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

Analysis approach

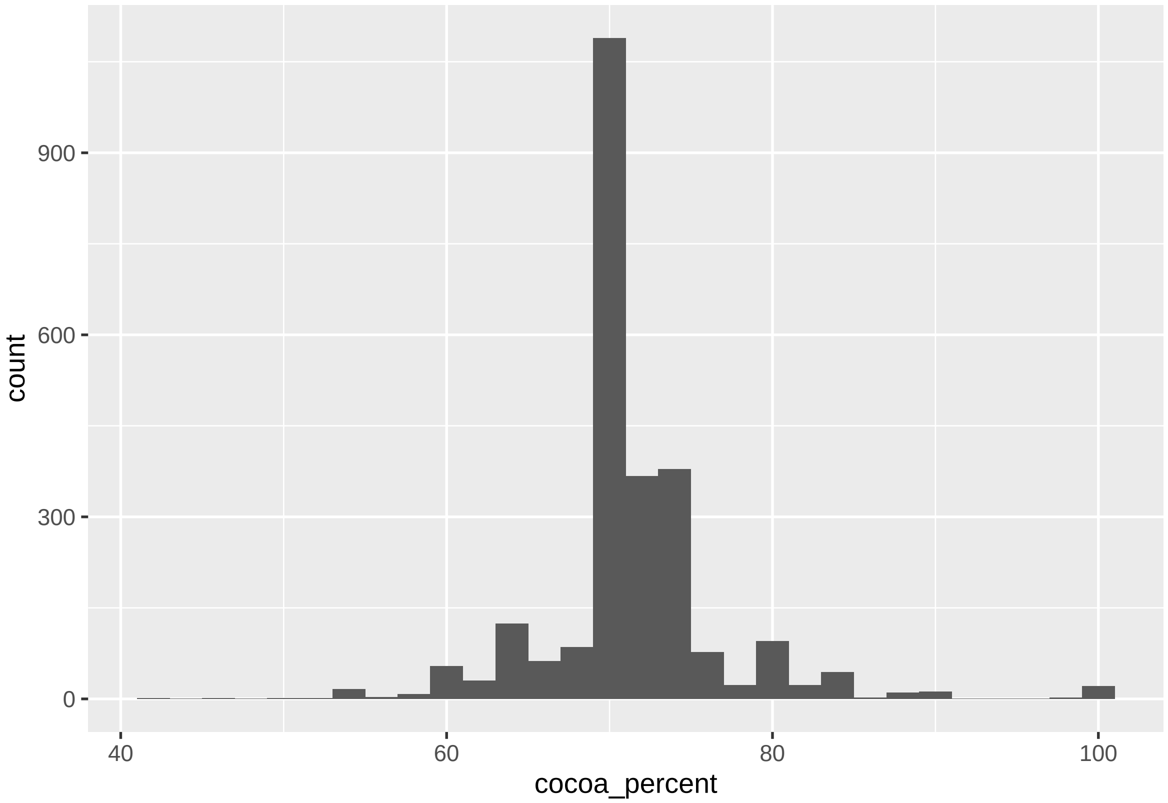
Shape of Ratings (already done)

