BÀI THỰC HÀNH 1: LÀM QUEN VỚI R STUDIO

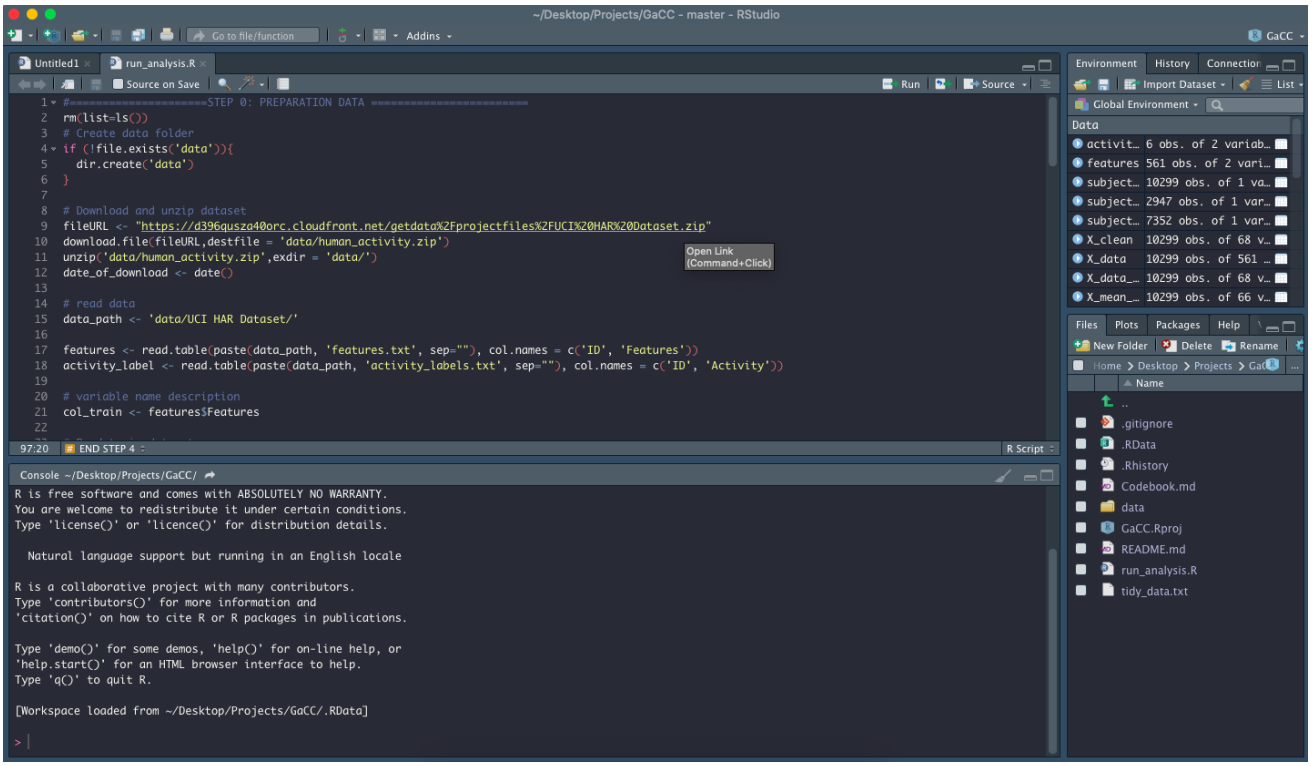
1. Giới thiệu về R Studio

R được tạo ra bởi 2 nhà thống kê học người New Zealand là Ross Ihanka và Robert Gentleman. Đây là ngôn ngữ chuyên dùng để phân tích và thống kê số liệu. R được cung cấp hoàn toàn miễn phí và có thể download tại địa chỉ:

Windows: https://cran.r-project.org/bin/windows/base/

R Studio là gói phần mềm mã nguồn mở.

Giao diện R Studio:



