



Data Transformation Live Class

Tags	Live Class	R
Class		
Finished Yet?	<input checked="" type="checkbox"/>	
Knowledge	Live Classes	

Bash Command Lines

-Command lines = คำสั่งที่เราใช้สั่งคอมของเรา (เช่น dir = โจร directory กั้งหมด, หรือ cd (change directory) = ย้าย directory เช่น cd desktop = ย้าย working directory ไปยัง desktop)

-Bash ย่อมาจาก Bourne Again Shell

-pwd = Print Working Directory

-ls = List File และดูไฟล์กั้งหมด

-touch ชื่อไฟล์ = สร้างไฟล์ใหม่กับชื่อตามที่เราตั้ง

-clear = clear terminal/shell (ใช้ CTRL + L ได้เช่นกัน)

-rm = Remove ใช้ลบไฟล์ (สามารถลบหลายไฟล์ได้พร้อมกัน เช่น rm *.txt จะลบไฟล์กั้งหมดที่มีนามสกุล txt)

-echo = print (เช่น echo hello world เป็นตัวบ) *ใช้ print ตัวแปรได้ เช่น A = 10 และ echo \$A จะออกมาเป็น 10

- เราสามารถเขียนค่าใหม่กับตัวแปรเก่าได้
 - เราสามารถเพิ่มข้อมูลในไฟล์ได้ เช่น echo cat > animal.txt (เพิ่ม cat ใน animal.txt) เป็นต้น
 - ถ้าอยากเพิ่มค่า (ไม่ใช่แทนที่ค่า) ให้ใช้ >> แทน > เช่น echo dog >> animal.txt เป็นต้น
 - cat = เรียกดู Content ใน File เช่น cat animal.txt เป็นต้น
 - mkdir = Make New Directory เช่น mkdir new_folder เป็นต้น (Folder จะไม่มีนามสกุลต่อท้าย)
 - เรายังสามารถใช้ option ใน ls ได้ เช่น ls -l (Long Format)
 - rmdir = Remove Directory (ลบแล้วกู้กลับยากมาก) *ถ้า Folder มีข้อมูลจะลบไม่ได้ ถ้าอยากรื้อ ให้ใช้ rm /ชื่อโฟลเดอร์ -r เช่น rm>Hello -r เป็นต้น (ໄລ่ลบทุกไฟล์ใน Folder)
 - cd = ย้าย Working Directory เช่น cd Hello → Bashtest/Hello เป็นต้น
 - ตอนที่เราพิมพ์ชื่อไฟล์ 2 ตัวแรก แล้วกด Tab จะระบบจะเติมชื่อไฟล์ให้เต็มโดยอัตโนมัติ ถ้าชื่อยังไม่ตรง สามารถกด Tab ไปเรื่อยๆ จนเจอนามไฟล์ที่ต้องการได้
-

Into the Tidyverse

- ใน Posit Cloud จะมี Terminal ให้เราใช้ Bash ได้ (เพราะโปรแกรมรันอยู่บน Linux Server)
- เราสามารถรัน R Script ใน Posit Cloud ด้วย Bash ได้ เช่น /cloud/projects\$ Rscript hello.R เป็นต้น

```
#Load Tidyverse, sqldf, and glue
library(tidyverse)
library(sqldf)
library(glue)

#Use glimpse, head, and tail to review data frame
#glimpse = check data type in columns
glimpse(mtcars)
head(mtcars, 3)
tail(mtcars, 3)

#Run SQL query in R
#You can assign the result of query to the variable
sqldf("SELECT * FROM mtcars WHERE mpg > 30")
df <- sqldf("SELECT mpg, wt, hp
            FROM mtcars
            WHERE wt < 2")
sqldf("SELECT am, avg(mpg), sum(mpg)
      FROM mtcars
      GROUP BY am")

#Glue = String template
my_name = "Poorin"
my_age = 23
```

```

glue("My name is {my_name} + I am {my_age} years old.")

#Dplyr = Data Transformation [Inspired by SQL]
#1. Select
#2. Filter
#3. Mutate
#4. Arrange
#5. Summarize + group_by

#Select
select(mtcars, mpg, hp, wt, am)
select(mtcars, contains("a"))
select(mtcars, 1, 3, 5)
select(mtcars, 1:5, am)
select(mtcars, mpg:disp)

#Select w/Pipe Operator
car_mpg30_hp100 <- mtcars %>%
  select(mpg, hp, wt) %>%
  filter(mpg > 30 & hp > 100) %>%
  rownames()

#Filter
mtcars %>%
  select(mpg, hp, wt, am) %>%
  filter(mpg > 30 | am == 1) %>%
  filter(mpg < 20)

mtcars %>%
  rownames_to_column() %>%
  select(model = rowname,
         miles_per_gallon = mpg,
         horsepower = hp,
         weight = wt) %>%
  head()

mtcars <- mtcars %>%
  rownames_to_column() %>%
  rename(model = rowname)

#Filter model names (Filter with condition)
mtcars %>%
  select(model, mpg, hp, wt) %>%
  filter(grepl("^M", model))

#Mutate (Create new column)
df <- mtcars %>%
  select(model, mpg, hp) %>%
  mutate(mpg_double = mpg * 2,
         mpg_log = log(mpg),
         hp_double = hp * 2)

#Create Label (am 0 = Auto, 1 = Manual)
mtcars <- mtcars %>%
  mutate(am = ifelse(am == 0, "Auto", "Manual"))

#Arrange (Sort data)
mtcars %>%
  select (model, mpg, am) %>%
  arrange(desc(mpg)) %>%

