

# R / RStudio の導入

Toshiki SHIBANO

2021-02-11

## 目次

対象	1
インストール	2
R のインストール . . . . .	2
RStudio のインストール . . . . .	2
RStudio の起動	2
R の実行と基礎	3
変数と代入 . . . . .	3
ベクトル . . . . .	4
型 . . . . .	4
データフレーム形式 . . . . .	5
ヘルプとエラー . . . . .	6
パッケージ . . . . .	6
バージョンの確認 . . . . .	7
これからのこと	8

## 対象

- R のインストールがまだの方で、プログラミングなどを初めてする方

- RStudio を入れたことのない方

(その割には不親切な内容かもしれません)

## インストール

### R のインストール

CRAN (<https://cran.ism.ac.jp>) から各自の OS に合わせてダウンロードします。基本的に最新版のインストールを推奨です。インストール途中で日本語設定が出来ると思います。特に難しいことはないと思いますが、分からないことがあれば、ネットで検索すればたくさん出て来ると思います。

### RStudio のインストール

RStudio (<https://rstudio.com/products/rstudio/download/>) からダウンロードします。無料の RStudio Desktop で十分だと思います。特に難しいことは (ry...

## RStudio の起動

1. RStudio のボタンを押して起動する
2. 上のメニューバーから **File** → **New Project**
3. Create Project 画面が出て来るので, **New Directory** → **New Project**
4. Directory Name を記入 (英語の方が良い) して, Create project as subdirectory の Browse ボタンを押して, フォルダを作るところを洗濯したら Create Project
5. 画面が更新されるので, メニューバーの File を押して **New File** → **R Script** or **R Markdown**  
R Markdown については後日まとめます。
6. スクリプト画面にコードを書いていきましょう! 実行は Shift + Enter (Mac なら command + Enter) で実行できます。

補足

ダークモードの設定はメニューバー **Tools** → **Global Options** → **Appearance** → **RStudio theme** から選択できます。

