Module 1: R, Rstudio, and Rmarkdown; Basic data types and structures

Yaqi Shi

07/09/2024

Methods and computing camp

This summer we will together learn and review materials of statistical computing and methods.

What do we do during lectures?

Materials will be available at course website. Lecture notes are created by Rmarkdown.

- **If you have questions, feel free to interrupt or send a message in the chat.**

We will cover 10 modules of statistical methods and computing.

- Each module takes ~1 hour.

What contents we will cover?

Module	Topics	References
1	R, Rstudio, and Rmarkdown Basic data types and structures	-
2	Reporting, data wrangling and graphing (I) LaTeX, tidyverse	-
3	Reporting, data wrangling and graphing (II) Elementary data analysis ggplot and Github	-
4	Probability distributions Statistical inference (I) Fundamental concepts in inference	AoS Chp 1-5 AoS Chp 6
5	Statistical inference (II) Maximum likelihood estimation	C&B Chp 6.3, 7 AoS Chp 3-4
6	Statistical inference (III) Hypothesis testing	AoS Chp 8 C&B Chap 8
7	Statistical models (I) Linear regression models	AoS Chp 13 C&B Chp 11
8	Statistical models (II) Generalized linear models	C&B Chp 12 AoS Chp 13
9	Simulation and parallel computing	C&B Chap 10 AoS Chp 24
10	Bootstrap	AoS Chp 5

Exercises

- Available before each module.
- Exercises can be difficult.
- Group discussion is recommended.
- Solutions will be posted after module.

Module 1: Basic programming in R

We will review R, Rstudio, and Syntax of R together.

- Rstudio (Knit)
- Basic data types
- Basic data structures
- Functions
- For loops

Useful resources:

- Tidyverse style guide
- The R Inferno

Introduction to R and Rstudio

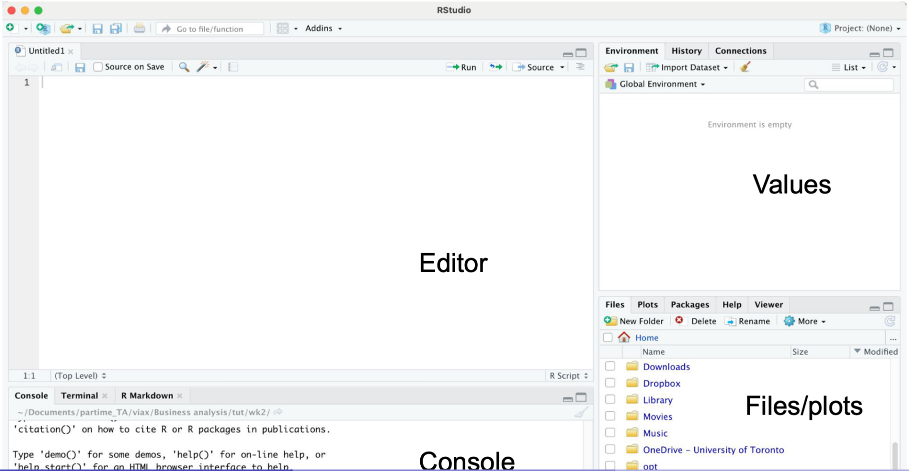
R is a free statistical software. We use R frequently/intensively during our study.

First please download R and its IDE Rstudio (if you haven't).

- https://www.r-project.org/
- https://www.rstudio.com/

Studio

- Editor: edit or save the file.
- Console: outputs.
- Values: store values of assigned.
- Files/plots/packages/help.



```
~/Documents/partime_TA/viax/Business analysis/tut/wk2/
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
```

How to set working directory?

Working directory is important since you might want to read and import data from other files. It is recommended to put these files under the same directory with your scripts.

- Method 1: Session -> set working directory.
- Method 2: Files -> Navigate to your directory.
- Method 3: `setwd()`/`getwd()`.

Alternatives: Set R.project.

How to install packages?

Common packages in statistical analysis with R:

- tidyverse/dpylr
- ggplot1
- kableExtra or gridExtra
- glmnet

Several options to install packages:

- Method 1: Tools -> install packages
- Method 2: Packages window.
- Method 3: `install.packages()`.

Ready with your Rstudio?

Let's code!

Vector

Can contain numerical, string, or Boolean value.