```

```

head(10)

mtcars %>%
  select (model, mpg, am) %>%
  arrange(am, desc(mpg)) %>%
  head(10)

#Create data frame from scratch
df <- data.frame(
  id = 1:5,
  country = c("Thailand", "Korea", "Japan", "Belgium", "USA")
)

df %>%
  mutate(region = case_when(
    country %in% c("Thailand", "Korea", "Japan") ~ "Asia",
    country == "USA" ~ "America",
    TRUE ~ "Other Regions"
  ))

df2 <- data.frame(
  id = 6:8,
  country = c("Germany", "Italy", "Spain")
)

#Append data frame
full_df <- df %>% bind_rows(df2)

df3 <- data.frame(
  id = 9:10,
  country = c("Canada", "Malaysia")
)

full_df <- df %>%
  bind_rows(df2) %>%
  bind_rows(df3)

list_df <- list(df, df2, df3)
full_df <- bind_rows(list_df)

full_df %>%
  mutate(region = case_when(
    country %in% c("Thailand", "Japan", "Korea", "Malaysia") ~ "Asia",
    country %in% c("Canada", "USA") ~ "America",
    TRUE ~ "Europe"
  ))

#Case when in SQL
sqldf("select *, case
      when country in ('USA', 'Canada') then 'America'
      when country in ('Thailand', 'Korea', 'Japan', 'Malaysia') then 'Asia'
      else 'Europe'
      end as region
      from full_df")

#Case when in R
full_df %>%
  mutate(region = case_when(
    country %in% c("Thailand", "Korea", "Japan", "Malaysia") ~ "Asia",
    country %in% c("Canada", "USA") ~ "America",

```

```

TRUE ~ "Europe"))

#Summarize + Group By
#Either summarize or summarise is OK
mtcars %>%
  group_by(am) %>%
  summarise(avg_mpg = mean(mpg),
            sum_mpg = sum(mpg),
            min_mpg = min(mpg),
            max_mpg = max(mpg),
            n = n())

result <- mtcars %>%
  mutate(vs = ifelse(vs==0, "v-shaped", "straight")) %>%
  group_by(am, vs) %>%
  summarise(avg_mpg = mean(mpg),
            sum_mpg = sum(mpg),
            min_mpg = min(mpg),
            max_mpg = max(mpg),
            n = n())
View(result)
write_csv(result, "result.csv")
df <- read_csv("result.csv")

#Missing Value (NA = Not Available)
v1 <- c(5, 10, 15, NA, 25)

#NA check by using is.na()
is.na(v1)

data("mtcars")

mtcars[5, 1] <- NA

#Filter NA
mtcars %>%
  filter(is.na(mpg))

#Filter complete case
mtcars %>%
  select(mpg, hp, wt) %>%
  filter(!is.na(mpg))

mtcars %>%
  filter(!is.na(mpg)) %>%
  summarise(avg_mpg = mean(mpg))

mtcars %>%
  summarise(avg_mpg = mean(mpg, na.rm = TRUE))

mean_mpg <- mtcars %>%
  summarise(mean(mpg, na.rm = TRUE)) %>%
  pull()

mtcars %>%
  select(mpg) %>%
  mutate(mpg2 = replace_na(mpg, mean_mpg))

#Looping over data frame
data(mtcars)