Hình 1: Giao diện của R Studio

2. Khởi tạo một Project mới trong R Studio

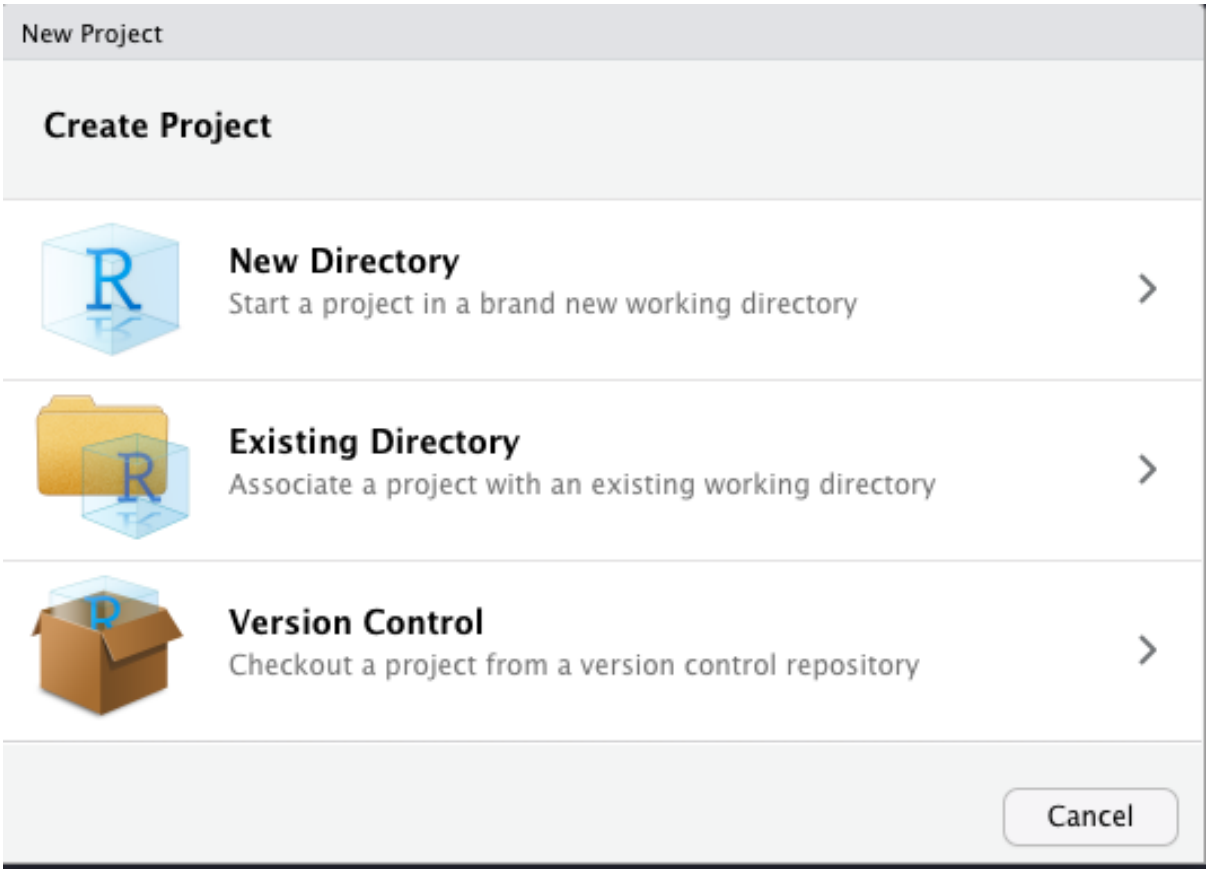
- Vào menu File -> New Project
- Cửa sổ New Project hiện ra:

    New Directory: Tạo 1 project trong thư mục mới.

    Existing Directory: Tạo project trong thư mục có sẵn

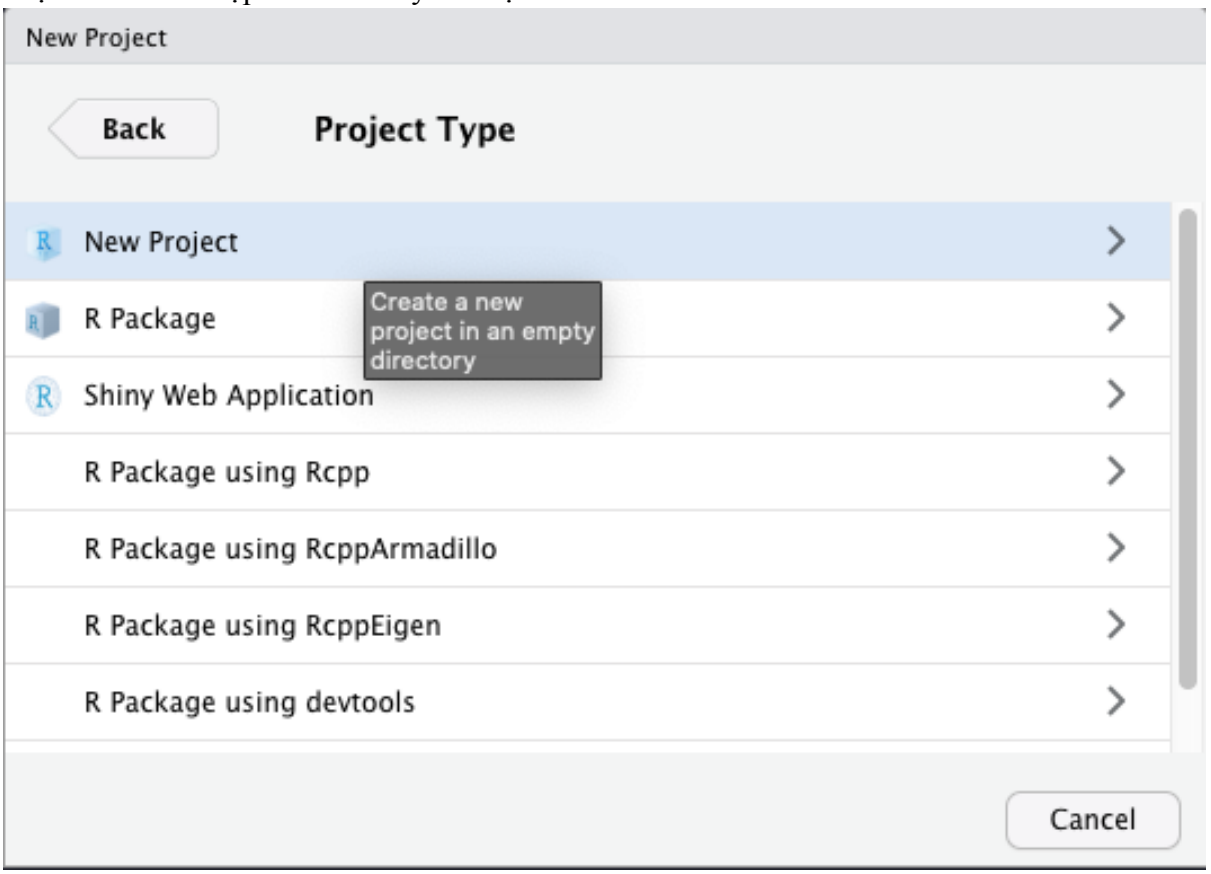
    Version Control: Tạo project từ 1 repository có sẵn trên github

