Advanced analytics on Budweiser data

Simerpreet Reddy

Jan/19/2020

## Purpose

The purpose of the code is to perform anlaysis on Budweiser Beers and Breweries data. Beers data provides the different measures of the Beer, such as it’s name, ABV(Alcohal By Volume), IBU(Internation Bitternes Unit), Beer Style and the Brewery it comes from.

Breweries data on the other hand provides the name of the Brewery along with the City and State it is located in. Analysis is done to provide insights from Beers and Breweries data on following aspects: Brewery and beer distribution by State. Comparison of ABV and IBU by-  
State, and Beer Style – IPA and Other Ales.

#Include the necessary libraries  
library(tidyr)  
library(tidyverse)

## -- Attaching packages ------------------------------------------------------------------------------ tidyverse 1.2.1 --

## v ggplot2 3.2.1 v purrr 0.3.3  
## v tibble 2.1.3 v dplyr 0.8.3  
## v readr 1.3.1 v stringr 1.4.0  
## v ggplot2 3.2.1 v forcats 0.4.0

## -- Conflicts --------------------------------------------------------------------------------- tidyverse\_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag() masks stats::lag()

library(dplyr)  
library(ggplot2)  
library(GGally)

## Registered S3 method overwritten by 'GGally':  
## method from   
## +.gg ggplot2

##   
## Attaching package: 'GGally'

## The following object is masked from 'package:dplyr':  
##   
## nasa

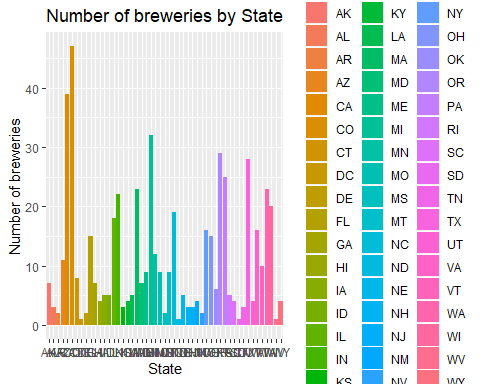
library(class)  
library(caret)

## Loading required package: lattice

##   
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':  
##   
## lift

#Step 1: Import the source files  
#The provided source file have commas in data values, so I used "|" as the delimiter.The files have headers so, mark header=TRUE  
beers\_raw <- read.csv(file.choose(), header = TRUE, sep = "|" )  
breweries\_raw <- read.csv(file.choose(), header = TRUE, sep = "|" )  
  
#Answering the questions  
#Question 1. How many breweries are present in each state  
#Solution  
#Plot beweries by State  
ggplot(breweries\_raw) + geom\_bar(mapping = aes(x=State, fill = State)) + ggtitle("Number of breweries by State") +  
 ylab("Number of breweries") + xlab("State")



# Question 2.Merge beer data with the breweries data. Print the first 6 observations and the last six observations to check the merged file. (RMD only, this does not need to be included in the presentation or the deck.)  
#Solution  
#Step 1 Left join Beers data with Breweries data based on the column "Brewery\_id" in Breweries data set and "Brew\_ID" in Beer data set  
Beer\_Breweries\_raw <- left\_join(beers\_raw, breweries\_raw, by = c("Brewery\_id" = "Brew\_ID" ), all = TRUE )  
  
#Step 2 Rename the columns of the merged dataset for better understanding  
colnames(Beer\_Breweries\_raw) <- c('Beer\_Name', 'Beer\_Id', 'ABV', 'IBU', 'Brewery\_Id', 'Beer\_Style', 'Ounces','Brewery\_Name', 'City','State')  
  
#Step 3 Printe first 6 rows of the data  
head(Beer\_Breweries\_raw)

## Beer\_Name Beer\_Id ABV IBU Brewery\_Id  
## 1 Pub Beer 1436 0.050 NA 409  
## 2 Devil's Cup 2265 0.066 NA 178  
## 3 Rise of the Phoenix 2264 0.071 NA 178  
## 4 Sinister 2263 0.090 NA 178  
## 5 Sex and Candy 2262 0.075 NA 178  
## 6 Black Exodus 2261 0.077 NA 178  
## Beer\_Style Ounces Brewery\_Name City  
## 1 American Pale Lager 12 10 Barrel Brewing Company Bend  
## 2 American Pale Ale (APA) 12 18th Street Brewery Gary  
## 3 American IPA 12 18th Street Brewery Gary  
## 4 American Double / Imperial IPA 12 18th Street Brewery Gary  
## 5 American IPA 12 18th Street Brewery Gary  
## 6 Oatmeal Stout 12 18th Street Brewery Gary  
## State  
## 1 OR  
## 2 IN  
## 3 IN  
## 4 IN  
## 5 IN  
## 6 IN

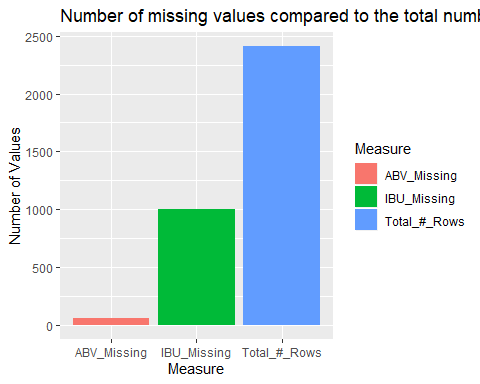
#Step 4 Print last 6 rows of the data  
tail(Beer\_Breweries\_raw)

## Beer\_Name Beer\_Id ABV IBU Brewery\_Id  
## 2405 Rocky Mountain Oyster Stout 1035 0.075 NA 425  
## 2406 Belgorado 928 0.067 45 425  
## 2407 Rail Yard Ale 807 0.052 NA 425  
## 2408 B3K Black Lager 620 0.055 NA 425  
## 2409 Silverback Pale Ale 145 0.055 40 425  
## 2410 Rail Yard Ale (2009) 84 0.052 NA 425  
## Beer\_Style Ounces Brewery\_Name City State  
## 2405 American Stout 12 Wynkoop Brewing Company Denver CO  
## 2406 Belgian IPA 12 Wynkoop Brewing Company Denver CO  
## 2407 American Amber / Red Ale 12 Wynkoop Brewing Company Denver CO  
## 2408 Schwarzbier 12 Wynkoop Brewing Company Denver CO  
## 2409 American Pale Ale (APA) 12 Wynkoop Brewing Company Denver CO  
## 2410 American Amber / Red Ale 12 Wynkoop Brewing Company Denver CO

#Question 3. Address the missing values in each column.  
#Solution  
#Check which columns of the data set have null values  
as.data.frame(lapply(Beer\_Breweries\_raw,function(x) { length(which(is.na(x)))}))

## Beer\_Name Beer\_Id ABV IBU Brewery\_Id Beer\_Style Ounces Brewery\_Name  
## 1 0 0 62 1005 0 0 0 0  
## City State  
## 1 0 0

#Create a dataframe to hold the column names with null values and the corresponding counts  
Missing\_Values <- data.frame(Measure = c("ABV\_Missing","IBU\_Missing", "Total\_#\_Rows") , Number\_of\_Values = c(62,1005,2410))  
  
#Create a plot to show the number of nulls in each column compared to the total number of rows in the data set.  
ggplot(data= Missing\_Values) + geom\_bar(mapping = aes(x= Measure, y= Number\_of\_Values, fill= Measure), stat="identity") + xlab("Measure") +  
 ylab("Number of Values") + ggtitle("Number of missing values compared to the total number of rows")



message("From the plot we see that the column ABV has 62 missing values and IBU has 1005 missing values.There's no other missing data.")

