Introduction to R for Data Management and Analysis

Marcel Ramos, MPH

Tuesday, June 18, 2019

Announcements

- Exercise 4
- Next week (conference)

Notes on Thursday's lecture

- Interactivity
- Plotting Odds Ratios
  - Base graphics
  - ggplot2

Review

- Notes on RMarkdown format
  - `install.packages` - installation needed once only, install via the console
- Settings in RStudio
  - Default to plot inline
  - Plot in plotting window (more space)
- Plotting systems in R
  - base, lattice, ggplot2
- Exploratory Data Analysis
- Graphics device
  - the default window / pane for all your plots
  - save to external files (`pdf`, `png`, etc.)

Code reading practice

- See `inClass_S5.R` file
  - First create a linear model from variables
  - Use `broom::tidy` on linear model object
  - Add a couple of columns using the `mutate` function
  - Remove teams that are missing using `is.na` and restrict to 2015
  - Reorder teams by some metric `estimate` and use `geom_pointrange`
  - Flip coordinates and add some labels

Today's lecture topics

- Repetitive code
- `for` loops
- Functions
- Functionals and functional programming
    - `apply` family

Examples

- Generate data using `replicate` and `sample` functions
- Replacing missing values with `NA`
- When to use a `for` loop
- When to write a `function`

`for` loops

- Repeat code a certain number of times
- Usually reserved for simple operations
- `for` <*variable*> in <*sequence of numbers*> { operation }
- Each step is visible
- Purpose may not be immediately clear

Functions

- Extend the language
  - Portable
- Group operations for *ideally* one purpose
- *Pure* functions
- Well defined inputs and output (usually)

Functionals

- An argument that itself is a function
- Many functions can accept other functions as an arguments
  - lapply
- Make coding more efficient and customizeable
- Increased flexibility but add a layer of complexity
- Why use them?
  - To avoid loops and simplify code