Unit 8 Presentation

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?

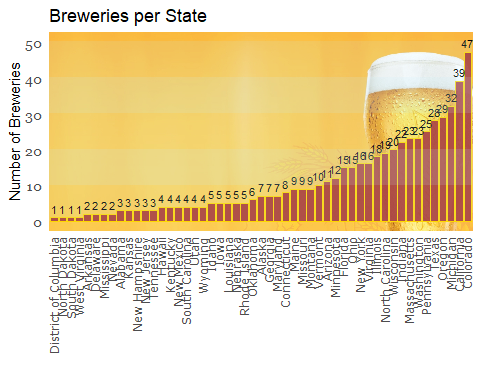
beers <- read.csv("Beers.csv")  
breweries <- read.csv("Breweries.csv")  
  
# Question 2  
# Merge beer data with the breweries data.   
# Print the first 6 observations and the last six observations to check the merged file.  
  
# Merge the two data frames into one  
data <- merge(beers,breweries,by.x = "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) # First six from merged set

## 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) # Last six from merged set

## 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() # first time on a machine only. Watch the "Console" window for a confirmation dialog  
  
# 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) # convert factors to characters  
brewery\_count\_by\_state$State = str\_trim(brewery\_count\_by\_state$State) # remove leading space from string  
brewery\_count\_by\_state = brewery\_count\_by\_state %>% mutate(state=State) # Capitolize State  
brewery\_count\_by\_state = brewery\_count\_by\_state[,3:2] # remove the old, non-capitolized state  
  
color\_boxes <- data.frame(y1=seq(0,40,10), # this sets up an array of x and y coords, and colors to  
 y2=seq(10,50,10), # zebra stripe the background later  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 5))  
  
# turn the states back into factors  
brewery\_count\_by\_state$state <- factor(brewery\_count\_by\_state$state, levels = brewery\_count\_by\_state$state[order(brewery\_count\_by\_state$count)])  
  
# read in the photo background  
image <- readJPEG("beer\_glass\_background.jpg")  
# scale the background for this chart  
bg <- rasterGrob(image, height = unit(1.9, "npc"), x=0.5, y=0.2)  
  
brewery\_count\_by\_state %>% # send in the brewery data  
 mutate(fullname = abbr2state(state)) %>% # change the state abbr to full name  
 ggplot() + # send the data in ascending order  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # add the background image  
 geom\_bar(aes(x=reorder(fullname, count), y = count),   
 stat="identity", fill = "brown", color = "gold", alpha = 0.9) + # plot the actual bars  
 geom\_text(aes(x=reorder(fullname, count), y = count,label = count),   
 vjust = -0.5, size = 3) + # draw the count up above the bars  
 geom\_rect(data=color\_boxes, aes(ymin = y1, ymax = y2,   
 xmin = -Inf, xmax = +Inf), fill = color\_boxes$color) + # zebra stripe the background  
 theme\_minimal() + # clear out the theme  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10), # set text attributes   
 axis.text.x = element\_text(family = "Verdana", angle=90, hjust = 1, vjust = .4)) +  
 labs(title = "Breweries per State", x="", y="Number of Breweries") # set legend

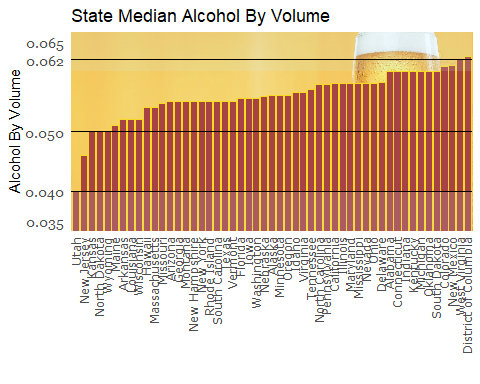


To perform the rest of the analysis, the data was checked for a number of inconsistencies including missing values. Over half of the records(1405 of 2410) in the dataset did not have values for the IBU, and 62 of the entries did not have a value for ABV.