## From the plot we see that the column ABV has 62 missing values and IBU has 1005 missing values.There's no other missing data.

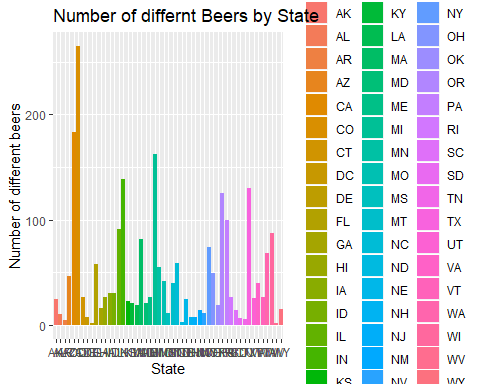
message(" The number of missing values is huge so we cannot just filter out the data. We have to replace the missing values with some meaningful data.")

## The number of missing values is huge so we cannot just filter out the data. We have to replace the missing values with some meaningful data.

#Step 2 Address missing values in column ABV  
  
message(" Factors that affect a beer's ABV and IBU - State the beer comes from and it's style")

## Factors that affect a beer's ABV and IBU - State the beer comes from and it's style

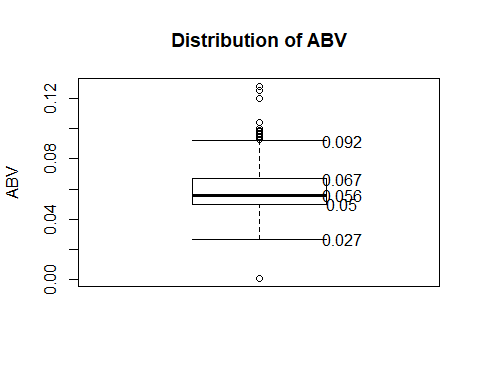
#Looking at beer distribution by State  
ggplot(Beer\_Breweries\_raw) + geom\_bar(mapping = aes(x=State, fill = State)) +   
 ggtitle("Number of differnt Beers by State") + ylab("Number of different beers")



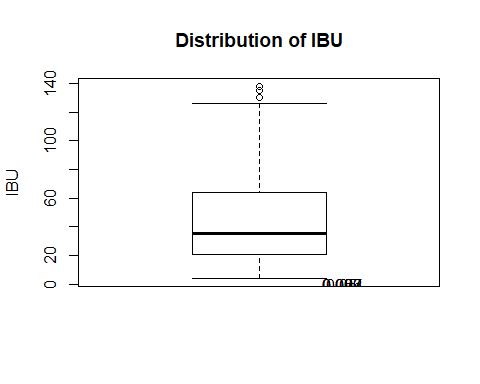
message("Looking at the beer distribution by State, we see that the data is not equally distributed. So, we do not want to consider State as a factor to calculate missing values. We will consider only Beer Category- based on beer style")

## Looking at the beer distribution by State, we see that the data is not equally distributed. So, we do not want to consider State as a factor to calculate missing values. We will consider only Beer Category- based on beer style

#Looking at ABV and IBU data to check if there are any outliers  
#ABV distribution  
boxplot(Beer\_Breweries\_raw$ABV, staplewex = 1, main = "Distribution of ABV", ylab = "ABV")  
text(y = boxplot.stats(Beer\_Breweries\_raw$ABV)$stats, labels = boxplot.stats(Beer\_Breweries\_raw$ABV)$stats, x = 1.25)



#IBU distribution  
boxplot(Beer\_Breweries\_raw$IBU, staplewex = 1, main = "Distribution of IBU", ylab = "IBU")  
text(y = boxplot.stats(Beer\_Breweries\_raw$ABV)$stats, labels = boxplot.stats(Beer\_Breweries\_raw$ABV)$stats, x = 1.25)



message("As the data has outliers and the mean is sensitive to outliers, we will replace the missing values by the corresponding median in each Beer Category")

## As the data has outliers and the mean is sensitive to outliers, we will replace the missing values by the corresponding median in each Beer Category

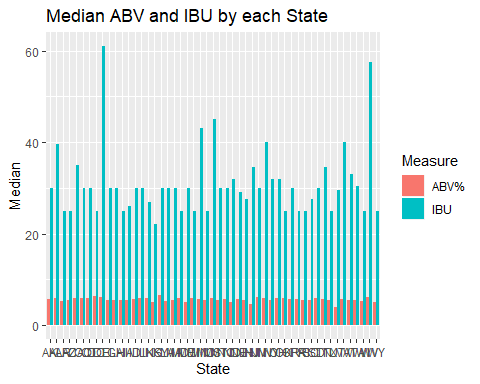
#Assigning Beer Categories: Read the Beer\_Style and assign the following categories: IPA- if the column Beer Style has IPA in its name, else 'Other Ales' if name has 'Ale' in its name and assign 'Other' to all the other types of beers.   
Beer\_Breweries\_raw$Beer\_Category <- ifelse(str\_detect(Beer\_Breweries\_raw$Beer\_Style, "IPA"), 'IPA', ifelse(str\_detect(Beer\_Breweries\_raw$Beer\_Name, "Ale"),'Other Ale','Other'))  
  
#Calculate median ABV for each Beer categy.  
Median\_ABV\_by\_St\_IPAorAle <- Beer\_Breweries\_raw%>% group\_by(Beer\_Category) %>% summarise(Median\_ABV = median(ABV, na.rm=TRUE))  
#Join the Median\_ABV data by State and Beer\_Category with the Beer\_Breweries\_raw data set  
Beer\_Breweries\_raw\_withMedianABV <-left\_join(Beer\_Breweries\_raw,Median\_ABV\_by\_St\_IPAorAle, by = c("Beer\_Category") )  
  