Store data = make an assignment. c() is for component.

```
v <- c(TRUE, FALSE, TRUE, TRUE, FALSE)
v <- c("python", "mathlab", "R")
v <- c(6, 5, 4, 3, 2, 1)
```

Are you familiar with the output of these?

```
which(v == 3)
```

```
## [1] 4
```

```
v[2]
v[-2]
v[2:3]
v[v < 4]
which(v == 3)
```

Matrix

```
mymat <- matrix(c(1:10), nrow = 2, ncol = 5,
                byrow = TRUE)
```

```
mymat[2, ]
```

```
## [1]  6  7  8  9 10
```

```
mymat <- matrix(c(1:10), nrow = 2, ncol = 5,
                byrow = TRUE)
mymat
mymat[1, 5]
mymat[2, ]
mymat[c(1:2), c(1:2)]
```

Matrix

```r
typeof(mymat)
```

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

```r
class(mymat)
```

```
## [1] "matrix" "array"
```

```r
is.matrix(mymat)
```

```
## [1] TRUE
```

```r
dim(mymat)
```

```
## [1] 2 5
```

Data frame

```
studentID <- c(1, 2, 3, 4)
age <- c(17, 18, 16, 19)
gender <-c("M", "F", "M", "M")
studentData <- data.frame(studentID, age, gender)
```

```
rownames(studentData) <- c("A", "B", "C", "D")
```

```
colnames(studentData) <- c("ID", "age", "gender")
studentData
```

```
##   ID age gender
## A  1  17      M
## B  2  18      F
## C  3  16      M
## D  4  19      M
```

Data frame

```
studentData[1,  ]
```

```
##   ID age gender
## A  1  17      M
```

```
studentData[ , 2]
```

```
## [1] 17 18 16 19
```

```
rownames(studentData)
```

```
## [1] "A" "B" "C" "D"
```

```
str(studentData)
```

```
## ’data.frame’:    4 obs. of  3 variables:
##  $ ID    : num  1 2 3 4
##  $ age   : num  17 18 16 19
##  $ gender: chr  "M" "F" "M" "M"
```

List

model fitting output

linear regression: $y_i = \beta x_i + \epsilon_i$

```r
x <- c(1:5)
eps <- rnorm(5)
y <- 2*x + eps
```

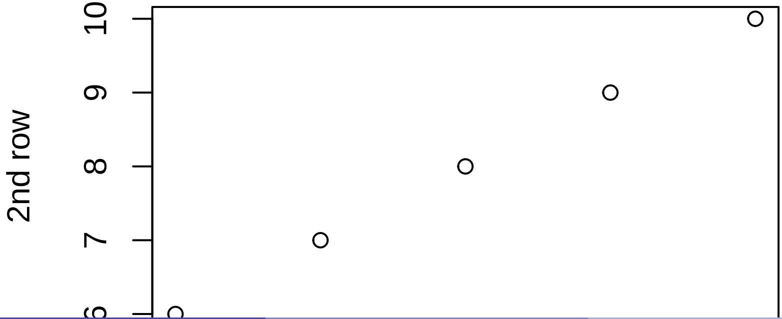
```r
mod <- lm(y ~ x)
summary(mod)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##        1        2        3        4        5
##  0.54674 -0.27591 -1.02200  0.68474  0.06642
##
```

Simple plotting

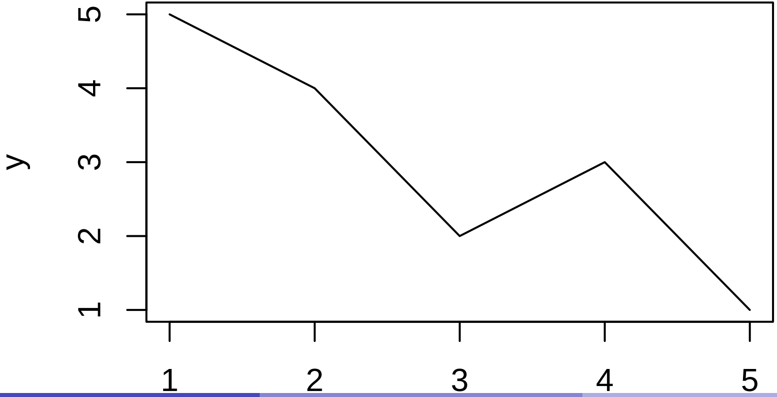
- For "quick and dirty" plots, use **plot**
- For more advanced and attractive data visualizations, use **ggplot**

```r
plot(mymat[1, ], mymat[2, ], ylab="2nd row", xlab="1st row")
```



Simple plotting

```r
y <- c(5, 4, 2, 3, 1)
x <- c(1, 2, 3, 4, 5)
plot(x, y, type = "l")
```



Special characters in R

- **NA**: Not Available (i.e. missing values)
- **NaN**: Not a Number (e.g. 0/0)
- **Inf**: Infinity
- **-Inf**: Minus Infinity. For instance 0 divided by 0 gives a NaN, but 1 divided by 0 gives Inf

```
0/0
```

```
## [1] NaN
```

```
1/0
```

```
## [1] Inf
```

Other

Important for stats research.

