Beers Project

#Title: "Beers Project"  
#MSDS 6306: Doing Data Science - Case Study 01  
#Group Members: Sowmya Mani & Migot Ndede  
#output:  
#word\_document: default  
#html\_document: default  
#pdf\_document: default  
#Date: March 06 2021  
  
#Introduction: This Case Study is about the Market Analysis of Beers and Breweries within the state of US  
  
#The data set used for this case study analysis consists of 2 Datasets:  
  
#Beers Dataset with 2410 Craft Beers along with their details about:  
#Name: Name of the beer  
#Beer\_ID: Unique identifier of the beer  
#ABV: Alcohol by volume of the beer  
#IBU: International Bitterness Units of the beer  
#Brewery\_ID: Brewery id associated with the beer  
#Style: Style of the beer  
#Ounces: Ounces of beer  
  
#Breweries Dataset with 558 Breweries across US along with their details about  
#Name: Name of the brewery  
#City: City where the brewery is located  
#State: U.S. State where the brewery is located  
#Brew\_ID: Unique identifier of the brewery  
  
#The goal of our team is to analyze the Beers market and present our analysis to the CEO & CFO of the Budweiser   
   
#Libraries loaded for the ANalysis  
library(XML)

## Warning: package 'XML' was built under R version 4.0.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 4.0.3

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

library(RCurl)

## Warning: package 'RCurl' was built under R version 4.0.3

library(httr)

## Warning: package 'httr' was built under R version 4.0.3

library(jsonlite)

## Warning: package 'jsonlite' was built under R version 4.0.3

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.0.3

## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3 v purrr 0.3.4  
## v tibble 3.0.4 v stringr 1.4.0  
## v tidyr 1.1.2 v forcats 0.5.0  
## v readr 1.4.0

## Warning: package 'ggplot2' was built under R version 4.0.3

## Warning: package 'tibble' was built under R version 4.0.3

## Warning: package 'tidyr' was built under R version 4.0.3

## Warning: package 'readr' was built under R version 4.0.3

## Warning: package 'purrr' was built under R version 4.0.3

## Warning: package 'stringr' was built under R version 4.0.3

## Warning: package 'forcats' was built under R version 4.0.3

## -- Conflicts ------------------------------------------ tidyverse\_conflicts() --  
## x tidyr::complete() masks RCurl::complete()  
## x dplyr::filter() masks stats::filter()  
## x purrr::flatten() masks jsonlite::flatten()  
## x dplyr::lag() masks stats::lag()

library(naniar)

## Warning: package 'naniar' was built under R version 4.0.3

library(GGally)

## Warning: package 'GGally' was built under R version 4.0.3

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

library(ggplot2)  
library(class)  
library(caret)

## Warning: package 'caret' was built under R version 4.0.3

## Loading required package: lattice

##   
## Attaching package: 'caret'

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

## The following object is masked from 'package:httr':  
##   
## progress

library(knnp)

## Warning: package 'knnp' was built under R version 4.0.3

##   
## Attaching package: 'knnp'

## The following object is masked from 'package:class':  
##   
## knn

library(e1071)

## Warning: package 'e1071' was built under R version 4.0.3

library(ggplot2)  
library(maps)

## Warning: package 'maps' was built under R version 4.0.3

##   
## Attaching package: 'maps'

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

library(dplyr)  
library(mapproj)

## Warning: package 'mapproj' was built under R version 4.0.3

library(ggplot2)  
library(dplyr)  
library(ggcorrplot)

## Warning: package 'ggcorrplot' was built under R version 4.0.3

#Import the Beers Data  
Beers\_orig<-read.csv('C:/Sowmya/SMU/04\_Doing Data Science/Unit-8 & Unit-9/Dataset-original/Beers\_original.csv',header = TRUE)  
  
#Quick Peek at the SUmmary data of the available dataset  
summary(Beers\_orig)

## Name Beer\_ID ABV IBU   
## Length:2410 Min. : 1.0 Min. :0.00100 Min. : 4.00   
## Class :character 1st Qu.: 808.2 1st Qu.:0.05000 1st Qu.: 21.00   
## Mode :character Median :1453.5 Median :0.05600 Median : 35.00   
## Mean :1431.1 Mean :0.05977 Mean : 42.71   
## 3rd Qu.:2075.8 3rd Qu.:0.06700 3rd Qu.: 64.00   
## Max. :2692.0 Max. :0.12800 Max. :138.00   
## NA's :62 NA's :1005   
## Brewery\_id Style Ounces   
## Min. : 1.0 Length:2410 Min. : 8.40   
## 1st Qu.: 94.0 Class :character 1st Qu.:12.00   
## Median :206.0 Mode :character Median :12.00   
## Mean :232.7 Mean :13.59   
## 3rd Qu.:367.0 3rd Qu.:16.00   
## Max. :558.0 Max. :32.00   
##

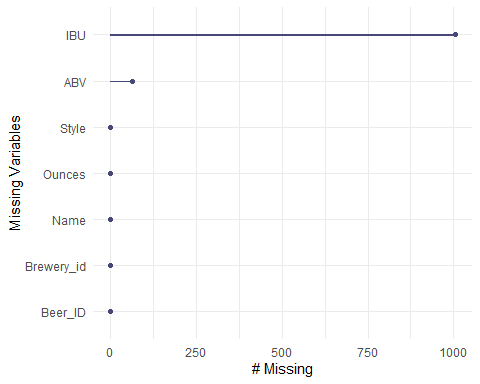
str(Beers\_orig)

## 'data.frame': 2410 obs. of 7 variables:  
## $ Name : chr "Pub Beer" "Devil's Cup" "Rise of the Phoenix" "Sinister" ...  
## $ Beer\_ID : int 1436 2265 2264 2263 2262 2261 2260 2259 2258 2131 ...  
## $ ABV : num 0.05 0.066 0.071 0.09 0.075 0.077 0.045 0.065 0.055 0.086 ...  
## $ IBU : int NA NA NA NA NA NA NA NA NA NA ...  
## $ Brewery\_id: int 409 178 178 178 178 178 178 178 178 178 ...  
## $ Style : chr "American Pale Lager" "American Pale Ale (APA)" "American IPA" "American Double / Imperial IPA" ...  
## $ Ounces : num 12 12 12 12 12 12 12 12 12 12 ...

#Checking for Missing Data  
sapply(Beers\_orig,function(x) sum(is.na(x)))

## Name Beer\_ID ABV IBU Brewery\_id Style Ounces   
## 0 0 62 1005 0 0 0

gg\_miss\_var(Beers\_orig)+xlab("Missing Variables")



#The Beer dataset has 2410 records with 7 variables providing more information on the different Craft Beers.Find that the below missing data in the dataset.ABV – 62 missing data,IBU – 1005 missing data.The variable summary and the variable types of the original data set has been provided for reference.Used gg\_miss\_var to find the missing records.  
  
#Researched the internet for the missing data and were able to fill the ABV and IBU.Searched with the below:Beer Name,Beer Style,Ounces,Populated the missing data:  
#ABV – 62 missing data  
#IBU – 1005 missing data  
  
Beers<-read.csv('C:/Sowmya/SMU/04\_Doing Data Science/Unit-8 & Unit-9/Beers.csv',header = TRUE)  
  
summary(Beers)

## Name Beer\_ID ABV IBU   
## Length:2410 Min. : 1.0 Min. :0.00100 Min. : 4.00   
## Class :character 1st Qu.: 808.2 1st Qu.:0.05000 1st Qu.: 22.00   
## Mode :character Median :1453.5 Median :0.05600 Median : 35.00   
## Mean :1431.1 Mean :0.05972 Mean : 41.32   
## 3rd Qu.:2075.8 3rd Qu.:0.06700 3rd Qu.: 60.00   
## Max. :2692.0 Max. :0.12800 Max. :138.00   
## Brewery\_id Style Ounces   
## Min. : 1.0 Length:2410 Min. : 8.40   
## 1st Qu.: 94.0 Class :character 1st Qu.:12.00   
## Median :206.0 Mode :character Median :12.00   
## Mean :232.7 Mean :13.59   
## 3rd Qu.:367.0 3rd Qu.:16.00   
## Max. :558.0 Max. :32.00

str(Beers)

## 'data.frame': 2410 obs. of 7 variables:  
## $ Name : chr "Pub Beer" "Devil's Cup" "Rise of the Phoenix" "Sinister" ...  
## $ Beer\_ID : int 1436 2265 2264 2263 2262 2261 2260 2259 2258 2131 ...  
## $ ABV : num 0.05 0.066 0.071 0.09 0.075 0.077 0.045 0.065 0.055 0.086 ...  
## $ IBU : int 22 45 70 75 70 35 90 32 45 75 ...  
## $ Brewery\_id: int 409 178 178 178 178 178 178 178 178 178 ...  
## $ Style : chr "American Pale Lager" "American Pale Ale (APA)" "American IPA" "American Double / Imperial IPA" ...  
## $ Ounces : num 12 12 12 12 12 12 12 12 12 12 ...

