Homework 5

James Young

6-28-2019

# Instructions

There are six exercises below. You are required to provide solutions for at least four of the five. You are required to solve at least one exercise in R, and at least one in SAS. You are required to provide five solutions, each solution will be worth 10 points. Thus, you may choose to provide both R and SAS solutions for a single exercise, or you may solve all five problems, mixing the languages as you wish.

*Warning* Starting with these exercises, I will be restricting the use of external libraries in R, particularly tidyverse libraries. Our goal here is to understand the R language and the mechanics of the R system. Much of the tidyverse is a distinct language, implemented in R. You will be allowed to use whatever libraries tickle your fancy in the midterm and final projects.

#### Experimental

Again, you will be allowed to provide one solution using Python. Elaborate on the similarities and differences between Ptyhon function definitions and R or IML or Macro language.

# Exercise 1. ##Please Grade

This exercise will repeat Exercise 1 from Homework 4, but using a data table.

### Part a.

Create a data table or frame with 3 defined columns: - Let M1 be a sequence of means from 320-420, incremented by 10. - Let M2 be 270. - Let SD be a pooled standard deviation of 150.

Define and print the tabke in the space below. Do not create individual vectors for this exercise, outside of the data frame, if you use R. In SAS, you may use IML to create a matrix and save the matrix as a data table, or define a sequence (DO) in the DATA step. I’ve included framework code in the SAS template.

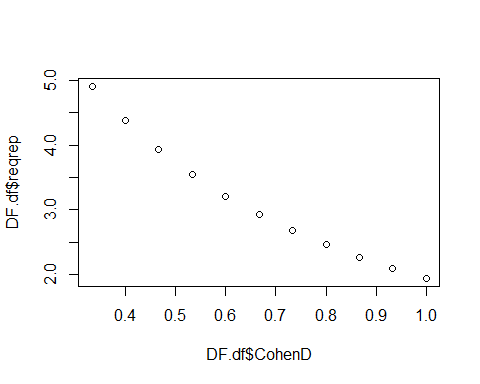
M1 = seq(320,420,by=10)  
M2 = c(270, 270, 270, 270, 270, 270, 270, 270, 270, 270, 270)  
SD = c(150, 150, 150, 150, 150, 150, 150, 150, 150, 150, 150)  
  
#create data frame using the previously defined vectors  
DF.df = data.frame(M1, M2, SD)  
DF.df

## M1 M2 SD  
## 1 320 270 150  
## 2 330 270 150  
## 3 340 270 150  
## 4 350 270 150  
## 5 360 270 150  
## 6 370 270 150  
## 7 380 270 150  
## 8 390 270 150  
## 9 400 270 150  
## 10 410 270 150  
## 11 420 270 150

Add to the data table a column containing required replicates, letting . Also add a column containing Cohen’s .

To show results, either print the table or plot required replicates versus as in the previous homework.

#cohen's d function from previous homework  
#SD pooled is 150  
cohen.d = function(M1, M2, SD){  
 d = (abs(M1 - M2)) / SD  
 return(d)  
}  
  
required.replicates <- function (m1, s1, m2, s2) {  
 CV <- 150/((m1+m2)/2)  
 Diff <- (m1-m2)/((m1+m2)/2)  
 2 \* ((CV/Diff)\*\*2) \* (((1-(0.05/2))+(1-0.2)))\*\*2  
}  
  
DF.df$CohenD = cohen.d(DF.df$M1, DF.df$M2, DF.df$SD)  
DF.df$reqrep = required.replicates(DF.df$M1, DF.df$M2, DF.df$SD)  
plot(DF.df$reqrep~DF.df$CohenD)



# Exercise 2 ##Please Grade

### Part a.

You will repeat the calculations from Homework 4, Ex 2, but this time, using a data table. However, instead of a matrix, the result with be a table with 30 rows, each corresponding to a unique combination of CV from and Diff from .

The table should look something like

Test your required replicates calculations by calculating required replicates for each combination of CV and Diff using the default values for and . Name this column Moderate.