# Create a new column ABV\_New containing ABV values and replacing missing values by the above calculated median ABV  
Beer\_Breweries\_raw\_withMedianABV$ABV\_New <- ifelse(is.na(Beer\_Breweries\_raw\_withMedianABV$ABV), Beer\_Breweries\_raw\_withMedianABV$Median\_ABV, Beer\_Breweries\_raw\_withMedianABV$ABV)  
  
#Calculate the median IBU grouping by Beer\_Category.  
Median\_IBU\_by\_St\_IPAorAle\_ABV\_CAT <- Beer\_Breweries\_raw\_withMedianABV%>% group\_by( Beer\_Category ) %>% summarise(Median\_IBU = median(IBU, na.rm=TRUE))  
  
##Join the Median\_IBU data by State, IPA/Ale category and ABV category with the data set above  
Beer\_Breweries\_raw\_withMedianABV\_IBU <- left\_join(Beer\_Breweries\_raw\_withMedianABV,Median\_IBU\_by\_St\_IPAorAle\_ABV\_CAT, by = c("Beer\_Category") )  
  
# Create a new column IBU\_New containing IBU values and replacing missing values by the above calculated median IBU  
Beer\_Breweries\_raw\_withMedianABV\_IBU$IBU\_New <- ifelse(is.na(Beer\_Breweries\_raw\_withMedianABV\_IBU$IBU), Beer\_Breweries\_raw\_withMedianABV\_IBU$Median\_IBU, Beer\_Breweries\_raw\_withMedianABV$IBU)  
  
#Filter out the missing values and remove the redundant columns  
Beer\_Breweries\_Clean <- filter(Beer\_Breweries\_raw\_withMedianABV\_IBU, !is.na(IBU\_New)) %>% select( -Median\_ABV, -Median\_IBU)  
  
# Step 5 Check the cleansed data to see if we still have any missing data. Look for ABV\_New and IBU\_New  
as.data.frame(lapply(Beer\_Breweries\_Clean,function(x) { length(which(is.na(x)))}))

## Beer\_Name Beer\_Id ABV IBU Brewery\_Id Beer\_Style Ounces Brewery\_Name  
## 1 0 0 62 1005 0 0 0 0  
## City State Beer\_Category ABV\_New IBU\_New  
## 1 0 0 0 0 0

message("As we do not have any missing values in the columns ABV\_New and IBU\_new, Beer\_Breweries\_Clean is the dataset we'll be doing further analysis on using the columns ABV\_New and IBU\_New")

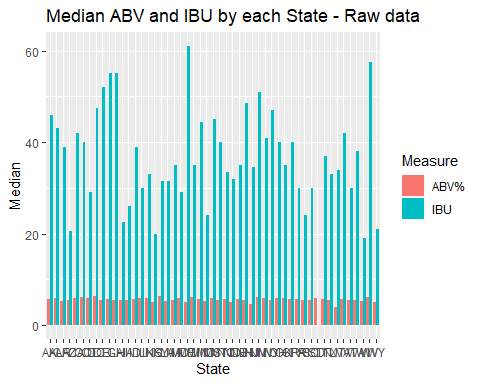
## As we do not have any missing values in the columns ABV\_New and IBU\_new, Beer\_Breweries\_Clean is the dataset we'll be doing further analysis on using the columns ABV\_New and IBU\_New

#Question 4.Compute the median alcohol content and international bitterness unit for each state. Plot a bar chart to compare.  
#Analysis on Cleansed data  
#Step 1 Calculate the median ABV by State  
#Step 1.1 Group by State and Summarise  
Answer4\_Median\_ABV <- Beer\_Breweries\_Clean %>% group\_by(State) %>%   
 summarise(Median = median(ABV\_New, na.rm = TRUE))  
  
#Step 1.2 Convert ABV to percentage (multiple by 100) for a more clear plot  
Answer4\_Median\_ABV$Median = Answer4\_Median\_ABV$Median \* 100  
  
#Step 1.3 Create a dataframe with columns Median (calculated above) and a column 'Measure' with it's value as 'ABV%'  
Answer4\_Median\_ABV$Measure = "ABV%"  
  
#Step 2 Calculate the median IBU by State  
#Step 2.1 Group by State and Summarise  
Answer4\_Median\_IBU <- Beer\_Breweries\_Clean %>% group\_by(State) %>%   
 summarise(Median = median(IBU\_New, na.rm = TRUE))  
  
#Step 2.2 Create a dataframe with columns Median (calculated above) and a column 'Measure' wit it's value as 'IBU'  
Answer4\_Median\_IBU$Measure = "IBU"  
  
#Step 3 Combine the baove created data frame to have Median ABV% abd IBU in one data frame  
Answer4 <- rbind(Answer4\_Median\_ABV,Answer4\_Median\_IBU )  
  
#Step 4 Plot the bar chart  
ggplot(Answer4) + geom\_bar(mapping= aes(x=State, y=Median, fill = Measure), stat= "identity", position = "dodge") + ggtitle("Median ABV and IBU by each State")



#Analysis on Raw data  
#Step 1 Calculate the median ABV by State  
#Step 1.1 Group by State and Summarise  
Answer4\_Median\_ABV <- Beer\_Breweries\_Clean %>% group\_by(State) %>%   
 summarise(Median = median(ABV, na.rm = TRUE))  
#Step 1.2 Convert ABV to percentage (multiple by 100) for a more clear plot  
Answer4\_Median\_ABV$Median = Answer4\_Median\_ABV$Median \* 100  
#Step 1.3 Create a dataframe with columns Median (calculated above) and a column 'Measure' wit it's value as 'ABV%'  
Answer4\_Median\_ABV$Measure = "ABV%"  
  
#Step 2 Calculate the median IBU by State  
#Step 2.1 Group by State and Summarise  
Answer4\_Median\_IBU <- Beer\_Breweries\_Clean %>% group\_by(State) %>%   
 summarise(Median = median(IBU, na.rm = TRUE))  
  
#Step 2.2 Create a dataframe with columns Median (calculated above) and a column 'Measure' wit it's value as 'IBU'  
Answer4\_Median\_IBU$Measure = "IBU"  
  
#Step 3 Combine the baove created data frame to have Median ABV% abd IBU in one data frame  
Answer4 <- rbind(Answer4\_Median\_ABV,Answer4\_Median\_IBU )  
  
#Step 4 Plot the bar chart  
ggplot(Answer4) + geom\_bar(mapping= aes(x=State, y=Median, fill = Measure), stat= "identity", position = "dodge") + ggtitle("Median ABV and IBU by each State - Raw data")

