

Intro to Data Transformation

Tags	Data Transformation	R
Class		
Finished Yet?	<input checked="" type="checkbox"/>	
Knowledge	 The Fourth Sprint: Data Transformation	

Core Data Analyst Skills

- ก้าวแรกที่สำคัญของ Data Analyst คือการเปลี่ยน Data Frame ให้อยู่ในรูปแบบที่เราต้องการ

- ใน R ใช้ dplyr จาก Tidyverse มี 5 Function หลัก ๆ ประกอบด้วย:

1. select()
2. filter()
3. mutate()
4. arrange()
5. summarise() หรือ summarize() [เขียนได้สองแบบ]

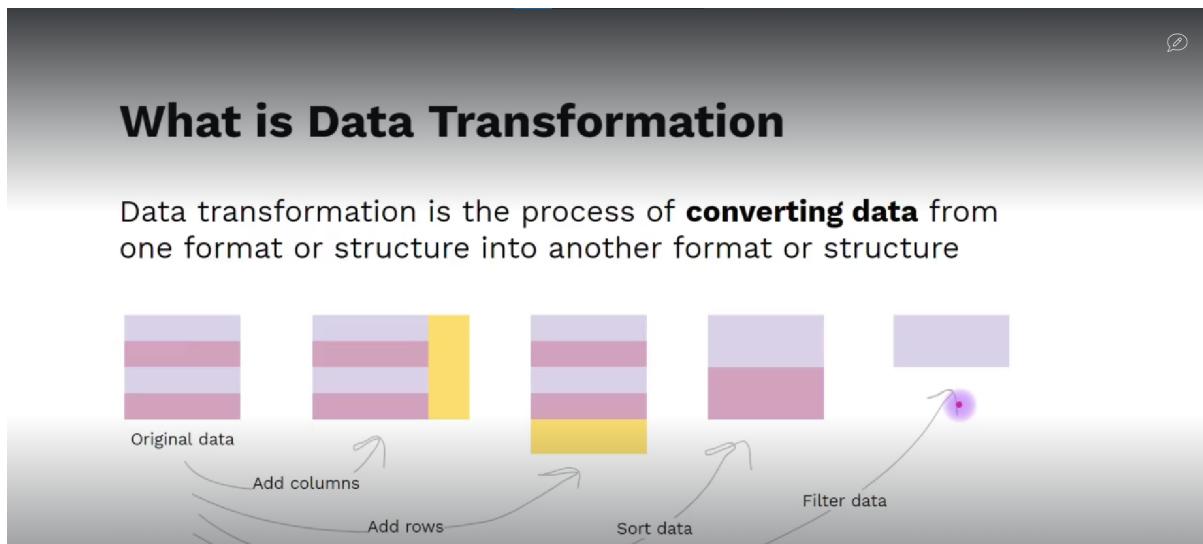
- Typical Workflow:

1. ดึงข้อมูลจาก SQL databases หรือ Data Format ต่าง ๆ เข้าสู่ R

2. เขียน dplyr เพื่อจัดการ Data Frame ด้วยวิธีการต่าง ๆ เช่น merge, join, union, transform เป็นต้น
 3. ส่งข้อมูลที่ได้รับการจัดรูปแบบแล้วให้ Users ของเรา (เช่น .csv, .xlsx, .json) หรือส่งไปให้ software อื่น ๆ ใช้งานต่อ เช่น Power BI, Tableau, Google Sheets, Data Studio
-

Lesson 1: Intro to Data Transformation

-Data Transformation คือการเปลี่ยน (Convert) รูปแบบข้อมูลจากรูปแบบหนึ่งไปเป็นอีกรูปแบบหนึ่ง เช่น การเพิ่ม rows และ/หรือ columns, การ Sort Data, ไปจนถึงการ Filter Data



```
#Install dplyr
install.packages("dplyr")
#Load dplyr
library(dplyr)
```

Core Functions in dplyr:

Functions	What it does? (EN)	What it does? (TH)
select()	select column	เลือกคอลัมน์
filter()	filter column with criteria	กรองข้อมูลด้วยเงื่อนไข
mutate()	create new column	สร้างตัวลัมบ์ใหม่
arrange()	arrange data	เรียงข้อมูล
summarise()	summarise as statistical data (aggregation)	สรุปผลสถิติ