…

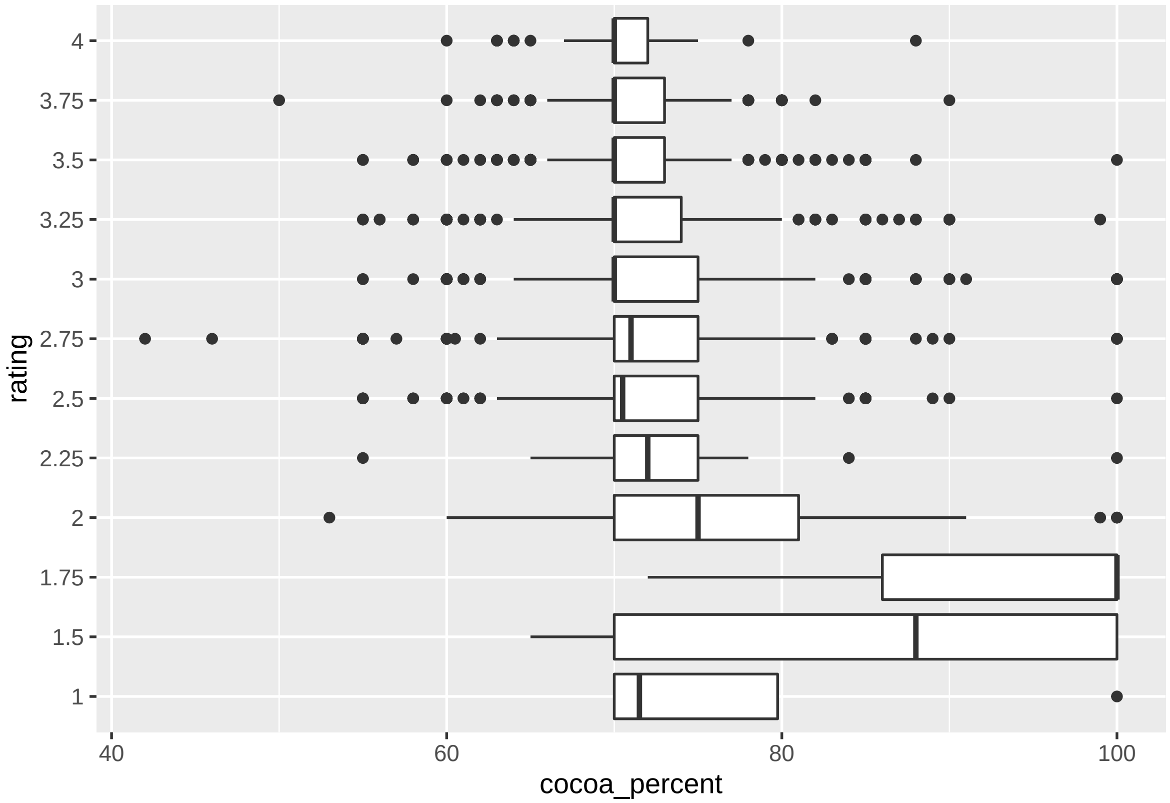
Cocoa Percent (Aimi)

```
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()
```

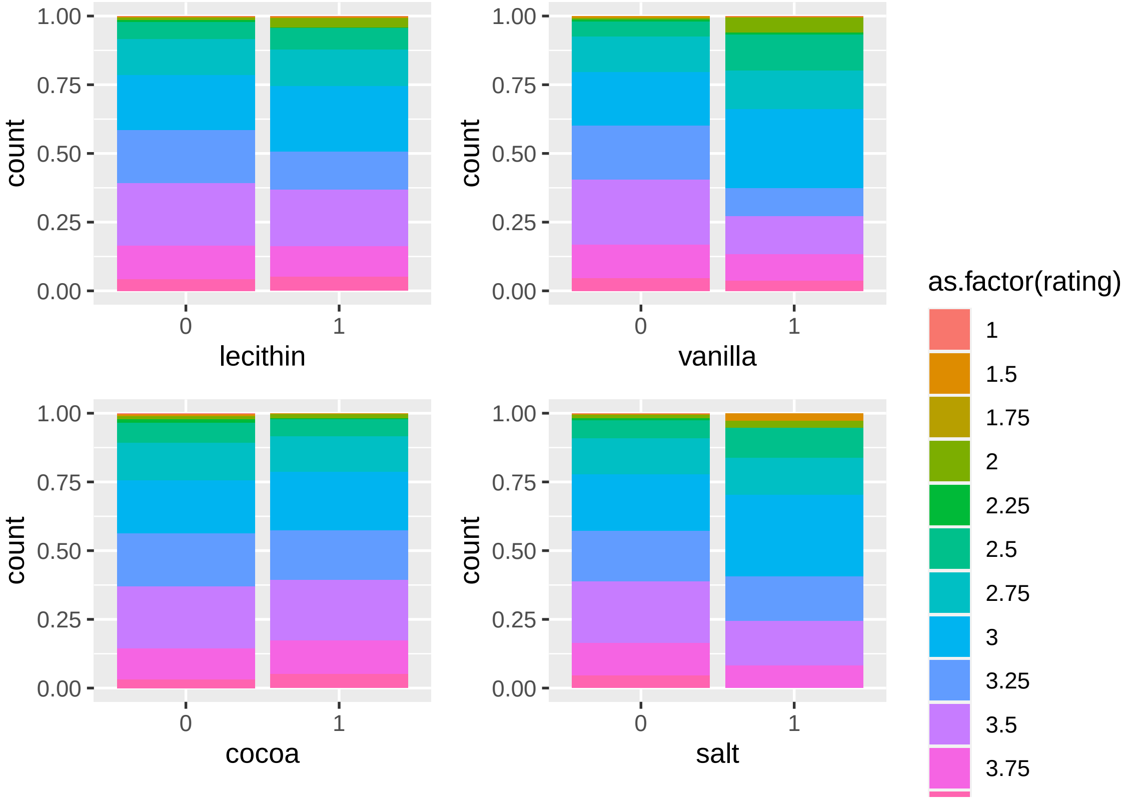


```
chocolate$rating <- as.numeric(chocolate$rating)
```

