Unit\_8\_Preso

Michael Hedge and Morgan Nelson

February 17, 2020

Greetings Mr. CEO and Mr. CFO, other distinguished guests.

Given the Beer and Brewery data, our research intends to answer several compelling questions for the business: 1. How many breweries are present in each state?

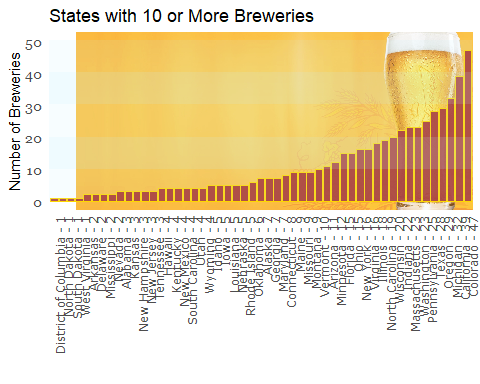
beers <- read.csv("./Beers.csv")  
breweries <- read.csv("Breweries.csv")  
  
# Question 2  
# Merge beer data with the breweries data.   
# Print the first 6 observations and the last six observations to check the merged file.  
  
# Merge the two data frames into one  
data <- merge(beers,breweries,by.x = "Brewery\_id", by.y = "Brew\_ID")  
data <- data %>%   
 rename(Beer = Name.x) %>% # clean up munged names from merge  
 rename(Brewery = Name.y) %>% # clean up munged names from merge  
 rename(Brewery\_ID = Brewery\_id) # Make anything with \_ID the same  
  
head(data, n = 6)

## Brewery\_ID Beer Beer\_ID ABV IBU  
## 1 1 Get Together 2692 0.045 50  
## 2 1 Maggie's Leap 2691 0.049 26  
## 3 1 Wall's End 2690 0.048 19  
## 4 1 Pumpion 2689 0.060 38  
## 5 1 Stronghold 2688 0.060 25  
## 6 1 Parapet ESB 2687 0.056 47  
## Style Ounces Brewery City  
## 1 American IPA 16 NorthGate Brewing Minneapolis  
## 2 Milk / Sweet Stout 16 NorthGate Brewing Minneapolis  
## 3 English Brown Ale 16 NorthGate Brewing Minneapolis  
## 4 Pumpkin Ale 16 NorthGate Brewing Minneapolis  
## 5 American Porter 16 NorthGate Brewing Minneapolis  
## 6 Extra Special / Strong Bitter (ESB) 16 NorthGate Brewing Minneapolis  
## State  
## 1 MN  
## 2 MN  
## 3 MN  
## 4 MN  
## 5 MN  
## 6 MN

tail(data,n = 6)

## Brewery\_ID Beer Beer\_ID ABV IBU  
## 2405 556 Pilsner Ukiah 98 0.055 NA  
## 2406 557 Heinnieweisse Weissebier 52 0.049 NA  
## 2407 557 Snapperhead IPA 51 0.068 NA  
## 2408 557 Moo Thunder Stout 50 0.049 NA  
## 2409 557 Porkslap Pale Ale 49 0.043 NA  
## 2410 558 Urban Wilderness Pale Ale 30 0.049 NA  
## Style Ounces Brewery City  
## 2405 German Pilsener 12 Ukiah Brewing Company Ukiah  
## 2406 Hefeweizen 12 Butternuts Beer and Ale Garrattsville  
## 2407 American IPA 12 Butternuts Beer and Ale Garrattsville  
## 2408 Milk / Sweet Stout 12 Butternuts Beer and Ale Garrattsville  
## 2409 American Pale Ale (APA) 12 Butternuts Beer and Ale Garrattsville  
## 2410 English Pale Ale 12 Sleeping Lady Brewing Company Anchorage  
## State  
## 2405 CA  
## 2406 NY  
## 2407 NY  
## 2408 NY  
## 2409 NY  
## 2410 AK

