データサイエンスをはじめましょう - Data Science for All -

John Doe

2023-02-18

目次

この文書について

データサイエンスを始めてみませんか。

データサイエンスは、広い意味をもったことばで、一口に、まなび始めると言っても、さまざまな始め方があると思います。本書では、そのひとつを提案するとともに、共に学んでいきたいと願って、書き始めました。

みなさんも一緒にデータサイエンスを学んでみませんか。

著者について

著者は、大学の学生の時以来、数学を学び、大学で教え、2019 年春に退職。それ以来、少しずつ、データサイエンスを学んでいます。

幸運にも、2019 年 9 月の日本数学会教育委員会主催教育シンポジウムで、「文理共通して行う数理□データサイエンス教育」という題で、話す機会が与えられ、その後、あることが契機となり、2020 年度から、毎年、冬に、大学院一般向け（分野の指定なし）の授業、「研究者のためのデータ分析（Data Analysis for Researchers)」を担当しています。複数の教員で担当しますが、基本的な部分は、わたしが教えています。受講生は 20 人程度ですが、殆どが、外国人。それも、多国籍で、多くても一国から三人程度。英語で教えています。

コンピュータ言語について

統計解析のために開発された R を使います。いずれは、python についても触れたいと思いますが、プログラミングの経験がない方も含めて、最初にデータサイエンスを学ぶには、R は最適だと考えています。特に、R Studio IDE（integrated development environment, 統合開発環境）で、R を使うことがとても、簡単になっています。さらに、簡単なものであれば、Posit Cloud で試したり、共有することも可能です。また、再現性（Reproducibility）や、なにを実行しているのかの説明を同時に記述すること（Literate Programming）は、非常に重要ですが、その記

述も、R Markdown によって、可能になっています。この文書も、R Markdown の一つの形式の、bookdown を利用しています。最後に、Bookdown に関連して、膨大な数の、参考書も、無償で提供されており、オンラインで読むことができることも、R をお薦めする理由です。

ただし、日本語のものは、まだ十分とは言えない状況です。この文書を書き始めたのも、すこしでも、お役に立つことができればとの、気持ちが背景にあります。

言語について

ご覧の通り、本書は、日本語で書かれています。用語は、英語、あるいは、英語を追記、または、英語をカタカナにしただけのものを使用する可能性が大きいですが、説明は、極力、日本語で書いていく予定です。

しかし、基本的に、コード（プログラムの記述）には、日本語を使わないで書いていく予定です。とくに、初心者にとっては、日本語の扱いは、負担になることが多いからです。最近は、コードの中で日本語を使用しても、ほとんど、問題は起きないように思います。そうであっても、世界の人の共通言語として、プログラム言語を学んでいくときには、日本語を使わないことは意義があると思います。

少し慣れてきて、日本語のデータなどを扱うときには、コードにも日本語を使う必要ができていますから、日本語の利用についても、追って説明していきます。APPENDIX **??** を参照してください。

最初は、みなさんも、変数（variable）や、オブジェクト（object）に名前をつけるときは、半角英数を使い、日本語は、使わないようにすることをお勧めします。

PDF、ePub 版について

実は、PDF 版と、ePub 版も作成しています。しかし、扱いが異なるので、ある程度完成するまでは、ほとんど更新しない予定です。いずれ、これらも、更新したものを公開できると良いのですが。試験公開版は、下のリンクにあります。

- PDF 版
- ePub 版

0.1 はじめに

Data Science: データ (Data) を活用して課題を発見□探求し、適切な解決策を探る意思決定のための科学(Decision Science)で、エンピリカル(Empirical Study) すなわち、理論ではなく、実証性を特徴とする。データから得られる特徴を表示するとともに、数理モデルを適用し□機械学習などで評価し□ア

ルゴリズムを策定する数理的思考を通して得られた結果を、可視化などによってコミュニケーションをおこない、共有し、他者の意見を聞き理解する努力をしながら、さらに課題について、あらたにデータを活用して考え、検証し、適切な解決策がもたらす新たな課題も予測しながら、調整をはかる。

第Ⅰ部

PART I PUBLIC DATA

0.2 Public Data

まずは、パブリックデータを見てみましょう。大きな機関のパグリックデータには、ダッシュボード（dashboard）と呼ばれている、パラメタを変更して、そのグラフを描くなどの機能が付いているものもあります。

0.2.1 World Bank

0.2.1.1 Open Government Data Toolkit: Open Data Defined

The term **Open Data** has a very precise meaning. Data or content is open if anyone is free to use, re-use or redistribute it, subject at most to measures that preserve provenance and openness.

1. The data must be *legally open*, which means they must be placed in the public domain or under liberal terms of use with minimal restrictions.
2. The data must be *technically open*, which means they must be published in electronic formats that are machine readable and non-proprietary, so that anyone can access and use the data using common, freely available software tools. Data must also be publicly available and accessible on a public server, without password or firewall restrictions. To make Open Data easier to find, most organizations create and manage Open Data catalogs.

0.2.2 Worldbank Data

- Climate Change Knowledge Portal: https://climateknowledgeportal.worldbank.org
  - country summary

0.2.3 World Bank: WDI - World Development Indicaters

- World Bank: https://www.worldbank.org
- Who we are:
  - To end extreme poverty: By reducing the share of the global population that lives in extreme poverty to 3 percent by 2030.
  - To promote shared prosperity: By increasing the incomes of the poorest 40 percent of people in every country.
- World Bank Open Data: https://data.worldbank.org
  - Data Bank, World Development Indicators, etc.

