Front Page

**DECLARATION**

**CERTIFICATE**

**ACKNOWLEDGEMENT**

**INDEX**

**ABSTRACT**

Our four-person team’s project on “Chocolate analysis” was completed as a part of the Probability and Statistics (BMAT202P) course at Vellore Institute of Technology, Chennai.

This report is based on the reviews of over 2000 chocolates, and their relation with factors such as ingredients, country of origin, etc. It applies statistics to make conclusions regarding the dataset. The project’s focus is on “Analysis of chocolate ratings using R.”

This is a case study on chocolates wherein questions regarding the dependence of rating on origin of ingredients, number of ingredients, etc are explored and answered. Questions about which country produces objectively better chocolates are difficult to answer, hence hypothesis testing is used for the same.

**INTRODUCTION**

Chocolates have become one of the most popular confectionaries in the world of today. They form the basic ingredient in many pastries and cakes. Chocolates can also be used as hot and cold beverages. Each manufacturer uses a special formula which combines the different varieties of the cocoa sweets to develop exclusive chocolates and try to make the exotic treat. Gifts of chocolates moulded to different shapes have become traditional on certain festivals and occasions.

Chocolates are made from the seeds of cocoa trees. Spanish mythologies consider that these trees were grown in the garden of Paradise and believe that the chocolate drink was Divine. The cocoa tree is a tropical plant, sometimes living and producing seeds for more than 200 years. Chocolates are made from the seeds of these trees. There are many varieties cultivated today and this farming is highly profitable.

Chocolate making is a highly commercialized and money-making programme. In modern factories, tons of bitter cocoa beans are turned into one of the world’s favourite confectionaries. These days, chocolates are available in a variety of flavours such as mint, coffee, orange, strawberry etc. Also, today the chocolates can contain ingredients such as peanuts, different types of walnuts, dry fruits, caramels, crisped rice etc.

Usually, the chocolates can be categorized into one the following groups.

1. Bitter

2. Bitter sweets

3. Unsweetened

4. Dark sweetened

5. Milk chocolates

6. Cocoa powder

7. Cocoa sauce/syrup

**PROJECT OBJECTIVES**

1) Finding summary statistics and finding the data types of each column to better understand the dataset

2) Finding the top 10 chocolate producers, considering the ingredients used

3) Plotting changes in ratings over time and distribution of ratings to analyse the given dataset

4) Finding the top 10 company locations, bean origin, review dates, and chocolate tastes from the dataset

5) Finding how many chocolates contain a particular ingredient and distinct values for all non-numeric columns

6) Finding the most common first taste and comparing the reviews of these chocolates with average review of all chocolates

7) Modelling the chocolate rating in terms of cocoa percent and count of ingredients

8) Testing whether top 10 companies producing chocolates is representative of the entire population

9) Testing whether <company A> produces objectively better chocolates than <company B>

**DATASET**

Dataset: <https://www.kaggle.com/datasets/soroushghaderi/chocolate-bar-2020>

**Context**

Chocolate is one of the most popular candies in the world. Each year, residents of the United States collectively eat more than 2.8 billion pounds. However, not all chocolate bars are created equal! This dataset contains expert ratings of over 1,700 individual chocolate bars, along with information on their regional origin, percentage of cocoa, the variety of chocolate bean used, and where the beans were grown.

**Flavors of Cacao Rating System:**

Rating Scale

4.0 - 5.0 = Outstanding  
3.5 - 3.9 = Highly Recommended  
3.0 - 3.49 = Recommended  
2.0 - 2.9 = Disappointing  
1.0 - 1.9 = Unpleasant

\*Not all the bars in each range are considered equal, so to show variance from bars in the same range I have assigned .25, .50 or .75.

Each chocolate is evaluated from a combination of both objective qualities and subjective interpretation. A rating here only represents an experience with one bar from one batch. Batch numbers, vintages, and review dates are included in the database when known. I would recommend people to try all the chocolate on the database regardless of the rating and experience for themselves.

