Introduction to R for Data Management and Analysis

Marcel Ramos, MPH

Session 5

Announcements

- Review Exercise 4 in class

Topics to review

- install.packages vs library
    - install.packages - run in the console once (if successful)
        - downloads the package from the internet and installs it
        - installation does not make the functions immediately available!
    - library - makes functions available in a package by loading it
- Notes for using RMarkdown
    - install.packages - installation needed once only, install via the console
    - do not include install.packages in the RMarkdown file
    - do not include help functions in the code, that's FYI only
    - load a package using library("pkgname") before using its functions
    - loading makes the functions available for use

Topics to review (2)

- Make use of the example code
  - library(dplyr); mtcars %>% ...
- Settings in RStudio
  - Default to plot inline
  - Plot in plotting window (more space)
  - Change the settings in Tools > Global Options > RMarkdown
- Avoid attach()!
  - Creates a mess of your global environment

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
- Don't worry about the specifics
- Understand the gist of what is happening
- Use what we know to look up functions: ?, help, Google

Today's lecture topics

- Plotting Systems
    - Types of graphs
    - ggplot2
    - Saving outputs
- Repetitive code
    - for loops
    - Functions
    - Functionals and functional programming
        - apply family

Plotting systems in R

- Three main systems:
  - ‘Base’ graphics
    - Already installed with R
  - lattice
    - Less common
  - ggplot2
    - Most popular
    - Syntax is different

Exploratory Data Analysis

- Informal representation data
- Looking for patterns, outliers, etc.
- Getting familiar with your data!

Types of graphs

- Historgram
- Scatterplot
    - Scatterplot matrix
- Boxplots / dotplots (ggplot2)
- Violin plots (ggplot2)
- Q-Q plots
- Mosaic plots
- and many more!

ggplot2 - Grammar of Graphics

- Different syntax
    - Slight learning curve
- Plots are built in layers
- Operations add layers to the plot

Saving outputs

- Common formats for saving plots:
  - PDF
  - SVG
  - PNG/TIFF
- but there are more
- ggsave

Output sandwhich

- Start with a function pdf, png, jpeg, etc.

- End in dev.off() for closing the graphics window

Saving plots in ggplot2

- ggplot2 graphics require a print (or a call) before it gets rendered in the file.
- ggsave - added to make it easier to save plotting objects

Recommended resources

- Fundamentals of Data Visualization
    - Claus O. Wilke
- R Graphics Cookbook
    - Winston Chang

Repetitive operations

- What strategies work to reduce the amount of repetitive coding?
  - for loops
  - function
- Instances where you might repeat code
  - Replacing missing values with NA
  - Data cleaning operations
- When to use a for loop / lapply
  - When you want to apply a series of operations more than once
- When to write a function
  - When you run a series of operations more than once

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 of the loop may not be immediately clear

Functions

- Main mechanism for extending the language
    - Packages make these portable
- Group operations for ideally one purpose
- Pure functions - input is the same class as the output
- Well defined inputs and outputs (usually)

Functionals

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