Project I\_B&B

Erica Brooks

2023-06-20

#Load Libraries  
library(class)  
library(caret)

## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.3

## Loading required package: lattice

library(e1071)

## Warning: package 'e1071' was built under R version 4.1.3

library(tm)

## Warning: package 'tm' was built under R version 4.1.3

## Loading required package: NLP

##   
## Attaching package: 'NLP'

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

library(plyr)  
library(dbplyr)

## Warning: package 'dbplyr' was built under R version 4.1.3

library(tidyverse)

## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v tibble 3.2.1 v dplyr 1.1.2  
## v tidyr 1.3.0 v stringr 1.5.0  
## v readr 2.1.2 v forcats 0.5.1  
## v purrr 1.0.1

## Warning: package 'tibble' was built under R version 4.1.3

## Warning: package 'tidyr' was built under R version 4.1.3

## Warning: package 'purrr' was built under R version 4.1.3

## Warning: package 'dplyr' was built under R version 4.1.3

## Warning: package 'stringr' was built under R version 4.1.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x NLP::annotate() masks ggplot2::annotate()  
## x dplyr::arrange() masks plyr::arrange()  
## x purrr::compact() masks plyr::compact()  
## x dplyr::count() masks plyr::count()  
## x dplyr::desc() masks plyr::desc()  
## x dplyr::failwith() masks plyr::failwith()  
## x dplyr::filter() masks stats::filter()  
## x dplyr::id() masks plyr::id()  
## x dplyr::ident() masks dbplyr::ident()  
## x dplyr::lag() masks stats::lag()  
## x purrr::lift() masks caret::lift()  
## x dplyr::mutate() masks plyr::mutate()  
## x dplyr::rename() masks plyr::rename()  
## x dplyr::sql() masks dbplyr::sql()  
## x dplyr::summarise() masks plyr::summarise()  
## x dplyr::summarize() masks plyr::summarize()

library(ggplot2)  
library(mapproj)

## Warning: package 'mapproj' was built under R version 4.1.3

## Loading required package: maps

## Warning: package 'maps' was built under R version 4.1.3

##   
## Attaching package: 'maps'

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

## The following object is masked from 'package:plyr':  
##   
## ozone

library(maps)  
library(naniar)

## Warning: package 'naniar' was built under R version 4.1.3

library(stringr)  
library(knitr)  
library(ggsn)

## Warning: package 'ggsn' was built under R version 4.1.3

## Loading required package: grid

library(usmap)

#Read in the Beers and Breweries data sets  
beerData = read.csv(file.choose(), header=TRUE)  
head(beerData)

## 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  
## Style Ounces  
## 1 American Pale Lager 12  
## 2 American Pale Ale (APA) 12  
## 3 American IPA 12  
## 4 American Double / Imperial IPA 12  
## 5 American IPA 12  
## 6 Oatmeal Stout 12

dim(beerData)

## [1] 2410 7

breweryData = read.csv(file.choose(), header=TRUE)  
head(breweryData)

## Brew\_ID Name City State  
## 1 1 NorthGate Brewing Minneapolis MN  
## 2 2 Against the Grain Brewery Louisville KY  
## 3 3 Jack's Abby Craft Lagers Framingham MA  
## 4 4 Mike Hess Brewing Company San Diego CA  
## 5 5 Fort Point Beer Company San Francisco CA  
## 6 6 COAST Brewing Company Charleston SC

dim(breweryData)

## [1] 558 4

#Change the beer "Name" column in the beerData to "Beer\_Name"   
#Change the brewery "Name" column in the breweryData to "Brewery\_Name"  
colnames(beerData)[1]="Beer\_Name"   
head(beerData)

## 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  
## Style Ounces  
## 1 American Pale Lager 12  
## 2 American Pale Ale (APA) 12  
## 3 American IPA 12  
## 4 American Double / Imperial IPA 12  
## 5 American IPA 12  
## 6 Oatmeal Stout 12

colnames(breweryData)[2] ="Brewery\_Name"  
head(breweryData)

## Brew\_ID Brewery\_Name City State  
## 1 1 NorthGate Brewing Minneapolis MN  
## 2 2 Against the Grain Brewery Louisville KY  
## 3 3 Jack's Abby Craft Lagers Framingham MA  
## 4 4 Mike Hess Brewing Company San Diego CA  
## 5 5 Fort Point Beer Company San Francisco CA  
## 6 6 COAST Brewing Company Charleston SC

#Question\_2 Merge Beer Data with Brewery Data; Outer Join  
BB.Data = merge(beerData, breweryData, by.x="Brewery\_id", by.y="Brew\_ID", all=TRUE)  
head(BB.Data)

## Brewery\_id Beer\_Name 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\_Name 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(BB.Data)

## Brewery\_id Beer\_Name 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\_Name 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