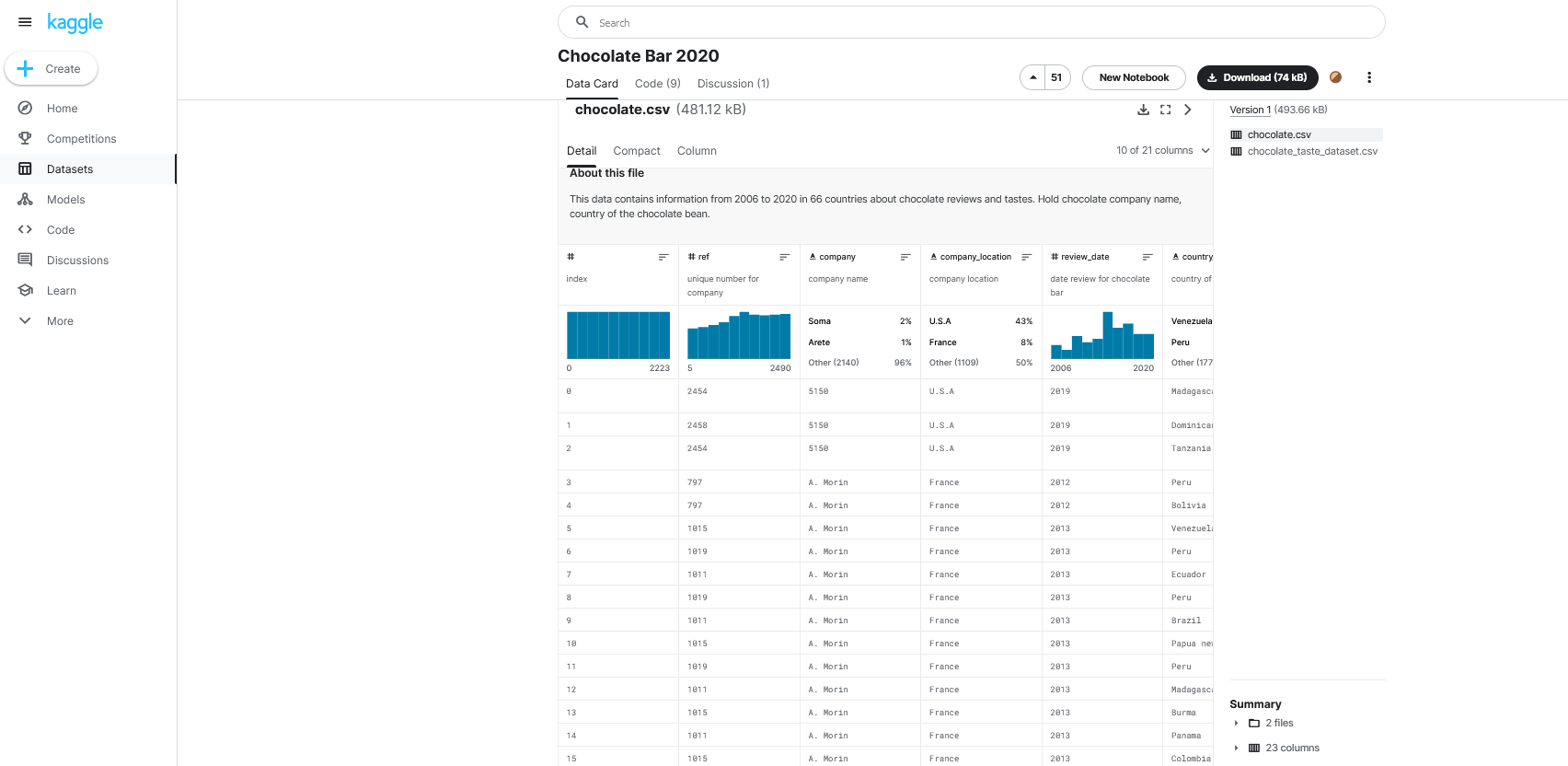
The database is narrowly focused on plain dark chocolate to appreciate the flavors of the cacao when made into chocolate. The ratings do not reflect health benefits, social missions, or organic status.

The flavor is the most important component of the Flavors of Cacao ratings. Diversity, balance, intensity, and purity of flavors are all considered. A straight forward single note chocolate can rate as high as a complex flavor profile that changes throughout. Genetics, terroir, post-harvest techniques, processing, and storage can all be discussed when considering the flavor component.

Texture has a great impact on the overall experience and it is also possible for texture related issues to impact flavor. It is a good way to evaluate the makers' vision, attention to detail, and level of proficiency.

Aftermelt is the experience after the chocolate has melted. Higher quality chocolate will linger and be long-lasting and enjoyable. Since the after melt is the last impression you get from the chocolate, it receives equal importance in the overall rating.

Overall Opinion is really where the ratings reflect a subjective opinion. Ideally, it is my evaluation of whether or not the components above worked together and opinion on the flavor development, character, and style. It is also here where each chocolate can usually be summarized by the most prominent impressions that you would remember about each chocolate



**DATA ANALYSIS**

1. **Data Cleaning**

Data\_Cleaning.R

##Chocolate Analysis  
library(readr) #to read the csv file

## Warning: package 'readr' was built under R version 4.2.3

chocolate <- read\_csv("data/raw\_chocolate.csv")

## New names:  
## Rows: 2224 Columns: 21  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (15): company, company\_location, country\_of\_bean\_origin, specific\_bean\_o... dbl  
## (6): ...1, ref, review\_date, cocoa\_percent, rating, counts\_of\_ingredients  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

View(chocolate)  
  