# font\_import()  
# extrafont::loadfonts(device="win")  
  
# The following lines create a table that has the number of breweries  
# per state, then tidys up the columns into a format the usmaps library  
# understands  
brewery\_count\_by\_state <- breweries %>% group\_by(State) %>% summarise(count=n())  
  
  
brewery\_count\_by\_state$State = as.character(brewery\_count\_by\_state$State)  
brewery\_count\_by\_state$State = str\_trim(brewery\_count\_by\_state$State)  
brewery\_count\_by\_state = brewery\_count\_by\_state %>% mutate(state=State)  
brewery\_count\_by\_state = brewery\_count\_by\_state[,3:2]  
  
color\_boxes <- data.frame(y1=seq(0,40,10),   
 y2=seq(10,50,10),  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 5))  
  
brewery\_count\_by\_state$state <- factor(brewery\_count\_by\_state$state, levels = brewery\_count\_by\_state$state[order(brewery\_count\_by\_state$count)])  
  
# style from https://rpubs.com/chidungkt/392980  
  
# read in the photo background  
image <- readJPEG("beer\_glass\_background.jpg")  
# scale the background for this chart  
bg <- rasterGrob(image, height = unit(1.5, "npc"), x=0.62, y=0.5)  
  
brewery\_count\_by\_state %>%   
 # filter(count > 9) %>%   
 mutate(fullname = paste(abbr2state(state), count, sep = " - ")) %>%   
 ggplot() +   
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_bar(aes(x=reorder(fullname, count), y = count),stat="identity",   
 fill = "brown",   
 color = "gold",  
 alpha = 0.9) +   
 geom\_rect(data=color\_boxes,  
 mapping=aes(ymin = y1, ymax = y2, xmin = -Inf, xmax = +Inf),  
 fill = color\_boxes$color) +  
 theme\_minimal() +  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10),  
 axis.text.x = element\_text(family = "Verdana", angle=90, hjust = 1)) +  
 labs(title = "States with 10 or More Breweries", x="", y="Number of Breweries")# +



# coord\_flip()

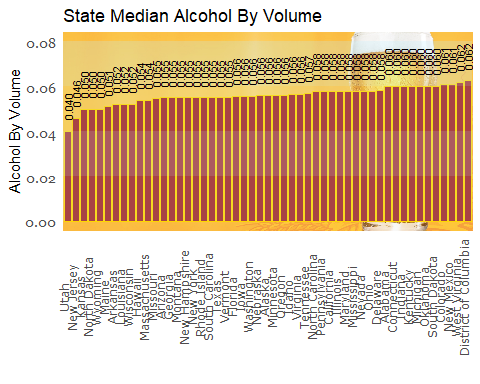
While accurate this graphic is less than helpful, so we’ve put the breweries on a map.

To perform the rest of the analysis, the data was checked for a number of inconsistencies including missing values. Over half of the records(1405 of 2410) in the dataset did not have values for the IBU, and 62 of the entries did not have a value for ABV.

To allow us to continue without removing any of the records for missing values, it was decided to impute the data with a k-NN on beers in the same “Style”. What this means is that for a given beer, the algorithm will look at the collection of beers, find ones that match with it best, determine an appropriate IBU or ABV value, and replace the missing value in the sample.

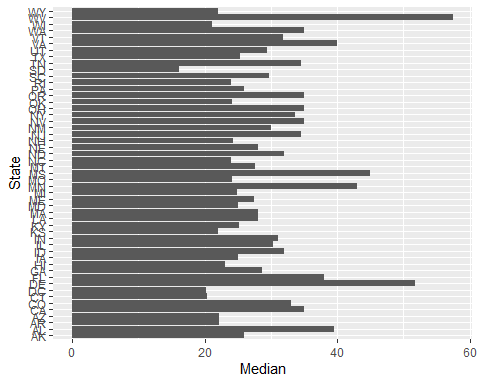
# use the knnImputation() function from the Date Mining with R (DMwR) library to fill in missing data  
adjData <- data %>% knnImputation()

# read in the photo background  
image <- readJPEG("beer\_glass\_background.jpg")  
# scale the background for this chart  
bg <- rasterGrob(image, height = unit(1.5, "npc"), x=0.5, y=0.5)  
  
