Draft

STA 210 - Project

Ginger and Stats - Aimi Wen, Rakshita Ramakrishna, Nathan Nguyen

```r
library(tidyverse)
library(tidymodels)
library(tidytext)
library(patchwork)
library(stringr)


library(ggplot2)
library(sf)
library(rnaturalearth)
library(rnaturalearthdata)
chocolate <- read_csv("../data/chocolate.csv")

world <- ne_countries(scale = "medium", returnclass = "sf")
```

```r
library(countrycode)
```

Exploratory Data Analysis

Data description

- Description of the observations in the data set:
  - The observations in this data set represent a review of general characteristics for different chocolate bars. A single observation in this data set represents a single chocolate bar.
  - The general characteristics are as follows:

        - Company (Manufacturer) lists who made the chocolate bar reviewed; the dataset also lists where this company is located under Company Location.
        - The dataset characterizes the Country of Bean Origin, Specific Bean Origin or name of bar, Percentage of Cocoa within the bar for each chocolate bar.
        - The data also shows which ingredients are used using letters, where B = Beans, S = Sugar, S* = Sweeteners other than white can or beet sugar, C = Cocoa Butter, V = Vanilla, L = Lecithin, Sa = Salt.
        - Finally, the data shows the rating (which ranges from 1-5, incrementing by 0.25) given under their rating system, which is linked above, as well as the date it was reviewed on

- Description of how the data was originally collected (not how you found the data but how the original curator of the data collected it).
    - Data is being continuously collected and added to the dataset after reviewing chocolate bars - this can be seen as the first review years for chocolate bars began in 2006 and have continued until 2021.

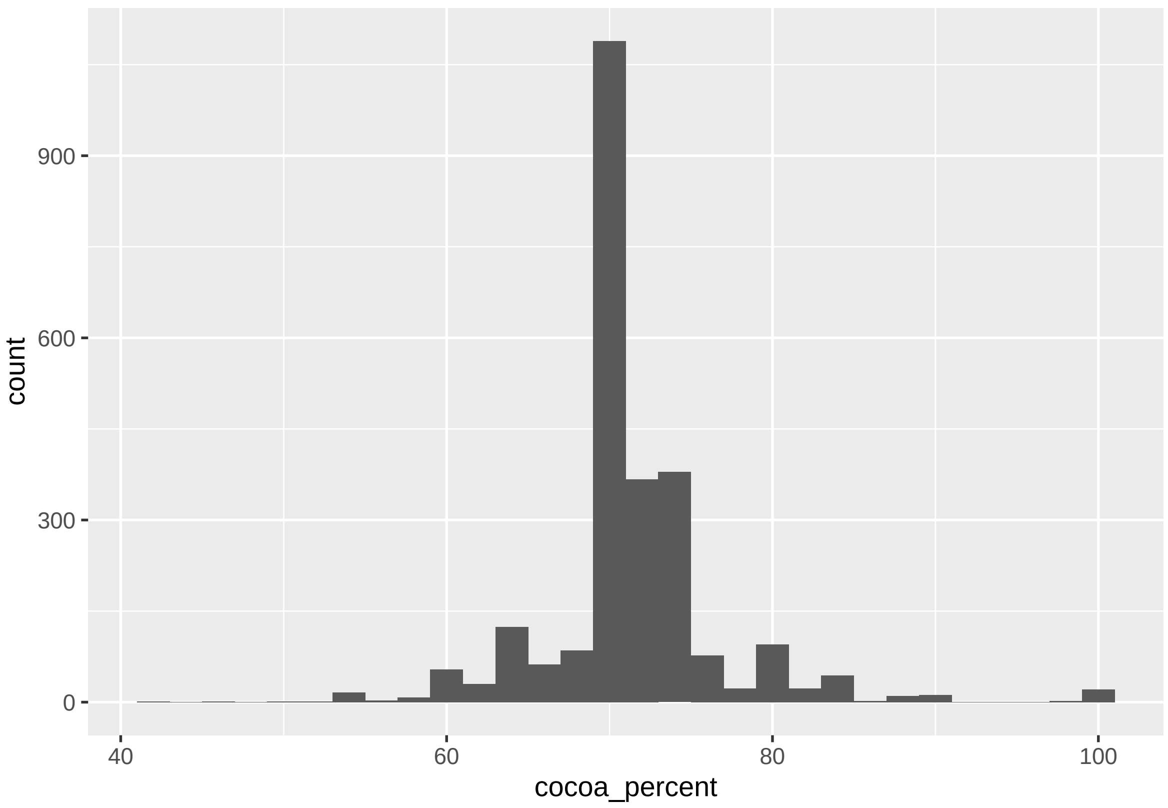
The data is collected by members of the Manhattan Chocolate Society reviewing chocolate bars using the rating system found at http://flavorsofcacao.com/review_guide.html and adding other characteristics about the bar itself.

