Beer Case Study: Jordan Eaddy
Code

```
---
title: "**Beer Case Study:Jordan Eaddy**"
output:
  pdf_document: default
  html_notebook: default
  html_document: default
   code_folding: show
  numbers_sections: true
   fig_width: 7
    fig_height: 6
    fig_caption: true
editor_options:
  chunk_output_type: inline
---
#This Chunk is where I check in on all of the missing values, load my libraries and merge the dataset by two different identifiers. I also took all the NA's and used the column average to fill those in. Then I scaled my data since we are going to use a KNN model later on.
```{r Setup, eval=FALSE}
library(tidyverse)
library(knitr)
library(class)
library(caret)
library(e1071)
library(stringr)
library(plotly)
library(ggthemes)

str(Beer)
str(Breweries)
#Join both datasets on their respective ID's and change the name of a few columns
Brew<-merge(Breweries,Beer, by.x ="Brew_ID", by.y = "Brewery_id")

Brew<- Brew %>%
  rename(
    Brewery=Name.x,
    Beer=Name.y
  )
#Checking in on missing values
library(naniar)
gg_miss_var(Brew)
#Sanity check
sapply(Brew,function(x)sum(is.na(x)))
```
```

```
#Replacing the NA's in IBU with the column average
Beer$IBU[is.na(Beer$IBU)] <-mean(Beer$IBU, na.rm=TRUE)
#Replacing the NA's in  ABV with the column average
Beer$ABV[is.na(Beer$ABV)] <-mean(Beer$ABV, na.rm=TRUE)
#Converting ABV to a percent for better readability
library(scales)
Beer$ABV<-percent(Beer$ABV, accuracy = 0.1)
Beer$ABV <- as.numeric(sub("%","",Beer$ABV))
```

Pretty straight forward here. I plotted the breweries for each state using a bar graph. I was able to plot the amount over head as well.

**##1.How many Brweries are present in each state?**

```{r}
table(Breweries$State)

Breweries%>%
  ggplot(aes(State))+
  geom_histogram(stat = "count")+
  geom_text(stat = 'count', aes(label=..count..), vjust=-1)+
  theme_economist()+
  ggtitle("Breweries Per State")+
  xlab("State")+
  theme_economist()+
  scale_x_discrete(guide = guide_axis(n.dodge = 2))
```

I merged my data in the first chunk. This is just showing the head and tail of my dataset called "Brew"

**#2. Merge the beer data with the breweries data. Print the first 6 observations and the last six observations to check the merged file.**

```{r}
head(Brew,10)
tail(Brew, 10)
```

For the missing values, I used the column mean for the respective variable. I figured since it was only a handful that were missing(realative to the size of the data set),I would not be better of dropping them

**#3.Address the missing values in each column**

```{r Missing Value}
sapply(Brew,function(x)sum(is.na(x)))

Beer$IBU[is.na(Beer$IBU)] <-mean(Beer$IBU, na.rm=TRUE)
#Replacing the NA's in  ABV with the column average
Beer$ABV[is.na(Beer$ABV)] <-mean(Beer$ABV, na.rm=TRUE)

```

I put the data into long form so I could plot a side-by-side bar chart. Once I was able to use pivot_longer and get it into the format I wanted. Plotting the bar chart became elementary. If I could do this again (without messing up the following outputs), I would have normalized the means so that ABV was not such a "little brother" to the IBU variable.

**#4. Compute the median alcohol content and international bitterness unit for each state. Plot a bar chart to compare.**

```{r}
brewmeans<-Brew %>%
  group_by(State) %>%
  summarise(Mean_IBU=mean(IBU),Mean_ABV=mean(ABV))
#Putting the data in long form
meansbrew<-tidyr::pivot_longer(brewmeans, cols=c('Mean_IBU', 'Mean_ABV'),
names_to='variable',
values_to="value")

#Visualizing Above
ggplot(meansbrew, aes(x=State, y= value, fill=variable))+
  geom_bar(stat='identity', position = 'dodge')+
  theme_economist()+
  ggtitle("Mean ABV and IBU Per State")+
  xlab("State")+
  ylab("IBU and ABV Units")+
  scale_x_discrete(guide = guide_axis(n.dodge = 3))
```

The TOP three states for IBU were WV, AL and ID. The top three for ABV were NV,DC and KY. I arranged this to get a tables view of each variable. I then assigned it a variable in case of later usage.