#View column names  
column\_names = colnames(chocolate)  
column\_names

## [1] "...1" "ref"   
## [3] "company" "company\_location"   
## [5] "review\_date" "country\_of\_bean\_origin"   
## [7] "specific\_bean\_origin\_or\_bar\_name" "cocoa\_percent"   
## [9] "rating" "counts\_of\_ingredients"   
## [11] "beans" "cocoa\_butter"   
## [13] "vanilla" "lecithin"   
## [15] "salt" "sugar"   
## [17] "sweetener\_without\_sugar" "first\_taste"   
## [19] "second\_taste" "third\_taste"   
## [21] "fourth\_taste"

#Removing columns not being used  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## 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

chocolate <- select(chocolate, -c(...1,fourth\_taste,specific\_bean\_origin\_or\_bar\_name))  
  
chocolate <- data.frame(chocolate)  
  
#Modifying the columns  
#Changing have/have not to 1/0  
  
chocolate = data.frame(chocolate)  
colnames(chocolate)[9] <- "bean"  
colnames(chocolate)[11] <- "vanila"  
  
#Columns to be changed: 9 to 15  
bool\_columns = colnames(chocolate[9:15])  
bool\_columns

## [1] "bean" "cocoa\_butter"   
## [3] "vanila" "lecithin"   
## [5] "salt" "sugar"   
## [7] "sweetener\_without\_sugar"

for (column in bool\_columns){  
 have\_not <- paste("have\_not\_", column, sep = "")  
 chocolate[column] = replace(chocolate[column], chocolate[column] == have\_not, 0);  
  
 have <- paste("have\_", column, sep = "")  
 chocolate[column] <- replace(chocolate[column], chocolate[column] == have, 1)  
}  
  
colnames(chocolate)[11] <- "vanilla"  
  
#Writing the cleaned data into a new file  
write.csv(chocolate, "data/chocolate.csv")

1. **Exploratory Analysis**

Exploratory\_Analysis.R

#Creating data frame  
library(readr)

## Warning: package 'readr' was built under R version 4.2.3

chocolate <- read\_csv("data/chocolate.csv")

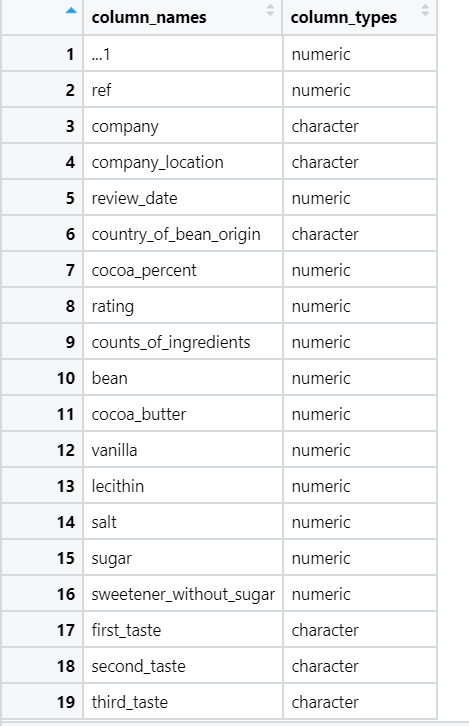
## New names:  
## Rows: 2224 Columns: 19  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (6): company, company\_location, country\_of\_bean\_origin, first\_taste, se... dbl  
## (13): ...1, ref, review\_date, cocoa\_percent, rating, counts\_of\_ingredien...  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

View(chocolate)  
chocolate <- data.frame(chocolate)

#SUMMARY STATISTICS  
  
#Finding class of each column  
library(tibble)

## Warning: package 'tibble' was built under R version 4.2.3

column\_names = colnames(chocolate)  
column\_types = sapply(chocolate, class)  
column\_statistics = as\_tibble(data.frame(column\_names, column\_types))  
View(column\_statistics)

  
  
summary(chocolate)

