

第01回ー統計解析環境（R）

ビジネス アナリティクス

竹田 恒

htakeda@tiu.ac.jp

令和4（2022）年度 春

1 はじめに

2 統計解析環境 R

3 基本操作

4 演算

5 入出力

1 はじめに

2 統計解析環境 R

3 基本操作

4 演算

5 入出力

はじめに

統計解析は、10 年ほど前までは、C や Fortran など、取扱いに専門的知識を要するプログラミング言語を用いて行われていました。これを、高度なプログラミング知識がなくても誰でも利用できる形にしたものが、統計解析環境 **R** です。Python とともにデータサイエンスの世界では標準のソフトウェアツールとなっています。

R を習得すれば、統計解析から業務効率化ツールの作成までオールマイティーに、そのスキルを活用できます。データ分析に携わる方は、様々な市販のソフトウェアに手を出さなくても、これ一本で十分です。手を動かして自分でやってみることが **R** 習得の近道です。そのため、各スライドには、必ず演習を載せてあります。

1 はじめに

2 統計解析環境 R

3 基本操作

4 演算

5 入出力

統計解析環境 R

Rとは、AT&Tベル研究所が開発した統計解析用のプログラミング言語（S言語）を参考にして作られたフリーのオープンソース言語（R言語）を使用できる統計解析環境。



図 1 The R environment



図 2 Microsoft R Open

Rにより、現代統計学をほぼ網羅する広範な統計解析や出版物品質のグラフ描画が容易に可能となる。Microsoft社により開発・保守されている高速版の**R**（Microsoft R Open）も存在する。

R パッケージ

R だけでも基本的な統計解析は可能だが，ユーザーの利用目的に応じて開発された R パッケージと呼ばれる統計解析ライブラリをインストールすることで機能を拡張できる．

R パッケージは，C/C++，Fortran，R 言語で記述されており，当初は，欧米大学の統計学科の教員らが中心となり開発・保守を行っていたが，近年は民間を含む様々な分野で広く開発が進められている．すでに，1,000 を超える R パッケージがウェブ上で公開されている．

CRAN

CRAN（包括的R保存庫網）とは、Rの本体やパッケージ、マニュアル類が無償公開されているウェブサイト。ユーザーは、最寄りのミラーサイトからソフトウェアをダウンロードする。



CRAN
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

About R
[R Homepage](#)
[The R Journal](#)

Software
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

Documentation
[Manuals](#)
[FAQs](#)
[Contributed](#)

The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- [Download R for Linux](#)
- [Download R for \(Mac\) OS X](#)
- [Download R for Windows](#)

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2019-12-12, Dark and Stormy Night) [R-3.6.2.tar.gz](#). read [what's new](#) in

日本語での R の情報源としては、筑波大学の岡田氏のウェブサイト [RjpWiki](#) が有名。多くの有用な情報が掲載されており、質問もできる。



RjpWiki

<http://www.okada.jp.org/RWiki/?RjpWiki>

[\[トップ \]](#) [\[Tips紹介 \]](#) [\[初級Q & A \]](#) [\[R掲示板 \]](#) [\[日本語化掲示板 \]](#) [\[リンク集 \]](#)

[編集](#) | [凍結](#) | [差分](#) | [バックアップ](#) | [添付](#) | [リロード](#) | [新規](#) | [一覧](#) | [単語検索](#) | [最終更新](#) | [ヘルプ](#)

最新の30件

2019-12-22

- Q & A (初級者コース)/17

2019-12-20

- [トップ](#) [ページ](#) [への](#) [コメ](#) [ント](#)

2019-11-22

- R揭示板

2019-08-12

- Rで項目反応理論

2019-07-25

- shinyapps.io
- Uber H3

2019-07-17

- REvolutionRは連邦の新型か
- okinawa

- [RipWikiとは](#)

- ### ○ 主要内容

- [公式](#)
- [マニュアル](#)
- [ウェブアプリケーション](#)
- [その他のR関連の記事のインターネット検索](#)

- 注音事項

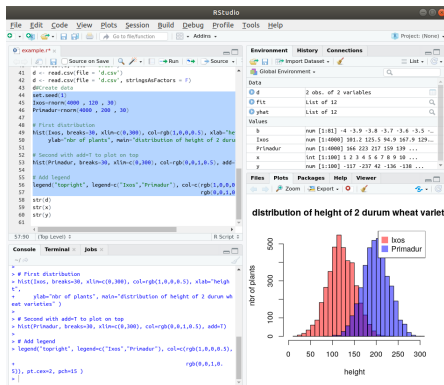
- コメント

RjpWikiとは +

RjpWikiはオープンソースの統計解析システムである**R**に関する情報交換を目的とした**Wiki**です。どなたでも自由にページを追加・編集できます。

統合開発環境 RStudio

RStudio とは、R 用の統合開発環境（IDE）で、ソースコードの編集、実行、ヘルプの表示、パッケージの作成など、プログラミングに必要な様々な便利な機能を持つ。Python の開発もできる。オープンソース版の RStudio を rstudio.com からダウンロードできる。



