

# 統計ソフトRの概要

データ集計・作図・レポート作成

千葉大学予防医学センター: 江口 哲史

# Rとは？

オープンソースの統計解析向けプログラミング言語

Windows, Mac, LinuxどちらでもOK

パッケージで機能を拡張

高度な可視化や最新の統計手法を利用可

コードを書いて処理を行う必要がある

# Rとは？

コードを書いて処理を行う必要がある

どうしても慣れが必要・とつきにくさ

敷居を下げるためのツールも存在

記述した内容が残ることはメリットでもある

レポート・資料を作成するためのプラットフォームがある

# R環境のセットアップ

# R環境のセットアップ

Base R: download from CRAN



# R環境のセットアップ

Base R: download from CRAN



RStudio IDE: download from RStudio



# R環境のセットアップ

Base R: download from CRAN



RStudio IDE: download from RStudio



# R起動

```
R version 4.1.2 (2021-11-01) -- "Bird Hippie"
Copyright (C) 2021 The R Foundation for Statistical Computing
Platform: x86_64-apple-darwin17.0 (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。

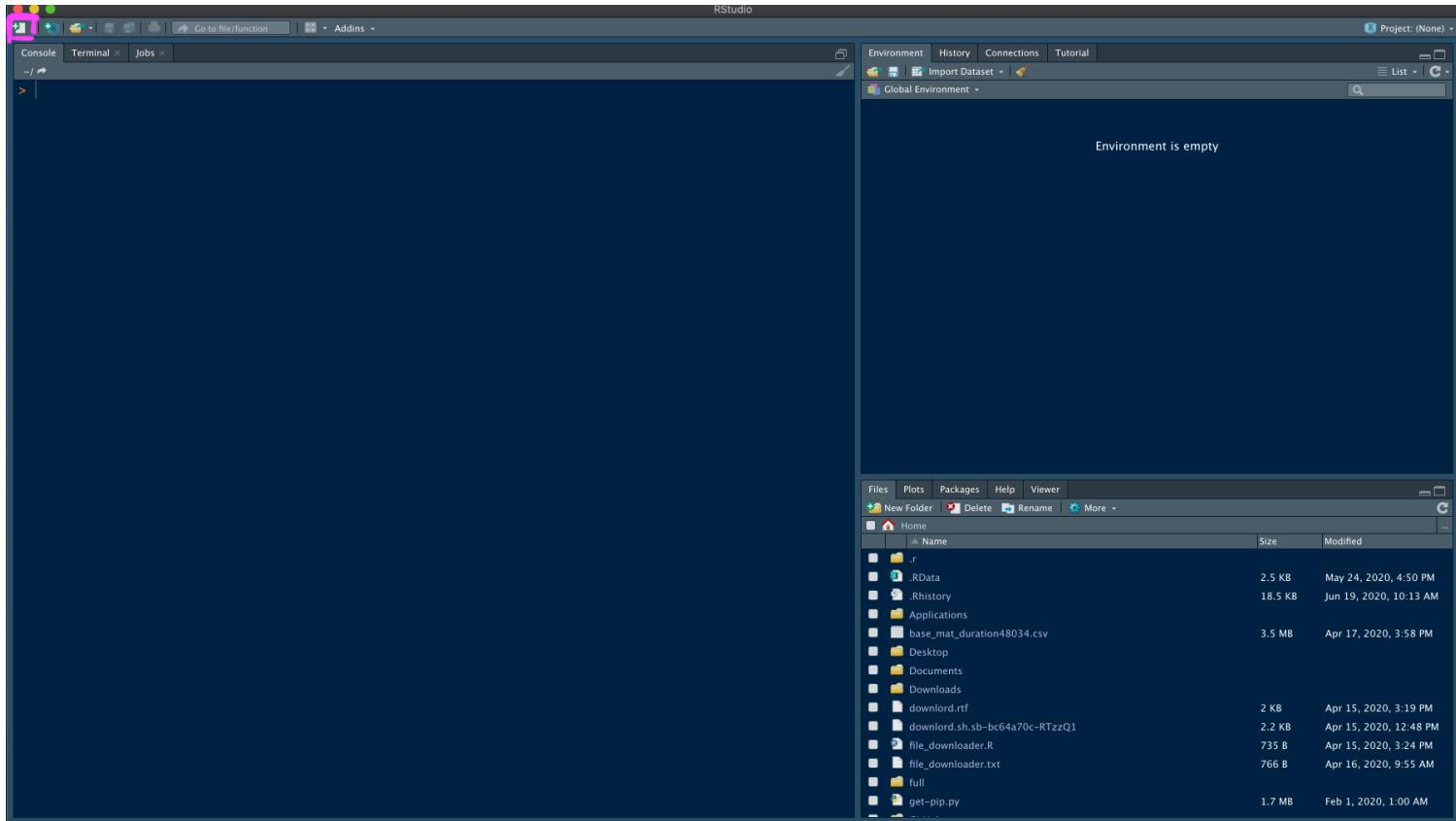
[R.app GUI 1.77 (8007) x86_64-apple-darwin17.0]

[ワークスペースが次のファイルから読み込まれました /Users/siero5335/.RData]
[履歴が次のファイルから読み込まれました /Users/siero5335/.Rapp.history]

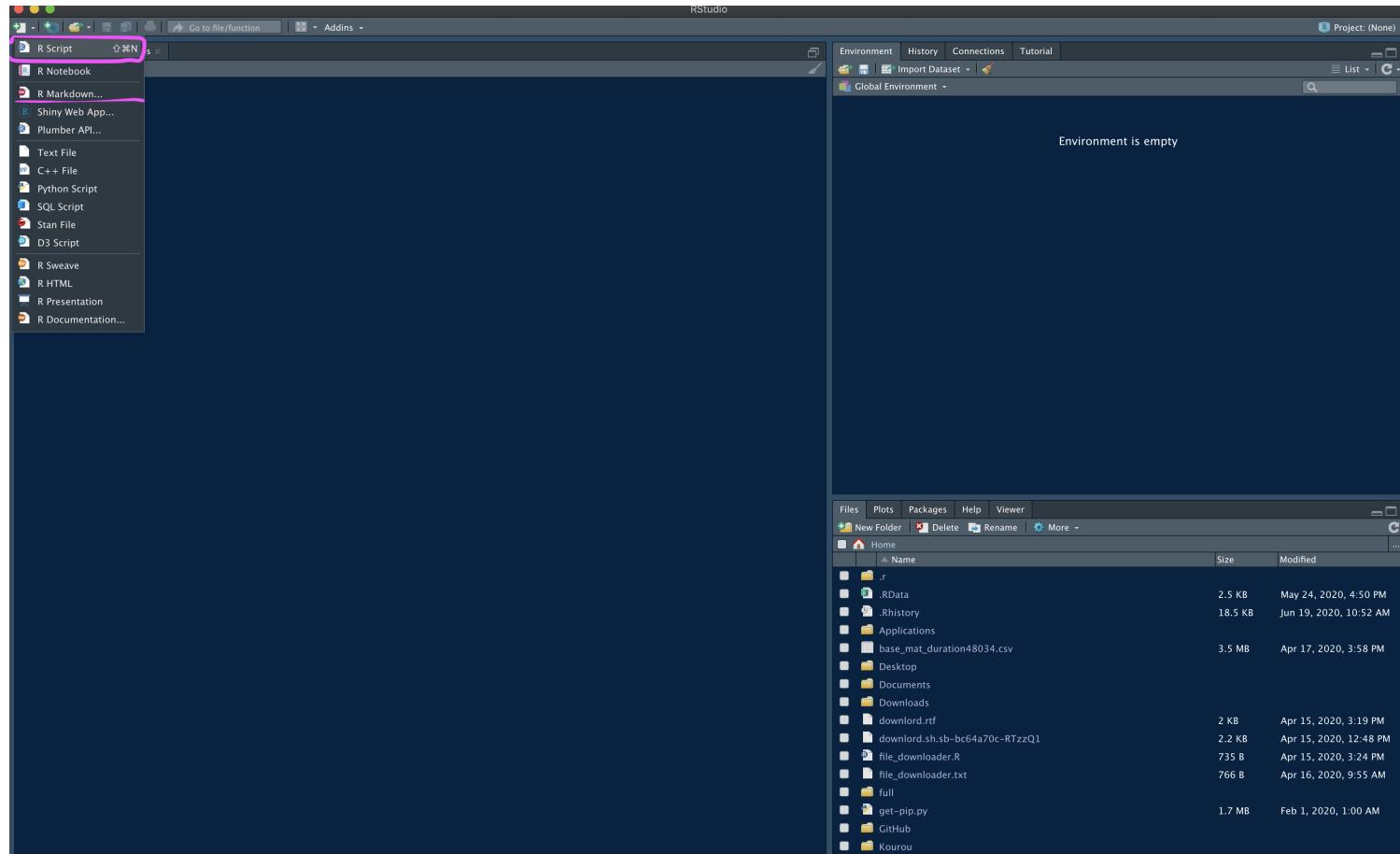
>
```

