all2 <- p.all2[order(rownames(p.all2)),]
> colnames(p.all2) <- c("aj", "bj", "cj")
> p.all2
      aj      bj      cj
s03 0.9459738 -1.13506592  0
s04 0.7548025 -2.24949057  0
s05 1.7066571 -0.86573343  0
s06 1.7895462 -0.80092762  0
s07 1.4794851 -1.74290303  0
s08 0.7880403  3.53510139  0
s09 0.7771965 -0.41847844  0
s12 1.5506062 -0.13159670  0
t14 0.9151254 -0.27259594  0
t15 1.0745288 -0.38574476  0
t16 2.1951115  0.08442242  0
>
>
>
```

## &gt; # 冊子tの特性値の推定

```
> # irtoys パッケージの読み込み
> library(irtoys)
> # 等化前
> th.bme.t <- mlebme(resp=dt, ip=p.t$est)
> th.bme.t.est <- th.bme.t[,c("est")]
> th.eap.t <- eap(resp=dt, ip=p.t$est, qu=normal.qu())
> th.eap.t.est <- th.eap.t[,c("est")]
```

## &gt; # 等化後

```
> th.bme.ts <- mlebme(resp=dt, ip=p.ts)
> th.bme.ts.est <- th.bme.ts[,c("est")]
> th.eap.ts <- eap(resp=dt, ip=p.ts, qu=normal.qu())
> th.eap.ts.est <- th.eap.ts[,c("est")]
>
```

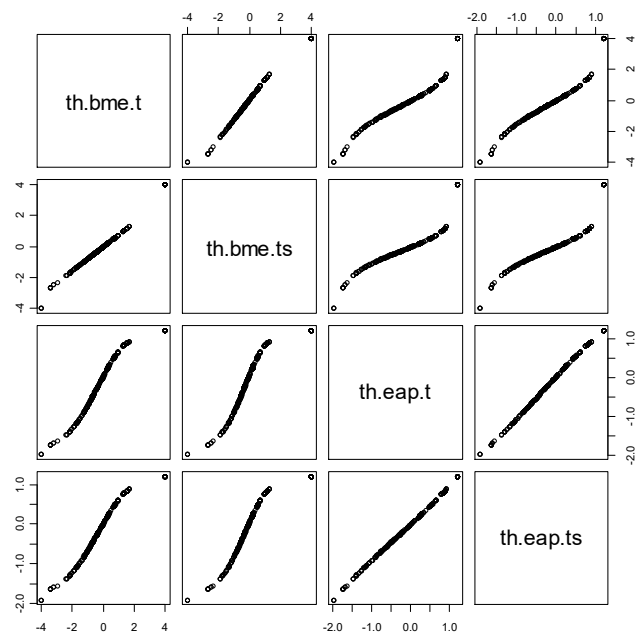
## &gt; # 特性値の散布図

```
> ests <- cbind(th.bme.t.est, th.bme.ts.est, th.eap.t.est, th.eap.ts.est)
> head(ests)
      th.bme.t.est th.bme.ts.est th.eap.t.est th.eap.ts.est
[1,] -1.2652978    -1.0086251   -0.90675320  -0.82829364
[2,] -0.1503728    -0.1441932   -0.08149246  -0.09018378
[3,] -0.7586096    -0.6157533   -0.54713042  -0.50443501
[4,]  3.9999352     3.9999494    1.20872178    1.19630733
[5,]  0.3409068     0.2367043    0.28316218    0.24147003
[6,] -2.0756327    -1.6368788   -1.35604568   -1.25386145
```

```
> pairs(ests)
> round(cor(ests), 3)
```

	th.bme.t	th.bme.ts	th.eap.t	th.eap.ts
th.bme.t	1.000	0.993	0.943	0.953
th.bme.ts	0.993	1.000	0.901	0.914
th.eap.t	0.943	0.901	1.000	0.999
th.eap.ts	0.953	0.914	0.999	1.000

```
# 同じ推定法であれば，等化の
# 前後の特性値はほぼ完全な相関
```



## DIF分析

## パッケージの読み込み

```
library(difR)
```

あらかじめdifRパッケージを読み込んでおく必要がある

## 一般化Lordカイ2乗法

```
difGenLord(データ名, group="グループ変数名", focal.names=c("焦点群1", "焦点群2", ...),
            nrFocal=焦点群の数, model="IRTモデル名")
```

## 一般化マンテル-ヘンツェル法

```
difGMH(データ名, group="グループ変数名", focal.names=c("焦点群1", "焦点群2", ...))
```

## 一般化ロジスティック回帰法

```
difGenLogistic(データ名, group="グループ変数名", focal.names=c("焦点群1", "焦点群2", ...))
```

## 3つの分析結果の比較

```
genDichoDif(データ名, group="グループ変数名", focal.names=c("焦点群1", "焦点群2", ...),
            method=c("GMH", "genLogistic", "genLord"), nrFocal=焦点群の数, model="IRTモデル名")
```

データは、分析に用いる項目と、グループ変数だけからなる。

IRTモデル名： 1PL, 2PL, 3PL

## カイ2乗値のプロット

```
plot(DIF分析オブジェクト名)
```

## スケーリングした項目母数

```
群名 <- c("参照群名", "焦点群1", "焦点群2", ...)
```

```
項目母数オブジェクト名 <- DIF分析オブジェクト名$itemParInit
```

```
rownames(項目母数オブジェクト名) <- NULL
```

```
items <- rep(DIF分析オブジェクト名$names, length(群名))
```

```
group <- rep(群名, each=length(DIF分析オブジェクト名$names))
```

```
(項目母数オブジェクト名 <- data.frame(items, 項目母数オブジェクト名, group))
```

## 項目特性曲線の描画

```
plot(DIF分析オブジェクト名, plot="itemCurve", item=変数番号, ref.name="参照群名")
```

itemは、"変数名"で指定してもよい

```
> setwd("d:¥¥")
```

```
> dA11 <- read.table("DIF10.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
```

```
> head(dA11)
```

```
Student.ID Age Group Gender Item1 Item2 Item3 Item4 Item5 Item6 Item7 Item8 Item9 Item10
1          1  14 Minhla female      1      1      1      0      0      1      0      0      0      0
2          1  14 Minhla female      1      1      1      1      1      1      1      1      1      1
3          2  14 Minhla  male      1      1      1      1      1      1      1      0      0      1
4          2  14 Minhla female      1      1      1      1      1      1      0      0      0      1
5          2  14 Minhla  male      1      1      1      1      1      1      1      1      1      1
6          3  14 Minhla  male      1      1      1      1      1      1      1      1      1      1>
```

```
> # dataset
```

```
> dEth <- dA11[, c("Item1", "Item2", "Item3", "Item4", "Item5",
+                 "Item6", "Item7", "Item8", "Item9", "Item10", "Group")]
```

```
>
```

```
> # DIF analysis
>
> library(difR)

> # 2PL model, "ltm" engine
>

> # Lord chi-squares, p values, and DIF items
> (rEth <- difGenLord(dEth, group="Group", focal.names=c("Sittway", "Minhla"),
+                   nrFocal=2, model="2PL"))
```

Detection of Differential Item Functioning using generalized Lord's method  
(2 focal groups), with 2PL model and without item purification

Engine 'ltm' for item parameter estimation

No set of anchor items was provided

No p-value adjustment for multiple comparisons

Generalized Lord's chi-square statistic:

	Stat.	P-value
Item1	5.1485	0.2724
Item2	2.2867	0.6832
Item3	0.6843	0.9533
Item4	6.2871	0.1787
Item5	5.1120	0.2760
Item6	4.2367	0.3749
Item7	7.1024	0.1306
Item8	5.9109	0.2059
Item9	11.1676	0.0247 *
Item10	5.9766	0.2009

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

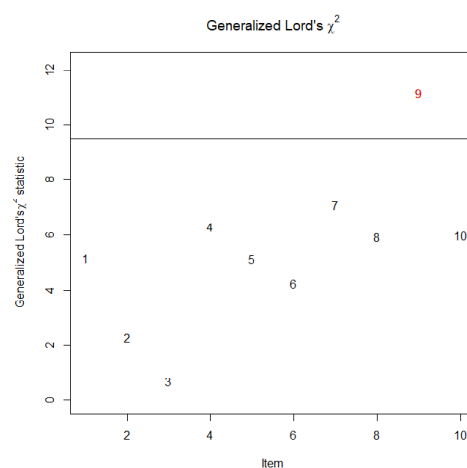
Detection threshold: 9.4877 (significance level: 0.05)

Items detected as DIF items:

Item9

Output was not captured!

```
> # plot of chi square statistics
> plot(rEth)
The plot was not captured!
>
```



```
> # M-H , p values, and DIF items
> (rMHEth <- difGMH(dEth, group="Group", focal.name=c("Sittway", "Minhla")))
```

Detection of Differential Item Functioning using Generalized Mantel-Haenszel method, without item purification and with 2 focal groups

Focal groups:

Sittway  
Minhla

Matching variable: test score

No set of anchor items was provided

No p-value adjustment for multiple comparisons

Generalized Mantel-Haenszel chi-square statistic:

	Stat.	P-value
Item1	3.3104	0.1911
Item2	3.5828	0.1667
Item3	0.0710	0.9651
Item4	1.3160	0.5179
Item5	1.7659	0.4136
Item6	7.8493	0.0197 *
Item7	4.1496	0.1256
Item8	3.3832	0.1842
Item9	3.3504	0.1873
Item10	0.2418	0.8861

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Detection threshold: 5.9915 (significance level: 0.05)

Items detected as DIF items:

Item6

Output was not captured!

```
> # Logistic, p values, and DIF items
> (rLogsEth <- difGenLogistic(dEth, group="Group", focal.name=c("Sittway", "Minhla")))
```

Detection of both types of Differential Item Functioning using Generalized logistic regression method, without item purification and with 2 focal groups

Focal groups:

Sittway  
Minhla

DIF flagging criterion: Likelihood ratio test

Matching variable: test score

No set of anchor items was provided

No p-value adjustment for multiple comparisons

Generalized Logistic regression statistic:

	Stat.	P-value
Item1	4.3921	0.3555
Item2	6.3923	0.1717
Item3	0.0854	0.9991
Item4	6.3593	0.1739
Item5	14.9842	0.0047 **
Item6	9.3672	0.0525 .
Item7	12.6210	0.0133 *
Item8	14.8718	0.0050 **
Item9	11.9417	0.0178 *
Item10	3.1416	0.5344

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Detection threshold: 9.4877 (significance level: 0.05)

Items detected as DIF items:

Item5  
Item7  
Item8  
Item9

Effect size (Nagelkerke's  $R^2$ ):

Effect size code:

'A': negligible effect  
'B': moderate effect  
'C': large effect

	$R^2$	ZT	JG
Item1	0.0127	A	A
Item2	0.0218	A	A
Item3	0.0003	A	A
Item4	0.0152	A	A
Item5	0.0282	A	A
Item6	0.0273	A	A
Item7	0.0237	A	A
Item8	0.0278	A	A
Item9	0.0242	A	A
Item10	0.0068	A	A

Effect size codes:

Zumbo & Thomas (ZT): 0 'A' 0.13 'B' 0.26 'C' 1  
Jodoin & Gierl (JG): 0 'A' 0.035 'B' 0.07 'C' 1

Output was not captured!

```
> # compare three results
> (gEth <- genDichoDif(dEth, group="Group", focal.names=c("Sittway", "Minhla"),
+   method=c("genLord", "GMH", "genLogistic"),
+   nrFocal=2, model="2PL"))
```

Comparison of DIF detection among multiple groups, using 3 methods

Generalized methods used: Lord's chi-square test, Mantel-Haenszel,  
Logistic regression

Matching variable: test score

No set of anchor items was provided

Parameters:

Significance level: 0.05

DIF effects tested by generalized logistic regression: both effects

DIF flagging criterion: Likelihood ratio test  
 Item response model: 2PL  
 Engine 'ltm' for item parameter estimation  
 Item purification: No

No p-value adjustment for multiple comparisons

Comparison of DIF detection results:

	Lord	M. -H.	Logistic	#DIF
Item1	NoDIF	NoDIF	NoDIF	0/3
Item2	NoDIF	NoDIF	NoDIF	0/3
Item3	NoDIF	NoDIF	NoDIF	0/3
Item4	NoDIF	NoDIF	NoDIF	0/3
Item5	NoDIF	NoDIF	DIF	1/3
Item6	NoDIF	DIF	NoDIF	1/3
Item7	NoDIF	NoDIF	DIF	1/3
Item8	NoDIF	NoDIF	DIF	1/3
Item9	DIF	NoDIF	DIF	2/3
Item10	NoDIF	NoDIF	NoDIF	0/3

Output was not captured!