1 はじめに

2 統計解析環境 R

3 基本操作

4 演算

5 入出力

基本操作 スカラの作成

手順

オブジェクト名の後に、代入（付置）記号「<-」と値を入力する。

- ・「<-」の代わりに「=」も使用できる（若干意味が異なる）。
- ・オブジェクト名は、大文字と小文字は区別される。

Console 1

```
> x <- 1  
  
> kw.pv <- 3.1
```

Console 2

```
> ls() # オブジェクト名表示  
  
> rm(list=ls()) # 全消去
```

Exercise 1

「Alt + -」で代入記号（<-）を 20 回入力せよ。

オブジェクト名を表示，オブジェクトを削除せよ。

基本操作 ベクトルの作成

手順（方法1）

結合関数「c」を用いて
作成

手順（方法2）

等差数列作成記号「:」
を用いて作成

手順（方法3）

等差数列作成関数

「seq」を用いて作成

「rep」関数で同一値ベクトル作成も可能 rep(NA, 3) → NA NA NA

「?関数名」をコンソールに入力するとヘルプが表示される。

Console 3

```
> v <- c(1,6,3)
[1] 1 6 3
```

Console 4

```
> v <- 1:3
```

```
[1] 1 2 3
```

Console 5

```
> v <- seq(1,6,2)
[1] 1 3 5
```

基本操作 ベクトルの作成

Exercise 2

次のベクトルを作成せよ.

$(3, 2, 1), (3, 6, 9), (4, 2, 0), (1.5, 2.5, 3.5), (1, 2, 3, 1, 2, 3)$

基本操作 行列の作成

手順

行列作成関数「matrix」を用いて作成。行数：nrow, 列数：ncol

Console 6

```
> m <- matrix(1:4, nrow = 2, ncol = 2)
```

(オプション byrow = T で行毎に値代入)

```
> m
```

```
 [,1] [,2]
```

```
[1,] 1 3
```

```
[2,] 2 4
```

基本操作 行列の作成

Exercise 3

次の行列を作成せよ.

$$\begin{bmatrix} -2 & 0 & 2 \\ 4 & 6 & 8 \end{bmatrix}, \begin{bmatrix} NA & NA \\ NA & NA \end{bmatrix}$$

基本操作 オブジェクトの画面表示

手順

コンソールに表示させたいオブジェクト名を入力する。または、ソースコード画面でオブジェクトを選択して Ctrl + Enter を押す。

Console 7

```
> v[3]
```

```
[1] 3
```

Console 8

```
> m[2, ]
```

```
[1] 2 4
```

Console 9

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

```
[1] 1 2
```

【オブジェクト要素の R 表記】

ベクトル v の要素 i : $v[i]$

行列 m の i 行 : $m[i,]$, j 列 : $m[, j]$, 要素 (i, j) : $m[i, j]$

基本操作 オブジェクトの画面表示

Exercise 4

スカラ，ベクトル，行列の値を表示せよ．

基本操作 データフレームの作成

手順

テーブル作成関数「data.frame」を用いて作成する。

Console 10

```
> d <- data.frame(name = c('panda', 'lion'),  
                  age = c(5, 7), male = c(T, F))  
  
> d  
  name age male  
1 panda 5 TRUE  
2  lion 7 FALSE
```

基本操作 データフレームの作成

Exercise 5

漢字，数値，論理値のカラムを持つ，データフレームを作成せよ（内容自由）.

基本操作 データフレームの操作 I

手順

アクセスしたいコラム（列）やレコード（行）のインデック番号を入力する。負の番号を入れると，そのコラムが除かれる。

Console 11

```
> d[, 1]
```

```
[1] "panda" "lion"
```

Console 12

```
> d[, c(1, 3)]
```

```
name male
```

```
1 panda TRUE
```

```
2 lion FALSE
```

Console 13

```
> d[, -1]
```

```
age male
```

```
1 5 TRUE
```

```
2 7 FALSE
```

基本操作 データフレームの操作 I

Exercise 6

レコード（行）にもインデックス番号でアクセスし，値の表示や代入を行え．

基本操作 データフレームの操作 2

手順

オブジェクト名のあとにアクセスしたいコラム（列）名を\$で結びつける。または、コラム（列）名をリテラルで囲み記入する。

データフレーム d のコラム：d\$コラム名 or d[, 'コラム名']

Console 14

```
> d$age
```

```
[1] 5 7
```

Console 15

```
> d[, c('name', 'age')]
```

```
  name age
```

```
1 panda 5
```

```
2  lion 7
```

基本操作 データフレームの操作 2

Exercise 7

レコード（行）にもインデックス番号でアクセスし，値の表示や代入を行え．カラム名，レコード名は `rownames(d)`, `colnames(d)` でアクセス可能．