0.2.3.1 World Development Indicator

- World Development Indicators (WDI) : the World Bank's premier compilation of cross-country comparable data on development; 1400 time series indicators
  - Themes: Poverty and Inequality, People, Environment, Economy, States and Markets, Global Links
  - Open Data & DataBank: Explore data, Query database
  - Bulk Download: Excel, CSV
  - API Documentation

0.2.4 OECD

OECD Data: https://data.oecd.org/

0.2.5 UN Data

UNdata: https://data.un.org

0.2.6 Our World in Data

owid: https://ourworldindata.org/

0.2.7 Eurostat

eurostat: https://ec.europa.eu/eurostat

第 II 部

PART II BASICS

0.3 R on R Studio

0.3.1 はじめに

R Studio で R を使うことを始めましょう。

このページの一番下に、簡単な解説ビデオがついています。

0.3.2 R と R Studio

> R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. https://www.r-project.org
>
> R は、無償で提供されている、統計解析とグラフを描写する環境です。Windows、MacOS や、Linux で利用することが可能です。
>
> RStudio is an integrated development environment (IDE) for R and Python. It includes a console, syntax-highlighting editor that supports direct code execution, and tools for plotting, history, debugging, and workspace management. RStudio is available in open source and commercial editions and runs on the desktop (Windows, Mac, and Linux). https://posit.co/products/open-source/rstudio/
>
> RStudio は、R と Python のための、総合開発環境です。RStudio には、プログラムを実行したり、制御やジョブ管理のための、コンソール（console)、コードを書いたり、実行したりする、文書の編集をする、エディター（Editor）とともに、グラフを表示したり、履歴や、プログラムを修正するなどのための、さまざまなツールが付属しています。RStudio はオープンソースで提供され、Windows、Mac および、Linux で利用可能で、有償版のサービスと無償版を提供しています。

R は、統計解析のためのシステムで、R Studio は、R（および Python）を利用するための、総合開発環境です。そこで、「R Studio で R を利用する」という表現をします。

0.3.3 R と R Studio のインストール

R と R Studio をインストールします。

両方とも、インストールすることが必要です。

0.3.3.1 R のインストール

https://cloud.r-project.org

上のリンクから、Windows、macOS または、Linux を選択して、インストールしてください。

macOS の場合は、M1, M2 など、最近の Apple Silicon の CPU で動くコンピュータか、以前の、Intel の CPU で動くものか、選択してください。Mac の左上の、りんごマークの、このコンピュータについてから、確認できます。

不明の場合は、「R のインストール」と検索してみてください。

0.3.3.2 R Studio のインストール

http://www.rstudio.com/download

上のリンクから、Windows 10/11 または、macOS 11+ を選択してください。これら以外の、古いシステムのコンピュータの場合は、下のサイトから、探してください。

https://docs.posit.co/previous-versions/

不明の場合は、「RStudio のインストール」と検索してみてください。

0.3.4 プロジェクト - Project

RStudio で R を利用する場合には、プロジェクトを作成することを強く勧めます。

1. まず、R Studio を起動します。
2. 上のメニューの、File から、New Project を選択します。New Directory（新しいディレクトリー）を選択し、プロジェクトを作成する Directory を決めて、名前をつけます。その名前が、プロジェクト名になります。

- Directory（フォルダー）を指定してその名前をつけて、プロジェクトを作成します。
- Directory が階層に分かれているときは、どこに作成するかを選択してから、名前をつけて、作成します。

3. 一旦、R Studio を終了してみましょう。
4. プロジェクトの起動には、いくつかの方法があります。

- まず、R Studio を起動。一つしかプロジェクトがない場合は、そのプロジェクトが起動すると思います、。上に、プロジェクト名が掲載されていれば、

問題ありません。

- File から、Open Project を選択し、起動したい、プロジェクトの Directory（フォルダー）を選択して起動します。
- File から、Recent Project（最近使ったプロジェクト）を選択すると、プロジェクト名が表示されますから、選択すると起動することができます。
- コンピュータのプロジェクト入っているディレクトリー（フォルダー）をさがし、そこに、プロジェクト名.Rproj とあるものを見つけて、それを開くと、そのプロジェクトが起動します。

5. 作業後は、保存しますかと聞かれますから、保存して終了してください。

0.3.5 コンソールで実行 - Run in Console

プログラム（コード）の実行は、いくつかの方法がありますが、一番、基本的な、コンソール（Console）での実行にすいて、説明します。Console は、R Studio の左下にあります。（左の枠が一つになっているかもしれません。その一番左のタブが Console です。選択されていない場合は、Console を選択してください。）

0.3.5.1 最初の四つ

下の、四つを、一つずつ、一番下の、> マークの次に書き（または、コピー□ペーストして）Return または、Enter キーを押してください。実行結果が、その下に出ます。最後の、`plot(cars)` は、`cars` というデータの、散布図が右下の、Plots タブに表示されます。

- `head(cars)`
- `str(cars)`
- `summary(cars)`
- `plot(cars)`

エラーが表示されたら、もう一度、スペルを確認して、入力してみてください。

次のような、結果が表示されると思います。簡単な説明をつけます。

```
head(cars)
#>   speed dist
#> 1     4    2
#> 2     4   10
#> 3     7    4
#> 4     7   22
#> 5     8   16
#> 6     9   10
```

`head(cars)` は、`cars` という、R に付属している、データの、最初（頭 head）の

6 行を、表示します。

```
str(cars)
#> 'data.frame':    50 obs. of  2 variables:
#>  $ speed: num  4 4 7 7 8 9 10 10 10 11 ...
#>  $ dist : num  2 10 4 22 16 10 18 26 34 17 ...
```

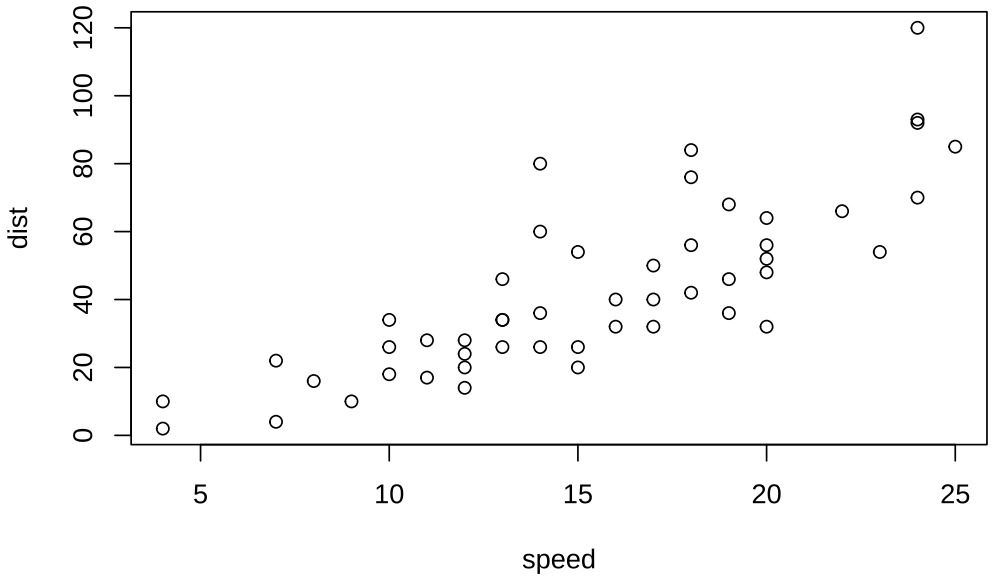
`str(cars)` は、`cars` という、R に付属している、データの構造（structure）を表示します。`data.frame` とありますが、これは、矩形になったデータ（各列の長さがおなじ）の一番簡単なクラスの名前で、2 変数、それぞれが、50 個の数値データ（numerical data）からなっていることがわかります。