```
>
>
>
> #all group values
> agnEth<-c("Myitkyinar","Sittway","Minhla")
>
>
>
> # discriminant and difficulty parameters after rescaling
> sIParEth <- rEth$itemParInit
> rownames(sIParEth) <-NULL
> items <- rep(rEth$names, length(agnEth))
> group <- rep(agnEth, each=length(rEth$names))
> (sIParEth <- data.frame(items, sIParEth, group))
```

	items	a	b	se. a.	se. b.	cov. a. b.	group
1	Item1	1.5380969	-1.83828193	0.4002920	0.3190842	0.109661508	Myitkyinar
2	Item2	2.0400659	-2.27464886	0.6365087	0.3862125	0.212230661	Myitkyinar
3	Item3	1.8825491	-2.24371338	0.5673737	0.3887347	0.192006403	Myitkyinar
4	Item4	1.0259302	-2.07074036	0.2852980	0.4638368	0.119533785	Myitkyinar
5	Item5	2.0394004	-1.03381906	0.4998997	0.1590402	0.054021960	Myitkyinar
6	Item6	1.6522546	-2.14141890	0.4639729	0.3791613	0.154066185	Myitkyinar
7	Item7	1.6141457	-0.90759031	0.3687412	0.1647958	0.039987083	Myitkyinar
8	Item8	1.4403277	-0.45802015	0.3197056	0.1367103	0.018433958	Myitkyinar
9	Item9	0.9098867	-0.10301649	0.2249735	0.1641164	0.004056429	Myitkyinar
10	Item10	0.3859422	<b>-2.49122292</b>	0.1874362	<b>1.1941683</b>	0.212574574	Myitkyinar
11	Item1	1.7485795	-2.07722535	0.6299268	0.2901455	0.087026849	Sittway
12	Item2	1.7896214	-2.00405041	0.6358558	0.2774932	0.075399445	Sittway
13	Item3	1.7366152	-2.36205236	0.6609115	0.3263546	0.122977902	Sittway
14	Item4	0.5252554	-2.32556132	0.2589588	0.7518019	0.144409051	Sittway
15	Item5	0.8503619	-1.66917263	0.3078722	0.3809860	0.050556450	Sittway
16	Item6	1.5452392	-2.50287354	0.5916726	0.3732614	0.142267478	Sittway
17	Item7	0.6821572	-0.86888040	0.2746271	0.3949254	-0.009275056	Sittway
18	Item8	0.7549751	-0.53868432	0.2879522	0.3872646	-0.034494214	Sittway
19	Item9	1.4346514	-0.04610927	0.5215074	0.3088151	-0.079355050	Sittway
20	Item10	1.0869772	-1.16786276	0.3611431	0.2927724	0.011058320	Sittway
21	Item1	1.4715802	-2.48778654	0.4395723	0.3773747	0.136199019	Minhla
22	Item2	<b>4.0055900</b>	-2.22321350	<b>2.8993142</b>	0.2457029	0.526190461	Minhla
23	Item3	1.4974351	-2.69569898	0.4888017	0.4433817	0.185256331	Minhla
24	Item4	1.2258118	-1.68406363	0.3235524	0.2642452	0.059505427	Minhla
25	Item5	1.5558869	-1.16685236	0.4192775	0.1804287	0.036124026	Minhla
26	Item6	1.0678651	-2.22144101	0.3130277	0.4000075	0.103828799	Minhla
27	Item7	0.7816893	-0.74675965	0.2257378	0.2377531	0.013165136	Minhla
28	Item8	0.6073449	-1.06547986	0.2036376	0.3266552	0.033399361	Minhla
29	Item9	0.2157375	-0.02383844	0.1708070	0.7936706	-0.057566944	Minhla
30	Item10	1.0188381	-1.24733840	0.2809263	0.2402838	0.038081481	Minhla

```
>
>
```

&gt;

```

> #draw ICCs
> par(mfrow=c(3, 3))
> plot(rEth, plot="itemCurve", item="Item2", ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item5", ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item6", ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item7", ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item8", ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item9", ref.name="Myitkyinar")
> plot(rEth, plot="itemCurve", item="Item10", ref.name="Myitkyinar")

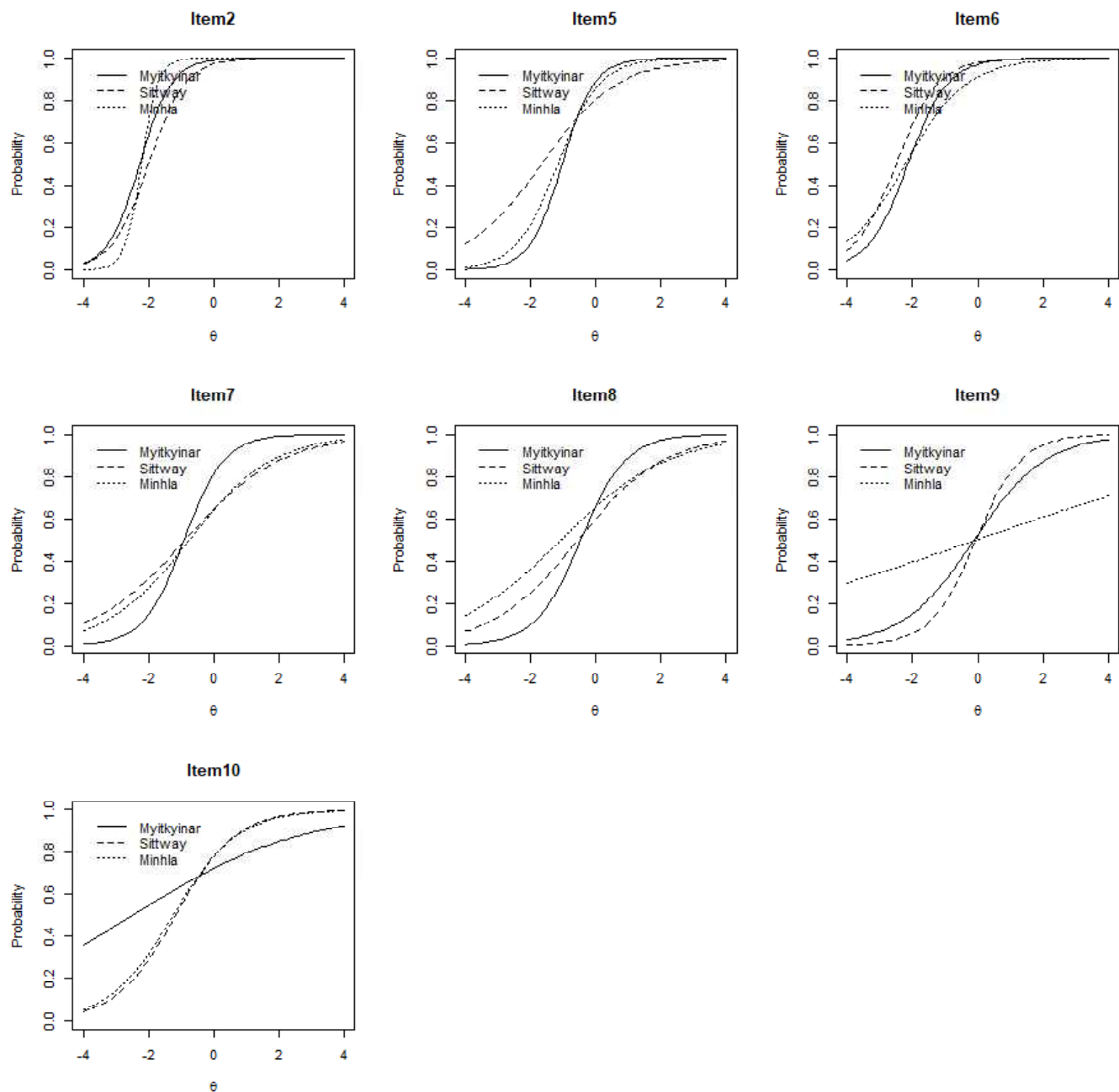
```

The plot was not captured!

```

> par(mfrow=c(1, 1))
>
>

```





## 24 付録

## 文字列関数 — 文字列の切り出し・結合・検索・置換

## 切り出し

`substr`(対象変数名, 始点, 終点)

始点, 終点で, 要素の何文字目から何文字目までを切り出すかを指定する

## 結合

`paste`(文字列の指定, `sep`="区切りの指定")

`paste0`(文字列の指定)

`paste`(ベクトル, `collapse`="区切りの指定")

`sep`="," でカンマ区切り, `sep`="" で区切りなし. `sep` を省略すると半角スペース区切りになる.  
`paste0` は, `sep`="" と指定したことと同じになる.  
 ベクトルの要素をつなげるときは, `collapse`を入れる. ""の間に何も書かないとベタづけになる.

## 検索

`grep`("検索する文字列", 検索対象, `fixed` = FALSE)

`grepl`("検索する文字列", 検索対象, `fixed` = FALSE)

`regexpr`("検索する文字列", 検索対象, `fixed` = FALSE)

`gregexpr`("検索する文字列", 検索対象, `fixed` = FALSE)

`grep`: 検索する文字列があったところの要素番号を返す.

`grepl`: 検索する文字列があるかどうか, すべての要素について TRUE, FALSE で返す.

`regexpr`: 検索する文字列があれば, 初出のもののみそれが何文字目かの値, なければ-1を返す.

`gregexpr`: 検索する文字列が複数あればそれがそれぞれ何文字目かの値, なければ-1を返す.

## 置換

`sub`("検索する文字列", "置き換える文字列", 検索対象, `fixed` = FALSE)

`gsub`("検索する文字列", "置き換える文字列", 検索対象, `fixed` = FALSE)

`sub`: 置換したい文字列があれば, 初出のもののみ置換する.

`gsub`: 置換したい文字列が複数あれば, それぞれすべてを置換する.

## 【注】

`fixed`=TRUE とすると完全一致のみ評価. カタカナや特殊記号を含む文字列を評価する場合, `fixed`=FALSE か `fixed`=TRUE の指定をしておかないとエラーになることがある.

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
> d1 <- read.table("文字列_データ.csv", header=TRUE, sep="," , fileEncoding="shift-jis")
> d1
  id      course
1  1      行動
2  2      臨床
3  3  行動コース
4  4  臨床コース
5  5   コース行動
6  6   コース臨床
7  7  行動行動コース
8  8  臨床臨床コース
>
```

	A	B
1	id	course
2	1	行動
3	2	臨床
4	3	行動コース
5	4	臨床コース
6	5	コース行動
7	6	コース臨床
8	7	行動行動コース
9	8	臨床臨床コース
10		

> # 文字列の切り出し

> d1\$course1 <- substr(d1\$course, 1, 3)

```
> d1
  id      course course1
1  1      行動   行動
2  2      臨床   臨床
3  3  行動コース 行動コ
4  4  臨床コース 臨床コ
5  5   コース行動 コース
6  6   コース臨床 コース
7  7  行動行動コース 行動行
8  8  臨床臨床コース 臨床臨
```

> # 文字列の結合

> d1\$course2 <- paste("心理", d1\$course)

> d1\$course3 <- paste("心理", d1\$course, sep="")

```
> d1
  id      course course1      course2      course3
1  1      行動   行動      心理 行動      心理行動
2  2      臨床   臨床      心理 臨床      心理臨床
3  3  行動コース 行動コ      心理 行動コース 心理行動コース
4  4  臨床コース 臨床コ      心理 臨床コース 心理臨床コース
5  5   コース行動 コース      心理 コース行動 心理コース行動
6  6   コース臨床 コース      心理 コース臨床 心理コース臨床
7  7  行動行動コース 行動行 心理 行動行動コース 心理行動行動コース
8  8  臨床臨床コース 臨床臨 心理 臨床臨床コース 心理臨床臨床
```

> # course2 は半角スペース区切り, course3 は区切りなしで, “心理” が結合されている。

> # ベクトルの要素の結合

> vitem <- c("x1", "x2")

> nitem <- paste(vitem, collapse=",")

```
> vitem
[1] "x1" "x2"
```

```
> nitem
[1] "x1,x2"
```

> data <- matrix(c(1, 1, 1, 0, 0, 1, 0, 0), nrow=4, ncol=2, byrow=TRUE)

```
> data
     [,1] [,2]
[1,]    1    1
[2,]    1    0
[3,]    0    1
[4,]    0    0
```

> paste(data, collapse="")

```
[1] "11001010"
```

> apply(data, 1, paste, collapse="")

```
[1] "11" "10" "01" "00"
```

> apply(data, 2, paste, collapse="")

```
[1] "1100" "1010"
```

## &gt; # 文字列の検索

```

> grep("行動", d1$course, fixed = FALSE)
[1] 1 3 5 7
# "行動"を含む行番号を表示

> grep("行動", d1$course, fixed = FALSE)
[1] TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE
# 各行が"行動"を含むかどうかを表示

> regexpr("行動", d1$course, fixed = FALSE)
[1] 1 -1 1 -1 4 -1 1 -1
attr(,"match.length")
[1] 2 -1 2 -1 2 -1 2 -1
# "行動"を含めばそれが何文字目か,
# 含まなければ-1を表示

> gregexpr("行動", d1$course, fixed = FALSE)
[[1]]
[1] 1
attr(,"match.length")
[1] 2
# "行動"を複数含めばそれぞれ何文字目か,
# 含まなければ-1を表示

# 中略

[[6]]
[1] -1
attr(,"match.length")
[1] -1

[[7]]
[1] 1 3
attr(,"match.length")
[1] 2 2

[[8]]
[1] -1
attr(,"match.length")
[1] -1
>

