



フリーソフト「R」入門

2010年3月26日

日本行動計量学会 第13回春の合宿セミナー A1コース

中山 厚穂 (長崎大学)

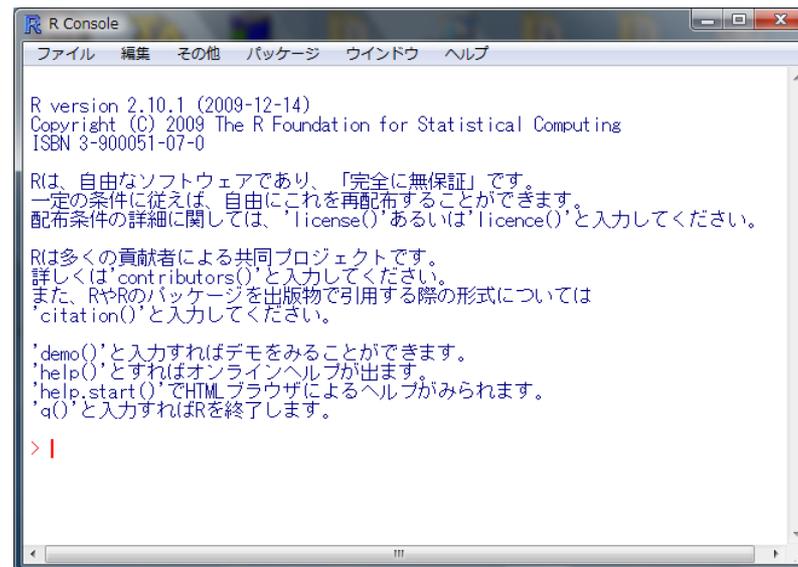
○横山 暁 (慶應義塾大学)



◦ はじめてのR

Rのインストールと起動

- RはCRANからDLできる
 - <http://cran.r-project.org/>
 - Linux, MacOS X, Windows版などがある
- 起動画面 (Windows版)



```
R Console
ファイル 編集 その他 パッケージ ウィンドウ ヘルプ

R version 2.10.1 (2009-12-14)
Copyright (C) 2009 The R Foundation for Statistical Computing
ISBN 3-900051-07-0

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。

> |
```

Rの基本操作 (1)

➤ 簡単な演算

➤ 加減乗除

- +
- -
- *
- /
- %/% : 整数商
- %% : 剰余

➤ 指数・対数, 三角関数

- x^y , **
- sqrt(x)
- log(x, a), log, log10, log2,
- exp(1)
- pi
- sin(x)
- cos(x)
- tan(x)
- factorial() : 階乗

Rの基本操作 (2)

- 四捨五入等
 - round(x) : 四捨五入
 - round(x, 桁)
 - floor : 小数切捨
 - ceiling : 小数切上
 - trunc : 整数部分

Rの基本操作 (3)

➤ 値の代入

➤ `<-`, `->`

➤ `=`

➤ `<<-`, `->>` : 永続代入

➤ 値の比較

➤ `>`, `>=`

➤ `<`, `<=`

➤ `==`, `!=`

➤ 論理

➤ `&` : 論理積

➤ `|` : 論理和

➤ `&&` : かつ

➤ `||` : または

ベクトル・行列・配列

- ベクトル

- `c(data)`

- 行列

- `matrix(data, 行数, 列数, byrow=F, dimnames)`

- `matrix(data, c(行数, 列数), byrow=F, dimnames)`

- 配列

- `array(data, dim=配列の大きさ, dimnames)`

- リストというものもあるがここでは省略

関数

▶ ベクトルに用いる関数

- ▶ length(x)
- ▶ sum(x)
- ▶ mean(x)
- ▶ median(x)
- ▶ quantile(x)
- ▶ var(x)
- ▶ cor(x)
- ▶ max(x)
- ▶ min(x)
- ▶ range(x)
- ▶ prod(x)
- ▶ sort(x, decreasing=F)

▶ 行列に用いる関数

- ▶ nrow(x)
- ▶ ncol(x)
- ▶ rowSums(x)
 - ▶ apply(x, 1, sum)
- ▶ colSums(x)
 - ▶ apply(x, 2, sum)
- ▶ rowMeans(x)
 - ▶ apply(x, 1, mean)
- ▶ colMeans(x)
 - ▶ apply(x, 2, mean)

ベクトル

- ▶ ベクトル同士の計算
- ▶ ベクトルとスカラーの計算
- ▶ ベクトルの統計量の計算

行列

➤ 行列の作成

➤ `matrix(1:6, nrow=2, ncol=3)`

```
      [,1] [,2] [,3]  
[1,]  1   3   5  
[2,]  2   4   6
```

