Peer Review 1

Natalia Zuniga-Garcia

October 1, 2017

1 Description

I went trough the documentation available in the student's GitHub page by October 1, 2017.

- Student evaluated: Yanxin Li (`https://github.com/Cindy-UTSDS/BigData`)
- Assignment reviewed: Exercises 1
- Document description:
  1. "exercises01-SDS385.pdf"
  2. "exercise 01 solutions.pdf"
  3. "exercise01 Generalized Linear Models.R"
  4. "exercise01 Linear Regression.R"

2 Code Style Evaluation

In this section I provide comments regarding the main code style characteristics: notation and naming, syntax, organization, and function definition. I based my evaluation criteria mainly on the documentation provided by the professor"Advanced R" by Hadley Wickham [1], and "Google's R Style Guide"[2].

2.1 Notation and Naming

- **File names**: I would suggest that the files' name do not include a space. You can use an hyphen (-) or a underscore (_) to separate the name of the files instead of a space.
- **Object names**: I like the way you named the objects and the functions. It makes it easier to follow the code. For instance, you called your inversion function "inv_method" and son on ("exercise01 Linear Regression.R", lines 10, 20). I think, in general, you demonstrated a good object naming.

2.2 Syntax

- **Spacing**: In general, the code presents adequate spacing. There are a few exceptions. For instance, sometimes when you use the simple multiplication operation (*), you don't provide an space in-between ("exercise01 Linear Regression.R" lines 22, 23, 41, 42). It is recommended to place spaces around all infix operators. Also, add spaces within the inputs' colons when writing a function ("exercise01 Linear Regression.R" lines 10, 20). You used it in others functions ("exercise01 Generalized Linear Models.R" lines 29, 42), but maybe try to keep it constant.

[1] `http://adv-r.had.co.nz/Style.html`

[2] `https://google.github.io/styleguide/Rguide.xml`

- **Curly braces**: You used appropriately the curly braces. I would only suggest to leave only one line as you did in “exercise01 Generalized Linear Models.R”, because in “exercise01 Linear Regression.R” (lines 15, 34, 53) you leaved two lines.

- **Line length**: There are some cases where you have long lanes, mainly when you comment the operations (“exercise01 Linear Regression.R” lines 22, 100). I would suggest to make the comment in the previous or posterior lane to avoid this. Also, if you have a long code line (“exercise01 Generalized Linear Models.R” lines 8, 122) you can break the line.

- **Indentations**: The indentation of your code seems appropriate to me.

- **Assignment**: You correctly used $< -$ to the assignments.

2.3 Organization

Trough the code you provided adequate comments that guided me easily. You used #——————- to separate your code into chunks, I like it because it makes it easy to read part by part (I will use it on my code next time!!). I just have an observation regarding commenting functions’ definition, but I will discuss it on the next section.

2.4 Function definition

I would suggest to comment more when defining a function. The professor made this comment when I presented my code to the class. He suggested me to define each input and output of the function more in detail. (I think) He said a function is like a contract you make and you have to be very specific in case someone else wants to use it. You did it appropriately on your gradient function (“exercise01 Generalized Linear Models.R” lines 37 to 47), maybe try to do it inside the function and keep it for the other functions you defined.

3 Code Efficiency Evaluation

- You multiply your matrices using %*%, I found out that *crossprod* function is more efficient. Check on R help: _Given matrices x and y as arguments, return a matrix cross-product. This is formally equivalent to (but usually slightly faster than) the call t(x) %*% y (crossprod) or x %*% t(y) (tcrossprod).)._

- In ‘exercise01 Generalized Linear Models.R” line 8, when you import a *.CSV* document, if you have it in the same carpet, you can just use *read.csv(”wdbc.csv”, header = FALSE)* instead specify the location into your computer. This makes it easier if someone else want to run your code outside your computer and use the same database.

- We both used Matrix library to exploit sparsity.

4 General Comments

- In “exercise 01 solutions.pdf” page 1, I like the way you explained all the other factorization methods.

- In “exercise 01 solutions.pdf” Figure 1 and 2, I would suggest next time show the results in a plot or with a summary table more specific on the result we are interested in (the mean or median time). Think like you are showing the results to attract readers and you want to make it easy read and understand. Figures are always good, it helps me to understand better too. You did this in Part

2 of the exercise, and it is very easy to see the difference between Gradient Descend and Newton Method.

- In “exercise 01 solutions.pdf” page 8, I like your code attachment. It seems like you are using R-Markdown. I would like you to include this file next time. I just started to used it based on Mauricio’s recommendation, I bet there is a lot I can learn from you.
- I would also encourage you to include any other document you used. For instance, a Latex documents if you used it, the R-Markdown files, or even papers if you used it. This can help others to understand better and to learn.

5 Summary

Dear Yanxin,

I tried to provide you good feedback. I tried check in detail your documentation so you can improve. In the process I learned a lot. I learned from your code and by reviewing it. For instance, I found out a lot of mistakes in my own code. Some of the comments I made to you are also in my code (I wish I knew about this before). I hope this help you in your next assignments, it did help me!