## ...1 ref company company\_location   
## Min. : 1.0 Min. : 5 Length:2224 Length:2224   
## 1st Qu.: 556.8 1st Qu.: 776 Class :character Class :character   
## Median :1112.5 Median :1381 Mode :character Mode :character   
## Mean :1112.5 Mean :1337   
## 3rd Qu.:1668.2 3rd Qu.:1928   
## Max. :2224.0 Max. :2490   
## review\_date country\_of\_bean\_origin cocoa\_percent rating   
## Min. :2006 Length:2224 Min. : 42.00 Min. :1.000   
## 1st Qu.:2011 Class :character 1st Qu.: 70.00 1st Qu.:3.000   
## Median :2014 Mode :character Median : 70.00 Median :3.250   
## Mean :2014 Mean : 71.49 Mean :3.199   
## 3rd Qu.:2016 3rd Qu.: 74.00 3rd Qu.:3.500   
## Max. :2020 Max. :100.00 Max. :4.000   
## counts\_of\_ingredients bean cocoa\_butter vanilla   
## Min. :1.000 Min. :1 Min. :0.0000 Min. :0.0000   
## 1st Qu.:2.000 1st Qu.:1 1st Qu.:0.0000 1st Qu.:0.0000   
## Median :3.000 Median :1 Median :1.0000 Median :0.0000   
## Mean :3.076 Mean :1 Mean :0.6902 Mean :0.1565   
## 3rd Qu.:4.000 3rd Qu.:1 3rd Qu.:1.0000 3rd Qu.:0.0000   
## Max. :6.000 Max. :1 Max. :1.0000 Max. :1.0000   
## lecithin salt sugar sweetener\_without\_sugar  
## Min. :0.0000 Min. :0.00000 Min. :0.0000 Min. :0.00000   
## 1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:1.0000 1st Qu.:0.00000   
## Median :0.0000 Median :0.00000 Median :1.0000 Median :0.00000   
## Mean :0.2154 Mean :0.01664 Mean :0.9631 Mean :0.03417   
## 3rd Qu.:0.0000 3rd Qu.:0.00000 3rd Qu.:1.0000 3rd Qu.:0.00000   
## Max. :1.0000 Max. :1.00000 Max. :1.0000 Max. :1.00000   
## first\_taste second\_taste third\_taste   
## Length:2224 Length:2224 Length:2224   
## Class :character Class :character Class :character   
## Mode :character Mode :character Mode :character   
##   
##   
##

#Finding summary statistics for numeric columns  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## Attaching package: 'dplyr'

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

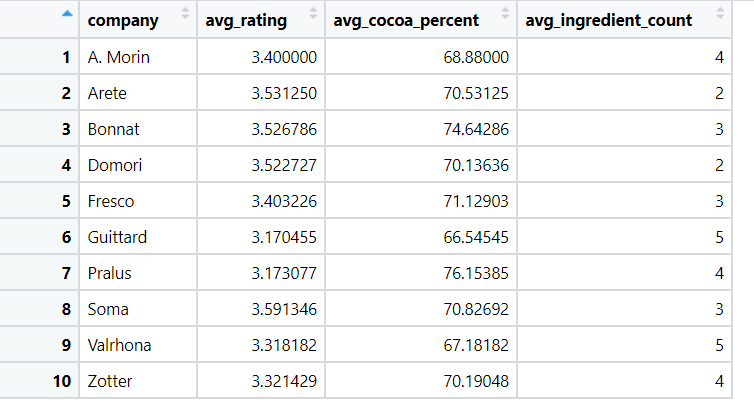
## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

summary(select(chocolate, c(review\_date, cocoa\_percent, rating, counts\_of\_ingredients)))

## review\_date cocoa\_percent rating counts\_of\_ingredients  
## Min. :2006 Min. : 42.00 Min. :1.000 Min. :1.000   
## 1st Qu.:2011 1st Qu.: 70.00 1st Qu.:3.000 1st Qu.:2.000   
## Median :2014 Median : 70.00 Median :3.250 Median :3.000   
## Mean :2014 Mean : 71.49 Mean :3.199 Mean :3.076   
## 3rd Qu.:2016 3rd Qu.: 74.00 3rd Qu.:3.500 3rd Qu.:4.000   
## Max. :2020 Max. :100.00 Max. :4.000 Max. :6.000

#Finding distinct values for non numeric columns  
Distinct\_Columns <- data.frame(matrix(nrow = 0, ncol = 2))  
colnames(Distinct\_Columns) = c("Column", "No. of distinct entries")  
  
index <- 1;  
for (number in 1:19){  
 column <- column\_statistics$column\_names[number]  
 type <- column\_statistics$column\_types[number]  
 if (type == 'character'){  
 unique\_values <- unique(na.omit(chocolate[column])) #unique values after omitting NA  
 distinct\_count <- nrow(unique\_values)  
   
 #Adding, column and distinct count to a separate dataframe  
 Distinct\_Columns[index,] <- list(column, distinct\_count)  
 index <- index + 1;  
   
 #Displaying first 5 unique values of category  
 first\_5 <- head(unique\_values, n = 5)  
 print(first\_5[1])  
 }  
}