- `floor(v)`, `ceiling(v)`
- `round(v, 2)`
- `rnorm()`, `rexp()`, `rbinom()`, etc generate random variable

```
floor(2.333333)
```

```
## [1] 2
```

```
ceiling(2.333333)
```

```
## [1] 3
```

```
round(2.333333, 2)
```

```
## [1] 2.33
```

```
rnorm(10)
```

```
##  [1] -0.68574579  1.50341759 -0.80660364 -0.03429308  0.73939655  0.65084975
##  [7]  1.07780203 -0.48943324 -0.36698526  0.32909995
```

Reading and writing data

- read.table/read.csv; write.table/write.csv
- data.table::fread
- readRDS; saveRDS

Function

```r
example_function <- function(x1, x2) {
   y <- 3 * x1^2 + 3 * x2^2 - 2* x1^2 *x2
   return(y)
}
example_function(1, 2)
```

```
## [1] 11
```

If else

What is the output?

```r
p <- 3
if (p <= 2) {
  print("p <= 2!")
} else {
  print("p > 2!")
}
```

for loop

```
v <- c(1, 2, 4, 3)
w <- c(0, 0, 0, 0)
t <- 0

for (i in v) {
  t <- t + 1
  w[t] <- w[t] + i
  print(w)
}
```

```
## [1] 1 0 0 0
## [1] 1 2 0 0
## [1] 1 2 4 0
## [1] 1 2 4 3
```

```
w
```

```
## [1] 1 2 4 3
```

while loop

```r
i <- 1
while (i <= 10) {
  print(i)
  i <- i + 1
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
```

next

```r
alphabet <- LETTERS[1:6]
for (i in alphabet) {
  if(i == 'D') {
    next
  }
 print(i)
}
```

```
## [1] "A"
## [1] "B"
## [1] "C"
## [1] "E"
## [1] "F"
```

break

```r
alphabet <- LETTERS[7:12]
for (i in alphabet) {
  if (i == 'K') {
    break
  }
  print(i)
}
```

```
## [1] "G"
## [1] "H"
## [1] "I"
## [1] "J"
```

apply

In R, we typically use `apply()` instead of for loop. It can be applied on matrix, vector, data frame, and loop through row or column (defined by the second input - `MARGIN`).

Syntax: `apply(X, MARGIN, FUN, ...)`

```r
f <- function(x) {
  ts <- 2*x^2
  return(ts)
}

ii <- matrix(1:4, nrow = 1)
apply(ii, 1, f)
```

```
##      [,1]
## [1,]    2
## [2,]    8
## [3,]   18
## [4,]   32
```

Knit in Rstudio

Common types of R files: `.R`, `.Rmd`

- Simulations, e.g. for loops, functions, I use `.R`
- Reporting, analysis, plotting, I use `.Rmd`

Rmd file can be converted to pdf, html through `Knit`.

- yaml style.

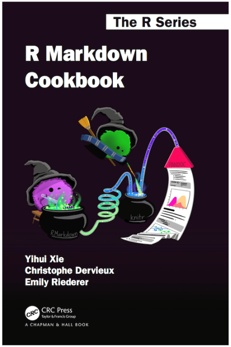
```
---
title: 'Module 1: Basic programming in R'
date: "04/15/2022"
output:
  beamer_presentation
---
```

More yaml style

```yaml
---
title: "A summary of xx"
date: "02/14/2022"
output:
  pdf_document:
    toc: true
    number_sections: true
---
```

Resource

- “R Markdown Cookbook” by Yihui Xie.
- https://bookdown.org/yihui/rmarkdown-cookbook/



Code style

- Google's R Style Guide
- `styler` software embedded in Rstudio - allows you to interactively restyle selected text, files, or entire projects.
- `Lintr` software embedded in Rstudio - performs automated checks to confirm that you conform to the style guide.

Exercise

Available on course website.

1. Matrix and vector operations - generate a 100×100 matrix for matrix inverse calculation.
2. For loops and plotting - understand for loops by plotting a complex polygon.