#Rechecking if the data set has any misisng data  
sapply(Beers,function(x) sum(is.na(x)))

## Name Beer\_ID ABV IBU Brewery\_id Style Ounces   
## 0 0 0 0 0 0 0

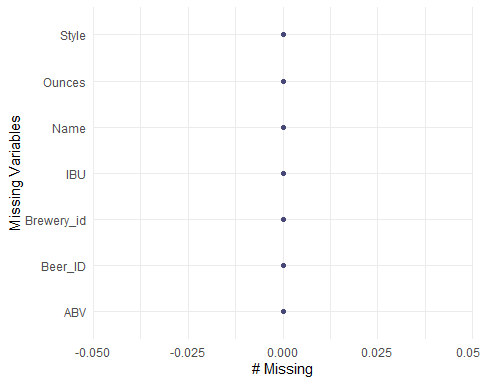
summary(Beers)

## Name Beer\_ID ABV IBU   
## Length:2410 Min. : 1.0 Min. :0.00100 Min. : 4.00   
## Class :character 1st Qu.: 808.2 1st Qu.:0.05000 1st Qu.: 22.00   
## Mode :character Median :1453.5 Median :0.05600 Median : 35.00   
## Mean :1431.1 Mean :0.05972 Mean : 41.32   
## 3rd Qu.:2075.8 3rd Qu.:0.06700 3rd Qu.: 60.00   
## Max. :2692.0 Max. :0.12800 Max. :138.00   
## Brewery\_id Style Ounces   
## Min. : 1.0 Length:2410 Min. : 8.40   
## 1st Qu.: 94.0 Class :character 1st Qu.:12.00   
## Median :206.0 Mode :character Median :12.00   
## Mean :232.7 Mean :13.59   
## 3rd Qu.:367.0 3rd Qu.:16.00   
## Max. :558.0 Max. :32.00

str(Beers)

## 'data.frame': 2410 obs. of 7 variables:  
## $ Name : chr "Pub Beer" "Devil's Cup" "Rise of the Phoenix" "Sinister" ...  
## $ Beer\_ID : int 1436 2265 2264 2263 2262 2261 2260 2259 2258 2131 ...  
## $ ABV : num 0.05 0.066 0.071 0.09 0.075 0.077 0.045 0.065 0.055 0.086 ...  
## $ IBU : int 22 45 70 75 70 35 90 32 45 75 ...  
## $ Brewery\_id: int 409 178 178 178 178 178 178 178 178 178 ...  
## $ Style : chr "American Pale Lager" "American Pale Ale (APA)" "American IPA" "American Double / Imperial IPA" ...  
## $ Ounces : num 12 12 12 12 12 12 12 12 12 12 ...

gg\_miss\_var(Beers)+xlab("Missing Variables")



## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

#Import the Breweries dataset  
Breweries<-read.csv('C:/Sowmya/SMU/04\_Doing Data Science/Unit-8 & Unit-9/Breweries.csv',header = TRUE)  
  
#Quick Peek at the SUmmary data of the available dataset  
summary(Breweries)

## Brew\_ID Name City State   
## Min. : 1.0 Length:558 Length:558 Length:558   
## 1st Qu.:140.2 Class :character Class :character Class :character   
## Median :279.5 Mode :character Mode :character Mode :character   
## Mean :279.5   
## 3rd Qu.:418.8   
## Max. :558.0

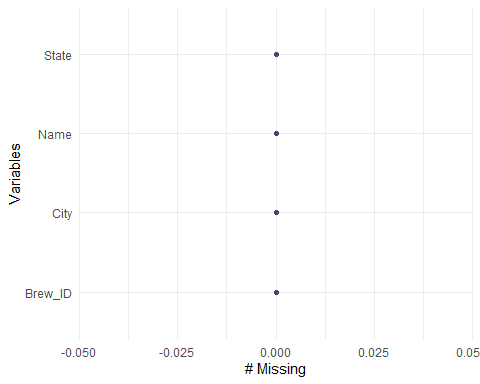
str(Breweries)

## 'data.frame': 558 obs. of 4 variables:  
## $ Brew\_ID: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Name : chr "NorthGate Brewing " "Against the Grain Brewery" "Jack's Abby Craft Lagers" "Mike Hess Brewing Company" ...  
## $ City : chr "Minneapolis" "Louisville" "Framingham" "San Diego" ...  
## $ State : chr " MN" " KY" " MA" " CA" ...

#Checking for Missing Data  
sapply(Breweries,function(x) sum(is.na(x)))

## Brew\_ID Name City State   
## 0 0 0 0

gg\_miss\_var(Breweries)



#The Brewery dataset has 558 records with 4 variables providing more information on the different Breweries across the states of US.  
  
#Find that the state variable had whitespaces on the left and would like to have it trimmed so we have a clean dataset.The dataset is ready to be merged with any dataset  
  
Breweries$State = str\_trim(Breweries$State)  
str(Breweries)

## 'data.frame': 558 obs. of 4 variables:  
## $ Brew\_ID: int 1 2 3 4 5 6 7 8 9 10 ...  
## $ Name : chr "NorthGate Brewing " "Against the Grain Brewery" "Jack's Abby Craft Lagers" "Mike Hess Brewing Company" ...  
## $ City : chr "Minneapolis" "Louisville" "Framingham" "San Diego" ...  
## $ State : chr "MN" "KY" "MA" "CA" ...

summary(Breweries)

## Brew\_ID Name City State   
## Min. : 1.0 Length:558 Length:558 Length:558   
## 1st Qu.:140.2 Class :character Class :character Class :character   
## Median :279.5 Mode :character Mode :character Mode :character   
## Mean :279.5   
## 3rd Qu.:418.8   
## Max. :558.0

## Including Plots

You can also embed plots, for example:

#Lets answer some of the questions of interest  
#How many breweries are present in each state?  
#Lets look at the Breweries dataset and analyze the data.Extracting the state and count of Breweries\_ID per state  
  
TotBreweries=Breweries%>%group\_by(State)%>%summarize(cnt = length(unique(Brew\_ID)))%>%arrange(cnt,State)%>%select(State,cnt)

## `summarise()` ungrouping output (override with `.groups` argument)

#Arranging the data in descending order  
TotBreweries

## # A tibble: 51 x 2  
## State cnt  
## <chr> <int>  
## 1 DC 1  
## 2 ND 1  
## 3 SD 1  
## 4 WV 1  
## 5 AR 2  
## 6 DE 2  
## 7 MS 2  
## 8 NV 2  
## 9 AL 3  
## 10 KS 3  
## # ... with 41 more rows

TotBreweries%>%arrange(desc(cnt))

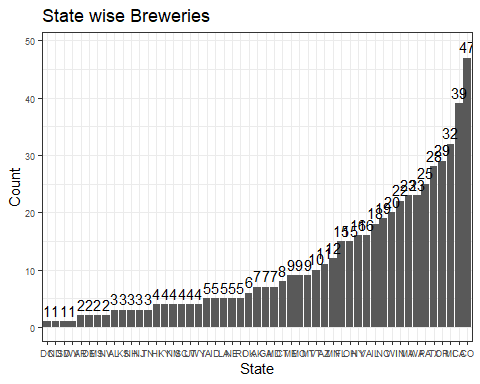
## # A tibble: 51 x 2  
## State cnt  
## <chr> <int>  
## 1 CO 47  
## 2 CA 39  
## 3 MI 32  
## 4 OR 29  
## 5 TX 28  
## 6 PA 25  
## 7 MA 23  
## 8 WA 23  
## 9 IN 22  
## 10 WI 20  
## # ... with 41 more rows

str(TotBreweries)

## tibble [51 x 2] (S3: tbl\_df/tbl/data.frame)  
## $ State: chr [1:51] "DC" "ND" "SD" "WV" ...  
## $ cnt : int [1:51] 1 1 1 1 2 2 2 2 3 3 ...