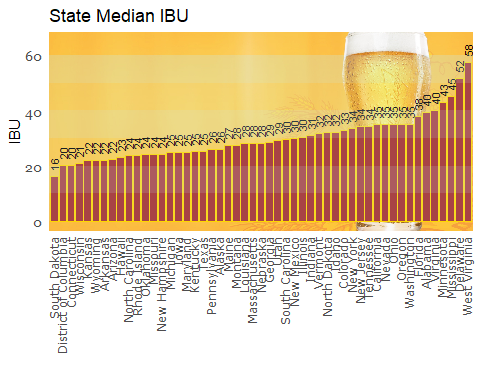
To allow us to continue without removing any of the records for missing values, it was decided to impute the data with a k-NN on beers in the same “Style”. What this means is that for a given beer, the algorithm will look at the collection of beers, find ones that match with it best, determine an appropriate 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  
# this method uses the available information in the remainder of the data frame to establish a kNN   
# relationship and fill in missing values  
adjData <- data %>% knnImputation()

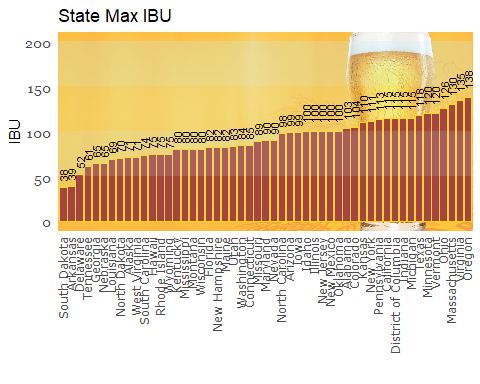
# read in the photo background  
image <- readJPEG("beer\_glass\_background.jpg")  
# scale the background for this chart  
bg <- rasterGrob(image, height = unit(1.5, "npc"), x=0.5, y=0.5)  
  
# create the alternating color background  
color\_boxes <- data.frame(y1=seq(0.00, .06, .02),   
 y2=seq(0.02, .08, .02),  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 4))  
  
adjData %>%   
 group\_by(State) %>%   
 summarize(medabv=median(ABV)) %>% # create a column named medabv from the mediab 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), # plot the actual data  
 stat="identity", color="gold", fill="brown") +  
 geom\_rect(data=color\_boxes, # zebra stripe the background  
 aes(ymin = y1, ymax = y2, xmin = -Inf, xmax = +Inf),  
 fill = color\_boxes$color) +  
 scale\_y\_continuous(breaks = sort(c(seq(0.035, 0.065, length.out = 3 ), # setup the y axis scale, and add 0.62  
 0.062, 0.04))) + # and 0.04 in it for reference  
 geom\_hline(yintercept = 0.062) + # highlight the max at 0.062 abv  
 geom\_hline(yintercept = 0.050) + # highlight the med at 0.050 abv  
 geom\_hline(yintercept = 0.04) + # highlight the min at 0.04 abv  
 theme\_minimal() + # clear the theme  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10), # set text attributes  
 axis.text.x = element\_text(family = "Verdana", angle=90, vjust = .4, hjust = 1)) + # rotate the x axis labels 90 degress  
 labs(title = "State Median Alcohol By Volume", x="", y="Alcohol By Volume") + # set the graph labels  
 coord\_cartesian(ylim = c(0.035, 0.065))



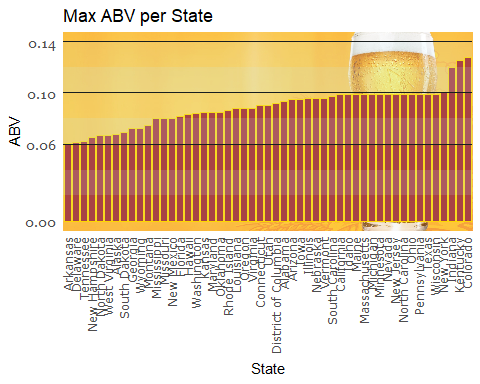
# create the alternating color background  
color\_boxes <- data.frame(y1=seq(0, 55, 10),   
 y2=seq(10, 65, 10),  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 6))  
  