簡素なUI・書いたコードの再利用が難しい  
→ RStudioの利用を推奨

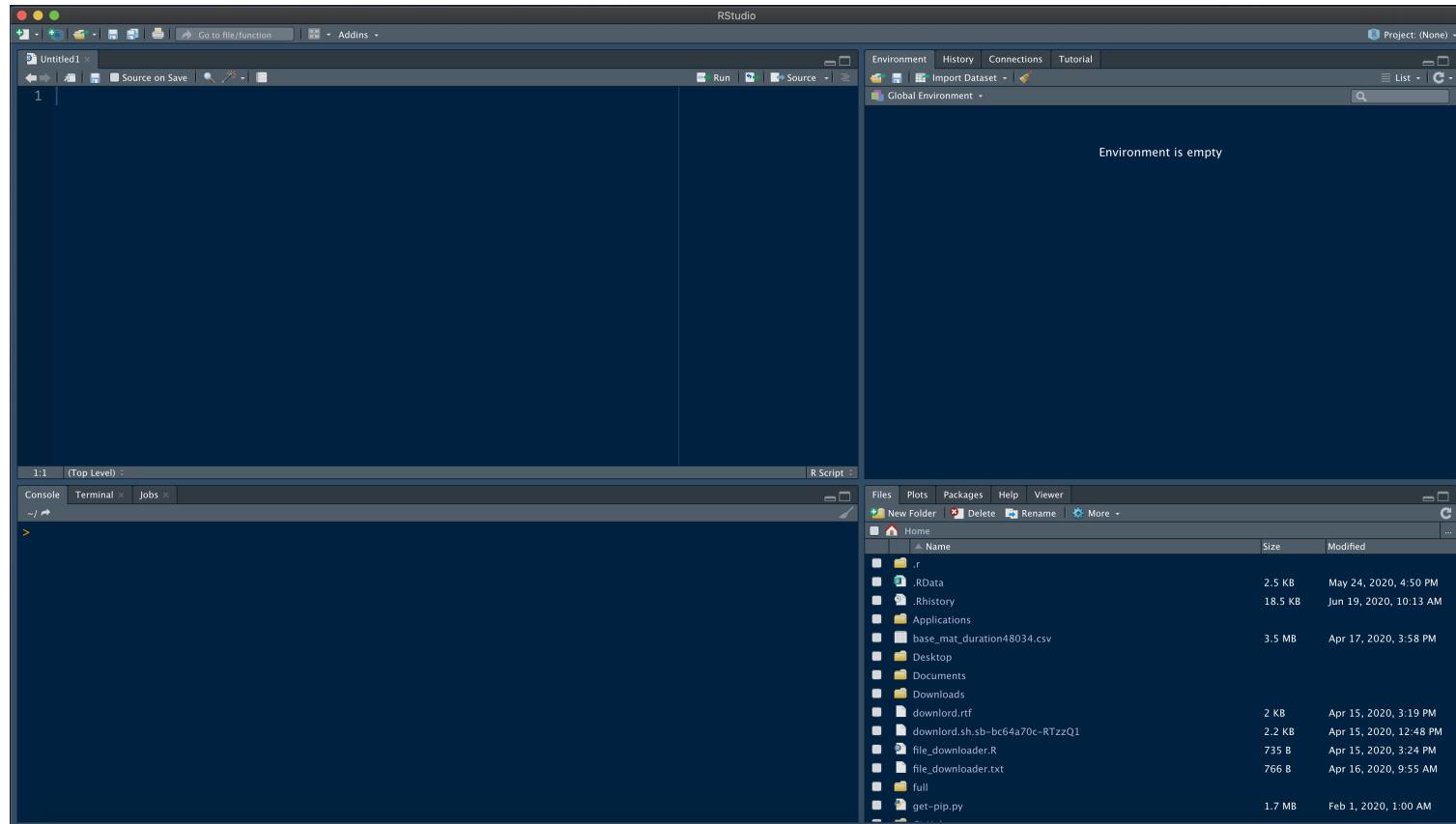
# RStudio起動



# RStudio起動



# RStudio起動



# パッケージ・データ読み込み

# パッケージの導入

[tidyverse](#) のパッケージの導入

CRAN:

```
install.packages("tidyverse")
install.packages("knitr")
install.packages("rmarkdown")
install.packages("GGally")
```

# パッケージの導入

[tidyverse](#) のパッケージの導入

CRAN:

```
install.packages("tidyverse")
install.packages("knitr")
install.packages("rmarkdown")
install.packages("GGally")
```

Rのパッケージはデータの集計や可視化・解析手法などの機能を追加するための外部ツール。[install.packages\(\)](#) 関数を使うことで CRAN に登録されているパッケージをインストールできる(2022/06現在18500種以上)。

# パッケージの読み込み

```
library("tidyverse"); library("GGally")
```

インストールしたパッケージを読み込む際には `library()` 関数を利用する。インストールはアップデートされたバージョンを使わないのであれば一度で大丈夫だが、パッケージの読み込みはRを立ち上げるたびに行う必要がある。

この `tidyverse` パッケージはデータの高速な読み込み・加工・集計・可視化などに関わるパッケージ群を一括で使えるようにするためのパッケージ。

`GGally` パッケージは後の可視化で少し紹介。

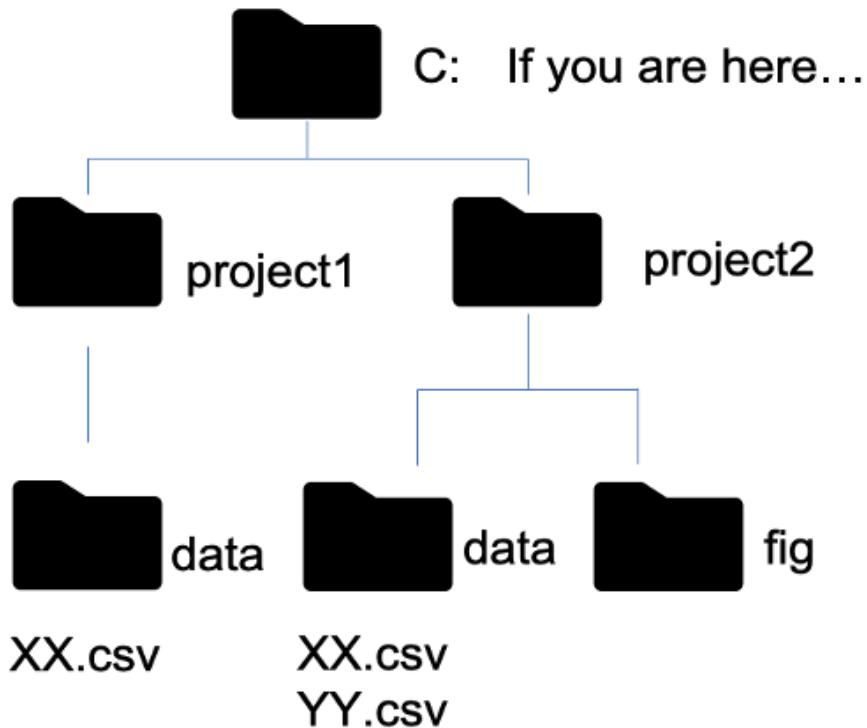
# データの読み込み

```
df <- read_csv("data/demo_data.csv")
```