A = seq(8, 28,by=4)  
B = seq(5, 25,by=5)  
  
  
d1 <- expand.grid(CV = B, Diff = A, KEEP.OUT.ATTRS = FALSE)

required.replicates <- function (CV, Diff) {  
 2 \* ((CV/Diff)\*\*2) \* (((1-(0.05/2))+(1-0.2)))\*\*2  
}  
  
d1$Moderate = required.replicates(d1$CV, d1$Diff)  
d1

## CV Diff Moderate  
## 1 5 8 2.4614258  
## 2 10 8 9.8457031  
## 3 15 8 22.1528320  
## 4 20 8 39.3828125  
## 5 25 8 61.5356445  
## 6 5 12 1.0939670  
## 7 10 12 4.3758681  
## 8 15 12 9.8457031  
## 9 20 12 17.5034722  
## 10 25 12 27.3491753  
## 11 5 16 0.6153564  
## 12 10 16 2.4614258  
## 13 15 16 5.5382080  
## 14 20 16 9.8457031  
## 15 25 16 15.3839111  
## 16 5 20 0.3938281  
## 17 10 20 1.5753125  
## 18 15 20 3.5444531  
## 19 20 20 6.3012500  
## 20 25 20 9.8457031  
## 21 5 24 0.2734918  
## 22 10 24 1.0939670  
## 23 15 24 2.4614258  
## 24 20 24 4.3758681  
## 25 25 24 6.8372938  
## 26 5 28 0.2009327  
## 27 10 28 0.8037309  
## 28 15 28 1.8083945  
## 29 20 28 3.2149235  
## 30 25 28 5.0233179

Calculate required replicaes again, but this time let and let . Label this column Conservative. Repeat the calculations, but this time let and let .

If you choose SAS, you can use the framework code from the first exercise.

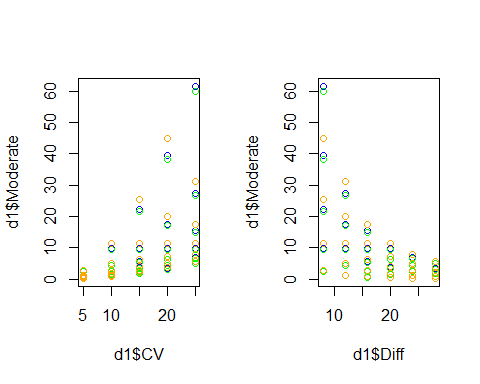
required.replicates <- function (CV, Diff) {  
 2 \* ((CV/Diff)\*\*2) \* (((1-(0.01/2))+(1-0.1)))\*\*2  
}  
  
d1$Conservative = required.replicates(d1$CV, d1$Diff)  
  
required.replicates <- function (CV, Diff) {  
 2 \* ((CV/Diff)\*\*2) \* (((1-(0.1/2))+(1-0.2)))\*\*2  
}  
  
d1$Liberal = required.replicates(d1$CV, d1$Diff)  
d1

## CV Diff Moderate Conservative Liberal  
## 1 5 8 2.4614258 2.8054883 2.3925781  
## 2 10 8 9.8457031 11.2219531 9.5703125  
## 3 15 8 22.1528320 25.2493945 21.5332031  
## 4 20 8 39.3828125 44.8878125 38.2812500  
## 5 25 8 61.5356445 70.1372070 59.8144531  
## 6 5 12 1.0939670 1.2468837 1.0633681  
## 7 10 12 4.3758681 4.9875347 4.2534722  
## 8 15 12 9.8457031 11.2219531 9.5703125  
## 9 20 12 17.5034722 19.9501389 17.0138889  
## 10 25 12 27.3491753 31.1720920 26.5842014  
## 11 5 16 0.6153564 0.7013721 0.5981445  
## 12 10 16 2.4614258 2.8054883 2.3925781  
## 13 15 16 5.5382080 6.3123486 5.3833008  
## 14 20 16 9.8457031 11.2219531 9.5703125  
## 15 25 16 15.3839111 17.5343018 14.9536133  
## 16 5 20 0.3938281 0.4488781 0.3828125  
## 17 10 20 1.5753125 1.7955125 1.5312500  
## 18 15 20 3.5444531 4.0399031 3.4453125  
## 19 20 20 6.3012500 7.1820500 6.1250000  
## 20 25 20 9.8457031 11.2219531 9.5703125  
## 21 5 24 0.2734918 0.3117209 0.2658420  
## 22 10 24 1.0939670 1.2468837 1.0633681  
## 23 15 24 2.4614258 2.8054883 2.3925781  
## 24 20 24 4.3758681 4.9875347 4.2534722  
## 25 25 24 6.8372938 7.7930230 6.6460503  
## 26 5 28 0.2009327 0.2290195 0.1953125  
## 27 10 28 0.8037309 0.9160778 0.7812500  
## 28 15 28 1.8083945 2.0611751 1.7578125  
## 29 20 28 3.2149235 3.6643112 3.1250000  
## 30 25 28 5.0233179 5.7254863 4.8828125