Chọn New Directory



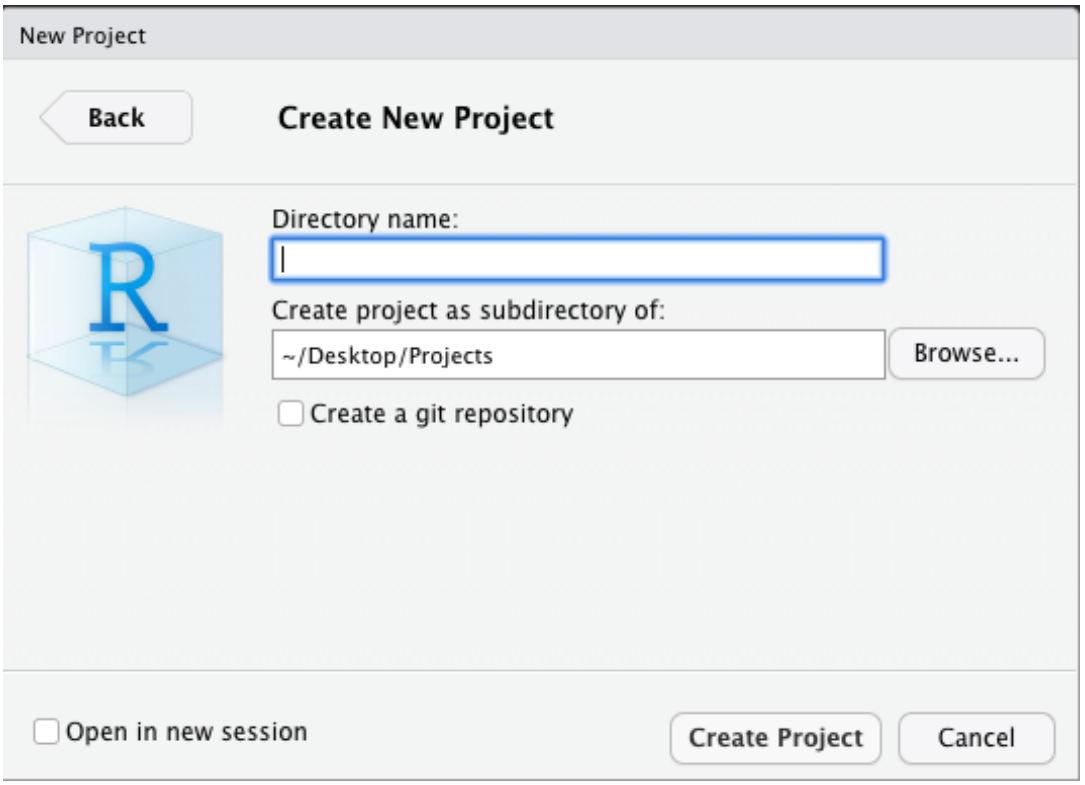
Hình 2: Cửa sổ tạo Project mới trong R Studio

- Chọn New Project



Hình 3: Cửa sổ chọn loại project muốn tạo trong R Studio

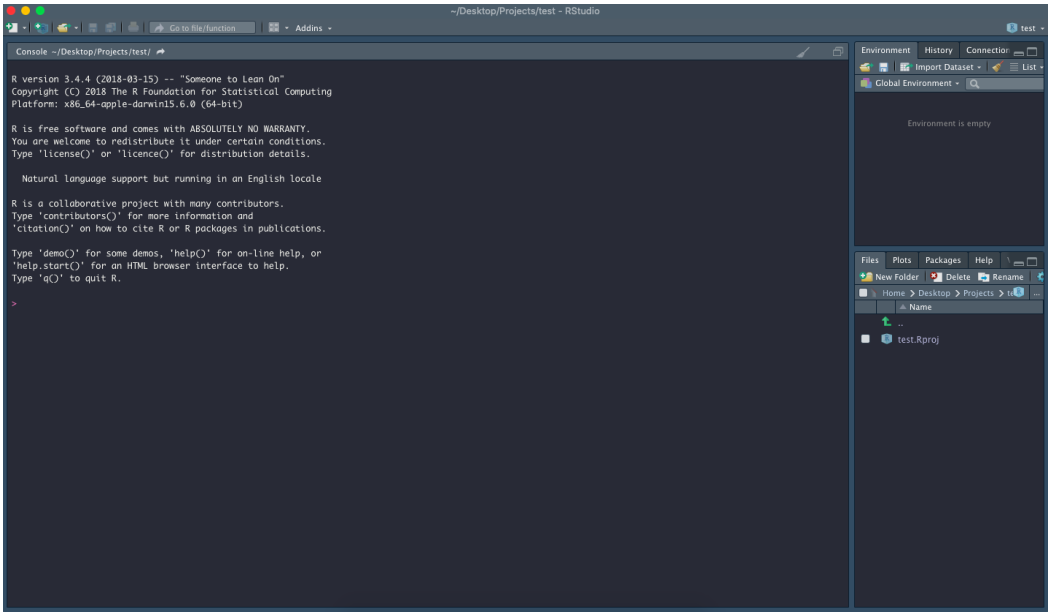
- Đặt tên cho thư mục chứa project trong Directory nam và chọn đường dẫn lưu thư mục project sẽ tạo ở mục Create project as subdirectory of



Hình 4: Cửa sổ đặt tên cho project

- Cuối cùng ta chọn nút create project.

