```
---
title: "Case Study 1: Beers and Breweries""
author: Samuel Vonpaays Soh
date: 01/24/2020
'''{r}
library(tidyverse)
library(ggplot2)
library(plotly)
library(forcats)
library(dplyr)
library(class)
library(caret)
library(maps)
library(mapproj)

#Reading Beers and Brewers files.
Beers = read.csv(file.choose(),header = TRUE)
Breweries=read.csv(file.choose(),header = TRUE)
count(Beers)
count(Breweries)
head(Beers)
head(Breweries)
str(Breweries)
#Q1. Count number of brewer in each State.
Breweries %>% group_by(State) %>% summarize(count=n())
#Plot the number of brewer in each State.
Breweries %>% ggplot(aes(x=forcats::fct_infreq(State,ordered = TRUE)))
+geom_bar()+ggtitle("Number of Breweries in each State")+xlab("State")
#text(x=p, y=Breweries$State,labels =
Breweries$State,pos=3,cex=0.6,col="red")
#Q2. Mergering Beer and Breweries.
BB=merge(Beers,Breweries,by.x="Brewery_id",by.y = "Brew_ID")
#rename column name
colnames(BB)[2]<-"Beer_Name"
colnames(BB)[8] <-"Brewery_Name"
head(BB, n=6)
tail(BB,n=6)
#Q3&4. Plot the median of ABV by grouping them on each State
BB %>% group_by(State) %>% summarize(median=median(ABV,na.rm=TRUE))%>%
ggplot(aes(x=State,y=(median)))+geom_bar(stat = "identity")+ggtitle("Median
Alcohol Content by State")

#Plot the median of IBU by grouping them on each State
BB %>% group_by(State) %>% summarize(median=median(IBU,na.rm=TRUE))%>%
ggplot(aes(x=State,y=median))+geom_bar(stat = "identity")
+ggtitle("International Bitterness by State")
#Q5. Plot the max of IBU by grouping them on each State
BB %>% group_by(State) %>% summarize(max=max(ABV,na.rm=TRUE)) %>%
ggplot(aes(x=State,y=max))+geom_bar(stat = "identity")+ggtitle("Max ABV by
State")
#Plot the max of IBU by grouping them on each State
BB %>% group_by(State) %>% summarize(max=max(IBU,na.rm=TRUE)) %>%
ggplot(aes(x=State,y=max))+geom_bar(stat = "identity")+ggtitle("Max IBU by
State")
```

```
#Q6. Summary Statistics on IBU and ABV
SummaryStat= BB %>% group_by(State) %>%
summarize(mean=mean(ABV,na.rm=TRUE),sd=sd(ABV,na.rm=TRUE),range=max(ABV,na.rm=TRUE)-
min(ABV,na.rm=TRUE),IQR=IQR(ABV,na.rm=TRUE),count=n())
print(SummaryStat,n=51)
SummaryStat%>%arrange(desc(sd))%>% print(n=51)
#Q7. Correlation between IBU and ABV
install.packages("GGally")
library(GGally)
BB %>% select(ABV,IBU) %>% ggpairs()
#Q8. IPA vs Ale in IBU and ABV
#BBIPA= BB %>% filter(!is.na(IBU)) %>% filter(str_detect(Style,"IPA")) %>%
select(Style, IBU, ABV)
#head(BBIPA)

BBAleIPA=BB %>% filter(!is.na(IBU)) %>% filter(str_detect(Style,"Ale")|
str_detect(Style,"IPA")) %>% select(Style, IBU, ABV, State)
head(BBAleIPA)
dummy <-str_sub(BBAleIPA$Style,-3,-1)
BBAleIPA$Style <- ifelse(dummy == "Ale", "Ale","IPA")
# Q8. New finding that will blow their socks off
-------------------------------------------------------
colnames(BBAleIPA)[4]="abb"
head(BBAleIPA)
str_view_all(BBAleIPA$abb,"\\s")
trimws(BBAleIPA$abb, which=c("left"))
str(lookup)
str(BBAleIPA)
lookup = data.frame(abb = state.abb, State = state.name)
head(lookup)
BBAleIPA$abb=trimws(BBAleIPA$abb, which=c("left"))
BBAleIPA$abb <- as.factor(BBAleIPA$abb)
str(BBAleIPA)
BBAleIPA2= merge(BBAleIPA,lookup,by="abb")
head(BBAleIPA2)
BBAleIPACount=count(BBAleIPA2,State, sort = TRUE)
head(BBAleIPACount)
BBAleIPACount$region <- tolower(BBAleIPACount$State)
BBAleIPACount2= BBAleIPACount[-1]
states <- map_data("state")
map.df <-merge(states,BBAleIPACount2,by="region",all.x=T)
map.df <-map.df[order(map.df$order),]
ggplot(map.df,aes(x=long,y=lat,group=group))+geom_polygon(aes(fill=n))
+geom_path()
+scale_fill_gradientn(colours=rev(heat.colors(10)),na.value="grey90")
+coord_map()+ggtitle("Number of IPA and Ale  in the State")



#----------------------------------------------------------
BBAleIPA$Style <- as.factor(BBAleIPA$Style)
str(BBAleIPA)
BBAleIPA %>% ggplot(aes(x=IBU,ABV,color=Style))+geom_point()+ggtitle(" IBU
vs ABV between IPA and Ale")
```

```
summary(BBAleIPA)

trainIndices=sample(seq(1:944),round(0.7*944))
trainBeers=BBAleIPA[trainIndices,]
testBeers=BBAleIPA[-trainIndices,]


classifications= knn(trainBeers[,c(2,3)],testBeers[,c(2,3)],
(trainBeers$Style),prob=TRUE,k=13)
table(classifications,testBeers$Style)
confusionMatrix(table(classifications,testBeers$Style))


#naive Bayes
iterations = 100

masterAcc = matrix(nrow = iterations)

splitPerc = .7 #Training / Test split Percentage

for(j in 1:iterations)
{

  trainIndices=sample(seq(1:944),round(0.7*944))
  train=BBAleIPA[trainIndices,]
  test=BBAleIPA[-trainIndices,]

  model = naiveBayes(train[,c(2,3)],train$Style)
  table(predict(model,test[,c(2,3)]),test$Style)
  CM = confusionMatrix(table(predict(model,test[,c(2,3)]),test$Style))
  masterAcc[j] = CM$overall[1]
}

MeanAcc = colMeans(masterAcc)

MeanAcc
```