# **Assignment related questions**

***Each week, make sure to add in the interpretation of the result - this can be a few sentences about if the model seemed accurate, whether the hypothesis was rejected, what is the difference between methods, or how results are interpreted - - beyond stating what the numeric values are; make sure to indicate why the approach you are taking makes sense.***

***(Here is an example for Assignment 1 – Hypothesis testing and Exploratory data analysis; each assignment may be focusing on a different data science method – such as Classification or Decision trees)***

Null Hypothesis (H0): A lab value level is equivalent to the national average of 150

μ = 150

Alternative Hypothesis (Ha): A lab value level is greater than the national average of 150 (One sided)

μ > 150

Here a t-test is being used, hypothesizing if the sample of a lab level to comparing if it is equivalent or greater than the national population.

The test found that the data sample was equivalent to the national average lab values, failing to reject the null hypothesis at a 5% significance level.

The t-test reports that the P-value is 0.117. That is, the test statistic does not fall in the "critical region." Since the P-value, 0.117, alpha is greater than = 0.05, the researcher fails to reject the null hypothesis.

# **R code:** Downloaded .R File from Rstudio

***Here are the Format of the .R file and R codes – other examples for your reference, can include the R codes that are provided as practice problems; this can be a long text file also – as it is a direct download of the code itself)***

*### Some useful keyboard shortcuts in R Studio:*

*### CTRL+ENTER Executes on the command line the script line containing the cursor*

*### CTRL+1 Moves cursor to script window*

*### CTRL+2 Moves cursor to command line*

*### Up Arrow [In console] Cycles through previously entered commands*

*### TAB Activates autocomplete - very useful for paths and variable names*

*### F1 Searches help for the word containing the cursor*

*##########################*

*# Basic Math in R*

*# R understands the four aritmetic functions +, -, \*, /*

*# as well as exponents via ^*

*3+2*

*4 - 6 #*

*4^3*

*# R is flexible about numeric values - don't need to worry about integer vs. float*

*2/3*

*2.0/3.0*

*#########################*

*# Storing values in variables*

*a<-2*

*b<-3*

*a/b*

*##########################*

*# Vector Math in R*

*# We can use the c(...) command to create vectors from a list of values.*

*# We reference entries by their location, given in square brackets.*

*prices <- c(5, 10, 12, 13)*

*quantities<-c(100, 3, 2000, 40)*

*prices\*quantities*

*sum(prices\*quantities)*

*taxrates.v<-c(1.07, 1.09, 1, 1.15)*

*sum(prices\*quantities)\*taxrates.v # compare cost of this basket of items in four regions*

*sum(prices\*quantities\*taxrates.v) # compute cost of buying 1st item in 1st region, etc.*

*##########################*

*# Gluing together vectors*

*cbind(prices, quantities)*

*rbind(prices, quantities)*

*##########################*

*# Example from slides:*

*### Data Creation*

*# create the tiny data frame:*

*r1 <- c(1, "Kirkwood", "MO", 63122)*

*r2 <- c(2, "St. Louis", "Missouri", 63108)*

*r3 <- c(3, "Kirkwood", "MO", 63212)*

*r4 <- c(4, "U City", "MO", 63130)*

*r5 <- c(5, "St. Louis", 63110, "MO")*

*r6 <- c(6, "Saint Louis", "MO", 63110)*

*# Glue all of those rows together with rbind()*

*tiny <- rbind(r1, r2, r3, r4, r5, r6)*

*# what does it look like?*

*tiny*

*# What does R think of this object?*

*class(tiny)*

*##########################*

*# How do we access entries?*

*# Similar to Excel, reference by row # and column #.*

*# Row 1, Column 2*

*tiny[1,2]*

*# To get all rows, leave row # blank, follow with comma and column of interest*

*# 3rd column*

*tiny[,3]*

*# To get all columns, enter rows of interest, follow with comma and close bracket*

*# 2nd row*

*tiny[2,]*

*### Notice all answers are being written horizontally as lists.*

*### In the Environment tab to the right, we see all the entries are stored as text.*

*### Let's change this from a matrix to a data frame.*

*tiny <- as.data.frame(tiny)*

*class(tiny)*

*tiny*

*# Give the columns names:*

*names(tiny)<-c("CaseNo", "City", "State", "Zip")*

*# Check the structure:*

*str(tiny)*

*# All values are showing up as "Factors" (Categorical variables).*

*# Let's try again:*

*tiny <- rbind(r1, r2, r3, r4, r5, r6)*

*tiny <- as.data.frame(tiny, stringsAsFactors=F)*

*names(tiny)<-c("CaseNo", "City", "State", "Zip")*

*str(tiny)*

*# Uh oh! There is a state in the zip column!*

*tiny$Zip*

*# What does the unexpected entry look like?*

*tiny[which(tiny$Zip=="MO"),]*

*# Let's fix it by referring to the row and column address.*

*# This data is in the 5th row, 3rd and 4th columns.*

*# (Note the colon lets you select a range of columns or rows):*

*tiny[5,3:4]*

*tiny[5,c(4,3)]*

*tiny[5,3:4]<-tiny[5,c(4,3)]*

*# And check our work:*

*tiny*

*# Let's order it by city:*

*tiny[order(tiny$City),]*

*# What cities are in our data?*

*unique(tiny$City)*

*# How many from each city:*

*table(tiny$City)*

*# Let's standardize names:*

*tiny[c(2,5),]*

*tiny[c(2,5),]$City*

*tiny[c(2,5),]$City<-"Saint Louis"*

*tiny*

*tiny[2,]$State*

*tiny[2,]$State <- "MO"*

*tiny*

*# Let's look at Kirkwood only:*

*tiny[which(tiny$City=="Kirkwood"),]*

*# All Kirkwood zips should be 63122.*

*tiny[which(tiny$City=="Kirkwood"),]$Zip*

*tiny[which(tiny$City=="Kirkwood"),]$Zip<-63122*

*tiny*

*###########################*

*# Saving data to CSV*

*### To check the default working directory, use "getwd()" without quotes.*

*getwd()*

*### To change the working directory, type the path in quotes inside*

*### the setwd() command.*

*### PC Example: setwd("C:/Users/YOURUSERNAMEHERE/Desktop/")*

*### Mac Example: setwd("/Users/YOURUSERNAMEHERE/Desktop/")*

*setwd() # Note: you must enter a valid folder path inside parentheses for this to work.*

*write.csv(tiny, "tiny.csv")*

*# Note: you can specify the full file path inside write.csv or read.csv*

*# write.csv(tiny, "D:/temp/myFirstFile.csv")*

*# Open your CSV in Excel, make some changes, save and close it*

*###########################*

*# Bringing data in from a file*

*not.so.tiny<-read.csv("tiny.csv")*

*not.so.tiny*

*# We don't need the "X" column. Check that removing it gives the ones we want.*

*not.so.tiny[,2:5]*

*# Redefine the data set*

*### WARNING - NO UNDO function in R.*

*### You can "start over" by reading in the csv and redefining not.so.tiny.*

*not.so.tiny <- not.so.tiny[,2:5]*

*not.so.tiny*

*############################*

*# Using additional packages*

*# install.packages("data.table")*

*# install.packages("ggplot2", dependencies = TRUE)*

*# library("data.table")*

*# library("ggplot2")*