FLS\_8\_Simi Rmd

Simi

2023-07-01

## 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:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



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

Beers = read.csv(file.choose(),header = TRUE)  
  
Breweries = read.csv(file.choose(),header = TRUE)  
  
#Question 1. How many breweries are present in each state?  
# Load the dplyr package  
library(dplyr)

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

##   
## Attaching package: 'dplyr'

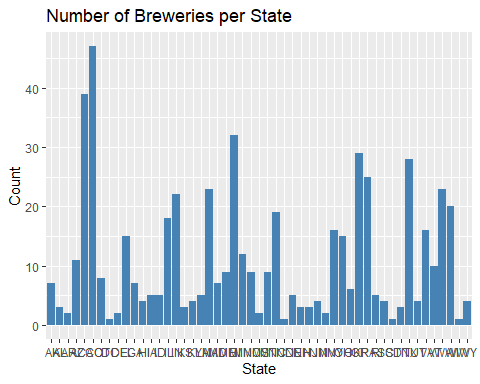
## The following objects are masked from 'package:stats':  
##   
## filter, lag

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

library(ggplot2)

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

# Assuming your dataset is named "breweries", use group\_by and summarise to count breweries per state  
Breweries\_per\_state <- Breweries %>%  
 group\_by(State) %>%  
 dplyr::summarise(Count = n())  
  
# Sort the breweries in descending order  
Breweries\_per\_state <- Breweries\_per\_state %>%  
 arrange(desc(Count))  
  
# Create a bar plot using ggplot  
ggplot(data = Breweries\_per\_state, aes(x = State, y = Count)) +  
 geom\_bar(stat = "identity", fill = "steelblue") +  
 labs(title = "Number of Breweries per State",  
 x = "State",  
 y = "Count")



#Qn 6. Comment on the summary statistics and distribution of the ABV variable.  
  
BeersData <- Breweries %>%  
 left\_join(Beers, by = c("Brew\_ID" = "Brewery\_id"))  
summary(BeersData$ABV)

## Min. 1st Qu. Median Mean 3rd Qu. Max. NA's   
## 0.00100 0.05000 0.05600 0.05977 0.06700 0.12800 62

#summary(filtered\_data$ABV)  
sd(BeersData$ABV)

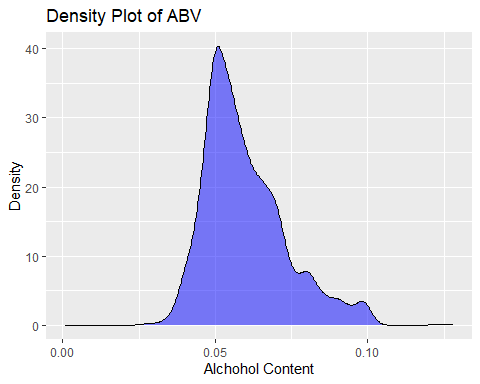
## [1] NA

# Create mode() function to calculate mode  
mode <- function(x, na.rm = FALSE) {  
   
 if(na.rm){ #if na.rm is TRUE, remove NA values from input x  
 x = x[!is.na(x)]  
 }  
   
 val <- unique(x)  
 return(val[which.max(tabulate(match(x, val)))])  
}  
  
mode(BeersData$ABV)

## [1] 0.05

#density plot  
ggplot(BeersData, aes(x = ABV)) +  
 geom\_density(color = "black", fill = "blue", alpha = 0.5) +  
 labs(x = "Alchohol Content", y = "Density", title = "Density Plot of ABV")

## Warning: Removed 62 rows containing non-finite values (`stat\_density()`).



#Question 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.  
  
# Filtering out zero values for alcohol\_content and bitterness  
library(plotly)

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

##   
## Attaching package: 'plotly'

## The following object is masked from 'package:ggplot2':  
##   
## last\_plot

## The following object is masked from 'package:stats':  
##   
## filter

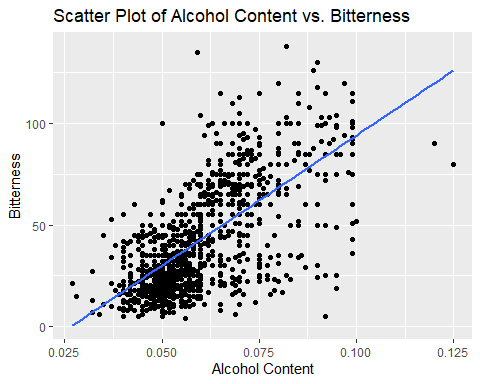
## The following object is masked from 'package:graphics':  
##   
## layout

filtered\_data <- BeersData[BeersData$ABV != 0 & BeersData$IBU != 0, ]  
  
ggplot(filtered\_data, aes(x = ABV, y = IBU)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(x = "Alcohol Content", y = "Bitterness", title = "Scatter Plot of Alcohol Content vs. Bitterness")