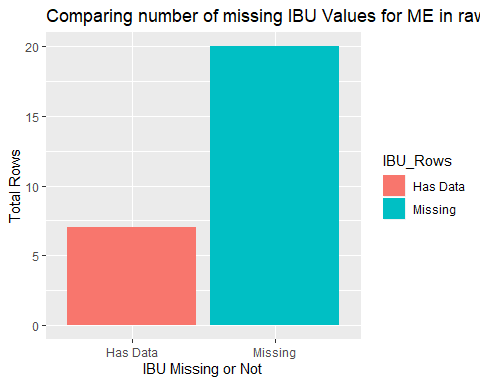
## Warning: Removed 1 rows containing missing values (geom\_bar).



message("As we see that in Cleansed data, WV has the highest Median IBU, whereas in the raw data ME has the highest median IBU. Let's deep dive into ME data")

## As we see that in Cleansed data, WV has the highest Median IBU, whereas in the raw data ME has the highest median IBU. Let's deep dive into ME data

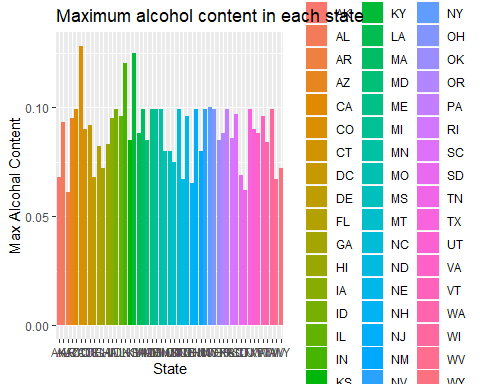
#Select only the ME data from the entire data set  
Beer\_Breweries\_Clean\_ME <- filter(Beer\_Breweries\_Clean, State=='ME')   
#Check which rows have missing IBU  
Beer\_Breweries\_Clean\_ME$IBU\_Rows <- ifelse(is.na(Beer\_Breweries\_Clean\_ME$IBU), "Missing", "Has Data")  
  
#Plot a gar char of the missing data  
ggplot(Beer\_Breweries\_Clean\_ME) + geom\_bar(mapping= aes( x=IBU\_Rows, fill=IBU\_Rows)) + ggtitle("Comparing number of missing IBU Values for ME in raw data") +  
 xlab("IBU Missing or Not") + ylab("Total Rows")



message(" We see that out of 27 rows for ME data, 20 have missing data in the raw data set. Thus the cleansed data is more reliable")

## We see that out of 27 rows for ME data, 20 have missing data in the raw data set. Thus the cleansed data is more reliable

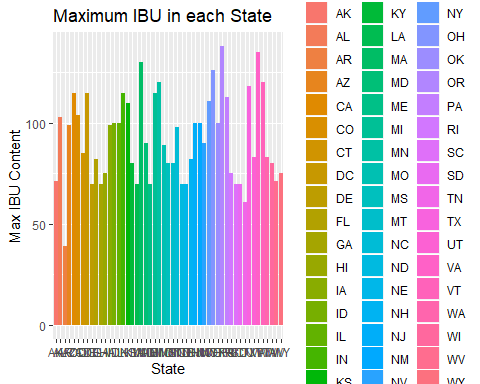
#Question 5. Which state has the maximum alcoholic (ABV) beer? Which state has the most bitter (IBU) beer?  
#Solution  
#Max ABV  
#Calculate max ABV by each state  
max\_alc\_content <-Beer\_Breweries\_Clean %>% group\_by(State) %>% summarise(Max\_Alcohol\_Content = max(ABV\_New, na.rm = TRUE))  
#Plot the data  
ggplot(max\_alc\_content) + geom\_bar(mapping= aes(x= State,y= Max\_Alcohol\_Content, fill= State), stat= "identity") + ggtitle("Maximum alcohol content in each state") + ylab("Max Alcohal Content")



message(" State with Maximum alcoholic beer is CO")

## State with Maximum alcoholic beer is CO

#Max IBU: Most bitter beer  
#Calculate max IBU by each state  
max\_IBU\_content <-Beer\_Breweries\_Clean %>% group\_by(State) %>% summarise(Max\_IBU = max(IBU\_New, na.rm = TRUE))  
#Plot the data  
ggplot(max\_IBU\_content) + geom\_bar(mapping= aes(x= State,y= Max\_IBU, fill= State), stat= "identity") + ggtitle("Maximum IBU in each State") + ylab("Max IBU Content")



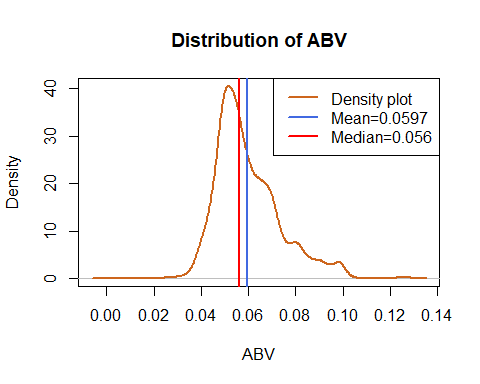
message(" State with most bitter beer is OR")

## State with most bitter beer is OR

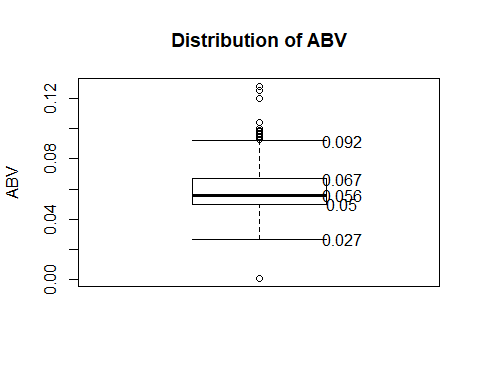
#Question 6. Comment on the summary statistics and distribution of the ABV variable.  
#Run a summary on the cleansed ABV column   
summary(Beer\_Breweries\_Clean$ABV\_New)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.0010 0.0500 0.0560 0.0597 0.0670 0.1280

#ABV Density Plot  
d <- density(Beer\_Breweries\_Clean$ABV\_New)  
plot(d, main = "Distribution of ABV" , xlab="ABV" )  
  
lines(density(Beer\_Breweries\_Clean$ABV\_New), # density plot  
 lwd = 2, # thickness of line  
 col = "chocolate3")  
abline(v = mean(Beer\_Breweries\_Clean$ABV\_New),  
 col = "royalblue",  
 lwd = 2)  
abline(v = median(Beer\_Breweries\_Clean$ABV\_New),  
 col = "red",  
 lwd = 2)  
legend(x = "topright", # location of legend within plot area  
 c("Density plot", "Mean=0.0597", "Median=0.056"),  
 col = c("chocolate3", "royalblue", "red"),  
 lwd = c(2, 2, 2))