head(cars) では、縦に表示されていたものが、横に表示されています。*speed*` `dist とありますが、cars*speed*`,`*cars*dist は、cars‘ データの、それぞれの列を意味します。

```
summary(cars)
#>      speed           dist
#>  Min.   : 4.0   Min.   :  2.00
#>  1st Qu.:12.0   1st Qu.: 26.00
#>  Median :15.0   Median : 36.00
#>  Mean   :15.4   Mean   : 42.98
#>  3rd Qu.:19.0   3rd Qu.: 56.00
#>  Max.   :25.0   Max.   :120.00
```

`cars` データの概要（summary）が表示されます。各列（変数）について、最小値（Minimum）、小さい方から、4 分の 1 を切り捨てたときの最小の値（1st Quadrant）、中央値（Median）、平均（Mean）、大きい方から、4 分の 1 を切り捨てたときの最大の値（3rd Quadrant）、最大値（Maximum）が表示されます。

```
plot(cars)
```



0.3.5.2 アサインメント、ヘルプ

コンソールで次のそれぞれを、試してみてください。

- `df <- cars`

`df` に、`cars` をアサインします。すなわち、`df` が、`cars` の内容に置き換わります。`cars` はデータですが、データを含む、オブジェクトの名前を設定するためにも使います。オブジェクト名は。英文字から始まれば、かなりの自由度がありますが、わたしは、英文字と数字と `_`（underscore）程度しか使わないようにしています。

- `head(df)`

`head(df)` は、`head(cars)` と同じ出力が得られます。

- `View(cars)`

左上の、窓枠が開き、`cars` の内容が表示されます。列名のところには、三角形も表示され、それを用いると、大きい順、小さい順などに、並び替えることも可能です。

- `?cars`

右下の、窓枠の　Help タブに、`cars` の情報が表示されます。Help タブにある、虫眼鏡がついた、検索窓（search window）に、`cars` といれても、同じ結果が得られます。内容を確認してください。

一番上には `cars {datasets}` とありますが、これは、`datasets` というパッケージの、`cars` だという意味です。そこで、`datasets` を調べてみましょう。

- `?datasets`

“The R Datasets Package” だと書かれていて、さらに、

This package contains a variety of datasets. For a complete list, use library(help = “datasets”).

さまざまなデータが含まれています。全てのリストをみるには、`library(help = "datasets")` を使ってください。

とありますから、`library(help = "datasets")` をコンソールに入力してみてください。

- `library(help = "datasets")`

左上の窓枠に、リストが表示されます。古いデータばかりですが、例として使うには、十分すぎるぐらいの、数のデータがあります。これらは、Toy Data（おもちゃのデータ）と呼ばれることもあります。

`cars` も見つかりましたか。

0.3.5.3 おすすめ

コンピュータのシステムが、日本語であると、R の言語も日本語になっているはずです。そこで、エラーが発生すると、一部、日本語で表示されます。しかし、ネット上などで、そのエラーの対応を検索するときは、英語のエラーメッセージで検索した方が、解決方法が得られる可能性が高いので、わたしは、英語に設定しています。英語にするには、Console で次のようにします。

言語を英語に設定: `Sys.setenv(LANG = "en")`

RStudio を終了して、もう一度起動すると、日本語に戻っていると思います。ですから、作業の最初、または、エラーが出たら、変更することをお勧めします。

日本語に戻したいときは、次のようにします。

言語を日本語に設定: `Sys.setenv(LANG = "ja")`

さまざまな Help なども、すべて日本語で表示されれば日本語を使うのは有効かもしれませんが、すくなくとも、現在は、そうではないので、上に説明したことから、英語に設定することをお勧めします。

0.3.5.4 練習

1. `head(cars, 10L)` は何が出力されますか。`head(cars, n=10L)` と同じですか。
2. `?head` または、Help の検索窓に `head` と入力して、説明を見てみてください。`head(cars, n=10L)` などについて、書いてありましたか。他には、どのようなことが分かりましたか。

3. `datasets` のデータのいくつかについて、そのデータの help や、`head`, `str`, `summary` などを使ってみてください。これらで表示できない場合はありますか。データについては、最初に、これら、三つを試してみることをお勧めします。わかったことをメモしておくと良いでしょう。`datasets` のリストをみるには、`library(help = "datasets")` でしたね。

0.3.6 RStudio について

RStudio は多くの機能を持っています。

0.3.6.1 四つの窓枠とタブ Four Panes and Tabs

- Top Left: Source Editor
- Top Right: Environment, History, etc.
- Bottom Left: Console, Terminal, Render, Background Jobs
- Bottom Right: Files, Plots, Packages, Help, Viewer, Presentation

0.3.7 R Script 実行記録

R Script を使って、コードを実行すると、その記録を残すことができます。

0.3.7.1 R Script の作成

- RStudio の上のメニュー□バーから File > New File > R Script を選択します。
- File > Save As で、名前をつけて保存します。{file_name}.R が作成されます。
  - 右下の、Files から、ファイルを確認してください。
- `head(cars)`, `str(cars)`, `summary(cars)`, `plot(cars)` などと改行をしながらコードを書きます。
- 実行するには、カーソルの場所で Ctrl+Shift+Enter (Win) または Cmd+Shift+Enter (Mac) とすると、カーソルのある行か、その下の行で、最初のコードが実行されます。
  - R Script エディターの上にある、Run ボタンを押しても、同様に実行されます。
  - Run ボタンの右の、Source ボタンを押すと、そのスクリプトの、最初からすべて実行されます。
- 最後には保存しておきましょう。