  
adjData %>% group\_by(State) %>%   
 summarize(Median=median(IBU)) %>%   
 mutate(State = abbr2state(str\_trim(as.character(State)))) %>% # turn 2 letter factor into full state name  
 ggplot() +  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_bar(aes(x=reorder(State, Median), y=Median),  
 stat="identity", color="gold", fill="brown") +  
 geom\_text(aes(x=reorder(State, Median), y = Median,label = round(Median), angle = 90),   
 hjust = -.5, size = 3) +   
 geom\_rect(data=color\_boxes, # zebra stripe the background  
 aes(ymin = y1, ymax = y2, xmin = -Inf, xmax = +Inf),  
 fill = color\_boxes$color) +  
 theme\_minimal() + # clear the theme  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10), # set text attributes  
 axis.text.x = element\_text(family = "Verdana", angle=90, vjust = .4, hjust = 1)) + # rotate the x axis labels 90 degress  
 labs(title = "State Median IBU", x="", y="IBU") +  
 coord\_cartesian(ylim = c(0, 65))



# create the alternating color background  
color\_boxes <- data.frame(y1=seq(0, 150, 50),   
 y2=seq(50, 200, 50),  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 4))  
  
  
adjData %>% group\_by(State) %>%  
 summarize(Max=max(IBU)) %>%   
 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, Max), y=Max),  
 stat="identity", color="gold", fill="brown") +  
 geom\_rect(data=color\_boxes, # zebra stripe the background  
 aes(ymin = y1, ymax = y2, xmin = -Inf, xmax = +Inf),  
 fill = color\_boxes$color) +  
 geom\_text(aes(x=reorder(State, Max), y = Max,label = round(Max), angle = 90),   
 hjust = -.5, size = 3) +  
 theme\_minimal() + # clear the theme  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10), # set text attributes  
 axis.text.x = element\_text(family = "Verdana", angle=90, vjust = .4, hjust = 1)) +# rotate the x axis labels 90 degress  
 labs(title = "State Max IBU", x="", y="IBU")



# create the alternating color background  
color\_boxes <- data.frame(y1=seq(0.00, .12, .02),   
 y2=seq(0.02, .14, .02),  
 color=rep(c("#BFEFFF20","#BFEFFF40"),  
 length.out = 7))  
  
adjData %>%   
 group\_by(State) %>%   
 summarize(Max=max(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, Max), y=Max),  
 stat="identity", color="gold", fill="brown") +  
 scale\_y\_continuous(breaks = sort(c(0.00, 0.06, 0.10, 0.14))) + # setup the y axis scale, and add 0.62  
 # and 0.04 in it for reference  
 geom\_hline(yintercept = 0.14) + # highlight the max at 0.062 abv  
 geom\_hline(yintercept = 0.10) + # highlight the med at 0.050 abv   
 geom\_hline(yintercept = 0.06) + # highlight the min at 0.04 abv  
   
 geom\_rect(data=color\_boxes, # zebra stripe the background  
 aes(ymin = y1, ymax = y2, xmin = -Inf, xmax = +Inf),  
 fill = color\_boxes$color) + labs(title="Max ABV per State", y="ABV", x="State") +  
 theme\_minimal() + # clear the theme  
 theme(panel.grid.minor = element\_blank(),  
 panel.grid.major = element\_blank(),  
 axis.ticks = element\_blank(),  
 axis.text.y = element\_text(family = "Georgia", size = 10), # set text attributes  
 axis.text.x = element\_text(family = "Verdana", angle=90, vjust = .4, hjust = 1)) # rotate the x axis labels 90 degress