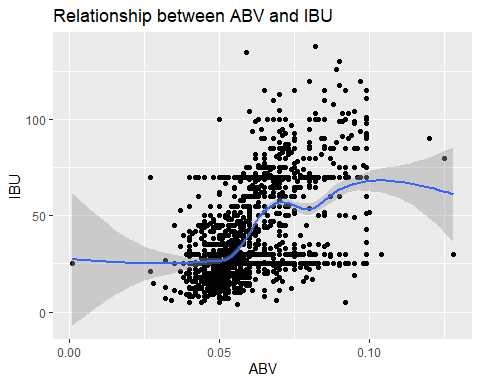


#ABV box plot  
boxplot(Beer\_Breweries\_Clean$ABV, staplewex = 1, main = "Distribution of ABV", ylab = "ABV")  
text(y = boxplot.stats(Beer\_Breweries\_Clean$ABV)$stats, labels = boxplot.stats(Beer\_Breweries\_Clean$ABV)$stats, x = 1.25)



#Question# 7  
#Create a Scatter plot of the data by ABV vs. IBU  
ggplot( data = Beer\_Breweries\_Clean) + geom\_point(mapping = aes(x = ABV\_New, y = IBU\_New)) +   
 geom\_smooth(mapping = aes(x = ABV\_New, y = IBU\_New)) + ggtitle("Relationship between ABV and IBU") +  
 xlab("ABV") + ylab("IBU")

## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



#Also find the correlation factor between ABV and IBU  
library("ggpubr")

## Warning: package 'ggpubr' was built under R version 3.6.2

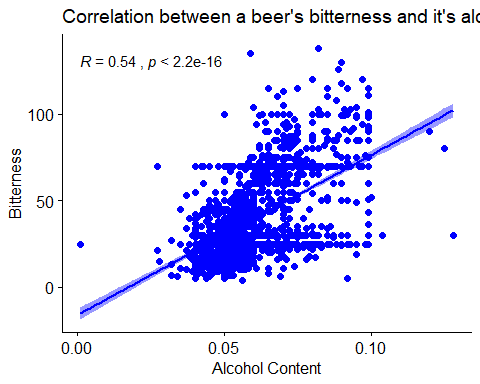
## Loading required package: magrittr

##   
## Attaching package: 'magrittr'

## The following object is masked from 'package:purrr':  
##   
## set\_names

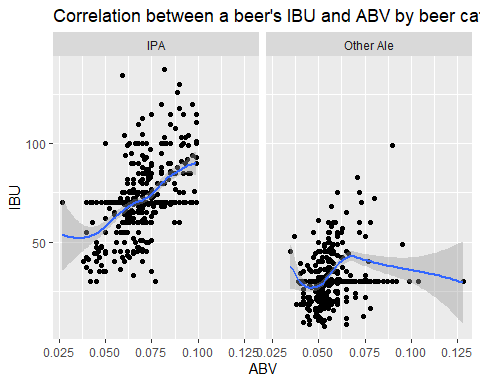
## The following object is masked from 'package:tidyr':  
##   
## extract

ggscatter(Beer\_Breweries\_Clean, x = "ABV\_New", y = "IBU\_New",   
 add = "reg.line", conf.int = TRUE,   
 cor.coef = TRUE, cor.method = "pearson",  
 xlab = "Alcohol Content", ylab = "Bitterness", color="blue") +   
 ggtitle("Correlation between a beer's bitterness and it's alcoholic content")



#Question 8: Find the correlation between ABV and IBU and IPA vs. Other Ales  
#Filter the data to include only IPA and Other Ales  
Beer\_Breweries\_Clean\_8 <- Beer\_Breweries\_Clean %>% filter(Beer\_Category %in% c("IPA", "Other Ale"))   
#Plot the data  
ggplot( data = Beer\_Breweries\_Clean\_8) + geom\_point(mapping = aes(x = ABV\_New, y = IBU\_New)) +   
 geom\_smooth(mapping = aes(x = ABV\_New, y = IBU\_New)) + xlab("ABV") + ylab("IBU") +  
 facet\_wrap(~Beer\_Category) + ggtitle("Correlation between a beer's IBU and ABV by beer category -IPA vs. Other Ales")

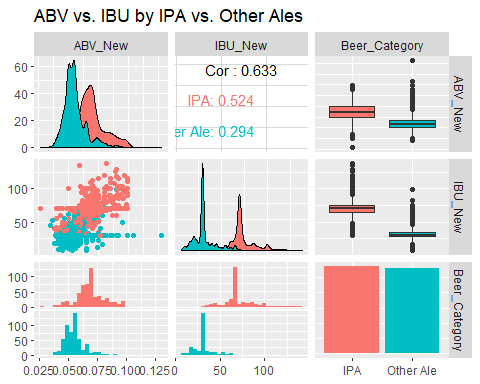
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



#Additional comparision of ABV and IBU by beer categories using ggpairs  
Beer\_Breweries\_Clean\_8 %>% select(ABV\_New, IBU\_New,Beer\_Category) %>%   
 ggpairs(aes(color=Beer\_Category) ,title="ABV vs. IBU by IPA vs. Other Ales")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

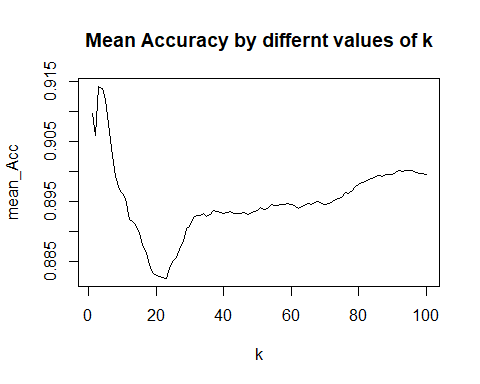
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



message("We can see from the plot that the IPAs have a higher median ABV and IBU compared to Other Ales. Most of the beers with higher ABV and IBU fall in IPA beer Category, whereas the Other Ales have comparatively lower ABV and IBU.")

## We can see from the plot that the IPAs have a higher median ABV and IBU compared to Other Ales. Most of the beers with higher ABV and IBU fall in IPA beer Category, whereas the Other Ales have comparatively lower ABV and IBU.