Shape of Ratings (already done)

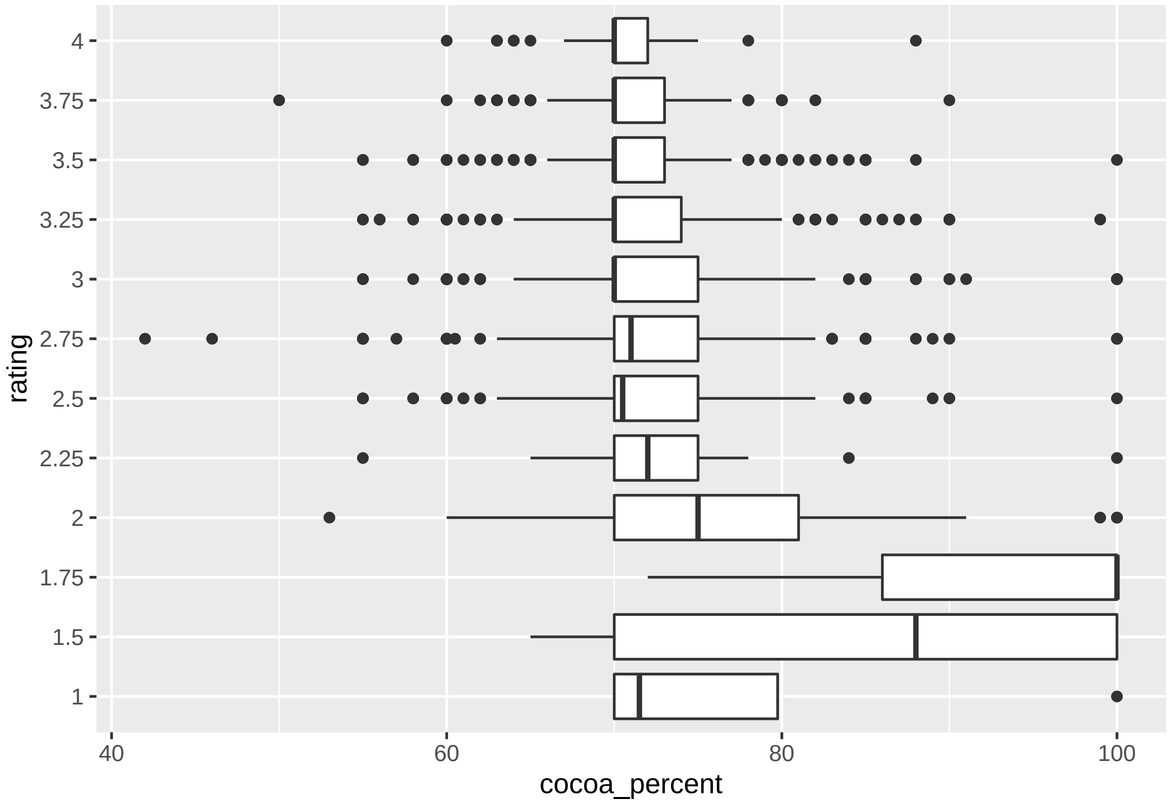
Cocoa Percent (Aimi)

```r
chocolate$cocoa_percent <- as.numeric(gsub('[,%]', '', chocolate$cocoa_percent))

chocolate$rating <- as.character(chocolate$rating)

ggplot(data= chocolate, aes(x= cocoa_percent)) + geom_histogram()
```

```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(data= chocolate, aes(x= cocoa_percent, y= rating)) + geom_boxplot()
```

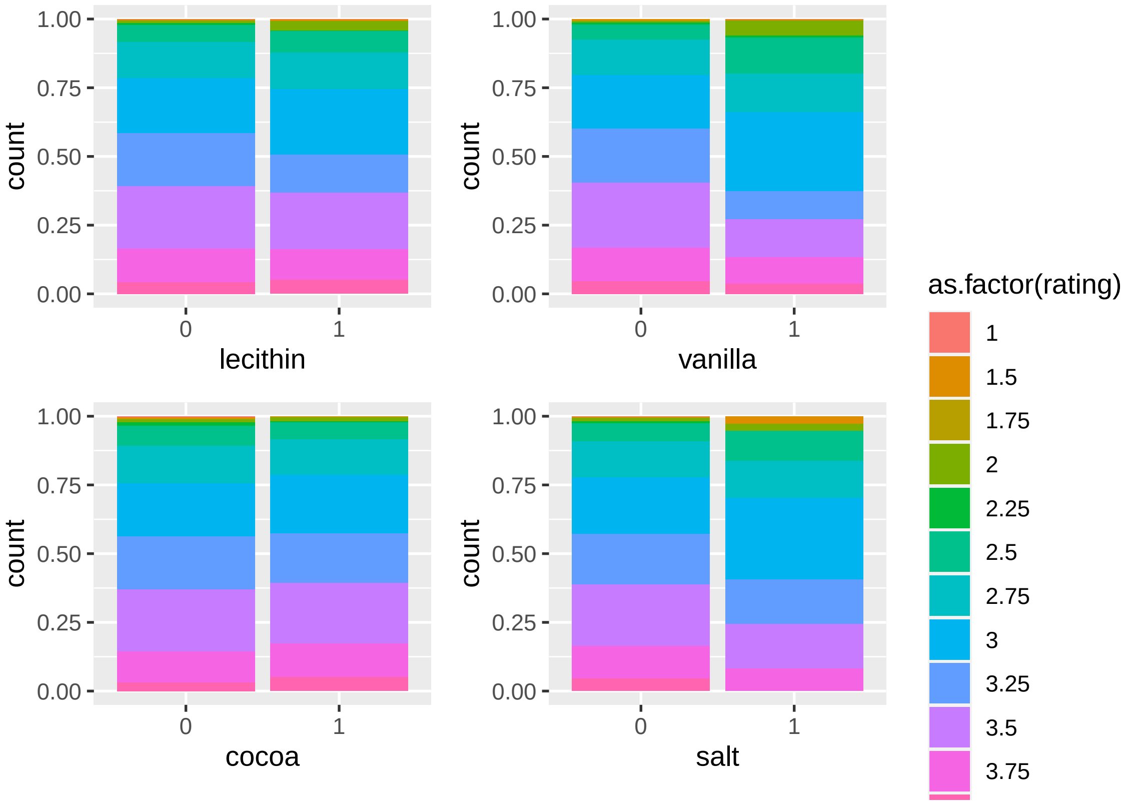


```r
chocolate$rating <- as.numeric(chocolate$rating)
```

