Statistics with Recitation: TA Session

Danny Po-Hsien Kang (康柏賢)

September 2, 2025

Today's agenda

1. Introduction to TA Session
2. Brief Introduction to R
3. Download R & RStudio
4. Basics of R
   - RStudio Interface
   - Assign Values
   - Data Types
   - Download Packages
   - Help Function

What will we do in the TA session?

1. Learn how to use R for statistical analysis
    - Basic syntax
    - Data manipulation
    - Data visualization
2. In-class practice
    - Some practices will be prepared to check your understanding in each TA session.
    - Finish them before leaving the classroom.
3. Quizzes
    - 6 quizzes in this semester (check syllabus!).
    - The graded answer sheet will be returned in the TA session in one week.

How to contact me?

1. Sending email
   - Please mail to r13323002@ntu.edu.tw with mail title including "[Statistics TA]".
2. Office hours
   - Tuesday, 4:30-5:20 p.m.
   - SSB. (社會科學院) Room 645.

What is R/RStudio?

- **R** is a free and open-source programming language for **statistical computing** and **data analysis**.
- **RStudio** is an integrated development environment (IDE) for R.

Icon for R

Icon for RStudio

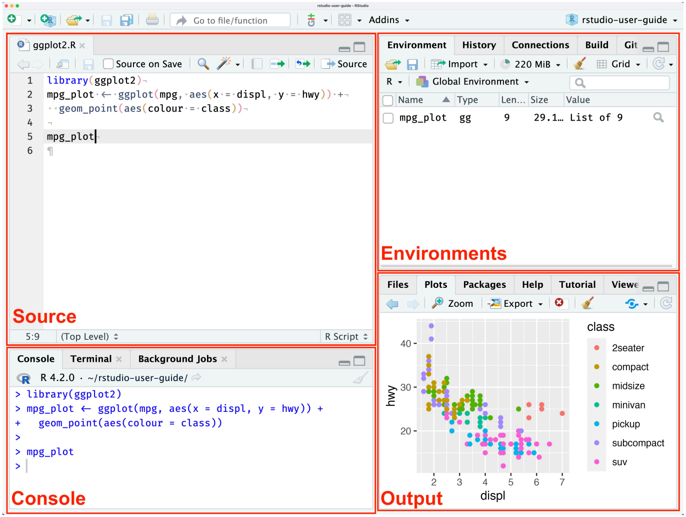
Why choose R?

Aspect	**R**	**Stata**	**Python**
Costs	Free, open-source	Paid	Free, open-source
Primary uses	Data analysis/ Statistics	Data analysis/ Statistics	Various purposes
Learning	Moderate	Easier	Moderate
Typical users	Econ/Stats/Soc. sci.	Econ/Stats/Soc. sci.	Popular in industries

Download R & RStudio

- Download R here.
- Download RStudio here.
- Install R first, and then install RStudio.

Basics of R: RStudio Interface



```
library(ggplot2)
mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
  geom_point(aes(colour = class))

mpg_plot
```

```
> library(ggplot2)
> mpg_plot <- ggplot(mpg, aes(x = displ, y = hwy)) +
+   geom_point(aes(colour = class))
>
> mpg_plot
>
```

Name	Type	Len...	Size	Value
mpg_plot	gg	9	29.1...	List of 9

Basics of R: RStudio Interface

- **Source** (Script): write your code here.
- **Environment**: check your data and objects here.
- **Console**: execute your code here.
- **Files/Plots/Packages/Help**: check your files, plots, packages, and help here.

Basics of R: Assign Values

- <- is the **assignment operator** in R. We use it to assign value to a variable.

```
# example
a <- 3
```

- A value assigned to a variable can be a number, a character, a dataframe, or the result of an expression.

```
# example
b <- a*a
# b = 9
x <- "hello" # x = "hello"
```

Assign Values (Supplement): <- vs =

- Can we use = instead of <-?
    - Sometimes the answer is Yes, but with cautions!
- Below two lines work the same:

```
x <- 5
y = 5
```

- But in function calls, = is used for naming arguments, not assignment.

```
mean(x <- 1:10)
x  ## [1]  1  2  3  4  5  6  7  8  9 10
mean(x = 1:10)
x ## Error: object 'x' not found
```

- To avoid ambiguity, it's better to use <- than = in general.