TotBreweries%>%group\_by(cnt,State)%>%arrange(desc(cnt))%>%ggplot(aes(x=reorder(factor(State),cnt),y=cnt))+geom\_bar(stat="identity")+geom\_text(aes(State, TotBreweries$cnt + 2, label = TotBreweries$cnt, fill = NULL), data = TotBreweries)+theme\_bw()+ylab("Count")+xlab("State")+ggtitle("State wise Breweries")+theme(axis.text.x = element\_text(size = 6.5),axis.text.y = element\_text(size = 7))

## Warning: Use of `TotBreweries$cnt` is discouraged. Use `cnt` instead.  
  
## Warning: Use of `TotBreweries$cnt` is discouraged. Use `cnt` instead.



#Top 10 states with Breweries  
Brewtop10=Breweries%>%group\_by(State)%>%summarize(Top10=length(unique(Brew\_ID)))%>%arrange(desc(Top10),State)%>%select(State,Top10)%>%head(10)

## `summarise()` ungrouping output (override with `.groups` argument)

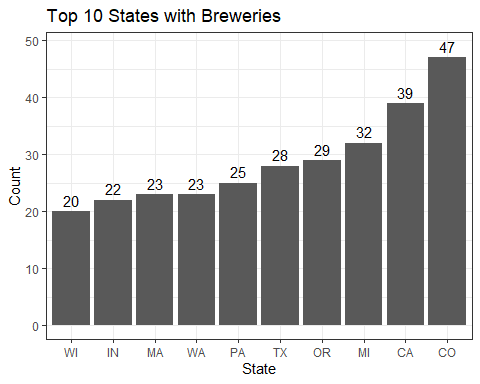
Brewtop10

## # A tibble: 10 x 2  
## State Top10  
## <chr> <int>  
## 1 CO 47  
## 2 CA 39  
## 3 MI 32  
## 4 OR 29  
## 5 TX 28  
## 6 PA 25  
## 7 MA 23  
## 8 WA 23  
## 9 IN 22  
## 10 WI 20

Brewtop10%>%group\_by(desc(Top10),State)%>%arrange(desc(Top10),State)%>%ggplot(aes(x=reorder(factor(State),Top10),y=Top10))+geom\_bar(stat="identity")+geom\_text(aes(State, Brewtop10$Top10 + 2, label = Brewtop10$Top10, fill = NULL), data = Brewtop10)+theme\_bw()+ylab("Count")+xlab("State")+ggtitle("Top 10 States with Breweries")

## Warning: Use of `Brewtop10$Top10` is discouraged. Use `Top10` instead.

## Warning: Use of `Brewtop10$Top10` is discouraged. Use `Top10` instead.



#Bottom 10 with Breweries  
Brewbottom10=Breweries%>%group\_by(State)%>%summarize(Bottom10 = length(unique(Brew\_ID)))%>%arrange(desc(Bottom10),State)%>%select(State,Bottom10)%>%tail(10)

## `summarise()` ungrouping output (override with `.groups` argument)

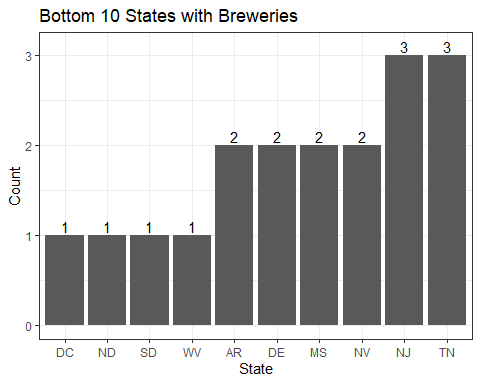
Brewbottom10

## # A tibble: 10 x 2  
## State Bottom10  
## <chr> <int>  
## 1 NJ 3  
## 2 TN 3  
## 3 AR 2  
## 4 DE 2  
## 5 MS 2  
## 6 NV 2  
## 7 DC 1  
## 8 ND 1  
## 9 SD 1  
## 10 WV 1

Brewbottom10%>%group\_by(desc(Bottom10),State)%>%arrange(desc(Bottom10),State)%>%ggplot(aes(x=reorder(factor(State),Bottom10),y=Bottom10))+geom\_bar(stat="identity")+geom\_text(aes(State, Brewbottom10$Bottom10+0.1, label = Brewbottom10$Bottom10, fill = NULL), data = Brewbottom10)+theme\_bw()+ylab("Count")+xlab("State")+ggtitle("Bottom 10 States with Breweries")

## Warning: Use of `Brewbottom10$Bottom10` is discouraged. Use `Bottom10` instead.

## Warning: Use of `Brewbottom10$Bottom10` is discouraged. Use `Bottom10` instead.



#Creating a map for the Breweries data  
lookup = data.frame(State = state.abb, State\_Name = state.name)  
MergeBrewup = left\_join(TotBreweries,lookup, by = "State")  
us <- map\_data("state")  
arr <- MergeBrewup %>%add\_rownames("region")%>%mutate(region=tolower(State\_Name))

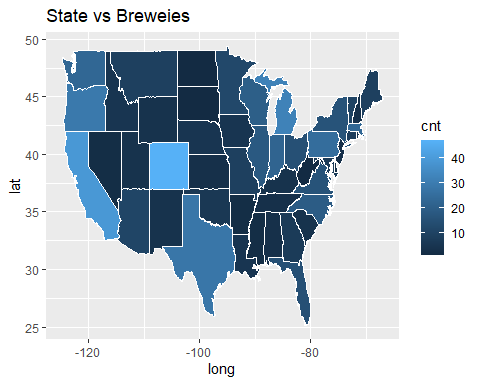
## Warning: `add\_rownames()` is deprecated as of dplyr 1.0.0.  
## Please use `tibble::rownames\_to\_column()` instead.  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

gg <- ggplot()  
gg <- gg + geom\_map(data=us, map=us,  
 aes(x=long, y=lat, map\_id=region),  
 fill="#ffffff", color="#ffffff", size=0.15)+expand\_limits(x = us$long, y = us$lat)

## Warning: Ignoring unknown aesthetics: x, y

gg <- gg + geom\_map(data=arr, map=us,  
 aes(fill=cnt, map\_id=arr$region),  
 color="#ffffff", size=0.15)+ggtitle("State vs Breweies")  
gg

## Warning: Use of `arr$region` is discouraged. Use `region` instead.



#We find that Colorado is the state with the largest number of breweries.California, Michigan, Oregon, Texas and Pennsylvania are the states with the next largest breweries after Colorado.West Virginia, South Dakota, North Dakota, Nevada and DC are the states with the least breweries

Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

#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.)  
  