```

```

#For other programming language, use for loop
for(i in 1 : ncol(mtcars)) {
  print(mean(mtcars[[i]]))
}

#In R, using apply() to loop over data frame:
#Apply mean to all columns in mtcars
apply(mtcars, 2, mean)

apply(mtcars, 2, sum)

#JOIN data frame
#JOINS In SQL: INNER, LEFT, RIGHT, FULL

band_members
band_instruments

left_join(band_members, band_instruments, by = "name")
#OR
band_members %>%
  left_join(band_instruments, by = "name")

band_members %>%
  inner_join(band_instruments, by = "name")

band_members %>%
  full_join(band_instruments, by = "name")

band_members %>%
  rename(member_name = name) -> band_members_2

band_members_2 %>%
  left_join(band_instruments, by = c("member_name" = "name"))

#Use larger library
library(nycflights13)

glimpse(flights)

flights %>%
  filter(year == 2013 & month == 9) %>%
  count(carrier) %>%
  arrange(-n) %>%
  head(5) %>%
  left_join(airlines, by = "carrier")

glimpse(airlines)

```

Web Scraping

```

#load rvest and tidyverse
library(rvest)
library(tidyverse)

```

```

url <- "https://www.imdb.com/search/title/?groups=top_100&sort=user_rating.desc"

#Static Website (Ex.: Wikipedia)

movie_name <- url %>%
  read_html() %>%
  html_elements("h3.lister-item-header") %>%
  html_text2()

ratings <- url %>%
  read_html() %>%
  html_elements("div.ratings-imdb-rating") %>%
  html_text2()

votes <- url %>%
  read_html() %>%
  html_elements("p.sort-num_votes-visible") %>%
  html_text2()

imdb_df <- data.frame(
  movie_name,
  ratings,
  votes
)

imdb_df %>%
  separate(votes, sep=" \| ", into=c("votes",
                                    "gross",
                                    "tops")) %>%
  View()

specphone_url <- "https://specphone.com/Samsung-Galaxy-S23-5G.html"

specphone_url %>%
  read_html() %>%
  html_elements("div.topic") %>%
  html_text2()

specphone_url %>%
  read_html() %>%
  html_elements("div.detail") %>%
  html_text2()

```

Homework:

1. IMDB web scraping ຈໍາຍ ໄ
2. ລອງຫາ static web ເພື່ອດຶງຂ້ອມູລອອກມາເລີນ ໄ ເຊັ່ນ Medium, Specphone

```

#Homework 1: IMDB web scraping
library(rvest)
library(tidyverse)

#Most popular video games sorted by user rating (desc)
imdb_url <- "https://www.imdb.com/search/title/?title_type=video_game&sort=user_rating,desc"

game_name <- imdb_url %>%
  read_html() %>%
  html_elements("h3.lister-item-header") %>%
  html_text2()

ratings <- imdb_url %>%
  read_html() %>%
  html_elements("div.ratings-imdb-rating") %>%
  html_text2()

votes <- imdb_url %>%
  read_html() %>%
  html_elements("p.sort-num_votes-visible") %>%
  html_text2()

imdb_df <- data.frame(
  game_name,
  ratings,
  votes
)

View(imdb_df)

#Homework 2: Static Web Scraping
library(rvest)
library(tidyverse)

#Notebook spec
nspec_url <- "https://notebookspec.com/chart/notebook-cpu.html"

model_name <- nspec_url %>%
  read_html() %>%
  html_elements("div.model.field") %>%
  html_text2()

base_clock <- nspec_url %>%
  read_html() %>%
  html_elements("div.baseclock.field") %>%
  html_text2()

turbo_clock <- nspec_url %>%
  read_html() %>%
  html_elements("div.turbo.field") %>%
  html_text2()

NBS_score <- nspec_url %>%
  read_html() %>%
  html_elements("div.score.field") %>%
  html_text2()

notebook_df <- data.frame(

```

```
model_name,  
base_clock,  
turbo_clock,  
NBS_score  
)  
  
notebook_df_new <- notebook_df[-1, ]  
  
rownames(notebook_df_new) <- NULL  
  
View(notebook_df_new)  
  
write_csv(notebook_df_new, "notebook_df.csv")
```

Homework in Colab:

[https://colab.research.google.com/drive/1rOyY_sH3XIo0I70Hh6yjBwnxUUEeiYJR?
usp=sharing](https://colab.research.google.com/drive/1rOyY_sH3XIo0I70Hh6yjBwnxUUEeiYJR?usp=sharing)