

R 103

Tags	Programming	R
Class		
Finished Yet?	<input checked="" type="checkbox"/>	
Knowledge	The Third Sprint: R	

R superpower!

- R เป็นภาษาที่ Fast Data Crunching ที่ถูกพัฒนาขึ้นมาสำหรับงาน Data โดยเฉพาะ เปิดโปรแกรมแล้วทำงานได้กันที
- R เป็น 1 ใน 3 ภาษาโปรแกรมที่ Data Analyst ต้องรู้จักและใช้ให้เป็น
- เราสามารถใช้ R อ่านไฟล์ข้อมูลได้หลายประเภท
- เราสามารถติดตั้ง library เพิ่มเติมนอกเหนือจาก base R เพื่อช่วยในการทำงานได้ เช่น:

```
#Install Packages
install.packages(c("readr",
                  "readxl",
                  "googlesheets4",
                  "jsonlite",
                  "dplyr",
                  "sqldf",
                  "RSQLite"))
```

```
#Load Library
library(readr)
library(readxl)
library(googlesheets4)
library(jsonlite)
library(dplyr)
library(sqldf)
library(RSQLite)
```

How to install R packages

-Package หรือ Library มีความหมายเหมือนกัน

-เราสามารถติดตั้ง package ใน R ได้ 2 วิธี คือ:

1. `install.packages()`
2. Packages Tab ใน RStudio → กด install

-Package ที่เราสามารถโหลดมาใช้ได้: https://cran.r-project.org/web/packages/available_packages_by_name.html

-เราสามารถกด Update Tab ใน RStudio เพื่อ Update library ได้

Lesson 1: Text File

-RStudio สามารถเขียน Script ได้มากกว่าภาษา R เช่น C++, Python, SQL เป็นต้น

-`read_table()` ใช้อ่าน file .txt ที่ใช้ whitespace (ช่องว่าง) เป็นตัวคั่นระหว่าง column (delimiter) [ก่อนใช้ ให้โหลด library `readr` มาก่อน] เช่น:

The screenshot shows the RStudio interface. On the left, there's a 'Text file' tab with a file named 'student'. The content of the file is:

```
1 id name
2 1 Diluc
3 2 Kaeya
4 3 Jean
5 4 Albedo
6 5 Lisa
```

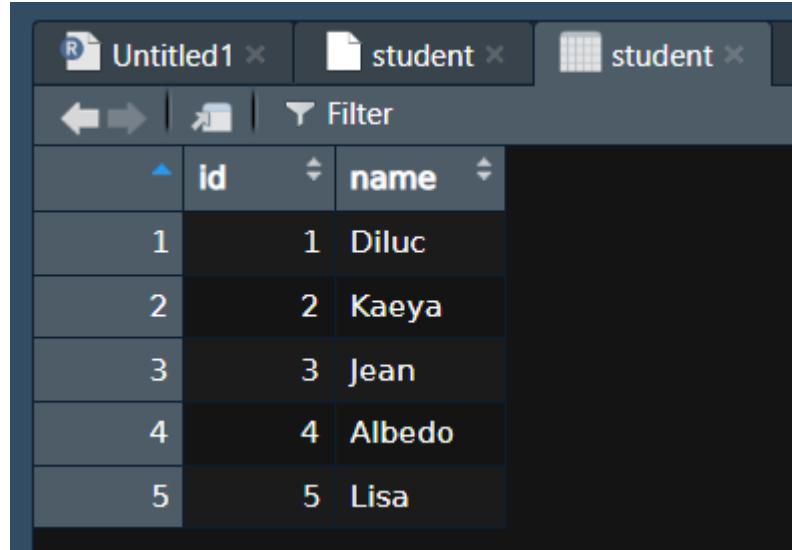
On the right, the R console window shows the command:

```
> student <- read_table("student")
```

Below the console, the 'Files' tab of the file browser is selected, showing the following files:

Name	Size	Modified
Function_kata.R	318 B	Dec 22, 2022, 11:20 PM
hotel_bookings.csv	16.1 MB	Sep 24, 2022, 5:32 PM
Looping_over_a_dataframe.R	430 B	Dec 22, 2022, 11:23 PM