Ingredients (Nathan)

```
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)
```

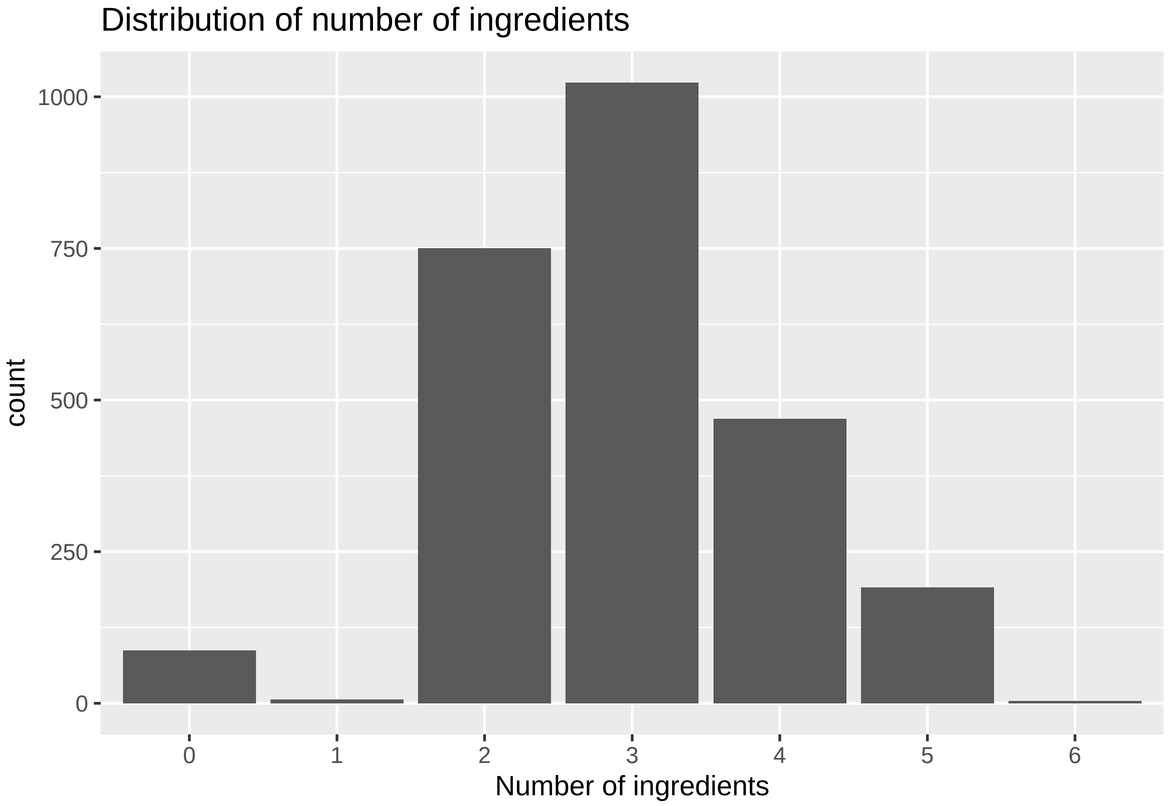


```
chocolate <- chocolate %>%
  mutate(
    num_ingres = if_else(is.na(ingredients), "0", str_sub(ingredients, 1, 1))
  )
```

