**R Library**

**R/S/S+**

* Object oriented, everything is an object other than operators
* Just type an object name to view data (no print function needed)
* No ; at the end of statements
* Use c() to concatenate

**Operators**

* Assigning variables
  + Use = or an arrow (<- or ->)
  + Point at the variable name when using an arrow
* Mathematical operators like other languages
* Comparison operators similar to other languages (<, >, ==, !=, <=, >=, &[is single & sign] )
  + Returns TRUE or FALSE
  + Can use to select data
    - Dataframe[column.name] > value & dataframe[2, ] == value

**Built-in Functions**

* class() tells you the class (character, number, logical[Boolean]) of an object
* as.character() convert to a string
* sqrt()
* log()
* sum()
* length() returns the length of the vector (array)
* rownames() provides a list of row names for a dataframe (sometimes they’re indexed numbers)
* attach() sets the data = dataframe.name
* detach() releases data = dataframe.name
* plot()
  + Can supply two variables to create one plot area (dependent~independent)
  + Can provide just dataframe to create a matrix of plots
  + Arguments
    - main = ‘Title’
    - xlab, ylab = ‘string’ for axis labels
    - xlim, ylim = [,] vector of length 2 specifying max/min for each axis
    - abline() to include a trendline
* points()
  + Works like a scatter plot
* summary()
* print()
  + Useful to store graphs/figures in a variable, then print(variable.name)
* unique(dataframe$column) will return a list of all unique values
* data.frame(vect1[, vect2]…)
* na.rm = TRUE (use as argument with other functions within () after specifying dataframe)
  + Removes NA (null) values
* is.na() returns TRUE if the argument has a value of NA and FALSE otherwise
  + If a vector/dataframe is supplied, it will provide a vector/dataframe of TRUE/FALSE values for each element
* help() can supply functions inside
* help.start()

**Boolean**

* Use a vector to select elements
  + Store an array of T, F values (selection.array)
  + Reference this as an index of a vector (vector)
    - vector[selection.array] would only return values where the index matched up with a T value (omits the indexes that correspond to F in the selection.array)

**Strings**

* Double or single quotes (works like other languages with quotes within strings)
* \ is the escape character

**If/else**

* Syntax similar to other languages
* if (conditional statement) {

code to execute

} else {

code to execute

}

**Dataframes**

* A matrix (essentially a table) stored in a variable (columns are vectors or field names and rows are data entries)
* Access one column (dataframe.name$column.name)
* Dataframes and vectors are 1 indexed (not 0 indexed like other languages)
* Access an element using dataframe.name[1, 2] (accesses row 1 column 2 data entry)
  + Can also be used to edit/assign data
  + dataframe.name[1, ] accesses all columns for row 1 (or [, 2] for all rows of column 2)
* Creating dataframes
  + Import csv files
  + Construct row by row or column by column
  + edit() to create new or edit(dataframe.name) to edit data
  + **Connect with a database**
  + Interface with other software

**Libraries and Associated Functions**

* Call a library
  + library(library.name)
* lattice
  + xyplot()
  + plot1 <- xyplot(dependent.variable ~ independent.variable, group = variable.name, data = dataframe.name

print(plot1)

* + - * Could also use (dep ~ ind | variable.name) syntax
    - xlab, ylab (like plot())
    - panel = function (…) { }
      * Can create custom functions between the { and }
      * panel.xyplot(…)
      * panel.lmline(…)
  + cloud() adds an extra variable/dimension
    - cloud(dep ~ ind1 + ind2, group = var1, data = dataframe)