Functions	What it does? (EN)	What it does? (TH)
group_by()	grouping data	จับกลุ่มข้อมูล

Lesson 2: Read CSV Files

- เราสามารถอ่านไฟล์ประเภท .csv ได้ด้วย `read.csv()` เป็น Base R Function มีอักษรหนึ่ง คือ การโหลด Library ซึ่ง `readr` และใช้ Function ซึ่ง `read_csv()` เช่น:

```
#Install readr
install.packages("readr")
#Load Library
library(readr)
#Read filename.csv
read_csv("filename.csv")
```

- ถ้าเราไม่อยากเปลี่ยนตัวอักษร (String) เป็น Factor ให้เช็ตแบบนี้:

```
imdb <- read.csv("imdb.csv", stringsAsFactors = FALSE)
```

- หลังจากที่เราเก็บค่าไว้ในตัวแปรเรียบร้อย เราสามารถเปิดไฟล์เป็น Data Frame ได้ด้วย `View()` ดังนี้:

```
View(imdb)
```

- `glimpse()` จะ Review ตัว Structure ของ Dataframe ให้เรารู้คร่าว ๆ

```
> glimpse(imdb)
Rows: 100
Columns: 7
$ NO          <int> 1, 2, 3, 4, 5...
$ MOVIE_NAME <chr> "The Shawshan...
$ YEAR        <int> 1994, 1972, 2...
$ RATING      <chr> "R", "R", "PG...
$ LENGTH      <int> 142, 175, 152...
$ GENRE       <chr> "Drama", "Cri...
```

-เราสามารถแสดงผล row ตัวบ ฯ และ row ก าวย ฯ ของ Data Frame ได้ด้วย head() และ tail()
โดยที่จะแสดงผล 6 row แรก และ 6 row สุดก าวย ตามลำดับ

```
> head(imdb)
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
```

	NO
1	1
2	2
3	3
4	4
5	5
6	6

```
> tail(imdb)
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
#> #> #> #> #> #>
```

	NO
95	95
96	96
97	97
98	98
99	99
100	100

-ถ้าอยากร ะให้แสดงผลทุก ๆ x row แรก หรือ x row สุดก าวย ให้พิมพ์ head(imdb, 10) หรือ tail(imdb, 10)

Lesson 3: Select Columns

-dplyr มา กับ Pipe Operator %>% ใช้ในการ เชื่อม Function ของเราร าได้หลาย Step เช่น:

```
df %>% select() %>% filter() %>% mutate() %>% arrange()
```

- เราสามารถเลือกข้อมูลเฉพาะจาก column ที่ต้องการได้ด้วย select() เช่น:

```
#Select MOVIE_NAME and RATING columns from imdb table  
select(imdb, MOVIE_NAME, RATING)  
#Select by using column index:  
select(imdb, 1, 5)
```

- เราสามารถเปลี่ยนชื่อ column ได้ดังนี้:

```
select(imdb, movie_name = MOVIE_NAME, release_year = YEAR)
```

- ตัวอย่างการใช้ %>%% ใน การสร้าง Data Pipeline:

```
#Example  
head(imdb)  
#Equals to  
imdb %>% head()  
  
#Another Example:  
select(imdb, movie_name = MOVIE_NAME, release_year = YEAR)  
#Equals to  
imdb %>% select(movie_name = MOVIE_NAME, release_year = YEAR)  
  
#Select MOVIE_NAME as movie_name, YEAR as release_year, and show first 10 columns  
imdb %>%  
  select(movie_name = MOVIE_NAME, release_year = YEAR) %>%  
  head(10)
```

Lesson 4: Filter Data Part 1

- เราสามารถกรองข้อมูลได้ด้วย filter() เช่น:

```
#Rows with Score >= 9.0 Only  
filter(imdb, SCORE >= 9.0)
```

- เราสามารถเรียกดูชื่อของ columns ใน Data Frame ได้ด้วย names() เช่น:

```
names(imdb)
```

- เราสามารถเปลี่ยนตัวอักษรให้เป็นตัวพิมพ์เล็กทั้งหมดได้ด้วย `tolower()` เช่น:

```
names(imdb) <- tolower(names(imdb))

#Select and Filter
imdb %>%
  select(movie_name, year, score) %>%
  filter(score >= 9 & year > 2000)

#Filter with %in%
imdb %>%
  select(movie_name, length, score) %>%
  filter(score %in% c(8.3, 8.8, 9.0))
```

