R による統計解析入門

小波秀雄

京都女子大学現代社会学部　名誉教授

May 31, 2016

R 事始め

R の操作について

- R はプログラミング言語
- 他のプログラミング言語みたいに何でもやるわけではない
- 決まった手順を守ってコマンドを打ち込んでいく感覚
- R Studio だと GUI のメニューで操作できるが，大して楽にはならない
- 守備範囲は統計の全分野にわたっていて，ライブラリ化されている

- R Console のアイコン

- R Studio のアイコン

このテキストの PDF ファイル，および関連のファイルは，下の URL からアクセスできます。

http://ruby.kyoto-wu.ac.jp/konami/Text/R

使い方は基本２通り

対話的な使い方 コンソール（端末）にコマンドを打ち込んで逐一操作
短いコマンドを試しに実行したり，ヘルプを参照するには便利

バッチ処理方式 ソースプログラム（スクリプト）を書いて，R で処理
修正や使いまわしが簡単，保存しておけるメリット大

どちらというのではなく，両方を併用するのが効率的なやり方

ここからしばらくは，コンソールを使った対話的なやり方で説明します。

R Console について Windows, Mac どちらも統合環境として GUI 端末に加えて簡単なエディタも備わっているので，小規模な処理には使いやすい。Unix 用は端末機能のみで，エディタはユーザーの好みでという発想。

R Studio について コンソール画面やフォルダブラウザなどのパネルを配して GUI 的な操作を狙ったサードパーティの製品だが，現状で言語サポートが不十分なので推奨はできない。

コンソールを開く

開始画面（終了の仕方に注意！）

```
R version 3.2.3 (2015-12-10) -- "Wooden Christmas-Tree"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-pc-linux-gnu (64-bit)
R は、自由なソフトウェアであり、「完全に無保証」です。
一定の条件に従えば、自由にこれを再配布することができます。
配布条件の詳細に関しては、'license()' あるいは 'licence()' と入力してください。

R は多くの貢献者による共同プロジェクトです。
詳しくは 'contributors()' と入力してください。
また、R や R のパッケージを出版物で引用する際の形式については
'citation()' と入力してください。

'demo()' と入力すればデモをみることができます。
'help()' とすればオンラインヘルプが出ます。
'help.start()' で HTML ブラウザによるヘルプがみられます。
'q()' と入力すれば R を終了します。

>
```

デモを見る

`demo()` と入力

こんな画面が開く

```
Demos in package ‘base’:

error.catching          More examples on catching and handling errors
is.things               Explore some properties of R objects and
                        is.FOO() functions. Not for newbies!
recursion               Using recursion for adaptive integration
scoping                 An illustration of lexical scoping.

Demos in package ‘grDevices’:

colors                  A show of R’s predefined colors()
hclColors               Exploration of hcl() space

(stdin):
```

↑↓キーでスクロール，'q' で入力画面に抜ける。

ライブラリとそれごとの機能のデモの一覧

base	error.catching, is.things, recursion, scoping
grDevices	colors, hclColors
graphics	Hershey, Japanese, graphics, image, persp, plotmath
stats	glm.vr, lm.glm, nlm, smooth

次のように入力して，デモを実行させることができる。これや colors, graphics, persp **などは，別画面でグラフィックス画面が開き，コンソールにスクリプトが表示される。モノによっては意味が分かりにくいものもある。**

```
> demo(recursion)
```

R の基本操作

代入（付値）と算術演算（# 以降はコメント）

```
> a <- 50    # 代入は = ではないことに注意
> b <- 30    # = も実は機能するが，他との関係で推奨できない
> c <- a + b
< c          # 変数名だけ入力
[1] 80       # 値が評価されて出力
```

使える変数名

abc, name, x, y
sum1(英字で始まり数字が含まれる形)
mean_total(アンダースコアも語頭以外は可)
Sum(大文字も可),
Member.female(ピリオドも可。他の言語では許容されないことが多い)