To show your work, some ideas for graphs:

* Plot required replicates vs CV and Diff, using different colors or symbols for Moderate,Conservative and Liberal

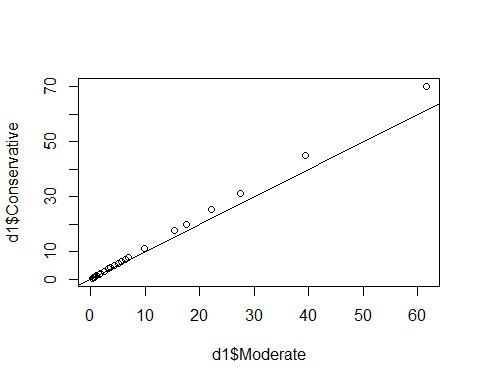
par(mfrow=c(1,2))  
plot(d1$Moderate~d1$CV, type ="p", col = "blue")  
points(d1$Liberal~d1$CV, type = "p", col = "green")  
points(d1$Conservative~d1$CV, type = "p", col = "orange")  
  
plot(d1$Moderate~d1$Diff, type ="p", col = "blue")  
points(d1$Liberal~d1$Diff, type = "p", col = "green")  
points(d1$Conservative~d1$Diff, type = "p", col = "orange")



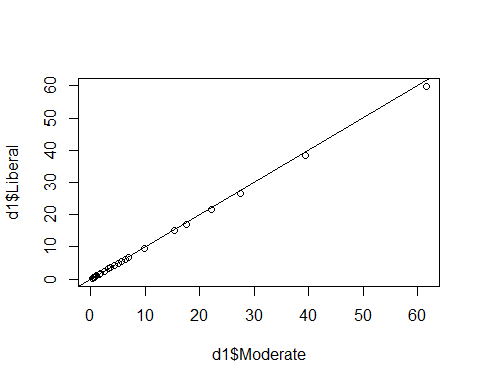
#Although moderate is the vertical axis label, all required replicate estimates are plotted according to the color coding demonstrated above.

* Plot Conservative vs Moderate and Liberal vs Moderate, including a line with slope 1 and intercept 0.

plot(d1$Conservative~d1$Moderate)  
abline(0, 1)



plot(d1$Liberal~d1$Moderate)  
abline(0, 1)



# Exercise 3 ##Please Grade

You’ll work with data from U.S. Wholesale price for pumpkins 2018 (<https://www.ers.usda.gov/newsroom/trending-topics/pumpkins-background-statistics/>, Table 1)

## Part a

Download the file pumpkins.csv from D2L and read the file into a data frame. Print a summary of the table.

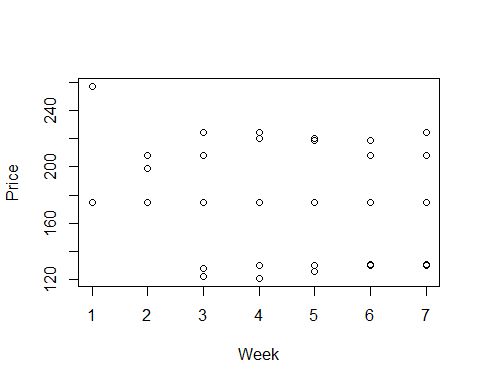
setwd("C://data/")  
pumpkins = read.csv("pumpkins.csv")

## Part b

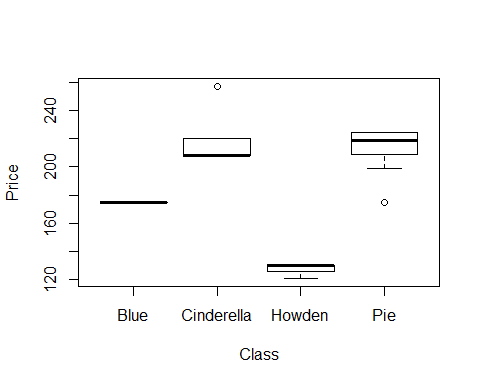
To show that the data was read correctly, create three plots. Plot 1. Price vs Week 2. Price vs Class 3. Size vs Class

These three plots should reproduce the three types of plots shown in the RegressionEtcPlots video, **Categorical vs Categorical**, **Continuous vs Continuous** and **Continuous vs Categorical**. Add these as titles to your plots, as appropriate.