**#5.Which state has the highest ABV and which state has the highest IBU?**

```{r}
arrange(Brew, desc(ABV)) #Colorado has the highest ABV
arrange(Brew, desc(IBU))#Oregon has the most bitter Beer
```

```
State_Mean<-Brew %>%
  group_by(State) %>%
  summarise(Mean_IBU=mean(IBU),Mean_ABV=mean(ABV))
arrange(State_Mean, desc(Mean_ABV))#Nevada has the highest average ABV
arrange(State_Mean, desc(Mean_IBU))#West Virginia has highest average IBU
```

The vtable call was new to me. I liked the output it gave, but I wish it came in a more usable format. The summary statistics and distribution graphic told me that there is some right skewness going on. I can also see that most of my data falls between 5.00 and 6.70. The standard deviation was small

**#6. Comment on the summary statistics and the distribution of the ABV variable**

```{r echo=TRUE}
library(vtable)
st(Brew)
summary(Brew)

#Plot of the Distribution
ggplot(Brew, aes(x=ABV))+
  geom_histogram(binwidth = .5, colour="black", fill="white")+
  geom_vline(aes(xintercept=mean(ABV,na.rm=T)),
                         color="red", linetype="dashed", size=1)+
  labs(title = "Distribution of ABV",
       caption="Red Line Indicates Mean ABV")+
  theme_economist()

#It looks like we have some right skewness going on in the plot below
```

Just to reiterate and make sure my data was scaled this time around, I think there are signs of a relationship here. We get some noise early on in the graphic, but once we get past that, there appears to be a linear trend.

**#7. Is there a relationship between the bitterness of the beer and its alcoholic content? Draw a scatter plot!**

```{r echo=TRUE}
#First Lets Standardize

Brew$ABVscaled<-scale(Brew$ABV)
Brew$IBUscaled<-scale(Brew$IBU)

#Now that we have added two new columns of standardized ABV and IBU, now we can create a scatter plot

Brew %>%
  ggplot(aes(x=IBUscaled, y=ABVscaled))+
```

```
  geom_point()+
  geom_smooth()+
  theme_economist()+
  labs(title = "Scatter Plot of Scaled IBU and ABV",
       x="IBU", y="ABV")

#There appears to be a relationship going on with noise around it. I have a preliminary hunch that this potentially dependent on style of beer
```

Before I began to model, I wanted to run a loop to get all the styles into two categories, ALE and IPA. I used grep to find all the styles that had either one of those in the name. I dropped the rest that had neither of the two

**#8A.Getting our data prepared to model**

```{r echo=TRUE}
#I want all the beers that have IPA or ALE in the Style column
ipale<-filter(Brew,grepl('IPA|Ale',Style))
sapply(ipale,function(x)sum(is.na(x))) #Quick sanity check
#In order to model, we need a consistent name in the style category, the following loop will use a logical call to do that
for (i in 1:1534) {
  if (is.na(str_match(ipale[i,9],".Ale"))) {
    ipale[i,9] <- "IPA"
  } else {
    ipale[i,9] <- "ALE"

  }
}

ipale
```

I wanted to tune k to find the most accurate one first. This came out to be 7 after running the average of 300 iterations

**#8B.With respect to IBU and ABV, use a KNN model to investigate the difference between IPAs and Ale's**

```{r echo=TRUE}
#First lets standardize by putting ABV and IBU on the same scale
ipale$ABVscaled<-scale(ipale$ABV)
ipale$IBUscaled<-scale(ipale$IBU)
#Model...BUT WAIT! How do we know what is the most accurate K to use?

splitpercentage=.75
iterations = 300 #300 INTERATIONS FOR EACH K UP TO 30
numks = 30
```

```
masterAcc = matrix(nrow = iterations, ncol = numks)

for(j in 1:iterations)
{
accs = data.frame(accuracy = numeric(30), k = numeric(30))
trainIndices = sample(1:dim(ipale)[1],round(splitpercentage * dim(ipale)[1]))
train = ipale[trainIndices,]
test = ipale[-trainIndices,]
for(i in 1:numks)
{
  classifications = knn(train[,c(11,12)],test[,c(11,12)],train$Style, prob = TRUE, k = i)
  table(classifications,test$Style)
  CM = confusionMatrix(table(classifications,test$Style))
  masterAcc[j,i] = CM$overall[1]
}

}

MeanAcc = colMeans(masterAcc) #THE MEAN OF THE ACCURACIES

plot(seq(1,numks,1),MeanAcc, type = "l")

#It looks like 7 is the most accurate K!
```

Ran the KNN model and it turned out to be pretty accurate at 85%. The number of correctly identified divided by the number of true successes(Sensitivity) came out to be 89%. Looking at balanced accuracy, which can help us evaluate how good a binary classifier is, it alluedes to this being a good model. Especially since there was some data imbalance with IPA and ALE.