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

## Warning: Removed 1005 rows containing non-finite values (`stat\_smooth()`).

## Warning: Removed 1005 rows containing missing values (`geom\_point()`).



scatter\_plot <- ggplot(filtered\_data, aes(x = ABV, y = IBU, text = Name.y)) +  
 geom\_point() + geom\_smooth(method = "lm", se = FALSE) +  
 labs(x = "Alcohol Content", y = "Bitterness", title = "Scatter Plot of Alcohol Content vs. Bitterness (excluding zeros)")  
  
interactive\_plot <- ggplotly(scatter\_plot)

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

## Warning: Removed 1005 rows containing non-finite values (`stat\_smooth()`).

interactive\_plot

cor(filtered\_data$ABV, filtered\_data$IBU)

## [1] NA

# Use knn 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)  
library(e1071)

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

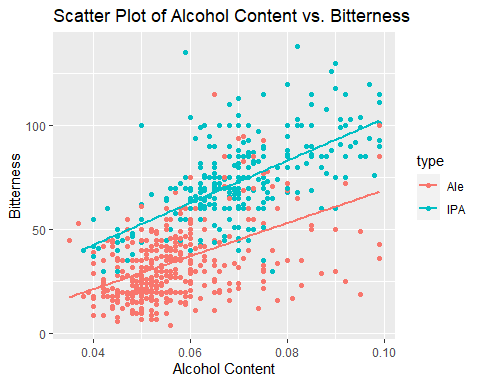
library(class)  
library(dplyr)  
library(caret)

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

## Loading required package: lattice

filtered\_data <- filtered\_data %>% mutate(type = ifelse (grepl('IPA', Style), "IPA", "Other"))  
filtered\_data$type = replace(filtered\_data$type , grepl('Ale', filtered\_data$Style), "Ale")  
IPA\_Ale <- filtered\_data[filtered\_data$type != "Other", ]  
  
  
ggplot(IPA\_Ale, aes(x = ABV, y = IBU, color = type)) +  
 geom\_point() +  
 geom\_smooth(method = "lm", se = FALSE) +  
 labs(x = "Alcohol Content", y = "Bitterness", title = "Scatter Plot of Alcohol Content vs. Bitterness")

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



predictors <- IPA\_Ale %>% select(IBU, ABV)  
response <- IPA\_Ale$type  
  
k <- 5 # Number of nearest neighbors  
train\_indices <- sample(1:nrow(IPA\_Ale), nrow(IPA\_Ale) \* 0.7)  
train\_data <- IPA\_Ale[train\_indices, ]  
test\_data <- IPA\_Ale[-train\_indices, ]  
  
knn\_result <- knn(train = train\_data %>% select(IBU, ABV), test = test\_data %>% select(IBU, ABV), cl = train\_data$type, k = k)  
accuracy <- mean(knn\_result == test\_data$type)  
print(paste("Accuracy:", accuracy))

## [1] "Accuracy: 0.855633802816901"

tbl\_knn = table(test\_data$type, knn\_result)  
CM = confusionMatrix(tbl\_knn)  
print(CM)

## Confusion Matrix and Statistics  
##   
## knn\_result  
## Ale IPA  
## Ale 146 21  
## IPA 20 97  
##   
## Accuracy : 0.8556   
## 95% CI : (0.8093, 0.8944)  
## No Information Rate : 0.5845   
## P-Value [Acc > NIR] : <2e-16   
##   
## Kappa : 0.7024   
##   
## Mcnemar's Test P-Value : 1   
##   
## Sensitivity : 0.8795   
## Specificity : 0.8220   
## Pos Pred Value : 0.8743   
## Neg Pred Value : 0.8291   
## Prevalence : 0.5845   
## Detection Rate : 0.5141   
## Detection Prevalence : 0.5880   
## Balanced Accuracy : 0.8508   
##   
## 'Positive' Class : Ale   
##