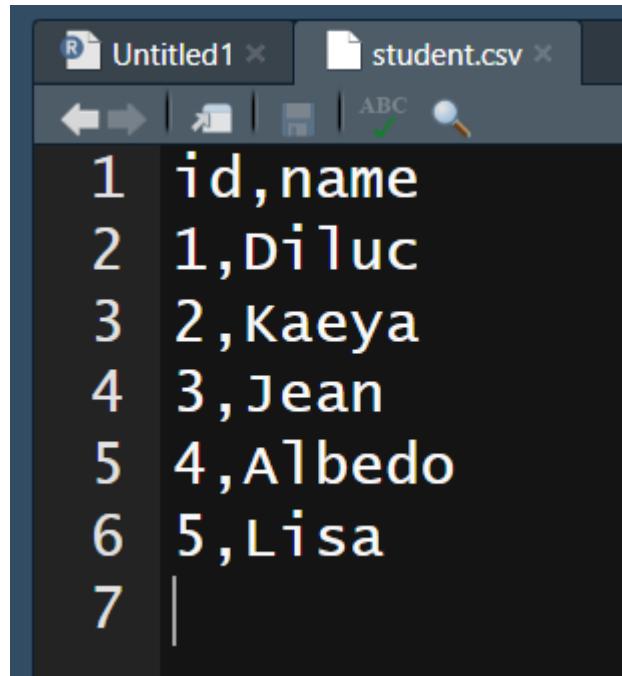
[หลังจากที่เรา assign ตัว text file ของเรามาเป็นตัวแปรแล้ว เราสามารถกดที่ไอคอนตารางตรง Data เพื่อให้แสดงผลออกมาเป็น Table ได้ หรือจะใช้ View(student) ก็ได้เช่นกัน]



	Id	name
1	1	Diluc
2	2	Kaeya
3	3	Jean
4	4	Albedo
5	5	Lisa

Lesson 2: CSV

-ไฟล์ประเภท CSV (Comma-Separated Values) คือไฟล์ที่ใช้ comma เป็น delimiter คั่นระหว่าง column เช่นไฟล์ที่เราทำงานใน Google Sheets หรือ Excel ซึ่งเราสามารถใช้ R อ่านไฟล์ประเภทนี้ได้ด้วย `read_csv` เช่น:



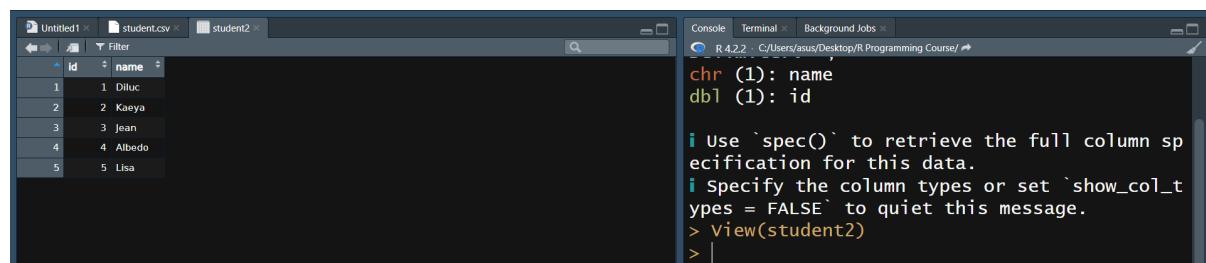
1	id, name
2	1, Diluc
3	2, Kaeya
4	3, Jean
5	4, Albedo
6	5, Lisa
7	

[ตัวอย่างของ .csv file]

```
> student2 <- read_csv("student.csv")
Rows: 5 Columns: 2
— Column specification —
Delimiter: ","
chr (1): name
dbl (1): id

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.
```

[read_csv ใช้อ่าน file ประเภท .csv ซึ่งเราสามารถ assign ผลกับตัวแปรเพื่อเรียกใช้งานต่อไปได้]



The screenshot shows the RStudio interface. On the left, there is a data viewer window titled 'student2' containing the following data:

		Filter
		id name
1	1	Diluc
2	2	Kaeya
3	3	Jean
4	4	Albedo
5	5	Lisa

On the right, the 'Console' tab shows the R code and its output:

```
R 4.2.2 C:\Users\asus\Desktop\R Programming Course>
chr (1): name
dbl (1): id

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.
> View(student2)
> |
```

[สังเกตว่าหลังจาก View แล้ว จะแสดงผลเป็น Table เมื่อบันกัน]

Lesson3 : Excel File

- เราสามารถใช้ read_excel() จาก library ชื่อ readxl เพื่ออ่านไฟล์ประเภท .xlsx ได้ เช่น:

```

> library("readxl")
> read_excel("students.xlsx", sheet = 1)
# A tibble: 5 × 3
  student_id name   major
      <dbl> <chr>  <chr>
1         1 toy    economics
2         2 john   economics
3         3 mary   economics