Ingredients (Nathan)

```r
chocolate <- chocolate %>%
  mutate(lecithin = case_when(
    grepl("L", ingredients) ~ 1,
    T ~ 0
  ),
  vanilla = case_when(
    grepl("V", ingredients) ~ 1,
    T ~ 0
  ),
  cocoa = case_when(
    grepl("C", ingredients) ~ 1,
    T ~ 0
  ),
  salt = case_when(
    grepl("Sa", ingredients) ~ 1,
    T ~ 0
  ),

  lecithin = as.factor(lecithin),
  vanilla = as.factor(vanilla),
  cocoa = as.factor(cocoa),
  salt = as.factor(salt)
  )
```

```
pL <- ggplot(chocolate, aes(lecithin, fill = as.factor(rating))) +
  geom_bar(position = "fill")+
  theme(legend.position = "none")
pV <- ggplot(chocolate, aes(vanilla, fill = as.factor(rating))) +
  geom_bar(position = "fill")+
  theme(legend.position = "none")
pC <- ggplot(chocolate, aes(cocoa, fill = as.factor(rating))) +
  geom_bar(position = "fill")+
  theme(legend.position = "none")
pSa <- ggplot(chocolate, aes(salt, fill = as.factor(rating))) +
  geom_bar(position = "fill")

(pL + pV)/(pC + pSa)
```

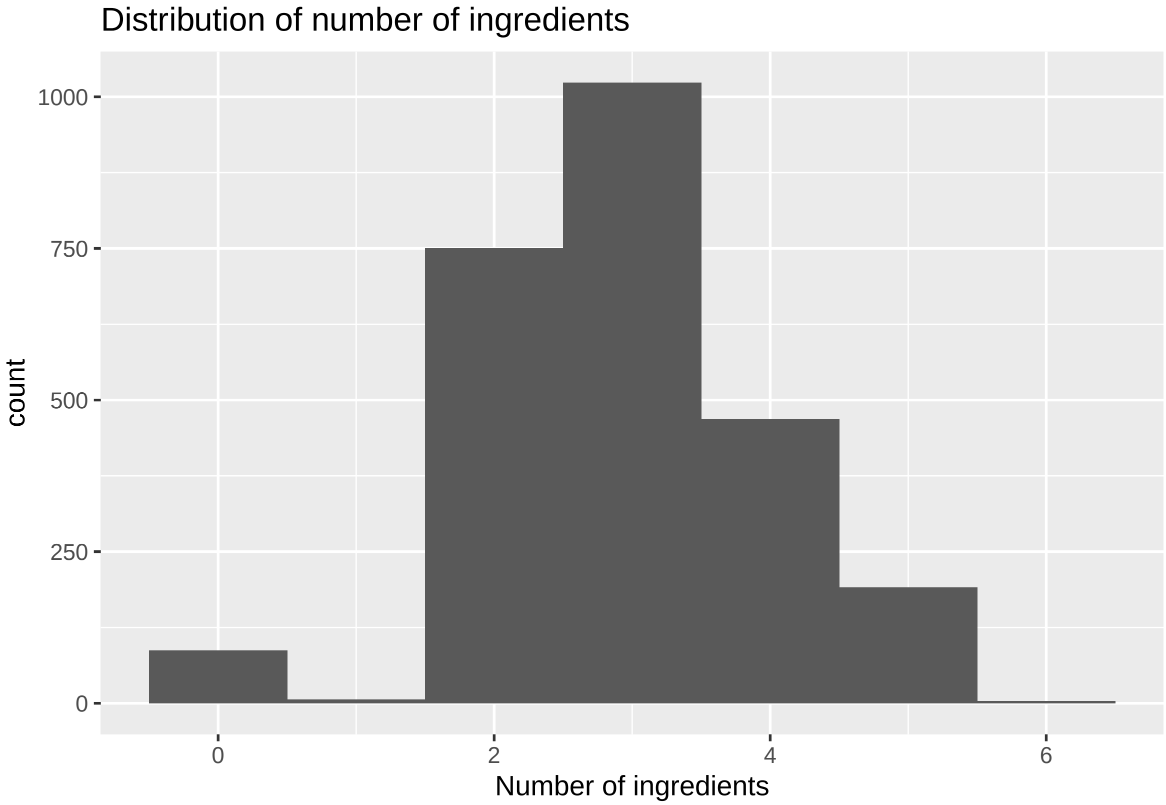


```
chocolate <- chocolate %>%
  mutate(
    num_ingres = if_else(is.na(ingredients), "0", str_sub(ingredients, 1, 1)),
    num_ingres = as.numeric(num_ingres)
  )
```

```
chocolate %>%
  drop_na(
    ingredients
  ) %>%
  count()
```

