

Stats Modeling 2

Peer Review For Zhenpei Yang

Jennifer Starling
Feb 17, 2017

You say this is your first time using R, but your code is really very good and clean! I did have a few tips for you that will make things flow better or run a bit faster.

1. Using `crossprod()`:

Instead of using `solve(t(x)%*%y)` for example, you can use `crossprod(x,x)`. This is equivalent but runs faster.

2. Function documentation:

When you write a function, you added a quick comment to describe what the function does in a sentence. It is also helpful to list the function inputs and outputs, so that anybody else who uses your function will understand the context and format of the inputs/outputs.

3. Housekeeping:

At the beginning of each R script, it is helpful (though of course not required) to add the line of code below. This removes all objects currently in the environment, so you start off with a clean slate. I find this helpful because sometimes if you were already working with R on a different script, you might accidentally use variables in the workspace from your previous work. This ensures you don't get any crazy unintended results.

```
rm(list=ls())
```

4. Code commenting:

I would add a few more descriptive code comments to help you remember what things are later. For example, in your exercise-02-1.R file, you have commented to 'define some variables' for the hyperparameters. It would be helpful to have a note that this section sets up the hyperparameters, and perhaps a comment reminding yourself of what each hyperparameter does. This helps me when I have a lot of different scripts for different models – you can quickly say “Oh yes, K is the prior precision matrix” when you look back at your code. Also, this helps you remember which parameterization you have used (ie precision versus covariance) in models where multiple choices are available.

5. Using “Source”:

This isn't necessary in Ex 1 and 2, since there aren't as many functions, but it is a thing I just started doing in Exercise 3, and you might also find it helpful.

You can save a separate R script with your functions, and then in the main script where you are using the functions, you do not need to write them out. Just use

```
source('filepath/filename.R')
```

And your functions will be loaded to the new script. This will help your code read cleanly, and will let you pull in the same set of functions to multiple scripts easily.