上記のような記述でcsvファイルを読み込むことができる。  
エクセルの場合はreadxl パッケージのread\_xlsx() 関数を使う。

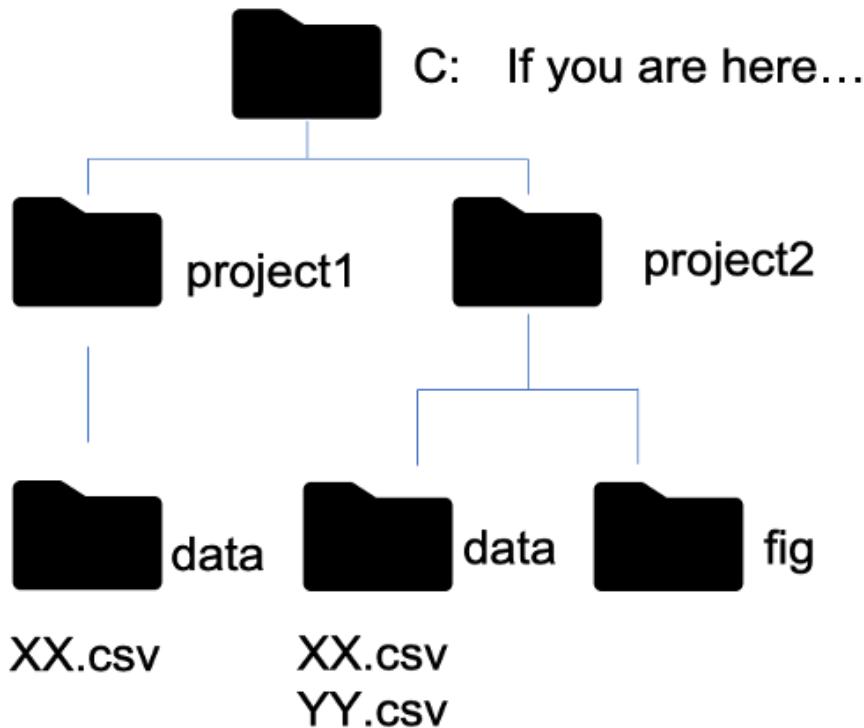
パス (コンピュータ内の住所) の概念を知らないとうまく読めない。

# パス



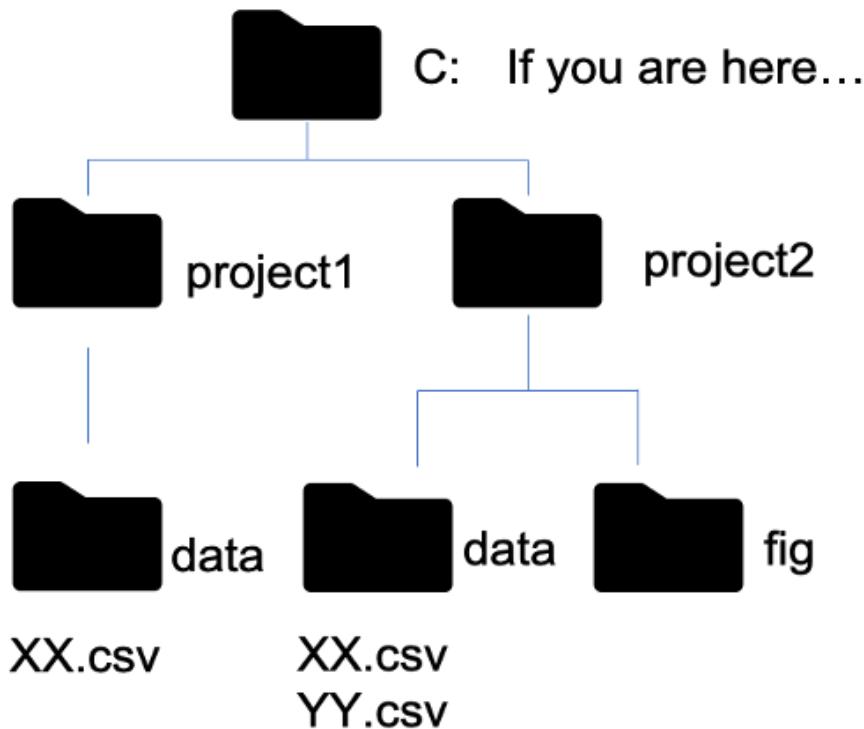
`getwd()` 関数で自分のRがどこの住所を参考にしているか調べることができる。

# パス



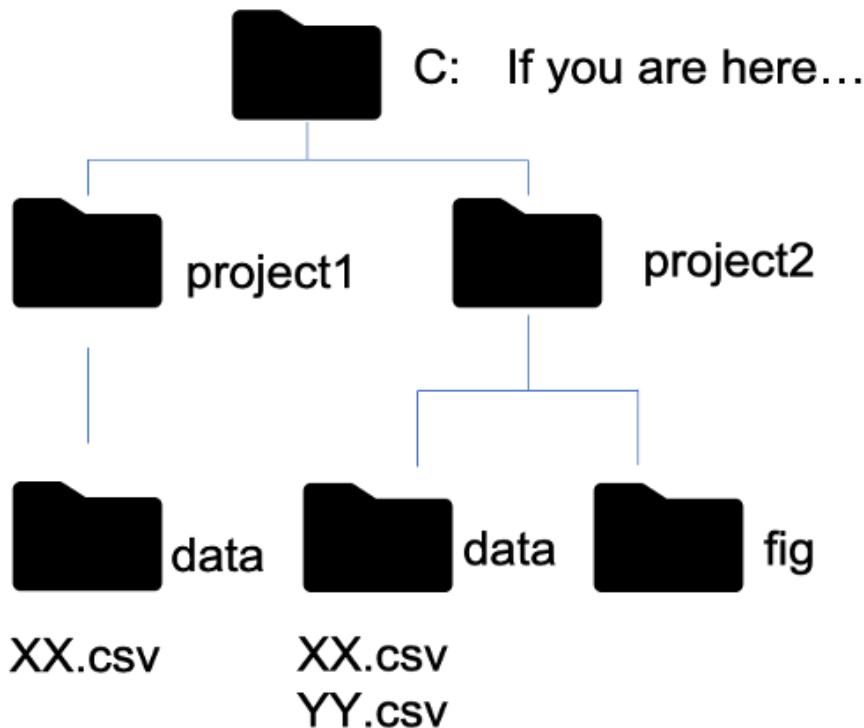
```
df <- read_csv("XX.csv") # NG
```

# パス



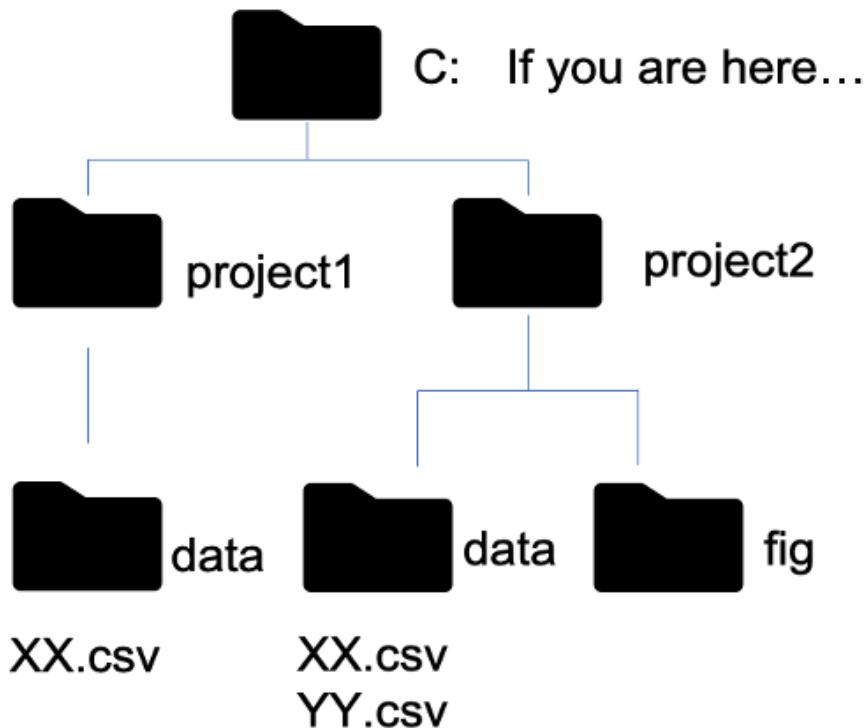
```
df <- read_csv("project1/XX.csv") # NG
```

# パス



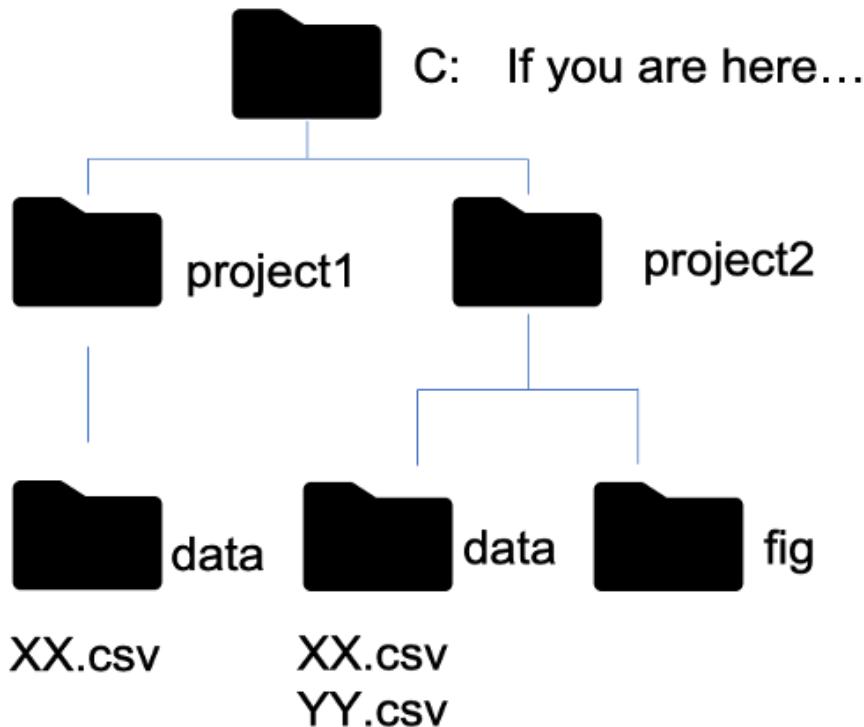
```
df <- read_csv("C:/project1/data/XX.csv") # OK
```