Basics of R: Data Types

- There are several basic data types in R:
  - **Numeric**: numbers with decimal points.
  - **Character**: text
  - **Logical**: `TRUE` or `FALSE`.
- To check the data type of a variable, use the `class()` function.

```r
# example
 x <- 5
 class(x) # Output: "numeric"
 y <- "Hello World!"
 class(y) # Output: "character"
 z <- TRUE
 class(z) # Output: "logical"
```

Basics of R: Data Types

- To store data in R, we have the following ways:
  - **Vector**: one-dimensional sequence of values of the same data type.
  - **List**: one-dimensional sequence of values (could) with different data types.
  - **Matrix**: two-dimensional structure of values of the same data type.
  - **Data frame**: two-dimensional structure of values (could) with different data types across columns.
  - **Tibble**: modernized version of a data frame used in the `tidyverse` package.

Basics of R: Data Types (Vector)

- Examples of `vector`: use `x:y` or `c()` to create vector

```
id <- 1:4 # 1 2 3 4
class(id)
# Output: "integer"
score <- c(30, 86, 50, 95)
class(score) # Output: "numeric"
name <- c("Sam", "John", "Andy", "Judy")
class(name) # Output: "character"
```

Basics of R: Data Types (List)

- Examples of `list`:

```
my_list <- list("Hello World!", 1, TRUE, score)
```

- Output

```
> print(my_list)
[[1]]
[1] "Hello World!"

[[2]]
[1] 1

[[3]]
[1] TRUE

[[4]]
[1] 30 86 50 95
```

Basics of R: Data Types (Matrix)

- Examples of `matrix(data, nrow=..., ncol=...)`:[1]

```
id <- 1:4
score <- c(30, 86, 50, 95)
id_score <- matrix(c(id, score), nrow = 4)
```

- Output

```
> print(id_score)
     [,1] [,2]
[1,]    1   30
[2,]    2   86
[3,]    3   50
[4,]    4   95
```

[1] Also try what happens when without `nrow` and when `nrow = 2`!

Basics of R: Data Types (Data Frame)

- Examples of `data.frame()`:

```
df <- data.frame(stu_id = id,
                 stu_score = score)
```

- Output

```
> print(df)
  stu_id stu_score
1      1        30
2      2        86
3      3        50
4      4        95
```

Data Types (Supplement): Type Coercion in R Vectors

- R vectors are of the same data type. Mixed types are coerced by hierarchy: `logical` → `numeric` → `character`.

Input	**Code**	**Result**
Logical + Numeric	`c(TRUE, 2)`	`[1] 1 2` (numeric)
Numeric + Character	`c(5, "a")`	`[1] "5" "a"` (character)
Logical + Character	`c(FALSE, "yes")`	`[1] "FALSE" "yes"` (character)
Numeric + Logical + Character	`c(1, TRUE, "hi")`	`[1] "1" "TRUE" "hi"` (character)
Mixed types in a List	`list(1, "a", TRUE)`	`[[1]] 1; [[2]] "a"; [[3]] TRUE`

- Example:

```r
x <- c(1, TRUE, "hi")
class(x) # Output: "character"
```

Basics of R: Download Packages

- To download and install a package, use the `install.packages()` function.

```r
install.packages("tidyverse")
```

- To load a package into your R session, use the library() function.

```r
library(tidyverse)
library(ggplot2)
```

- Add `""` when use `install.packages()` install, but no need when use `library()`.

Basics of R: Help Function

- To get help on a specific function, use the `help()` function or the ? operator to check the documentation for the function.

```
help(mean)
?mean
```

- This will open a help page with information about the function, its arguments, and examples of usage.
- Other types of help function: `help.start()` and `help.search()`.