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

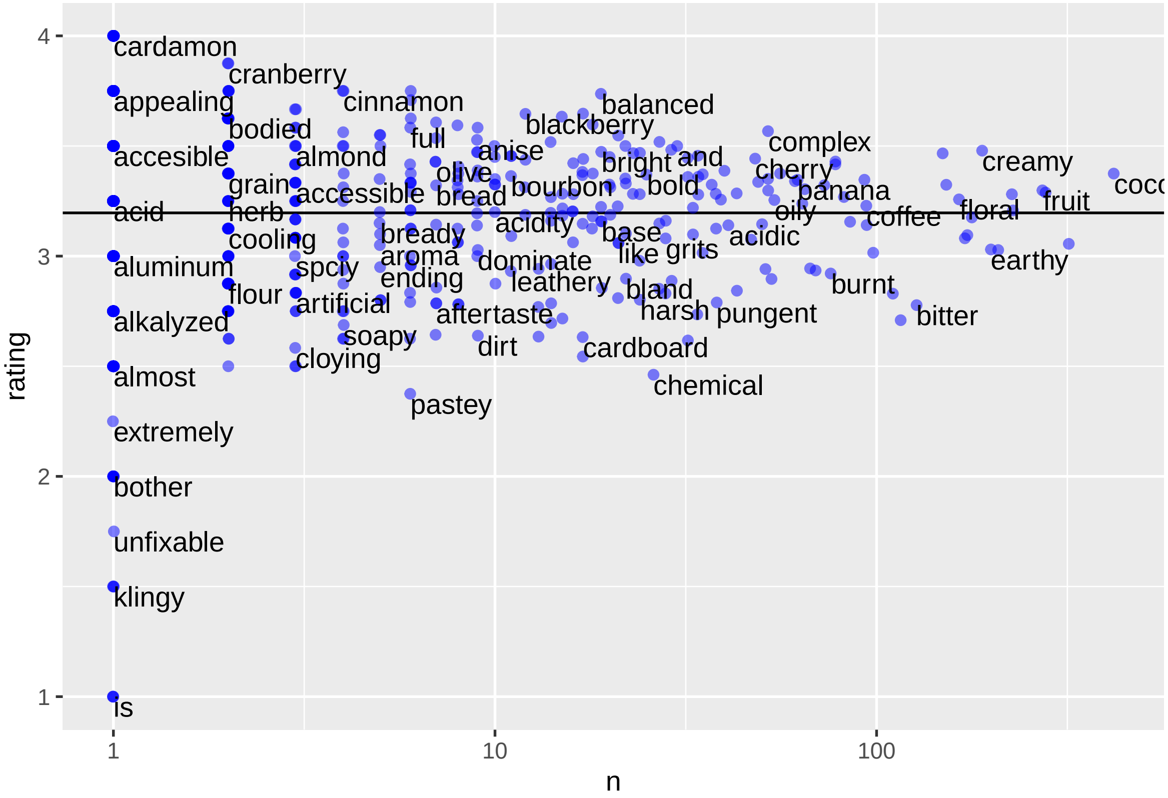
```
# A tibble: 1 x 1
      n
  <int>
1  2443
```