```

## &gt; # 文字列の置換

```

> sub("行動", "教心", d1$course, fixed = FALSE)
[1] "教心" "臨床" "教心コース" "臨床コース"
[5] "コース教心" "コース臨床" "教心行動コース" "臨床臨床コース"
# 同一要素内の1つめの"行動"は"教心"に置換されるが、それ以降のものは置換されない。

> gsub("行動", "教心", d1$course, fixed = FALSE)
[1] "教心" "臨床" "教心コース" "臨床コース"
[5] "コース教心" "コース臨床" "教心教心コース" "臨床臨床コース"
>
# 同一要素内のすべての"行動"が"教心"に置換される。

```

## 演算記号・算術関数

```
> x <- 5
> y <- 2
>
```

```
> # 和
> x + y
[1] 7
>
```

```
> # 差
> x - y
[1] 3
>
```

```
> # 積
> x * y
[1] 10
>
```

```
> # 除算
> x / y
[1] 2.5
>
```

```
> # 商
> x %% y
[1] 2
>
```

```
> # 余り
> x %% y
[1] 1
>
```

```
> # べき乗
> x ^ y
[1] 25
>
```

```
> # 平方根
> sqrt(x)
[1] 2.236068
>
```

```
> # 指数
> exp(x)
[1] 148.4132
>
```

```
> # 自然対数
> log(x)
[1] 1.609438
>
```

```
> # 三角関数
```

```
> sin(x)
[1] -0.9589243
```

```
> cos(x)
[1] 0.2836622
```

```
> tan(x)
[1] -3.380515
> u <- 0.5
```

```
> asin(u)
[1] 0.5235988
```

```
> acos(u)
[1] 1.047198

> atan(u)
[1] 0.4636476
>
>

> # 最大値・最小値
> a <- c(1, 3, 2, 5, 4)

> max(a)
[1] 5

> min(a)
[1] 1

>

> # 最大値・最小値の要素番号

> which.max(a)
[1] 4

> which.min(a)
[1] 1
>

> # 異なるベクトルの同じ位置にある要素の最大値・最小値
> b <- c(2, 4, 4, 4, 3)

> pmax(a, b)
[1] 2 4 4 5 4

> pmin(a, b)
[1] 1 3 2 4 3
>
>
```

## 切り上げ・切り下げ・四捨五入

```

> a <- 2.525
> b <- 2.515
> c <- -2.515
> d <- -2.525
>
>
> # 切り上げ, Xを超える最小の整数値
> ceiling(a)
[1] 3
> ceiling(b)
[1] 3
> ceiling(c)
[1] -2
> ceiling(d)
[1] -2
>
>
> # 切り下げ, Xを超えない最小の整数値
> floor(a)
[1] 2
> floor(b)
[1] 2
> floor(c)
[1] -3
> floor(d)
[1] -3
>
>
> # Xの小数点以下を削除
> trunc(a)
[1] 2
> trunc(b)
[1] 2
> trunc(c)
[1] -2
> trunc(d)
[1] -2
>
>
> # 四捨五入. ただし, X.5はXが偶数ならX, Xが奇数ならX+1に変換
> # round(x, p) x:数値, p: 小数点以下桁数
>
> p <- 2
> round(a, p)
[1] 2.52
> round(b, p)
[1] 2.52
> round(c, p)
[1] -2.52
> round(d, p)
[1] -2.52
>
>
> # 通常四捨五入
> p <- 2
> sign(a)*floor(abs(a)*10^p + 0.5)/10^p
[1] 2.53
> sign(b)*floor(abs(b)*10^p + 0.5)/10^p
[1] 2.52
> sign(c)*floor(abs(c)*10^p + 0.5)/10^p
[1] -2.52
> sign(d)*floor(abs(d)*10^p + 0.5)/10^p
[1] -2.53
>

```

```
>
> # 十方向に四捨五入
> p <- 2
> floor(a*10^p + 0.5)/10^p
[1] 2.53
> floor(b*10^p + 0.5)/10^p
[1] 2.52
> floor(c*10^p + 0.5)/10^p
[1] -2.51
> floor(d*10^p + 0.5)/10^p
[1] -2.52
>
>
```



## 集合関数

```
> x <- c("a", "c", "e")
> y <- c("a", "b", "c", "d")
>
```

```
> # 和集合
> (z <- union(x, y))
[1] "a" "c" "e" "b" "d"
>
```

```
> # 積集合
> (z <- intersect(x, y))
[1] "a" "c"
>
```

```
> # xの要素のうち, yに含まれるものを除いた集合
> (z <- setdiff(x, y))
[1] "e"
>
```

```
> # xの各要素はyの要素であるか
> (z <- is.element(x, y))
[1] TRUE TRUE FALSE
```

```
> (z <- x %in% y)
[1] TRUE TRUE FALSE
>
```

```
> # 集合として等しいか
> (z <- setequal(x, y))
[1] FALSE
```

```
> w <- c("a", "c", "e", "a", "c", "e")
> (z <- setequal(x, w))
[1] TRUE
>
>
```

## 比較演算子

```

> x <- 5
> y <- 5
> z <- 5.00000001
>
> w <- NA
>
> t <- c(1, 1, 1)
> u <- c(2, 2, 2)
> v <- c(1, 2, 3)
>
>

> # 比較演算子
>
> # 等しいか
> x == y
[1] TRUE

> identical(x, y)
[1] TRUE

> identical(x, z)
[1] FALSE
>

> # ほとんど等しいか
> all.equal(x, z)
[1] TRUE
>

> # 等しくないか
> x != y
[1] FALSE
>

> # 以上か
> x >= y
[1] TRUE
>

> # 超か
> x > y
[1] FALSE
>

> # 以下か
> x <= y
[1] TRUE
>

> # 未満か
> x < y
[1] FALSE
>
>
> # 欠測値か
> is.na(w)
[1] TRUE
>
>
>

```

## 論理演算子

```
> x <- 5
> y <- 5
> z <- 5.00000001
>
> w <- NA
>
> t <- c(1, 1, 1)
> u <- c(2, 2, 2)
> v <- c(1, 2, 3)
>
>
```

## &gt; # 論理演算子

```
>
> # かつ(スカラー)
> (x==5) && (y==4)
[1] FALSE
```

## &gt; # または(スカラー)

```
> (x==5) || (y==4)
[1] TRUE
```

## &gt; # かつ(ベクトル)

```
> # if文でベクトルを評価した場合は、先頭要素の真偽で判
> (t==v) & (u==v)
[1] FALSE FALSE FALSE
```

## &gt; # または(ベクトル)

```
> # if文でベクトルを評価した場合は、先頭要素の真偽で判
> (t==v) | (u==v)
[1] TRUE TRUE FALSE
>
```

## &gt; # でない

```
> !(x==4)
[1] TRUE
```

```
> !(u==v)
[1] TRUE FALSE TRUE
>
```

```
> !(v==2)
[1] TRUE FALSE TRUE
```

## ベクトルの作成

```
> # ベクトルの生成
```

```
> (x <- c(1, 4, 3))
[1] 1 4 3
```

```
> (y <- c(2, 6, 8))
[1] 2 6 8
```

```
> # ベクトルの各要素に名前をつける
```

```
> names(x) <- c("a", "b", "c")
```

```
> x
a b c
1 4 3
```

```
> names(y) <- c("d", "e", "f")
```

```
> y
d e f
2 6 8
```

```
>
```

```
> # ベクトルの連結
```

```
> (z <- c(x, y))
```

```
a b c d e f
1 4 3 2 6 8
```

```
> # 値が1ずつ増えていくベクトル
```

```
> (x <- c(1:9))
[1] 1 2 3 4 5 6 7 8 9
```

```
> (x <- seq(1, 9))
[1] 1 2 3 4 5 6 7 8 9
```

```
> # 指定した値ずつ増えていくベクトル
```

```
> (x <- seq(1, 9, by=2))
[1] 1 3 5 7 9
```

```
> # ベクトルの要素を逆順にする
```

```
> (y <- rev(x))
[1] 9 8 7 6 5 4 3 2 1
```

```
> # 同一ベクトルを指定個つなげる
```

```
> (x <- rep(c("a", "b", "c"), 2))
[1] "a" "b" "c" "a" "b" "c"
```

```
> # 各要素を指定回繰り返す
```

```
> (x <- rep(c("a", "b", "c"), each=2))
[1] "a" "a" "b" "b" "c" "c"
```

```
> # 重複要素のあるベクトルから素な要素だけを抽出する
```

```
> unique(x)
[1] "a" "b" "c"
```

## 行列の作成

> # データフレームを行列に変える

> d1 <- read.table("制御\_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")

> head(d1)

```
  x1 x2 x3
1 NA  3  2
2  3  3  2
3  1  3  2
4  3  3  2
5  3  3  3
6  3  3  2
```

> x <- as.matrix(d1)

> head(x)

```
  x1 x2 x3
[1,] NA  3  2
[2,]  3  3  2
[3,]  1  3  2
[4,]  3  3  2
[5,]  3  3  3
[6,]  3  3  2
```

> # 要素を入力して行列を作成する

> (x <- matrix(c(1, 3, 5, 2, 4, 6), nrow=3, ncol=2))

```
  [,1] [,2]
[1,]  1  2
[2,]  3  4
[3,]  5  6
```

>

> (y <- matrix(c(1, 1, 0, 1, 1, 0), nrow=3, ncol=2, byrow=TRUE))

```
  [,1] [,2]
[1,]  1  1
[2,]  0  1
[3,]  1  0
```

> # 行列の各行に名前をつける

> y <- matrix(c(1, 1, 0, 1, 1, 0), nrow=3, ncol=2, byrow=TRUE)

> rownames(y) <- c("r1", "r2", "r3")

> y

```
  [,1] [,2]
r1    1    1
r2    0    1
r3    1    0
```

> # 行列の各列に名前をつける

> y <- matrix(c(1, 1, 0, 1, 1, 0), nrow=3, ncol=2, byrow=TRUE)

> colnames(y) <- c("c1", "c2")

> y

```
  c1 c2
[1,]  1  1
[2,]  0  1
[3,]  1  0
```

> # 行列の各行・各列に名前をつける

> y <- matrix(c(1, 1, 0, 1, 1, 0), nrow=3, ncol=2, byrow=TRUE)

> dimnames(y) <- list(c("r1", "r2", "r3"), c("c1", "c2"))

> y

```
  c1 c2
r1  1  1
r2  0  1
r3  1  0
```

```
> # ベクトルを行列に変形する
```

```
> z <- c(1, 2, 3, 4, 5, 6)
```

```
> dim(z) <- c(3, 2)
```

```
> z
      [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
```

```
>
> dim(z) <- c(2, 3)
```

```
> z
      [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
```

```
> # 行列をベクトルに変形する
```

```
> (y <- matrix(c(1, 2, 3, 4, 5, 6), nrow=3, ncol=2, byrow=TRUE))
```

```
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

```
> (z <- as.vector(y))
```

```
[1] 1 3 5 2 4 6
```

```
> # 単位行列
```

```
> diag(3)
```

```
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
```

```
> # 対角行列
```

```
> diag(c(1, 2, 3))
```

```
      [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
```

## ベクトル・行列の要素の評価

> # ベクトル1の各要素が、ベクトル2のいずれかの要素であるか否かを返す

```
> c(1:6) %in% c(1, 4, 5)
[1] TRUE FALSE FALSE TRUE TRUE FALSE
```

> # ベクトル1の要素が、ベクトル2に初めて出てくる要素番号を返す

```
> match(c(1), 1:10)
[1] 1
```

> # ベクトル1の要素を含む、ベクトル2の要素番号をすべて返す

```
> grep(c(1), 1:10)
[1] 1 10
```

> # 文字データの長さを返す

```
> nchar(c("a", "bcd", "ef"))
[1] 1 3 2
```

> # ベクトル1の各要素を含む、ベクトル2の要素が1つであるとき、その要素番号を返す

> # 完全一致が部分一致に優先される

> # 一致する部分がなかったり、複数一致する場合は0を返す

```
> charmatch("m", c("mean", "median", "mode"))
[1] 0
```

```
> charmatch("med", c("mean", "median", "mode"))
[1] 2
```

```
> charmatch(c("", "ab", "ab"), c("abc", "ab"))
[1] 0 2 2
```

> # ベクトル1の各要素を含む、ベクトル2の要素が1つであるとき、その要素番号を返す

> # 完全一致が部分一致に優先される

> # dup=FALSE とすると、一度 match された要素は除外される

> # 一致する部分がなかったり、複数一致する場合はNAを返す

```
> pmatch("m", c("mean", "median", "mode"))
[1] NA
```

```
> pmatch("med", c("mean", "median", "mode"))
[1] 2
```

```
> pmatch(c("", "ab", "ab"), c("abc", "ab"), dup = FALSE)
[1] NA 2 1
```

```
> pmatch(c("", "ab", "ab"), c("abc", "ab"), dup = TRUE)
[1] NA 2 2
```

> #factor型データのレベル番号を要素とするベクトルの作成

```
> (x <- c("a", "b", "c", "d", "a", "b", "c"))
[1] "a" "b" "c" "d" "a" "b" "c"
```