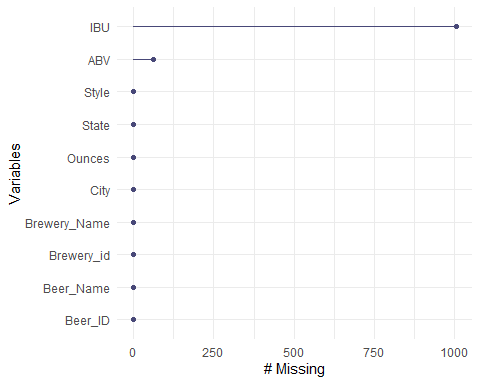
dim(BB.Data)

## [1] 2410 10

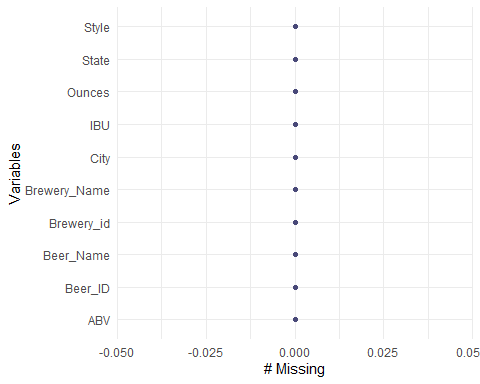
#Question\_3 Address the missing values in each column  
s=sapply(BB.Data, function(x) sum(is.na(x))) #count missing values in each column  
s

## Brewery\_id Beer\_Name Beer\_ID ABV IBU Style   
## 0 0 0 62 1005 0   
## Ounces Brewery\_Name City State   
## 0 0 0 0

gg\_miss\_var(BB.Data) #graphical view of missing values



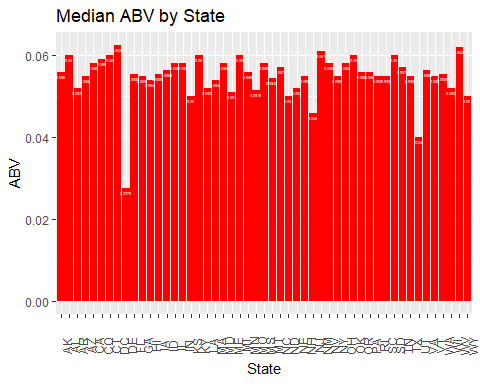
BB.Data$ABV <- replace(BB.Data$ABV, is.na(BB.Data$ABV), 0.0) #Impute the missing values in the ABV column with 0.0  
BB.Data$IBU <- replace(BB.Data$IBU, is.na(BB.Data$IBU), 0) #Impute the missing values (NA) in IBU column with 0  
gg\_miss\_var(BB.Data)



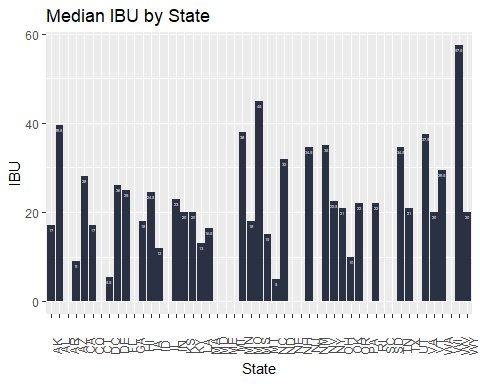
#Question\_4 Compute the median alcohol content and the international bitterness for each state  
median.Data <- BB.Data %>% filter(!is.na(ABV)) %>% group\_by(State) %>% summarize(medianABV = median(ABV, na.rm=TRUE), medianIBU = median(IBU, na.rm=TRUE))  
head(median.Data)

## # A tibble: 6 x 3  
## State medianABV medianIBU  
## <chr> <dbl> <dbl>  
## 1 " AK" 0.056 17   
## 2 " AL" 0.06 39.5  
## 3 " AR" 0.052 0   
## 4 " AZ" 0.055 9   
## 5 " CA" 0.058 28   
## 6 " CO" 0.059 17

median.Data <- median.Data %>% arrange(desc(medianABV)) #Sort the ABV column in ascending order  
  
  
#Plot the median data for the medianABV  
median.Data %>% ggplot(aes(x=State))+  
 geom\_bar(aes(y=medianABV), fill="red", stat="identity")+  
 xlab("State")+  
 ylab("ABV")+  
 ggtitle("Median ABV by State")+  
 geom\_text(aes(y=medianABV, label = medianABV), size = 1, vjust = 1.5, colour = "white")+  
 theme(axis.text.x=element\_text(angle=90,margin = margin(.5, unit = "cm"),vjust = 1),legend.position = "none")