- 1:6は、1から6まで1ずつ増加（もしくは減少）させたベクトル
- `nrow=2`もしくは`ncol=3`が無くても同様
- 両方指定した際、データの要素数が合わないとエラー
- 片方しか指定しないと、データの要素数に合わせて行列の大きさが決まる

➤ `matrix(1:6, nrow=2, ncol=3, byrow=T)`

```
      [,1] [,2] [,3]  
[1,]  1   2   3  
[2,]  4   5   6
```

- `byrow=T`を指定すると、行方向優先でデータが入る
- 標準では`byrow=F`となっている

行列

- ▶ ゼロ行列

- ▶ `matrix(0, nrow=2, ncol=3)`で作成できる

```
      [,1] [,2] [,3]  
[1,]  0   0   0  
[2,]  0   0   0
```

- ▶ ちなみに

- ▶ `matrix(1, nrow=2, ncol=3)`で要素が全て1の行列

- ▶ 行列の転置

- ▶ `t(x)`

行列

▶ 行列要素の抽出

▶ `x<-matrix(1:6, nrow=2, ncol=3)`

▶ `x[i, j]`で(i, j)成分を抽出できる

▶ `x[1,2]`

▶ `[1] 3`

▶ `x[,2]`

▶ `[1] 3 4`

▶ `x[2,]`

▶ `[1] 2 4 6`

▶ `x[,2, drop=F]` : 2列目を行列の形で抽出

▶ `x[c(1,2), c(1,3)]` : 1,2行目と1,3列目を抽出

▶ `x[,-c(1,2)]` : 1,2列を除外した行列を抽出

行列

➤ 行列の結合

➤ rbind : 行ベクトル同士を結合

➤ rbind(c(1,2,3), c(4,5,6))

➤ cbind : 列ベクトル同士を結合

➤ cbind(c(1,2,3), c(4,5,6))

➤ もちろん3つ以上の行・列ベクトルの結合も可

➤ 行列の大きさの確認

➤ dim(x) : 行列の大きさ

➤ nrow(x) : 行数

➤ ncol(x) : 列数

行列

▶ 行列の計算

- ▶ `x<-matrix(0:3, 2, 2)`
- ▶ `y<-matrix(1:4, 2, 2)`

▶ 和・差・積・商（要素同士の積・商）

- ▶ `+, -, *, /`
- ▶ （行列としての）積
 - ▶ `%*%`
 - ▶ 外積(`%o%`), クロネッカー積(`%x%`)も計算できる
 - ▶ ベクトル2つを作成しておいてクロス積(`crossprod(a, b)`)の計算もできる (`t(a)%*%b`の計算)
 - ▶ `crossprod(a)`ならa自身のクロス積

行列

- 行列の計算
 - `nrow(x)`
 - `ncol(x)`
 - `rowSums(x)`
 - `apply(x, 1, sum)`
 - `colSums(x)`
 - `apply(x, 2, sum)`
 - `rowMeans(x)`
 - `apply(x, 1, mean)`
 - `colMeans(x)`
 - `apply(x, 2, mean)`

行列

- 単位行列・対角行列
 - `diag(2)` : 2x2の単位行列
 - `diag(1, ncol=2, nrow=2)`でも可
 - `diag(1:3)`なら・・・？
- 上三角・下三角行列
 - `x<-matrix(1:9, 3)`
 - `x[upper.tri(x)]<-0`
 - `x[upper.tri(x, diag=T)]<-0`
 - `upper`を`lower`で下三角

行列

- 逆行列

- `solve(x)`

- `solve(A, b)`は $Ax=b$ の連立方程式を解くことに相当

- `b`が単位行列なら省略することで`A`の逆行列が求まる

- 一般逆行列 (ムーア・ペンローズ)

- `library(MASS)`

- MASSパッケージが必要

- `ginv(x)`

行列

- 固有値・固有ベクトル

```
x<-matrix(c(0,2,-1,2,0,0,-1,0,0), 3,3)
```

```
> eigen(x)
```

```
$values
```

```
[1] 2.236068 0.000000 -2.236068
```

```
$vectors
```

```
      [,1] [,2] [,3]  
[1,] -0.7071068 0.0000000 0.7071068  
[2,] -0.6324555 0.4472136 -0.6324555  
[3,] 0.3162278 0.8944272 0.3162278
```

- z<-eigen(x)

- z\$values

- z\$vectors

行列

- 分解
 - QR分解 : $\text{qr}(x)$
 - 特異値分解 : $\text{svd}(x)$
 - $X=U*%D*%t(V)$
 - など

行列

- ▶ 行・列に名前を付ける
 - ▶ names属性を与える
 - ▶ `x<-matrix(1:6, nrow=2, ncol=3)`
 - ▶ `rownames(x)<-c("ue", "shita")`
 - ▶ `colnames(x)<-c("hidari","mannaka","migi")`
- ▶ 名前の初期化
 - ▶ `rownames(x)<-NULL`
 - ▶ `colnames(x)<-NULL`
- ▶ 行列だけでなくベクトルにも要素にラベルを付けることができる