```
> x <- factor(x)
```

```
> (y <- pmatch(x, levels(x), dup=TRUE))
[1] 1 2 3 4 1 2 3
```

```
> # 条件を満たすベクトル, 行列の要素の位置
```

```
> # 奇数データベクトルの作成
> (vodd <- seq(1, 19, by=2))
[1] 1 3 5 7 9 11 13 15 17 19
```

```
> # 条件を満たす要素の位置
> which(vodd>5)
[1] 4 5 6 7 8 9 10
```

```
> # 条件を満たす要素の位置を指定して, ベクトルから値を取り出す
> vodd[which(vodd>5)]
[1] 7 9 11 13 15 17 19
```

```
> # こうしても同じ
> vodd[(vodd>5)]
[1] 7 9 11 13 15 17 19
```

```
> # 行列の場合
> (mnum <- matrix(c(1:15), nrow=3, ncol=5, byrow=TRUE))
      [,1] [,2] [,3] [,4] [,5]
[1,]    1    2    3    4    5
[2,]    6    7    8    9   10
[3,]   11   12   13   14   15
```

```
> # 条件 (3で割り切れる) を満たす要素の行列要素番号
```

```
> # 要素番号で表示
> which(mnum %% 3 == 0)
[1] 2 6 7 11 15
>
> # 行番号, 列番号で表示
> which(mnum %% 3 == 0, arr.ind=TRUE)
      row col
[1,]    2    1
[2,]    3    2
[3,]    1    3
[4,]    2    4
[5,]    3    5
```

```
> # 条件 (3で割り切れる) を満たす行列要素の位置を指定して値を取り出す
> mnum[which(mnum %% 3 == 0)]
[1] 6 12 3 9 15
```

```
> # こうしても同じ
> mnum[(mnum %% 3 == 0)]
[1] 6 12 3 9 15
```



## 行列演算

&gt; # 行列の生成

&gt; (x &lt;- matrix(c(1, 3, 5, 2, 4, 6), nrow=3, ncol=2))

	[, 1]	[, 2]
[1, ]	1	2
[2, ]	3	4
[3, ]	5	6

&gt; (y &lt;- matrix(c(1, 1, 0, 1, 1, 0), nrow=3, ncol=2, byrow=TRUE))

	[, 1]	[, 2]
[1, ]	1	1
[2, ]	0	1
[3, ]	1	0

&gt; # 行列の足し算

&gt; (x+y)

	[, 1]	[, 2]
[1, ]	2	3
[2, ]	3	5
[3, ]	6	6

&gt; # 行列の引き算

&gt; (x-y)

	[, 1]	[, 2]
[1, ]	0	1
[2, ]	3	3
[3, ]	4	6

&gt; # 要素の2乗

&gt; x^2

	[, 1]	[, 2]
[1, ]	1	4
[2, ]	9	16
[3, ]	25	36

&gt; # 行列のかけ算

&gt; (z &lt;- t(x) %\*% y)

	[, 1]	[, 2]
[1, ]	6	4
[2, ]	8	6

&gt; (w &lt;- x %\*% t(y))

	[, 1]	[, 2]	[, 3]
[1, ]	3	2	1
[2, ]	7	4	3
[3, ]	11	6	5

&gt; # クロネッカー積

&gt; x %x% y

	[, 1]	[, 2]	[, 3]	[, 4]
[1, ]	1	1	2	2
[2, ]	0	1	0	2
[3, ]	1	0	2	0
[4, ]	3	3	4	4
[5, ]	0	3	0	4
[6, ]	3	0	4	0
[7, ]	5	5	6	6
[8, ]	0	5	0	6
[9, ]	5	0	6	0

```
> # 転置行列
```

```
> t(x)
      [,1] [,2] [,3]
[1,]     1     3     5
[2,]     2     4     6
```

```
> # 対角要素の取り出し
```

```
> diag(w)
[1] 3 4 5
```

```
> # 対角要素だけを残した行列の作成
```

```
> diag(diag(w))
      [,1] [,2] [,3]
[1,]     3     0     0
[2,]     0     4     0
[3,]     0     0     5
```

```
> # 三角行列
```

```
> u <- v <- w
> u[lower.tri(u)] <- 0
> v[upper.tri(v)] <- 0
```

```
> u
      [,1] [,2] [,3]
[1,]     3     2     1
[2,]     0     4     3
[3,]     0     0     5
```

```
> v
      [,1] [,2] [,3]
[1,]     3     0     0
[2,]     7     4     0
[3,]    11     6     5
```

```
> # 行列式
```

```
> det(z)
[1] 4
```

```
> det(w)
[1] -4.440892e-16
```

```
> round(det(w), 4)
[1] 0
```

```
> # 逆行列
```

```
> (invz <- solve(z))
      [,1] [,2]
[1,]  1.5 -1.0
[2,] -2.0  1.5
```

```
> # 一般逆行列
```

```
> library(MASS)
> (ginv(w))
      [,1] [,2] [,3]
[1,] -0.08333333 8.257284e-16 0.08333333
[2,]  1.16666667 3.333333e-01 -0.50000000
[3,] -1.25000000 -3.333333e-01 0.58333333
```

```

> # 固有値分解
> eigen(z)
$values
[1] 11.6568542  0.3431458

$vectors
      [,1]      [,2]
[1,] 0.5773503 -0.5773503
[2,] 0.8164966  0.8164966

> # 特異値分解
> svd(x)
$d
[1] 9.5255181 0.5143006

$u
      [,1]      [,2]
[1,] -0.2298477  0.8834610
[2,] -0.5247448  0.2407825
[3,] -0.8196419 -0.4018960

$v
      [,1]      [,2]
[1,] -0.6196295 -0.7848945
[2,] -0.7848945  0.6196295

> # QR分解
> qr(x)
$qr
      [,1]      [,2]
[1,] -5.9160798 -7.4373574
[2,]  0.5070926  0.8280787
[3,]  0.8451543  0.9935832

$rank
[1] 2

$qlaux
[1] 1.169031 1.113104

$pivot
[1] 1 2

attr(,"class")
[1] "qr"

> # コレスキー分解
> chol(z)
      [,1]      [,2]
[1,] 2.44949  1.632993
[2,] 0.00000  1.825742

> chol(w)
      [,1]      [,2]      [,3]
[1,] 1.732051  1.154701  0.5773503
[2,] 0.000000  1.632993  1.4288690
[3,] 0.000000  0.000000  1.6201852

```

## 確率関数

## # 確率関数

dxxxxx : 確率密度関数  
 pxxxxx : 確率分布関数  
 qxxxxx : 分位点関数  
 rxxxxx : 乱数

## # ベータ分布

dbeta(x, shape1, shape2), pbeta(x, shape1, shape2), qbeta(p, shape1, shape2), rbeta(n, shape1, shape2)

## # 二項分布

dbinom(x, n0, p0), pbinom(x, n0, p0), qbinom(p, n0, p0), rbinom(n, n0, p0)

## # カイ 2 乗分布

dchisq(x, df), pchisq(x, df), qchisq(p, df), rchisq(n, df)

## # 指数分布

dexp(x, rate), pexp(x, rate), qexp(p, rate), rexp(n, rate)

## # ガンマ分布

dgamma(x, shape, rate), pgamma(x, shape, rate), qgamma(p, shape, rate), rgamma(n, shape, rate)

## # 正規分布

dnorm(x, mean, sd), pnorm(x, mean, sd), qnorm(p, mean, sd), rnorm(n, mean, sd)

## # F 分布

df(x, df1, df2), pf(x, df1, df2), qf(p, df1, df2), rf(n, df1, df2)

## # t 分布

dt(x, df), pt(x, df), qt(p, df), rt(n, df)

## # 一様分布

dunif(x, min, max), punif(x, min, max), qunif(p, min, max), runif(n, min, max)

## # ワイブル分布

dweibull(x, shape, scale), pweibull(x, shape, scale), qweibull(p, shape, scale), rweibull(n, shape, scale)

# 非心分布は、非心パラメタ ncp= を指定すればよい。

```
> dt(2.7, df=3)
```

```
[1] 0.03124146
```

```
> pt(2.7, df=30)
```

```
[1] 0.9943578
```

```
> pt(2.7, df=30, ncp=1)
```

```
[1] 0.9436329
```

```
> qt(.995, df=30)
```

```
[1] 2.749996
```

```
> rt(n=5, df=30)
```

```
[1] 1.331593 1.638198 0.912146 -1.974527 -2.168895
```

```
>
```

## 多変量正規乱数の発生

母集団値を指定した標本の発生

```
library(MASS)
Z <- mvrnorm(n, M, S, tol = 1e-6, empirical = FALSE)
```

標本値を指定した標本の発生

```
library(MASS)
Z <- mvrnorm(n, M, S, tol = 1e-6, empirical = TRUE)
```

MASSパッケージは最初からインストールされている.

```

個体数          n
平均ベクトル    M
共分散行列      S
を指定する.
```

標準偏差ベクトル  $s$  と 相関係数行列  $R$  から 共分散行列  $S$  を作成するのが分かりやすい.

```

> setwd("d:¥¥")
>

> # 個体数
> n <- 245
>

> # 変数の数
> p <- 4
>

> # 標本平均ベクトル
> M <- c(15, 20, 18, 16)
>

> # 標本標準偏差ベクトル
> s <- c(5, 6, 5, 7)
>

> # 標本相関係数行列
> R <- matrix(c(
+ 1.0, 0.4, -0.3, -0.6,
+ 0.4, 1.0, 0.0, -0.3,
+ -0.3, 0.0, 1.0, 0.5,
+ -0.6, -0.3, 0.5, 1.0
+ ), p, p)
>

> # 標本共分散行列の計算
> D <- diag(s)
> S <- D %*% R %*% D
>

> # 母集団値を指定した2変量標準正規乱数の発生
> library(MASS)
> d1 <- mvrnorm(n, M, S, tol = 1e-6, empirical = FALSE)
> colnames(d1) <- c("x1", "x2", "x3", "x4")
> head(d1)
      x1      x2      x3      x4
[1,] 13.600134 19.83877 19.063658 20.2046010
[2,] 25.510293 19.20538  9.955946  6.3096060
[3,] 12.291883 12.83929 15.114641 27.1414162
[4,]  2.399108 10.44224 20.258530 24.6136281
[5,] 11.171811 20.51997 18.866531 18.3840800
[6,] 20.365074 36.70374 15.259708  0.4885998
>
```

```

> # 記述統計量
> dtmp <- d1
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp

  N Mean  SD   x1   x2   x3   x4
1 245 15.37 4.99  1.00  0.40 -0.30 -0.63
2 245 20.38 6.15  0.40  1.00  0.02 -0.30
3 245 18.20 4.58 -0.30  0.02  1.00  0.44
4 245 15.59 6.53 -0.63 -0.30  0.44  1.00

>
> # write.table(d1, "Z1.csv", row.names=FALSE, sep=" ", fileEncoding="shift-jis")
>
>
>
>
>
>
> # 標本値を指定した2変量標準正規乱数の発生
> library(MASS)
> d2 <- mvrnorm(n, M, S, tol = 1e-6, empirical = TRUE)
> colnames(d2) <- c("x1", "x2", "x3", "x4")
> head(d2)
      x1      x2      x3      x4
[1,] 28.242407 23.24588 12.789406  6.237946
[2,] 18.001552 18.94479  6.832597 14.702094
[3,]  2.473954 26.19450 26.504034 25.892094
[4,] 20.085287 21.52668 17.093125  6.084849
[5,] 22.069442 27.97013 15.041955  7.074444
[6,] 10.253586 14.12464 24.115007 24.726687

>
> # 記述統計量
> dtmp <- d2
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp

  N Mean  SD   x1   x2   x3   x4
1 245   15   5   1.0   0.4  -0.3  -0.6
2 245   20   6   0.4   1.0   0.0  -0.3
3 245   18   5  -0.3   0.0   1.0   0.5
4 245   16   7  -0.6  -0.3   0.5   1.0

>
>
> # write.table(d2, "Z2.csv", row.names=FALSE, sep=" ", fileEncoding="shift-jis")
>
>

```

## 項目反応モデルに基づく0/1データの発生

## irtoysパッケージを使う方法

```
detach("package:psych")
library(irtoys)
データ名 <- sim(ip=項目パラメタ行列, x=能力値ベクトル)
```

あらかじめirtoysパッケージをインストールしておく必要がある。irtoysパッケージのsim関数を使う。項目パラメタ行列は、行方向に項目を取り、識別力(a), 困難度(b), あて推量(c)の3列からなる行列。2PLモデルのときは、あて推量パラメタの値をすべて0に設定する。

## psychパッケージを使う方法

```
detach("package:irtoys")
library(irtoys)
オブジェクト名 <- sim.irt(nvar=項目数, n=個体数, a=識別力, d=困難度, c=あて推量, theta=能力値)
データ名 <- オブジェクト名$items
```

あらかじめipsychパッケージをインストールしておく必要がある。psyパッケージのsim.irt関数を使う。2PLモデルのときは、あて推量パラメタの値をすべて0に設定する。

## 【重要！】

irtoysパッケージにも、psychパッケージにも、simという名前の関数があるので、一方のパッケージを使うときは、他方のパッケージを取り外しておくのが良い。

