HW5

Jacob Williams

September 20, 2018

Problem 3

A good figure I believe should be a very easy/clear figure to read. The goal of the presenter should be very evident in the figure itself, so when a reader looks at the figure they can understand what the figure is showing immediately. With this as the goal, a informative title, axis titles, coloring by groups, and a legend is all ways to obtain this clear figure outcome.

Problem 4

Problm 4a

If the successes are denoted as a 1 and failures as a 0. A function in base R to compute this proportion is simply the mean().

```
prop <- function(x) {
    if (!is.vector(x) | !is.numeric(x) | sum(!is.na(x)) != length(x)) {
        return("This is either not a vector, not numeric, or has NA's in it")
    } else {
        proportion <- sum(x)/length(x)
        return(proportion)
    }
}
frt <- c(1, 0, NA, 1, 0)
prop(frt)
```

```
## [1] "This is either not a vector, not numeric, or has NA's in it"
```

Problem 4b

```
set.seed(12345)
hut <- (30:40)/100
P4b_data <- matrix(NA, nrow = length(hut), ncol = 10)
for (i in 1:length(hut)) {
    P4b_data[i, ] <- rbinom(10, 1, prob = hut[i])
}
apply(P4b_data, 1, prop)
```

```
##  [1] 0.6 0.2 0.3 0.4 0.3 0.4 0.6 0.3 0.3 0.5 0.6
```

Problem 4c

The row proportion of success should be centered around .35 and the column proportion of success should be centered around the same value of .35. The row proportion of success should have a larger variance due to

the fact that your comparing values with different centers. The probabilities for the columns in theory will increase by .01 for every single column, but this would happen as are sample sizes go to infinity.

Problem 4d

```
set.seed(12345)
library(data.table)
flips <- function(probability) {
    if (!is.vector(probability)) {
        return("Not a vector")
    } else {
        flip_data <- vector()
        flip_data <- rbinom(10, 1, probability)
        return(flip_data)
    }
}
probs <- (30:40)/100
P4d_data <- matrix(unlist(lapply(probs, flips)), nrow = length(probs),
    ncol = 10, byrow = T)
# The step above could easily be down in the functions itself
apply(P4d_data, 1, prop)
```

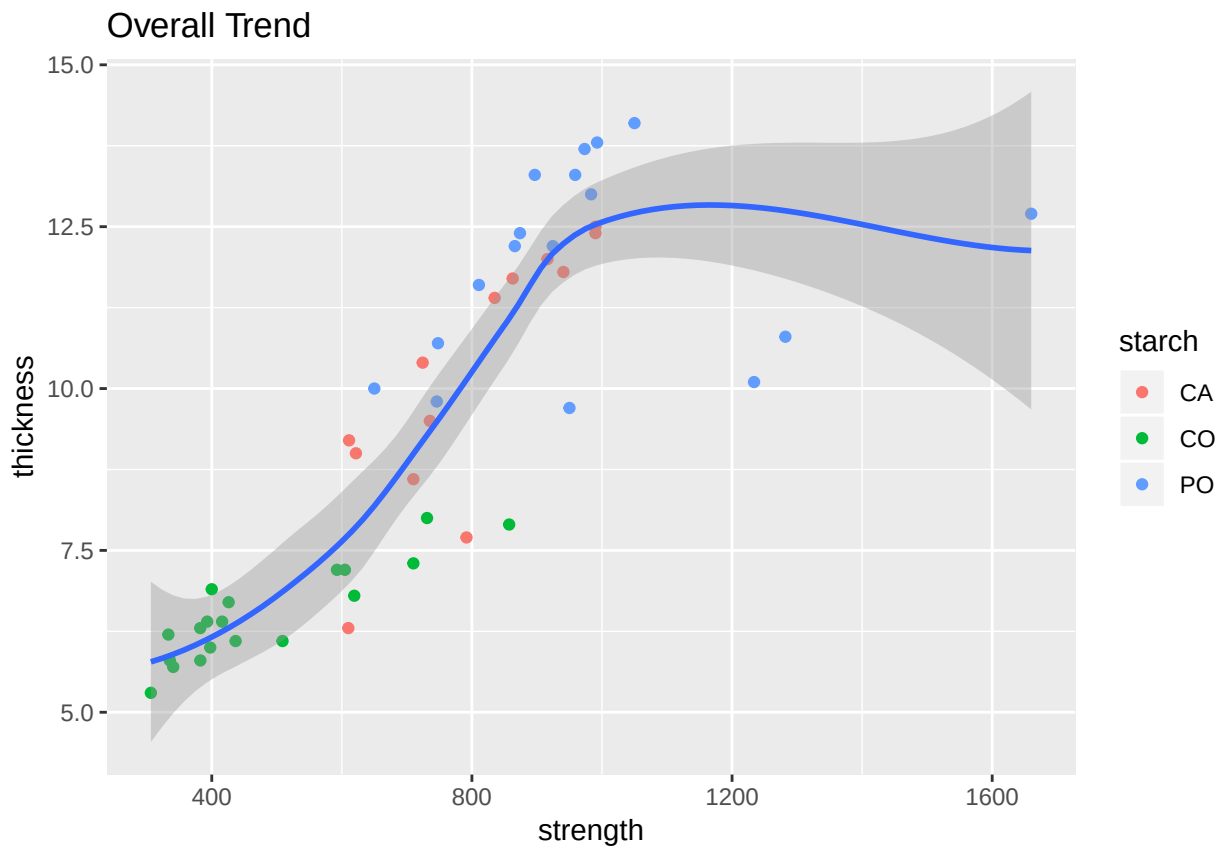
```
##  [1] 0.6 0.2 0.3 0.4 0.3 0.4 0.6 0.3 0.3 0.5 0.6
```