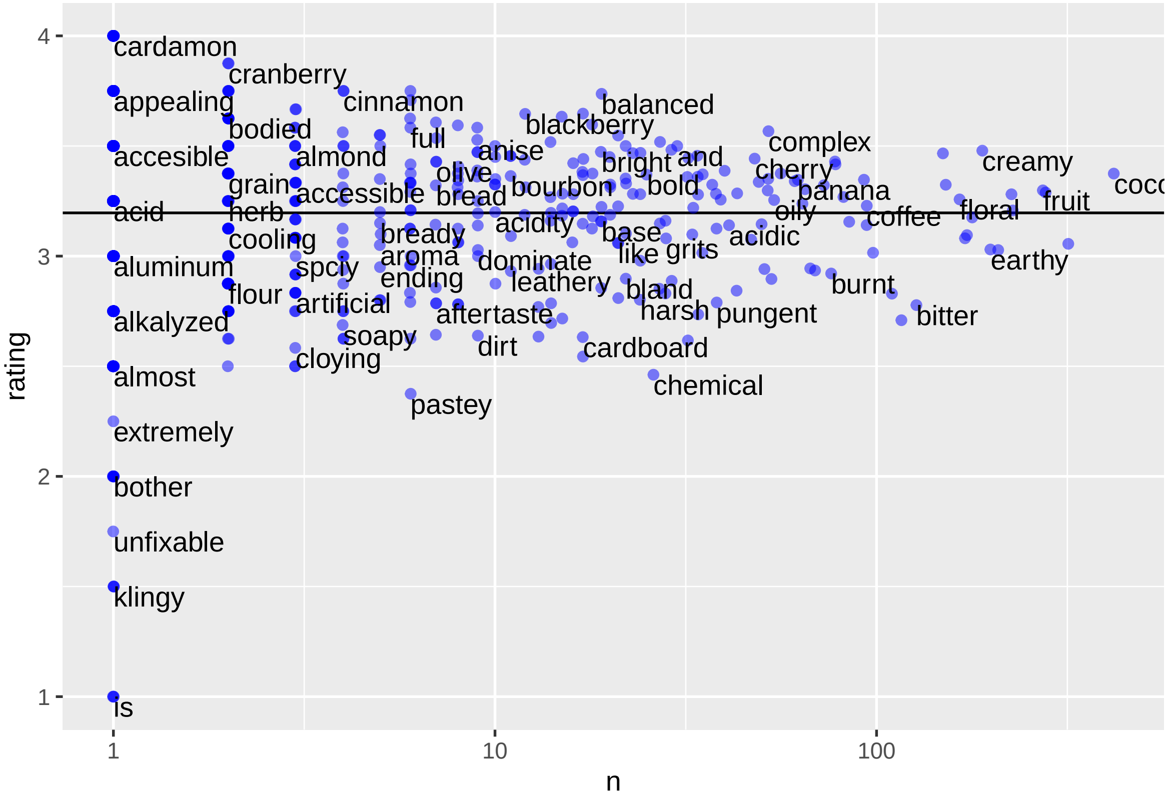
```
# A tibble: 1 x 1
      n
  <int>
1  2443
```

```
ggplot(chocolate, aes(num_ingres))+
  geom_histogram(binwidth = 1)+
  labs(
    title = "Distribution of number of ingredients",
    x = "Number of ingredients"
  )
```



Most Memorable Characteristic (Aimi)

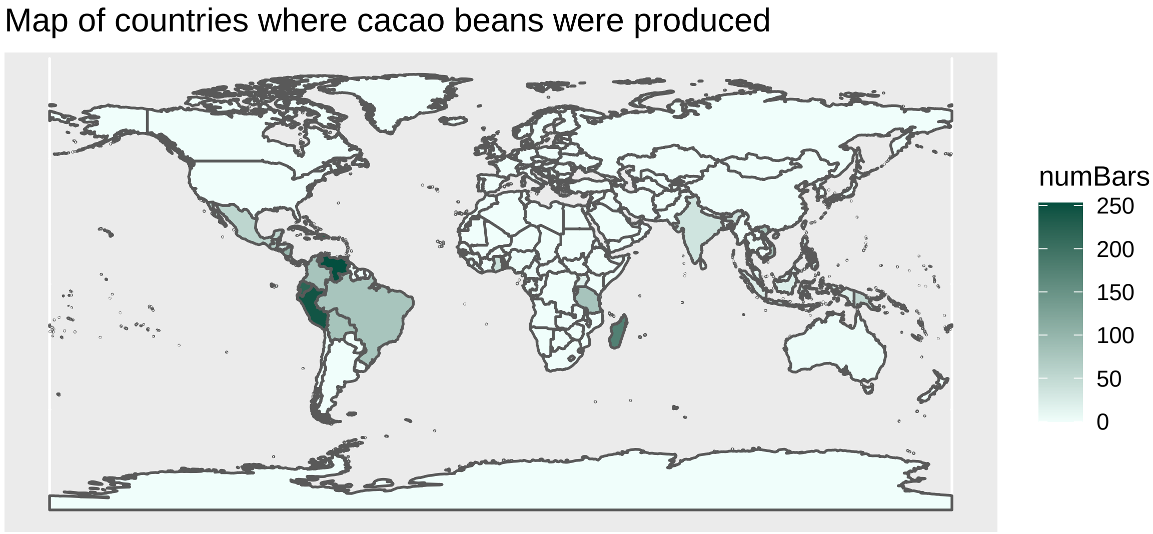
```
tidy_chocolate<- chocolate %>%
  unnest_tokens(word, most_memorable_characteristics)

tidy_chocolate %>%
  group_by(word) %>%
  summarize( n= n(),
             rating= mean(rating) ) %>%
  ggplot(aes(n, rating)) +
  geom_hline(yintercept= mean(chocolate$rating)) +
  geom_jitter(color= "blue", alpha= 0.5) +
  geom_text(aes(label= word),
            check_overlap= TRUE,
            vjust= "top",
            hjust= "left") +
  scale_x_log10()
```



Country Bean of Origin (Rakshita)