#Renaming the Breweries ID column to be in sync across "Breweries" and "Beers" dataset  
colnames(Beers)[5] = "Brew\_ID"  
colnames(Beers)[1] = "Beer\_Name"  
colnames(Breweries)[2] = "Brew\_Name"  
#Merging the 2 dataset  
Beer\_Merg = left\_join(Beers,Breweries, by = "Brew\_ID")  
head(Beer\_Merg,n=6)

## Beer\_Name Beer\_ID ABV IBU Brew\_ID Style  
## 1 Pub Beer 1436 0.050 22 409 American Pale Lager  
## 2 Devil's Cup 2265 0.066 45 178 American Pale Ale (APA)  
## 3 Rise of the Phoenix 2264 0.071 70 178 American IPA  
## 4 Sinister 2263 0.090 75 178 American Double / Imperial IPA  
## 5 Sex and Candy 2262 0.075 70 178 American IPA  
## 6 Black Exodus 2261 0.077 35 178 Oatmeal Stout  
## Ounces Brew\_Name City State  
## 1 12 10 Barrel Brewing Company Bend OR  
## 2 12 18th Street Brewery Gary IN  
## 3 12 18th Street Brewery Gary IN  
## 4 12 18th Street Brewery Gary IN  
## 5 12 18th Street Brewery Gary IN  
## 6 12 18th Street Brewery Gary IN

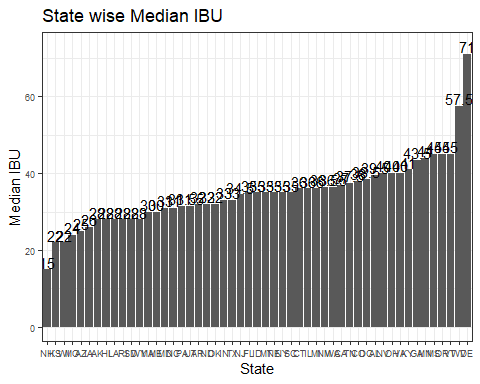
tail(Beer\_Merg,n=6)

## Beer\_Name Beer\_ID ABV IBU Brew\_ID  
## 2405 Rocky Mountain Oyster Stout 1035 0.075 75 425  
## 2406 Belgorado 928 0.067 45 425  
## 2407 Rail Yard Ale 807 0.052 22 425  
## 2408 B3K Black Lager 620 0.055 12 425  
## 2409 Silverback Pale Ale 145 0.055 40 425  
## 2410 Rail Yard Ale (2009) 84 0.052 22 425  
## Style Ounces Brew\_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

#The dataset Beers and Breweries have been merged on Brew\_ID.The columns Brewery\_ID has been renamed to be in sync with the Brewery dataset.The columns Name on both Beer and Brewery dataset has been updated to Brew\_Name and Beer\_Name to ensure we have unique columns

#4.Compute the median alcohol content and international bitterness unit for each state. Plot a bar chart to compare  
  
#Lets look at the Median IBU data per state  
Beer\_Merg%>%select(State, IBU)%>%group\_by(State)%>%arrange(desc(IBU))%>%summarize(Med.IBU = median(IBU))%>%ggplot(aes(x=reorder(factor(State),Med.IBU),y=Med.IBU),fill = State)+geom\_col()+geom\_text(aes(State, Med.IBU + 2, label = Med.IBU, fill = NULL))+theme\_bw()+ylab("Median IBU")+xlab("State")+ggtitle("State wise Median IBU")+theme(axis.text.x = element\_text(size = 6.5),axis.text.y = element\_text(size = 7))

## `summarise()` ungrouping output (override with `.groups` argument)



#Summarizing the Medium IBU data per Stater  
Beer\_IBU = Beer\_Merg%>%select(State, IBU)%>%group\_by(State)%>%arrange(desc(IBU))%>%summarize(Med.IBU = median(IBU))

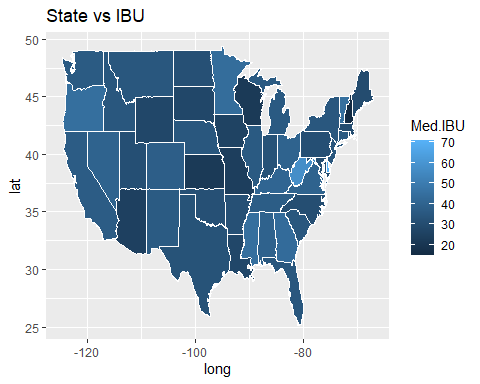
## `summarise()` ungrouping output (override with `.groups` argument)

#Creating a Map to see the spread across the states  
MergeBrewup = left\_join(Beer\_IBU,lookup, by = "State")  
us <- map\_data("state")  
arr <- MergeBrewup %>%add\_rownames("region")%>%mutate(region=tolower(State\_Name))  
gg <- ggplot()  
gg <- gg + geom\_map(data=us, map=us,  
 aes(x=long, y=lat, map\_id=region),  
 fill="#ffffff", color="#ffffff", size=0.15)+expand\_limits(x = us$long, y = us$lat)

## Warning: Ignoring unknown aesthetics: x, y

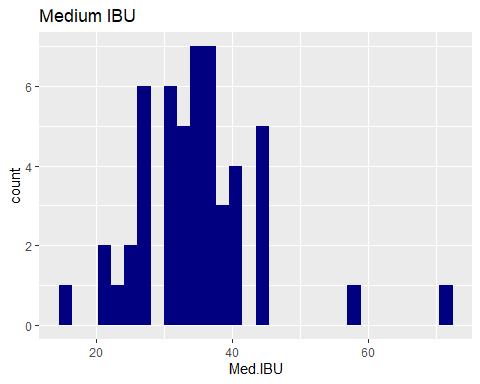
gg <- gg + geom\_map(data=arr, map=us,  
 aes(fill=Med.IBU, map\_id=arr$region),  
 color="#ffffff", size=0.15)+ggtitle("State vs IBU")  
gg

## Warning: Use of `arr$region` is discouraged. Use `region` instead.



#Histogram to analyze the data  
Beer\_IBU%>%ggplot(aes(x=Med.IBU))+geom\_histogram(fill = "Navy Blue")+ggtitle("Medium IBU")

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

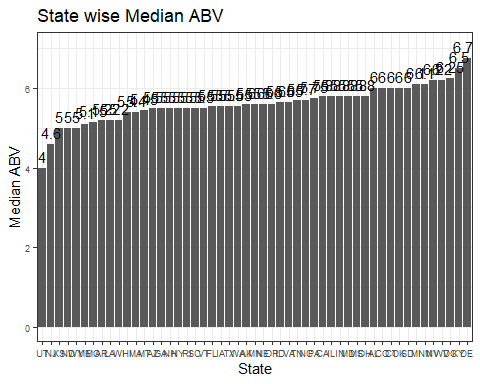


#Summary of Medium IBU  
summary(Beer\_IBU$Med.IBU)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 15.00 30.00 35.00 34.84 38.25 71.00

#Delaware is the state with medium IBU of 71 and West Virginia with median IBU of 57.5.New Hampshire is the state with the least median IBU of 15.Summary of IBU looks normally distributed and with this dataset we have no evidence to sugest the data is not normally distributed.  
  
  
#Lets look at the Median ABV data per state  
Beer\_Merg%>%select(State, ABV)%>%group\_by(State)%>%arrange(desc(ABV))%>%summarize(Med.ABV = median(ABV\*100))%>%ggplot(aes(x=reorder(factor(State),Med.ABV),y=Med.ABV),fill = State)+geom\_col()+geom\_text(aes(State, Med.ABV + 0.3, label = Med.ABV, fill = NULL))+theme\_bw()+ylab("Median ABV")+xlab("State")+ggtitle("State wise Median ABV")+theme(axis.text.x = element\_text(size = 6.5),axis.text.y = element\_text(size = 7))

## `summarise()` ungrouping output (override with `.groups` argument)



#Summarizing the Medium IBU data per Stater  
Beer\_ABV = Beer\_Merg%>%select(State, ABV)%>%group\_by(State)%>%arrange(desc(ABV))%>%summarize(Med.ABV = median(ABV\*100))

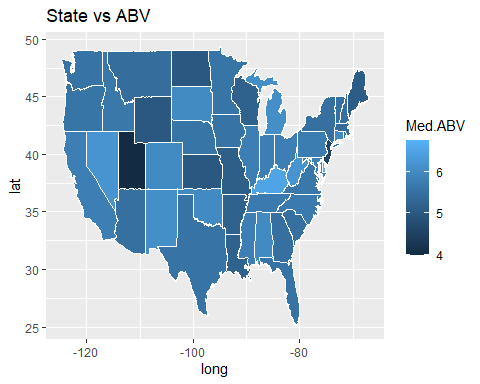
## `summarise()` ungrouping output (override with `.groups` argument)