Cửa sổ làm việc cho 1 project mới hiện ra như sau:



Hình 5: Giao diện Project mới sau khi tạo xong

<u>Để thao tác với R, ta có 2 cách:</u>

<u>Cách 1</u>: Gõ lệnh chạy trực tiếp tại dấu nhắc lệnh > (Xem Hình 5)

<u>Cách 2:</u> Viết các dòng lệnh vào file script, sau đó tiến hành chạy file script chứa dòng lệnh đó.

Lưu ý: Đối với các project cần xử lý nhiều và phức tạp thì khuyến khích nên chọn cách 2. R Studio cho phép quét khối và chạy từng lệnh tương ứng trong file script như SQL Server.

VD: Gõ lệnh: `>rnorm(10)`

Kết quả trả về:

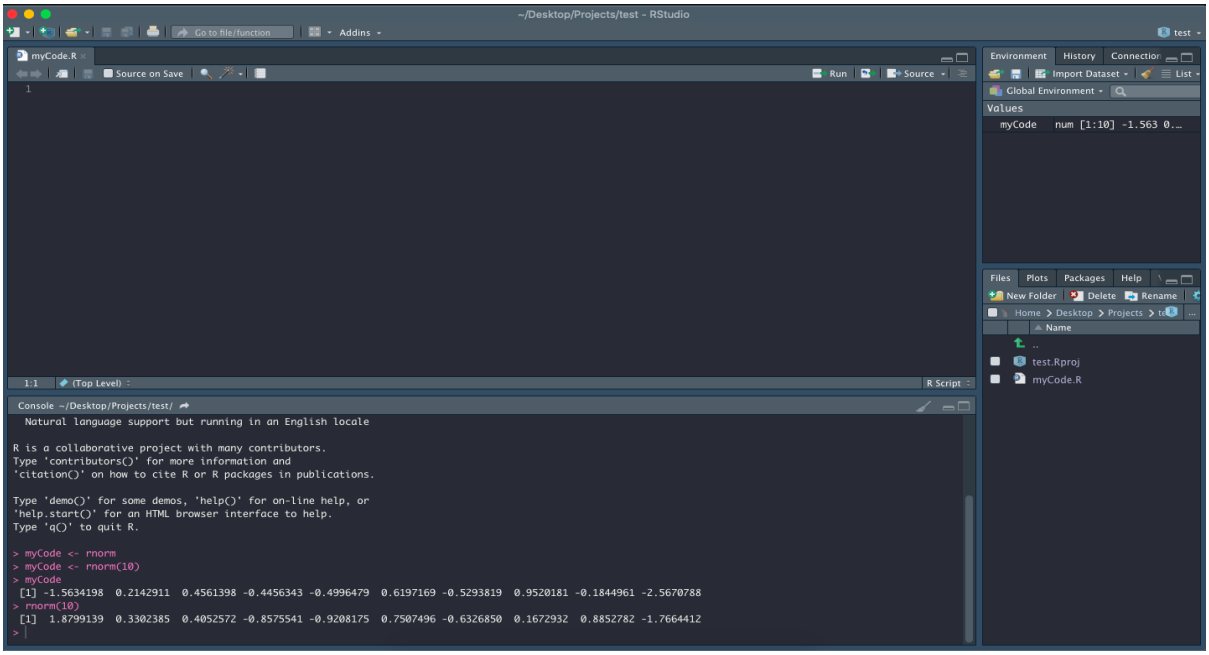
```
[1]  1.8799139  0.3302385  0.4052572 -0.8575541
-0.9208175  0.7507496 -0.6326850  0.1672932  0.8852782
-1.7664412
```

Để tạo file script, ta làm như sau:

- Vào File -> New File -> RScript.

- Lưu lại file với phần mở rộng là: .R

VD: myCode.R



Hình 6: Giao diện file script nhập lệnh vào (myCode.R)

3. Thực hành thao tác cơ bản với dữ liệu trong R Studio

Dữ liệu mẫu:

Bộ dữ liệu: **Novel Corona Virus 2019**

Link tải:

https://www.kaggle.com/sudalairajkumar/novel-corona-virus-2019-dataset

Bước 1: Khởi tạo Project trong R Studio và đặt tên là *corona_virus*

Bước 2: Tạo file chứa source code và đặt tên là *myCode.R*

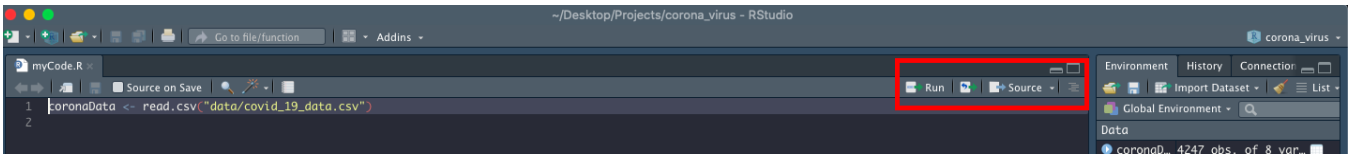
Bước 3: Tạo thư mục *data/* chứa trong thư mục project.

Bước 4: Giải nén bộ dữ liệu Novel Corona Virus 2019 và copy file covid_19_data.csv vào thư mục data/ đã tạo ở Bước 3.

