Topic 2: Basics for Coding in R

1. Initial Guidelines for R code
2. Objects
3. Examples of Assigning Objects
4. Functions
5. Getting Help with R

1. Elements of R Coding

- R is case-sensitive
- Number of spaces generally not important for syntax
  - But spacing can make code easier to read
- Wrapping lines of code ok
  - Limiting lines to 80 characters
- # is Comment character for line (or for after line of R code)
  - Get in habit of commenting code, for others and yourself
  - Narrative in R Markdown
    - Before/after chunk of code
    - No need for comment character, but can still use # within code chunk

Elements of R Coding, running R Code

- Can code in R script or Create RMarkdown file
  - Code in script with no need for any mark-up to code
  - RMarkdown, must code inside ```` ```{r} ```` and closing ticks ```` ``` ````

- To Run code in R Script
  - Highlight code to run, click Run button
  - Or to run one line of code,
    - just put cursor ANYWHERE on line, click Run
    - Notice that this advance to next line.... Can keep clicking to run sequential lines
  - Instead of clicking Run button
    - **Windows: Push Control + Enter keys**
    - **Mac: Push Command + Return keys**

Elements of R Coding, running code continued

- Running R Markdown Code
  - Text outside the 3 ‘open and closing ticks’ will not run
  - Only run code in grey area: a chunk
  - As before, to run an entire chunk, use play button
  - Otherwise, can run code inside chunk as with regular R script
    - See previous slide
    - The Run button provides more options
- Revisit methods for running R code (will remind through out Bootcamp)
  - Save time by running certain parts of code
  - Learn efficient key strokes
  - Use **copying and pasting effectively** to rerun certain parts of altered code
  - Comment out parts that you don’t want to run using #

Elements of R Coding continued

- Errors
  - Will learn to interpret some types of errors
  - Good red and bad red (warnings are good and often allow code to run anyway)
- Keep track of open and closed parentheses, brackets, braces
- Keep track of commas
  - Errors caused by parens and commas can be hard to interpret
- The two basic elements of R code: objects & functions
  - Objects store information
  - Functions do something

2. Objects

- Types of objects
  - Vectors (R doesn't not really have scalars, vectors length = 1)
  - Matrices (not really covered here)
  - Data.frames (tables of data, we will refer to this object a lot)
  - Define function results as object (more on this later)
- Stored Objects appear in Global Environment window
- $ operator
  - References element of an object
  - Syntax: `object_name$element_name`

3. Assigning Objects

- `<-` vs `=`
  - The same assignment operator (almost)
  - '`<-`' might be thought of as defining an object to a name
  - '`=`' might be thought of as the mathematical equality
- Assign a vector
  - `Y <- c(2,3,4)`
  - `y = c(5, 6, 7)`
- Assigning data.frame
  - `Data <- cars`
  - Will load data in R as part of Topic 4.

4. Functions

- Obvious functions
  - `mean()`
  - `round()`
  - `median()`
  - `anova()`
- Not Obvious
  - `sd()` standard deviation
  - `lm()` linear model (later)
- Function can depend on type of object
  - Same function name can perform different tasks depending on object
  - e.g. `plot()`, example later
- Function with same name in different packages (more on this later)

Functions... continued

- Functions require argument(s)
- Often functions need an object in parentheses
  - `mean(data$y)`
- Can specifying detail of a function (often multiple specifications)
  - Separated arguments by commas
  - `round(data$y, 2)`
- Default arguments if not specified
  - `round(data$y,2)`
- Functions within functions
  - Parenthetical caution – easy to loose track of open and closed parens
  - `round(mean(data$y),2)`

5. Getting Help with R

- Functions
  - Can hit tab key to see arguments
  - Note suggested arguments when typing
- Help in Rstudio
  - Help in Menu
  - Help Tab in lower right window R documentation (when package loaded)
  - Includes examples
  - Examples often use built-in R datasets (copy, paste, and run)
- `help()` or `?function_name`
- Google
  - This is a skill
  - You will become increasingly better at finding answers/examples online
- Ask colleague