Lecture Note for Applied Econometrics

Yuta Toyama

Last updated: 2019/04/16

Chapter 1

Preface

Welcome to Applied Econometrics using R!

About this

This lecture note is maintained by Yuta Toyama.

Acknowledgement (as of April 10, 2019)

- Chapter 2 and 3 are largely based on Applied Statistics with `R`. https://daviddalpiaz.github.io/appliedstats/

Chapter 2

Introduction to the course

Placeholder

What is econometrics?

Why do we need to learn computation

Why do we use R?

Getting Started

Helps

Quick tour of Rstudio

Basic Calculations

Addition, Subtraction, Multiplication and Division

Exponents

Mathematical Constants

Logarithms

Trigonometry

Getting Help

Installing Packages

Chapter 3

Data and Programming

Placeholder

Data Types

Data Structures

Vectors

Basics of vectors

Useful functions for creating vectors

Subsetting

Vectorization

Logical Operators

Short exercise

More Vectorization

Matrices

Basics

Matrix calculations

Exercise

Getting information for matrix

Calculations with Vectors and Matrices

Lists

Data Frames

Programming Basics -Control flow-

if/else

`for` loop

Functions

Data frame

Chapter 4

Problem Set 1

- Due date: April 22th (Monday) 11pm

Rules for Problem Sets

- If you are enrolled in Japanese class (i.e., Wednesday 2nd), you can use both Japanese and English to write your answer.
- Submit your solution through `CourseN@vi`.
- Submit both your answer and R script.
- Using `Rmarkdown` would be appreciated, though not mandatory.
    - `Rmarkdown` introduction in Japanese: https://kazutan.github.io/kazutanR/Rmd_intro.html
    - `Rmarkdown` introduction in English: https://rmarkdown.rstudio.com/articles_intro.html
- I might cover `Rmarkdown` in the course later.

Question: Examine the law of large numbers through numerical simulations

Consider the random sample of $\{x_i\}_{i=1}^N$ drawn from the random variable X. The law of large numbers implies that

$$\frac{1}{N}\sum_{i=1}^{N} x_i \xrightarrow{p} E[X]$$

In other words, the sample mean converges to the population mean in probability as the sample size goes to infinity (i.e., $N \to \infty$).

Similary, the sample variance also converges to the population variance in probability

$$\frac{1}{N}\sum_{i=1}^{N} (x_i - \bar{x})^2 \xrightarrow{p} V[X]$$

(This is an application of the law of large numbers, though it is a bit involved to prove this.)

The goal of this problem set is to demonstrate these two properties through numerical simulations. Here is what we are going to do:

1. For a certain sample size N, draw N random numbers from the normal distribution with known mean and standard deviation.

2. Calculate the sample mean and the sample variance for the "data" you draw.
3. Repeat this for many different sample sizes.
4. Examine to see whether the sample mean and the variance are getting closer to the true value, which you set when you draw the random numbers, as the sample size gets larger.

I explain how to implement this in `R` step by step below.

1. Prepare a function like this
    1. There are two inputs: (1) a vector that contains the data $ { x_i }_{i=1}^{N} $ and (2) the indicator of whether you calculate the mean or the standard deviation.

```
fun_something = function(firstinput, secondinput){

  # Two inputs: firstinput, secondinput
  # One output: output

  # Do something.

  return(output)

}
```

    2. Use if/else sentence. Example:

```
# "secondinput" is the name of the input variable in your function
if ( secondinput == "mean"){
  ## calculate mean of the data (firstinput)

} else if ( secondinput == "sd")
  ## Calculate standard deviation of the data (firstinput)
}
```

    3. Use `return` function to define the output of the function.
2. Construct a vector that contains the sample size you want to use in your simulation. For example:

```
samplesize_vec = c(100, 200, 300, 400)
```

Here, let's try 100 different sample sizes that ranges from 100 to 50000. (Hint: `rep` function)

3. Prepare two vectors that contain the result in the `forloop` below. Since we are trying 100 different sample sizes, let's create a vector with the length of 100.

```
# Hint:
# numeric(k) returns a zero vector with the length of k
# length( vector) returns the length of `vector`

# result_mean = ....
# result_variance = ....
```

4. To create the random draw from the normal distribution, use below

```
# You can choose the mean and the standard deviation as you like.
rnorm(n = 100, mean = 2, sd = 5)
```

4. Use `forloop` to calculate both mean and the standard deviation for each sample size. For example:

```
for (i in 1:length(samplesize_vec)){

  # Draw the random number

  # Calculate the mean using the function you construct.
  # Calculate the variance using the function you construct.

}
```

5. Plot the result with `ggplot2`.
    1. Install the package if you have not done it yet.
    2. Load `ggplot2` by `library(ggplot2)`
    3. Use `qplot` command to make a figure

```
# Create plot and save it as the variable `plot1`
plot1 <- qplot(x = samplesize_vec, y = yourresult, geom = "line")

# print "plot1"
print(plot1)

# save the plot as PNG file
ggsave(file = "filename.png", plot = plot1)
```