Slide 01: Data

Richard Dien Giau Bui

Load packages

- First, before a code session, think which packages you need to use today
  - Like before cooking, which knifes we should use
- Load it with `library(name_of_package)`:

```
1 library(tidyverse)
2 library(skimr)
```

Code block

- Insert a new code block with **CTRL+ALT+I** then run with the green **RUN** button

```
60 + 8
```

```
[1] 68
```

Syntax of R language

- Like any language, R also has their own syntax:

```
1 OUTPUT = FUNCTION(INPUT, OTHER_OPTIONS)
```

- For example: we take function `sqrt` for number 4 (as input) and the results will be saved to new object `y` . How much is `y` ?

Comments in R

- Some notes to our codes
- Not run and just give information

```
[1] 9
[1] 9
```

Four main data types in R

- `numeric`: for example, counting, km, scores, money, …
- `character`: for example, “Wednesday”, “Today I failed the coding class”, …
- `factor`: e.g., a column with only 4 possible categories: A, B, C, D
- `date`: `2024-02-28`

Type 1. numeric

- Create a numeric variable

```
1 x = c(1:10) # other way: x <- c(1:10)
2 typeof(x)
```

```
[1] "integer"
```

```
1 x = c(0.1, 0.4, 0.5)
2 typeof(x)
```

```
[1] "double"
```

- Some math manipulation with numeric

```
1 length(x)
```

```
[1] 3
```

```
1 mean(x)
```

```
[1] 0.3333333
```

```
1 median(x)
```

```
[1] 0.4
```

```
1 min(x)
```

```
[1] 0.1
```

```
1 max(x)
```

```
[1] 0.5
```

- Quick way:

```
1 skim(x)
```

Type 2: `character`

- For text data
- Can't use with math calculation, for example, below code will raise error:

```
1 "60" + "8"
```

- Create text vector

```
1 s = c("ann", "betty", "candy")
```

- Some text manipulation with `stringr` package:

```
1 length(s)
```

```
[1] 3
```

```
1 str_length(s)
```

```
[1] 3 5 5
```

```
1 str_to_upper(s)
```

```
[1] "ANN"   "BETTY" "CANDY"
```

```
1 str_replace(string = s, pattern = "a", replacement = "$")
```

```
[1] "$nn"   "betty" "c$ndy"
```

```
1 paste0("student_", s)
```

```
[1] "student_ann"   "student_betty" "student_candy"
```

Type 3: `factor`

- Some category only, not continuous numeric variable
- For example:

```
1  f1 = c(0, 1, 1, 0 ,1)
2  f2 = c("Monday", "Tuesday", "Wednesday", "Tuesday", "Friday")
```

- Sometimes, we need `factor` to do some special manipulation, e.g., plotting data by groups

Type 4: `date`

- Make date:

```
d = c(20211231, 20221231, 20231231)
typeof(d)
```

```
[1] "double"
```

- Convert `numeric` in yyyymmdd format to `date`:

```
d = ymd(d)
typeof(d)
```

```
[1] "double"
```

Missing data: NA

- Sometimes, we do not have data for some cases in the data
- In data science, people call it missing data
- In R, it is denoted as NA (pronounced as “Non-Applicable”)

Vector vs dataframe

- Vector: One column of data
  - E.g., `x = c(5,7,9)` is a vector with three elements, where the first element can be index/choose by `x[1]`
- Dataframe: Many columns put together in the same table
  - A dataframe has rows/observations and columns/variables, such as `Data[rows, columns]`
  - To select first 5 rows rows: `Data[1:5,]`
  - To select column 2 and 3: `Data[,c(2,3)]`

Example

```
1 mtcars[,1]
```

```
 [1] 21.0 21.0 22.8 21.4 18.7 18.1 14.3 24.4 22.8 19.2 17.8 16.4 17.3 15.2
10.4
[16] 10.4 14.7 32.4 30.4 33.9 21.5 15.5 15.2 13.3 19.2 27.3 26.0 30.4 15.8
19.7
[31] 15.0 21.4
```

```
1 mtcars[1,]
```

```
          mpg cyl disp  hp drat   wt  qsec vs am gear carb
Mazda RX4  21   6  160 110  3.9 2.62 16.46  0  1    4    4
```

Input dataframe to R

- Use interactive import panel
- Use functions
  - Depending on data, we have corresponding functions
  - For example, `read_csv` to read CSV files to R, `read_excel` to read Excel files (we may need to load libraries to use these functions)

Example

- Read the file `hsb.csv` to R by two ways

Save dataframe from R to a file

- Save output dataframe from R to a file to use later
- Many possible file types: CSV, `RDS`, `parquet`
  - E.g., `saveRDS(OUTPUT, "path_to_file.rds")`, `readr::write_csv(OUTPUT, "path_to_file.csv")`

Some tasks with the dataframe

- Make data smaller
- Make data larger
- Don’t change the data size

Make data smaller

Function	When to use
`select`	Choose a few variables
`filter`	Choose few rows
`group_by` and `summarize`	Summarize some statistics by groups

Make data larger

Function	When to use
`mutate`	Create new variables
`transmute`	Also create new variables, but keep only these new variables and drop all old variables

Don’t change the data size

Function	When to use
`arrange`	Sort data
`rename`	To change variable name

How to use a function?

- Google
  - ChatGPT
- Use `?` before the function name:

```
1 ?mutate
```

- Practice more to remember more functions

Thank you

We are ready to clean a data in the next slide.