```
> rm(list=ls())
>
> setwd("d:¥¥scripts¥¥")
```

```
> # 個体数
> ns <- 1000
```

```
> # 項目数
> ni <- 20
```

```
> # 項目識別力
> aL <- 0.2
> aH <- 2
> a <- runif(ni, aL, aH)
```

```
> # 項目困難度
> bL <- -2.4
> bH <- 2.4
> b <- runif(ni, bL, bH)
```

```
> #  $\gamma$  (あて推量パラメタ)
> c <- rep(0, ni)
>
```

```
> #  $\theta$ 
> theta <- rnorm(ns)
```

```

> # irtoys パッケージ
> detach("package:psych")

detach("package:psych") でエラー: 無効な 'name' 引数です

> library(irtoys)

> # パラメタ値
> (abc <- cbind(a, b, c))
      a      b      c
[1,] 0.3548740 1.3100336 0
[2,] 1.6516841 -1.6007484 0
[3,] 0.5331020 -2.0386886 0
[4,] 0.4072748 2.2815836 0
[5,] 0.6050733 2.3217993 0
[6,] 1.3102981 0.1314779 0
[7,] 1.9314691 1.2579182 0
[8,] 1.0195482 0.7493610 0
[9,] 0.4823217 -2.2525355 0
[10,] 0.8388784 0.8041410 0
[11,] 0.9463370 -1.5890344 0
[12,] 1.2605833 1.2025271 0
[13,] 1.3065001 -1.9461953 0
[14,] 0.5934825 2.3430172 0
[15,] 1.1020518 -1.0440412 0
[16,] 1.0327555 0.3014254 0
[17,] 1.7413587 2.2671554 0
[18,] 1.0652509 2.3937353 0
[19,] 0.7999865 -2.0905235 0
[20,] 0.2778654 1.7751244 0

> head(theta)
[1] -2.44917438 -0.10362513 0.23303718 -1.67709759 -0.06559557 0.37971671

```

### > # 0/1データの発生

```

> d1 <- sim(ip=abc, x=theta)

> head(d1)
      [, 1] [, 2] [, 3] [, 4] [, 5] [, 6] [, 7] [, 8] [, 9] [, 10] [, 11] [, 12] [, 13] [, 14] [, 15] [, 16]
[1,]      0      0      1      0      0      0      0      0      0      0      0      0      0      0      0      0
[2,]      1      1      1      1      1      1      0      1      0      0      1      0      1      0      1      0
[3,]      0      1      1      0      0      0      0      1      1      0      1      0      1      1      1      1
[4,]      0      1      0      1      0      0      0      0      0      0      0      0      0      0      0      0
[5,]      1      1      1      0      1      1      0      0      1      1      1      1      1      0      1      0
[6,]      0      1      1      0      0      1      0      0      1      1      1      0      1      0      1      1
      [, 17] [, 18] [, 19] [, 20]
[1,]      0      0      0      1
[2,]      0      0      0      0
[3,]      0      0      1      1
[4,]      0      0      1      0
[5,]      0      0      1      1
[6,]      0      0      1      0

```

### > # 発生させたデータからパラメタ推定

```

> abc1 <- est(resp=d1, model="2PL", engine="ltn")
> a.irtoys <- abc1$est[, 1]
> b.irtoys <- abc1$est[, 2]
>
> theta1 <- ability(resp=d1, ip=abc1[1], method="MLE")
> theta.irtoys <- theta1[, "est"]

```



```
> # psychパッケージ
> detach("package:irtoys")
> library(psych)
```

次のパッケージを付け加えます: ‘psych’

以下のオブジェクトは ‘package:ltm’ からマスクされています:

```
factor.scores
```

以下のオブジェクトは ‘package:polycor’ からマスクされています:

```
polyserial
```

```
> # 0/1データの発生
```

```
> d2a <- sim.irt(nvar=ni, n=ns, a=a, d=b, c=c, theta=theta)
> d2 <- d2a$items
```

```
> head(d2)
      V1 V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13 V14 V15 V16 V17 V18 V19 V20
[1,]  0  0  0  0  0  0  0  0  1  0  0  0  0  0  0  0  0  0  1  0
[2,]  1  0  1  0  0  1  0  0  0  0  1  0  1  0  1  0  0  0  1  0
[3,]  1  1  1  0  0  1  0  1  1  0  1  1  1  0  0  0  0  0  1  1
[4,]  0  0  1  1  1  0  0  0  1  0  1  0  1  0  0  0  0  0  1  0
[5,]  0  1  1  0  0  1  0  0  1  1  0  0  1  0  0  0  0  0  1  0
[6,]  0  1  1  0  0  1  0  0  1  0  0  0  1  1  1  1  0  1  1  0
```

```
> # 発生させたデータからパラメタ推定
```

```
> detach("package:psych")
> library(irtoys)
```

```
> abc2 <- est(resp=d2, model="2PL", engine="ltm")
> a.psych <- abc2$est[,1]
> b.psych <- abc2$est[,2]
```

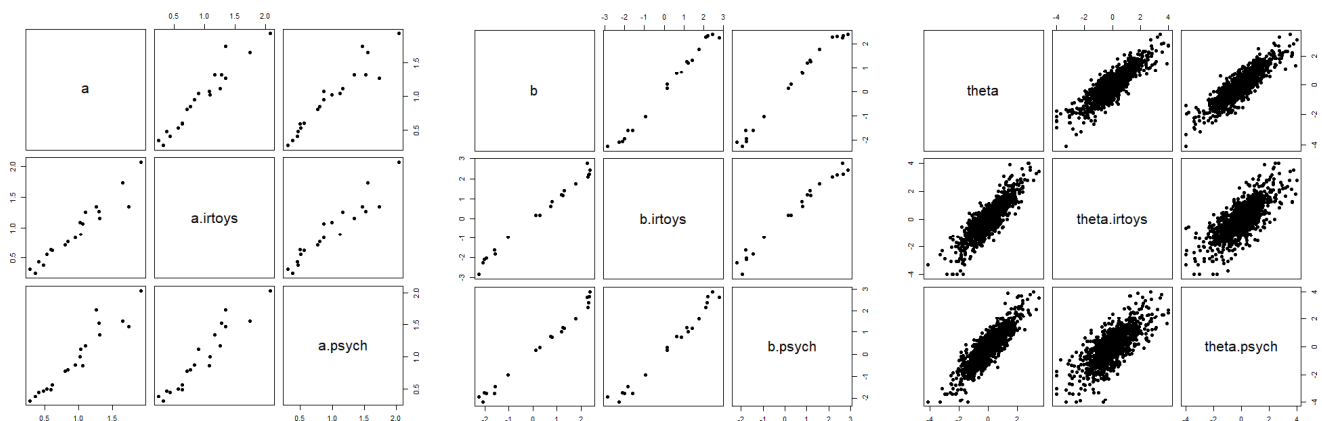
```
> theta2 <- ability(resp=d2, ip=abc2[1], method="MLE")
> theta.psych <- theta2[, "est"]
```

```
> # パラメタ値と推定値の図示
```

```
> pairs(cbind(a, a.irtoys, a.psych), pch=19)
```

```
> pairs(cbind(b, b.irtoys, b.psych), pch=19)
```

```
> pairs(cbind(theta, theta.irtoys, theta.psych), pch=19)
```



## 制御コマンド

評価する条件式に NA が入るとエラーになってしまうので、`is.na`(変数名) などとして、NAであればTRUE, NAでなければFALSEを返すように、まずしておくといよい。

```
for(ループ変数 in 値を変化させる範囲){
  ループ変数の値の変化に伴い実行される式の群
}
```

```
if (条件式1) { 条件式1がTRUEの時に実行される式の群 }
```

```
if (条件式1) {条件式1がTRUEの時に実行される式の群} else if (条件式2) {
  条件式1がFALSEで、条件式2がTRUEの時に実行される式の群
} else{ 以上の条件式がすべてFALSEのときに実行される式の群}
```

実行される式が1つであれば{}で囲わなくてもよい。

}とelseの間に改行しないほうがよい。

else if 以下は省略することができる。

if ((条件式1a) & (条件式1b)) とすれば、条件式1a および 条件式1bの両方がTRUEの場合、となる  
if ((条件式1a) | (条件式1b)) とすれば、条件式1a または 条件式1bの一方がTRUEの場合、となる

`ifelse`(ある変数についての条件式, 条件式がTRUEの時の当該変数の値, 条件式がFALSEの時の当該変数の値)

```
while(条件式){
  実行される式の群
}
```

式を実行する前に条件式の評価を行い、条件式がTRUEである間、式を実行する。

条件式が常にTRUEであれば、いつまでも関数が動いてしまうので注意。

```
repeat{
  実行される式の群
  if(条件式) break
}
```

式を実行した後に条件式の評価を行い、条件式がTRUEになれば終了する。

if(条件式) break を入れるのを忘れない。

`break`

`for`, `while`, `repeat` などのループから強制的に抜け出すコマンド。

```
> setwd("d:¥¥¥")
> d1 <- read.table("制御_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> d1
  x1 x2 x3
1  NA  3  2
2   3  3  2
3   1  3  2
4   3  3  2
5   3  3  3
6   3  3  2
7   2  3  3
8   2  3  2
9   3  3  2
```

```

10 2 3 2
11 1 3 2
12 3 3 2
13 1 3 3
14 2 3 2
15 NA NA 2
>

```

```

> # d1の行数
> nr <- nrow(d1)
>

```

```

> # d1の変数名
> cnames1 <- colnames(d1)
>

```

```

> # d1の行数, 2列, 要素の値0の行列を生成
> d2 <- matrix(c(0), nr, 2)
>

```

```

> # d1とd2を結合
> d1 <- data.frame(d1, d2)
> colnames(d1) <- c(cnames1, "s1", "m1")
> d1

```

```

  x1 x2 x3 s1 m1
1 NA 3 2 0 0
2 3 3 2 0 0
3 1 3 2 0 0
4 3 3 2 0 0
5 3 3 3 0 0
6 3 3 2 0 0
7 2 3 3 0 0
8 2 3 2 0 0
9 3 3 2 0 0
10 2 3 2 0 0
11 1 3 2 0 0
12 3 3 2 0 0
13 1 3 3 0 0
14 2 3 2 0 0
15 NA NA 2 0 0
>

```

```

> #for, if
> for (i in 1 : nr){
+   if (is.na(d1$x1[i])!=TRUE){
+     if (d1$x1[i]==1) d1$s1[i] <- 0.5   else
+       if (d1$x1[i]==2) d1$s1[i] <- 1   else
+         d1$s1[i] <- 0
+   } else {
+     d1$s1[i] <- 0
+     d1$m1[i] <- 1
+   }
+ }
> d1

```

```

  x1 x2 x3 s1 m1
1 NA 3 2 0.0 1
2 3 3 2 0.0 0
3 1 3 2 0.5 0
4 3 3 2 0.0 0
5 3 3 3 0.0 0
6 3 3 2 0.0 0
7 2 3 3 1.0 0
8 2 3 2 1.0 0
9 3 3 2 0.0 0
10 2 3 2 1.0 0
11 1 3 2 0.5 0
12 3 3 2 0.0 0
13 1 3 3 0.5 0
14 2 3 2 1.0 0
15 NA NA 2 0.0 1

```

	A	B	C
1	x1	x2	x3
2		3	2
3	3	3	2
4	1	3	2
5	3	3	2
6	3	3	3
7	3	3	2
8	2	3	3
9	2	3	2
10	3	3	2
11	2	3	2
12	1	3	2
13	3	3	2
14	1	3	3
15	2	3	2
16			2
17			

```

# iを1からnrまで変化させる
# x1が欠測でなければ,
# x1が1ならs1を0.5とする
# x1が2ならs1を1とする
# x1がそれ以外なら, s1を0とする

# x1が欠測値なら, s1を0とし,
# さらに, m1を10とする

```

```

>
>
> # ifelse
> ifelse(d1$x3==2, 1, 0) # x3が2なら1, それ以外なら0を表示
[1] 1 1 1 1 0 1 0 1 1 1 1 1 0 1 1
>

> # while
> j <- 1 # はじめにjの値を1と指定
> while(j <= 10){ # jが10以下である間は次を繰り返す
+   print(d1[j,]) # d1の第j行を表示
+   j <- j+1 # jの値を1だけ増加させる
+ }
  x1 x2 x3 s1 m1
1 NA 3 2 0 1
  x1 x2 x3 s1 m1
2 3 3 2 0 0
  x1 x2 x3 s1 m1
3 1 3 2 0.5 0
  x1 x2 x3 s1 m1
4 3 3 2 0 0
  x1 x2 x3 s1 m1
5 3 3 3 0 0
  x1 x2 x3 s1 m1
6 3 3 2 0 0
  x1 x2 x3 s1 m1
7 2 3 3 1 0
  x1 x2 x3 s1 m1
8 2 3 2 1 0
  x1 x2 x3 s1 m1
9 3 3 2 0 0
  x1 x2 x3 s1 m1
10 2 3 2 1 0
>