# create the alternating color background  
color\_boxes <- data.frame(y1=seq(0.00, .06, .02),   
 y2=seq(0.02, .08, .02),  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 4))  
  
adjData %>%   
 group\_by(State) %>%   
 summarize(medabv=median(ABV)) %>%  
 mutate(State = abbr2state(str\_trim(as.character(State)))) %>% # turn 2 letter factor into full state name  
 ggplot() +  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_bar(aes(x=reorder(State, medabv), y=medabv),stat="identity", color="gold", fill="brown") +  
 geom\_rect(data=color\_boxes,  
 mapping=aes(ymin = y1, ymax = y2, xmin = -Inf, xmax = +Inf),  
 fill = color\_boxes$color) +  
 geom\_text(aes(x=reorder(State, medabv), y=medabv, label=format(round(medabv,3), nsmall = 3)), size = 3, angle=90, hjust = -.4) +  
 scale\_y\_continuous(limits = c(0, 0.08)) +  
 theme\_minimal() +  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10),  
 axis.text.x = element\_text(family = "Verdana", angle=90, vjust = .02)) +  
 labs(title = "State Median Alcohol By Volume", x="", y="Alcohol By Volume") # +

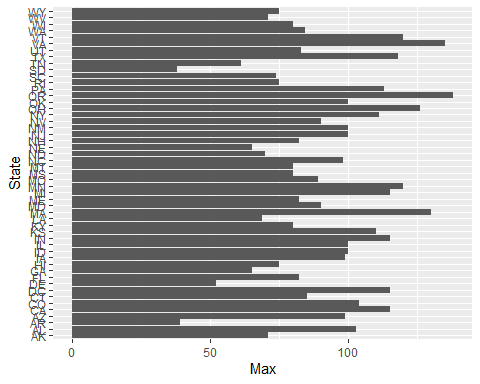


# coord\_fixed(ratio=200)

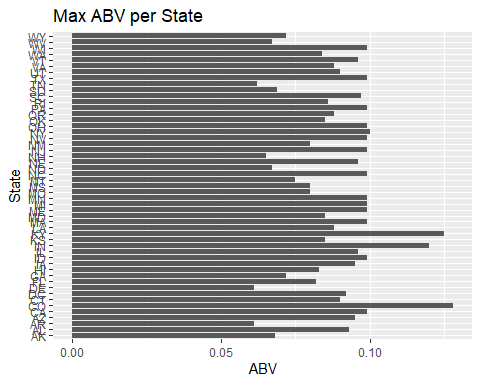
adjData %>% group\_by(State) %>% summarize(Median=median(IBU)) %>%   
 ggplot(aes(x=State, y=Median)) +  
 geom\_bar(stat="identity") +  
 coord\_flip()



adjData %>% group\_by(State) %>% summarize(Max=max(IBU)) %>%   
 ggplot(aes(x=State, y=Max)) +  
 geom\_bar(stat="identity") +  
 coord\_flip()



adjData %>% group\_by(State) %>% summarize(Max=max(ABV)) %>%   
 ggplot(aes(x=State, y=Max)) +  
 geom\_bar(stat="identity") +  
 coord\_flip() +  
 labs(title="Max ABV per State", y="ABV", x="State")



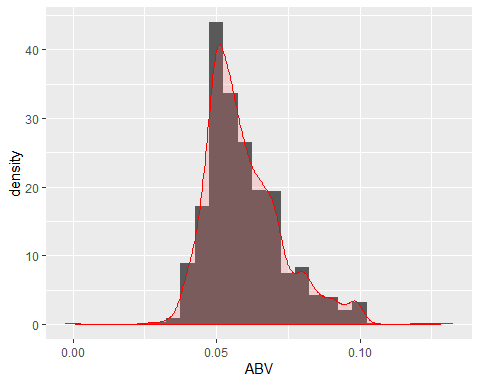
The summary statistics of the ABV by state show a minimum ABV of 0.001, or 0.1% Alcohol by Volume. This is either a data collection error or a set of Non-Alcoholic beers. The maximum ABV is 12.8%, and the mean ABV is 5.9% with 50% of the beers in the sample ranging from 5.0% to 6.7%. A case could be made to discard the beers with minimum ABV as “not-beers” but rather “beer-like-beverages”, as they are not subject to the same laws regulating the sale of alcoholic beverages. Alternately, many persons are not allowed to drink alcohol because of religious beliefs or medical reasons, so it may be worth while to keep the non-alcoholic beers in the study.

adjData %>% select(ABV) %>% summary()