Problem 5

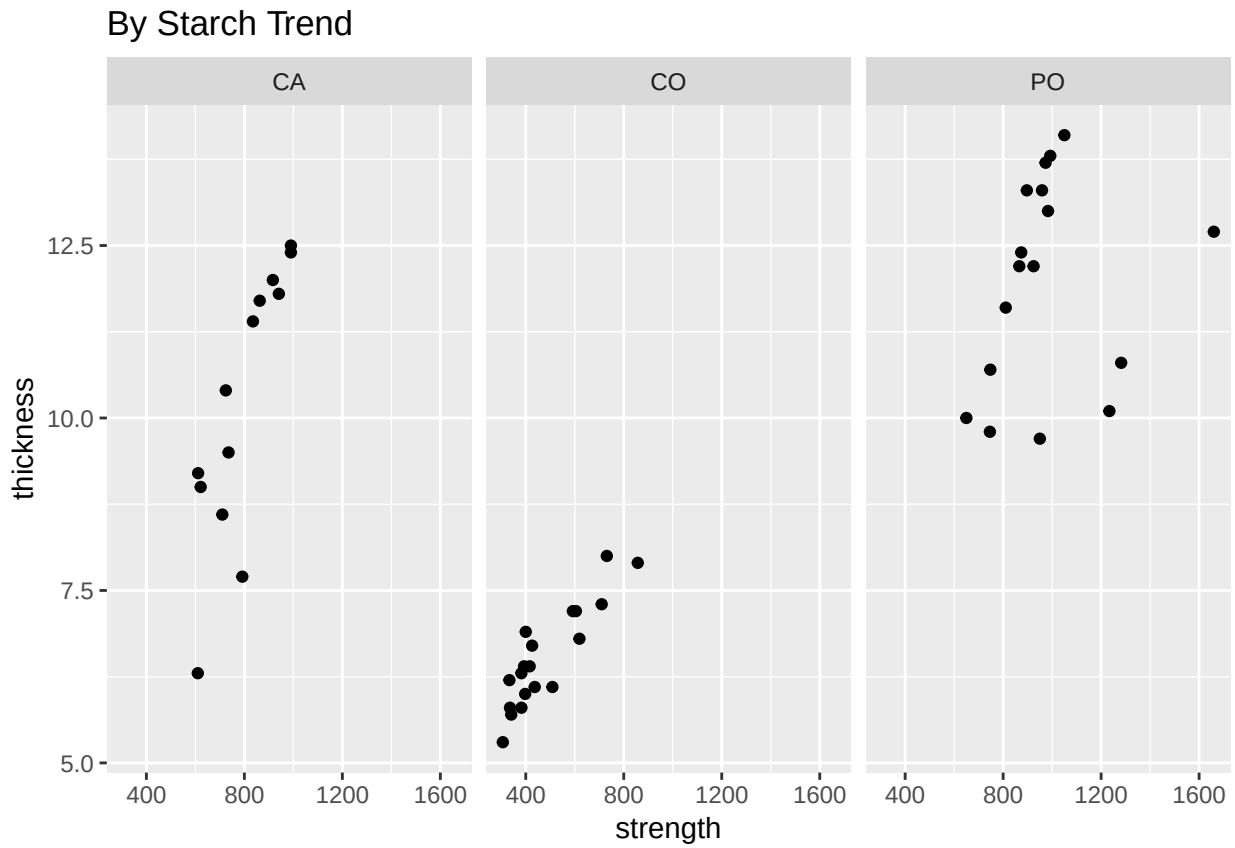
```
library(data.table)
library(tidyr)
library(ggplot2)
url1 <- "https://www2.isye.gatech.edu/~jeffwu/book/data/starch.dat"
Dat1 <- fread(url1, sep = " ")
str(Dat1)
```

```
## Classes 'data.table' and 'data.frame':   49 obs. of  3 variables:
##  $ starch   : chr  "CA" "CA" "CA" "CA" ...
##  $ strength : num  792 610 710 941 990 ...
##  $ thickness: num  7.7 6.3 8.6 11.8 12.4 12 11.4 10.4 9.2 9 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