#Creating a Map to see the spread across the states  
MergeBrewup = left\_join(Beer\_ABV,lookup, by = "State")  
us <- map\_data("state")  
arr <- MergeBrewup %>%add\_rownames("region")%>%mutate(region=tolower(State\_Name))  
gg <- ggplot()  
gg <- gg + geom\_map(data=us, map=us,  
 aes(x=long, y=lat, map\_id=region),  
 fill="#ffffff", color="#ffffff", size=0.15)+expand\_limits(x = us$long, y = us$lat)

## Warning: Ignoring unknown aesthetics: x, y

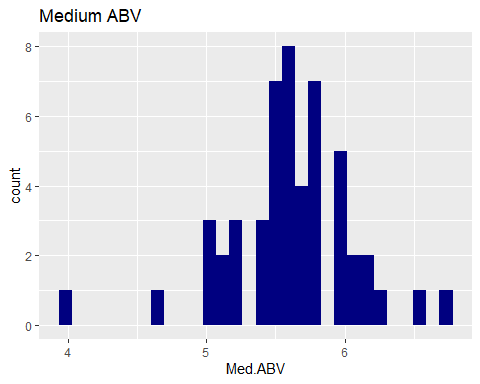
gg <- gg + geom\_map(data=arr, map=us,  
 aes(fill=Med.ABV, map\_id=arr$region),  
 color="#ffffff", size=0.15)+ggtitle("State vs ABV")  
gg

## Warning: Use of `arr$region` is discouraged. Use `region` instead.



#Histogram to analyze the data  
Beer\_ABV%>%ggplot(aes(x=Med.ABV))+geom\_histogram(fill = "Navy Blue")+ggtitle("Medium ABV")

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



#Summary of Medium IBU  
summary(Beer\_ABV$Med.ABV)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 4.000 5.475 5.600 5.611 5.800 6.750

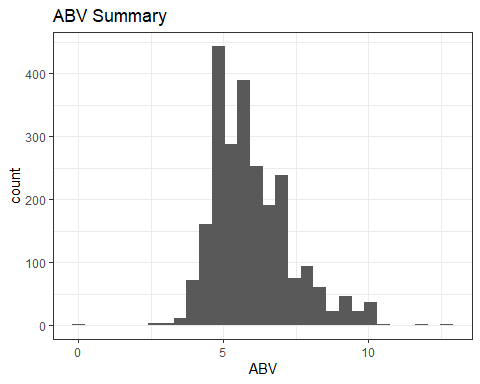
#Comment on the summary statistics and distribution of the ABV variable.  
summary(Beer\_Merg)

## Beer\_Name Beer\_ID ABV IBU   
## Length:2410 Min. : 1.0 Min. :0.00100 Min. : 4.00   
## Class :character 1st Qu.: 808.2 1st Qu.:0.05000 1st Qu.: 22.00   
## Mode :character Median :1453.5 Median :0.05600 Median : 35.00   
## Mean :1431.1 Mean :0.05972 Mean : 41.32   
## 3rd Qu.:2075.8 3rd Qu.:0.06700 3rd Qu.: 60.00   
## Max. :2692.0 Max. :0.12800 Max. :138.00   
## Brew\_ID Style Ounces Brew\_Name   
## Min. : 1.0 Length:2410 Min. : 8.40 Length:2410   
## 1st Qu.: 94.0 Class :character 1st Qu.:12.00 Class :character   
## Median :206.0 Mode :character Median :12.00 Mode :character   
## Mean :232.7 Mean :13.59   
## 3rd Qu.:367.0 3rd Qu.:16.00   
## Max. :558.0 Max. :32.00   
## City State   
## Length:2410 Length:2410   
## Class :character Class :character   
## Mode :character Mode :character   
##   
##   
##

#Delaware is the state with maximum ABV of 6.75.Kentucky is the next largest state with maximum ABV of 6.5.Utah and New Jersey are the least states with ABV of 4 and 4.6 respectively.Summary of ABV looks normally distributed and with this dataset we have no evidence to sugest the data is not normally distributed.

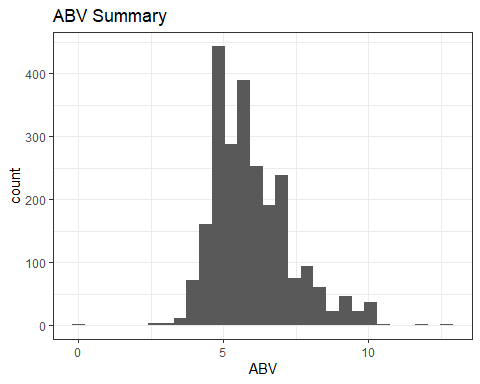
#6.Comment on the summary statistics and distribution of the ABV variable  
Beer\_Merg%>%ggplot(aes(x=ABV\*100))+geom\_histogram()+theme\_bw()+ggtitle("ABV Summary")+xlab("ABV")

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

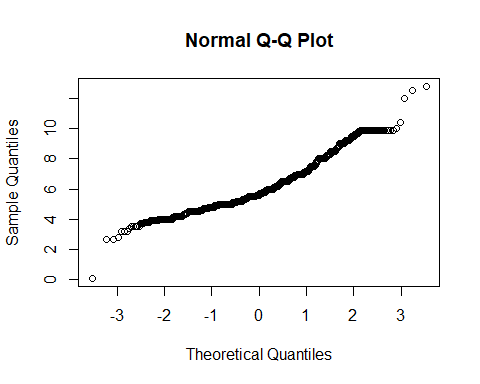


#Histogram  
Beer\_Merg%>%ggplot(aes(x=ABV\*100))+theme\_bw()+ggtitle("ABV Summary")+xlab("ABV")+geom\_histogram()

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



#QQ Plot  
qqnorm(Beer\_Merg$ABV\*100)

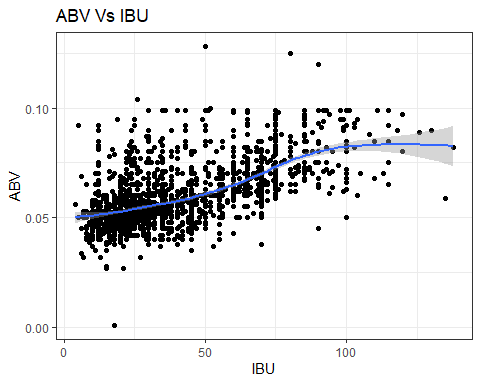


#Summary Stats of ABV  
summary(Beer\_Merg$ABV\*100)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 0.100 5.000 5.600 5.972 6.700 12.800

#The histogram shows that ABV data is normally distributed.The QQ plot shows that the data has constant variance.The mean ABV is 7.972 and the median ABV is 5.600.The min ABV is .1 and Max ABV is 12.8   
  
#7.Is there an apparent relationship between the bitterness of the beer and its alcoholic content? Draw a scatter plot. Make your best judgment of a relationship and EXPLAIN your answer.  
Beer\_Merg%>%ggplot(aes(x=IBU,y=(ABV)))+geom\_point()+geom\_smooth()+theme\_bw()+ylab("ABV")+xlab("IBU")+ggtitle("ABV Vs IBU")

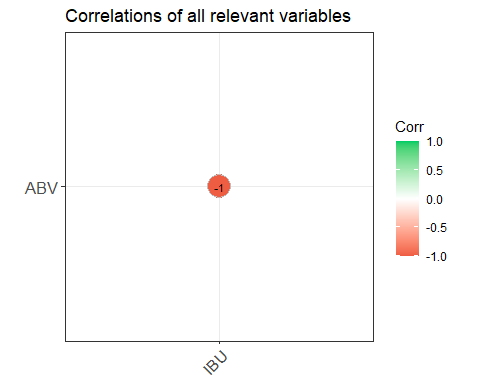
## `geom\_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'



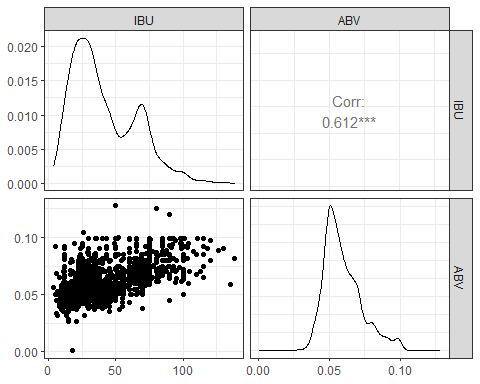
#There is a positive linear relationship with ABV and IBU.For every 1 unit increase in ABV there is an increase in the IBU  
  
#Correltion Function  
corr <- cor(Beer\_Merg%>%select(ABV,IBU), use = "complete.obs")  
corr

## ABV IBU  
## ABV 1.0000000 0.6118091  
## IBU 0.6118091 1.0000000

corr <- round(cor(corr), 2)  
ggcorrplot(corr, type = "lower",  
 lab = TRUE, lab\_size = 3, method = "circle",  
 colors = c("tomato2", "white", "springgreen3"),  
 title = "Correlations of all relevant variables",  
 ggtheme = theme\_bw())



#The IBU and ABV is highly negatively correlated which is helpful in predicting the future observations or any missing observations.  
  