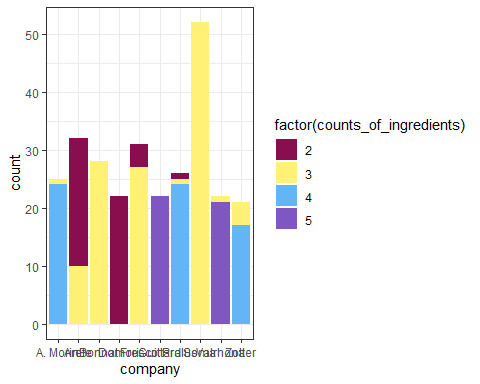
## company  
## 1 5150  
## 4 A. Morin  
## 29 Acalli  
## 33 Adi aka Fijiana (Easy In Ltd)  
## 37 Aelan  
## company\_location  
## 1 U.S.A  
## 4 France  
## 33 Fiji  
## 37 Vanuatu  
## 41 Ecuador  
## country\_of\_bean\_origin  
## 1 Madagascar  
## 2 Dominican republic  
## 3 Tanzania  
## 4 Peru  
## 5 Bolivia  
## first\_taste  
## 1 cocoa  
## 3 rich cocoa  
## 4 fruity  
## 5 vegetal  
## 6 oily  
## second\_taste  
## 1 blackberry  
## 2 vegetal  
## 3 fatty  
## 4 melon  
## 5 nutty  
## third\_taste  
## 1 full body  
## 2 savory  
## 3 bready  
## 4 roasty  
## 6 caramel

#Finding average rating, cocoa percent and count of ingredients for top 10 companies  
  
#Selecting top 10 companies  
freq\_Origin = as.data.frame(table(chocolate$company))  
colnames(freq\_Origin) = c("Company", "Frequency")  
  
# The top 10 chocolate bar producers  
top10s = dplyr::arrange(freq\_Origin, desc(Frequency))[1:10,]  
onlytops = dplyr::filter(chocolate, company %in% top10s[,1])   
  
Company\_Rating\_data <- select(onlytops, c(company, rating, cocoa\_percent, counts\_of\_ingredients))  
Company\_Rating\_data = Company\_Rating\_data %>% group\_by(company) %>%  
 summarise(avg\_rating = mean(rating),  
 avg\_cocoa\_percent = mean(cocoa\_percent),  
 avg\_ingredient\_count = round(mean(counts\_of\_ingredients)),  
 .groups = 'drop')  
  
View(Company\_Rating\_data) 

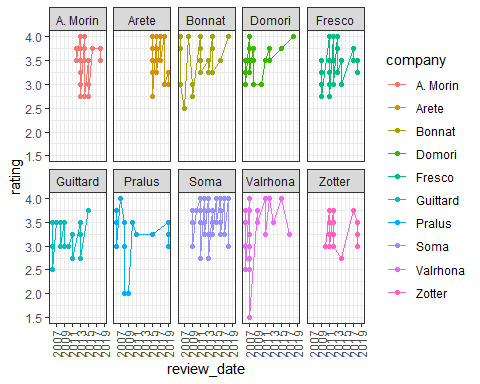
#PLOTTING  
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.2.3

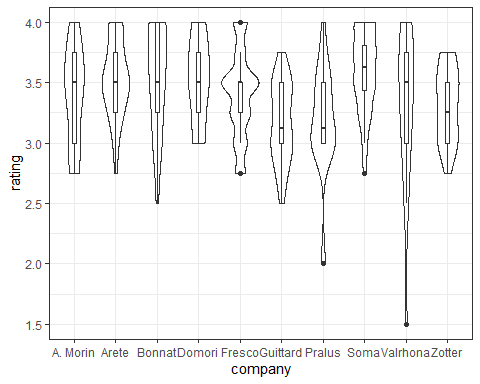
#First plot  
#1. Bar plot of the top 10 producers considering the number of ingredients used  
ggplot(onlytops, aes(x = company, fill = factor(counts\_of\_ingredients))) + geom\_bar() +  
 theme\_bw() + scale\_fill\_manual(values = c("2" = "#880E4F", "3" = "#FFF176","4" = "#64B5F6", "5" = "#7E57C2"))



#Second plot  
#2. Scatter plot of changes in the ratings of chocolate bars (2006-2020)  
ggplot(onlytops, aes(x = review\_date, y = rating, color = company)) +   
 geom\_point() + facet\_wrap(~ company, nrow = 2) + geom\_line() + theme\_bw() +  
 scale\_x\_continuous(breaks = seq(2005,2021,2), guide = guide\_axis(angle = 90))



# aes() function is often used within other graphing elements to specify the desired aesthetics.  
# the function geom\_point() adds a layer of points, which creates a scatterplot.  
# facet\_wrap() makes a long ribbon of panels (generated by any number of variables) and wraps it into 2d  
# nrow control how many columns  
# scale\_x\_continuous is a scale breaks: manually label the ticks  
# guide = rotate the year labels  
  
#Third plot  
#3. Violin plot of the distribution of ratings  
ggplot(onlytops, aes(x = company, y = rating)) + geom\_violin() +  
 geom\_boxplot(width=0.1, fill="white") +theme\_bw()



# How many chocolates contain a particular ingredient?  
total\_chocolates = nrow(chocolate)  
print(total\_chocolates)

## [1] 2224

ingredients = colnames(chocolate[10:16])  
for (ingredient in ingredients) {  
 chocolates\_having\_ingredient = sum(chocolate[ingredient] == 1)  
 percent\_chocolates = as.character(formatC((chocolates\_having\_ingredient/total\_chocolates) \* 100, digits=2, format="f"))  
 cat(sprintf("%s percent of chocolates have %s in them\n", percent\_chocolates, ingredient))  
}