Đọc dữ liệu và gán vào biến coronaData

```
coronaData <- read.csv("data/covid_19_data.csv")
```

Ghi chú: Cách thực thi lệnh code trong R Studio

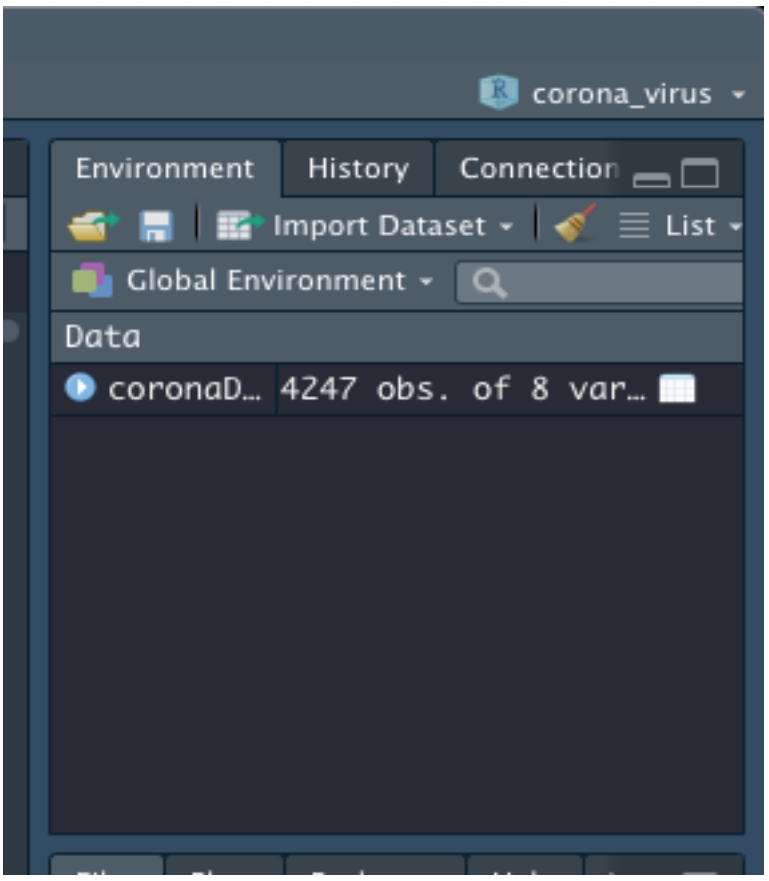


Hình 7: Giao diện các nút lệnh thực thi các đoạn lệnh trong file script

Để thực thi một dòng lệnh code, chúng ta quét khối dòng lệnh code cần thực hiện và bấm nút lệnh Run (trong ô màu đỏ như hình) để thực thi.

Nút lệnh Source sẽ thực thi toàn bộ các lệnh chứa trong file script.

Sau khi thực hiện lệnh đọc file covid_19_data.csv và gán vào biến coronaData, ta được biến coronaData là biến chứa dữ liệu được biểu diễn ở dạng Dataframe trong R.



Hình 8: Giao diện quản lý các biến trong môi trường R Studio

Để xem nội dung của dữ liệu vừa đọc, click vào biến dữ liệu như Hình 8

myCode.R | coronaData

Filter

	SNo	ObservationDate	Province.State	Country.Region	Last.Update	Confirmed	Deaths	Recovered
1	1	01/22/2020	Anhui	Mainland China	1/22/2020 17:00	1	0	0
2	2	01/22/2020	Beijing	Mainland China	1/22/2020 17:00	14	0	0
3	3	01/22/2020	Chongqing	Mainland China	1/22/2020 17:00	6	0	0
4	4	01/22/2020	Fujian	Mainland China	1/22/2020 17:00	1	0	0
5	5	01/22/2020	Gansu	Mainland China	1/22/2020 17:00	0	0	0
6	6	01/22/2020	Guangdong	Mainland China	1/22/2020 17:00	26	0	0
7	7	01/22/2020	Guangxi	Mainland China	1/22/2020 17:00	2	0	0
8	8	01/22/2020	Guizhou	Mainland China	1/22/2020 17:00	1	0	0
9	9	01/22/2020	Hainan	Mainland China	1/22/2020 17:00	4	0	0
10	10	01/22/2020	Hebei	Mainland China	1/22/2020 17:00	1	0	0
11	11	01/22/2020	Heilongjiang	Mainland China	1/22/2020 17:00	0	0	0
12	12	01/22/2020	Henan	Mainland China	1/22/2020 17:00	5	0	0
13	13	01/22/2020	Hong Kong	Hong Kong	1/22/2020 17:00	0	0	0
14	14	01/22/2020	Hubei	Mainland China	1/22/2020 17:00	444	17	28

Hình 9: Nội dung dữ liệu vừa đọc vào từ file ở dạng Dataframe.

Ghi chú: Dữ liệu sau khi được đọc vào R sẽ được lưu trữ ở dạng Dataframe. Dataframe là cấu trúc dùng để biểu diễn dữ liệu 2 chiều trong R.