#Running the ggpairs to see the relationship between IBU and ABV  
Beer\_Merg %>%  
select(IBU, ABV) %>%  
ggpairs()+theme\_bw()



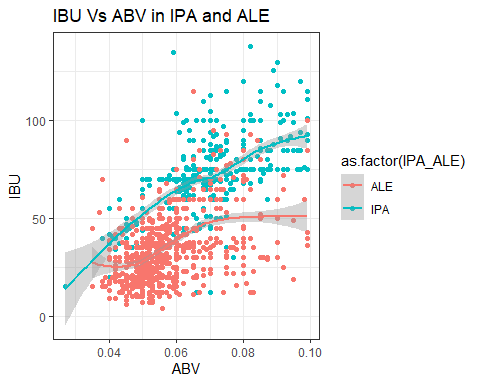
# We can see the data is slightly skewed but using the Central limit theroem with the provided dataset there is not enough evidence to suggest IBU and ABV is not normally distributed.The scatter plot shows even spread which does not provide enough evidence to suggest that the data is does not have a constant variance.

#8.Budweiser would also like to investigate the difference with respect to IBU and ABV between IPAs (India Pale Ales) and other types of Ale (any beer with “Ale” in its name other than IPA). You decide to use KNN classification to investigate this relationship. Provide statistical evidence one way or the other. You can of course assume your audience is comfortable with percentages … KNN is very easy to understand conceptually.  
grepl("IPA", Beer\_Merg$Style)

## [1] FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [13] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [25] FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE FALSE FALSE FALSE  
## [37] FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [49] TRUE FALSE TRUE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [61] TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [73] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE  
## [85] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE  
## [97] TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [109] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE  
## [121] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [133] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE  
## [145] FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE  
## [157] FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [169] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE  
## [181] TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE  
## [193] TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [205] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE TRUE  
## [217] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE FALSE FALSE TRUE  
## [229] FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [241] TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE  
## [253] TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [265] TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [277] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [289] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE  
## [301] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [313] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [325] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE  
## [337] FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [349] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE  
## [361] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [373] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE  
## [385] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE  
## [397] FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [409] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE  
## [421] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE  
## [433] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [445] FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [457] TRUE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [469] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE  
## [481] FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [493] FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [505] TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE  
## [517] FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE  
## [529] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE  
## [541] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE TRUE  
## [553] FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [565] FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [577] FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [589] FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE  
## [601] FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE TRUE TRUE TRUE FALSE  
## [613] FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [625] FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [637] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [649] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [661] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [673] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [685] TRUE TRUE FALSE FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE  
## [697] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE  
## [709] TRUE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE  
## [721] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE  
## [733] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [745] TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [757] FALSE TRUE TRUE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE  
## [769] FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE  
## [781] TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE FALSE  
## [793] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [805] TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE FALSE  
## [817] FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE TRUE TRUE  
## [829] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [841] TRUE FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE  
## [853] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [865] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE  
## [877] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE  
## [889] FALSE TRUE FALSE FALSE TRUE TRUE TRUE TRUE FALSE TRUE TRUE FALSE  
## [901] FALSE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE  
## [913] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [925] TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [937] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [949] FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE  
## [961] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [973] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [985] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [997] TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE  
## [1009] FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE  
## [1021] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1033] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE FALSE FALSE FALSE  
## [1045] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1057] TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE FALSE  
## [1069] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1081] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE  
## [1093] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1105] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE  
## [1117] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1129] TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1141] TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [1153] FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE  
## [1165] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1177] TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1189] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE  
## [1201] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1213] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1225] FALSE TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1237] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1249] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [1261] FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE  
## [1273] FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE  
## [1285] TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1297] FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1309] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1321] TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE  
## [1333] FALSE FALSE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE TRUE TRUE  
## [1345] TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE  
## [1357] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1369] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [1381] FALSE TRUE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1393] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1405] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1417] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1429] FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1441] FALSE TRUE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1453] FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1465] FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE TRUE FALSE  
## [1477] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1489] TRUE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1501] TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [1513] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1525] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1537] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1549] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE  
## [1561] FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE  
## [1573] FALSE FALSE FALSE TRUE TRUE TRUE FALSE FALSE TRUE FALSE TRUE FALSE  
## [1585] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1597] TRUE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE  
## [1609] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1621] FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [1633] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1645] TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1657] FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE  
## [1669] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1681] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1693] FALSE FALSE TRUE TRUE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE  
## [1705] FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1717] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [1729] FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1741] FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE  
## [1753] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1765] TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1777] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1789] TRUE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1801] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [1813] FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE  
## [1825] FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1837] FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1849] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [1861] FALSE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [1873] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE TRUE  
## [1885] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE TRUE  
## [1897] FALSE FALSE TRUE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE  
## [1909] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1921] TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1933] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE  
## [1945] TRUE TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [1957] FALSE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [1969] TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE  
## [1981] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [1993] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [2005] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2017] FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [2029] FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2041] TRUE TRUE TRUE TRUE TRUE FALSE FALSE TRUE FALSE FALSE FALSE FALSE  
## [2053] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [2065] TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE TRUE  
## [2077] TRUE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE FALSE FALSE  
## [2089] FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE  
## [2101] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [2113] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE FALSE FALSE  
## [2125] TRUE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE  
## [2137] FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE TRUE TRUE FALSE TRUE  
## [2149] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE  
## [2161] TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2173] FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [2185] TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE TRUE FALSE FALSE FALSE  
## [2197] FALSE TRUE FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE TRUE TRUE  
## [2209] FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE TRUE FALSE TRUE  
## [2221] FALSE TRUE FALSE TRUE TRUE FALSE TRUE FALSE TRUE FALSE FALSE FALSE  
## [2233] TRUE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2245] TRUE FALSE FALSE TRUE TRUE FALSE FALSE FALSE FALSE TRUE TRUE FALSE  
## [2257] FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2269] FALSE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE  
## [2281] FALSE FALSE FALSE FALSE TRUE FALSE TRUE FALSE FALSE FALSE FALSE FALSE  
## [2293] FALSE FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE TRUE TRUE FALSE  
## [2305] TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2317] TRUE FALSE TRUE TRUE TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE  
## [2329] TRUE FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE  
## [2341] FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [2353] FALSE TRUE FALSE FALSE TRUE FALSE FALSE FALSE TRUE FALSE FALSE FALSE  
## [2365] TRUE TRUE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  
## [2377] FALSE FALSE FALSE TRUE FALSE FALSE TRUE FALSE TRUE FALSE TRUE FALSE  
## [2389] TRUE FALSE TRUE FALSE FALSE TRUE FALSE TRUE TRUE FALSE TRUE TRUE  
## [2401] FALSE FALSE FALSE FALSE FALSE TRUE FALSE FALSE FALSE FALSE

IPA<-Beer\_Merg[grepl("IPA",Beer\_Merg$Style),]  
IPA\_updated<-IPA%>%mutate(IPA\_ALE="IPA")  
Ale<-Beer\_Merg[grepl("Ale",Beer\_Merg$Style),]  
Ale1<-Ale[!grepl("IPA",Ale$Style),]  
Ale\_updated<-Ale1%>%mutate(IPA\_ALE="ALE")  
IPA\_ALE = union(IPA\_updated,Ale\_updated)  
  