#Build KNN data model to analyze Beer Categories  
#Filter the data to limit it to IPA and Other Ale beer categories, and also select only the columns required for analysis  
Beer\_Breweries\_8\_knn\_data <- Beer\_Breweries\_Clean %>% filter(Beer\_Category %in% c("IPA", "Other Ale")) %>% select(Beer\_Id,Beer\_Name, ABV\_New,IBU\_New, Beer\_Category)  
  
#Create a placeholder for Accuracy  
Acc= matrix(nrow=100, ncol=100)  
  
#For loop to create 100 different training and test data sets  
for(seed in 1:100) {  
 set.seed(seed)  
 trainIndices = sample(seq(1:length(Beer\_Breweries\_8\_knn\_data$Beer\_Category)), round(.7\*length(Beer\_Breweries\_8\_knn\_data$Beer\_Category)))  
 trainBew\_Brewries = Beer\_Breweries\_8\_knn\_data[trainIndices,]  
 testBew\_Brewries = Beer\_Breweries\_8\_knn\_data[-trainIndices,]  
#For loop to run KNN model for k=1-100   
 for( i in 1:100) {  
 knn1 <- knn(trainBew\_Brewries[, c(3,4)],testBew\_Brewries[, c(3,4)], trainBew\_Brewries$Beer\_Category, k = i )  
 t\_knn <- table(knn1,testBew\_Brewries$Beer\_Category )  
 #Create confusion matrix  
 Cm\_Knn <- confusionMatrix(t\_knn)  
 #Captures Accuracy  
 Acc[seed,i] = Cm\_Knn$overall[1]  
   
 }   
   
}  
#Caluclate mean accuracy for each of the seed/knn iteritions  
mean\_Acc = colMeans(Acc)  
#Plot the data by knnvalue and accuracy  
plot(seq(1,100,1),mean\_Acc, type = "l", xlab ='k', main = "Mean Accuracy by differnt values of k" )



#See where we had the max accuracy  
which.max(mean\_Acc)

## [1] 3

#Print the max accuracy of the model  
max(mean\_Acc)

## [1] 0.9141298

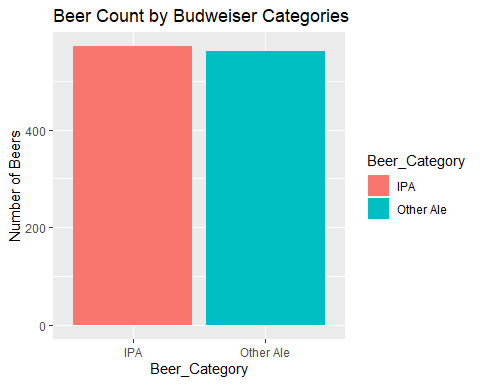
message("By running KNN for different samples and different values of K, we can see the knn model accuracy is highest at k=3. So performing internal validation at KNN at k=3")

## By running KNN for different samples and different values of K, we can see the knn model accuracy is highest at k=3. So performing internal validation at KNN at k=3

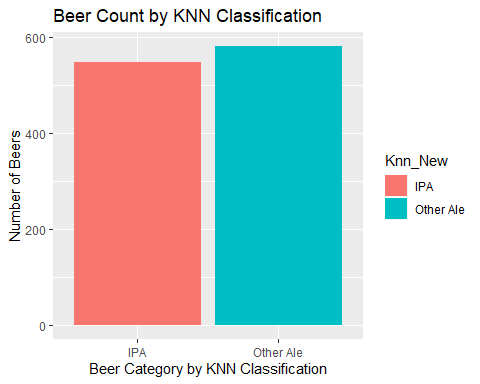
#Perform internal validation with knn with knn=3  
  
knn3 <- knn.cv(Beer\_Breweries\_8\_knn\_data[, c(3,4)],Beer\_Breweries\_8\_knn\_data$Beer\_Category, k=3 )  
  
t\_knn <- table(knn3,Beer\_Breweries\_8\_knn\_data$Beer\_Category )  
  
Cm\_Knn <- confusionMatrix(t\_knn)  
  
#Train the model with knn analysis  
model\_knn <- train(Beer\_Breweries\_8\_knn\_data[, c(3,4)], Beer\_Breweries\_8\_knn\_data$Beer\_Category, method='knn')  
  
#Predict the beer categories using the model  
Beer\_Breweries\_8\_knn\_data$Knn\_New <-as.factor(predict(model\_knn,Beer\_Breweries\_8\_knn\_data[, c(3,4)]))  
  
#Question 9: Check for misclassification of beers  
  
#Compare the Beer Category assignment by knn Model vs. the provided Budweiser data  
Mismatched\_data <- filter(Beer\_Breweries\_8\_knn\_data,Beer\_Breweries\_8\_knn\_data$Knn\_New != Beer\_Breweries\_8\_knn\_data$Beer\_Category)  
  
#No of rows of mismatched data  
nrow(Mismatched\_data)

## [1] 82

#Plot the total mismatched data  
#Plor beer categories by Budweiser asssignment  
ggplot(Beer\_Breweries\_8\_knn\_data) + geom\_bar(mapping = aes(x = Beer\_Category, fill= Beer\_Category)) +  
ylab("Number of Beers") + ggtitle("Beer Count by Budweiser Categories")



#Plor beer categories by KNN asssignment  
ggplot(Beer\_Breweries\_8\_knn\_data) + geom\_bar(mapping = aes(x = Knn\_New, fill= Knn\_New))+  
 ylab("Number of Beers") + xlab("Beer Category by KNN Classification")+ ggtitle("Beer Count by KNN Classification")



#Find the number of beer misclassified as IPA  
nrow(filter(Mismatched\_data, Beer\_Category == "IPA" ))

## [1] 52

#Find the number of beer misclassified as Other Ales  
nrow(filter(Mismatched\_data, Beer\_Category != "IPA" ))

## [1] 30

message("We can see that around 23 beers were misclassified as 'IPA' and 51 were misclassified as 'Other Ale' ")

## We can see that around 23 beers were misclassified as 'IPA' and 51 were misclassified as 'Other Ale'

#Print a few samples of the mismatched data  
#Top 6  
head(Beer\_Breweries\_8\_knn\_data)

## Beer\_Id Beer\_Name ABV\_New IBU\_New Beer\_Category Knn\_New  
## 1 2264 Rise of the Phoenix 0.071 70 IPA IPA  
## 2 2263 Sinister 0.090 70 IPA IPA  
## 3 2262 Sex and Candy 0.075 70 IPA IPA  
## 4 2131 Cone Crusher 0.086 70 IPA IPA  
## 5 1980 Troll Destroyer 0.085 70 IPA IPA  
## 6 799 21st Amendment IPA (2006) 0.070 70 IPA IPA

#Bottm 6  
tail(Beer\_Breweries\_8\_knn\_data)

## Beer\_Id Beer\_Name ABV\_New IBU\_New Beer\_Category  
## 1126 1316 Colorojo Imperial Red Ale 0.082 30 Other Ale  
## 1127 1045 Wynkoop Pumpkin Ale 0.055 30 Other Ale  
## 1128 928 Belgorado 0.067 45 IPA  
## 1129 807 Rail Yard Ale 0.052 30 Other Ale  
## 1130 145 Silverback Pale Ale 0.055 40 Other Ale  
## 1131 84 Rail Yard Ale (2009) 0.052 30 Other Ale  
## Knn\_New  
## 1126 Other Ale  
## 1127 Other Ale  
## 1128 Other Ale  
## 1129 Other Ale  
## 1130 Other Ale  
## 1131 Other Ale