# パス



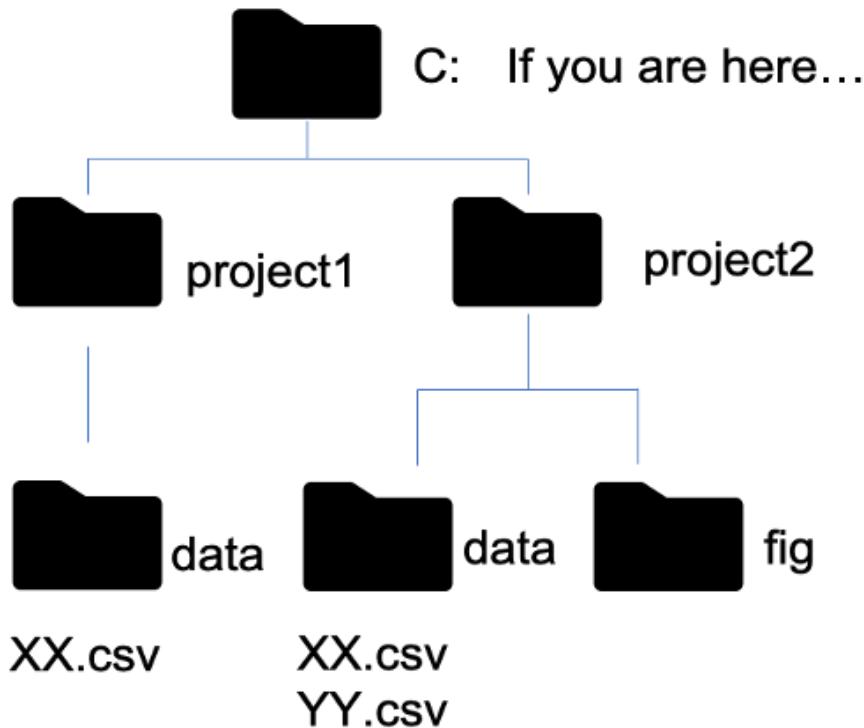
```
df <- read_csv("C:/project1/data/YY.csv") # NG
```

# パス



```
df <- read_csv("C:/project2/data/XX.csv") # Other file
```

# パス



```
df <- read_csv("C:/project2/data/YY.csv") # OK
```

# パス

正確にパスを書かないといけないのでタイプ等があるだけで読みなくなってしまう

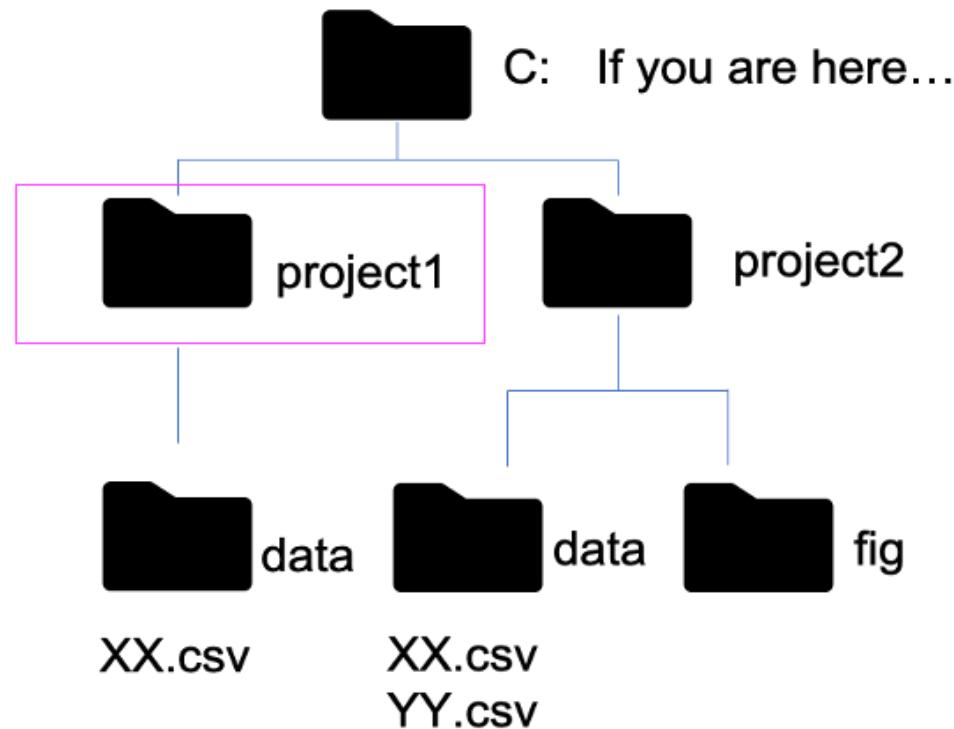
# パス

正確にパスを書かないといけないのでタイプ等があるだけで読みなくなってしまう

```
setwd("C:¥/project1/")
df <- read_csv("data/XX.csv") # OK
```

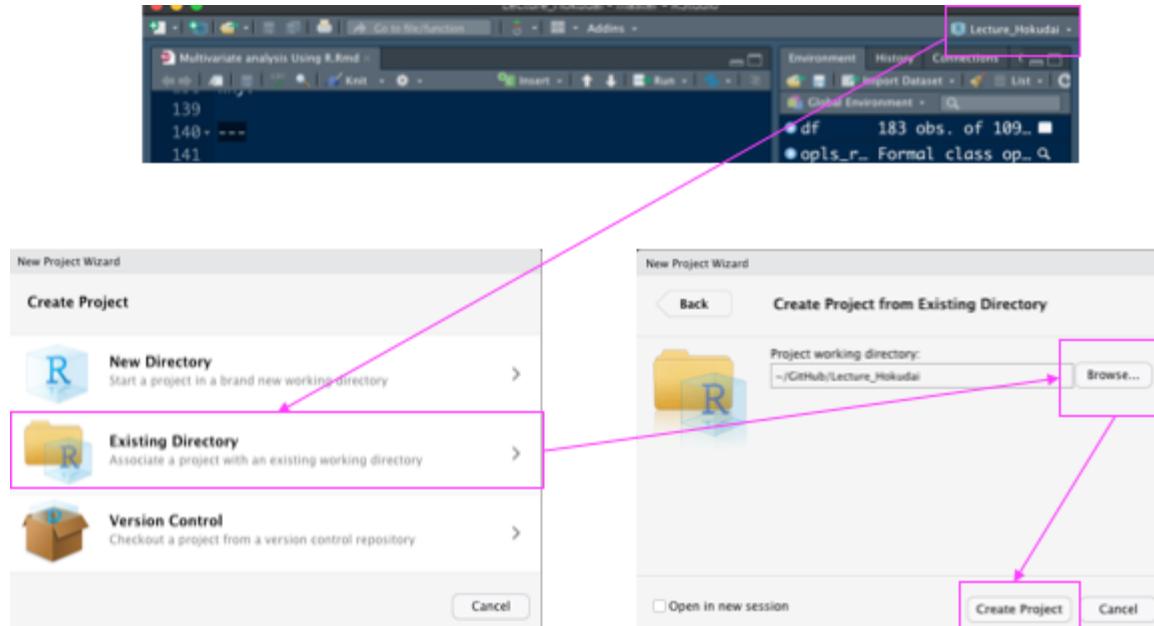
`setwd()`関数で自分のRが参考にしている住所を指定できる

# パス



`setwd()` 関数で自分のRが参考にしている住所を指定できる

# プロジェクト



プロジェクトを作ることで特定のフォルダを参照フォルダにできる

# プロジェクト

```
df <- read_csv("data/demo_data.csv") # OK
```

一番シンプルな記述で読み込みができるように

## データ確認・加工

# データの確認

```
head(df)
```

```
## # A tibble: 6 × 6
##   `Sample ID`  Age Gender Height Weight outcome
##       <dbl> <dbl> <chr>    <dbl>   <dbl>    <dbl>
## 1         1     36 Female    162      52      18
## 2         2     13 Female    160      43       8
## 3         3     20 Female    153      46      37
## 4         4     24 Male     167      54      57
## 5         5     22 Female    153      43      14
## 6         6     48 Male     168      60      35
```

ID含めて変数は7個、性別以外は数値

# データの確認

```
summary(df)
```

```
##      Sample ID          Age         Gender        Height
## Min.   : 1.00   Min.   :10.00   Length:80    Min.   :129.0
## 1st Qu.:20.75  1st Qu.:25.75  Class  :character  1st Qu.:153.0
## Median :40.50  Median :37.00  Mode   :character  Median :158.5
## Mean   :40.50  Mean   :35.60                    Mean   :158.6
## 3rd Qu.:60.25  3rd Qu.:46.00                    3rd Qu.:165.2
## Max.   :80.00  Max.   :60.00                    Max.   :177.0
## 
##      Weight        outcome
## Min.   :30.00   Min.   :  7.20
## 1st Qu.:44.75  1st Qu.: 37.75
## Median :50.00  Median : 59.00
## Mean   :52.08  Mean   : 71.27
## 3rd Qu.:56.25  3rd Qu.: 93.25
## Max.   :80.00  Max.   :360.00
```

min, max, mean, median, 1st, 3rd quartilesを [summary\(\)](#) 関数1つで確認できる

# データの集計

`tidyverse`に含まれる`dplyr`はtableデータの操作に適したパッケージ

データの加工・集計の際に便利

内部で読み込まれる`magrittr`パッケージの機能でパイプ`%>%`を使ってパイプの左側に記述したデータをパイプの右側の処理に受け渡す

連続してパイプをつないでまとめていろいろな処理ができる

# dplyrの主な関数

`select()`: 特定の列を選択

`filter()`: 特定の行を選択

`group_by()`: 特定の変数に基づいてグループを作る

`summarise()`: グループ化したデータに基づいて集計する

`mutate()`: 新しい変数を作る

`xx_join()`: 共通する変数に基づいて2つのデータを結合する (今回は解説せず)