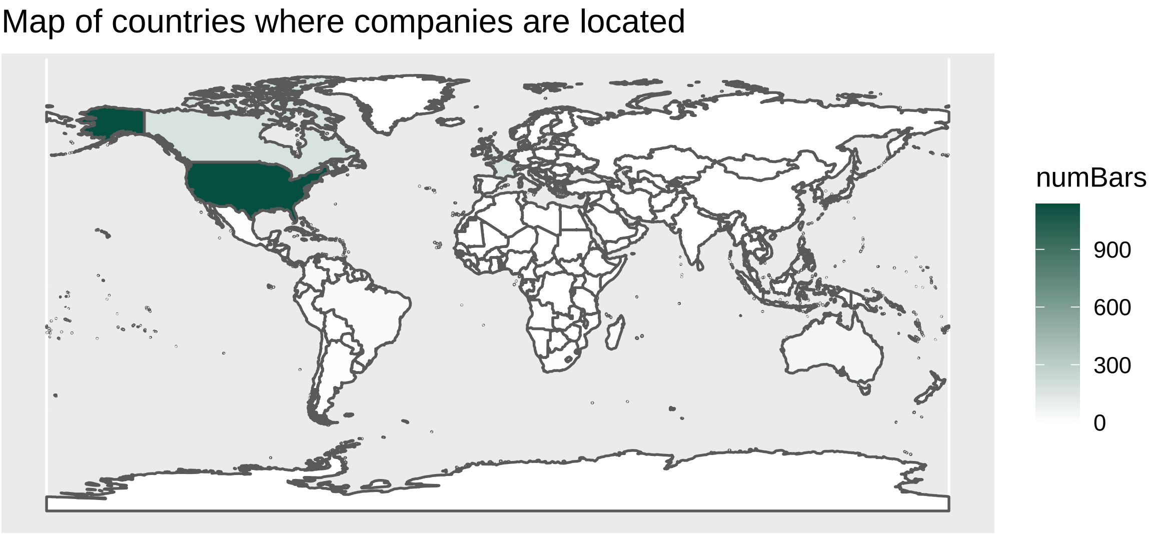
```
chocolate_modified <- chocolate %>%
  mutate(name_long = country_of_bean_origin) %>%
  group_by(name_long) %>%
  count(name_long)
```

```
chocworld_data <- world %>%
  full_join(y = chocolate_modified,
  by = "name_long") %>%
  mutate(numBars = ifelse(is.na(n), 0, n))

ggplot(data = chocworld_data) +
  scale_fill_gradient(low = "#F0FEFB", high = "#044F3F") +
  geom_sf(aes(fill = numBars, geometry = geometry)) +
  labs(title = "Map of countries where cacao beans were produced")
```



Company Location (Rakshita)

```
chocolate_modified2 <- chocolate %>%
  mutate(name_long = case_when(
    company_location == "U.S.A." ~ "United States",
    company_location == "U.K." ~ "United Kingdom",
    company_location == company_location ~ company_location)) %>%
  group_by(name_long) %>%
  count(name_long)

chocworld_data1 <- world %>%
  full_join(y = chocolate_modified2,
  by = "name_long") %>%
  mutate(numBars = ifelse(is.na(n), 0, n))

ggplot(data = chocworld_data1) +
  scale_fill_gradient(low = "#ffffff", high = "#044F3F") +
  geom_sf(aes(fill = numBars, geometry = geometry)) +
  labs(title = "Map of countries where companies are located")
```

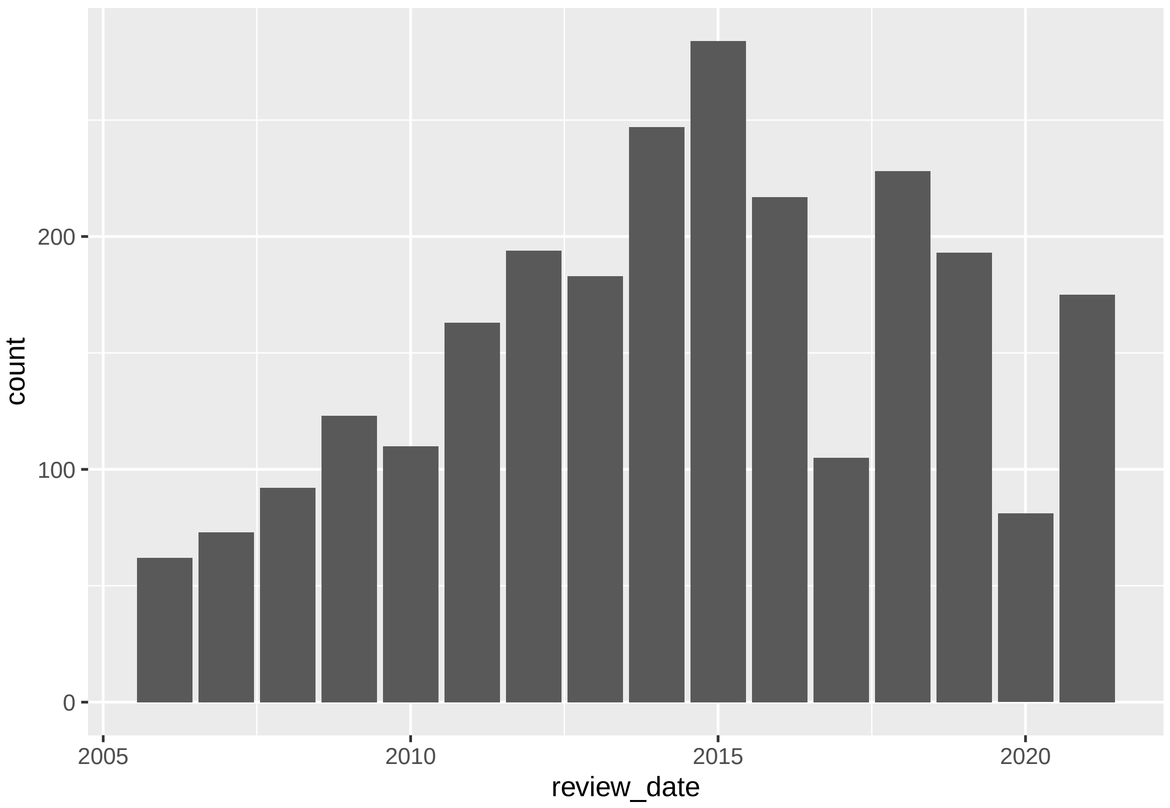


```
chocolate %>%
    count(company_location, sort = TRUE)
```

```
# A tibble: 67 x 2
   company_location     n
   <chr>            <int>
 1 U.S.A.            1136
 2 Canada             177
 3 France             176
 4 U.K.               133
 5 Italy               78
 6 Belgium             63
 7 Ecuador             58
 8 Australia           53
 9 Switzerland         44
10 Germany             42
# ... with 57 more rows
```