#Relationship between IBU and ABV for ALE and IPA using scatter plot  
IPA\_ALE %>% ggplot(aes(x = ABV, y = IBU, color = as.factor(IPA\_ALE))) + geom\_point()+geom\_smooth()+theme\_bw()+ggtitle("IBU Vs ABV in IPA and ALE")

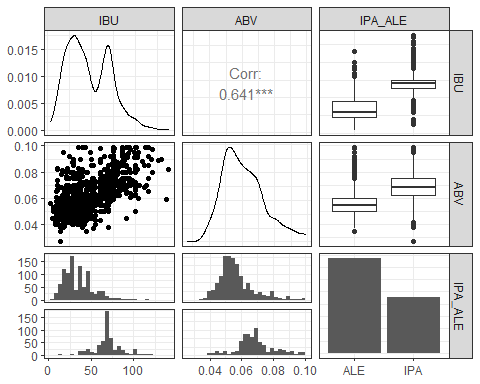
## `geom\_smooth()` using method = 'loess' and formula 'y ~ x'



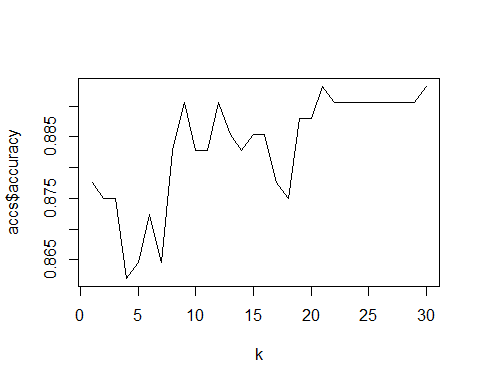
IPA\_ALE %>% select (IBU,ABV, IPA\_ALE) %>% ggpairs()+theme\_bw()

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

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



#knn  
#Split the dataset into train and test datset.75% would remain as training dataet and 25% will remain as test dataset.  
set.seed(32)   
library(caret)  
iterations = 100  
accs = data.frame(accuracy = numeric(30), k = numeric(30))  
  
splitPerc = .75  
trainIndicesknn = sample(seq(1:length(IPA\_ALE$IPA\_ALE)),round(splitPerc \* length(IPA\_ALE$IPA\_ALE)))  
trainknn = IPA\_ALE[trainIndicesknn,]  
testknn = IPA\_ALE[-trainIndicesknn,]  
  
#Running the KNN classifier  
for(i in 1:30)  
{  
model = class::knn(trainknn[,c(3,4)],testknn[,c(3,4)],(trainknn$IPA\_ALE),k=i,prob=TRUE)  
table(model,testknn$IPA\_ALE)  
CM = confusionMatrix(table(model,testknn$IPA\_ALE))  
 accs$accuracy[i] = CM$overall[1]  
 accs$k[i] = i  
}  
#Plot to find the mean k and mean accuracy  
plot(accs$k,accs$accuracy, type = "l", xlab = "k")



MeanAcc = colMeans(accs)  
MeanAcc

## accuracy k   
## 0.8833333 15.5000000

#Display the Confusion Matrix  
model = class::knn(trainknn[,c(3,4)],testknn[,c(3,4)],(trainknn$IPA\_ALE),k=22,prob=TRUE)  
table(model,testknn$IPA\_ALE)

##   
## model ALE IPA  
## ALE 228 20  
## IPA 22 114

CM = confusionMatrix(table(model,testknn$IPA\_ALE))  
  
CM

## Confusion Matrix and Statistics  
##   
##   
## model ALE IPA  
## ALE 228 20  
## IPA 22 114  
##   
## Accuracy : 0.8906   
## 95% CI : (0.855, 0.92)  
## No Information Rate : 0.651   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.7601   
##   
## Mcnemar's Test P-Value : 0.8774   
##   
## Sensitivity : 0.9120   
## Specificity : 0.8507   
## Pos Pred Value : 0.9194   
## Neg Pred Value : 0.8382   
## Prevalence : 0.6510   
## Detection Rate : 0.5938   
## Detection Prevalence : 0.6458   
## Balanced Accuracy : 0.8814   
##   
## 'Positive' Class : ALE   
##

#Extrapolated a new variable IPA\_ALE which segregates all beers with style IPA to IPA and the rest of the Ale s to ALE.Initial Analysis show that the IBU and ABV between IPA and ALE is linearly correlated.There is an interaction between the IPA and ALE beers.IPA seems to have a greater IBU and ABV compared to ALE.The assumption of normal distribution and constant variance is satisfied.The observations is considered to be independent   
  
#The K-NN classification is used to predict a beer type based on the IBU and ABV.Our strategy is to find the closest applicants and let the majority rule on whether the beer type will be ABV or IBV.The ALE Vs IPA data has been plotted on the scatter plot.The best value of k = 22 and accuracy is .8906.The average accuracy is 0.8833 and k=15.5  
  
#NB  
  
iterations = 100  
accsNB = data.frame(accuracy = numeric(30), k = numeric(30))  
#Split the dataset into train and test datset.75% would remain as training dataet and 25% will remain as test dataset.  
splitPerc = .75  
  
trainIndicesknb = sample(seq(1:length(IPA\_ALE$IPA\_ALE)),round(splitPerc \* length(IPA\_ALE$IPA\_ALE)))  
trainknb = IPA\_ALE[trainIndicesknb,]  
testknb = IPA\_ALE[-trainIndicesknb,]  
  
#Running the NB classifier  
for(i in 1:30)  
{  
model = naiveBayes(trainknb[,c(3,4)],as.factor(trainknb$IPA\_ALE))  
table(predict(model,testknb[,c(3,4)]),as.factor(testknb$IPA\_ALE))  
CM = confusionMatrix(table(predict(model,testknb[,c(3,4)]),as.factor(testknb$IPA\_ALE)))  
 accsNB$accuracy[i] = CM$overall[1]  
 accsNB$k[i] = i  
}  
MeanAccNB = colMeans(accsNB)  
#Display the mean accuracy  
MeanAccNB

## accuracy k   
## 0.8463542 15.5000000

model = naiveBayes(trainknb[,c(3,4)],as.factor(trainknb$IPA\_ALE))  
table(predict(model,testknb[,c(3,4)]),as.factor(testknb$IPA\_ALE))

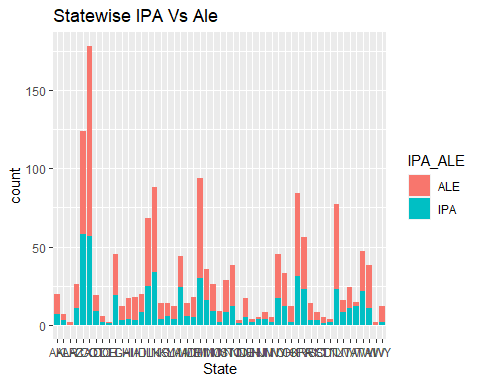
##   
## ALE IPA  
## ALE 215 28  
## IPA 31 110

CM = confusionMatrix(table(predict(model,testknb[,c(3,4)]),as.factor(testknb$IPA\_ALE)))  
CM

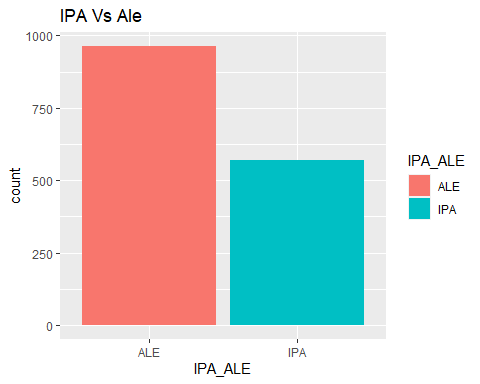
## Confusion Matrix and Statistics  
##   
##   
## ALE IPA  
## ALE 215 28  
## IPA 31 110  
##   
## Accuracy : 0.8464   
## 95% CI : (0.8063, 0.8809)  
## No Information Rate : 0.6406   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.6679   
##   
## Mcnemar's Test P-Value : 0.7946   
##   
## Sensitivity : 0.8740   
## Specificity : 0.7971   
## Pos Pred Value : 0.8848   
## Neg Pred Value : 0.7801   
## Prevalence : 0.6406   
## Detection Rate : 0.5599   
## Detection Prevalence : 0.6328   
## Balanced Accuracy : 0.8355   
##   
## 'Positive' Class : ALE   
##

