Topic 4: Basics of Importing Data into R

1. Data files for use in R
2. Ways to Import Data
3. Basic Functions to Check Data

1. Data files in R

- Built-in data files
  - Base R (and R packages) include example data frames
  - Use data() to view built-in datasets
    - Use '`data(package = .packages(all.available = TRUE))`' to list the data sets in all *available* packages
  - Easiest way to load data (use with examples in R help)
- Importing datasets
  - Comma separated values (.csv files) are common
  - Can import from MS Excel (as .xlsx, but more easily as .csv)
  - Multiple functions and methods to load data
  - Important to examine data once imported!

2. Ways of Importing Data

- Point and click via Global Environment window
  - Can click Import Dataset button 
  - Not recommended
    - Some datatype errors may occur
    - Must reload and browse for data each time running new session
    - Can have difficulty maintaining data frame naming throughout code
- Running code to load data
  - `read.table()`
  - `read.csv()` this is the one we will use!
- Other methods (beyond scope of coursework)
  - API's -- Application Programming Interface
  - Read in from webpage – (some instructors may use this to provide datasets)
  - etc...

Importing Data via `read.csv()`

- `read.table()` more versatile, but works similarly as `read.csv()`
- Can supply pathname in R Code (preferred)
- Or can browse for Dataset using `file.choose()`
- Pathname of dataset
  - Can type path name into code (preferred)
  - Must not move file
  - May need different paths if using different computers to analyze dataset
- Browsing for dataset
  - Must search for data file each time wanting to load
  - Not ideal for using with R Markdown

Importing Data via `read.csv()`

- Pathname syntax: `dataframe_name <- read.csv("pathname")`
- Getting the pathname with Mac!
  - Use find  to find dataset
  - bottom of finder window shows path
  - Right click on the datafile, then push and HOLD option key
  - 2/3 down is: Copy "datafilename.csv" as Pathname
  - Then paste pathname into R code inside quotes of `read.csv("")`
  - May also use '~' to shorten path name
    - Change:
      `"/Users/bensharp/Dropbox/STAT511/Assignments/Assign6/RatLiver.csv"`
    - to:
      `"~Dropbox/STAT511/Assignments/Assign6/RatLiver.csv"`

Importing Data via `read.csv()`

- Getting the pathname with Windows
  - Use Windows Explorer 
  - path name can be viewed and copied
  - Paste pathname into R code inside of quotes of `read.csv("")`
  - BUT need forward slashes in R !!!
  - Change all backslashes to forward slashes
  - Example code:

```
RatLiver<-read.csv("C:/Users/sharp/Dropbox/STAT511/Assignments/Assign6/RatLiver.csv")
```

  - OR can right click on datafilename.csv
    - copy/pasting into Rstudio editor already converts forward slashes
    - Must delete file:/// that appears at the beginning of this pathname
  - May be a way to shorten pathname as with Mac?

Importing Data via `read.csv()`

- `file.choose()`
  - Syntax: `dataframename <- read.csv(file.choose())`
  - Run code, RStudio gives window to browse and select datafilename.csv
  - Run `file.choose()` without `read.csv`, click data file to see file pathname in console

- Other considerations for long pathnames
  - Can create a folder for datafiles near root directory (of cloud storage)
  - No path necessary if working directory is in same folder of datafiles
    - Possible to open R script files from Windows Explorer for default working directory to be set in same folder (this may not work for everyone)
    - Try: `getwd()` to see pathname of R session working directory
    - Try: `setwd()` to establish pathname to folder where working with datafiles
    - Syntax without pathname just becomes: `read.csv("datafilename.csv")`

3. Checking Data Loaded into R

- "running" data frame is simplest way to see your data
  - Create data frame object using `read.csv()`
  - "Send" data frame object to Console, i.e. just run the data frame
  - Will show limited rows of what was loaded as data frame
- `View(dataframename)`
  - Yes, capital 'V'
  - Creates a spreadsheet view of data, but in new tab among R Code tabs
  - May copy/paste more easily
  - Requires toggling between R code and the View of data
  - Not ideal for R Markdown

Checking Data Loaded into R continued

- `head(dataframename)`
  - Shows the first few rows of the data frame
  - Can specify number of rows to be shown

- `tail(dataframename)`
  - Shows the LAST few rows of the data frame
  - Can specify number of rows to be shown
  - Depending on sorting, can show if last few rows were properly loaded

Checking Data Loaded into R continued

- `str(dataframename)`
  - Super helpful to see information about the data
  - Shows list of columns of data
  - Data types are shown -- important to know what R believes the data types are!!
  - First few elements of each column shown

- Other self-explanatory functions for data frames
  - `length()`
  - `nrow(), ncol()`
  - `dimensions()`