Review Date (Nathan)

```
ggplot(chocolate, aes(review_date))+
  geom_bar()
```

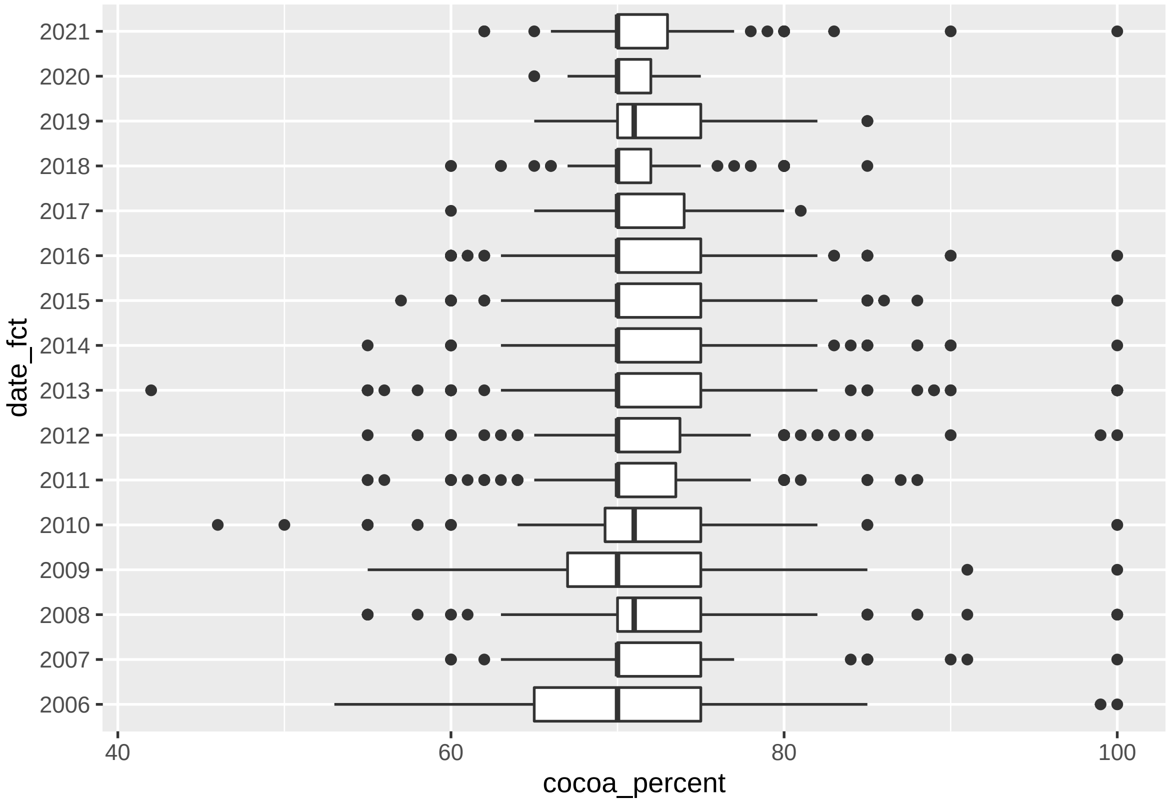


```
# statistics of review dates

chocolate %>%
  summarise(mean = mean(review_date),
            median = median(review_date),
            sd = sd(review_date))
```