次は予約語なので使用不可

if else repeat while function for in next break TRUE FALSE NULL NA Inf NaN

変数名について

data, sum, summary, mean, median, max, mean, sqrt, sin, cos **のような統計用語，数学関数名はすでに関数として存在するので，使わないほうがよい。使うことはできる。**

変数名の工夫

大文字を使う Data, DATA, DT
ピリオドを使って連結 Data.sum, DATA.mean
ただし D **は** 1 **文字で微分演算子なので注意！** C, F, I, T **も意味あり。**

変数名がすでに関数になっているか調べる

```
> sum
function (..., na.rm = FALSE)  .Primitive("sum")
> Sum
 エラー:  オブジェクト 'Sum' がありません
```

sum **は関数名に使われているが，**Sum **は未使用。大文字を使うのは安全なやり方。**

ベクター（配列）

ベクターを c 関数で作る

```
> X <- c(10.2,20.5,-2.9,13.8,9.1,23.1)
> X
[1] 10.2 20.5 -2.9 13.8  9.1 23.1
> X[4]
[1] 13.8  # 1 からカウントされていることに注意
> X[2:4]
[1] 20.5 -2.9 13.8
> X[-3]
[1] 10.2 20.5 13.8  9.1 23.1  # 3 番目を除外したベクターが返る
> order(X)
[1] 3 5 1 4 2 6  # 順位に置き換えたベクターが返る
> sort(X)
[1] -2.9  9.1 10.2 13.8 20.5 23.1
```

簡単な集計

sum, mean, median, min, max, length 関数を上記の X に適用してみよう。

いろいろな統計量

```
> var(X)              # 標本不偏分散
[1] 86.364
> sd(X)               # 標本不偏標準偏差
[1] 9.293223
> quantile(X)         # 四分位数
    0%    25%    50%    75%   100%
-2.900  9.375 12.000 18.825 23.100
> quantile(X,0.25) # 第１四分位数
  25%
9.375
> quantile(X,0.75) # 第３四分位数
   75%
18.825
> summary(X)          # ５数要約（＋平均）
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 -2.900   9.375  12.000  12.300  18.820  23.100
```

統計量の意味

標本不偏分散，標準偏差

$$\sigma^2 = \frac{1}{n-1}\sum_{1}^{n}(x_i - \mu)^2,\ \sigma = \sqrt{\sigma^2}$$

これらは，母集団から無作為抽出したサイズ n の標本のデータ $x_1, x_2, \cdots, x_n$ から母集団の分散，標準偏差を求めるための推定式である。
単に n 個のデータの分散，標準偏差は次の通り。

$$\sigma^2 = \frac{1}{n}\sum_{1}^{n}(x_i - \mu)^2,\ \sigma = \sqrt{\sigma^2}$$

四分位数，五数要約

ある値以下のデータ数が全データ数の 25% となる値を第一四分位数，75% のところを第三四分位数と呼ぶ。メジアン（中央値）は第二四分位数である。
これらに最小値と最大値を付け加えた 5 つの数をまとめて五数要約という。

マトリクス（行列）

複数のデータ列を入力

```
> X <- c(10.2,20.5,-2.9,13.8,9.1,23.1)
> Y <- c(6.8,10.1,5.4,9.1,9.1,1.1)
> Z <- c(1.32,3.24,0.98,0.55,1.67,2.3)
```

cbind 関数で列 (column) として結合 (bind)

```
> Mat <- cbind(X,Y,Z)
> Mat
        X    Y    Z
[1,] 10.2  6.8 1.32
[2,] 20.5 10.1 3.24
[3,] -2.9  5.4 0.98
[4,] 13.8  9.1 0.55
[5,]  9.1  9.1 1.67
[6,] 23.1  1.1 2.30
```

rbind 関数で列 (row) として結合もできる

```
< rbind(X,Y,Z)
   [,1]  [,2]  [,3]  [,4] [,5] [,6]
X 10.20 20.50 -2.90 13.80 9.10 23.1
Y  6.80 10.10  5.40  9.10 9.10  1.1
Z  1.32  3.24  0.98  0.55 1.67  2.3
```