右の二つの画面は設定で色々変更出来ます。例えば, グラフを描くと右下の画面がファイル画面からプロット画面に切り替わります。

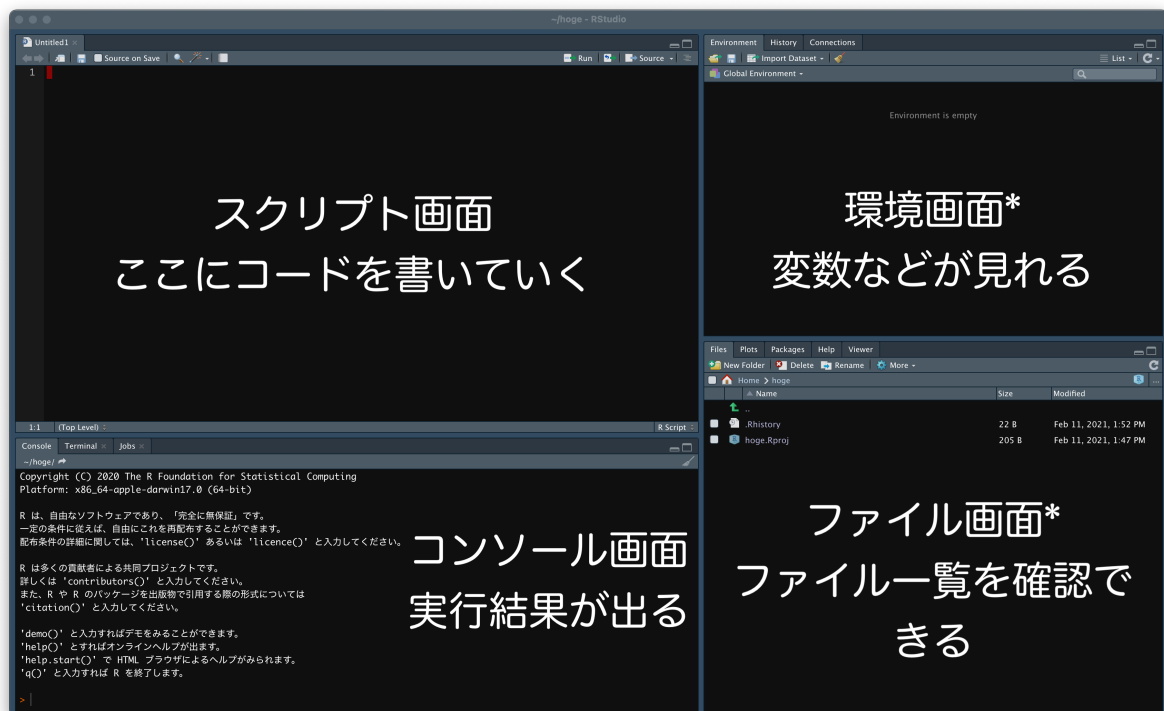


図1 RStudio の画面

## R の実行と基礎

### 変数と代入

スクリプト画面で以下のようにうって実行してみましょう。

```
a <- 10
b <- 20
a + b
```

```
## [1] 30
```

```
a * b
```

```
## [1] 200
```

```
a ^ 2
```

```
## [1] 100
```

<- 記号は代入を表します。

1 行目は、a という**変数**に 10 を代入するということを表しています。

変数とは様々なデータを保存しておく箱のようなものです。

## ベクトル

次にベクトルという重要な概念を説明します。

```
vec <- c(1, 2, 3, 4, 5)
vec
```

```
## [1] 1 2 3 4 5
```

ベクトルとはその名の通り、複数個の似たデータを一度に扱うことが出来ます。この c というのがベクトルを作成するための関数になります。

関数とは何らかの機能をもったものです。機械をイメージすると分かりやすいです。

ある命令を受けたらそれを実行して、何かしらの結果を返すものです。

つまり、上の実行結果について、

c() という関数は、1, 2, 3, 4, 5 と与えられたデータに対して、ベクトルとして vec という変数に保存することが行われています。

- 作図
- 統計検定

などを行う関数もあり、基本的にプログラミングは色々な関数を使って様々なことを実行していきます。

## 型

変数には型というものが存在します。

簡単に言うとその変数の中に入っているデータの種類です。これは重要な概念です。

例えば、数値 (numeric) や文字 (character), 論理値 (logical), 因子 (factor) など様々あります。

```
# データ型を調べる
class(1)
```

```
## [1] "numeric"
```

```
class("hoge")
```

```
## [1] "character"
```

```
a <- c(1, 2, 3)
class(a)
```

```
## [1] "numeric"
```

```
class(TRUE)
```

```
## [1] "logical"
```

```
class(NULL)
```

```
## [1] "NULL"
```

補足

本当はもう少し複雑なようです。モード (mode), クラス (class), タイプ (typeof) の3つに大別され、カッコ内の関数でそれぞれの型を調べることが出来ます。

- モード: オブジェクトに格納されている要素に対する型
- クラス: オブジェクトの属性に対する型
- タイプ: データを保存できる変数のオブジェクトに対する型

参考 biostatistics(<https://stats.biopapyrus.jp/r/basic/data-type.html>). 2021/02/11

## データフレーム形式

Rで最も重要と思うのが**データフレーム**です。データフレームとは表で、同じ列 (column, 縦) に同じ型の変数を取り、異なる列同士は異なる型の変数で良いものです。

つまり、データシートみたいなものです。

```
y <- c(1, 3, 5, 7, 9, 11)
x1 <- c(1.1, 2.3, 3.6, 4.8, 5.4, 6.0)
x2 <- c("A", "A", "A", "B", "B", "B")
df <- data.frame(y = y,
                 x1 = x1,
                 x2 = x2)
df
```

y	x1	x2
1	1.1	A

y	x1	x2
3	2.3	A
5	3.6	A
7	4.8	B
9	5.4	B
11	6.0	B

このような表をデータフレームといいます。y, x1 列は数値が入っており, x2 は文字が入っています。またこのようなデータ形式のことを tidy データといいます。tidy データは非常に重要であり, 多くの R の関数はこの tidy データを対象にしています。詳細は R for Data Science (by Hadley Wickham & Garrett Grolemund) (URL: <https://r4ds.had.co.nz/index.html>) の 12 章を参照して下さい。