#Plot the median data for the medianIBU  
median.Data %>% ggplot(aes(x=State))+  
 geom\_bar(aes(y=medianIBU), fill="#2A3142", stat="identity")+  
 xlab("State")+  
 ylab("IBU")+  
 ggtitle("Median IBU by State")+  
 geom\_text(aes(y=medianIBU, label = medianIBU), size = 1, vjust = 1.5, colour = "white")+  
 theme(axis.text.x=element\_text(angle=90,margin = margin(.5, unit = "cm"),vjust = 1),legend.position = "none")



#Question\_5 State with the maximum ABV and the state with the maximum IBU  
#Create a heat map for ABV  
  
BB.Data$State <- trimws(BB.Data$State, which=c("left")) #Remove the leading trail spaces from the "State" column  
str(BB.Data)

## 'data.frame': 2410 obs. of 10 variables:  
## $ Brewery\_id : int 1 1 1 1 1 1 2 2 2 2 ...  
## $ Beer\_Name : chr "Get Together" "Maggie's Leap" "Wall's End" "Pumpion" ...  
## $ Beer\_ID : int 2692 2691 2690 2689 2688 2687 2686 2685 2684 2683 ...  
## $ ABV : num 0.045 0.049 0.048 0.06 0.06 0.056 0.08 0.125 0.077 0.042 ...  
## $ IBU : num 50 26 19 38 25 47 68 80 25 42 ...  
## $ Style : chr "American IPA" "Milk / Sweet Stout" "English Brown Ale" "Pumpkin Ale" ...  
## $ Ounces : num 16 16 16 16 16 16 16 16 16 16 ...  
## $ Brewery\_Name: chr "NorthGate Brewing " "NorthGate Brewing " "NorthGate Brewing " "NorthGate Brewing " ...  
## $ City : chr "Minneapolis" "Minneapolis" "Minneapolis" "Minneapolis" ...  
## $ State : chr "MN" "MN" "MN" "MN" ...

ABV.Data <- BB.Data[, c(4,10)] #Create a dataframe with just the ABV data  
head(ABV.Data)

## ABV State  
## 1 0.045 MN  
## 2 0.049 MN  
## 3 0.048 MN  
## 4 0.060 MN  
## 5 0.060 MN  
## 6 0.056 MN

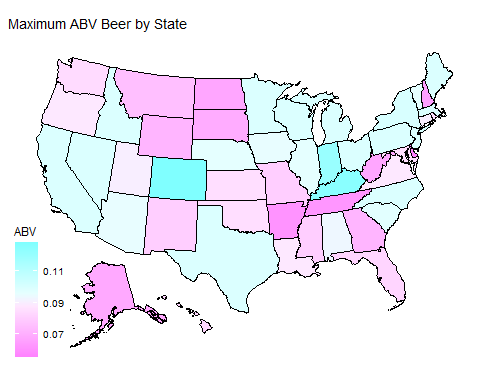
#Get the state with maximum alcoholic beer  
max\_ABV <- ABV.Data %>% group\_by(State) %>%  
 slice\_max(ABV, n=1)  
max\_ABV

## # A tibble: 65 x 2  
## # Groups: State [51]  
## ABV State  
## <dbl> <chr>  
## 1 0.068 AK   
## 2 0.093 AL   
## 3 0.061 AR   
## 4 0.095 AZ   
## 5 0.099 CA   
## 6 0.099 CA   
## 7 0.099 CA   
## 8 0.099 CA   
## 9 0.128 CO   
## 10 0.09 CT   
## # i 55 more rows

#Remove duplicate values  
max\_ABV <- distinct(max\_ABV, ABV, State, .keep\_all= TRUE)  
max\_ABV

## # A tibble: 51 x 2  
## # Groups: State [51]  
## ABV State  
## <dbl> <chr>  
## 1 0.068 AK   
## 2 0.093 AL   
## 3 0.061 AR   
## 4 0.095 AZ   
## 5 0.099 CA   
## 6 0.128 CO   
## 7 0.09 CT   
## 8 0.092 DC   
## 9 0.055 DE   
## 10 0.082 FL   
## # i 41 more rows

colnames(max\_ABV)[2] = "state" #change the name of column "region" to "state"  
  
#Plot ABV Data  
plot\_usmap(data = max\_ABV, values = "ABV") +   
 scale\_fill\_gradientn(colours=rev(cm.colors(10)), na.value = "grey90") +   
 labs(title = "Maximum ABV Beer by State")



#Create a heat map for IBU  
IBU.Data <- BB.Data[, c(5,10)] #Create a df with just the IBU data  
head(IBU.Data)

