HW5_House_Matthew

Matthew House

10/3/2017

Problem 1

Completed

Problem 2

Completed

Problem 3

What are you thoughts for what makes a good figure?

```
* A good figure is informative and can convey, on its own, what the author is attempting to say. It shou
```

Problem 4

a. Create a function that computes the proportion of successes in a vector. Use good programming practices.

```
# this  was taken from "https://www.r-bloggers.com/r-function-of-the-day-tapply-2/" and edited to work
 my_vect <-
    data.frame(patient = 1:100,
               age = rnorm(100, mean = 60, sd = 12),
               treatment = gl(2, 50,
                 labels = c("Treatment", "Control")),
               success = sample(c(0,1), replace = TRUE, size = 100))

lapply(my_vect[4:4], mean)
```

```
## $success
## [1] 0.52
```

```
#This is a randomly generated value vector so the probability of a success changes each time you run it

#tapply(flags$animate, flags$landmass, mean)
```

b. Create a matrix to simulate 10 flips of a coin with varying degrees of "fairness" as follows:

```
set.seed(12345)
P4b_data <- matrix(rbinom(10, 1, prob = (30:40)/100), nrow = 10, ncol = 10)
```

c. Use your function in conjunction with apply to compute the proportion of success in P4b_data by column and then by row. What do you observe? What is going on?

- It reads it a long vector without row and column designations... unless I didn't do something correctly.

```
#P4b_data
```

d. You are to fix the above matrix by creating a function whose input is a probability and output is a vector whose elements are the outcomes of 10 flips of a coin. Now create a vector of the desired probabilities. Using the appropriate apply family function, create the matrix we really wanted above. Prove this has worked by using the function created in part a to compute and tabulate the appropriate marginal successes.

```
ht <- function(){
  coin_flips <- sample(1:2, size = 10, replace = TRUE)
  return(c(coin_flips))
}
ht()
# Couldn't figure out how to use the apply functions without changing this to have row and column names
Heads_or_Tails <- replicate(10, ht())
table(Heads_or_Tails)
table(Heads_or_Tails)/length(Heads_or_Tails)
```

Problem 5

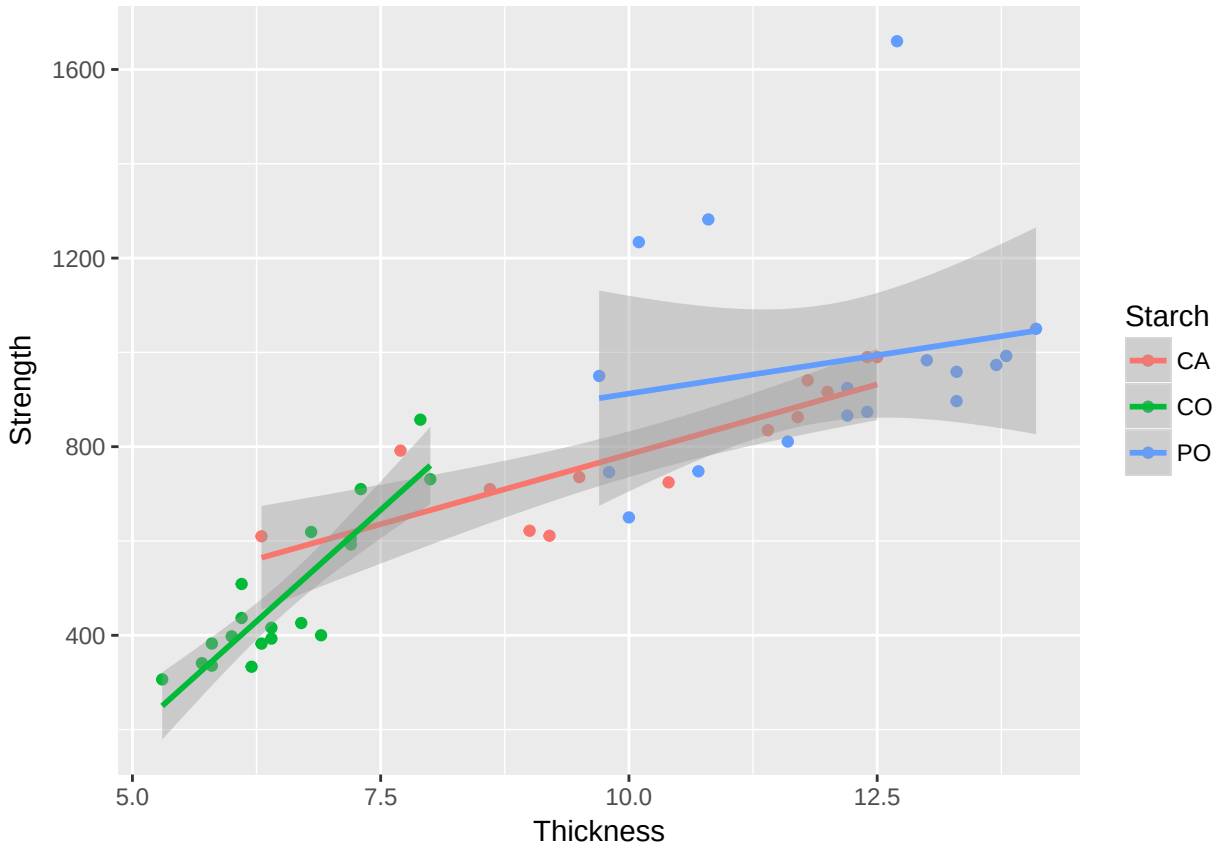
Load, munge, and explore the data given in Wu and Hamada from the starch experiment. Consider strength as the response. You do not need to form a model or otherwise analyze the dataset, you do need to explore the data, make any figures/tables necessary to make observations about the data, and generally annotate the process in text.