通常のデータ操作では，多数のデータを含む列を縦に並べたものを扱うことが多いので，cbind **のほうがよく用いられる。**

マトリクスの次元属性を知る

```
> dim(Mat)
[1] 6 3
> dim(rbind(X,Y,Z))
[1] 3 6
> dim(Mat)[1]  # dim() の出力はベクターとして振る舞う
[1] 6
> dim(Mat)[2]
[1] 3
```

マトリクスの要素を指定する

```
> Mat[2,3]
   Z
3.24
> Mat[2,]
    X     Y     Z
20.50 10.10  3.24
> Mat[,3]
[1] 1.32 3.24 0.98 0.55 1.67 2.30
> Mat[,]
        X    Y    Z
[1,] 10.2  6.8 1.32
[2,] 20.5 10.1 3.24
[3,] -2.9  5.4 0.98
[4,] 13.8  9.1 0.55
[5,]  9.1  9.1 1.67
[6,] 23.1  1.1 2.30
```

Mat[,], Mat[2,], Mat[,3] **のように添字指定がなければ全データを代表する**

欠損値の扱い

欠損値は NA (not available) で埋める

```
> V <- c(10,20,30,NA,50,60)
> sum(V)
[1] NA
> mean(V)
[1] NA
> var(V)
[1] NA
> sum(V,na.rm=TRUE)
[1] 170
> mean(V,na.rm=TRUE)
[1] 34
> var(V,na.rm=TRUE)
[1] 430
```

データに欠損値が含まれると，sum, mean, var **は** NA **を返すが，オプション引数として** na.rm = TRUE **または** na.rm = T **を与えると，欠損値を取り除いたデータで計算した値を返す。**

list 関数で混在するデータをまとめる

データごとに名前を付けて list に渡す

```
> N <- c("a","b","c","d")
> List <- list(Mat=Mat,val=V,name=N)
> List
$Mat
         X    Y    Z
[1,] 10.2  6.8 1.32
[2,] 20.5 10.1 3.24
[3,] -2.9  5.4 0.98
[4,] 13.8  9.1 0.55
[5,]  9.1  9.1 1.67
[6,] 23.1  1.1 2.30

$val
[1] 10 20 30 NA 50 60

$name
[1] "a" "b" "c" "d"
```

$ を使ってリストに含まれる情報にアクセス

```
> names(List)    # List に含まれる名前を一覧
[1] "Mat"  "val"  "name"

> List$Mat
        X    Y    Z
[1,] 10.2  6.8 1.32
[2,] 20.5 10.1 3.24
[3,] -2.9  5.4 0.98
[4,] 13.8  9.1 0.55
[5,]  9.1  9.1 1.67
[6,] 23.1  1.1 2.30
> List$val
[1] 10 20 30 NA 50 60
> List$name
[1] "a" "b" "c" "d"
```

R の統計処理の結果はリストとして出力されることが多く，この形が使われる。

ファイルデータを扱う

利用できるファイルの種類

- タブ・スペースでデータが区切られたテキストファイル
- カンマでデータが区切られたテキストファイル (CSV ファイル)
- Excel のデータ（拡張子 .xls, .xlsx）

Excel のデータをテキストファイルに

ファイルメニュー→名前を付けて保存→この画面で「すべての形式」をクリックして「テキスト CSV 形式」を選ぶ。

注意

- xls, xlsx ファイルの読み込みは特別なライブラリを用いており処理系に依存するので，テキストファイルに変換して使うことを推奨する。

Windows での実行について

Windows で R を実行する際の留意点

- ルートディレクトリはドライブレター (C:, D:など)
- ディレクトリの区切りはバックスラッシュ '\' だが，環境によって '¥' が表示される。
- R ではディレクトリの区切りをスラッシュ '/' としている（Unix の仕様)。
- Windows の文字コードは CP932 (Shit-JIS) であり，Mac, Linux は UTF-8
- Windows 版 Rscript にはパスが通っていないので，コマンドプロンプトからスクリプトを端末（コマンドプロンプト）で実行するにはパスの設定が必要
- Windows ユーザーはホームディレクトリの概念が希薄なので，作業ディレクトリを意識することが必要

