STAT 230/530 Lecture 2: R & R Studio Tips and Best Practices

Business

- Make sure you are officially shopping so we have a record of you and so that you receive official course announcements!
- This term, drop-in extra help will be held in 17 Hillhouse Room 115. Days and times to be announced (soon I hope, and perhaps starting immediately).
- HW 1 will be posted over the weekend, due Monday September 12.
- My intended new section organization is in processing. We'll figure it out.

Tips and Best Practices

- Based on years and years of experience working with undergraduate and graduate students, high school students, government and industry professionals, . . .
- Will save you time
- Will help reduce confusion
- Will help avoid frustration
- Will make it more likely you're doing what you want to do
- Will save us time

Change some R Studio default options (required)

- Open R Studio
- `Tools > Global Options...`
- Look for the first drop-down: `Save the workspace to .RData on exit`
- Change this first drop-down to `Never`
- All of the check-boxes ***above*** this dropdown should be ***unchecked***. That is, no, don't restore anything at startup.

Tip 1: Organize your data analysis world with folders

- Create a STAT230 folder someplace. Maybe on your desktop?
- Use subfolders in a sensible way, and use a lot of them!
- Example: Each homework assignment should have its own folder.
- Philosophy: one folder, one "data set" (even if a single data set might involve multiple files), and one or more scripts that are a complete record of your work.

Tip 2: Do all of your work in a script

- Avoid typing in the console unless it is really something interactive that you don't need a record of.
- Save your work regularly (that is, save the script)
- Only work on one thing at a time (more on this later)

Tip 3: Only work on one thing at a time!

- Constantly close up R Studio (don't just minimize)
- Whenever possible, start R Studio by double-clicking a script for your analysis – by doing this, R Studio will automatically be working in the right place (if you organize yourself as recommended above)
- Don't have multiple scripts open simultaneously relating to different data analysis problems

Example 1: BB2012 (after data scrape done on Wednesday)

We'll do this interactively in class.

Tip 5: Learn a few shortcuts

- With your cursor on some line of a script, do -enter on the Mac or -enter on the PC. Some people say “return” instead of “enter”, or “ctrl” instead of “control” or (on the Mac “squiggley” instead of “command”).
  - This runs that line of the script, equivalent to a copy-paste from the script into the console
  - If it doesn’t work or doesn’t do what you expect, then perhaps you didn’t run the preceeding code in your script? The script is a sequential record of your work. Actual work is done in the console.
- If you highlight a block of code, the same shortcut will run the block

Tip 6: Other customization (optional)

- Again in `Tools > Global Options...`
- Under the "Appearance" section, choose a font size large enough so you can read your code easily
- Under "Code" there are a bunch of options that are a matter of personal preference.
- Example: "Insert matching parens/quotes" is selected by default, but I turn it off (I don't like it). Some students love this. Your choice.

Tip 7: How much code should go on a line and is indentation important?

- Not more than 80 characters
- Under Code > Display there is an option to show a vertical line at column 80, and I strongly recommend sticking to this.
- A single command can be broken onto multiple lines.
- Indendatation is important and easy in R Studio. Don't be sloppy!
- -I or -I can automatically do some nice indendatation of your code... another useful shortcut!

Example 2: Environmental Performance Index 2012

We'll do this interactively in class. The year 2012 shared between the basketball and this environmental performance index (EPI) work is purely coincidental.

Other notes

- Some of these tips (and more) appear in the `ExtraIntroMaterial.pdf`. That's required reading for shopping period.
- This course focuses on the core (base, standard) R language for much of the term. Use of additional packages (i.e. `ggplot2`, etc...) require formal approval. Ask before using.
- An exception to the point above: yes, you can let R Studio install packages that might include `rmarkdown` or `markdown` or `knitr`... to support production of nice documents. More on this later.
- The "answer" is less interesting than the "process". Assigned problems may have "answers" in the traditional sense, but the learning objects are broad. Don't lose sight of the forest through the trees.

Time permitting

```
page <- 'http://www.realclearpolitics.com/epolls/2016/pres
x <- scan(page, what="", sep="\n")
temp <- x[grep("Polling Data", x)[2]]
```