MTH208: Worksheet 10

Streamlining your codes

Now is a good time to learn how to streamline codes that take time, and to ensure reproducibility of our work. Recall in Worksheet 8, our scraping code can take a large amount of time. We will learn how to work with such codes.

Further, this is also how I run your codes, so you will learn to ensure you do not make mistakes in your submissions.

Operating Systems

There are three main operating systems for machines

- Windows - least preferred OS for programming
- MacOS - better for programming than Windows, but it's a closed community
- Linux (different versions) - most flexible for programming

All our machines in the lab are linux based machines. The particular operating system is CentOS7. Default linux machines come with a non-GUI interface. That is, there is no mouse and click! All your machines are equipped with a GUI interface, where it feels like a usual Windows-type computer.

However, for reproducibility and to use the full landscape, it is often advisable to use the **terminal.**

Terminal

A terminal is essentially a "Command Line Interface", where you give computers commands for different tasks. These commands rely on "code" or types text, and not on the mouse.

1. Go to your desktop, right-click, and choose "Open Terminal". This opens the terminal window.
2. Go to the website here https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners and look at the various commands. Now try the following commands for yourself:
    - `pwd`
    - `ls`
    - `cd`
    - `mkdir`

- `rmdir`
- `rm`
- `touch`

3. We can also open `R` in the terminal. Type `R` and press enter. An `R` session will open in the terminal itself. Next, type `q()` to quit `R`.

4. Now in the terminal, type `R --vanilla`. This tells R to open an R session without loading any information and also without saving anything at the end. Now try `q()` to exit `R` and notice the difference.

R with the Terminal

You can see that opening `R` with the Terminal only opens the console, and not the script or other features.

Any `*.R` file can be opened with any text editor (the default on your machines is I think Gedit). And then that editor works like the `RStudio` script editor. You can also use the script editor in `Rstudio` as your editor as well!

Using the Terminal for `R` console is not as useful, we might as well use `Rstudio`. Instead we can use the Terminal to run `R` scripts in the *background*.

1. In the Terminal, using commands, go to your working directory for Worksheet 2, where you have saved `worksheet2.R`.

2. Once inside that directory, run the following
   `R CMD BATCH --vanilla worksheet2.R &`

   This command will run the whole worksheet2.R code in a new clean session.

   - `R` in the beginning says we will be running `R`
   - `CMD` means "command". We are giving an R command.
   - `BATCH` means we are running this `R` script in the background
   - `--vanilla` means run a clean session
   - `&` sign at the end means that the the terminal will remain active for other tasks while the code runs.

3. Once you run the above, whatever is usually printed in the console will be saved in a `worksheet2.Rout` file. Go check this file and see the contents.
   If the file runs successfully (without errors), then a `proc time` is shown at the end telling us how long the code ran for. If there is an error, then the error message is shown.

4. Notice that there is no presentable output from the code! The code ran, but we do not know what is inside any objects, and we cannot check either. Suppose I want to save the `allround data.frame` we

created in Worksheet 2. Add the lines

```
save(allround, file = "cricket_a.Rdata")
```

This saves the *object* `allround` (in the exact format it is currently present in) in an `Rdata` file called `cricket_a.Rdata` in your working directory. Now re-run Part(2).

In this way, you can save multiple objects as well

```
save(allround, msc, file = wksheet2.Rdata")
```

5. Using this information, now rerun Worksheet 6 code in the terminal so that the final IMDb dataset from this code is saved in an `*.Rdata` object. (Replace the * with your choice of name).

6. Go back to all worksheets (except worksheet 8) so far and make sure they are:

   - Organized in their own subdirectories.
   - `R CMD BATCH` commands on them work without any errors. If you find any errors, fix them so that the codes run cleanly.

7. Load an `R` session from your working repository for Worksheet 6 in the Terminal. Type `ls()` in the console. It should return `character(0)`.
   Now type `load("*.Rdata")` (where replace the * with the correct name). This loads the `R` objects. Check by typing `ls()`.

8. Repeat Part (6) for all three of your assignment codes as well. You will then see the exact errors I got when I ran your code.

9. Repeat Part (6) above again for Worksheet 8 (this code takes time, that is why we are leaving it to the end).