R Workshop – GW Gelman Library – January 31, 2020

1. Function: typeof(x) = tells you the type of the variable
2. If typeof = “logical” = explains TRUE or FALSE values
3. Function: c -> to compose a vector (comma separated list of values)
   1. Degrees <- c(‘BA’, ‘BS’)
4. Vector: sequence of data elements of the same basic type. Members of a vector are called Components
5. Data visualizations
   1. height <- c(72, 81, 71, 60, 67, 60, 65, 68, 67, 69, 70, 71, 73, 60, 68, 64, 59)
   2. hist(height) => will give you a histogram of the heights
   3. boxplot(height) => gives u the median in visualization form
6. Dataframe -> basically a table with rows and columns
7. Python pandas is similar to read function in R
   1. You have to install and load the package
8. Dataset projects:
   1. Read.csv(‘file\_name’) set it to a vector: for example gap\_df <- read.csv(‘gap.csv’)
   2. Summary(gap\_df) = quick summary and statistics of the data
   3. Install.packages(‘dplyr’) – package useful for wrangling data
      1. dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges:
         1. mutate() adds new variables that are functions of existing variables
         2. select() picks variables based on their names.
         3. filter() picks cases based on their values.
         4. summarise() reduces multiple values down to a single summary.
         5. arrange() changes the ordering of the rows.
      2. Gap\_df %>% select(\*column\_names without quotes\*)
         1. %>% is a pipe in dplyr.
      3. Function: filter() – selects or filters the rows of the data table that meet certain criteria creating a new data subset.
      4. Function: mutate() – used to create a new variable from a dataset
      5. Function: arrange() – used to reorder rows by one or more variables (default is ascending)
         1. Descending – add a negative sign (-) or DESC
   4. Install.packages(‘ggplot2’) – data visualization package for the statistical programming language R
      1. Setting up: ggplot(data = gap\_df)
   5. Function: aes() – aesthetic mapping – mapping data with visual features
      1. ggplot(data = gap\_df) + aes(x = lifeExp, y = gdpPercap) + geom\_point()  
         - plots the axis (without points)  
         - plots some scatterpoints
      2. function: geom\_point() – plots points (there are different ‘geoms’ for ggplot)
      3. function: scale\_x\_log10() – scales to tens
   6. gap.lm -> lm stands for linear model
      1. y ~ x -> way to express a relationship between y and x
      2. gap\_df2007$loggdp <- log10(gap\_df2007$gdpPercap) – adds a new column called loggdp that is the log 10 of gdppercap