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



Most Memorable Characteristic (Aimi)

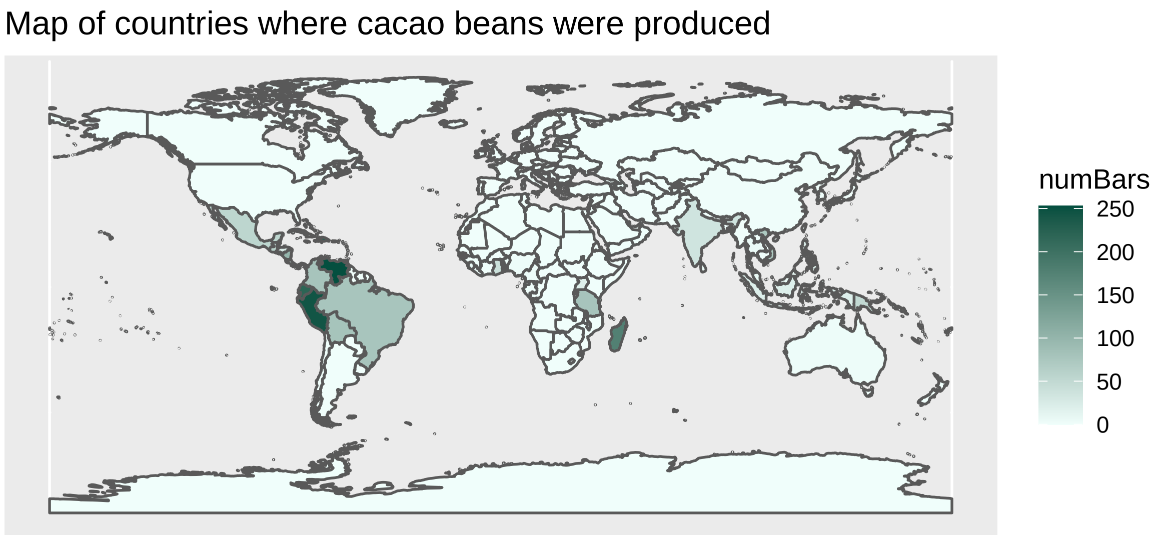
```
tidy_chocolate<- chocolate %>%
  unnest_tokens(word, most_memorable_characteristics)

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)

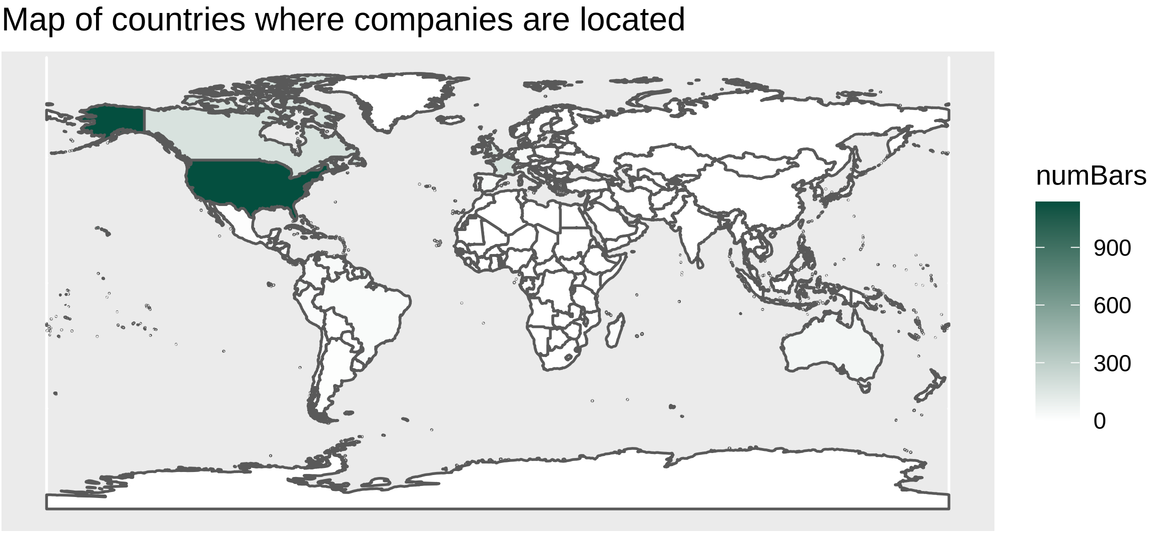
```
chocolate_modified <- chocolate %>%
  mutate(name_long = country_of_bean_origin) %>%
  group_by(name_long) %>%
  count(name_long)
```

