DSL Seminar: MCMC (1)

Kyung-han Kim

Data Science Lab

January, 2023

Contents

- Seminar Orientation
- Overview of **Statistical Computing** and **Bayesian Statistics** course
- R basics
- R Markdown basics

Orientation - Weekly Schedule

- Week 1: Course Overview, R basics, RMD basics
- Week 2: Markov Chain, Monte Carlo
- Week 3: Markov Chain Monte Carlo (MCMC)
- Week 4: Bayesian Statistics Introduction
- Week 5: Bayesian Regression
- Week 6: Statistical Computing - Random Number Generation
- Week 7: Statistical Computing - Gibbs Sampler
- Week 8: Statistical Computing - Final Exam (2022-2)
- We are going to use R as a programming language.
  Also, some assignments will be re-used in our seminar.
- Schedule may be changed during the seminar.

Course Overview - Statistical Computing

- Professor: Seonghyun Jeong (2021-2, 2022-2)
- Assignments (60%/100%):
  - 8 assignments in total, 20 points each.
  - No deduction until -5 points. (5 make-up points)
- Final Exam (30%/100%):
  - 3.5 hours open-book exam. (Live coding exam!)
  - Some questions require mathematical proof (ex. Find the mgf of ...), and the others require live coding!
  - Lecture note, previous assignments, Internet searching: All available!
  - Communication among students is the only prohibited way of solving.
- Prerequisite: **Mathematical Statistics (1) (STA3126)**
- You need to be very familiar with **variable transformation** and **pdf, cdf, mgf**.
- You cannot use any built-in packages in R.
  Sometimes, you can only use runif($\cdot$) for random number generation.

Course Overview - Bayesian Statistics

- Professor: Jaewoo Park (2022-2)
- Assignments (60%/100%):
  - 6 assignments in total, 100 points each.
  - Be careful with minor deductions.
- Final Exam (30%/100%):
  - 3 hours open-book exam. (Live coding exam!)
  - Some questions require mathematical proof (ex. Find the conditional distribution of ...), and the others require live coding!
  - Lecture note, previous assignments, Internet searching: All available!
  - Communication among students is the only prohibited way of solving.
- Prerequisite: Mathematical Statistics (1), Mathematical Statistics (2)
- You can freely use every built-in package in R.
  Some of questions are designated for using specific package.

Why R?

- R is the only programming language used in Statistical Computing and Bayesian Statistics course.
- In every assignment and final exams, we must use R and write a report with its result.
- However, R programming is not included in the syllabus.
  We need to do it by ourselves.
- We need to know...
  - Vector and matrix
  - How to make custom function
  - for, while loop and if-else condition
  - Distribution-based functions
    (i.e., r, d, p, q + norm/gamma/binom/pois/$\cdots$)
  - How to draw plot or histograms
  - How to import library/packages in R (Bayesian Statistics only)
- Be familiar with Rstudio!

R Basics (1): Vector, Matrix

- 1) Vector
  - 1-dimensional data form
  - Similar to list in Python
  - No need to take care of data type (numeric, character, ...)
  - Use 'c()' function ('rep()', 'seq()': special case)
  - Size can be checked with 'length()' function.
- 2) Matrix
  - 2-dimensional data form
  - Similar to 2-dimensional array in Python
  - Use 'matrix()' function
  - Vector should be included inside the 'matrix()' function as a data part.
  - *nrow* or *ncol* option should be mentioned inside the function if needed.
  - *byrow* (T/F): Determines the direction of the data inside the matrix.
  - Size can be checked with 'dim()' function.
- Let's use rep(NA, _)!

R Basics (2): Custom Function

- We can make our own function with 'function(){}' function.
- Inside the parenthesis, we can designate necessary inputs.
  If not needed, we can leave it empty.
- Calculation or formula should be defined in the curly bracket, "{}".
- At the last part of the curly bracket,
  we use 'return()' to designate the final result of the function.
  Note that 'return()' part is not necessary.
  However, in most cases, we are going to need it.

R Basics (3): Distribution-based functions

- R is a statistician-specialized programming language,
  so it has various types of probability distribution-based functions in it!
- In Statistical Computing and Bayesian Statistics course, we need:
  - Uniform distribution (unif)
    - Until Homework 4, this is the only random number generating function you can use in Statistical Computing.
  - Normal distribution (norm)
  - Binomial/Bernoulli distribution (binom)
  - Gamma distribution (gamma)
  - Inverse-gamma distribution (invgamma): package needed
    - Note that there are two types of 'invgamma' function!!
      One is in 'invgamma' package, and the other is in 'nimble' package.
      Both two functions work properly, but have different name with rate/scale parameter. If you import both package in one R file, 'invgamma' function will follow the most recently imported one.
- We need to specify the proper parameter(s) inside the parenthesis.
  We can check proper parameter with 'help()' function or '?'.