#### 注意点

R を使う時に Excel で保存する時は tidy データになるように保存しましょう。

しかしながら, Excel だけで完結する場合はあなたが扱いやすいようにまとめてください。

## ヘルプとエラー

関数の使い方が分からない場合はヘルプを見ましょう。help() 関数を用いる, もしくは関数の前に? をつけることでヘルプを見ることが出来ます。

```
# c() 関数について知りたい
# これはコメントと言って、
# # より後の文章は実行されません
help(c)
?c
```

そして, R を実行しているとよくエラーがでます。しかしながら, エラーは怖くありません。まずエラー文を読み, スペルミスやカッコを疑いましょう。分からなければエラー文をコピーして検索しましょう。そこに答えがあると思います。他人に聞く時はエラー文とともに R やパッケージのバージョンも確認しましょう。

## パッケージ

パッケージとは便利な関数などをまとめたものです。多くの統計処理について, パッケージが配布されており, 車輪の再発明になることを防いでくれます。自分でもパッケージを作ることが出来ます。

install.packages 関数を用いてパッケージをインストールします。一度インストールしたパッケージは library() 関数を用いてよみこむことで使用できます。RStudio を終了・再起動すると, もう一度 library() 関数を用いてパッケージを読み込む必要があります。

試しに以下のパッケージをインストールしましょう。tidyverse パッケージは非常に便利なパッケージを集めたパッケージになっています。もし CRAN の選択画面が出てきたら Japan のところを選択しましょう。

```
# 次の文章の先頭の # を消して実行して下さい
```

```
#install.packages("tidyverse")
```

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3
```

```
## v ggplot2 3.3.2      v purrr  0.3.4
```

```
## v tibble  3.0.1      v dplyr  1.0.2
```

```
## v tidyr   1.1.2      v stringr 1.4.0
```

```
## v readr   1.3.1      v forcats 0.5.0
```

```
## -- Conflicts ----- tidyverse_conflicts
```

```
## x dplyr::filter() masks stats::filter()
```

```
## x dplyr::lag()     masks stats::lag()
```

補足 Windows の場合、library() 関数を実行すると、エラーが出る場合があります、そこに OneDrive が関わっていることがあります。OneDrive との連携を切って再インストールすると治るようです。詳細は調べて下さい。

## バージョンの確認

バージョンの確認は大事です。バージョンによって挙動が異なることがあります。質問する時は自身の OS とともに、R やパッケージのバージョンを相手に伝えるようにしましょう。

```
# R のバージョン確認
```

```
version$version.string
```

```
## [1] "R version 4.0.3 (2020-10-10)"
```

```
version # 詳しい環境をみたい時
```

```
##
## platform      x86_64-apple-darwin17.0
## arch          x86_64
## os            darwin17.0
## system        x86_64, darwin17.0
## status
## major         4
```

```
## minor          0.3
## year           2020
## month          10
## day            10
## svn rev        79318
## language       R
## version.string R version 4.0.3 (2020-10-10)
## nickname       Bunny-Wunnies Freak Out
```

```
# RStudio のバージョン
# R Markdown ではエラーが出るので、コメントアウトしています
# メニューバーの Help → About RStudio からでも確認できます
#RStudio.Version()

# パッケージバージョンの確認
packageVersion("tidyverse")
```

```
## [1] '1.3.0'
```

## これからのこと

今回は本当に基本的なことしか述べていません。一度、書籍や PDF など勉強するのをお勧めします。つぎはぎの知識では、より複雑なことをする時に苦労すると思います (私がそうです)。多くの R の本はその特色上、統計解析とセットになっていて、両方を同時に勉強することができます。本屋さんなどで前書きを読んだり、本を流し読みしたりして、自分にあった本を探して下さい。一例として以下に私が (一部) 読んだことのある本・PDF を上げておきます。

- R によるやさしい統計学 by 山田剛史・杉澤武俊・村井潤一郎
- R による統計解析 by 青木繁伸
- R Tips (<http://cse.naro.affrc.go.jp/takezawa/r-tips/r.html>)

英語が大丈夫な方は、調べれば日本語訳の本は有料だが、英語版は無料な PDF がたくさん出て来ると思います。次のは私が読んでみたい本です。

- Hands-On Programming with R (<https://rstudio-education.github.io/hopr/>) by Garrett Grolemund