## 100.00 percent of chocolates have bean in them  
## 69.02 percent of chocolates have cocoa\_butter in them  
## 15.65 percent of chocolates have vanilla in them  
## 21.54 percent of chocolates have lecithin in them  
## 1.66 percent of chocolates have salt in them  
## 96.31 percent of chocolates have sugar in them  
## 3.42 percent of chocolates have sweetener\_without\_sugar in them

**Questions**

**<list down questions>**

Chocolate\_Analysis.R

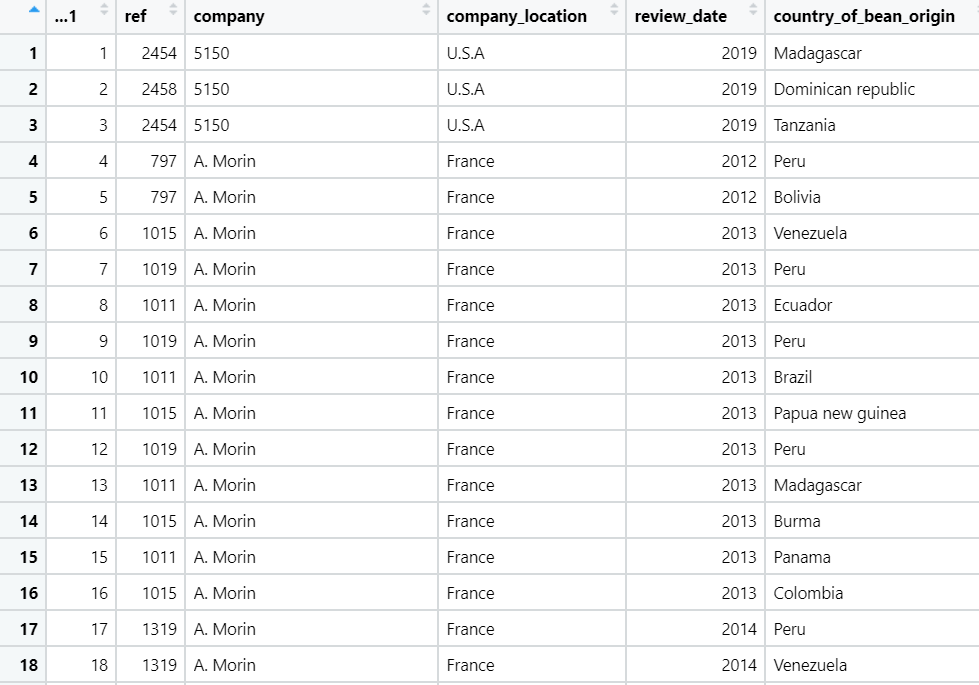
#Creating data frame  
library(readr)

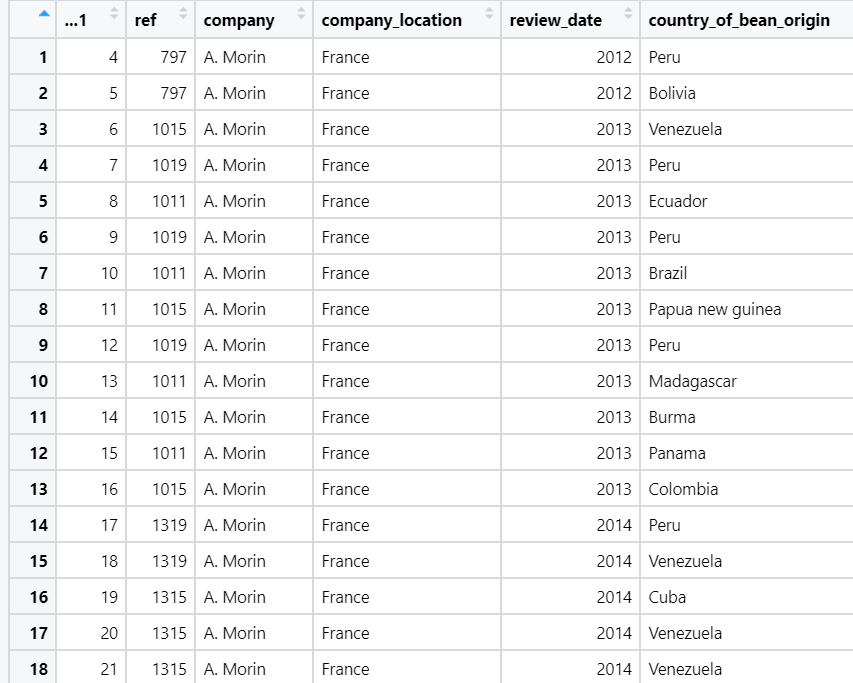
## Warning: package 'readr' was built under R version 4.2.3

chocolate <- read\_csv("data/chocolate.csv")