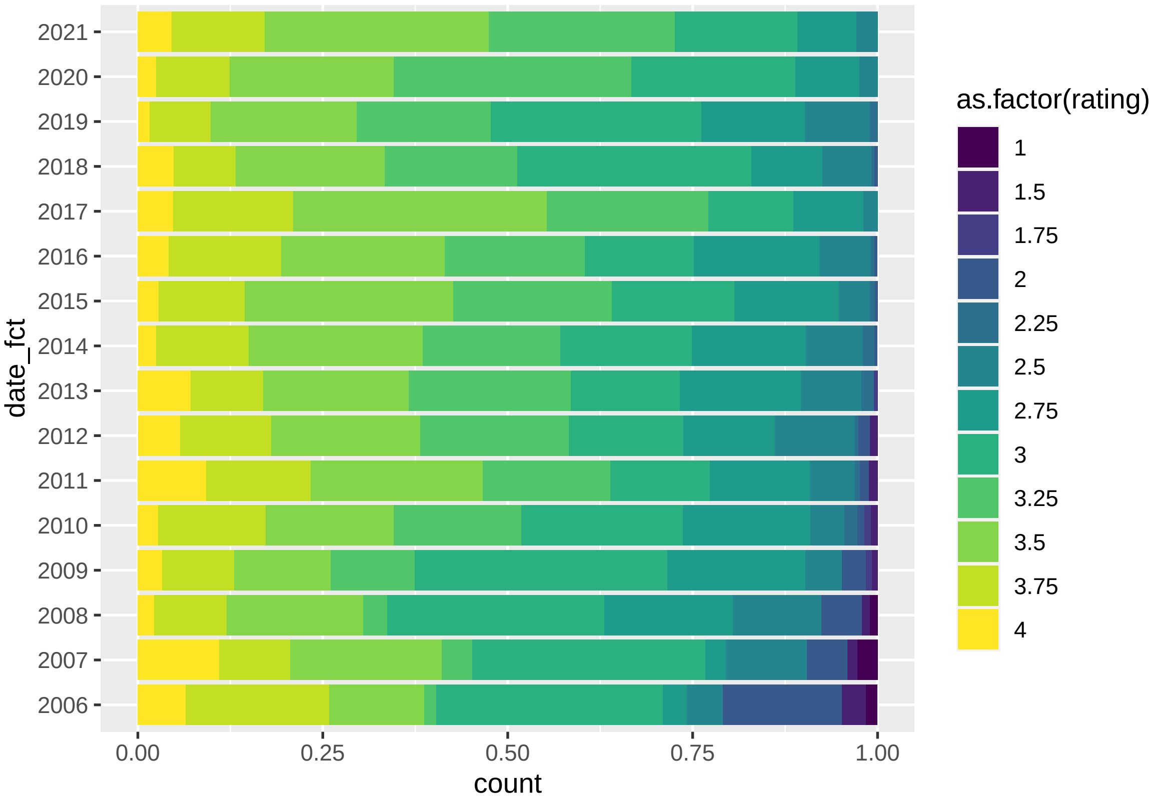
```
# A tibble: 1 x 3
   mean median    sd
  <dbl>  <dbl> <dbl>
1 2014.   2015  3.97
```

```
#review date vs cocoa_percent and ratings

chocolate <- chocolate %>%
  mutate(
    date_fct = as.factor(review_date)
  )
```

```
ggplot(chocolate, aes(date_fct, cocoa_percent))+
  geom_boxplot()+
  coord_flip()
```



```
ggplot(chocolate, aes(date_fct, fill = as.factor(rating)))+
  geom_bar(position = "fill")+
  coord_flip()+
  scale_fill_viridis_d()
```



```
chocolate_clean <- chocolate %>%
  separate(most_memorable_characteristics, sep= ",", into= c("most_memorable", "other_memoral
  select(-other_memorable)
```

```
Warning: Expected 2 pieces. Missing pieces filled with `NA` in 95 rows [14, 34,
39, 41, 99, 145, 168, 228, 240, 264, 281, 290, 357, 365, 368, 405, 426, 433,
442, 477, ...].
```

```
#|label: cleaning-dataset
chocolate_clean <- chocolate_clean %>%
  mutate(
    top_memorable= case_when(
      str_detect(most_memorable, "cream") ~ "fatty_smooth",
      str_detect(most_memorable, "fatty") ~ "fatty_smooth",
      str_detect(most_memorable, "smooth") ~ "fatty_smooth",
      str_detect(most_memorable, "dairy") ~ "fatty_smooth",
      str_detect(most_memorable, "roast") ~ "roast",
      str_detect(most_memorable, "earth") ~ "roast",
      str_detect(most_memorable, "smoke") ~ "roast",
      str_detect(most_memorable, "wood") ~ "roast",
      str_detect(most_memorable, "bitter") ~ "roast",
      str_detect(most_memorable, "intense") ~ "strong_sweet",
      str_detect(most_memorable, "sweet") ~ "strong_sweet",
      str_detect(most_memorable, "cocoa") ~ "strong_sweet",
      str_detect(most_memorable, "caramel") ~ "strong_sweet",
      str_detect(most_memorable, "brownie")~ "strong_sweet",
      str_detect(most_memorable, "sandy") ~ "rough_texture",
      str_detect(most_memorable, "dry") ~ "rough_texture",
      str_detect(most_memorable, "gritty") ~ "rough_texture",
      str_detect(most_memorable, "coarse") ~ "rough_texture",
      str_detect(most_memorable, "chalky") ~ "rough_texture",
      str_detect(most_memorable, "powdery") ~ "rough_texture",
      str_detect(most_memorable, "nut") ~ "nutty",
      str_detect(most_memorable, "sticky") ~ "greasy",
      str_detect(most_memorable, "oily") ~ "greasy",
      str_detect(most_memorable, "spic") ~ "spiced",
      str_detect(most_memorable, "molasses") ~ "spiced",
      str_detect(most_memorable, "floral") ~ "floral",
      str_detect(most_memorable, "grassy") ~ "floral",
      str_detect(most_memorable, "vanilla") ~ "floral",
      str_detect(most_memorable, "fruit") ~ "fruity",
      str_detect(most_memorable, "tart") ~ "fruity",
      str_detect(most_memorable, "banana") ~ "fruity",
      str_detect(most_memorable, "berry") ~ "fruity",
      str_detect(most_memorable, "berries") ~ "fruity",
      str_detect(most_memorable, "citrus") ~ "fruity",
      str_detect(most_memorable, "lemon") ~ "fruity",
      str_detect(most_memorable, "complex") ~ "complex",
      TRUE ~ "other"
    )
  )
```

```
chocolate_clean$continent_bean <- countrycode(sourcevar= chocolate_clean[["country_of_bean_o
                                              destination= "continent")
```

```
Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, :
```

```
chocolate_clean <- chocolate_clean %>%
  mutate(continent_bean= ifelse(
    country_of_bean_origin== "U.S.A.", "North America", continent_bean
  ))

chocolate_clean <- chocolate_clean %>%
  mutate(continent_bean= ifelse(
    continent_bean== "Americas", "South America", continent_bean
  ))
```

```
chocolate_clean <- chocolate_clean %>%
  mutate(continent_bean= case_when(
    continent_bean== "South America" ~ "South America",
    continent_bean== "Africa" ~ "Africa",
    continent_bean== "Asia" ~ "Asia",
    TRUE ~ "Other"
  ))
```

```
chocolate_clean$continent_company <- countrycode(sourcevar= chocolate_clean[["company_locati
                                                  destination= "continent")
```

```
Warning in countrycode_convert(sourcevar = sourcevar, origin = origin, destination = dest, :
```

```
chocolate_clean <- chocolate_clean %>%
  mutate(continent_company= ifelse(
    company_location== "U.S.A.", "North America", continent_company
  )) %>%
  mutate(continent_company=ifelse(
    company_location== "Canada", "North America", continent_company
  )) %>%
  mutate(continent_company= ifelse(
    continent_company== "Americas", "South America", continent_company
    )
  )
```

```
chocolate_clean <- chocolate_clean %>%
  mutate(continent_company= case_when(
    continent_company== "North America" ~ "North America",
    continent_company== "Europe" ~ "Europe",
    TRUE ~ "Other"
  ))
```

