ST 516: Foundations of Data Analytics

General coding tips for larger simulations

Code is a form of communication

Just like written prose, there is easy to read code and hard to read code.

Your goal is to make your code easy to read for yourself, for your future self, and for anyone else that may read it.

Part of having easy to read code is using a consistent style. You might like to look into a style guide and start following it: `http://adv-r.had.co.nz/Style.html`

Naming variables

- Use short but descriptive names for variables
- Don't use the same name for different things in your code
- Avoid using names that belong to R functions
- Different people follow different conventions for multi-word names: `long_name`, `long.name`, `longName`. Pick one and **be consistent**.

Long lines and indentation

It's easiest to read code if it isn't one really long line.

Don't be afraid to start a new line, if the command is continuing add an indent.

Bad

```
long_line <- paste("This is a really long line", "that I thought I woul
```

Good

```
long_line <- paste("This is a really long line",
  "that I thought I would cram onto",
  "one line to make it hard", "to read")
```

Don't be afraid to use multiple files

Once your analysis prokects get larger, sometimes it's easier to put parts in different files.

If the parts need to be run in order, use numbers in the file names: 1-clean-data.R, 2-perform-analysis.R

Save relevant objects and plots

When you start to run larger simulations, you don't want to rerun long computations just to reexamine the results.

You can save R objects using the `saveRDS` function and read them back in later with `readRDS`

```
x <- rnorm(10000)
saveRDS(x, file = "x-sim.rds")
# then in a new session
x <- loadRDS("x-sim.rds")
```

Also, save plots using code in your scripts, that way if you need to adjust them, it's easy to find where you made it in your code:

```
x <- rnorm(1000)
qplot(x, binwidth = 0.2)
ggsave("x-hist.pdf", width = 4, height = 2)
```

Comments

Use comments:

- to help navigate the structure of your code:
  `# === Question 1 ==================`
- to help explain the why not the what:
  `# Unusual observation in histogram, examine in more detail`