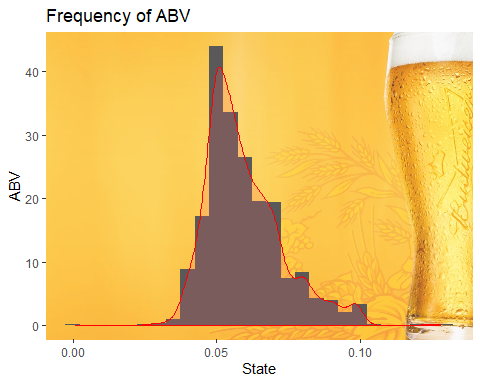
The summary statistics of the ABV by state show a minimum ABV of 0.001, or 0.1% Alcohol by Volume. This is either a data collection error or a set of Non-Alcoholic beers. The maximum ABV is 12.8%, and the mean ABV is 5.9% with 50% of the beers in the sample ranging from 5.0% to 6.7%. A case could be 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

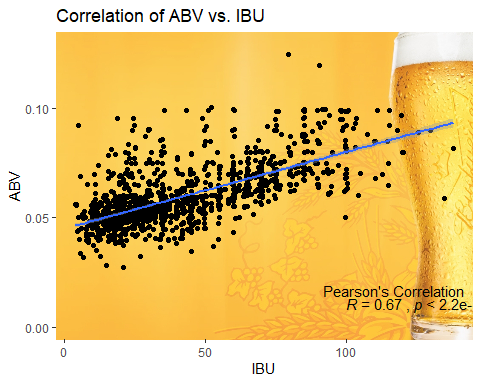
The distribution of Alcohol by Volume peaks at 5.9%, and shows a right skew, indicating that the set of beers sampled show a tendency to start at around 5% ABV, with a large grouping of beers in that 5.0%-6.7%, then a long tail reaching into higher and higher ABV numbers. Fewer beers were below the 5.0% ABV mark. This shows a trend towards making beers with higher and higher ABV, indicating a potential untapped market at lower alcohol contents.

adjData %>% filter(!is.na(ABV)) %>%   
 ggplot(aes(x=ABV)) +  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_histogram(aes(y=..density..),binwidth = 0.005, na.rm = TRUE, show.legend = TRUE) +  
 labs(title="Frequency of ABV", y="ABV", x="State") +  
 geom\_density(alpha=0.2, fill="#FF6666", color="red")

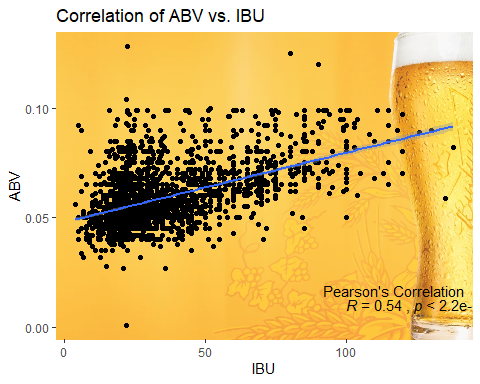


Using the original data set to plot the ABV vs. IBU with a linear model regression line, we see an “up and to the right” correlation between IBU and ABV. Pearson’s Correlation is 0.67, indication a fair amount of positive correlation.

data %>% select(c(IBU, ABV)) %>% ggplot() +  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_point(aes(x=IBU, y=ABV), position = "jitter", na.rm = TRUE) +  
 geom\_smooth(aes(x=IBU, y=ABV), method="lm", na.rm = TRUE) +  
 stat\_cor(aes(x=IBU, y=ABV),method = "pearson", label.x = 100, label.y = 0.010, na.rm = TRUE) +  
 annotate("text", x=117, y=0.017, label="Pearson's Correlation") +  
 labs(title = "Correlation of ABV vs. IBU", x="IBU", y="ABV")

 If we use the imputed dataset however, the additional data tends toward the mean and the correlation drops to 54%. Even though it lost 13% correlation points, it is still a significant positive correlation.

adjData %>% select(c(IBU, ABV)) %>% ggplot() +  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_point(aes(x=IBU, y=ABV), position = "jitter", na.rm = TRUE) +  
 geom\_smooth(aes(x=IBU, y=ABV), method="lm", na.rm = TRUE) +  
 stat\_cor(aes(x=IBU, y=ABV),method = "pearson", label.x = 100, label.y = 0.010, na.rm = TRUE) +  
 annotate("text", x=117, y=0.017, label="Pearson's Correlation") +  
 labs(title = "Correlation of ABV vs. IBU", x="IBU", y="ABV")