Analysis approach

Ratings vs cocoa percent, ingredients, most memorable characteristics

```
choco_fit <- linear_reg() %>%
  set_engine("lm") %>%
  fit(rating ~ cocoa_percent + vanilla + salt +
        num_ingres + top_memorable, data = chocolate_clean)
```

```
tidy(choco_fit)
```

```
# A tibble: 15 x 5
   term                       estimate std.error statistic   p.value
   <chr>                         <dbl>     <dbl>     <dbl>     <dbl>
 1 (Intercept)                  4.29     0.139       30.7  1.90e-176
 2 cocoa_percent               -0.0119   0.00152     -7.80 8.78e- 15
 3 vanilla1                    -0.317    0.0306     -10.4  1.19e- 24
 4 salt1                       -0.277    0.0704      -3.93 8.73e-  5
 5 num_ingres                   0.0529   0.0103       5.13 3.07e-  7
 6 top_memorablefatty_smooth   -0.174    0.0799      -2.18 2.94e-  2
 7 top_memorablefloral         -0.388    0.0862      -4.50 7.09e-  6
 8 top_memorablefruity         -0.134    0.0815      -1.64 1.00e-  1
 9 top_memorablegreasy         -0.357    0.0851      -4.19 2.82e-  5
10 top_memorablenutty          -0.285    0.0845      -3.37 7.66e-  4
11 top_memorableother          -0.415    0.0778      -5.34 1.00e-  7
12 top_memorableroast          -0.421    0.0807      -5.22 1.96e-  7
13 top_memorablerough_texture  -0.521    0.0796      -6.55 7.02e- 11
14 top_memorablespiced         -0.297    0.0842      -3.52 4.32e-  4
15 top_memorablestrong_sweet   -0.328    0.0792      -4.15 3.50e-  5
```

```
glance(choco_fit) %>%
  select(adj.r.squared, AIC, BIC)
```

```
# A tibble: 1 x 3
  adj.r.squared   AIC   BIC
          <dbl> <dbl> <dbl>
1         0.129 2755. 2849.
```

All predictors

```
choco_fit_full <- linear_reg() %>%
  set_engine("lm") %>%
  fit(rating ~ cocoa_percent + vanilla + salt +
        num_ingres + top_memorable + continent_company,
      data = chocolate_clean)

tidy(choco_fit_full)
```

```
# A tibble: 17 x 5
   term                           estimate std.error statistic   p.value
   <chr>                             <dbl>     <dbl>     <dbl>     <dbl>
 1 (Intercept)                      4.28     0.142      30.0   1.40e-169
 2 cocoa_percent                   -0.0119   0.00153    -7.78  1.06e- 14
 3 vanilla1                        -0.319    0.0308    -10.4   9.57e- 25
 4 salt1                           -0.277    0.0706     -3.92  9.17e-  5
 5 num_ingres                       0.0545   0.0105      5.17  2.54e-  7
 6 top_memorablefatty_smooth       -0.173    0.0800     -2.16  3.09e-  2
 7 top_memorablefloral             -0.386    0.0862     -4.48  7.77e-  6
 8 top_memorablefruity             -0.138    0.0815     -1.69  9.18e-  2
 9 top_memorablegreasy             -0.355    0.0851     -4.17  3.19e-  5
10 top_memorablenutty              -0.283    0.0845     -3.35  8.24e-  4
11 top_memorableother              -0.414    0.0778     -5.32  1.13e-  7
12 top_memorableroast              -0.420    0.0807     -5.21  2.05e-  7
13 top_memorablerough_texture      -0.520    0.0796     -6.53  7.97e- 11
14 top_memorablespiced             -0.297    0.0842     -3.53  4.21e-  4
15 top_memorablestrong_sweet       -0.328    0.0792     -4.14  3.54e-  5
16 continent_companyNorth America   0.0155   0.0201      0.773 4.40e-  1
17 continent_companyOther          -0.0183   0.0246     -0.744 4.57e-  1
```

```
glance(choco_fit_full) %>%
  select(adj.r.squared, AIC, BIC)
```

```
# A tibble: 1 x 3
  adj.r.squared   AIC   BIC
          <dbl> <dbl> <dbl>
1         0.129 2757. 2862.
```

Data

The data dictionary can be found here.