# select(): 列の選択

```
df %>% select(Age, Gender, outcome)
```

```
## # A tibble: 80 × 3
##       Age Gender outcome
##   <dbl> <chr>    <dbl>
## 1     36 Female     18
## 2     13 Female      8
## 3     20 Female     37
## 4     24 Male       57
## 5     22 Female     14
## 6     48 Male       35
## 7     46 Female     88
## 8     49 Male      100
## 9     26 Female     37
## 10    50 Female     7.2
## # ... with 70 more rows
```

目的の列のみを `select()` で抽出

# select(): 列の選択

```
df %>% select(!c(Age, Gender, outcome))
```

```
## # A tibble: 80 × 3
##   `Sample ID` Height Weight
##       <dbl>    <dbl>   <dbl>
## 1 1           162     52
## 2 2           160     43
## 3 3           153     46
## 4 4           167     54
## 5 5           153     43
## 6 6           168     60
## 7 7           153     44
## 8 8           157     44
## 9 9           159     48
## 10 10          153     42
## # ... with 70 more rows
```

!をつけると選択した列以外を抽出できる

# filter(): 行の選択

```
df %>% filter(Gender == "Female")  
  
## # A tibble: 45 × 6  
##   `Sample ID`  Age Gender Height Weight outcome  
##       <dbl> <dbl> <chr>    <dbl>   <dbl>    <dbl>  
## 1           1     36 Female    162      52     18  
## 2           2     13 Female    160      43      8  
## 3           3     20 Female    153      46     37  
## 4           5     22 Female    153      43     14  
## 5           7     46 Female    153      44     88  
## 6           9     26 Female    159      48     37  
## 7          10     50 Female    153      42     7.2  
## 8          11     30 Female    154      43     16  
## 9          13     45 Female    160      50     28  
## 10         14     24 Female    168      58    110  
## # ... with 35 more rows
```

`filter()` では行の抽出が可能

上記では性別 (Gender) がFemaleの行のみを抽出  
文字列の場合は""で囲んで文字列として取り扱う

# filter(): 行の選択

```
head( df %>% filter(Height > 150 & Gender == "Female") )
```

組み合わせも可能 (身長150cm以上の女性)

# group\_by(): グルーピング

主にデータの集計に用いられる関数

関数の()内にグルーピングに使いたい変数を記入する

```
head( df %>% group_by(Gender) )
```

```
## # A tibble: 6 × 6
## # Groups:   Gender [2]
##   `Sample ID`  Age Gender Height Weight outcome
##       <dbl> <dbl> <chr>    <dbl>   <dbl>    <dbl>
## 1         1    36 Female    162     52      18
## 2         2    13 Female    160     43       8
## 3         3    20 Female    153     46      37
## 4         4    24 Male     167     54      57
## 5         5    22 Female    153     43      14
## 6         6    48 Male     168     60      35
```

単にgroup\_by()関数に渡すだけではデータの見た目は変わらない

# summarise(): グルーピングしたデータの集計

```
df %>% group_by(Gender) %>%  
  summarise(N = n(), H = mean(Height), W = max(Weight))
```

```
## # A tibble: 2 × 4  
##   Gender     N      H      W  
##   <chr>   <int>  <dbl>  <dbl>  
## 1 Female     45    155.     66  
## 2 Male       35    164.     80
```

`summarise()`は名の通りグルーピングしたデータを集計するための関数  
`n()`, `mean()`, `sum()`, `sd()`などの関数を使って、  
`group_by()`関数で指定した変数で集計値を計算できる

# mutate(): 変数の追加

```
head( df %>% mutate(BMI = Weight / (Height/100)^2,
                      BMI = round(BMI, digits = 1)) )
```

```
## # A tibble: 6 × 7
##   `Sample ID`  Age Gender Height Weight outcome    BMI
##       <dbl> <dbl> <chr>    <dbl>   <dbl>    <dbl>   <dbl>
## 1         1     36 Female    162      52     18    19.8
## 2         2     13 Female    160      43      8    16.8
## 3         3     20 Female    153      46     37    19.7
## 4         4     24 Male     167      54     57    19.4
## 5         5     22 Female    153      43     14    18.4
## 6         6     48 Male     168      60     35    21.3
```