> # repeat
> k <- 1 # はじめにkの値を1と指定
> repeat{
+   print(d1[k,]) # d1の第k行を表示
+   k <- k+1 # kの値を1だけ増加させる
+   if (k > 8) break # もしkが8を超えたら終了. そうでなければ上を繰り返す
+ }
  x1 x2 x3 s1 m1
1 NA 3 2 0 1
  x1 x2 x3 s1 m1
2 3 3 2 0 0
  x1 x2 x3 s1 m1
3 1 3 2 0.5 0
  x1 x2 x3 s1 m1
4 3 3 2 0 0
  x1 x2 x3 s1 m1
5 3 3 3 0 0
  x1 x2 x3 s1 m1
6 3 3 2 0 0
  x1 x2 x3 s1 m1
7 2 3 3 1 0
  x1 x2 x3 s1 m1
8 2 3 2 1 0
>

```

## コマンドを生成して実行

```
eval(parse(text=paste(コマンド文を生成する指定, sep="")))
```

for文で、iやjの値を変数の指定に用いて、コマンドを生成して実行するのに用いることも多い。

```
> x <- 2
> y <- 4
>
>

> # x,yを数値として扱うか, "文字"として扱うかで paste の結果が異なる
>
> # xの値とyの値をpaste
> paste(x, y, sep="")
[1] "24"
>

> # x+yの値
> paste(x+y, sep="")
[1] "6"
>

> # xの値とyの値を足す式
> paste(x, "+", y, sep="")
[1] "2+4"
>

> # xとyを足す式
> paste("x+y", sep="")
[1] "x+y"
>
>

> # pasteされたものをスクリプトとして実行する
>
> eval(parse(text=paste(x, y, sep="")))
[1] 24
>
> eval(parse(text=paste(x+y, sep="")))
[1] 6
>
> eval(parse(text=paste(x, "+", y, sep="")))
[1] 6
>
> eval(parse(text=paste("x+y", sep="")))
[1] 6
>
>
```

```

> # ベクトルがある場合
> z <- c(1,3)
>
> # xとzの各要素をそれぞれpasteした値のベクトル
> paste(x, z, sep="")
[1] "21" "23"
>
> # xとzの各要素をそれぞれ足した値のベクトル
> paste(x+z, sep="")
[1] "3" "5"
>
> # xとzの要素の値をそれぞれ足す式
> paste(x, "+", z, sep="")
[1] "2+1" "2+3"
>
> # xとzを足す式
> paste("x+z", sep="")
[1] "x+z"
>
>
> # pasteされたものをスクリプトとして実行する
>
> eval(parse(text=paste(x, z, sep="")))
[1] 23
>
> eval(parse(text=paste(x+z, sep="")))
[1] 5
>
> eval(parse(text=paste(x, "+", z, sep="")))
[1] 5
>
> eval(parse(text=paste("x+z", sep="")))
[1] 3 5
>
>
> # zの要素を結合した文字データ
> (zz <- paste(z, collapse=""))
[1] "13"
>
> # xとzの要素をすべてpaste
> paste(x, zz, sep="")
[1] "213"
>
> # xは数値, zは文字なので, 足せない
> paste(x+zz, sep="")
x + zz でエラー: 二項演算子の引数が数値ではありません
>
> # xの値とzの値を足す式
> paste(x, "+", zz, sep="")
[1] "2+13"
>
> # xとzzと足す式
> ppaste("x+zz", sep="")
[1] "x+zz"
>
>
> # pasteされたものをスクリプトとして実行する
>
> eval(parse(text=paste(x, zz, sep="")))
[1] 213
>
> eval(parse(text=paste(x+zz, sep="")))
x + zz でエラー: 二項演算子の引数が数値ではありません
>
> eval(parse(text=paste(x, "+", zz, sep="")))
[1] 15
>
> eval(parse(text=paste("x+zz", sep="")))
x + zz でエラー: 二項演算子の引数が数値ではありません
>

```

```

>
>
> # zの要素を結合した文字データを数値化
> (zz <- as.numeric(paste(z, collapse="")))
[1] 13
>
> # xとzの要素をすべてpaste
> ppaste(x, zz, sep="")
[1] "213"
>
> # xとzzの値を足したもの
> paste(x+zz, sep="")
[1] "15"
>
> # xとzzの値を足す式
> paste(x,"+",zz, sep="")
[1] "2+13"
>
> # xとzzを足す式
> paste("x+zz", sep="")
[1] "x+zz"
>

> # pasteされたものをスクリプトとして実行する
>
> eval(parse(text=paste(x, zz, sep="")))
[1] 213
> eval(parse(text=paste(x+zz, sep="")))
[1] 15
> eval(parse(text=paste(x,"+",zz, sep="")))
[1] 15
> eval(parse(text=paste("x+zz", sep="")))
[1] 15
>
>
>
>

```

## 一括分析・総当たりの分析

分析に含める変数名を書いたベクトルの作成

ベクトル名 &lt;- c(変数名の列)

## 一括分析・総当たりの分析

```
for(i in ベクトル名){
  オブジェクト名 <- eval(parse(text=paste(分析関数や変数の設定, sep="")))
  print(変数, quote = FALSE)
  print(オブジェクト名)
}
```

対応のある変数の総当たりの分析の場合は、for文を二重にして、iとjで回す。

```
> setwd("d:¥¥¥")
> d1 <- read.table("総当たりの分析_データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  id class x1 x2 x3
1  1     a  3  3  2
2  2     a  3  3  2
3  3     a  3  3  2
4  4     a  3  3  2
5  5     b  3  3  3
6  6     a  3  3  2
>
```

	A	B	C	D	E
1	id	class	x1	x2	x3
2	1	a	3	3	2
3	2	a	3	3	2
4	3	a	3	3	2
5	4	a	3	3	2
6	5	b	3	3	3
7	6	a	3	3	2
8	7	a	2	3	3
9	8	a	2	3	2
10	9	a	3	2	2
11	10	b	3	3	3
12	11	b	3	3	2
13	12	b	2	2	2
14	13	a	3	3	2
15	14	a	3	3	2
16	15	a	2	3	2
17	16	a	3	3	2
18	17	a	3	3	2
19	18	a	3	3	3
20	19	a	2	3	2
21	20	a	2	3	2

```
> # 分析に用いる変数名
> v.names <- c("x1", "x2", "x3")

>
> # 各群の記述統計量
> dtmp <- d1[d1$class=="a", v.names]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
   N Mean  SD  x1  x2  x3
x1 43 2.53 0.55 1.00 0.37 0.55
x2 43 2.84 0.43 0.37 1.00 0.20
x3 43 2.16 0.57 0.55 0.20 1.00
>
```

```
> dtmp <- d1[d1$class=="b", v.names]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
   N Mean  SD  x1  x2  x3
x1 43 2.21 0.67 1.00 0.61 0.34
x2 43 2.44 0.70 0.61 1.00 0.40
x3 43 2.19 0.55 0.34 0.40 1.00
```



> # 一括分析 — 対応のない t 検定を指定したすべての変数について行う

```
for(i in v.names){
  tmp <- eval(parse(text=paste("t.test(d1$", i, "~d1$class)", sep="")))
  print(i, quote = FALSE)
  print(tmp)
}
```

[1] x1

Welch Two Sample t-test

```
data: d1$x1 by d1$class
t = 2.4531, df = 80.715, p-value = 0.01632
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.06148781 0.58967498
sample estimates:
mean in group a mean in group b
 2.534884      2.209302
```

[1] x2

Welch Two Sample t-test

```
data: d1$x2 by d1$class
t = 3.1491, df = 69.974, p-value = 0.002409
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.1449566 0.6457410
sample estimates:
mean in group a mean in group b
 2.837209      2.441860
```

[1] x3

Welch Two Sample t-test

```
data: d1$x3 by d1$class
t = -0.1925, df = 83.781, p-value = 0.8479
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.2635701 0.2170585
sample estimates:
mean in group a mean in group b
 2.162791      2.186047
```

> # 総当たりの分析 — 無相関検定を指定した変数においてすべての組合せについて行う

> # 分析に用いる変数名

```
> v.names <- c("x1", "x2", "x3")
```

> # 記述統計量

```
> dtmp <- d1[, v.names]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean  SD  x1  x2  x3
x1 86 2.37 0.63 1.00 0.56 0.41
x2 86 2.64 0.61 0.56 1.00 0.29
x3 86 2.17 0.56 0.41 0.29 1.00
```

```

>
>
> # 無相関検定を総当たりで行う
> p <- length(v.names)
> for(i in 1:(p-1)){
+   for(j in (i+1):p){
+     tmp <- eval(parse(text=paste("cor.test(d1$", v.names[i], ", d1$", v.names[j],
+                                     ", use='paired.complete.obs')", sep="")))
+     tcomp <- paste(v.names[i], "and", v.names[j])
+     print(tcomp, quote = FALSE)
+     print(tmp)
+   }
+ }

```

```
[1] x1 and x2
```

Pearson's product-moment correlation

```

data: d1$x1 and d1$x2
t = 6.2385, df = 84, p-value = 1.713e-08
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.3982996 0.6920578
sample estimates:
      cor
0.5626897

```

```
[1] x1 and x3
```

Pearson's product-moment correlation

```

data: d1$x1 and d1$x3
t = 4.1692, df = 84, p-value = 7.397e-05
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.2216375 0.5754582
sample estimates:
      cor
0.4140683

```

```
[1] x2 and x3
```

Pearson's product-moment correlation

```

data: d1$x2 and d1$x3
t = 2.7785, df = 84, p-value = 0.006735
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.08337366 0.47292836
sample estimates:
      cor
0.290124
>

```

## 関数の作成

```
関数名 <- function(引数, デフォルト値の設定) {
  ...
  return(出力をまとめたもの)
}
```

関数を使用するには、Rを立ち上げた後に、関数部分を一度実行しておく

関数の中で生成した値を関数の外でも使いたい場合  
オブジェクト名 <<- 計算式

## 2群の平均値差の効果量を推定する関数

```
# -----
# Effect Size d

effectD <- function(n1, n2, m1, m2, s1, s2, conf.level=0.95) {
  library(MBESS)
  esdf <- n1+n2-2
  ess <- sqrt((n1*s1*s1 + n2*s2*s2)/esdf)
  esg <- (m1-m2)/ess
  esn <- sqrt(n1*n2/(n1+n2))
  est <- esg*esn
  escl <- conf.limits.nct(t.value=est, df=esdf, conf.level=conf.level)
  eslow <- escl$Lower.Limit/esn
  esup <- escl$Upper.Limit/esn
  op <- round(c(esg, eslow, esup), 2)
  esout <- round(c(n1, n2, m1, m2, s1, s2, op), 2)
  names(esout) <- c("n1", "n2", "m1", "m2", "s1", "s2", "G", "Lower", "Upper")
  return(esout)
  detach("package:MBESS")
}

# -----

> setwd("d:¥¥¥")
> d1 <- read.table("対応のない2群の平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
>
>
> # 各群のデータを抽出
> data1 <- d1[d1$説明=="ビデオ", "不安"]
> data2 <- d1[d1$説明=="看護師", "不安"]

> # 記述統計量
> (m1 <- mean(data1))
[1] 27.03883
> (s1 <- sd(data1))
[1] 4.315777
> (n1 <- length(data1))
[1] 103
> (m2 <- mean(data2))
[1] 25.43617
> (s2 <- sd(data2))
[1] 6.220947
> (n2 <- length(data2))
[1] 94

> # 関数の実行
> effectD(n1=n1, n2=n2, m1=m1, m2=m2, s1=s1, s2=s2)

      n1      n2      m1      m2      s1      s2      G  Lower  Upper
103.00  94.00  27.04  25.44   4.32   6.22   0.30   0.02   0.58
```

## スクリプトの呼び出し

別ファイルに保存してあるスクリプトを呼び出して実行する。

`source("ファイル名")`

ディレクトリが異なる場合はディレクトリから指定する。

自作関数を別ファイルにして保存しておき、`source` を使って、必要な箇所でそのファイル（関数）を呼び出すという使い方ができる。

```
> setwd("i:¥¥Rdocuments¥¥scripts¥¥")
>
> d1 <- read.table("対応のない2群の平均値データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
  番号 説明 不安
1    1 ビデオ 26
2    2 看護師 24
3    3 ビデオ 29
4    4 看護師 23
5    5 ビデオ 25
6    6 看護師 23
>
>
> # 各群のデータを抽出
> data1 <- d1[d1$説明=="ビデオ", "不安"]
> data2 <- d1[d1$説明=="看護師", "不安"]
>
>
> # 記述統計量
> (m1 <- mean(data1))
[1] 27.03883
> (s1 <- sd(data1))
[1] 4.315777
> (n1 <- length(data1))
[1] 103

> (m2 <- mean(data2))
[1] 25.43617
> (s2 <- sd(data2))
[1] 6.220947
> (n2 <- length(data2))
[1] 94

> # 確認
> library(psych)
> describeBy(d1[,c("不安")], d1$説明, mat=TRUE, digits=2)
  item group1 vars   n mean   sd median trimmed  mad min max range  skew kurtosis   se
X11    1 ビデオ    1 103 27.04 4.32    27   27.12 4.45  17  37    20 -0.10   -0.44 0.43
X12    2 看護師    1  94 25.44 6.22    25   25.43 5.93   9  40    31 -0.03   -0.15 0.64

> # スクリプトの呼び出し
> source("300_自作関数.R")