http://www2.isye.gatech.edu/~jeffwu/book/data/starch.dat

```
url<- "http://www2.isye.gatech.edu/~jeffwu/book/data/starch.dat"
starch <- read.table(url, header = F, skip = 1, fill = T, stringsAsFactors = F)
TidyStarch <- transmute(.data = starch, Starch = V1, Strength = V2, Thickness = V3)
head(TidyStarch)
```

```
##   Starch Strength Thickness
## 1     CA    791.7       7.7
## 2     CA    610.0       6.3
## 3     CA    710.0       8.6
## 4     CA    940.7      11.8
## 5     CA    990.0      12.4
## 6     CA    916.2      12.0
```

```
grouped <-  group_by(TidyStarch,Starch)
ggplot(grouped, aes(x = Thickness, y = Strength, colour = Starch)) +
  geom_point() +
    geom_smooth(method='lm',formula=y~x)
```




```
#+
#  facet_wrap( ~ starch)
```

Problem 6

Our ultimate goal in this problem is to create an annotated map of the US. I am giving you the code to create said map, you will need to customize it to include the annotations.

Part a. Get and import a database of US cities and states. Here is some R code to help:

```
    #we are grabbing a SQL set from here
    # http://www.farinspace.com/wp-content/uploads/us_cities_and_states.zip

    #download the files, looks like it is a .zip
#    install.packages("downloader")
    library(downloader)
    download("http://www.farinspace.com/wp-content/uploads/us_cities_and_states.zip",dest="us_cities_sta
    unzip("us_cities_states.zip", exdir="./")

    #read in data, looks like sql dump, blah
    library(data.table)
    states <- fread(input = "./us_cities_and_states/states.sql",skip = 23,sep = "'", sep2 = ",", header
    ### YOU do the CITIES
    ### I suggest the cities_extended.sql may have everything you need
    ### can you figure out how to limit this to the 50?
```

Part b. Create a summary table of the number of cities included by state.

```
cities <- fread(input = "./us_cities_and_states/cities_extended.sql", sep = "'", sep2 = ",", header = F

 colnames(states) <- c("State Name", "State Abbv.")
 colnames(cities) <- c("City", "State Abbv.")

 citystate <- inner_join(cities, states, by = "State Abbv.")

 statesum = table(citystate$`State Name`)
 Numcitystate = as.data.frame(statesum)

 colnames(Numcitystate) <- c("State Name", "Freq")

knitr::kable(Numcitystate)

#This was used from Shane's code, as I couldn't get it to count the
#way I was doing it.
```

Part c. Create a function that counts the number of occurances of a letter in a string. The input to the function should be "letter" and "state_name". The output should be a scalar with the count for that letter.

Create a for loop to loop through the state names imported in part a. Inside the for loop, use an apply family function to iterate across a vector of letters and collect the occurance count as a vector.

```
    ##pseudo code
    letter_count <- data.frame(matrix(NA,nrow=50, ncol=26))
    getCount <- function(){
        temp <- strsplit(state_name)
        # how to count??
        return(count)
    }
    for(i in 1:50){
        letter_count[i,] <- xx-apply(args)
    }
```

Part d.

Create 2 maps to finalize this. Map 1 should be colored by count of cities on our list within the state. Map 2 should highlight only those states that have more than 3 occurances of ANY letter in thier name.

Quick and not so dirty map:

```
    #https://cran.r-project.org/web/packages/fiftystater/vignettes/fiftystater.html
    library(ggplot2)
#   install.packages("fiftystater")
#   install.packages("mapproj")
    library(fiftystater)
    library(mapproj)

    data("fifty_states") # this line is optional due to lazy data loading
    crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)
    # map_id creates the aesthetic mapping to the state name column in your data
    p <- ggplot(crimes, aes(map_id = state)) +
      # map points to the fifty_states shape data
      geom_map(aes(fill = Assault), map = fifty_states) +
      expand_limits(x = fifty_states$long, y = fifty_states$lat) +
      coord_map() +
```

```
  scale_x_continuous(breaks = NULL) +
  scale_y_continuous(breaks = NULL) +
  labs(x = "", y = "") +
  theme(legend.position = "bottom",
        panel.background = element_blank())

p
#ggsave(plot = p, file = "HW5_Problem6_Plot_Settlage.pdf")
```

Problem 7

Push your homework and submit a pull request.

When it is time to submit, –ONLY– submit the .Rmd and .pdf solution files. Names should be formatted HW4_lastname_firstname.Rmd