`mutate()` 関数は新しい変数を追加するための関数

データの組み合わせて様々な加工も可能

上記では身長と体重からBMIをつくっている

# 作図

# 作図

`tidyverse`に含まれる`ggplot2`パッケージが広く利用されている

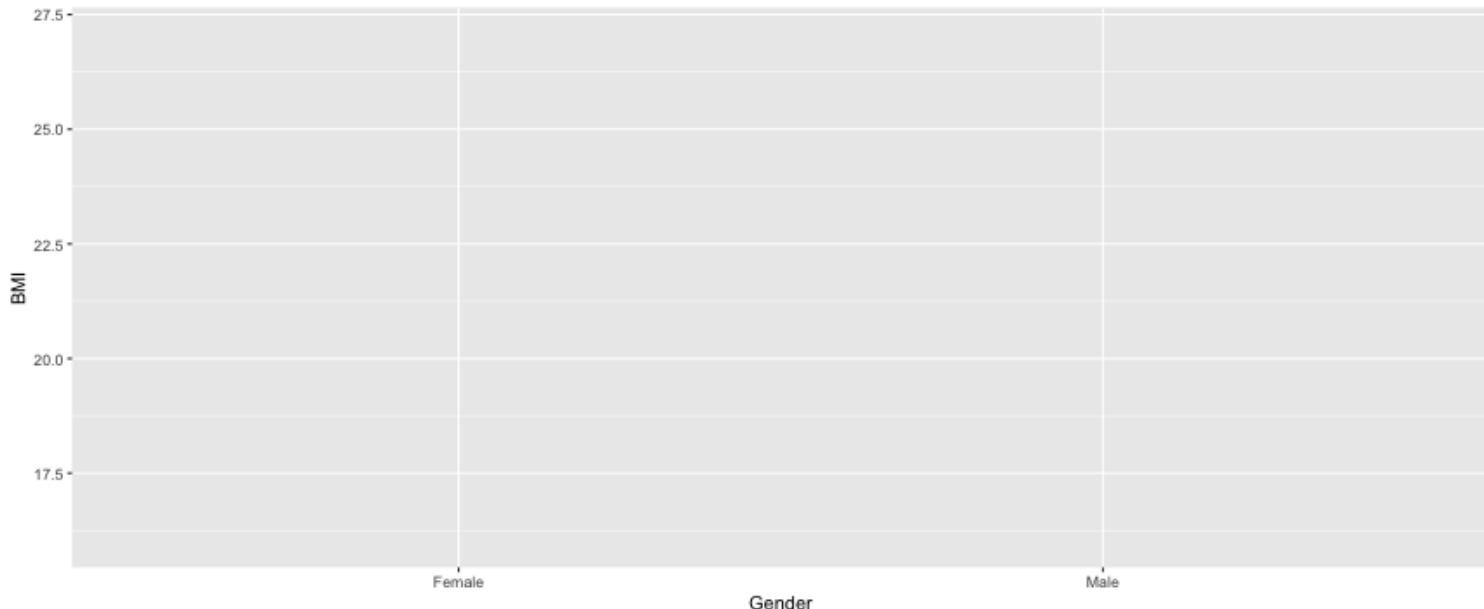
作ることができる図の種類が多い

レイヤー構造になっており柔軟性が高い

きれいな図を作ることができる（好みはあるが…）

# ggplot2

```
ggplot(df, aes(x = Gender, y = BMI))
```

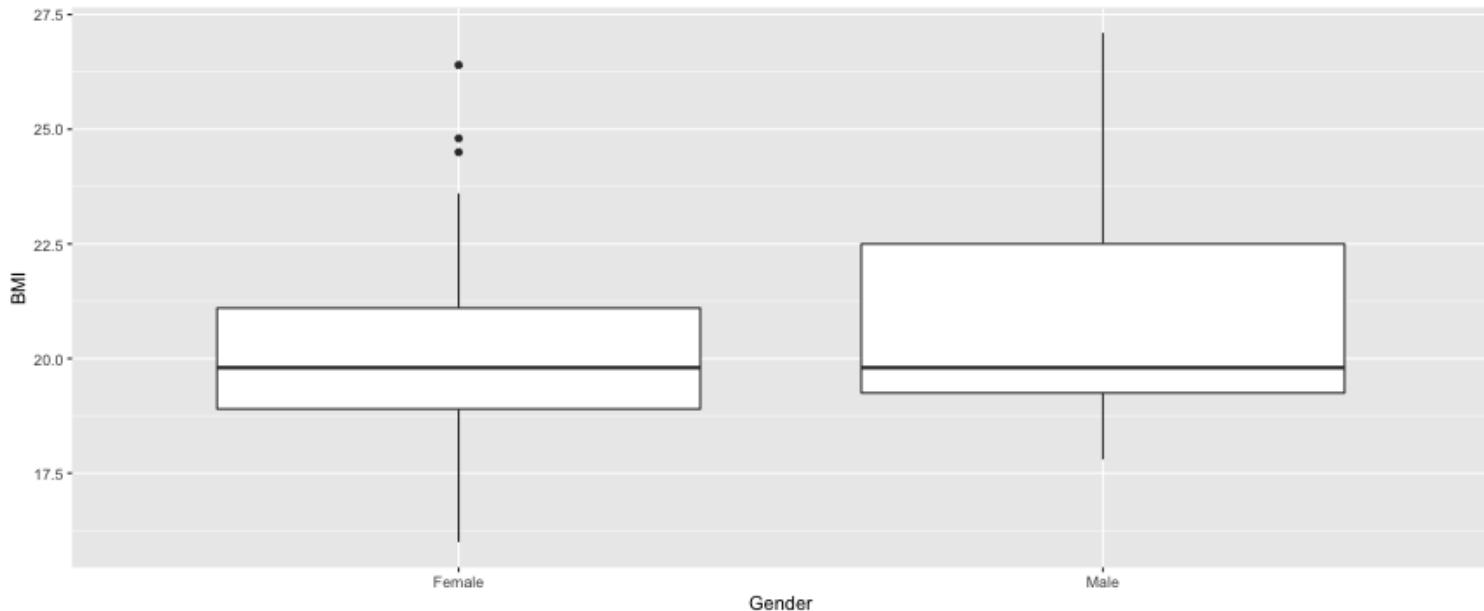


レイヤーの1段目

`ggplot()` 関数内にデータを指定し、`aes()` 関数で x, y 軸などを指定しておく x, y 軸がそれぞれ指定されるが、出力される図は外側だけ

# ggplot2

```
ggplot(df, aes(x = Gender, y = BMI)) + geom_boxplot()
```



`geom_XX()`関数をつなげ、作りたい出力レイヤーを指定する

# 代表的なgeom\_XX

`geom_boxplot()`: 箱ひげ図をつくる

`geom_point()`: 散布図をつくる

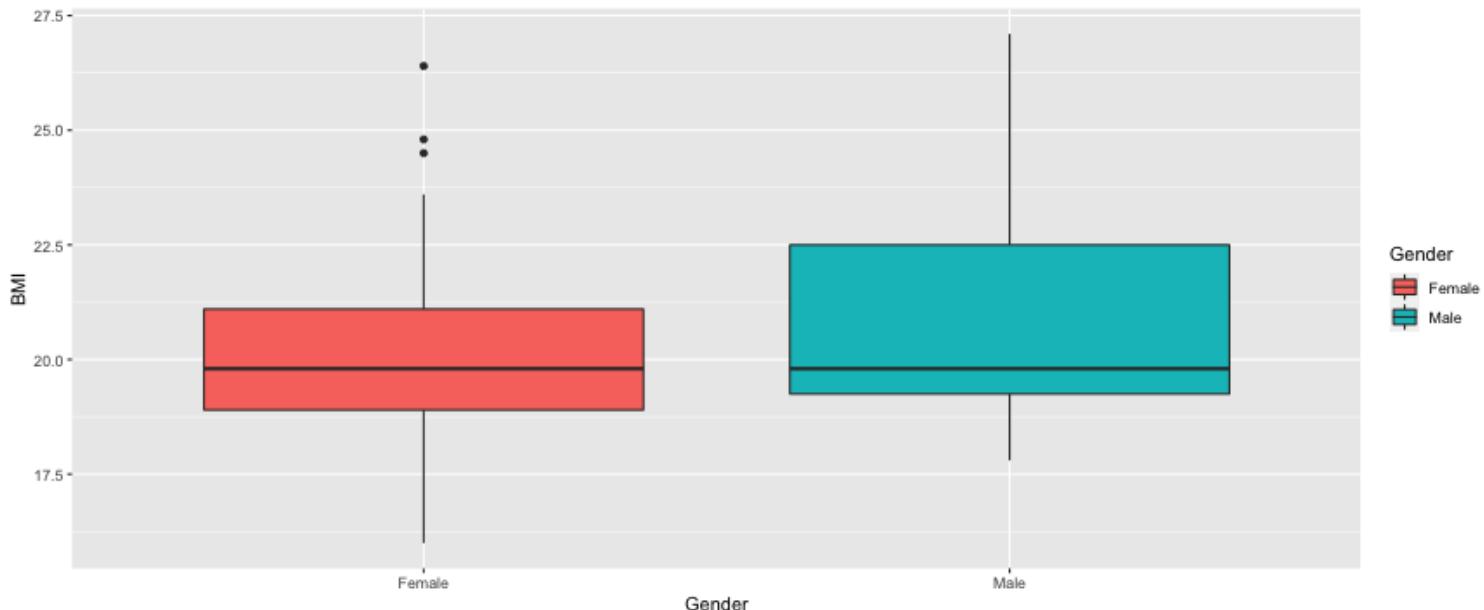
`geom_smooth()`: 回帰直線を引く

`geom_histogram()`: ヒストグラムをつくる

`geom_bar()`: 棒グラフをつくる

# geom\_boxplot()

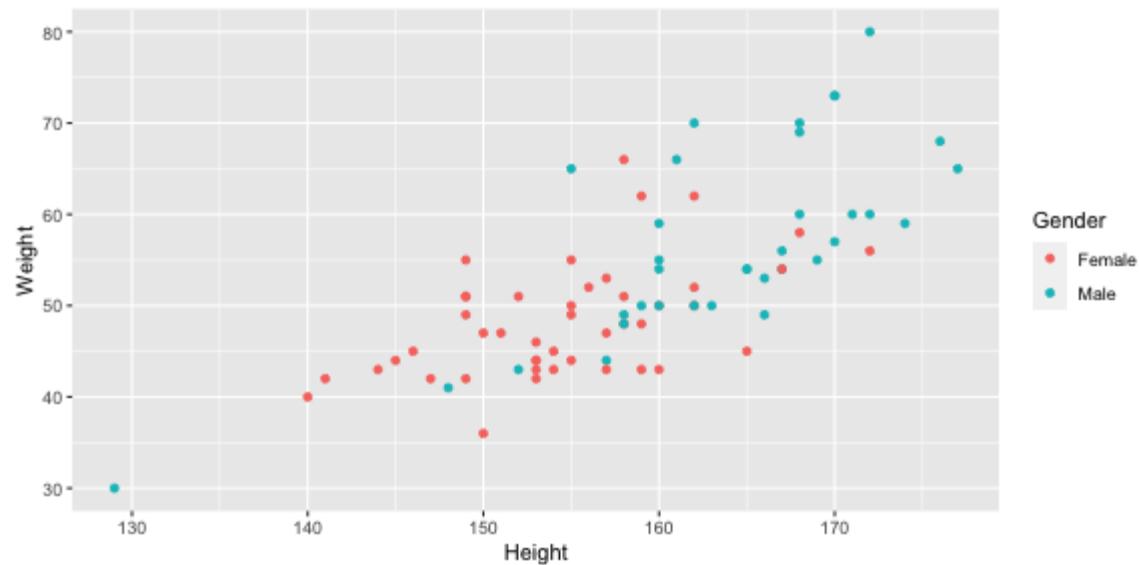
```
ggplot(df, aes(x = Gender, y = BMI, fill = Gender)) + geom_boxplot()
```



`fill`, `color`引数に変数を指定することで色分けも可能

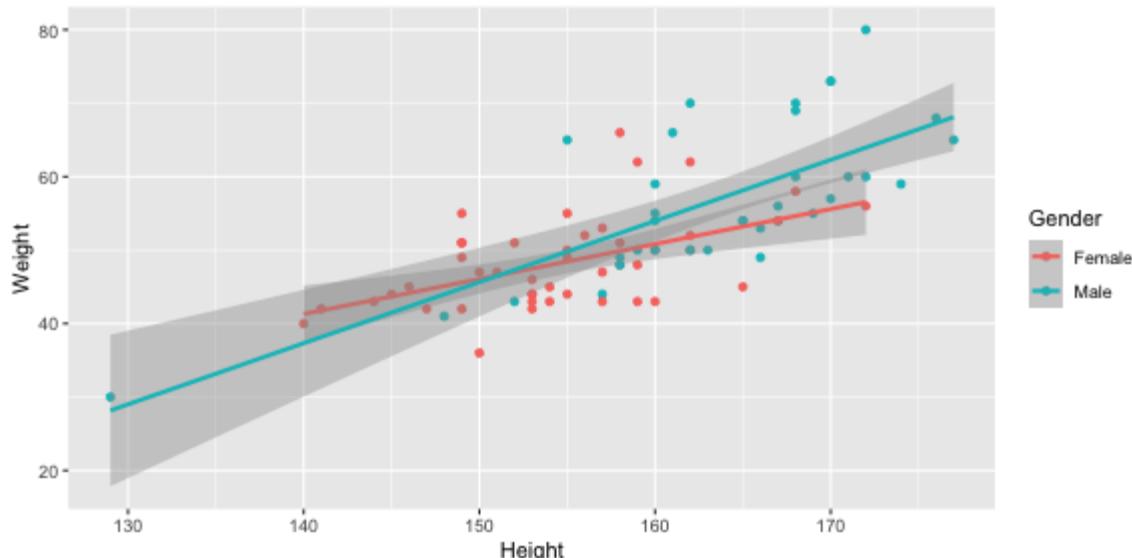
# 散布図

```
ggplot(df, aes(x = Height, y = Weight, color = Gender)) +  
  geom_point()
```



