エラーが発生したときの対処

スクリプトを実行するとエラーが出てしまうときは，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）ということもしばしば起こります。このような状態について，よくある誤りと対処法を以下に例示します。

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いおいて，使わないで下さい。

・Apple 社製のパソコン（Mac）を使っていませんか？

本スクリプト集は 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，スクリプト集で「**MacOS でＲを使用するときの注意点**」のページを探して，そこの説明を読んでください。

・incomplete final line found by readTableHeader というメッセージが出ていませんか？

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もこのメッセージが出る場合は，テキストエディットやメモ帳などを起動して，そこから対象の CSV ファイルを開き，最終行の末尾に改行記号を入れて保存してください。

・エクセルでデータを保存するとき「CSV UTF-8（コンマ区切り）」で保存していませんか？

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必要になります。文字コードには，Shift JIS や UTF-8 など，いくつかの種類があります。

「CSV UTF-8（コンマ区切り）」でデータを保存した場合は，BOM（バイトオーダーマーク）というものが自動的に付加された UTF-8 形式で，データが保存されます。

この形式のデータを R で読み込む場合は，次のように fileEncoding="UTF-8-BOM" というオプションを追加する必要があります。

d1 <- read.table("dta1.csv", header=TRUE, sep=",", fileEncoding="UTF-8-BOM")

しかし，少なくとも現在は，BOM はつけないほうが良いとされているので，この形式でデータを保存することはお勧めしません。スクリプト集にあるように「CSV（コンマ区切り）」で保存するようにしてください。

・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，新しく分析を始めたいのに，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がよくあるケースです。

例）

setwd("D:¥¥Report¥¥") を選択すべきところを etwd("D:¥¥Report¥¥")　# 先頭の s 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。

ダブルクオーテーションで囲うのも忘れないでください。

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=",")
```

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。ファイル名の「.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。

エクスプローラで拡張子を表示するには，エクスプローラ画面の上のほうにある「表示」をクリックして，「ファイル名拡張子」に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。

拡張子を表示したとき，ファイル名が「data1.csv.csv」のようになっていることがあります。これは，データを csv 形式で保存する際，ファイル名を「data1.csv」と書いたうえに，拡張子が自動で追加されたためです。この場合は，エクスプローラ上でファイル名を右クリックして「名前の変更」を選択し，
「data1.csv」のように，拡張子が 1 つになるようにファイル名を修正して下さい。

・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。

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半角スペース，全角スペースがあるとき，R では「.」（ピリオド）に変換されます。あまり好ましくないので，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大変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　ｘ１　　# 半角x1を全角ｘ１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 xl　　# 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 % と間違えている
d1[d1$sex=="f",] を d1[d1$sex="f",]　　# == を = と間違えている
```

<- を < または <=　# 代入記号「<-」を，大小関係を表す「<」や「<=」と間違えている

・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ら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ませんか？

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，途中で改行せす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 in1: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いても，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。

・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）head(d1)

nrow(d1)

を

head(d1) nrow(d1)　　(head(d1); nrow(d1) なら OK)

・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「;」で区切りるようにします。

例）if (x==5){y <- 1} else y <- 0 を

・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？

数式をの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。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と解決することが多いです。

・R Studio を使っていませんか？

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角にできます。

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。

・コンピュータのシステムアップデートがあって，コンピュータの再起動待ち状態など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。システムの更新版のチェックを行って，システムを最新の状態にしておくのが良いです。

・全角文字を使っていませんか？

全角文字に対応していない関数を使う場合などは，半角カナだけでなく，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。

この場合は，データ，変数名，フォルダ名，ファイル名等をすべて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ませんか？

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。

・再起動待ちでもなく，原因は分からないがとにかく 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 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，「ヘルプの使い方」の項を参照してください。

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。

・R を再インストールするのも有効です。

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。再インストールする場合，ミラーサイトは 0-cloud を選択するのが無難のようです。