0.3.7.2 R Script による実行

新しく、R Script を作成し、この下の、コード（ハイライトされている部分）をコピー□ペーストして、保存し、実行してみてください。

それぞれ、どのようなことをしているでしょうか。

```
#################
#
# basics.R
#
#################
# 'Quick R' by DataCamp may be a handy reference:
#     https://www.statmethods.net/management/index.html
# Cheat Sheet at RStudio: https://www.rstudio.com/resources/cheatsheets/
# Base R Cheat Sheet: https://github.com/rstudio/cheatsheets/raw/main/base-r.
# To execute the line: Control + Enter (Window and Linux), Command + Enter (M
## try your experiments on the console

## calculator

3 + 7

### +, -, *, /, ^ (or **), %%, %/%

3 + 10 / 2

3^2

2^3

2*2*2

### assignment: <-, (=, ->, assign())

x <- 5

x
```

```
#### object_name <- value, '<-' shortcut: Alt (option) + '-' (hyphen or minus)
#### Object names must start with a letter and can only contain letter, numbers, _ and .

this_is_a_long_name <- 5^3

this_is_a_long_name

char_name <- "What is your name?"

char_name

#### Use 'tab completion' and 'up arrow'

### ls(): list of all assignments

ls()
ls.str()

#### check Environment in the upper right pane

### (atomic) vectors

5:10

a <- seq(5,10)

a

b <- 5:10

identical(a,b)

seq(5,10,2) # same as seq(from = 5, to = 10, by = 2)

c1 <- seq(0,100, by = 10)

c2 <- seq(0,100, length.out = 10)

c1
```

```
c2

length(c1)

#### ? seq   ? length   ? identical

(die <- 1:6)

zero_one <- c(0,1) # same as 0:1

die + zero_one # c(1,2,3,4,5,6) + c(0,1). re-use

d1 <- rep(1:3,2) # repeat


d1

die == d1

d2 <- as.character(die == d1)

d2

d3 <- as.numeric(die == d1)

d3

### class() for class and typeof() for mode
### class of vectors: numeric, charcters, logical
### types of vectors: doubles, integers, characters, logicals (complex and ra

typeof(d1); class(d1)

typeof(d2); class(d2)

typeof(d3); class(d3)

sqrt(2)
```

```
sqrt(2)^2

sqrt(2)^2 - 2

typeof(sqrt(2))

typeof(2)

typeof(2L)

5 == c(5)

length(5)

### Subsetting

(A_Z <- LETTERS)

A_F <- A_Z[1:6]

A_F

A_F[3]

A_F[c(3,5)]

large <- die > 3

large

even <- die %in% c(2,4,6)

even

A_F[large]

A_F[even]

A_F[die < 4]
```

```r
### Compare df with df1 <- data.frame(number = die, alphabet = A_F)
df <- data.frame(number = die, alphabet = A_F, stringsAsFactors = FALSE)

df

df$number

df$alphabet

df[3,2]

df[4,1]

df[1]

class(df[1])

class(df[[1]])

identical(df[[1]], die)

identical(df[1],die)

####################
# The First Example
####################

plot(cars)

# Help

? cars

# cars is in the 'datasets' package

data()

# help(cars) does the same as ? cars
# You can use Help tab in the right bottom pane
```

```
help(plot)
? par

head(cars)

str(cars)

summary(cars)

x <- cars$speed
y <- cars$dist

min(x)
mean(x)
quantile(x)

plot(cars)

abline(lm(cars$dist ~ cars$speed))

summary(lm(cars$dist ~ cars$speed))

boxplot(cars)

hist(cars$speed)
hist(cars$dist)
hist(cars$dist, breaks = seq(0,120, 10))
```

0.3.7.2.1 スクリプト 1: basics.R

```
# https://coronavirus.jhu.edu/map.html
# JHU Covid-19 global time series data
# See R package coronavirus at: https://github.com/RamiKrispin/coronavirus
# Data taken from: https://github.com/RamiKrispin/coronavirus/tree/master/csv
# Last Updated
Sys.Date()

## Download and read csv (comma separated value) file
```

```
coronavirus <- read.csv("https://github.com/RamiKrispin/coronavirus/raw/maste
# write.csv(coronavirus, "data/coronavirus.csv")

## Summaries and structures of the data
head(coronavirus)
str(coronavirus)
coronavirus$date <- as.Date(coronavirus$date)
str(coronavirus)

range(coronavirus$date)
unique(coronavirus$country)
unique(coronavirus$type)

## Set Country
COUNTRY <- "Japan"
df0 <- coronavirus[coronavirus$country == COUNTRY,]
head(df0)
tail(df0)
(pop <- df0$population[1])
df <- df0[c(1,6,7,13)]
str(df)
head(df)
### alternatively,
head(df0[c("date", "type", "cases", "population")])
###

## Set types
df_confirmed <- df[df$type == "confirmed",]
df_death <- df[df$type == "death",]
df_recovery <- df[df$data_type == "recovery",]
head(df_confirmed)
head(df_death)
head(df_recovery)

## Histogram
plot(df_confirmed$date, df_confirmed$cases, type = "h")
plot(df_death$date, df_death$cases, type = "h")
# plot(df_recovered$date, df_recovered$cases, type = "h") # no data for recov

## Scatter plot and correlation
```

```
plot(df_confirmed$cases, df_death$cases, type = "p")
cor(df_confirmed$cases, df_death$cases)



## In addition set a period
start_date <- as.Date("2022-07-01")
end_date <- Sys.Date()
df_date <- df[df$date >=start_date & df$date <= end_date,]
##

## Set types
df_date_confirmed <- df_date[df_date$type == "confirmed",]
df_date_death <- df_date[df_date$type == "death",]
df_date_recovery <- df_date[df_date$data_type == "recovery",]
head(df_date_confirmed)
head(df_date_death)
head(df_date_recovery)

## Histogram
plot(df_date_confirmed$date, df_date_confirmed$cases, type = "h")
plot(df_date_death$date, df_date_death$cases, type = "h")
# plot(df_date_recovered$date, df_date_recovered$cases, type = "h") # no data for recovery

plot(df_date_confirmed$cases, df_date_death$cases, type = "p")
cor(df_date_confirmed$cases, df_date_death$cases)

#### Extra
plot(df_confirmed$date, df_confirmed$cases, type = "h",
     main = paste("Comfirmed Cases in",COUNTRY),
     xlab = "Date", ylab = "Number of Cases")
```