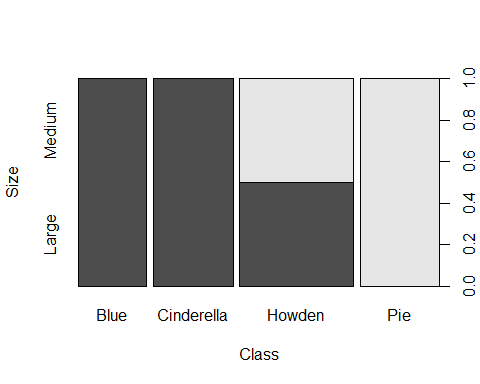
plot(Price~Week, pumpkins)



plot(Price~Class, pumpkins)



plot(Size~Class, pumpkins)



From these plots, you should be able to answer these questions: 1. Are some Weeks missing Price observations? No, all weeks have price observations. 2. Do Prices vary more for some Classes? Yes, for example Blue is only one price while other class options such as pie are variable. 2. Do all Classes have the same Sizes? No, we can see from the graph that there is variation in pumpkin size by class.

# Exercise 5 ##Please Grade

## Part a

Go to <http://www.itl.nist.gov/div898/strd/anova/SiRstv.html> and use the data listed under Data File in Table Format (<https://www.itl.nist.gov/div898/strd/anova/SiRstvt.dat>)

## Part b

Edit this into a file that can be read into R or SAS, or find an appropriate function that can read the file as-is. You will need to upload the edited file to D2L along with your Rmd/SAS files. Provide a brief comment on changes you make, or assumptions about the file needed for you file to be read into R/SAS. Read the data into a data frame or data table.

setwd("C://data/")  
NIST = read.csv("NIST.csv")

## Part c

There are 5 columns in these data. Calculate mean and sd and sample size for each column in this data, using column summary functions. Print the results below

sumitup <- function(x){  
 c(mean(x), sd(x), length(x))  
}  
Col2 = sumitup(NIST$Col2)  
Col1 = sumitup(NIST$ï..Col1)  
Col3 = sumitup(NIST$Col3)  
Col4 = sumitup(NIST$Col4)  
Col5 = sumitup(NIST$Col5)  
  
df = data.frame(Col1, Col2, Col3, Col4, Col5)  
df

## Col1 Col2 Col3 Col4 Col5  
## 1 196.24308000 196.244300 196.16702000 196.1481400 196.14324000  
## 2 0.08747329 0.137975 0.09372413 0.1042267 0.08844797  
## 3 5.00000000 5.000000 5.00000000 5.0000000 5.00000000

Determine the largest and smallest means, and their corresponding standard deviations, and calculate an effect size and required replicates to experimentally detect this effect.

If you defined functions in the previous exercises, you should be able to call them here.

#Largest mean is col2 with 196.244300 with sd 0.137975  
#Smalles mean is col5 with 196.14324000 with sd 0.08844797  
cohen.d = function(M1, S1, M2, S2){  
 SD = (((S1\*\*2+S2\*\*2)/2)^(1/2))  
 d = (abs(M1 - M2)) / SD  
 return(d)  
}  
  
required.replicates <- function (m1, s1, m2, s2) {  
 CV <- (((s1\*\*2+s2\*\*2)/2)^(1/2))/((m1+m2)/2)  
 Diff <- (m1-m2)/((m1+m2)/2)  
 sdpooled <- ((((s1\*\*2 + s2\*\*2))/2)^(1/2))  
 2 \* ((CV/Diff)\*\*2) \* (((1-(0.05/2))+(1-0.2)))\*\*2}  
  
reqrep = required.replicates(196.244300, 0.137975, 196.14324000, 0.08844797)  
cohen =cohen.d(196.244300, 0.137975, 196.14324000, 0.08844797)  
reqrep

## [1] 8.286029

cohen

## [1] 0.8720475

Done