> # 呼び出した関数の実行（効果量の推定）
> effectD(n1=n1, n2=n2, m1=m1, m2=m2, s1=s1, s2=s2)
  n1    n2    m1    m2    s1    s2    G  Lower  Upper
103.00 94.00 27.04 25.44  4.32  6.22  0.30  0.02  0.58
```

## いくつかの図をまとめてPDFファイルに出力

単回帰分析において、説明変数x、基準変数y、誤差e、予測値pとしたときの、次の6個の散布図を3行2列にならべて、PDFファイルに出力する。

x-y, p-y, x-e, p-e, x-p y-e

```
> setwd("d:¥¥")
> d1 <- read.table("回帰分析データ.csv", header=TRUE, sep=",", fileEncoding="shift-jis")
> head(d1)
```

```
  id stress kyoufu support utsu work result
1  1     20    2.2     17    18     0     0
2  2     23    4.8     18    21     1     0
3  3     30    5.8     12    29     1     1
4  4     25    5.2     18    29     0     1
5  5     26    2.0      8    22     1     0
6  6     21    5.0     26    19     1     0
```

1	id	stress	kyoufu	support	utsu	work	result
2	1	20	2.2	17	18	0	0
3	2	23	4.8	18	21	1	0
4	3	30	5.8	12	29	1	1
5	4	25	5.2	18	29	0	1
6	5	26	2	8	22	1	0
7	6	21	5	26	19	1	0
8	7	14	2.2	24	12	0	0
9	8	22	4.4	17	19	1	0
10	9	26	4.2	11	27	0	1
11	10	26	4.2	18	18	1	0
12	11	21	2	27	18	0	0
13	12	24	4.8	19	29	1	1
14	13	24	3.4	23	19	1	0
15	14	23	2.2	10	24	1	0
16	15	23	4.4	24	20	0	0
17	16	35	7.2	12	31	0	1
18	17	25	3.2	17	19	1	0
19	18	33	3.4	14	26	0	1
20	19	30	4.4	20	30	1	1
21	20	19	5.6	18	12	1	0

```
> # 記述統計量
> dtmp <- d1[, c(-1)]
> ntmp <- nrow(dtmp)
> mtmp <- colMeans(dtmp)
> stmp <- apply(dtmp, 2, sd)
> ctmp <- cor(dtmp)
> ktmp <- round(data.frame(ntmp, mtmp, stmp, ctmp), 2)
> colnames(ktmp) <- c("N", "Mean", "SD", colnames(ctmp))
> ktmp
```

```
      N Mean SD stress kyoufu support utsu work result
stress 245 22.94 5.25 1.00 0.40 -0.34 0.62 0.03 0.44
kyoufu 245 4.05 1.17 0.40 1.00 -0.03 0.31 0.10 0.20
support 245 18.42 4.96 -0.34 -0.03 1.00 -0.51 -0.03 -0.39
utsu    245 20.29 6.49 0.62 0.31 -0.51 1.00 0.02 0.76
work    245 0.50 0.50 0.03 0.10 -0.03 0.02 1.00 -0.08
result  245 0.26 0.44 0.44 0.20 -0.39 0.76 -0.08 1.00
```

```
> # 単回帰分析
> reg.1 <- lm(utsu ~ stress, data=d1)
> summary(reg.1)
```

```
Call:
lm(formula = utsu ~ stress, data = d1)
```

```
Residuals:
    Min       1Q   Median       3Q      Max
-11.6277  -3.2181  -0.0374   3.0771  15.6674
```

```
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  2.73611    1.46777   1.864   0.0635 .
stress       0.76506    0.06238  12.264 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 5.114 on 243 degrees of freedom
Multiple R-squared:  0.3823,    Adjusted R-squared:  0.3798
F-statistic: 150.4 on 1 and 243 DF,  p-value: < 2.2e-16
```

```
>
>
> #偏回帰係数の信頼区間
> confint(reg.1)
            2.5 %    97.5 %
(Intercept) -0.1550688 5.6272812
stress       0.6421862 0.8879398
>
>
```

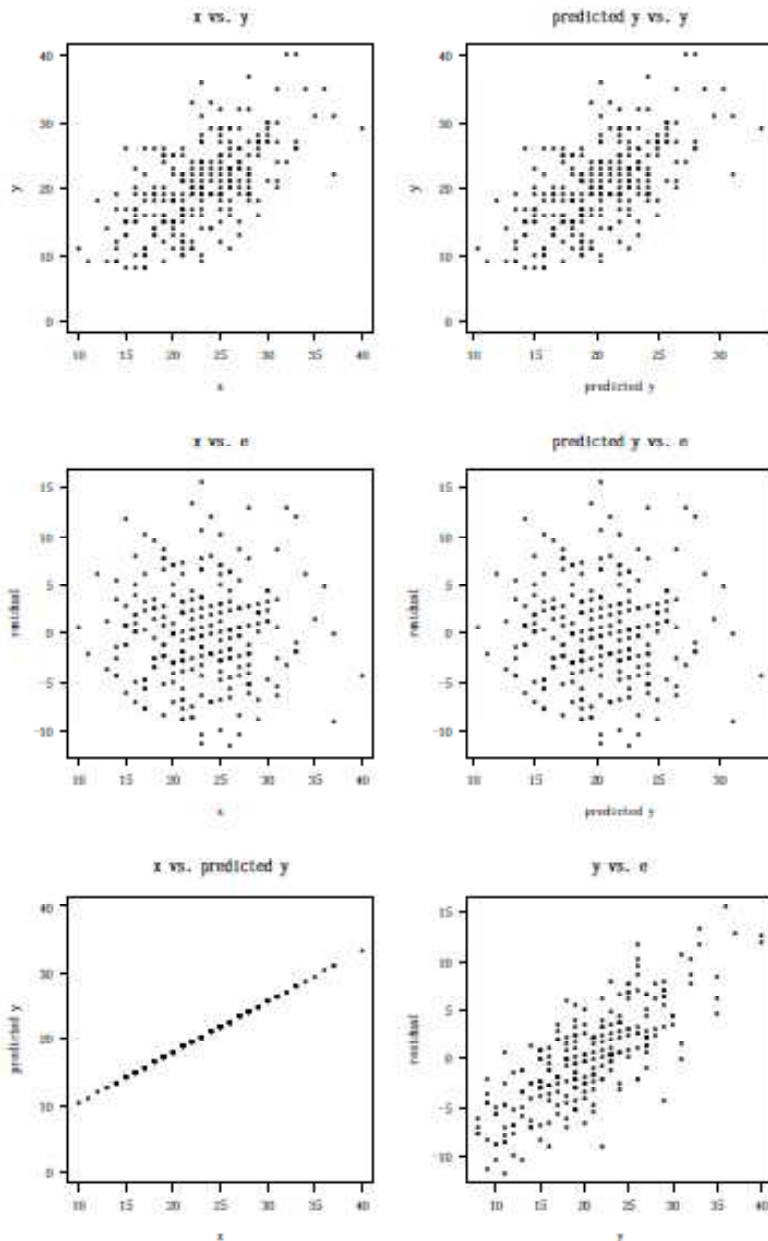
## &gt; #x, y, e の 散布図をPDFに出力

```

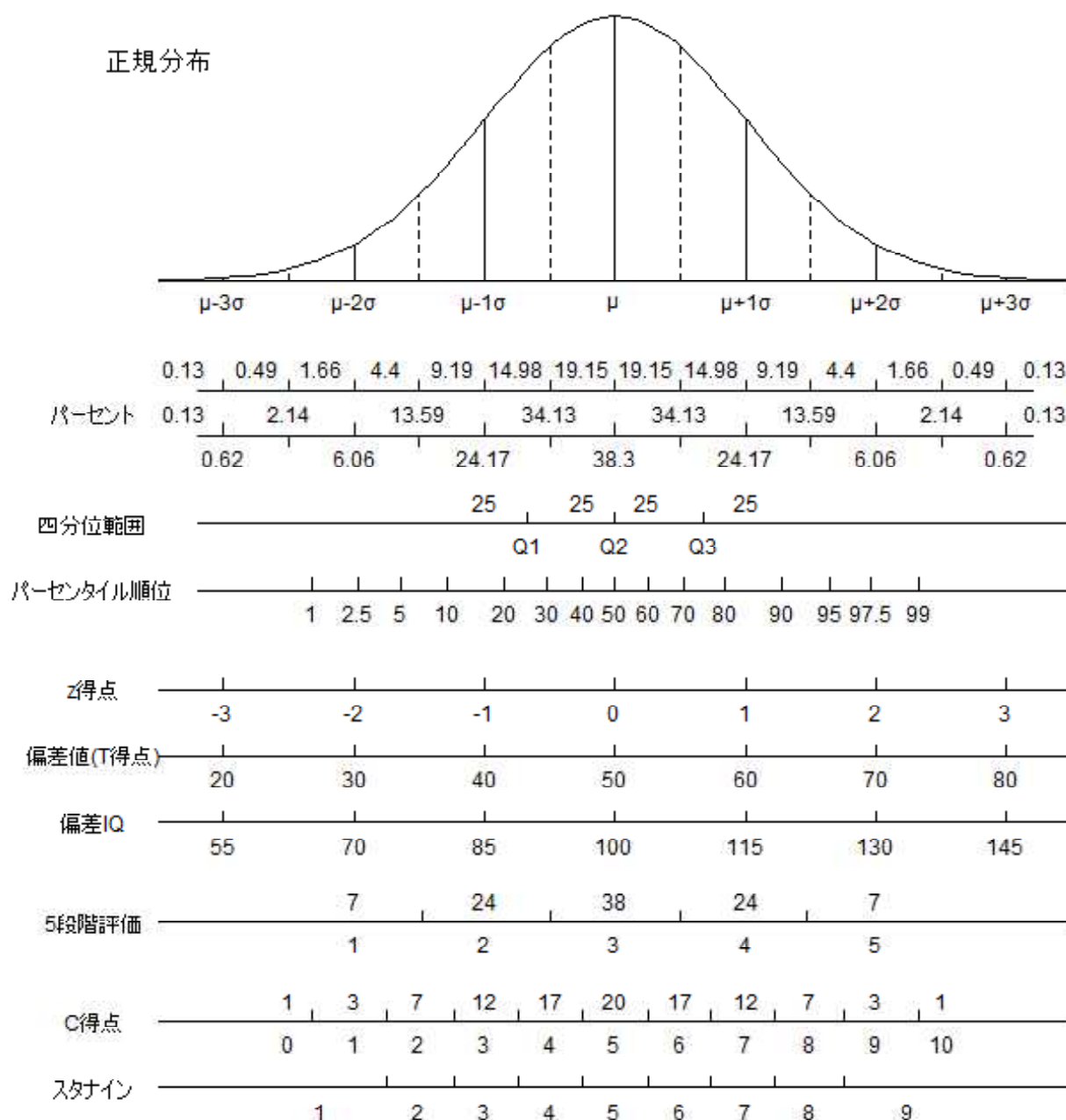
> pdf("残差プロット.pdf", paper="a4", width=7, height=15, family="Japan1")
> layout(matrix(c(1, 2, 3, 4, 5, 6), 3, 2, byrow=T)) # 7×15インチの中に6個の図を3行2列に配置
>
> plot(d1$stress, d1$utsu, xlab="x", ylab="y", ylim=c(0, 40), pch=20, las=1, main="x vs. y")
>
> plot(reg.1$fitted.values, d1$utsu, xlab="predicted y", ylab="y",
+      ylim=c(0, 40), pch=20, las=1, main="predicted y vs. y")
>
> plot(d1$stress, reg.1$residuals, xlab="x", ylab="residual", pch=20, las=1,
+      main="x vs. e")
>
> plot(reg.1$fitted.values, reg.1$residuals, xlab="predicted y",
+      ylab="residual", pch=20, las=1, main="predicted y vs. e")
>
> plot(d1$stress, reg.1$fitted.values, xlab="x", ylab="predicted y",
+      ylim=c(0, 40), pch=20, las=1, main="x vs. predicted y")
>
> plot(d1$utsu, reg.1$residuals, xlab="y", ylab="residual", pch=20, las=1,
+      main="y vs. e")
>
> layout(1)
> dev.off()
null device
1

```

## # 出力されたファイル



## 正規分布と標準得点の対応関係の図を描く



## # 正規分布と標準得点

```
setwd("c:\\home\\Rdocuments\\scripts\\")
par(mar=c(0, 1, 0, 1))
```

```
ycep <- 0.75
dprp <- 0.6
xvals.a <- seq(-3.5, 3.5, length=71)
dvals.a <- dnorm(xvals.a)* dprp + ycep
plot(xvals.a, dvals.a, type="l",
      xlim=c(-4.5, 3.5), ylim=c(0, 1),
      axes=FALSE, ann=FALSE)
segments(-3.5, ycep, 3.5, ycep)

x1 <- c(-3.0, -2.0, -1.0, 0, 1.0, 2.0, 3.0)
for(i in 1:length(x1)){
  y1 <- dnorm(x1[i])*dprp+ycep
  segments(x1[i], ycep, x1[i], y1)
```