## New names:  
## Rows: 2224 Columns: 19  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (6): company, company\_location, country\_of\_bean\_origin, first\_taste, se... dbl  
## (13): ...1, ref, review\_date, cocoa\_percent, rating, counts\_of\_ingredien...  
## ℹ Use `spec()` to retrieve the full column specification for this data. ℹ  
## Specify the column types or set `show\_col\_types = FALSE` to quiet this message.  
## • `` -> `...1`

View(chocolate)



chocolate <- data.frame(chocolate)  
  
#Questions  
#Selecting top 10 companies  
freq\_Origin = as.data.frame(table(chocolate$company))  
colnames(freq\_Origin) = c("Company", "Frequency")  
  
# The top 10 chocolate bar producers  
top10s = dplyr::arrange(freq\_Origin, desc(Frequency))[1:10,]  
onlytops = dplyr::filter(chocolate, company %in% top10s[,1])  
  
View(onlytops)  


#1. Find most common first taste and compare rating with other tastes  
  
# find most common first taste  
common\_taste = names(which.max(table(chocolate$first\_taste)))  
  
# find the avg ratings  
average\_rating = 0  
average\_count = nrow(chocolate["rating"])  
common\_taste\_rating = 0  
common\_taste\_rating\_count = 0  
  
taste = chocolate["first\_taste"]  
rating = chocolate["rating"]  
  
for (index in 1:nrow(chocolate["rating"])) {  
 if (taste[[1]][[index]] == common\_taste) {  
 common\_taste\_rating = common\_taste\_rating + rating[[1]][[index]]  
 common\_taste\_rating\_count = common\_taste\_rating\_count + 1  
 }  
 average\_rating = average\_rating + rating[[1]][[index]]  
}  
  
message("Average rating of chocolates is: ", average\_rating / average\_count)

## Average rating of chocolates is: 3.19856115107914

message("Rating for chocolates with most popular taste is: ", common\_taste\_rating / common\_taste\_rating\_count)

## Rating for chocolates with most popular taste is: 3.484375

#2. Multiple regression (model rating in terms of cocoa percent and count of ingredients)  
library(dplyr)

## Warning: package 'dplyr' was built under R version 4.2.3

##   
## 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

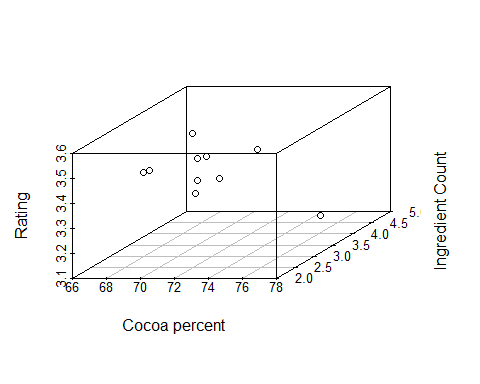
Rating\_data <- select(onlytops, c(company, rating, cocoa\_percent, counts\_of\_ingredients))  
Rating\_data = Rating\_data %>% group\_by(company) %>%  
 summarise(avg\_rating = mean(rating),  
 avg\_cocoa\_percent = mean(cocoa\_percent),  
 avg\_ingredient\_count = round(mean(counts\_of\_ingredients)),  
 .groups = 'drop')  
  
RegModel = lm(Rating\_data$avg\_rating ~ Rating\_data$avg\_cocoa\_percent + Rating\_data$avg\_ingredient\_count)  
RegModel

##   
## Call:  
## lm(formula = Rating\_data$avg\_rating ~ Rating\_data$avg\_cocoa\_percent +   
## Rating\_data$avg\_ingredient\_count)  
##   
## Coefficients:  
## (Intercept) Rating\_data$avg\_cocoa\_percent   
## 4.61467 -0.01119   
## Rating\_data$avg\_ingredient\_count   
## -0.12244

summary(RegModel)

##   
## Call:  
## lm(formula = Rating\_data$avg\_rating ~ Rating\_data$avg\_cocoa\_percent +   
## Rating\_data$avg\_ingredient\_count)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.09964 -0.05898 -0.03310 0.06209 0.13657   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.61467 0.87982 5.245 0.00119 \*\*  
## Rating\_data$avg\_cocoa\_percent -0.01119 0.01175 -0.953 0.37255   
## Rating\_data$avg\_ingredient\_count -0.12244 0.03222 -3.800 0.00671 \*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.09634 on 7 degrees of freedom  
## Multiple R-squared: 0.6781, Adjusted R-squared: 0.5862   
## F-statistic: 7.374 on 2 and 7 DF, p-value: 0.01892

library(scatterplot3d)  
scatterplot3d(Rating\_data$avg\_rating ~ Rating\_data$avg\_cocoa\_percent + Rating\_data$avg\_ingredient\_count,  
 xlab = "Cocoa percent",  
 ylab = "Ingredient Count",  
 zlab = "Rating")