```

[sheet = 1 หมายความว่า อ่าน sheet แผ่นที่ 1 ใน file xlsx. มีได้หลาย sheet ซึ่งเราสามารถพิมพ์ sheet = “ชื่อของ sheet นั้น ๆ” เพื่อดู sheet ที่ต้องการได้เช่นกัน]

The screenshot shows the RStudio interface. On the left, the code editor window titled 'Untitled1' contains the following R code:

```

1 result <- list()
2
3 for (i in 1:3) {
4   result[[i]] <- read_excel("students.xlsx", sheet = i)
5 }
6
7 for (i in 1:3) {
8   print(result[[i]])
9 }
10

```

On the right, the 'Console' window shows the output of the code:

```

1       1 toy    economics
2       2 john   economics
3       3 mary   economics
4       4 anna  economics
5       5 lisa   economics
# A tibble: 5 × 3
  student_id name   major
      <dbl> <chr>  <chr>
1         1 toy    economics

```

```

#for loop to assign each sheet to result variable
for (i in 1:3) {
  result[[i]] <- read_excel("students.xlsx", sheet = i)

#for loop to print all sheets
for (i in 1:3) {
  print(result[[i]])
}

```

Lesson 4: Google Sheets

-read_sheet() ใช้อ่านข้อมูลจาก Google Sheets

-ถ้าเป็น public link ก่อนใช้คำสั่ง read_sheet() ให้รัน gs4_deauth() ก่อน (แปลว่า link ที่เราจะอ่าน file เป็น public link ไม่ต้องมีการ login)

-อ่านข้อมูลเพิ่มเติมเกี่ยวกับ gs4deauth() ได้ที่:

<https://googlesheets4.tidyverse.org/articles/auth.html>

-ตัวอย่าง code ในการอ่านข้อมูลจาก Google Sheets:

```
#Run the library necessary to use read_sheet()
library(googlesheets4)
#Assign Google Sheets url to a variable
url <- "Google Sheets Link"
#If the file is open to public, use gs4_deauth
gs4_deauth()
#Then, use read_sheet to read Google Sheets's spreadsheet
read_sheet(url, sheet = "sheet's Name")
```

Lesson 5: JSON

-JSON = JavaScript Object Notation

-เราสามารถใช้ fromJSON() เพื่ออ่าน file ประเภท JSON เข้ามาเป็น List ใน R และค่อยใช้ data.frame() เพื่อเปลี่ยน List → Data Frame ได้

-Key ใน JSON ต้องเขียนด้วย “ ” (Double Quote) เท่านั้น

-อ่านเพิ่มเติมเกี่ยวกับ jsonlite ได้ที่: <https://cran.r-project.org/web/packages/jsonlite/index.html>

```
teyvat2.json x teyvat2 x
1 {  
2   "id" : [1,2,3,4,5] ,  
3   "name" : ["Diluc","Kaeya","Jean","Albedo","Lisa"] ,  
4   "tall" : [true,true,true,false,true]  
5 }
```

1	Diluc	TRUE	
2	Kaeya	TRUE	
3	Jean	TRUE	
4	Albedo	FALSE	
5	Lisa	TRUE	

```
Console Terminal Background Jobs x
R 4.2.2 - C:/Users/asus/Desktop/R Programming Course/▶
> teyvat2 <- data.frame(fromJSON("teyvat2.json"))
> View(teyvat2)
> |
```

[ใช้ fromJSON("ชื่อไฟล์ JSON") เพื่ออ่านไฟล์ JSON จากนั้น ทำเป็น Data Frame และเก็บค่าไว้ในตัวแปรเพื่อเรียกductหรือใช้งานต่อไปได้]

Lesson 6: Bind Rows (UNION ALL)

-bind_rows() ใน R = UNION ALL ใน SQL

```
library(dplyr)
library(readxl)
#Read Excel file
econ <- read_excel("students.xlsx", sheet = 1)
business <- read_excel("students.xlsx", sheet = 2)
data <- read_excel("students.xlsx", sheet = 3)
#bind_rows() = UNION ALL in SQL
```

-ตัวอย่างการใช้งาน:

```
library(dplyr)
library(readxl)
#Read Excel file
econ <- read_excel("students.xlsx", sheet = 1)
business <- read_excel("students.xlsx", sheet = 2)
data <- read_excel("students.xlsx", sheet = 3)
#bind_rows() = UNION ALL in SQL
bind_rows(econ, business)
```

```
> bind_rows(econ, business)
# A tibble: 10 × 3
  student_id name   major
        <dbl> <chr> <chr>
1          1 toy    economics
2          2 john   economics
3          3 mary   economics
4          4 anna   economics
5          5 lisa   economics
6          6 happy  business
```