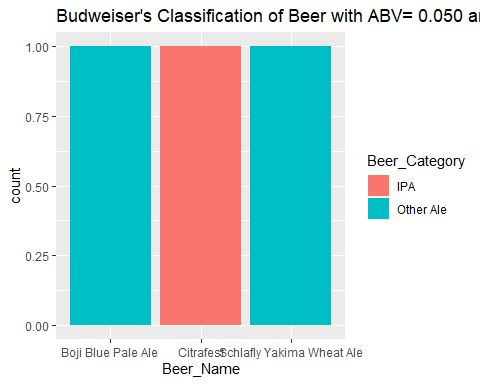
#Run a summary of the data set  
summary(Beer\_Breweries\_8\_knn\_data)

## Beer\_Id Beer\_Name ABV\_New   
## Min. : 1 Dale's Pale Ale : 6 Min. :0.0270   
## 1st Qu.: 709 Dagger Falls IPA : 3 1st Qu.:0.0520   
## Median :1318 312 Urban Pale Ale : 2 Median :0.0600   
## Mean :1316 312 Urban Wheat Ale : 2 Mean :0.0622   
## 3rd Qu.:1886 Bitter Biker Double IPA: 2 3rd Qu.:0.0700   
## Max. :2692 Citra Ass Down : 2 Max. :0.1280   
## (Other) :1114   
## IBU\_New Beer\_Category Knn\_New   
## Min. : 7.00 Length:1131 IPA :549   
## 1st Qu.: 30.00 Class :character Other Ale:582   
## Median : 52.00 Mode :character   
## Mean : 52.01   
## 3rd Qu.: 70.00   
## Max. :138.00   
##

Beer\_Breweries\_8\_knn\_data$Beer\_Category<- as.factor(Beer\_Breweries\_8\_knn\_data$Beer\_Category)  
#colnames(Beer\_Breweries\_8\_knn\_data$Knn\_New.predict(model\_knn, Beer\_Breweries\_8\_knn\_data[, c(3, 4)])) <- c("Beer\_Category\_knn")  
  
#Check Budweisers classification of beer from mismatched data  
filter(Beer\_Breweries\_Clean, Beer\_Id %in% c(2602, 2105, 1021, 1930)) %>% select(Beer\_Id,Beer\_Name,ABV\_New,IBU\_New,Beer\_Style)

## Beer\_Id Beer\_Name ABV\_New IBU\_New Beer\_Style  
## 1 2602 Citrafest 0.050 45 American IPA  
## 2 2105 Even Keel 0.038 40 American IPA  
## 3 1930 Summer Session Ale 0.050 61 American Pale Wheat Ale  
## 4 1021 Watership Brown Ale 0.072 55 American Brown Ale

#Check Beer categories for beers with ABV = 0.050 & IBU\_New = 45  
invaid\_class <- filter(Beer\_Breweries\_Clean, ABV\_New == 0.050 & IBU\_New == 45 )%>% select(Beer\_Id,Beer\_Name,ABV\_New,IBU\_New,Beer\_Style,Beer\_Category)  
  
#Plot the Budweised classification for beers with ABV = 0.050 & IBU\_New = 45   
ggplot(invaid\_class) + geom\_bar(mapping= aes(x= Beer\_Name, fill=Beer\_Category)) +   
 ggtitle("Budweiser's Classification of Beer with ABV= 0.050 and IBU = 45")



message("We see that 2 of the 3 beers with ABV=0.050 and IBU=45 were classified at Other Ales, but one was classified as IPA")

## We see that 2 of the 3 beers with ABV=0.050 and IBU=45 were classified at Other Ales, but one was classified as IPA

nrow(Mismatched\_data)

## [1] 82

message("In total there are 82 beers, out of 2410, which, according to the KNN model, seem to have been incorrectly categoried as IPA or Other Ales by Budweiser")

## In total there are 82 beers, out of 2410, which, according to the KNN model, seem to have been incorrectly categoried as IPA or Other Ales by Budweiser

message("Take away from the analysis:   
Conflicting ‘Beer Style’ and ‘Expected Taste’ can affect the sale of the beer, so we need to correct the classifications.  
82 beers are candidates for re-classification for Beer Style based on their Alcohol content and bitterness levels. The list is included below.")

## Take away from the analysis:   
## Conflicting ‘Beer Style’ and ‘Expected Taste’ can affect the sale of the beer, so we need to correct the classifications.  
## 82 beers are candidates for re-classification for Beer Style based on their Alcohol content and bitterness levels. The list is included below.

#Renaming columns for better clarity  
colnames(Mismatched\_data) <- c("Beer Id", "Beer Name","ABV","IBU","Budweiser\_Beer\_Category", "KNN\_Predicted\_Beer\_Category")  
  
#List of 82 beers  
print.data.frame(Mismatched\_data)