0.3.7.2.2 スクリプト 2: coronavirus.T

0.3.7.3 練習

上の、coronavirus.R について

1. COUNTRY <- "Japan" の Japan を他の国に変えてみましょう。
2. start_date <- as.Date("2022-07-01") の日付を、他の日付に変えてみましょう。
3. df_confirmed$cases と df_death$cases についてどんなことがわかりま

すか。

4. 発見や、問いがあれば、書き出してみましょう。

0.3.7.4 Tips

キーボード□ショートカットと言われる、さまざまな機能があります。

- 上のメニュー□バー: Help > Keyboard Short Cut Help 確認してみてください。
- 右下の窓枠: Files タブから、ファイルの確認ができます。

0.3.8 パッケージ - Packages

R packages are extensions to the R statistical programming language containing code, data, and documentation in a standardised collection format that can be installed by users of R using Tool > Install Packages in the top menu bar of R Studio. https://en.wikipedia.org/wiki/R_package

R パッケージは、R の拡張機能で、コード、データ、ドキュメントを標準化されたコレクション形式で含んでおり、標準的なものは、R Studio の Top Bar の Tool > Install Packages からインストールできます。

0.3.8.1 パッケージのインストール

いずれ使いますので、まずは、三つのパッケージをインストールしてみましょう。

- `tidyverse`
- `rmarkdown`
- `tinytex`

インストール方法はいくつかあります。

一つ目は、上のメニューバーの Tool から、Install Packages ... を選択します。そして、パッケージーズにインストールしたい、パッケージ名を入力します。そのパッケージ名が下にも出れば、Install ボタンを押してください。入力した名前の下にパッケージ名が出ない場合は、スペルが間違っている可能性がありますから、確認して、入れ直してください。

Console に、`install.packages("tidyverse")` などと表示され、たくさんメッセージが出ます。終了すると、> のマークがでます。

二つ目は、`install.packages("tidyverse")` のような書式で書いて、Console に入れる方法です。

三つ目は、右下の窓枠の Packages のタブにある、Install というボタンを押す方法です。すると、一番目の方法に、戻り、パッケージ名を入力できるようになりま

す。

この Packages タブにある、ものが、すでに、インストールされているパッケージです。そのなかで、`base` や、`datasets` などいくつかに、チエックがついていると思いますが、それらは、ロードされていて、いつでも、使える状態になっていることを意味しています。ロードは、たとえば、`library(tidyverse)` のようにしますが、それは、いずれもう一度説明します。

インストールは一回だけ。ときどき、Tools > Check for Package Update をつかって、Update しておくと良いでしょう。

0.3.8.2 備考

Package によっては、Source から Compile するかと聞いてくる場合があります。どちらでも、良いのですが、特に、問題が起こっていなければ、No でよいと思います。コンピュータにあった形でインストールすることが必要な場合は、Yes とします。

同じパッケージをもう一度、インストールしたり、または、関連するパッケージがあるような場合、R をリスタートするかと聞いてくることがあります。特に問題が起こらなければ、No で構いません。ただ、エラーが起こって、それに関連して、特別なパッケージをインストールする必要がある場合がありますが、そのときは、Yes としてください。

0.3.9 クラウド - Posit Cloud

RStudio Cloud は、誰でもオンラインでデータサイエンスを行い、共有し、教え、学ぶことができる、軽量でクラウドベースのソリューションです。2022 年 11 月に、会社名が、RStuio から Posit に変更になったこともあり、Posit Cloud となっていますが、また、RStudio Cloud と表示されている箇所もありますので、併記しておきます。

0.3.9.1 クラウドサービス How to Start Posit Cloud

まず、サインアップして、使ってください。一ケ月の利用時間の限度など、設定されていますが、どこからでも、インターネットにつながっていれば使えるので、わたしは、いつくかアカウントを持って、活用しています。

1. Go to https://posit.cloud/
2. Sign Up: top right
3. Email address or Google account
4. New Project: Project Name

0.3.10 練習問題 Posit Primers

Posit Primers https://posit.cloud/learn/primers

教科書 “R for Data Science” は、`tidyverse` パッケージを中心に、データサイエンスについて解説したものですが、Posit Primers は、演習問題をしながら、教科書の内容を理解できるように構成されています。

0.3.10.1 最初の演習 The Basics – r4ds: Explore, I

- Visualization Basics
- Programming Basics

ぜひこれら二つの演習問題を、トライしてください。解説を読んでいただけでは、データサイエンスは身につきません。

0.3.11 参考文献 References

一番目は、すでに紹介した、教科書です。二番目は、この文書を作成している、Bookdown というパッケージのサイトですが、そこに、たくさんの本が、無償で公開されています。素晴らしい本がたくさん含まれています。

- R For Data Science, by H. Wickham: https://r4ds.had.co.nz
  - Introduction: https://r4ds.had.co.nz/explore-intro.html#explore-intro
- Bookdown: https://bookdown.org, Archive

下の一番目は、R 入門を、 2 時限の講義でしたときのものです。二番目と三番目は、講義で使ったものを、まとめたものです。教科書のようには、できていませんが、参考になる部分もあるかと思いますので、紹介しておきます。

- Introducton to R
- Data Analysis for Researchers 2022
- Data Analysis for Researchers 2021

0.3.12 YouTube Video - getstarted

- ファイル: https://ds-sl.github.io/intro2r/getstarted.html

0.3.13 追記

R Studio または、RStudio Cloud（Posit Cloud）以外で、R を使われる方のために、少しだけ書いておきます。個人的には、Google colab と、Cocalc を利用しています。