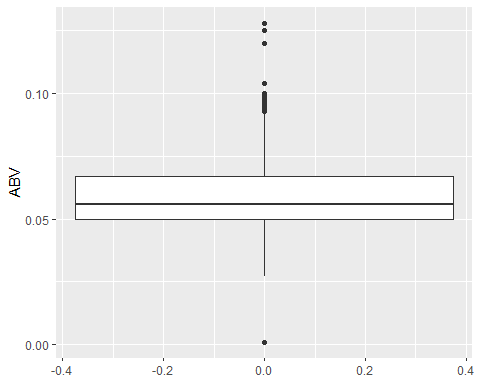
## ABV   
## Min. :0.00100   
## 1st Qu.:0.05000   
## Median :0.05600   
## Mean :0.05971   
## 3rd Qu.:0.06700   
## Max. :0.12800

The distribution of Alcohol by Volume peaks at 5.9%, and shows a right skew, indicating that the set of beers sampled show a tendency to start at around 5% ABV, with a large grouping of beers in that 5.0%-6.7%, then a long tail reaching into higher and higher ABV numbers. Fewer beers were below the 5.0% ABV mark. From this graph, the distribution of ABV in beers appears to be more logarithmic than normal.

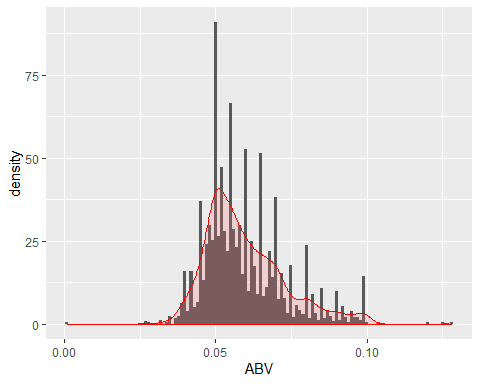
adjData %>% filter(!is.na(ABV)) %>% ggplot(aes(x=ABV)) +  
 geom\_histogram(aes(y=..density..),binwidth = 0.005, na.rm = TRUE, show.legend = TRUE) +  
 geom\_density(alpha=0.2, fill="#FF6666", color="red")

 The Boxplot of ABV gives a better look at the number of outliers on each end of the ABV values. Below the bottom whisker, we see a small number of outliers near 0.1% ABV, with many more outliers above the top whisker ranging from about 6.7% to 13% ABV.

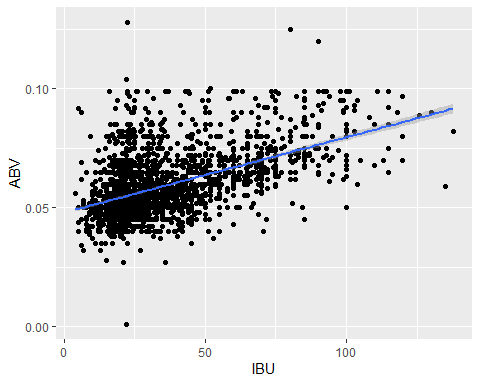
adjData %>% filter(!is.na(ABV)) %>% ggplot() +  
 geom\_boxplot(aes(y=ABV))

 Another interesting point about the histogram, the values of ABV do not appear to be purely continuous. Instead, if we choose a smaller bin width on the ABV values, we can see a definite trend to have a beer with a specific ABV.

adjData %>% filter(!is.na(ABV)) %>% ggplot(aes(x=ABV)) +  
 geom\_histogram(aes(y=..density..),binwidth = 0.001, na.rm = TRUE, show.legend = TRUE) +  
 geom\_density(alpha=0.2, fill="#FF6666", color="red")

 Plotting ABU vs. ABV as dots, then adding a linear model regression line shows an “up and to the right” correlation between IBU and ABV.

adjData %>% select(c(IBU, ABV)) %>% ggplot() +  
 geom\_point(aes(x=IBU, y=ABV), position = "jitter", na.rm = TRUE) +  
 geom\_smooth(aes(x=IBU, y=ABV), method="lm", na.rm = TRUE)

 A more concrete indication of this correlation is given by a different plot. Here, we are shown that the correlation is 0.67, or 67%. While this shows a correlation, it is not a “strong” one, as there is a lot of variance between IBU and ABV.

adjData %>% select(c(IBU,ABV)) %>% ggpairs()