基本操作 データフレームの操作 3

手順

アクセスしたいコラム（列）やレコード（行）に論理値を入力する。T（TRUE）の論理値箇所のデータが抽出される。

Console 16

```
> d[, c(T, F, T)]  
  name male  
1 panda TRUE  
2  lion FALSE
```

Console 17

```
> d[d$age > 6, ]  
  name age male  
2  lion   7 FALSE
```

基本操作 データフレームの操作 3

Exercise 8

レコード（行）に論理値ベクトルでアクセスし，値の表示や代入を行え．

1 はじめに

2 統計解析環境 R

3 基本操作

4 演算

5 入出力

演算 四則

手順

和「+」、減「-」、積「*」、除「/」、乗「^」、剰余「%%」、剰商「%/」の
算法記号を使って演算する。要素ごとの演算となる。

この他、行列演算用の積「%*%」、転置「t()」、逆行列「solve()」などもある。

Console 18

```
> x <- 1:3; y <-  
1:3  
  
> x + y  
[1] 2 4 6
```

Console 19

```
> x <- 1:3; y <-  
1:3  
  
> x * y  
[1] 1 4 9
```

Console 20

```
> x <- 9; y <- 2  
  
> x %% y  
[1] 4
```

演算 四則

Exercise 9

既出のすべての演算記号を使って計算せよ（内容自由）.

演算 組込関数

手順

平均値「mean」、中央値「median」、最大「max」、最小「min」

範囲「range」、平方根「sqrt」、絶対値「abs」、丸め「round」など

Console 21

```
> x <- 1:3
```

```
> mean(x)
```

```
[1] 2
```

Console 22

```
> x <- 1:3
```

```
> range(x)
```

```
[1] 1 3
```

Console 23

```
> x <- 3.14
```

```
> round(x, 1)
```

```
[1] 3.1
```

演算 組込関数

Exercise 10

既出のすべての組み込み関数を使って、計算せよ（内容自由）。また、その他の関数、定数（ π ）は、どのようなものがあるかウェブで検索せよ。

演算 自作関数 I

手順

関数オブジェクト名 <- function (引数1, 引数2, ...) 関数式
の形式で関数を作成する。引数は値渡しとなる。

Console 24

```
> get.mbe <- function(yhat, y) mean(yhat - y)
```

```
> mbe <- get.mbe(yhat = 1:3, y = 4:6)
```

```
[1] -3
```


演算 自作関数 I

Exercise 11

RMSE（平均 2 乗誤差平方）を求める関数を作成せよ.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\hat{y}_i - y_i)^2}$$

cf. 平均: `mean()`, 平方根: `sqrt()`

演算 自作関数2

手順

複数行の関数を作成するときは、関数のスコープを示す `{}`（中括弧）や変数名を指定して出力する `return` 関数を用いる。

Console 25

```
> f <- function(x) {  
  y <- 1 + x + x ^2  
  z <- log(y)  
  return(z)  
}
```

Console 26

```
> f(2)  
[1] 1.94591
```

最後のオブジェクトを返り値とする場合は `return` はなくても良い。

Exercise 12

複数行の関数を作成せよ.

入出力 テキストデータの出力

手順

write.csv 関数を使用して、オブジェクトデータをファイルに CSV ファイル形式*で書き込む。 * CSV：カンマ区切

Console 27

```
> d0 <- data.frame(name=c('panda','lion'),age=c(5,7))  
> write.csv(d0, file = 'd0.csv')
```

quote = F オプションをつけると出力ファイル内の文字列引用符「"」を削除できる。 write.csv(d0, file = 'd0.csv', quote = F)

入出力 テキストデータの出力

Exercise 13

データフレームを作成しファイルに出力せよ.

入出力 テキストデータの出力

手順

`read.csv` 関数を使用して、CSV ファイルを読み込みオブジェクトに格納する。

Console 28

```
> d1 <- read.csv(file = 'd0.csv')
```

```
> str(d1)
```

`stringsAsFactors = F` オプションを付加すると文字列の自動因子化を抑制する。

```
read.csv(file = 'd0.csv', stringsAsFactors = F)
```

入出力 テキストデータの出力

Exercise 14

CSV ファイルを読み込み，オブジェクトに格納せよ．

入出力 Excel データの入力

手順

excel.link パッケージを利用し、R とリンクさせる。

パッケージの利用コマンド：`library(excel.link)`

Console 29

```
> library(excel.link)

> xl.workbook.open('test.xlsx')           # Excel を開く

> d <- data.frame(x=3:1, y=-1:1)

> xl['Sheet1!A1'] <- d    # Sheet1 の A 1 を起点としてデータを書き込み

> xl['Sheet1!B2'] -> x     # Sheet1 の B 2 からデータを読み込み

> xl.workbook.save('test.xlsx')           # Excel を保存
```

xlr：行名付き，xlc：列名付き，xlrc：行列名付き入出力

入出力 テキストデータの出力

Exercise 15

パッケージのヘルプにあるサンプルコードを用いて **Excel** を操作せよ.