R Basics (3): Distribution-based functions (cont'd)

- These functions cannot be used by itself, but we need to combine with one of *r,d,p,q* in front of them.
- Each letters work as follows:
  - *r*: Generates a random number from a specific distribution.
  - *d*: Gives a pdf/pmf value.
  - *p*: Gives a cdf value of a specific distribution.
  - *q*: Gives a quantile of a specific distribution.
    In fact, quantile function is an inverse of cdf. ($p^{-1} = q$)
- For instance,
  rnorm(1, mean=0, sd=1) = ??? (one random number from Z)
  dnorm(1, 0, 1) = f(1) = $\frac{e^{-0.5}}{\sqrt{2\pi}}$ where $f(x)$ is a pdf of Z.
  pnorm(1, 0, 1) = P(Z $\leq$ 1)
  qnorm(1, 0, 1) = Inf

R Basics (4): Plot and Histogram(1)

- We use 'plot()' function in order to draw a scatterplot or curves.
- 'x' and 'y' must be mentioned in the parenthesis as a vector.
- If x = c(x1, x2, x3, x4) and y = c(y1, y2, y3, y4),
  then **plot(x, y)** gives us a scatterplot of four points,
  $(x1, y1), (x2, y2), (x3, y3), (x4, y4)$.
- We can draw a curve or line with a simple trick -
  define 'x' with very small interval!
  - If x = seq(-10, 10, **0.001**), and y = dnorm(x, 0, 1),
    then **plot(x, y)** gives us a pdf of Z in interval [-10, 10].
- If we want to draw/overlap another curve at the original plot,
  we use 'lines()' function. This can be applied in histogram too.
- Straight lines can be drawn with 'abline()' function.
  - abline(h=0): horizontal line with value 0,
    abline(v=0): vertical line with value 0.
- There are many 'aesthetic' options which can change our plot fancier!
  (Check R file)

R Basics (4): Plot and Histogram(2)

- We use 'hist()' function in order to draw a histogram.
- Unlike 'plot()', we don't need x values. This means we only need one vector which we want to visualize as a histogram.
- Most of 'aesthetic' options can also be applied in 'hist()' function.
- Note that some of questions in Statistical Computing gives a full credit only when you use 'nclass' option.
  Unless, you can get a minor deduction as 'coarse histogram'.
  - Be careful when you need to overlap a density curve with a histogram! Also, you need to use 'freq=FALSE' option when you want to overlap a density curve with a histogram. (This direction will be given in a homework.)
- With 'nclass' option, we can designate the number of 'bar' of a histogram.
- par(mfrow=c(a,b)) determines the alignment of multiple plots. Multiple plots will be aligned in *a* rows and *b* columns.

R Basics (5): Install and Import Packages in R

- Sometimes, we need to use additional packages or libraries.
- If such packages are not installed, we need to install it first.
- 'install.packages()' function helps us to install additional R packages.
- Once the package is installed, we need to import it using 'library()' function.
- You only need to install package once. Later, just import it directly!
- Be careful - Statistical Computing does not allow any kind of additional packages!!
- On the other hand, some of the questions in Bayesian Statistics homework cannot be solved without additional packages...

Why R Markdown (RMD)?

- Both Statistical Computing and Bayesian Statistics course has a lot of assignments. $(8 + 6 = 14)$
- Moreover, all of them require us to write a report.
- A report should include R code and its result with your interpretation.
- Without RMD, we need to capture all codes and their results manually. This time-wasting work is quite annoying.
- RMD helps us to write our report conveniently!
- All codes work as the exact same way with normal R codes. We can mix R code, plots, and personal writings easily with RMD.
- Also we can add mathematical equations such as

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-\frac{(x-\mu)^2}{2\sigma^2})$$

easily!

RMD (1): Code Chunk

- RMD is a markdown document.
  So, in default, your writing is not regarded as a code.
  They are just Korean or English.
- If we want to add an R code, we need to define a code chunk.
- You can make a code chunk with Ctrl + Alt + I command.
  Or, manually type three ‘s and r at the beginning and
  type three ‘s at the end.
- Do not erase the first code chunk which includes
  knitr::opts_chunk$set(echo = TRUE)!
  RMD does not work without this line.
- The green triangle button on the right side is ‘run current chunk’ button.

RMD (2): Mathematical Equations

- We need to know some rules and commands to write a mathematical equation in RMD.
  - All equations should be inside of $ signs.
  - If you want to write an equation independently, write an equation inside \[and \].
  - Greek letters/some famous things: put \ in front of their names.
    (Ex) σ: \sigma, exp: \exp, $\sqrt{2}$: \sqrt{2})
  - Subscript works with _, and superscript works with ^.
  - Other useful math symbols can be found in
    https://www.math.uci.edu/ xiangwen/pdf/LaTeX-Math-Symbols.pdf
  - We can use \frac{}{} to write a fraction.
    First bracket is a numerator, and second bracket is a denominator.
    (Ex) $\frac{1}{2}$: \frac{1}{2})
  - Write $\sim$ with \sim command!
    RMD doesn't recognize it if we simply type it from keyboard...

RMD (3): Knit

- When you finished your writing, press Knit button.
  Then you can get an HTML document. (or Word, LaTeX document)
- Submit it on LearnUs!