Research Project Workshop 07

Exploratory Data Analysis (EDA) & Data Visualization

Exercises

1. Creating and running a script of basic functions (slide 42)
   1. Create a new R Project and name it RPW\_07
   2. Save a new R script file and name it “exercises1”
   3. Add a few functions to your script file, use the # to annotate your work
      1. > print(“data viz”)
      2. > 3\*7/(5-4)
      3. Create an object (using *assignedname* <- ) named test\_calc that computes the square root (sqrt) of 100
   4. Install a package, use the function > install.package(“*nameofpackage”)*
      1. gdata, ggplot2; gcookbook; knitr; swirl
      2. for series installation use the function > install.package(c(“gdata”, “ggplot2”))
   5. Load a package, use the function: > library(“*nameofpackage*”)
   6. Save and select Code (menu at top); Run Region; Run All to test run

\*TIP: notice how comments are really useful in your script for documentation

\*TIP: also notice, your script recreates everything you need for what you’ve done!

1. Exploring workspace - do this exploring in the console area (slide 43)
   1. To see where RStudio is “looking” for files ask what is the working directory, use the function > getwd( )
   2. To change the working directory, use the function > setwd(“*folder/filename path*”)
   3. To see a list of files in current workspace, use the function > list.files( )
   4. Adjust working directory if needed

\*TIP: you should already be working in created projects space – best practice to create project

1. Importing data - save in script!
   1. Install and load the “gdata” package to work with Excel spreadsheets
   2. Use the provided smoking data spreadsheet and assign it as smoking\_df (this type of object is called a data frame). Use the function > read.xls(“*filename*”)
   3. Alternatively, import data from the web using this url: (<http://jwhollister.com/public/files/example.csv> and assign it as jhweb\_df (this type of object is called a data frame).
   4. Explore: Use the function > str(*nameofdata*) to understand what the structure of your data looks like
   5. Explore: Use the function > summary(*nameofdata*) to see descriptive stats summary.
   6. EXTRAS: Use the function > names(*nameofdata*) to see variable names
   7. EXTRAS: Use the function > is.na (*nameofdata*) to identify missing values

\*TIP: there’s pre-installed data you can locate with the data( ) function

1. EDA – save in script!
   1. Find the mean of a variable, use the function:

> mean(*nameofdata$variablename*)

* 1. Find the standard deviation of a variable, use the function:

> sd(nameofdata$*nameofvariable*)

* 1. Find the quartiles of a variable, use the function:

> quantile(*nameofdata$nameofvariable*)

* 1. Perform a Welch’s T-test, use the function:

> t.test(*nameofvariable*)

* 1. Perform a two-sample t-test, use the function:

> t.test(smoking\_df$tot\_cig ~ smoking\_df$sex, var.equal=TRUE, conf.level=0.80)

* 1. Perform a correlation of two variables, use the function:

> cor(smoking\_df$sys, smoking\_df$tot\_cig)

* 1. Perform test of correlation, use the function:

> cor.test(smoking\_df$sys, smoking\_df$tot\_cig)

\*TIP: think of tilde to mean “as a function of” or can variable2 (with two levels) explain the change in variable1?

1. Visualizations – select one or more and save in script! The [cookbook](http://hbanaszak.mjr.uw.edu.pl/TempTxt/Mittal_2011_R%20Graphs%20Cookbook.pdf) will help customize.
   1. Create a scatterplot (page 9), use the function:

> plot(smoking\_df$bmi, smoking\_df$tot\_cig)

* 1. Bar chart (page 14), use the function:

> barplot(smoking\_df$tot\_cig, names.arg = smoking\_df$sex, col = "black")

* 1. Bar chart of means, use a series of functions:

> mean.sex <-aggregate(formula = tot\_cig ~ sex, data = smoking\_df, FUN = mean)

> barplot(mean.sex$tot\_cig, names.arg = mean.sex$sex, col = "black")

* 1. Histogram of variable in data (page 18), use the function:

> hist(smoking\_df$bmi)

* 1. Box plot (page 20), use the function:

> boxplot(smoking\_df$ss\_base, smoking\_df$ss\_wk2, las=1)

> boxplot(smoking\_df$tot\_cig ~ smoking\_df$sex, main = "Boxplot of baseline smoking by Group", ylab="cigs smoked (total)", las=1)

1. Saving visualizations
   1. Can save png (or substitute pdf) all using the function:

> png("*figurename.png*")+

barplot(mean.sex$tot\_cig, names.arg = mean.sex$sex, col = "black")+

dev.off( )