Google colab は、Google アカウントの作成、Cocalc は、Cocalc アカウントの作成、または、Google アカウントか、GitHub アカウントのリンクが必要です。

Google アカウントをお持ちの方は多いと思うので、Google colab について、最低限のことのみ、書いておきます。

0.3.13.1 Google colab で R

基本的に、`python` 開発環境として構築されているものですが、R でも使うことができます。

1. Google アカウントにログインします。
2. ここ をクリックして起動します。
3. 一番上に、ノートブック名が `Untitled0.ipynb` などと表示されますから、適当に変更します。
4. ＋コード、＋テキスト　とあり、最初のコードの 1 行が表示されていますから、たとえば、`head(cars)` と入れて、左の三角を押します。すると、最初だけ少し時間がかかりますが、その下に結果がでます。
5. 次に、上や、最後の行の直下に、表示される、＋コード、＋テキストをクリックして、あたらしい、コード□チャンクか、テキスト□チャンクを書き入れていきます。
6. `tidyverse` などは、すでにインストールされていますが、使いたいときは、`library(tidyverse)` とし、インストールされていないときは、`install.packages("WDI")` などとします。

ノートを、保存、印刷、ダウンロードなど可能です。

フォルダーを作成して、外部ファイルを読み込んだり、書き出したりすることも可能です。

0.3.13.1.1 参考にしたもの

- How to use R in Google Colab:

0.4 R Markdown

0.4.1 Reproducible and Literate Programming

データサイエンスは、サイエンス（科学）ということばもついていますが、特に、根拠に基づいた（evidence based）とか、データに基づいた（data based）ということばを使うときには、なおさら、再現可能性（reproducibility）や、コードの内容の説明などのコミュニケーションにも注力する必要があります。このことを心がけて、データサイエンスを学んでいきましょう。

表題にある、“Reproducible and Literate Programming” は、Reproducible（再現可能）かつ、Literate　な（理解できるように記述した）Program（プログラム□コード）を共有することをたいせつにしましょうということです。

0.4.1.1 目的、問いなど

プロジェクトの目的、問いなどは、途中で変わっていくこともありますが、その都度に、メモをしておくと良いでしょう。

0.4.1.2 データについて

どのようなデータをどのように取得してきたかを、記録し、伝えられるようにすることが、必要です。データを取得するときから、取得方法や、それを伝える方法にも常に気をつけましょう。

0.4.1.3 コードについて

どのようなコードでそのグラフ（chart）などが得られたかも、単にコードを記述するだけでなく、それぞれの部分に、説明を付与することも有効です。

0.4.1.4 グラフについて

視覚化（visualization）は、とても有効です。そこで、見て理解したこと、観察したこと（observations）などは、簡単でも構いませんから、必ず、記録しておきましょう。

0.4.1.5 まとめ: R Markdown の目的

まさに、このようなことを可能にするのが、R Markdown です。少しずつ学んでいきましょう。

0.4.2 準備：パッケージのインストール

R パッケージは、R の拡張機能で、コード、データ、ドキュメントを標準化されたコレクション形式で含んでおり、標準的なものは、R Studio の Top Bar の Tool > Install Packages からインストールできます。

- `tidyverse`
- `rmarkdown`
- `tinytex`

インストールを複数回しても問題はありませんが、インストールされているかどうかは、Packages タブから確認することができます。

インストールは一回だけ。ときどき、Tools > Check for Package Update をつかって、Update しておくと良いでしょう。

0.4.3 R Notebook

R Markdown はデータサイエンスのためのオーサリングフレームワーク。

コード（プログラム）とその実行結果、を記録□表示し、高品質のレポートの作成を可能にします。

R Notebook は、独立してインタラクティブに実行できるチャンクを持つ R Markdown ドキュメントの一つの形式で、入力のすぐ下に出力が表示することができます。

1. File > New File > R Notebook
2. Save with a file name, say, test-notebook
3. Preview by [Preview] button
4. Run Code Chunk plot(cars) and then Preview again.

0.4.4 日本語のテンプレート

下のリンクを開き、右上の Code ボタンから、Download Rmd を選択すると、ダウンロードできますから、ダインロードしたものを、プロジェクト□フォールダーに移動またはコピーしてください。ダウンロードできないときは、Ctrl を押しながら、Download Rmd をクリックすると、Save As で保存できると思います。ブラウザーによって仕様が異なりますから、適切な方法を選んでください。

- https://ds-sl.github.io/intro2r/RNotebook-J.nb.html
- https://ds-sl.github.io/intro2r/Rmarkdown-J.nb.html

Windows でも、Mac でも提供されている、Google Chrome の場合には、Code ボタンから、ダンロードされるはずです。

0.4.5 R Markdown いくつかの Output

```
---
title: "Testing R Markdown Formats"
author: "ID Your Name"
header-includes:
  - \usepackage{ctex}
output:
  html_notebook: default
  html_document: default
  pdf_document: default
    latex_engine: xelatex
  word_document: default
  powerpoint_presentation: default
  ioslides_presentation: default
---
```

PDF でエラー? コンソールで `tinytex::install_tinytex()`

- TeX システムがインストールされている場合は不要

0.4.6 YouTube Video - rmarkdown

第 III 部

PART III INSTITUTIONAL DATA

0.5 World Bank

0.5.1 World Development Indicator (WDI)

パッケージと tidyverse と WDI を使いますから、下のコードによって、ロードします。

```
library(tidyverse)
#> -- Attaching packages ------------------- tidyverse 1.3.2 --
#> v ggplot2 3.4.1     v purrr   1.0.1
#> v tibble  3.1.8     v dplyr   1.1.0
#> v tidyr   1.3.0     v stringr 1.5.0
#> v readr   2.1.4     v forcats 1.0.0
#> -- Conflicts ---------------------- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
library(WDI)
```