配列

➤ 配列の作成

➤ `array(data, dim=配列の大きさ, dimnames)`

➤ `array(1:8, dim=c(1,4,2))`

, , 1

```
      [,1] [,2] [,3] [,4]  
[1,]    1    2    3    4
```

, , 2

```
      [,1] [,2] [,3] [,4]  
[1,]    5    6    7    8
```

- `data`の要素数は、指定した配列に必要な要素数と異なっていても良い
 - 少なければ、繰り返し代入される
 - 多ければ、無視される

グラフ

- `plot(x軸のデータ, (y軸のデータ), オプション)`
 - `x<-y<-c(1:10)`
 - `plot(x, y)`

 - `plot(sin, -pi, pi)`
- `hist()` : ヒストグラム
- `boxplot()` : 箱ひげ図

Plot

- typeオプション
 - `plot(x, y, typeオプション)`
 - `type="p"` : 点 (散布図)
 - `l` : 折れ線
 - `b` : 点と線
 - `c` : `b`の線が無いバージョン
 - `n` : 軸だけ描いてプロットしない
 - など
- 図の消去
 - `frame()` or `plot.new()`

作業ディレクトリ（フォルダ）

- ▶ 作業ディレクトリの確認
 - ▶ `getwd()`
- ▶ 作業ディレクトリの変更
 - ▶ `setwd("C:/Users/user/Documents")`
 - ▶ バックスラッシュ(/)もしくは¥¥

結果の書き出し

- 結果の書き出し方法
 - `sink("ファイル名")`
 - 作業ディレクトリ以外に保存したいときには絶対パス
 - 描きだしたい計算等
 - `sink()`

- `writetable()`という関数もある
 - `writetable(data, file="ファイル名", append, sep, row.names, col.names)`

データの読み込み

- scanを用いる
 - scan(data)
 - カンマ区切りのデータならscan(data, sep=",")
- read.table (read.csv)を用いる
 - read.table(data, header=F, sep, row.names, col.names)
 - オプションは様々あるが・・・
 - read.csvは, オプションでheader=T, sep=","が標準になっている

データフレーム

➤ data.frame

- 数値ベクトルだけでなく、文字ベクトル等のデータをまとめて1つの変数とすることができる
- 各行・各列はラベルを必ずもつ
 - ラベルによる操作が可能
 - `summary()`で列ごとの特徴を見ることができる

回帰分析

➤ 回帰分析を行う

➤ データの準備

➤ `ssales<-read.table("sample2.txt", header=T, row.names=1)`

➤ 回帰分析のコマンド(lm)

➤ `lm(formula, data)` の形

➤ formulaはy~xという形

➤ 重回帰ならy~x1+x2+...

➤ オプションは様々あるが...

➤ `sales.lm<-lm(売上~気温, ssales)`

Call:

`lm(formula = 売上 ~ 気温, data = ssales)`

Coefficients:

(Intercept)	気温
-30.428	4.136

回帰分析

- 概要の出力
 - `summary(sales.lm)`

Call:

```
lm(formula = 売上 ~ 気温, data = sales)
```

Residuals:

```
   Min     1Q  Median     3Q      Max
-10.241 -3.818  1.259  2.747 11.623
```

Coefficients:

```
              Estimate Std. Error t value Pr(> |t|)
(Intercept) -30.4276    19.9695  -1.524 0.166086
  気温         4.1359     0.7427   5.569 0.000529 ***
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 7.431 on 8 degrees of freedom

Multiple R-squared: 0.7949, Adjusted R-squared: 0.7693

F-statistic: 31.01 on 1 and 8 DF, p-value: 0.0005294

回帰分析

➤ Excelの分析ツールだと

概要

回帰統計	
重相関 R	0.89158
重決定 R2	0.794915
補正 R2	0.76928
標準誤差	7.430951
観測数	10

分散分析表

	自由度	変動	分散	割られた分散	有意 F
回帰	1	1712.248	1712.248	31.00829	0.000529
残差	8	441.7522	55.21903		
合計	9	2154			

	係数	標準誤差	t	P-値	下限 95%	上限 95%	下限 95.0%	上限 95.0%
切片	-30.4276	19.96947	-1.5237	0.166086	-76.4772	15.6221	-76.4772	15.6221
気温	4.135864	0.742724	5.568509	0.000529	2.42314	5.848588	2.42314	5.848588

回帰分析

- 散布図
 - `plot(ssales$気温, ssales$売上)`
 - さらに回帰直線を追加
 - `abline(sales.lm$coef)`
- 残差プロット
 - `plot(resid(sales.lm))`
- Q-Qプロット
 - `qqnorm(resid(sales.lm))`