```
ggplot(Dat1, aes(strength, thickness)) + geom_point(aes(col = starch)) +
    geom_smooth() + ggtitle("Overall Trend")
```




```
ggplot(Dat1, aes(strength, thickness)) + geom_point() + facet_wrap(~starch) +
    ggtitle("By Starch Trend")
```



It appears overall that there is a moderate trend between strength and thickness, as well as trends formed by the starch variables. It seems that the starch variable causes increases in both thickness and strength depending on what starch type you are looking at, the CO starch is lowest in both strength and thickness while the PO starch is highes in both strength and thickness. When looking at the overall plot it is easy to see that there appears to be at least 1 outlier (1700,12.5), this point definitely would affect the overall slope and intercept coefficients of a regression line and should be investigated. But when you look at the plots by starch the same point does not seem as extreme, but I would still suggest investigating that point.

Problem 6

```
library(knitr)
library(downloader)
library(data.table)
states <- fread(input = "./us_cities_and_states/states.sql",
    skip = 23, sep = "'", sep2 = ",", header = F, select = c(2,
        4))
colnames(states) <- c("Name", "Initial")
states_missing <- data.frame(Name = c("Alabama", "Alaska", "Arizona",
    "Arkansas"), Initial = c("AL", "AK", "AZ", "AR"))
states <- data.frame(rbind(states_missing, states))
states$Name <- as.character(states$Name)
states$Initial <- as.character(states$Initial)
cities <- fread(input = "./us_cities_and_states/cities_extended.sql",
    sep = "'", sep2 = ",", header = F, select = c(2, 4))
```

```
cities <- subset(cities, V4 %in% states$Initial)
table(cities$V4)
```

```
##
##   AK   AL   AR   AZ   CA   CO   CT   DC   DE   FL   GA   HI   IA   ID   IL
##  273  838  709  532 2651  659  438  284   98 1487  972  139 1060  325 1587
##   IN   KS   KY   LA   MA   MD   ME   MI   MN   MO   MS   MT   NC   ND   NE
##  989  756  961  725  703  619  489 1170 1031 1170  533  405 1090  407  620
##   NH   NJ   NM   NV   NY   OH   OK   OR   PA   RI   SC   SD   TN   TX   UT
##  284  733  426  253 2207 1446  774  484 2208   91  539  394  795 2650  344
##   VA   VT   WA   WI   WV   WY
## 1238  309  732  898  859  195
```

```
getCount <- function(letter, state_name) {
    if (!is.character(letter) | !is.character(state_name)) {
        return("Either the state name of letter was not a character")
    } else {
        state_name <- tolower(state_name)
        temp <- strsplit(state_name, "")
        count <- vector()
        for (i in 1:length(letter)) {
            count[i] <- sum(unlist(temp) %in% letter[i])
        }
        return(count)
    }
}
letter_count <- data.frame(matrix(NA, nrow = 51, ncol = 26))
for (i in 1:51) {
    letter_count[i, ] <- getCount(letters, states$Name[i])
}
colnames(letter_count) <- letters
rownames(letter_count) <- states$Name
# ALABAMA
getCount(letters, states$Name[1])
```