まず、三つの例を見てみましょう。なにをしているかわかりますか。考えて見てください。

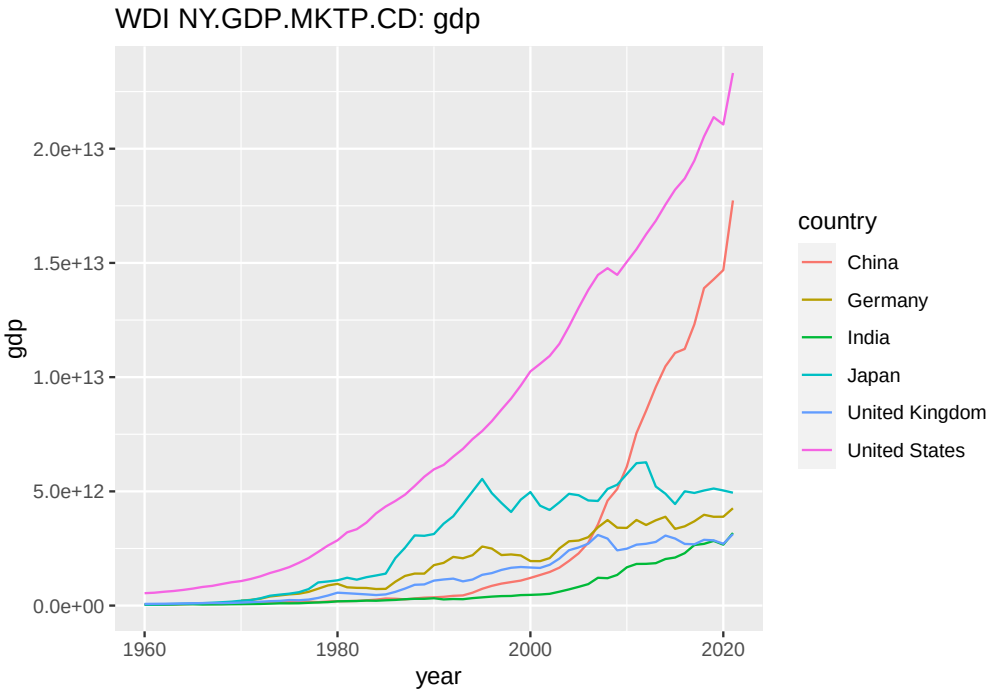
```
WDI(country = "all", indicator = c(gdp = "NY.GDP.MKTP.CD"),
    extra=TRUE) %>% drop_na(gdp) %>%
  filter(year==max(year), income !="Aggregates") %>%
  drop_na(region) %>% arrange(desc(gdp))
```

```
#> Rows: 16492 Columns: 13
#> -- Column specification --------------------------------------
#> Delimiter: ","
#> chr  (7): country, iso2c, iso3c, region, capital, income...
#> dbl  (4): year, gdp, longitude, latitude
#> lgl  (1): status
#> date (1): lastupdated
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
#> # A tibble: 196 x 13
#>    country       iso2c iso3c  year     gdp status lastupda~1
#>    <chr>         <chr> <chr> <dbl>   <dbl> <lgl>  <date>
#>  1 United States US    USA    2021 2.33e13 NA     2022-12-22
#>  2 China         CN    CHN    2021 1.77e13 NA     2022-12-22
```

```
#>  3 Japan          JP    JPN    2021 4.94e12 NA     2022-12-22
#>  4 Germany        DE    DEU    2021 4.26e12 NA     2022-12-22
#>  5 India          IN    IND    2021 3.18e12 NA     2022-12-22
#>  6 United Kingd~ GB    GBR    2021 3.13e12 NA     2022-12-22
#>  7 France         FR    FRA    2021 2.96e12 NA     2022-12-22
#>  8 Italy          IT    ITA    2021 2.11e12 NA     2022-12-22
#>  9 Canada         CA    CAN    2021 1.99e12 NA     2022-12-22
#> 10 Korea, Rep.    KR    KOR    2021 1.81e12 NA     2022-12-22
#> # ... with 186 more rows, 6 more variables: region <chr>,
#> #   capital <chr>, longitude <dbl>, latitude <dbl>,
#> #   income <chr>, lending <chr>, and abbreviated variable
#> #   name 1: lastupdated
```