Liệt kê số cột trong bảng dữ liệu: *Lệnh ncol()*

```
> ncol(coronaData)
```

Kết quả: 8

Liệt kê số dòng: *lệnh nrow()*

```
> nrow(coronaData)
```

Kết quả: 4247

In ra 10 dòng đầu trong bảng dữ liệu: *Lệnh head()*

```
> head(coronaData, 10)
```

In ra tên các biến (tên cột) của dữ liệu: *lệnh names()*

```
> names(coronaData)
```

Tạo biến countryCorona lưu giá trị là các quốc gia có dịch Corona (Cột Country.Region)

Phép gán trong R ký hiệu là: <-

```
> countryCorona <- coronaData['Country.Region']
```

Liệt kê số lượng ca lây nhiễm được xác nhận (biến Confirmed) nhiều nhất và lưu vào biến maxConfirmedCases. Sử dụng lệnh *max()*

```
> maxConfirmedCases <- max(coronaData['Confirmed'])
```

Liệt kê các dữ liệu về covid-19 tại quốc gia Trung Quốc đại lục (Mainland China) và lưu vào biến coronaChina.

Để trích xuất dữ liệu từ 1 cột, ta sử dụng cú pháp: **tên_dataset$tên_cột**.

VD: corona$Country.Region sẽ trích xuất dữ liệu từ cột Country.Region.

```
> coronaChina <-
coronaData[which(coronaData$Country.Region=='Mainland
China'),]
```

Tìm quốc gia (Country.Region) có số ca lây nhiễm nhiều nhất: sử dụng lệnh *which(<điều kiện>).*

Để lấy dữ liệu của 1 bộ dữ liệu theo điều kiện của một cột, ta dùng cú pháp: *tên_dataset[<điều kiện>,]*

VD: Câu lệnh dưới đây, sẽ lấy ra 1 dòng dữ liệu thoả điều kiện: cột Confirmed = số ca nhiễm lớn nhất

```
> maxCountryConfirmedCorona <-
coronaData[which(coronaData$Confirmed==maxConfirmedCases)
,]['Country.Region']
```

Tìm tỉnh (Province.State) có số ca lây nhiễm nhiều nhất

```
> maxStateConfirmedCorona <-
coronaData[which(coronaData$Confirmed==maxConfirmedCases)
,]['Province.State']
```

Lấy dữ liệu theo ngày tháng

Định dạng lại cột dữ liệu ngày tháng: Sử dụng hàm: *as.Date(<danh sách ngày tháng>, <định dạng ngày tháng trong dữ liệu>)*

```
coronaData$ObservationDate <-
as.Date(coronaData$ObservationDate, "%m/%d/%Y")
```

Lấy dữ liệu trong tháng 1/2020: bắt đầu từ 01/01/2020 đến 31/01/2020.

```
data_jan <- coronaData[which(coronaData$ObservationDate>=
"2020-01-01" & coronaData$ObservationDate <=
"2020-01-31"), ]
```

4. Bài tập

a) Code lại các ví dụ trong Phần 3.

b) Tìm dữ liệu về số ca lây nhiễm tại Vietnam (Country.Region == 'Vietnam') và lưu vào biến c*oronaVietnam.*

c) In ra số ca lây nhiễm nhiều nhất tại Việt Nam (Sử dụng lệnh print() trong R)

d) Tìm dữ liệu về số ca lây nhiễm tại Việt Nam trong tháng 02 năm 2021.

e) In ra số dữ liệu về ca lây nhiễm nhiều nhất trong khoảng tháng 01 và 02 tại Việt Nam (Lấy năm 2021).

f) Thực hiện tương tự câu e) cho Indonesia và Philipine.

g) In ra dữ liệu về ca nhiễm ghi nhân (*Confirmed*) của Trung Quốc trong khoảng thời gian từ 01/02/2021 cho đến 15/02/2021. In ra màn hình sử dụng lệnh print().

h) Thống kê số lượng record theo từng tỉnh của Trung Quốc trong tháng 02/2021.

Gợi ý: Dùng hàm *table()*. Đếm từng dòng theo từng tỉnh.

i) Đếm **số lượng ca nhiễm** theo từng tỉnh của Trung Quốc trong tháng 02/2021.

Gợi ý: *Lấy số ca nhiễm được ghi nhận (Confirmed) của ngày cuối cùng trong tháng trừ đi cho số ca nhiễm được ghi nhận (Confirmed) của ngày đầu tiên trong tháng*

k) Tìm dữ liệu ca tử vong của Trung Quốc trong khoảng thời gian từ 01/02/2021 cho đến 15/02/2021. In ra màn hình sử dụng lệnh *print()*.

*l) *Có nhận xét gì về số ca nhiễm mới tại Việt Nam giữa tháng 05/2020 và tháng 05/2021. Vẽ biểu đồ đường thể hiện số ca nhiễm mới trong 2 tháng trên. Gợi ý: Dùng hàm plot() trong R.*

*m) * Vẽ biểu đồ về số ca lây nhiễm nhiều nhất của 3 quốc gia: Vietnam, Indonesia và Philippine trong 2 tháng gồm 01 và tháng 02 năm 2021.*

Nộp bài: Các bạn nộp file source code và file báo cáo đi kèm.