```
##  [1] 4 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
```

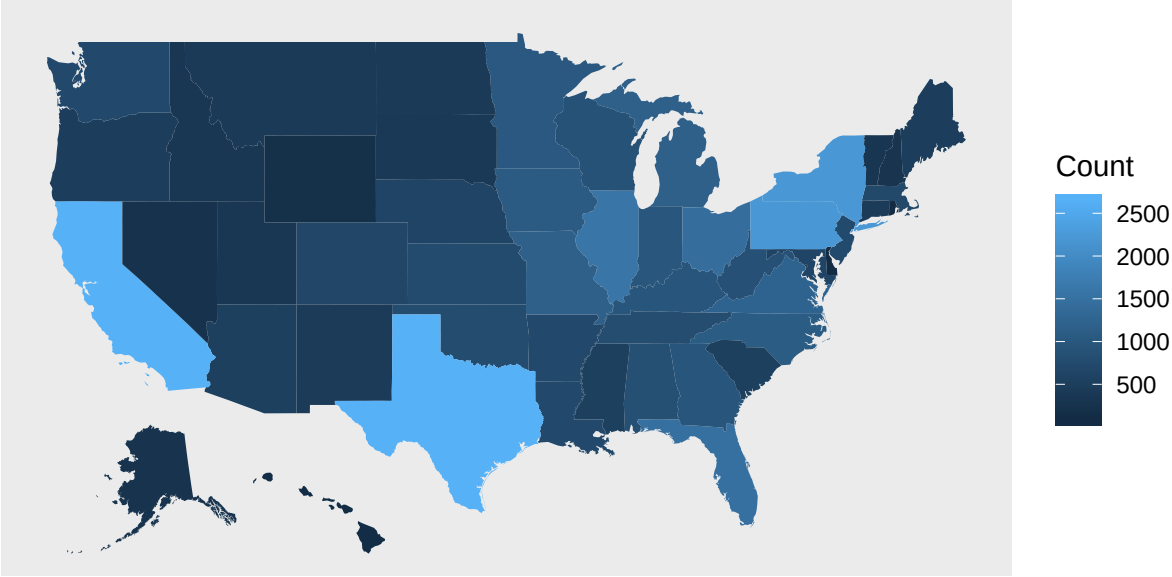
```
library(ggplot2)
colnames(cities) <- c("Name", "Initial")
freq <- data.frame(table(cities$Initial))
colnames(freq) <- c("Initial", "Count")
states <- merge(states, freq, by = "Initial")
states$Name <- tolower(states$Name)


library(fiftystater)
```

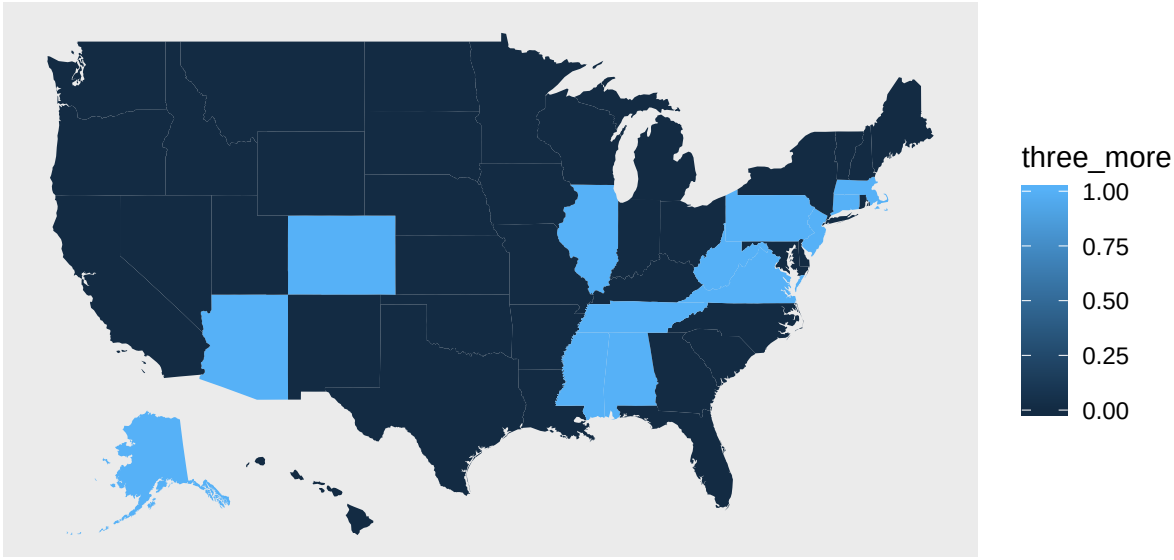
```
## Warning: package 'fiftystater' was built under R version 3.3.3
```

```
data("fifty_states")  # this line is optional due to lazy data loading
crimes <- data.frame(state = tolower(rownames(USArrests)), USArrests)

ggplot(states, aes(map_id = Name)) + geom_map(aes(fill = Count),
    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 = "")
```




```
states_binary <- states[, c(2)]
three_more <- vector()
three_more <- ifelse(unname(apply(letter_count, 1, max)) > 2,
    1, 0)
states_binary <- data.frame(cbind(states_binary, three_more))
colnames(states_binary) <- c("Name", "three_more")
states_binary$Name <- as.character(states_binary$Name)
states_binary$three_more <- as.numeric(as.character(states_binary$three_more))
ggplot(states_binary, aes(map_id = Name)) + geom_map(aes(fill = three_more),
    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 = "")
```


three_more
1.00
0.75
0.50
0.25
0.00