#install.packages("ggvis")  
library(ggvis)

## Warning: package 'ggvis' was built under R version 3.6.3

##   
## Attaching package: 'ggvis'

## The following object is masked from 'package:ggplot2':  
##   
## resolution

library(class)  
library(caret)

##   
## Attaching package: 'caret'

## The following object is masked from 'package:openintro':  
##   
## dotPlot

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

# we will use the un-imputed data for this, as the data is much more accurate  
# pull out the IPAs first  
ipas <- data %>%   
 filter(!is.na(IBU) & !is.na(ABV)) %>%   
 filter(str\_detect(Style, regex("\\bipa\\b", ignore\_case = TRUE, multiline = FALSE))) %>%   
 mutate(ipa=TRUE)  
  
# then the "other" ales  
otherAles <- data %>%   
 filter(!is.na(IBU) & !is.na(ABV)) %>%   
 filter(str\_detect(Style, regex("\\bale\\b", ignore\_case = TRUE, multiline = FALSE))) %>%   
 filter(!str\_detect(Style, regex("\\bipa\\b", ignore\_case = TRUE, multiline = FALSE))) %>%   
 mutate(ipa=FALSE)  
  
allAles <- rbind(ipas, otherAles)

if(!require(ggforce)) {install.packages("ggforce")}

## Loading required package: ggforce

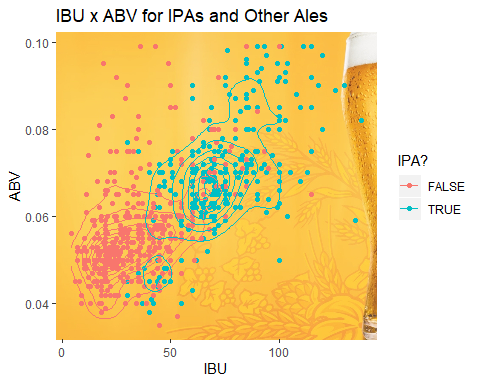
## Warning: package 'ggforce' was built under R version 3.6.3

if(!require(concaveman)) {install.packages("concaveman")}

## Loading required package: concaveman

## Warning: package 'concaveman' was built under R version 3.6.3

library(ggforce)  
library(concaveman)  
  
allAles %>% ggplot(aes(x=IBU, y=ABV)) +  
 annotation\_custom(bg, -Inf, Inf, -Inf, Inf) + # beer background  
 geom\_point(aes(color = ipa)) +  
 geom\_density2d(aes(color=ipa)) +  
  
 labs(title = "IBU x ABV for IPAs and Other Ales", color = "IPA?")

 If we map the IBU vs. the ABV for the groups “IPA” and “Others”, then add contour lines to show the greatest concentration of values, we can clearly see the groupings of IBU and ABV for Others to be much lower in IBU and ABV than the IPAs. While we can see that there are clearly some Other ales in the region associated with IPAs, we’d like to know if the two groups are distinct enough that we can predict the Style of beer from the IBU/ABV alone.

summary(ipas$IBU)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 30.00 60.00 70.00 71.95 85.00 138.00

summary(otherAles$IBU)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.00 20.00 30.00 34.33 44.25 115.00

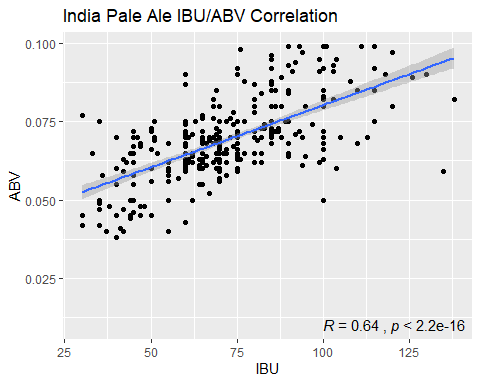
summary(ipas$ABV)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.03800 0.06200 0.06800 0.06914 0.07500 0.09900