```

}
x1 <- c(-2.5, -1.5, -0.5, 0.5, 1.5, 2.5)
for(i in 1:length(x1)){
  y1 <- dnorm(x1[i])*dprp+ycep
  segments(x1[i],ycep,x1[i], y1, lty="dashed")
}

ycepl <- ycep-0.02
text(-3, ycepl, " $\mu - 3\sigma$ ", cex=0.75)
text(-2, ycepl, " $\mu - 2\sigma$ ", cex=0.75)
text(-1, ycepl, " $\mu - 1\sigma$ ", cex=0.75)
text(0, ycepl, " $\mu$ ", cex=0.75)
text(1, ycepl, " $\mu + 1\sigma$ ", cex=0.75)
text(2, ycepl, " $\mu + 2\sigma$ ", cex=0.75)
text(3, ycepl, " $\mu + 3\sigma$ ", cex=0.75)

# percent
ycepl <- ycep-0.06
x1 <- c(-3.3, -2.75, -2.25, -1.75, -1.25, -.75, -.25, .25, .75, 1.25, 1.75, 2.25, 2.75, 3.3)
y1 <- c(0.13, 0.49, 1.66, 4.40, 9.19, 14.98, 19.15, 19.15, 14.98, 9.19, 4.40, 1.66, 0.49, 0.13)
for(i in 1:length(x1)){
  text(x1[i], ycepl, y1[i], cex=0.75)
}
ycepl <- ycepl-0.02
yceplu <- ycepl + 0.01
segments(-3.2,ycepl, 3.2,ycepl)
x1 <- c(-3.0, -2.5, -2.0, -1.5, -1.0, -0.5, 0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0)
for(i in 1:length(x1)){
  segments(x1[i],ycepl,x1[i], yceplu)
}

ycepl <- ycepl-0.02
x1 <- c(-3.3, -2.5, -1.5, -.5, .5, 1.5, 2.5, 3.3)
y1 <- c(0.13, 2.14, 13.59, 34.13, 34.13, 13.59, 2.14, 0.13)
for(i in 1:length(x1)){
  text(x1[i], ycepl, y1[i], cex=0.75)
}

text(-4, ycepl, "パーセント", cex=0.75)
ycepl <- ycepl-0.02
yceplu <- ycepl + 0.01
ycepls <- ycepl - 0.01
segments(-3.2,ycepl, 3.2,ycepl)
x1 <- c(-3.0, -2.0, -1.0, 0, 1.0, 2.0, 3.0)
y1 <- c(-2.5, -1.5, -0.5, 0.5, 1.5, 2.5)
for(i in 1:length(x1)){
  segments(x1[i],ycepl,x1[i], yceplu)
  segments(y1[i],ycepl,y1[i], ycepls)
}

ycepl <- ycepl-0.02
x1 <- c(-3, -2, -1, 0, 1, 2, 3)
y1 <- c(0.62, 6.06, 24.17, 38.30, 24.17, 6.06, 0.62)
for(i in 1:length(x1)){
  text(x1[i], ycepl, y1[i], cex=0.75)
}

# Q1-Q3
ycepl <- ycepl-0.06
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "四分位範囲", cex=0.75)
segments(-3.2,ycepl, 3.5,ycepl)
x1 <- c(0.25, 0.50, 0.75)
y1 <- c("Q1", "Q2", "Q3")
for(i in 1:length(x1)){
  xq <- qnorm(x1[i])
  segments(xq,ycepl, xq, yceplm)
  text(xq, ycepls, y1[i], cex=0.75)
}

```



```

}
z1 <- c(25, 25, 25, 25)
w1 <- c(-1, -0.25, 0.25, 1)
for(i in 1:length(z1)){
  text(w1[i], yceplu, z1[i], cex=0.75)
}

# percintile order
ycepl <- ycepl-0.06
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "パーセンタイル順位", cex=0.75)
segments(-3.2, ycepl, 3.5, ycepl)
x1 <- c(0.01, 0.025, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.975, 0.99)
y1 <- c(1, 2.5, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, 95, 97.5, 99)
for(i in 1:length(x1)){
  xq <- qnorm(x1[i])
  segments(xq, ycepl, xq, yceplm)
  text(xq, ycepls, y1[i], cex=0.75)
}

# z score
ycepl <- ycepl-0.09
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "z得点", cex=0.75)
segments(-3.5, ycepl, 3.5, ycepl)
x1 <- c(-3.0, -2.0, -1.0, 0, 1.0, 2.0, 3.0)
y1 <- c(-3.0, -2.0, -1.0, 0, 1.0, 2.0, 3.0)
for(i in 1:length(x1)){
  segments(x1[i], ycepl, x1[i], yceplm)
  text(x1[i], ycepls, y1[i], cex=0.75)
}
x1 <- c(-2.5, -1.5, -0.5, 0.5, 1.5, 2.5)
y1 <- c(2.15, 13.59, 34.13, 34.13, 13.59, 2.15)
for(i in 1:length(x1)){
  #text(x1[i], yceplu, y1[i], cex=0.75)
}

# T score
ycepl <- ycepl-0.06
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "偏差値(T得点)", cex=0.75)
segments(-3.5, ycepl, 3.5, ycepl)
x1 <- c(-3.0, -2.0, -1.0, 0, 1.0, 2.0, 3.0)
y1 <- c(20, 30, 40, 50, 60, 70, 80)
for(i in 1:length(x1)){
  segments(x1[i], ycepl, x1[i], yceplm)
  text(x1[i], ycepls, y1[i], cex=0.75)
}

# IQ
ycepl <- ycepl-0.06
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "偏差IQ", cex=0.75)
segments(-3.5, ycepl, 3.5, ycepl)
x1 <- c(-3.0, -2.0, -1.0, 0, 1.0, 2.0, 3.0)
y1 <- c(55, 70, 85, 100, 115, 130, 145)
for(i in 1:length(x1)){
  segments(x1[i], ycepl, x1[i], yceplm)
  text(x1[i], ycepls, y1[i], cex=0.75)
}

```

```

# 5 rank
ycepl <- ycepl-0.09
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "5段階評価", cex=0.75)
segments(-3.5, ycepl, 3.5, ycepl)
x1 <- c(0.07, 0.24, 0.38, 0.24)
x2 <- 0
for(i in 1:length(x1)){
  x2 <- x2+x1[i]
  xq <- qnorm(x2)
  segments(xq, ycepl, xq, yceplm)
}
x1 <- c(-2, -1, 0, 1, 2)
y1 <- c(1, 2, 3, 4, 5)
z1 <- c(7, 24, 38, 24, 7)
for(i in 1:length(x1)){
  text(x1[i], yceplu, z1[i], cex=0.75)
  text(x1[i], ycepls, y1[i], cex=0.75)
}

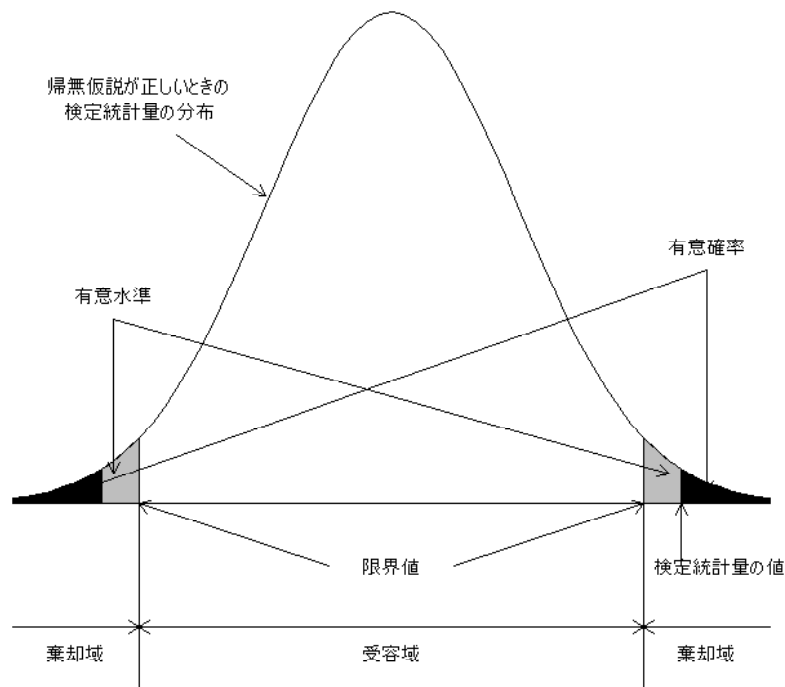
# C score
ycepl <- ycepl-0.09
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "C得点", cex=0.75)
segments(-3.5, ycepl, 3.5, ycepl)
x1 <- c(0.01, 0.03, 0.07, 0.12, 0.17, 0.20, 0.17, 0.12, 0.07, 0.03)
x2 <- 0
for(i in 1:length(x1)){
  x2 <- x2+x1[i]
  xq <- qnorm(x2)
  segments(xq, ycepl, xq, yceplm)
}
x1 <- c(-2.5, -2.0, -1.5, -1.0, -0.5, 0, 0.5, 1.0, 1.5, 2.0, 2.5)
y1 <- c(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
z1 <- c(1, 3, 7, 12, 17, 20, 17, 12, 7, 3, 1)
for(i in 1:length(x1)){
  text(x1[i], yceplu, z1[i], cex=0.75)
  text(x1[i], ycepls, y1[i], cex=0.75)
}

# stanine
ycepl <- ycepl-0.06
yceplm <- ycepl + 0.01
yceplu <- ycepl + 0.02
ycepls <- ycepl - 0.02
text(-4, ycepl, "スタナイン", cex=0.75)
segments(-3.5, ycepl, 3.5, ycepl)
x1 <- c(0.04, 0.07, 0.12, 0.17, 0.20, 0.17, 0.12, 0.07)
x2 <- 0
for(i in 1:length(x1)){
  x2 <- x2+x1[i]
  xq <- qnorm(x2)
  segments(xq, ycepl, xq, yceplm)
}
x1 <- c(-2.25, -1.5, -1.0, -0.5, 0, 0.5, 1.0, 1.5, 2.25)
y1 <- c(1, 2, 3, 4, 5, 6, 7, 8, 9)
for(i in 1:length(x1)){
  text(x1[i], ycepls, y1[i], cex=0.75)
}

text(-3.5, 0.95, "正規分布")

```

## 統計的有意性検定の概念図を描く



```
par(mar=c(0, 1, 0, 1))
```

```
ycep <- 0.2
xvals.a <- seq(-3, 3, length=61)
dvals.a <- dnorm(xvals.a)+ycep
plot(xvals.a, dvals.a, type="l", ylim=c(0, 0.65), axes=FALSE, ann=FALSE)
segments(-3, ycep, 3, ycep)
```

```
xvals.l <- seq(-3, -2, length=21)
dvals.l <- dnorm(xvals.l)+ycep
xvals.u <- seq(2, 3, length=21)
dvals.u <- dnorm(xvals.u)+ycep
polygon(c(xvals.l, rev(xvals.l)), c(rep(ycep, 21), rev(dvals.l)), col="gray")
polygon(c(xvals.u, rev(xvals.u)), c(rep(ycep, 21), rev(dvals.u)), col="gray")
```

```
xvals.pl <- seq(-3, -2.3, length=8)
dvals.pl <- dnorm(xvals.pl)+ycep
xvals.pu <- seq(2.3, 3, length=8)
dvals.pu <- dnorm(xvals.pu)+ycep
polygon(c(xvals.pl, rev(xvals.pl)), c(rep(ycep, 8), rev(dvals.pl)), col="black")
polygon(c(xvals.pu, rev(xvals.pu)), c(rep(ycep, 8), rev(dvals.pu)), col="black")
```

```
segments(-2, ycep, -2, 0)
segments(2, ycep, 2, 0)
arrows(-2, ycep, -0.5, ycep-0.05, code=1, length=0.1)
arrows(2, ycep, 0.5, ycep-0.05, code=1, length=0.1)
text(0, ycep-0.05, "限界値")
arrows(2.3, ycep, 2.3, ycep-0.05, code=1, length=0.1)
text(2.3, ycep-0.05, "検定統計量の値")
```

```
arrows(-2.2, ycep+0.025, -2.2, ycep+0.15, code=1, length=0.1)
arrows(2.2, ycep+0.025, 2.2, ycep+0.15, code=1, length=0.1)
text(-2.2, ycep+0.17, "有意水準")
arrows(-2.5, ycep+0.01, 2.5, ycep+0.19, code=1, length=0.1)
arrows(2.5, ycep+0.01, 2.5, ycep+0.19, code=1, length=0.1)
```

```
text(2.5, ycep+0.21, "有意確率")

arrows(-1, ycep+0.25, -1.7, ycep+0.3, code=1, length=0.1)
text(-2, ycep+0.34, "帰無仮説が正しいときの")
text(-2, ycep+0.32, "検定統計量の分布")

arrows(-2, ycep-0.1, 2, ycep-0.1, code=3, length=0.1)
arrows(-2, ycep-0.1, -3, ycep-0.1, code=1, length=0.1)
arrows(2, ycep-0.1, 3, ycep-0.1, code=1, length=0.1)
text(0, ycep-0.12, "受容域")
text(-2.5, ycep-0.12, "棄却域")
text(2.5, ycep-0.12, "棄却域")
```