# 散布図

```
ggplot(df, aes(x = Height, y = Weight, color = Gender)) +  
  geom_point() +  
  geom_smooth(method = "lm")
```

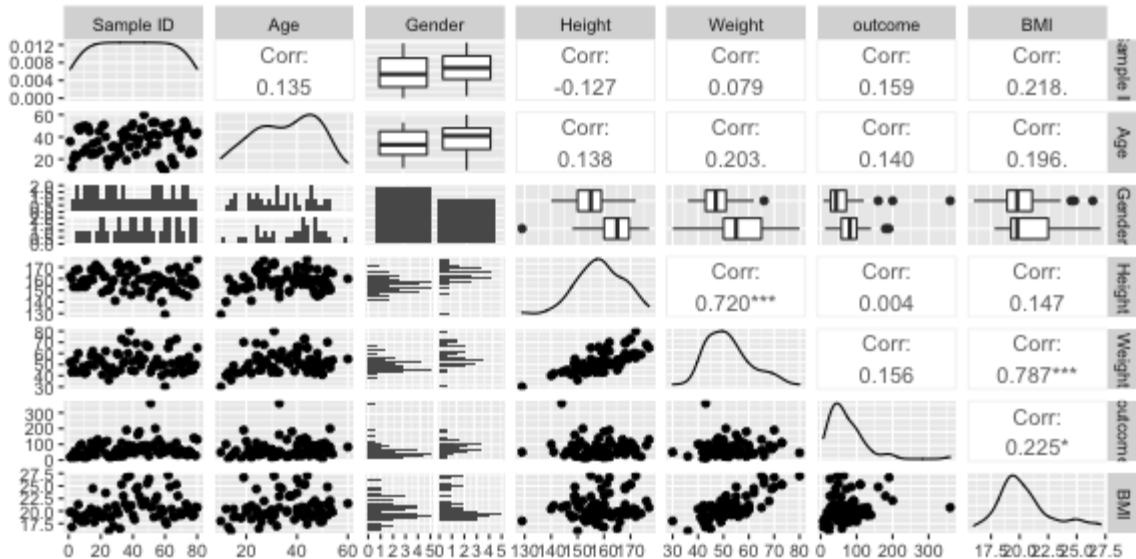


geom\_smooth()で

先程の図にレイヤーを追加する形で回帰直線を追加

# 散布図行列

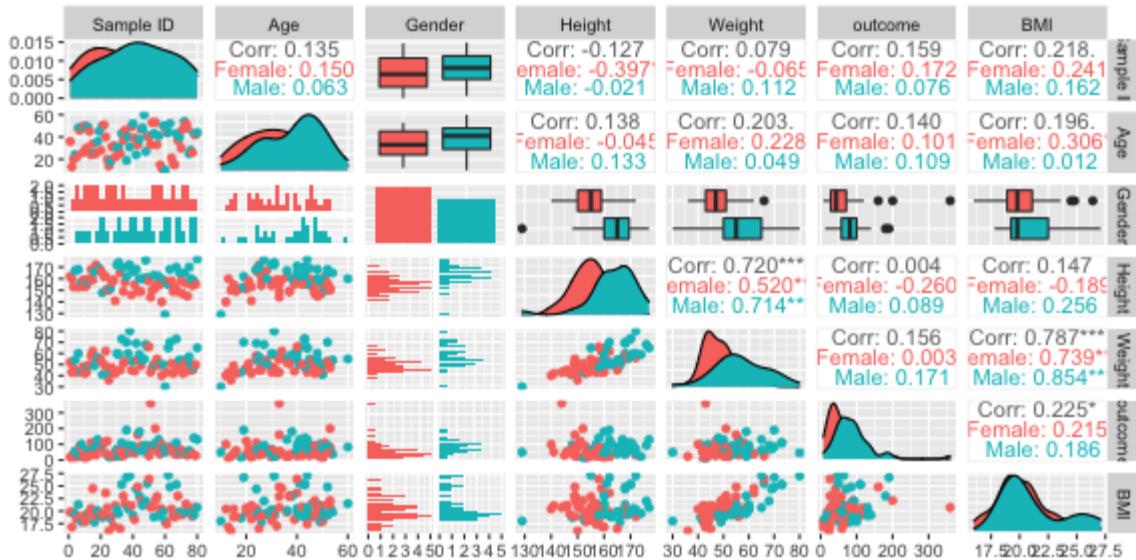
```
ggpairs(df)
```



[GGally](#)パッケージを利用することで相関行列を一気に作ることもできる

# 散布図行列

```
ggpairs(df, aes(color = Gender))
```



`aes()` 関数に変数を渡して色分けした図にもできる  
そのまま論文の図にするにはうるさいかもしれないが探索に有効

# レポートティング

# RMarkdown

Markdownの派生

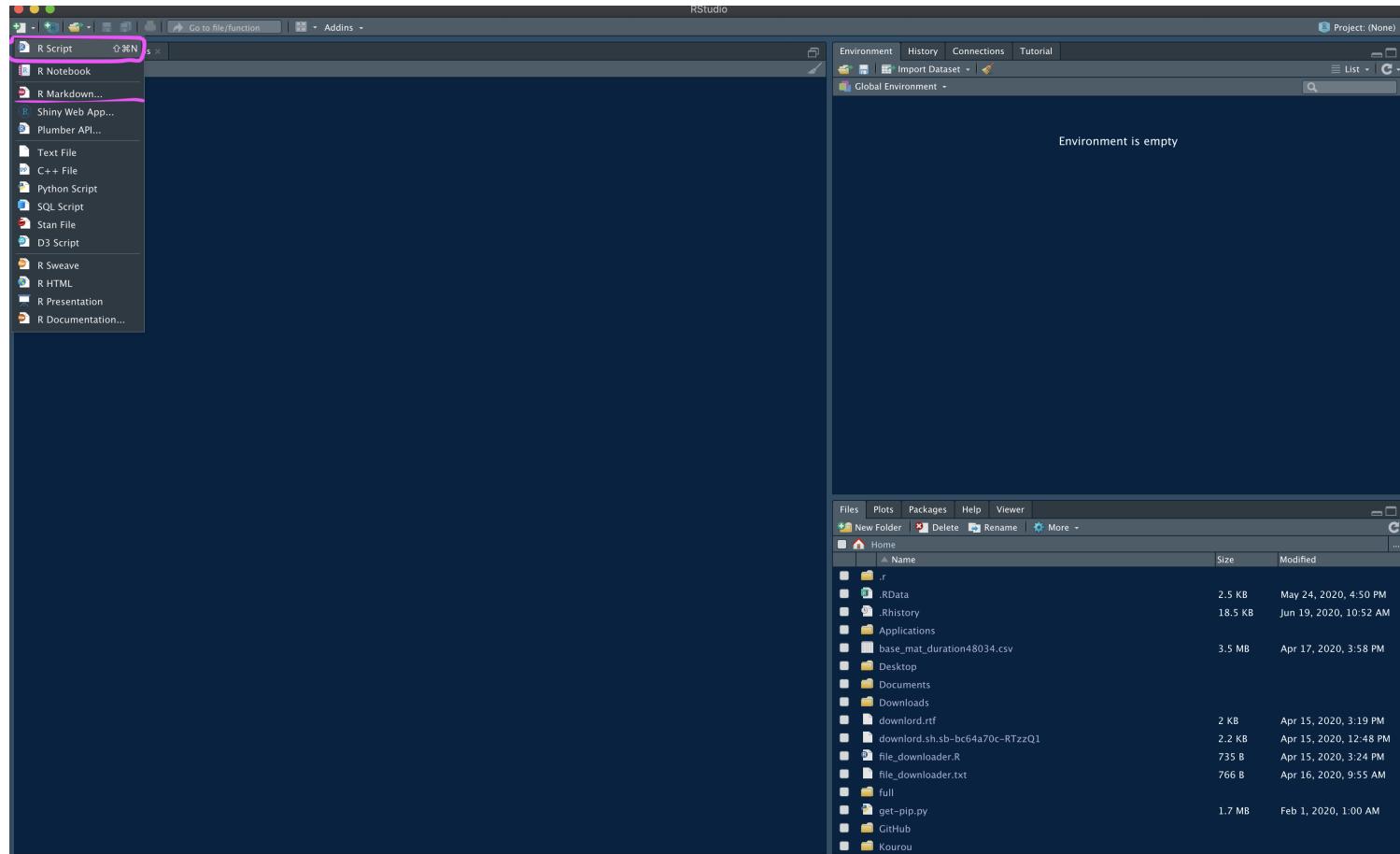
文章とコードをひとまとめにして出力するための機能

html, word, PDF, pptxなど、様々な形式に出力可能

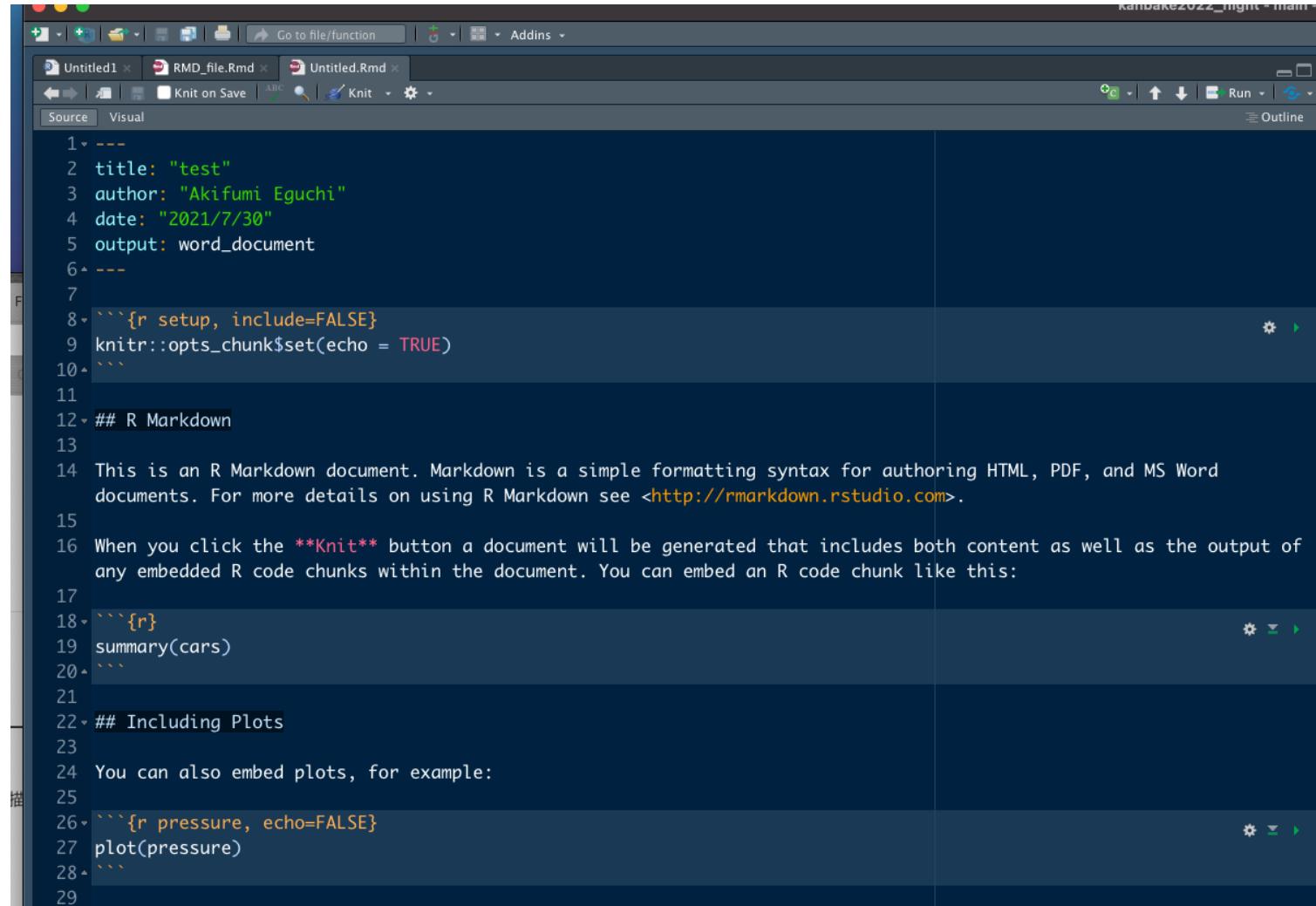
データ・乱数・パッケージなどが固定されていればいつも同じ出力

sessionInfo()関数で現在のR、パッケージのバージョン情報を出力可

# RMarkdown



# RMarkdown



The screenshot shows the RStudio interface with an R Markdown file open. The title bar indicates the session is named "Kanbake2022\_night - main". The left sidebar shows three tabs: "Untitled1", "RMD\_file.Rmd", and "Untitled.Rmd". The main pane displays the R Markdown code:

```
1 ---  
2 title: "test"  
3 author: "Akifumi Eguchi"  
4 date: "2021/7/30"  
5 output: word_document  
6 ---  
7  
8 ```{r setup, include=FALSE}  
9 knitr::opts_chunk$set(echo = TRUE)  
10```  
11  
12## R Markdown  
13  
14 This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.  
15  
16 When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:  
17  
18```{r}  
19 summary(cars)  
20```  
21  
22## Including Plots  
23  
24 You can also embed plots, for example:  
25  
26```{r pressure, echo=FALSE}  
27 plot(pressure)  
28```  
29
```

# RMarkdown

test<sup>1</sup>

Akifumi Eguchi<sup>2</sup>

2021/7/30<sup>3</sup>

• R Markdown<sup>4</sup>

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.<sup>5</sup>

When you click the Knit button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:<sup>6</sup>

```
summary(cars)7
```

```
## #> 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
```

• Including Plots<sup>8</sup>

You can also embed plots, for example:<sup>9</sup>

pressure

temperature

# RMarkdown

bibtexと連携して引用文献をつけることもできる

一部論文誌フォーマットも用意されている

メモ帳代わりにでもコードと結果・考察をまとめておくことは助けになる