Windows に R をインストールしたら最初にやること

- エクスプローラを使って C:\R とか C:\home\R というフォルダを作成する
- （可能ならここまで）コントロールパネル→システム→詳細設定→環境変数の設定と進んで，システムの Path に次の文字列を追加する (3.3.0 は最新版のバージョン番号なので，そうでないケースでは適宜読み替える)

  C:\Program Files\R\R-3.3.0\bin

データを読み込む

元のテキストデータ（ファイル名は BMI.txt）

```
Name   Sex Height   Weight
Yuri   F    155.6    54.3
Miwa   F    164.2    63.2
Saki   F    158.3    52.3
Taiki   M    171.4    84.4
Tarou M    191.5    76.4
Kei    M    178.5    75.3
```

データフレームに読み込んで大略をつかむ

```
> setwd("C:/R/Mydata")    # データファイルのあるパスを指定
> BMI <- read.table("BMIdata.txt",header=TRUE) # データフレームに読み込み。ヘッダ１行あり
> class(BMI)     # BMI が属しているデータ構造は何か？
[1] "data.frame"
> names(BMI)     # 個別データごとの名前を表示
[1] "Name"    "Sex"     "Height" "Weight"
```

```
> BMI     # データを表示させる（小さいデータの時のみ意味あり）
   Name Sex Height Weight
1  Yuri   F  155.6   54.3
2  Miwa   F  164.2   63.2
3  Saki   F  158.3   52.3
4 Taiki   M  171.4   84.4
5 Tarou   M  191.5   76.4
6   Kei   M  178.5   75.3
> str(BMI)       # BMI の構造を表示させる
'data.frame': 6 obs. of  4 variables:
 $ Name  : Factor w/ 6 levels "Kei","Miwa","Saki",..: 6 2 3 4 5 1
 $ Sex   : Factor w/ 2 levels "F","M": 1 1 1 2 2 2
 $ Height: num  156 164 158 171 192 ...
 $ Weight: num  54.3 63.2 52.3 84.4 76.4 75.3
```

データから情報を得る

```
> summary(BMI$Height)    # Height のまとめ
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  155.6   159.8   167.8   169.9   176.7   191.5
> summary(BMI$Sex)       # カテゴリカル変数（ファクター）のまとめは頻度を与える
F M
3 3
```

条件を付けてデータを絞る

```
> BMI_F <- BMI[BMI$Sex=="F",]    # 女性のみ，列はすべて
> BMI_M <- BMI[BMI$Sex=="M",]    # 男性のみ，列はすべて
> max(BMI_F$Height)              # 女性の身長の最大値
[1] 164.2
> mean_H <- mean(BMI$Height)          # 平均身長
> Higher <- BMI[BMI$Height>mean_H,]   # 平均よりも身長の高い人
> Higher
   Name Sex Height Weight
4 Taiki   M  171.4   84.4
5 Tarou   M  191.5   76.4
6   Kei   M  178.5   75.3
```

練習問題

下のように試験の成績のデータを記録した CSV ファイル Score.csv がある。試験を受けなかった場合の点数は NA となっている。このファイルを読み込んで，以下の情報を取り出しなさい。なお，このファイルは以下の URL からアクセスできるようになっている。

`http://ruby/kyoto-wu.ac.jp/konami/Text/R/`

1. **男子，女子の人数**
2. **男女それぞれについて，試験をすべて受けた人数**
3. **男子，女子，全体それぞれの平均点，最高点，最低点**

氏名，よみ，性別，英語，数学，国語
青木紗也加，あおきさやか,F,69,80,58
浅田菜摘，あさだなつみ,F,92,60,78
朝日友紀，あさひゆうき,M,59,79,63
足立敦志，あだちあつし,M,NA,NA,89
（以下続く）