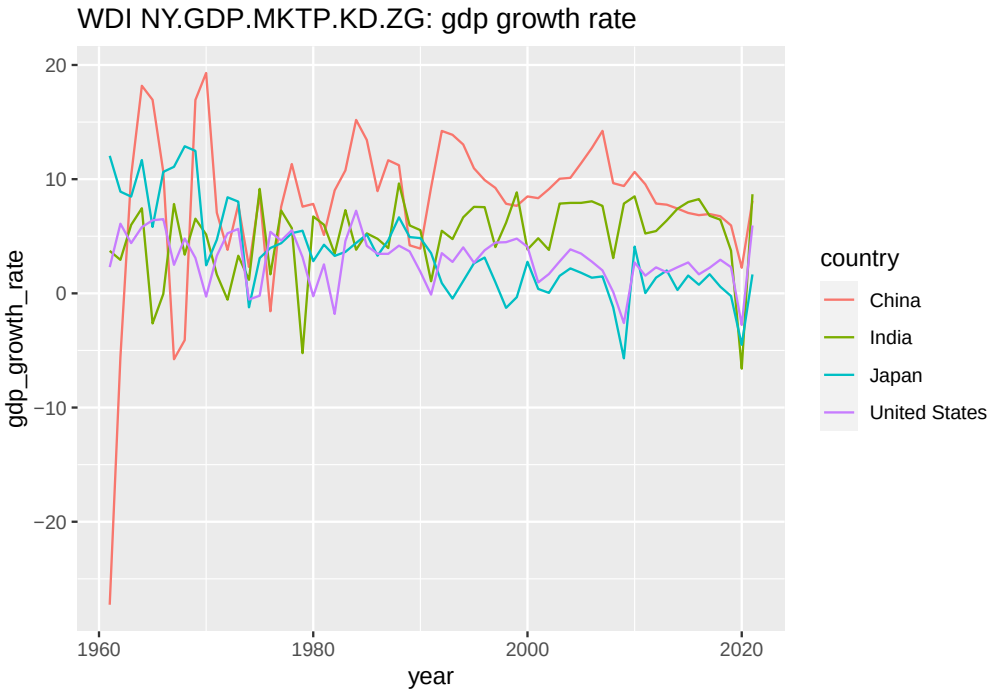
```
WDI(country = c("CN","GB","JP","IN","US","DE"), indicator = c(gdp = "NY.GDP.M
  ggplot(aes(year, gdp, col = country)) + geom_line() +
  labs(title = "WDI NY.GDP.MKTP.CD: gdp")
```

```
#> Rows: 372 Columns: 13
#> -- Column specification --------------------------------------
#> Delimiter: ","
#> chr  (7): country, iso2c, iso3c, region, capital, income...
#> dbl  (4): year, gdp, longitude, latitude
#> lgl  (1): status
#> date (1): lastupdated
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this m
```




```
WDI(country = c("CN","IN","JP","US"),
    indicator = c(gdp_growth_rate = "NY.GDP.MKTP.KD.ZG"), extra=TRUE) %>%
  drop_na(gdp_growth_rate) %>%
  ggplot(aes(year, gdp_growth_rate, col = country)) + geom_line() +
  labs(title = paste("WDI NY.GDP.MKTP.KD.ZG: gdp growth rate"))
```

```
#> Rows: 248 Columns: 13
#> -- Column specification --------------------------------------
#> Delimiter: ","
#> chr  (7): country, iso2c, iso3c, region, capital, income...
#> dbl  (4): year, gdp_growth_rate, longitude, latitude
#> lgl  (1): status
#> date (1): lastupdated
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```



まず、世界の国々の、GDP（gross domestic product 国内総生産）のデータを、取得して、2021 年の GDP を大きな順に並べています。

値は、たとえば、$2.331508e+13$ のように書かれていますが、これは、科学的記法と呼ばれるもので、2.331508×10^{13} を意味しています。約 23 兆ドルです。

次に、3 兆ドル以上の、6 カ国を選択し、その、iso2c と呼ばれるコードを使って、それらの国のデータをもう一度取得し、年次変化をあらわすグラフを描いています。

さらにその中から、4 カ国を選んで、今度は、GDP の年次変化率を描いています。単位は、パーセントです。

これは、ひとつの例ですが、ここで使われているのが、WDI World Development Indicator というもので、世界銀行が、いくつかの指標を定めて、編纂しているものです。

0.5.1.1 指標 Indicators (WDI)

上の例では、次の二つの指標のコード Indicator Code (WDI Code) が使われました。

- NY.GDP.MKTP.CD: GDP (current US$)
- NY.GDP.MKTP.KD.ZG: GDP growth (annual %)

0.5.1.2 指標 WDI (World Development Indicators)

The World Development Indicators is a compilation of relevant, high-quality, and internationally comparable statistics about global development and the fight against poverty. The database contains 1,400 time series indicators for 217 economies and more than 40 country groups, with data for many indicators going back more than 50 years.

WDI は、世界の開発状況と、貧困との戦いに関する、適切で上質、かつ、国際的に比較可能な時系列の統計データを編纂したものです。このデータベースは、217 の経済と 40 以上の国グループについて 1,400 の時系列指標を含み、指標のデータの多くは 50 年以上前に遡ることができます。

- 世界銀行（World Bank）: https://www.worldbank.org
- World Bank Open Data: https://data.worldbank.org
  - Country / Indicator > Featured & All > Details
- World Development Indicators (WDI) :
  - Themes: Poverty and Inequality, People, Environment, Economy, States and Markets, Global Links
  - Open Data & DataBank: Explore data, Query database

0.5.1.3 指標のコード、WDI code を探してみよう

いくつかの探し方があります。まず、ここでは、World Bank のサイトから探す方法を説明しましょう。

ふた通りあります。

1. World Bank Open Data にいくと、表題の下の検索窓の下に、Country / Indicator とありますから、Indicator を選択します。すると、そこに、項目のリストが、Featured と All という二つのタブに分かれて出ています。かなり膨大です。それを選択すると、その項目のサイトに行きます。それが、指標のサイトです。図などの、右上に、Details とありますから、それを選択すると、その中に、Indicator が書かれています。実は、指標のサイトのアドレス（URL）を見ると、そこにも、この Indicator が書かれていることがわかります。

2. World Development Indicators (WDI) にいくと、下のようなテーマに分かれています。

Themes: Poverty and Inequality, People, Environment, Economy, States and Markets, Global Links

その中から、選択して、スクロールすると、そこに、指標が書かれています。

Indicator, Code, Time coverage, Region coverage, Get data

とあり、Code が、指標のコードです。実は、すべての年や、すべての地域のデータが揃っているわけではないので、この情報を見ておくことはとても重要です。ほとんど、データがない場合もあります。

一番右端の Get data からは、CSV や、データバンク（Data Bank）へのリンクがあります。

それぞれの方法で、上で使った、二つの指標およびそのコードは見つかりましたか。

1 の方法の途中に出てきた、検索窓から検索することも可能です。

0.5.1.4 指標 WDI の例

このあとの、例で使う指標を書いておきます。

- NY.GDP.MKTP.CD: GDP (current US$)
- NY.GDP.DEFL.KD.ZG: Inflation, GDP deflator (annual %)
- SL.UEM.TOTL.NE.ZS: Unemployment, total (% of total labor force) (national estimate)
- CPTOTNSXN: CPI Price, nominal
- SL.TLF.CACT.MA.NE.ZS: Labor force participation rate, male (% of male population ages 15+) (national estimate)
- SL.TLF.CACT.FE.NE.ZS: Labor force participation rate, female (% of male population ages 15+) (national estimate)

0.5.1.5 練習 1. - 調べてみたい WDI 指標とそのコード

いくつか、リストしてみましょう。

0.5.2 WDI パッケージ

`WDI` パッケージの使い方を紹介します。

`WDI` パッケージで、データをダウンロードしたり、探したり、詳細情報を得たりできます。

0.5.2.1 指標 WDI 検索

まず、検索です。上で、サイトから調べる方法を紹介しましたが、`WDI` パッケージの、`WDIsearch` でも探すことができます。詳細は、右下の窓枠の Help タブの検索窓に、WDIsearch といれて調べて見てください。ここでは、二種類の検索方法を紹介します。