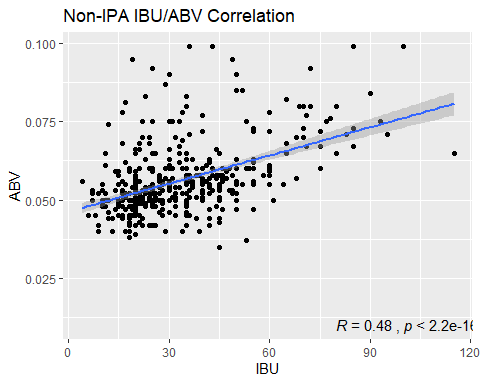
summary(otherAles$ABV)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.03500 0.05000 0.05400 0.05656 0.06000 0.09900

ipas %>% ggplot(aes(x=IBU, y=ABV)) +  
 geom\_point() + geom\_smooth(method="glm") +   
 stat\_cor(aes(x=IBU, y=ABV),method = "pearson", label.x = 100, label.y = 0.010, na.rm = TRUE) +  
 labs(title = "India Pale Ale IBU/ABV Correlation")

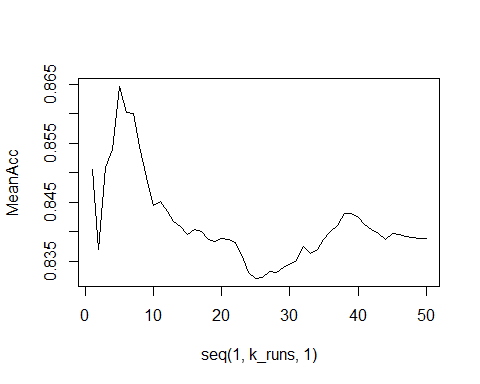


otherAles %>% ggplot(aes(x=IBU, y=ABV)) +   
 geom\_point() + geom\_smooth(method="glm") +   
 stat\_cor(aes(x=IBU, y=ABV),method = "pearson", label.x = 80, label.y = 0.010, na.rm = TRUE) +  
 labs(title = "Non-IPA IBU/ABV Correlation")



# run a set of tests to determine best k value  
iterations = 100 # the number of tests to run  
k\_runs = 50 # the number of "k"s to run in each test  
  
# a place to hold the master data  
masterAcc = matrix(nrow = iterations, ncol=k\_runs)  
  
for(j in 1:iterations) {  
 # accs = data.frame(accuracy = numeric(k\_runs), k=numeric(k\_runs))  
  
 # randomize the sample set for this test  
 ran <- sample(1:nrow(allAles), 0.9 \* nrow(allAles))  
 ipa\_train <- allAles[ran,]  
 ipa\_test <- allAles[-ran,]  
   
 # run the 'k's  
 for(i in 1:k\_runs) {  
 classifications = knn(ipa\_train[,c(4,5)], ipa\_test[,c(4,5)], ipa\_train$ipa, k=i, prob = TRUE)  
 CM = confusionMatrix(table(ipa\_test$ipa, classifications))  
 masterAcc[j,i] = CM$overall[1]  
 }  
}

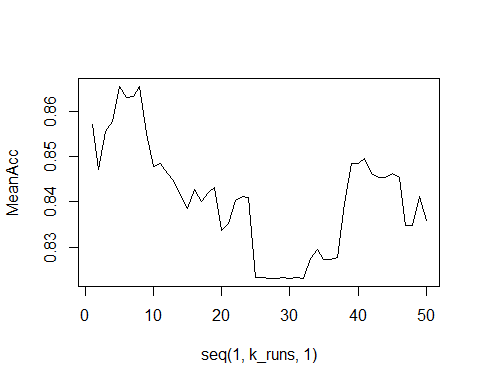
MeanAcc = colMeans(masterAcc)  
plot(seq(1,k\_runs,1),MeanAcc, type="l")