Quy định nộp bài: Nộp 2 file code và báo cáo, không nén file, đặt tên theo cú pháp:

<MSSV>_<Họ tên>_BT1.pdf

và

<MSSV>_<Họ tên>_BT1.R

Chúc tất cả các bạn học tốt

5. Phụ lục: Một số lệnh thông dụng trong R.

Lệnh về môi trường vận hành của R

`getwd()`	Cho biết directory hiện hành là gì
`setwd(c:/works)`	Chuyển directory vận hành về c:\works (chú ý R dùng "/")
`options(prompt="R>")`	Đổi prompt thành R>
`options(width=100)`	Đổi chiều rộng cửa số R thành 100 characters
`options(scipen=3)`	Đổi số thành 3 số thập phân (thay vì kiểu 1.2E-04)
`options()`	Cho biết các thông số về môi trường hiện nay của R

Lệnh cơ bản

`ls()`	Liệt kê các đối tượng (objects) trong bộ nhớ
`rm(object)`	Xóa bỏ đối tượng
`seach()`	Tìm hướng

Kí hiệu tính toán

`+`	Cộng
`-`	Trừ
`*`	Nhân
`/`	Chia
`^`	Lũy thừa
`%/%`	Chia số nguyên
`%%`	Số dư từ chia hai số nguyên

Kí hiệu logic

`==`	Bằng
`!=`	Không bằng
`<`	Nhỏ hơn
`>`	Lớn hơn
`<=`	Nhỏ hơn hoặc bằng
`>=`	Lớn hơn hoặc bằng
`is.na(x)`	Có phải x là biến số missing
`&`	Và (AND)

`|`	Hoặc (OR)
`!`	Không là (NOT)

Tạo số

`numeric(n)`	Cho ra n số 0
`character(n)`	Cho ra n kí tự ""
`logical(n)`	Cho ra n FALSE
`seq(-4,3,0.5)`	Dãy số -4.0, -3.5, -3.0, …, 3.0
`1:10`	Giống như lệnh seq(1, 10, 1)
`c(5,7,9,1)`	Nhập số 5, 7, 8 và 1
`rep(1, 5)`	Cho ra 5 sô 1: 1, 1, 1, 1, 1.
`Gl(3,2,12)`	Yếu tố 3 bậc, lặp lại 2 lần, tổng cộng 12 số: 1 1 2 2 3 3 1 1 2 2 3 3

Tạo nên số ngẫu nhiên bằng mô phỏng theo các luật phân phối (simulation)

`rnorm(n, mean=0, sd=1)`	Phân phối chuẩn (normal distribution) với trung bình = 0 và độ lệch chuẩn = 1.
`rexp(n, rate=1)`	Phân phối mũ (exponential distribution)
`rgamma(n,shape,scale=1)`	Phân phối gamma
`rpois(n, lambda)`	Phân phối Poisson
`rweibull(n,shape,scale=1)`	Phân phối Weibull
`rcauchy(n,location=0,scale=1)`	Phân phối Cauchy
`rbeta(n, shape1, shape2)`	Phân phối beta
`rt(n, df)`	Phân phối t
`rchisq(n, df)`	Phân phối Chi bình phương
`rbinom(n, size, prob)`	Phân phối nhị phân (binomial)
`rgeom(n, prob)`	Phân phối geometric
`rhyper(nn, m, n, k)`	hypergeometric
`rlnorm(n,meanlog=0,sdlog=1)`	Phân phối log normal
`rlogis(n,location=0,scale=1)`	Phân phối logistic
`rnbinom(n,size,prob)`	Phân phối negative Binomial
`runif(n,min=0,max=1)`	Phân phối uniform

Biến đổi số thành kí tự (character) và ngược lại

`as.numeric(x)`	Biến đổi x thành biến số số học để có thể tính toán
`as.character(x)`	Biến đổi x thành biến số chữ (character) để phân loại
`as.logical(x)`	Biến đổi x thành biến số logic
`factor(x)`	Biến đổi x thành biến số yếu tố

Data frames

`data.frame(x,y)`	Nhập x và y thành một data frame
`tuan$age`	Chọn biến số `age` từ dataframe `tuan`.

`attach(tuan)`	Đưa dataframe `tuan` vào hệ thống R
`detach(tuan)`	Xóa bỏ dataframe `tuan` khỏi hệ thống R

Hàm số toán

`log(x)`	Logarít bậc e
`log10(x)`	Logarít bậc 10
`exp(x)`	Số mũ
`sin(x)`	Sin
`cos(x)`	Cosin
`tan(x)`	Tangent
`asin(x)`	Arcsin (hàm sin đảo)
`acos(x)`	Arccosin (hàm cosin đảo)
`atan(x)`	Arctang(hàm tan đảo)

Hàm số thống kê

`min(x)`	Số nhỏ nhất của biến số x
`max(x)`	Số lớn nhất của biến số x
`which.max(x)`	Tìm dòng nào có giá trị lớn nhất của biến số x
`which.min(x)`	Tìm dòng nào có giá trị nhỏ nhất của biến số x
`length(x)`	Tổng số yếu tố (elements) trong một biến số (hay số mẫu)
`sum(x)`	Số tổng của biến số x
`range(x)`	Khác biệt giữa max(x) và min(x)
`mean(x)`	Số trung bình của biến số x
`median(x)`	Số trung vị (median) của biến số x
`sd(x)`	Độ lệch chuẩn (standard deviation) của biến số x
`var(x)`	Phương sai (variance) của biến số x
`cov(x,y)`	Hiệp biến (covariance) giữa hai biến số x và y
`cor(x,y)`	Hệ số tương quan (coefficient of correlation) giữa biến số x và y.
`quantile(x)`	Chỉ số của biến số x
`cor(x,y)`	Hệ số tương quan (correlation coefficient) giữa biến số x và y
`is.na(x)`	Kiểm tra xem x có phải là số trống không (missing value)
`complete.cases(x1,x2,...)`	Kiểm tra nếu tất cả x1, x2, … đều không có số trống.