本スライドもRMarkdownおよびxaringanパッケージを使って作ったもの

コードはアップロード予定

RMarkdownでも可能だが、開発中のQuartoを利用することでR, Python, Juliaなど、複数のデータ解析に関わる言語にまたがった処理も可能になりつつある

# 参考資料

RユーザのためのRStudio "実践" 入門

Rが生産性を高める

データ分析のためのデータ可視化入門

再現可能性のすゝめ

自然科学研究のためのR入門

# 実行環境

```
sessioninfo::session_info()

## - Session info -
## setting value
## version R version 4.1.2 (2021-11-01)
## os      macOS Big Sur 10.16
## system x86_64, darwin17.0
## ui      X11
## language (EN)
## collate ja_JP.UTF-8
## ctype   ja_JP.UTF-8
## tz      Asia/Tokyo
## date    2022-06-03
## pandoc  2.17.1.1 @ /Applications/RStudio.app/Contents/MacOS/quarto/bin/ (via rmarkdown)
##
## - Packages -
## package * version date (UTC) lib source
## assertthat 0.2.1 2019-03-21 [1] CRAN (R 4.1.0)
## backports  1.4.1 2021-12-13 [1] CRAN (R 4.1.0)
## bit        4.0.4 2020-08-04 [1] CRAN (R 4.1.0)
## bit64      4.0.5 2020-08-30 [1] CRAN (R 4.1.0)
## broom     0.8.0 2022-04-13 [1] CRAN (R 4.1.2)
## bslib      0.3.1 2021-10-06 [1] CRAN (R 4.1.0)
## cellranger 1.1.0 2016-07-27 [1] CRAN (R 4.1.0)
## cli        3.3.0 2022-04-25 [1] CRAN (R 4.1.2)
## colorspace 2.0-3 2022-02-21 [1] CRAN (R 4.1.2)
## crayon    1.5.1 2022-03-26 [1] CRAN (R 4.1.2)
## DBI        1.1.2 2021-12-20 [1] CRAN (R 4.1.0)
## dplyr     2.1.1 2021-04-06 [1] CRAN (R 4.1.0)
## digest    0.6.29 2021-12-01 [1] CRAN (R 4.1.0)
## dplyr     * 1.0.9 2022-04-28 [1] CRAN (R 4.1.2)
## ellipsis   0.3.2 2021-04-29 [1] CRAN (R 4.1.0)
## evaluate  0.15 2022-02-18 [1] CRAN (R 4.1.2)
## fansi      1.0.3 2022-03-24 [1] CRAN (R 4.1.2)
## farver    2.1.0 2021-02-28 [1] CRAN (R 4.1.0)
## fastmap   1.1.0 2021-01-25 [1] CRAN (R 4.1.0)
## forcats   * 0.5.1 2021-01-27 [1] CRAN (R 4.1.0)
## fs         1.5.2 2021-12-08 [1] CRAN (R 4.1.0)
## generics   0.1.2 2022-01-31 [1] CRAN (R 4.1.2)
## GGally    * 2.1.2 2021-06-21 [1] CRAN (R 4.1.0)
## ggplot2   * 3.3.6 2022-05-03 [1] CRAN (R 4.1.2)
## glue       1.6.2 2022-02-24 [1] CRAN (R 4.1.2)
## gtable    0.3.0 2019-03-25 [1] CRAN (R 4.1.0)
## haven     2.5.0 2022-04-15 [1] CRAN (R 4.1.2)
## highr     0.9 2021-04-16 [1] CRAN (R 4.1.0)
## hms       1.1.1 2021-09-26 [1] CRAN (R 4.1.0)
```

Enjoy !