# run a single test with the experimentally gathered 'k' value  
ipa\_pr <- knn(ipa\_train[,c(4,5)], ipa\_test[,c(4,5)], ipa\_train$ipa, k=5, prob = TRUE)  
  
# get the statistics of the   
confusionMatrix(table(ipa\_test$ipa, ipa\_pr))

## Confusion Matrix and Statistics  
##   
## ipa\_pr  
## FALSE TRUE  
## FALSE 57 2  
## TRUE 6 30  
##   
## Accuracy : 0.9158   
## 95% CI : (0.8408, 0.9629)  
## No Information Rate : 0.6632   
## P-Value [Acc > NIR] : 7.377e-09   
##   
## Kappa : 0.8171   
##   
## Mcnemar's Test P-Value : 0.2888   
##   
## Sensitivity : 0.9048   
## Specificity : 0.9375   
## Pos Pred Value : 0.9661   
## Neg Pred Value : 0.8333   
## Prevalence : 0.6632   
## Detection Rate : 0.6000   
## Detection Prevalence : 0.6211   
## Balanced Accuracy : 0.9211   
##   
## 'Positive' Class : FALSE   
##

set.seed(1972)  
iterations = 100  
k\_runs = 50  
  
masterAcc = matrix(nrow = iterations, ncol = k\_runs)  
for(j in 1:iterations) {  
 accs = data.frame(accuracy = numeric(k\_runs), k = numeric(k\_runs))  
 for(i in 1:k\_runs) {  
 classifications = knn.cv(allAles[,c(4,5)], allAles$ipa, k=i, prob = TRUE)  
 CM = confusionMatrix(table(allAles$ipa, classifications))  
 masterAcc[j,i] = CM$overall[1]  
 }  
}  
  
MeanAcc = colMeans(masterAcc)  
plot(seq(1,k\_runs,1), MeanAcc, type="l")



set.seed(1972)  
classifications = knn.cv(allAles[,c(4,5)], allAles$ipa, k=5, prob = TRUE)  
confusionMatrix(table(allAles$ipa, classifications))

## Confusion Matrix and Statistics  
##   
## classifications  
## FALSE TRUE  
## FALSE 489 63  
## TRUE 63 329  
##   
## Accuracy : 0.8665   
## 95% CI : (0.8432, 0.8876)  
## No Information Rate : 0.5847   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.7252   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.8859   
## Specificity : 0.8393   
## Pos Pred Value : 0.8859   
## Neg Pred Value : 0.8393   
## Prevalence : 0.5847   
## Detection Rate : 0.5180   
## Detection Prevalence : 0.5847   
## Balanced Accuracy : 0.8626   
##   
## 'Positive' Class : FALSE   
##

The Internal Cross Validation k-Nearest Neighbor classification tells us that there is enough difference in the IBU/ABV of India Pale Ales (IPAs) to determine the Style of beer roughly 86% of the time, just from the values of IBU and ABV. We have confidence that 95% of the time, the average accuracy of the kNN model used would be between 84% and 88% accuracy.

# we must recode string factors as numbers for the distance algorithm in  
# the kNN classifier to work properly  
allAles <- allAles %>%   
 mutate(city\_id = as.integer(City)) %>%   
 mutate(style\_id = as.integer(Style))  
  
classifications = knn.cv(allAles[,c(4,5,13)], allAles$State, k=1, prob = TRUE)  
CM = confusionMatrix(table(allAles$State, classifications))  
CM$overall[1]

## Accuracy   
## 0.3040254

If we assume that:

* current breweries are selling most of their beer locally
* that they are making beers which sell well in their areas

It could be beneficial to know what types of beer would sell best in a given state based on what is currently being produced in that state. Using a kNN classification that takes into account IBU, ABV and Style of beer, we can predict with around a 30% success rate the state that beer would most fit into. 30% may not sound like much, but compared to a purely random 1-in-50 chance of picking the right state (a 2% probability), 30% is a 15x improvement.