#8C.KNN Model

```{r echo=TRUE}
train_knn = sample(1:dim(ipale)[1],round(splitpercentage * dim(ipale)[1]))
# I want to sample from 1: all the way to the dimensions of ipale
# Then I want to round that split percentage times that dimension
train = ipale[train_knn,]
# creating the training set
test = ipale[-train_knn,]

classifications = knn(train[,c(11,12)],test[,c(11,12)],train$Style, prob = TRUE, k = 7)
table(classifications,test$Style) #Prints the Truth
confusionMatrix(table(classifications,test$Style))

```

Here I visualized the model output. This shows us that data imbalance I spoke about previously.
#8D. Visual of Model output
```{r echo=TRUE}
CM=table(classifications,truth)
CM2=confusionMatrix(CM)

table <- data.frame(CM2$table)

plotTable <- table %>%
  mutate(goodbad = ifelse(table$classifications == table$truth, "good", "bad")) %>%
  group_by(truth) %>%
  mutate(prop = Freq/sum(Freq))

ggplot(data = plotTable, mapping = aes(x = truth, y = classifications, fill = goodbad, alpha =
prop)) +
  geom_tile() +
  geom_text(aes(label = Freq), vjust = .5, fontface  = "bold", alpha = 1) +
  scale_fill_manual(values = c(good = "blue", bad = "gray")) +
  theme_economist() +
  xlim(rev(levels(table$truth)))+
  labs(title="Visualization of KNN Confusion Matrix",
       caption="by Jordan Eaddy",
       x="TRUTH",
       y="CLASSIFICATIONS")

```

I was a business major in undergrad so wanting to know target markets was the first thing that came to my head on this question. I filtered all the states that were below average in IBU,ABV and Ounce of beer. I then plotted those to see which states could possibly be targets for Budweiser if they chose to expand.

**#9. Find a useful observation that Budweiser may be able to find value in**

```{r echo=TRUE}
bbrewmean<-ipale %>%
  select(State,IBU,ABV,Ounces,Style) %>%
  group_by(State) %>%
  summarise(Mean_IBU=mean(IBU),Mean_ABV=mean(ABV),mean_OU=mean(Ounces))

Works<-merge(bbrewmean,ipale,by="State")


#Amount of Below Average per city
table(Lowstates$City)
#Amount of Below Average per state
table(Lowstates$State)
```

```
#Plotly of STYLES
p1<-Works %>%
  ggplot(aes(x=mean_OU,y=Mean_ABV, color=State))+
  geom_jitter()+
  facet_wrap(~Style)+
  theme_economist_white()+
   labs(title="Styles in Respect to ABV and Ounces",
       caption="by Jordan Eaddy",
       x="Average Ounces",
       y="Average ABV")



#Categorizing the underachieving states
str(Lowstates)
Lowstates<-Works %>%
  filter(Mean_ABV<6,Mean_IBU<47,mean_OU<13)


#Blew Average Plotly of Ounces vs ABV
p2<-Lowstates %>%
 ggplot(aes(x=mean_OU,y=Mean_ABV, color=State))+
  geom_jitter()+
  facet_wrap(~Style)+
  theme_economist_white()+
   labs(title="Target States ABV",
       caption="by Jordan Eaddy",
       x="Ounces",
       y="ABV")

#Below Average Plotly of Ounces vs IBU
p2a<-Lowstates %>%
 ggplot(aes(x=mean_OU,y=Mean_IBU, color=State))+
  geom_point()+
  facet_wrap(~Style)+
  theme_economist_white()+
   labs(title="Target States IBU",
       caption="by Jordan Eaddy",
       x="Ounces",
       y="IBU")

#Plotly outputs
```

```
ggplotly(p1)
ggplotly(p2)
ggplotly(p2a)
```