## Beer Id Beer Name ABV IBU  
## 1 2602 Citrafest 0.050 45  
## 2 1875 Naked Pig Pale Ale 0.060 43  
## 3 2105 Even Keel 0.038 40  
## 4 1900 Lost Sailor IPA 0.055 40  
## 5 1456 Powder Hound Winter Ale 0.072 60  
## 6 1898 Blackmarket Rye IPA 0.075 35  
## 7 1896 Aftermath Pale Ale 0.058 44  
## 8 1850 American India Red Ale 0.071 83  
## 9 1547 Full Nelson Pale Ale 0.059 60  
## 10 119 Full Nelson Pale Ale (2010) 0.059 60  
## 11 692 Blue Point White IPA 0.060 40  
## 12 1891 Longhop IPA 0.042 30  
## 13 1629 Rebel IPA 0.065 45  
## 14 28 Tule Duck Red Ale (Current) 0.062 42  
## 15 1954 Ghost Ship White IPA 0.056 55  
## 16 1439 Boat Beer 0.042 35  
## 17 1247 Ray Ray’s Pale Ale 0.052 42  
## 18 2380 IPA #11 0.057 58  
## 19 569 Tocobaga Red Ale 0.072 75  
## 20 1393 Cascadian Dark Ale 0.060 75  
## 21 1406 Savannah Brown Ale 0.062 55  
## 22 2029 Tropicalia 0.065 65  
## 23 2292 Little Sister India Style Session Ale 0.043 60  
## 24 935 Fascist Pig Ale 0.080 72  
## 25 1223 All Day IPA 0.047 42  
## 26 672 Hop A Potamus Double Dark Rye Pale Ale 0.090 99  
## 27 2314 Wild Trail Pale Ale 0.057 44  
## 28 1505 Flying Jenny Extra Pale Ale 0.060 54  
## 29 693 Flying Jenny Extra Pale Ale (2012) 0.060 54  
## 30 2007 Norm's Gateway IPA 0.040 55  
## 31 2026 Pursuit 0.070 40  
## 32 1653 The Long Thaw White IPA 0.062 45  
## 33 1833 Hop Farm IPA 0.058 45  
## 34 1605 Festeroo Winter Ale 0.078 60  
## 35 2193 Abominable Winter Ale 0.073 70  
## 36 916 Abominable Winter Ale (2012) 0.073 70  
## 37 759 Double Haul IPA (2009) 0.065 65  
## 38 758 Double Haul IPA (2006) 0.065 65  
## 39 86 Double Haul IPA 0.065 65  
## 40 1396 Laughing Dog IPA 0.064 66  
## 41 2329 Peacemaker Pale Ale 0.057 47  
## 42 2176 Mauna Kea Pale Ale 0.054 42  
## 43 1460 El Conquistador Extra Pale Ale 0.048 44  
## 44 1148 Nebraska India Pale Ale 0.065 65  
## 45 2475 Slow Ride 0.045 40  
## 46 2692 Get Together 0.045 50  
## 47 1226 Three Skulls Ale Pale Ale 0.063 42  
## 48 2302 Pinner Throwback IPA 0.049 35  
## 49 1908 Fresh Slice White IPA 0.055 45  
## 50 1930 Summer Session Ale 0.050 61  
## 51 2630 98 Problems (Cuz A Hop Ain't One) 0.065 65  
## 52 2628 Grapefruit IPA 0.050 35  
## 53 711 Over the Rail Pale Ale 0.057 68  
## 54 1021 Watership Brown Ale 0.072 55  
## 55 2515 Pump House IPA 0.055 45  
## 56 945 Long Hammer IPA 0.065 44  
## 57 583 Long Hammer IPA 0.065 44  
## 58 1771 Lil SIPA 0.050 55  
## 59 1078 Schlafly IPA 0.045 30  
## 60 1386 Righteous Ale 0.063 57  
## 61 423 Righteous Ale (2011) 0.063 57  
## 62 1606 Snow King Pale Ale 0.060 55  
## 63 1971 Texas Pale Ale (TPA) 0.055 40  
## 64 2486 Baby Daddy Session IPA 0.047 35  
## 65 90 Third Eye Pale Ale 0.065 65  
## 66 1749 Just IPA 0.046 45  
## 67 1912 40 Mile IPA 0.060 50  
## 68 951 Alloy 0.058 36  
## 69 433 Greenville Pale Ale 0.055 52  
## 70 2235 Day Hike Session 0.041 41  
## 71 1661 Trailhead ISA 0.048 48  
## 72 482 Trailhead India Style Session Ale (2011) 0.048 48  
## 73 1925 Trader Session IPA 0.040 42  
## 74 2190 Campside Session IPA 0.045 50  
## 75 1932 Thai Style White IPA 0.065 33  
## 76 1846 Wachusett Light IPA 0.040 37  
## 77 1844 Wachusett IPA 0.056 50  
## 78 1029 Wachusett Light IPA (2013) 0.040 37  
## 79 1550 Charlie in the Rye 0.058 55  
## 80 2146 #004 Session I.P.A. 0.048 38  
## 81 1513 Lights Out Vanilla Cream Extra Stout 0.077 30  
## 82 928 Belgorado 0.067 45  
## Budweiser\_Beer\_Category KNN\_Predicted\_Beer\_Category  
## 1 IPA Other Ale  
## 2 Other Ale IPA  
## 3 IPA Other Ale  
## 4 IPA Other Ale  
## 5 Other Ale IPA  
## 6 IPA Other Ale  
## 7 Other Ale IPA  
## 8 Other Ale IPA  
## 9 Other Ale IPA  
## 10 Other Ale IPA  
## 11 IPA Other Ale  
## 12 IPA Other Ale  
## 13 IPA Other Ale  
## 14 Other Ale IPA  
## 15 IPA Other Ale  
## 16 IPA Other Ale  
## 17 Other Ale IPA  
## 18 IPA Other Ale  
## 19 Other Ale IPA  
## 20 Other Ale IPA  
## 21 Other Ale IPA  
## 22 IPA Other Ale  
## 23 IPA Other Ale  
## 24 Other Ale IPA  
## 25 IPA Other Ale  
## 26 Other Ale IPA  
## 27 Other Ale IPA  
## 28 Other Ale IPA  
## 29 Other Ale IPA  
## 30 IPA Other Ale  
## 31 IPA Other Ale  
## 32 IPA Other Ale  
## 33 IPA Other Ale  
## 34 Other Ale IPA  
## 35 Other Ale IPA  
## 36 Other Ale IPA  
## 37 IPA Other Ale  
## 38 IPA Other Ale  
## 39 IPA Other Ale  
## 40 IPA Other Ale  
## 41 Other Ale IPA  
## 42 Other Ale IPA  
## 43 Other Ale IPA  
## 44 IPA Other Ale  
## 45 IPA Other Ale  
## 46 IPA Other Ale  
## 47 Other Ale IPA  
## 48 IPA Other Ale  
## 49 IPA Other Ale  
## 50 Other Ale IPA  
## 51 IPA Other Ale  
## 52 IPA Other Ale  
## 53 Other Ale IPA  
## 54 Other Ale IPA  
## 55 IPA Other Ale  
## 56 IPA Other Ale  
## 57 IPA Other Ale  
## 58 IPA Other Ale  
## 59 IPA Other Ale  
## 60 Other Ale IPA  
## 61 Other Ale IPA  
## 62 Other Ale IPA  
## 63 IPA Other Ale  
## 64 IPA Other Ale  
## 65 IPA Other Ale  
## 66 IPA Other Ale  
## 67 IPA Other Ale  
## 68 IPA Other Ale  
## 69 Other Ale IPA  
## 70 IPA Other Ale  
## 71 IPA Other Ale  
## 72 IPA Other Ale  
## 73 IPA Other Ale  
## 74 IPA Other Ale  
## 75 IPA Other Ale  
## 76 IPA Other Ale  
## 77 IPA Other Ale  
## 78 IPA Other Ale  
## 79 IPA Other Ale  
## 80 IPA Other Ale  
## 81 IPA Other Ale  
## 82 IPA Other Ale

## Take away from the analysis

Conflicting ‘Beer Style’ and ‘Expected Taste’ can affect the sale of the beer, so we need to correct the classifications.

82 beers are candidates for re-classification for Beer Style based on their Alcohol content and bitterness levels.