#Extrapolated a new variable IPA\_ALE which segregates all beers with style IPA to IPA and the rest of the Ale s to ALE.Initial Analysis show that the IBU and ABV between IPA and ALE is linearly correlated.There is an interaction between the IPA and ALE beers.IPA seems to have a greater IBU and ABV compared to ALE.The assumption of normal distribution and constant variance is satisfied.The observations is considered to be independent   
  
#The NB classification is used to predict a beer type based on the IBU and ABV.Our strategy is to predict the beer type based on the ABV or IBV.The ALE Vs IPA data has been plotted on the scatter plot.The accuracy is .8828.Positive Class is ALE

#Summary  
IPA\_ALE%>%ggplot(aes(x=State,fill=IPA\_ALE))+geom\_bar(position = "stack")+ggtitle("Statewise IPA Vs Ale")



IPA\_ALE%>%ggplot(aes(x=IPA\_ALE,fill=IPA\_ALE))+geom\_bar()+ggtitle("IPA Vs Ale")



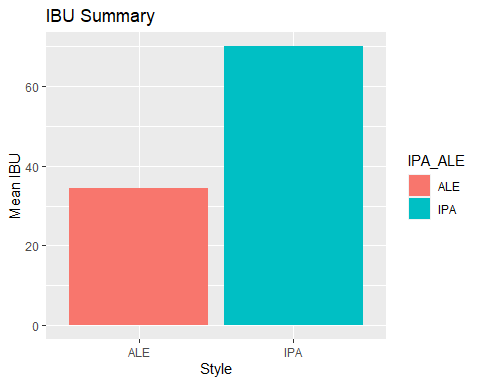
IPA\_ALE\_SUm=IPA\_ALE%>%group\_by(IPA\_ALE)%>%select(IBU,ABV,IPA\_ALE)%>%summarize(Mean.IBU=mean(IBU),Mean.ABV=mean(ABV))

## `summarise()` ungrouping output (override with `.groups` argument)

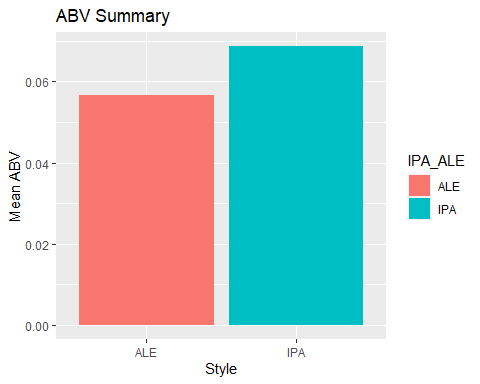
IPA\_ALE\_SUm

## # A tibble: 2 x 3  
## IPA\_ALE Mean.IBU Mean.ABV  
## <chr> <dbl> <dbl>  
## 1 ALE 34.5 0.0568  
## 2 IPA 70.1 0.0687

IPA\_ALE\_SUm%>%ggplot(aes(x=IPA\_ALE,y=Mean.IBU,fill=IPA\_ALE))+geom\_col()+ggtitle("IBU Summary")+ylab("Mean IBU")+xlab("Style")



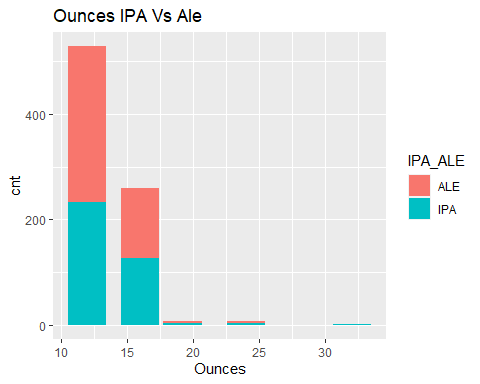
IPA\_ALE\_SUm%>%ggplot(aes(x=IPA\_ALE,y=Mean.ABV,fill=IPA\_ALE))+geom\_col()+ggtitle("ABV Summary")+ylab("Mean ABV")+xlab("Style")



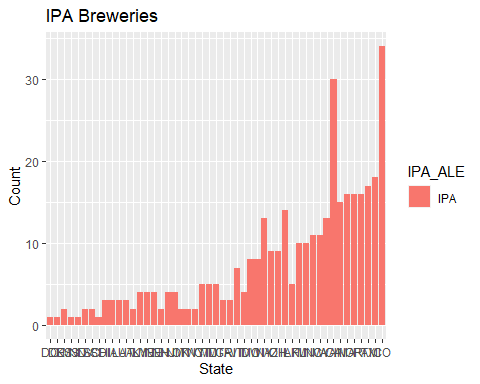
Brewsummary=IPA\_ALE%>%group\_by(State,IPA\_ALE,Ounces)%>%summarize(cnt = length(unique(Brew\_ID)))%>%arrange(cnt,State,IPA\_ALE,Ounces)

## `summarise()` regrouping output by 'State', 'IPA\_ALE' (override with `.groups` argument)

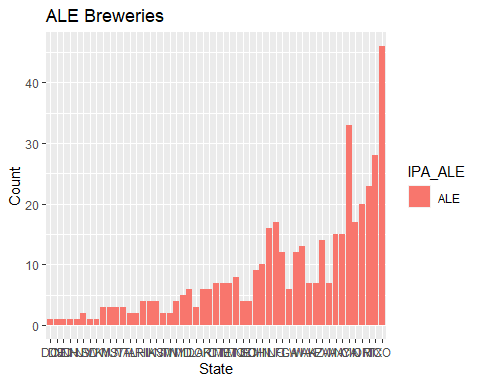
Brewsummary%>%ggplot(aes(x=Ounces,y=cnt,fill=IPA\_ALE))+geom\_col()+ggtitle("Ounces IPA Vs Ale")



Brewsummary%>%filter(IPA\_ALE=='IPA')%>%arrange(desc(cnt,State))%>%ggplot(aes(x=reorder(factor(State),cnt),y=cnt,fill=IPA\_ALE))+geom\_col()+ggtitle("IPA Breweries")+xlab("State")+ylab("Count")



Brewsummary%>%filter(IPA\_ALE=='ALE')%>%arrange(desc(cnt,State))%>%ggplot(aes(x=reorder(factor(State),cnt),y=cnt,fill=IPA\_ALE))+geom\_col()+ggtitle("ALE Breweries")+xlab("State")+ylab("Count")



#ALE is brewed more than IPA in major breweries across the states of US.The volume of ALE brewed is more than IPA.12 ounce is brewed more than the 24 and 32 ounces. ALE is sold more than IPA in 12 ounces and IPA constitutes more than ALE in 16 ounces.Colorado & California are the major breweries that brew ALE and IPA across US.DC and Delaware are the states with the least breweries for ALE and IPA.The Mean IBU of IPA is 70 and ALE is 34.The Mean ABV of IPA is 6.8 and ALE is 5.6  
  
#As this is an observational study there is no cause and effect but association to the conclusion.We cannot conclude that IPA is more sought out beer than ALE from the mean IBU and ABV values (pvalue <0.0005 from the knn and NB test).In conclusion the association between IBU and ABV can be generalized to identify IPA and ALE beer styles in the selected area of study in the US but cannot be generalized to other countries across the world.Though the production of ALE is more than IPA, we have enough evidence to suggest that the market for IPA is increasing and there is potential for business in the states of US   
  
write.csv(Beer\_Merg, "C:/Sowmya/SMU/04\_Doing Data Science/Unit-8 & Unit-9/Final Submission/Beer\_Merg.CSV")