Chỉ số ma trận

`x[1]`	Số đầu tiên của biến số x
`x[1:5]`	Năm số đầu tiên của biến số x
`x[y<=30]`	Chọn x sao cho y nhỏ hơn hoặc bằng 30
`x[sex=="male"]`	Chọn x sao cho `sex` bằng `male`

Nhập dữ liệu

`data(name)`	Xây dựng một kho dữ liệu

`read.table("name")`	Đọc / nhập số liệu từ file `name`
`read.csv("name")`	Đọc / nhập số liệu dạng excel (cách nhau bằng ",") từ file `name`
`read.delim("name")`	Đọc / nhập số liệu dạng tab delimited
`read.delim2("name")`	Đọc / nhập số liệu dạng tab delimited, cách nhau bằng ";" và số thập phân là ","
`read.csv2("name")`	Đọc / nhập số liệu dạng csv, cách nhau bằng ";" và số thập phân là ","

Phần phụ trong read.table

`header=TRUE`	Hàng đầu tiên của dữ liệu là tên của biến số
`sep=","`	Số liệu ngăn cách bằng dấu hiệu ","
`dec=","`	Số thập phân là "," (để phân biệt với ".")
`na.strings="."`	Số liệu trống (missing value) là "."

Phân phối xác suất

`pnorm(x,mean,sd)`	Phân phối chuẩn
`plnorm(x,mean,sd)`	Phân phối chuẩn logarit
`pt(x,df)`	Phân phối t
`pf(x,n1,n2)`	Phân phối F
`pchisq(x,df)`	Phân phối Chi bình phương
`ppois(x,lambda)`	Phân phối Poisson
`punif(x,min,max)`	Phân phối uniform (đồng dạng)
`pexp(x,rate)`	Phân phối hàm mũ
`pgamma(x,shape,scale)`	Phân phối gamma
`pbeta(x,a,b)`	Phân phối beta

Phân tích thống kê

`t.test`	Kiểm định t
`pairwise.t.test`	Kiểm định t cho paired design
`cor.test`	Kiểm định hệ số tương quan
	`method = "kendall"`
	`method = "spearman"`
`var.test`	Kiểm định phương sai
`bartlett.test`	Kiểm định nhiều phương sai
`wilcoxon.test`	Kiểm định Wilcoxon
`kruskal.test`	Kiểm định Kruskal
`friedman.test`	Kiểm định Friedman

`lm(y ~ x)`	Phân tích hồi qui tuyến tính (linear regression)
`lm(y ~ factor)`	Phân tích phương sai 1 chiều (1-way analysis of variance)
`lm(y ~ factor+x)`	Phân tích hiệp biến (analysis of covariance)
`lm(y ~ x1+x2+x3)`	Phân tích hồi qui tuyến tính đa biến số (multiple linear regression)
`binom.test`	Kiểm định nhị phân (Binomial test)
`prop.test`	Kiểm định so sánh nhiều tỉ số
`prop.trend.test`	Kiểm định so sánh nhiều tỉ số theo xu hướng
`fisher.test`	Kiểm định Fisher
`chisq.test`	Kiểm định Chi bình phương
`glm(y~x1+x2+x+x3)`	Phân tích hồi qui logistic
`s<-Surv(time,event)`	Phân tích survival
`survfit(s)`	Biểu đồ Kaplan-Meier
`survdiff(s~g)`	Kiểm định Log-rank giữa hai nhóm g
`coxph(s ~ x1+x2)`	Phân tích hồi qui Cox

Tìm mô hình dựa vào tiêu chuẩn AIC

```
cox <- coxph(Surv(y, death) ~ ., data=simdata)
searchAIC <- step(cox, direction="both")
summary(searchAIC)
```

Bayesian model average

```
time <- simdata$y
death <- simdata$death
xvars <- simdata[,c(3,4,5,6,7)]
bma <- bic.surv(xvars, time, death)
summary(bma)
imageplot.bma(bma)
```

Biểu đồ

`plot(y~x)`	Vẽ đồ thị y và x (scatter plot)
`hist(x)`	Vẽ đồ thị y và x (scatter plot)
`plot(y ~ x | z)`	Vẽ hai biểu đồ x và y theo từng nhóm của z
`pie(x)`	Vẽ đồ thị tròn
`boxplot(x)`	Vẽ đồ thị theo dạng hình hộp
`qqnorm(x)`	Vẽ phân phối quantile của biến số x
`qqplot(x, y)`	Vẽ phân phối quantile của biến số y theo x
`barplot(x)`	Vẽ biểu đồ hình khối cho biến số x
`hist(x)`	Vẽ histogram cho biến số x
`stars(x)`	Vẽ biểu đồ sao cho biến số x
`abline(a, b)`	Vẽ đường thẳng với intercept=a và slope=b
`abline(h=y)`	Vẽ đường thẳng ngang
`abline(v=x)`	Vẽ đường thẳng đứng

`abline(lm.object)`	Vẽ đồ thị theo mô hình tuyến tính

Một số thông số cho biểu đồ

`pch`	Kí hiệu để vẽ đồ thị (pch = *p*lotting *ch*aracters)
`mfrow, mfcol`	Tạo ra nhiều cửa sổ để vẽ nhiều đồ thị cùng một lúc (*m*ulti*fr*ame)
`xlim, ylim`	Cho giới hạn của trục hoành và trục tung
`xlab, ylab`	Viết tên trục hoành và trục tung
`lty, lwd`	Dạng và kích thước của đường biểu diễn
`cex, mex`	Kích thước và khoảng cách giữa các kí tự.
`col`	Màu sắc