#3. Hypothesis testing (take top 10 companies producing chocolates as a sample and see if it's representative of the entire population) mean of the rating  
sample\_mean=0  
sample\_size=0  
temp = data.frame(sort(table(chocolate$company), decreasing=TRUE)[1:10])  
temp

## Var1 Freq  
## 1 Soma 52  
## 2 Arete 32  
## 3 Fresco 31  
## 4 Bonnat 28  
## 5 Pralus 26  
## 6 A. Morin 25  
## 7 Domori 22  
## 8 Guittard 22  
## 9 Valrhona 22  
## 10 Zotter 21

soma = data.frame(chocolate[chocolate$company=='Soma' , ])  
tempmean=mean(soma$rating)  
n=nrow(soma)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
arete = data.frame(chocolate[chocolate$company=='Arete' , ])  
tempmean=mean(arete$rating)  
n=nrow(arete)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
fresco = data.frame(chocolate[chocolate$company=='Fresco' , ])  
tempmean=mean(fresco$rating)  
n=nrow(fresco)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
bonnat = data.frame(chocolate[chocolate$company=='Bonnat' , ])  
tempmean=mean(bonnat$rating)  
n=nrow(bonnat)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
pralus = data.frame(chocolate[chocolate$company=='Pralus' , ])  
tempmean=mean(pralus$rating)  
n=nrow(pralus)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
amorin = data.frame(chocolate[chocolate$company=='A. Morin' , ])  
tempmean=mean(amorin$rating)  
n=nrow(amorin)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
domori = data.frame(chocolate[chocolate$company=='Domori' , ])  
tempmean=mean(domori$rating)  
n=nrow(domori)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
guittard = data.frame(chocolate[chocolate$company=='Guittard' , ])  
tempmean=mean(guittard$rating)  
n=nrow(guittard)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
valrhona = data.frame(chocolate[chocolate$company=='Valrhona' , ])  
tempmean=mean(valrhona$rating)  
n=nrow(valrhona)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
zotter = data.frame(chocolate[chocolate$company=='Zotter' , ])  
tempmean=mean(zotter$rating)  
n=nrow(zotter)  
sample\_size=sample\_size+n  
sample\_mean=sample\_mean+tempmean  
  
sample\_mean=sample\_mean/10  
population\_mean=mean(chocolate$rating)  
sigma=sd(chocolate$rating)  
  
z=(sample\_mean-population\_mean)/(sigma/sqrt(sample\_size))  
  
alpha=0.05  
zhalfalpha=qnorm(1-(alpha/2))  
c(-zhalfalpha,zhalfalpha)

## [1] -1.959964 1.959964

pval=2\*pnorm(z)  
  
if(pval>alpha){print("Accept Null hypothesis")} else{print("Reject Null hypothesis")}

## [1] "Accept Null hypothesis"

#4. Hypothesis testing (take 2 companies, compare the average reviews) - check of one country produces objectively better chocolates   
ratingmeans=aggregate(onlytops$rating, list(onlytops$company), FUN=mean)  
ratingmeans

## Group.1 x  
## 1 A. Morin 3.400000  
## 2 Arete 3.531250  
## 3 Bonnat 3.526786  
## 4 Domori 3.522727  
## 5 Fresco 3.403226  
## 6 Guittard 3.170455  
## 7 Pralus 3.173077  
## 8 Soma 3.591346  
## 9 Valrhona 3.318182  
## 10 Zotter 3.321429

P1=(ratingmeans$Group.1[1])  
m1=ratingmeans$x[1]  
P2=(ratingmeans$Group.1[2])  
m2=ratingmeans$x[2]  
n1=sum(onlytops$company==P1)  
n2=sum(onlytops$company==P2)  
n2

## [1] 32

ratingsd=aggregate(onlytops$rating, list(onlytops$company), FUN=sd)  
sd1=ratingsd$x[1]  
sd2=ratingsd$x[2]  
  
#t test for 2 means  
#H0 : Both companies have the same quality/popularity as the other  
#H1 : The companies are not on the same level of quality/popularity  
# testing at 5% level of significance  
t= (m1-m2)/sqrt(((sd1\*sd1)/n1)+((sd2\*sd2)/n2))  
t

## [1] -1.318355

cv=qt(0.975,(n1+n2-2))  
cv

## [1] 2.004045

if(cv <=t){print("Accept Ho")} else{print("Reject Ho")}

## [1] "Reject Ho"

**CONCLUSION**

1. ***Find most common first taste and compare rating with other tastes***
2. ***Multiple regression - Modelling chocolate rating in terms of cocoa percent and count of ingredients***
3. **Q3**
4. **Q4**