Exploratory\_Analysis.R

skhim

2023-04-04

#Creating data frame  
library(readr)

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

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

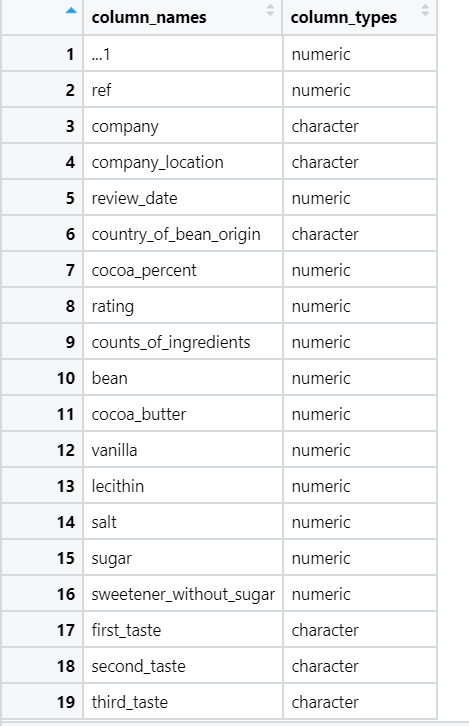
## New names:  
## Rows: 2224 Columns: 19  
## ── Column specification  
## ──────────────────────────────────────────────────────── Delimiter: "," chr  
## (6): company, company\_location, country\_of\_bean\_origin, first\_taste, se... dbl  
## (13): ...1, ref, review\_date, cocoa\_percent, rating, 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

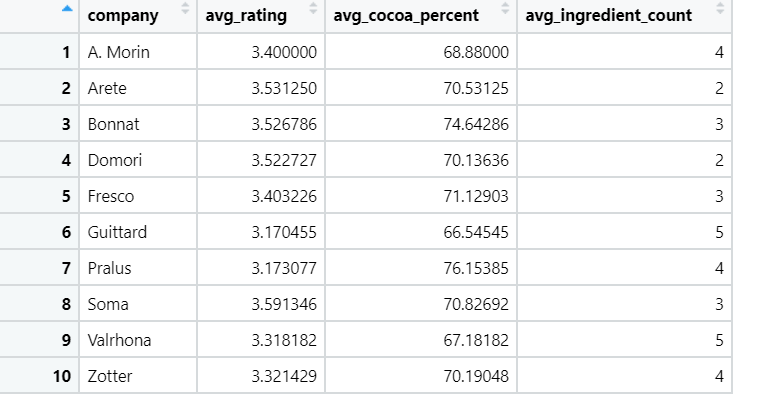
summary(select(chocolate, c(review\_date, cocoa\_percent, rating, counts\_of\_ingredients)))

## 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

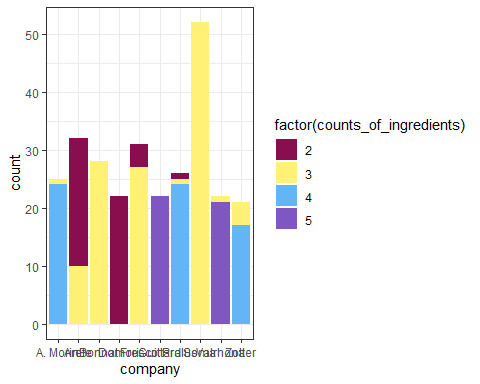
#Finding average rating, cocoa percent and count of ingredients for top 10 companies  
  
#Selecting top 10 companies  
freq\_Origin = as.data.frame(table(chocolate$company))  
colnames(freq\_Origin) = c("Company", "Frequency")  
  
# The top 10 chocolate bar producers  
top10s = dplyr::arrange(freq\_Origin, desc(Frequency))[1:10,]  
onlytops = dplyr::filter(chocolate, company %in% top10s[,1])   
  
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')  
  
View(Rating\_data)



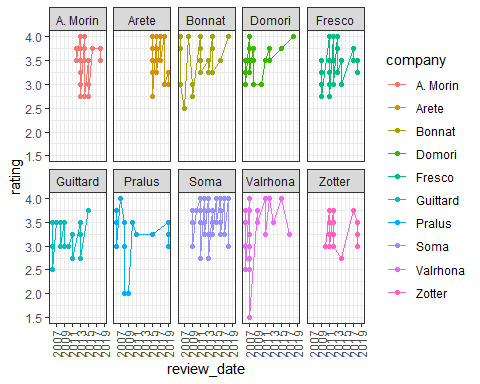
#PLOTTING  
library(ggplot2)

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

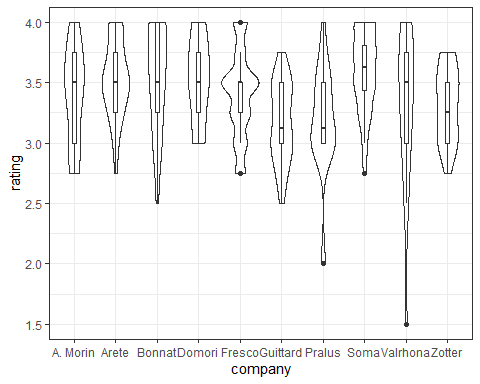
#First plot  
#1. Bar plot of the top 10 producers considering the number of ingredients used  
ggplot(onlytops, aes(x = company, fill = factor(counts\_of\_ingredients))) + geom\_bar() +  
 theme\_bw() + scale\_fill\_manual(values = c("2" = "#880E4F", "3" = "#FFF176","4" = "#64B5F6", "5" = "#7E57C2"))



#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