- & คือ AND ส่วน | คือ OR และ == คือ ‘เท่ากับ’

Lesson 5: Filter Data Part 2

- เราสามารถกรองด้วยค่าที่เป็น String ได้เช่นกัน เช่น:

```
imdb %>% filter(rating == "R")
```

- `grepl()` เป็น Function พิเศษของ R ใช้ในการหา Pattern เช่น:

```
grepl("Drama", imdb$genre)

imdb %>%
  select(movie_name, genre, rating) %>%
  filter(grepl("Drama", imdb$genre))

imdb %>%
  select(movie_name) %>%
  filter(grepl("The", imdb$movie_name))
```

*`grepl()` เป็น Function ที่ Case Sensitive

Lesson 6: Create New Columns

- เราสามารถสร้าง column ใหม่ได้ด้วย mutate() เช่น:

```
#Select then create new columns
imdb %>%
  select(movie_name, score, length) %>%
  mutate(score_group = if_else(score >= 9, "High Rating", "Low Rating"),
  length_group = if_else(length >= 120, "Long Film", "Short Film"))
```

- เราสามารถ Update ข้อมูลได้ด้วย column_update() ในวงเล็บ mutate() เช่น

```
#Update score
imdb %>%
  select(movie_name, score) %>%
  mutate(score_update = score + 0.1) %>%
  head(10)
```

Lesson 7: Arrange Data

- เราสามารถเรียงค่าใน column มากจากไปน้อยได้ด้วย desc() ในวงเล็บ arrange() เช่น:

```
imdb %>%
  arrange(rating, desc(length)) %>%
  head(10)
```

*ถ้าไม่ครอบ desc() ระบบจะเรียงค่าจากน้อยไปมากให้โดยอัตโนมัติ

Lesson 8: Summary Statistics

- เราสามารถใช้ Summarise() ในการหาค่าทางสถิติของ Data Frame ของเราได้ (Aggregation) เช่น:

```
#Finding mean, sd, min, max, and count of length
imdb %>%
  summarise(mean_length = mean(length),
  sd_length = sd(length),
```

```
min_length = min(length),  
max_length = max(length),  
n = n())
```

- เราสามารถใช้ `group_by()` เพื่อจัดกลุ่มของข้อมูลตาม column ที่เราต้องการได้ เช่น:

```
imdb %>%  
  filter(rating != "") %>%  
  group_by(rating) %>%  
  summarise(mean_length = mean(length),  
            sd_length = sd(length),  
            min_length = min(length),  
            max_length = max(length),  
            n = n())
```

* เราสามารถ `group_by()` ได้มากกว่า 1 column

Lesson 9: Join Tables

- เราสามารถ JOIN TABLE ใน R ได้ด้วย `inner_join()` เช่น:

```
favourite_films <- data.frame(id = c(5, 10, 25, 30, 98))  
  
favourite_films %>%  
  inner_join(imdb, by = c("id" = "no"))
```

Lesson 10: Export CSV File

* Function `write.csv()` เป็นของ Base R ใน Library `readr` จะมี `write_csv()` เป็น Function ที่ใหญ่กว่า ทำงานไว้กว่าเล็กน้อยและไม่มีพฤติกรรมแปลง ๆ เหมือนของ Base R แต่ผลลัพธ์ที่ได้ไม่ต่างกัน

- เราสามารถ Export Data Frame ของเราเป็นไฟล์ `.csv` ได้ด้วย `write.csv` เช่น:

```
#Create imdb_prep  
imdb_prep <- imdb %>%  
  select(movie_name, released_year = year, rating, length, score) %>%  
  filter(rating == "R" & released_year > 2000)
```

```
#Export imdb_prep  
write.csv(imdb_prep, "imdb_prep.csv", row.names = FALSE)
```

-การ Export ไฟล์ออกจาก Directory ของเรา ให้ทำการเช็คที่กล่องหน้าไฟล์ และคลิก more → export → download ตามลำดับ

*เราสามารถหาได้ว่า Function ไหนอยู่ใน Base R หรือ Library อื่น ๆ ได้ด้วย
help("function_name") เช่น:

```
help("glue")
```