-ในการทำงานจริง เราอาจมี Data Frame มากมาย ดังนั้น เราควรสร้าง List ของ Data Frame ขึ้นมา ก่อน จากนั้นถึงค่อยใช้ bind_rows(list.df) เช่น:

```
list_df <- list(econ, business, data)
full_df <- bind_rows(list_df)
```

```
library(readxl)
#Read Excel file
econ <- read_excel("students.xlsx", sheet = 1)
business <- read_excel("students.xlsx", sheet = 2)
data <- read_excel("students.xlsx", sheet = 3)
#bind_rows() = UNION ALL in SQL
bind_rows(econ, business)
list_df <- list(econ, business, data)
full_df <- bind_rows(list_df)
```

```
> full_df
# A tibble: 15 × 3
  student_id name   major
        <dbl> <chr> <chr>
1          1 toy    economics
2          2 john   economics
3          3 mary   economics
4          4 anna   economics
5          5 lisa   economics
6          6 happy  business
```

Lesson 7: Bind Cols (\neq JOIN)

-Bind Cols ไม่เท่ากับการเขียน JOIN เพราะไม่จำเป็นต้องใช้ Key ใด ๆ เพียงเอา Data Frame มาต่อข้างกันเท่านั้น

-rep() ช่วยในการทำซ้ำในกรณีที่ข้อมูลเหมือนกัน เช่น

```

city = c("Mondstadt", "Mondstadt", "Mondstadt")
#Equals to:
city = c( rep("Mondstadt", 3) )

```

The screenshot shows the RStudio interface with two panes. The left pane contains R code for creating two data frames, df1 and df2. The right pane shows the resulting data frames in the console.

```

12
13 df1 <- data.frame(
14   id = 1:3,
15   name = c("Amber", "Lisa", "Kaeya")
16 )
17 df2 <- data.frame(
18   city = c( rep("Mondstadt", 3) ),
19   country = c( rep("Teyvat", 3) )
20 )
21

```

	id	name
1	1	Amber
2	2	Lisa
3	3	Kaeya

	city	country
1	Mondstadt	Teyvat
2	Mondstadt	Teyvat
3	Mondstadt	Teyvat

[เราจะทำการ Bind Cols 2 Data Frame นี้ด้วยกัน]

The screenshot shows the RStudio interface with two panes. The left pane contains R code for creating two data frames, df1 and df2, and then merging them using bind_cols. The right pane shows the merged data frame in the console.

```

15   name = c("Amber", "Lisa", "Kaeya")
16 )
17 df2 <- data.frame(
18   city = c( rep("Mondstadt", 3) ),
19   country = c( rep("Teyvat", 3) )
20 )
21
22 bind_cols(df1, df2)
23 bind_cols(df2, df1)
24

```

	id	name	city	country
1	1	Amber	Mondstadt	Teyvat
2	2	Lisa	Mondstadt	Teyvat
3	3	Kaeya	Mondstadt	Teyvat

	city	country	id	name
1	Mondstadt	Teyvat	1	Amber
2	Mondstadt	Teyvat	2	Lisa

[สังเกตว่า bind_cols(df1, df2) จะให้ผลลัพธ์ไม่เหมือนกับ bind_cols(df2, df1)]

-R สามารถ JOIN ได้เหมือนกับ SQL เช่น:

```

#df1
df1 <- data.frame(
  id = 1:3,
  name = c("Amber", "Lisa", "Kaeya")
)
#df2
df2 <- data.frame(
  id = 1:3,
  city = c( rep("Mondstadt", 3) ),
  country = c( rep("Teyvat", 3) )
)
#LEFT JOIN by id
left_join(df1, df2, by = "id")

```

The screenshot shows the RStudio interface with the following details:

- Editor:** The left pane displays R code in a script named "Untitled1.R". The code creates two data frames, df1 and df2, and then performs a left join on them.
- Console:** The right pane shows the R session output. It starts with the R version (R 4.2.2), the working directory (C:/Users/susus/Desktop/R Programming Course), and the creation of city and country vectors. Then, it shows the result of the left join between df1 and df2 on the "id" column. The resulting data frame has columns: id, name, city, and country. The data is as follows:

id	name	city	country
1	Amber	Mondstadt	Teyvat
2	Lisa	Mondstadt	Teyvat
3	Kaeya	Mondstadt	Teyvat