## IBU State  
## 1 50 MN  
## 2 26 MN  
## 3 19 MN  
## 4 38 MN  
## 5 25 MN  
## 6 47 MN

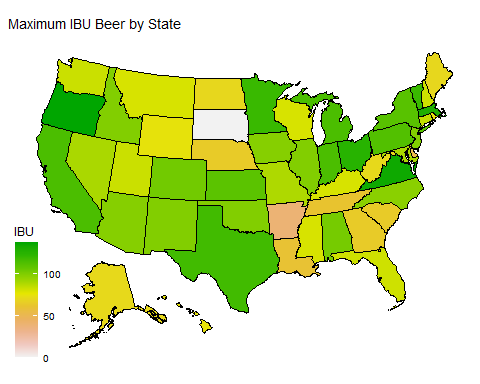
#Get the maximum value of IBU for each state  
max\_IBU <- IBU.Data %>% group\_by(State) %>%  
 slice\_max(IBU, n=1)  
max\_IBU

## # A tibble: 67 x 2  
## # Groups: State [51]  
## IBU State  
## <dbl> <chr>  
## 1 71 AK   
## 2 103 AL   
## 3 39 AR   
## 4 99 AZ   
## 5 115 CA   
## 6 104 CO   
## 7 85 CT   
## 8 85 CT   
## 9 115 DC   
## 10 52 DE   
## # i 57 more rows

#Remove duplicate values  
max\_IBU <- distinct(max\_IBU, IBU, State, .keep\_all= TRUE)  
max\_IBU

## # A tibble: 51 x 2  
## # Groups: State [51]  
## IBU State  
## <dbl> <chr>  
## 1 71 AK   
## 2 103 AL   
## 3 39 AR   
## 4 99 AZ   
## 5 115 CA   
## 6 104 CO   
## 7 85 CT   
## 8 115 DC   
## 9 52 DE   
## 10 82 FL   
## # i 41 more rows

colnames(max\_IBU)[2] = "state" #change the name of column "State" to "state"  
  
plot\_usmap(data = max\_IBU, values = "IBU") +   
 scale\_fill\_gradientn(colours=rev(terrain.colors(10)), na.value = "grey90") +   
 labs(title = "Maximum IBU Beer by State")



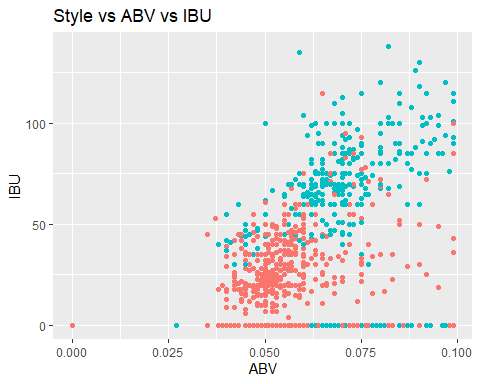
#Question\_8 KNN Model   
set.seed(6) #Make the data reproducible. Get the same random data.  
  
#Filtered the beers that were IPA and Ale   
BB.Data$Style = replace(BB.Data$Style, grepl("IPA", BB.Data$Style), "IPA")  
BB.Data$Style = replace(BB.Data$Style, grepl("Ale", BB.Data$Style), "Ale")  
  
IPA\_Ale = BB.Data %>% filter(Style == "IPA" | Style == "Ale")  
  
#Partition the 1534 Observations into a smaller training set and testing set  
splitPerc = .75  
trainInd = sample(1:dim(IPA\_Ale)[1], round(splitPerc \* dim(IPA\_Ale)[1]))  
train = IPA\_Ale[trainInd,]  
dim(train)

## [1] 1150 10

test = IPA\_Ale[-trainInd,]  
dim(test)

## [1] 384 10

#Plot the data   
IPA\_Ale %>% ggplot(aes(x = ABV, y =IBU, color = Style)) + geom\_point() +   
 ggtitle("Style vs ABV vs IBU") +   
 xlab("ABV")+   
 theme(legend.position = "none")



#KNN Classification with a k = 5  
classifications = knn(train[,c(4,5)],test[,c(4,5)],train$Style, prob = TRUE, k = 5)  
table(test$Style,classifications)

## classifications  
## Ale IPA  
## Ale 224 14  
## IPA 47 99

cm = confusionMatrix(table(test$Style,classifications))  
cm

## Confusion Matrix and Statistics  
##   
## classifications  
## Ale IPA  
## Ale 224 14  
## IPA 47 99  
##   
## Accuracy : 0.8411   
## 95% CI : (0.8007, 0.8763)  
## No Information Rate : 0.7057   
## P-Value [Acc > NIR] : 5.129e-10   
##   
## Kappa : 0.6475   
##   
## Mcnemar's Test P-Value : 4.182e-05   
##   
## Sensitivity : 0.8266   
## Specificity : 0.8761   
## Pos Pred Value : 0.9412   
## Neg Pred Value : 0.6781   
## Prevalence : 0.7057   
## Detection Rate : 0.5833   
## Detection Prevalence : 0.6198   
## Balanced Accuracy : 0.8513   
##   
## 'Positive' Class : Ale   
##