-bind_cols() อาจไม่ได้ใช้บ่อยเท่า JOIN เพราะการเขียน JOIN สามารถใช้ match id ของ 2 Data Frames หรือมากกว่าได้ เช่น customerid เป็นต้น

-ref. how to cross join in R?: <https://stackoverflow.com/questions/10600060/how-to-do-cross-join-in-r>

Lesson 8: SQL

- เราสามารถใช้ `sqldf()` เพื่อใช้เขียน SQL จัดการกับ Data Frame ใน R ได้ เช่น:

```
#Load library sqldf and readr
library(sqldf)
library(readr)

#Assign school.csv to variable "school"
school <- read_csv("school.csv")

#Select all columns from school dataframe
sqldf("select * from school;")

#Select average and sum of student data from the dataframe (aggregation)
sqldf("select avg(student), sum(student) from school;")

#Select school_id, school_name, and country column from school dataframe
sqldf("select school_id, school_name, country from school;")

#Store SQL query as a variable
sql_query <- "select * from school where country = 'USA';"

#Use stored query to assign values in variable "usa_school"
usa_school <- saldf(sql_query)
```

The screenshot shows the RStudio interface. On the left, the code editor window titled 'sqlldr.readr.R' contains R code for reading data from an SQLite database. On the right, the 'Console' tab shows the output of the code execution, displaying a data frame named 'usa_school' with columns: school_id, school_name, country, and student.

```

sqlldr.readr.R* [1]
1 # Load library
2 library(sqldf)
3
4 # Read data from SQLite database
5 usa_school <- sqldf("select * from school")
6
7 # Print the first few rows of the data frame
8 head(usa_school)
9
10 # Calculate average and sum of student data
11 avg_student <- sqldf("select avg(student), sum(student) from school")
12
13 # Store SQL query as a variable
14 sql_query <- "select * from school where country = 'USA'"
15
16 # Use stored query to assign values in another variable
17 usa_school <- sqldf(sql_query)

```

```

Console Terminal Background Jobs
> usa_school <- sqldf(sql_query)
> usa_school
  school_id school_name country student
1          4      Harvard    USA     48000
2          5         MIT    USA     62000
> top10
1 TRUE
2 TRUE
>

```

Lesson 9: SQLite

- เราสามารถใช้ library ชื่อ RSQLite เพื่อจัดการกับข้อมูลใน sqlite .db file

- ขั้นตอนการทำงานแบ่งเป็น 3 ขั้นตอน ได้แก่:

1. Connect to Database (Open Connection)
2. Get Data with SQL
3. Disconnect from Database (Close Connection)

- อ่านเพิ่มเติมเกี่ยวกับ Database ใน R ได้ที่: <https://dbi.r-dbi.org/>

- Code ตัวอย่างในการใช้ RSQLite:

```

#Load library
library(RSQLite)

#Connect to SQLite database (.db file)
#1. Open Connection
conn <- dbConnect(SQLite(), "chinook.db")

#2. Get Data
dbListTables(conn)
dbListFields(conn, "customers")

df <- dbGetQuery(conn, "select * from customers where country = 'USA'")
df2 <- dbGetQuery(conn, "select * from customers where country = 'United Kingdom'")

#3. Close Connection
dbDisconnect(conn)

```

*อย่าลืมเช็คให้ดีว่า Database File ของเรายู่ใน Working Directory และหรือยัง

Lesson 10: How to save data in R

-เราสามารถ save data ใน R ได้ 2 แบบเพื่อนำกลับมาใช้ในอนาคต ต่อให้เราลบ data ทั้งหมด (ไม่มี data หลงเหลือบนหน้าต่าง environment และล้วง) ก็สามารถโหลดกลับมาใช้ใหม่ได้

1. `save.image()`: ใช้ save objects ทั้งหมดที่อยู่ใน environment (เช่น workspace) ของเรา เข้าไปที่ไฟล์ `.RData`
2. `saveRDS()`: ใช้ save single object แค่ไฟล์เดียวที่ไฟล์ `.rds`

-ถ้าต้องการโหลด `.RData` ให้ใช้ function `load()`

-ถ้าต้องการโหลด `.rds` ให้ใช้ function `readRDS()`

ยกตัวอย่างเช่น:

```
#save.image() + load()
save.image(file = "data.RData")
load(file = "data.RData")

#saveRDS() + readRDS()
saveRDS(business, file = "business.rds")
business <- readRDS("business.rds")
```