```
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")
```

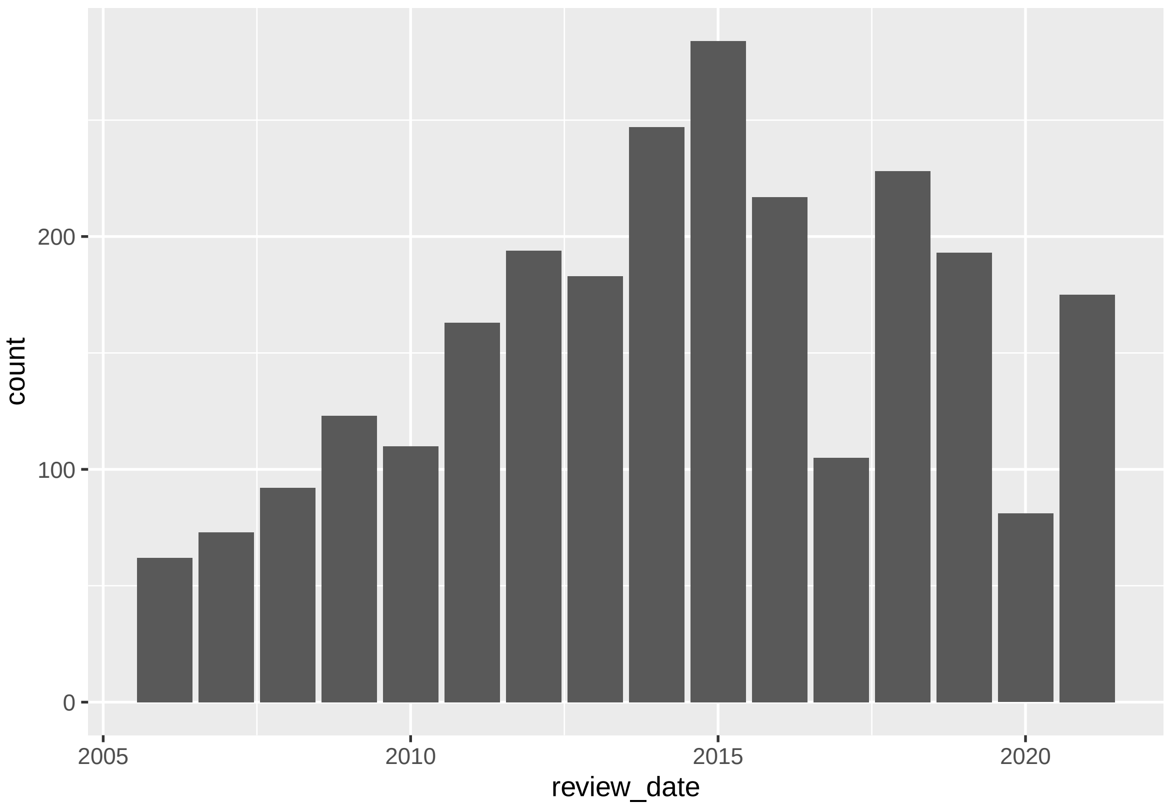


```
chocolate %>%
    count(company_location, sort = TRUE)
```

```
# 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()
```

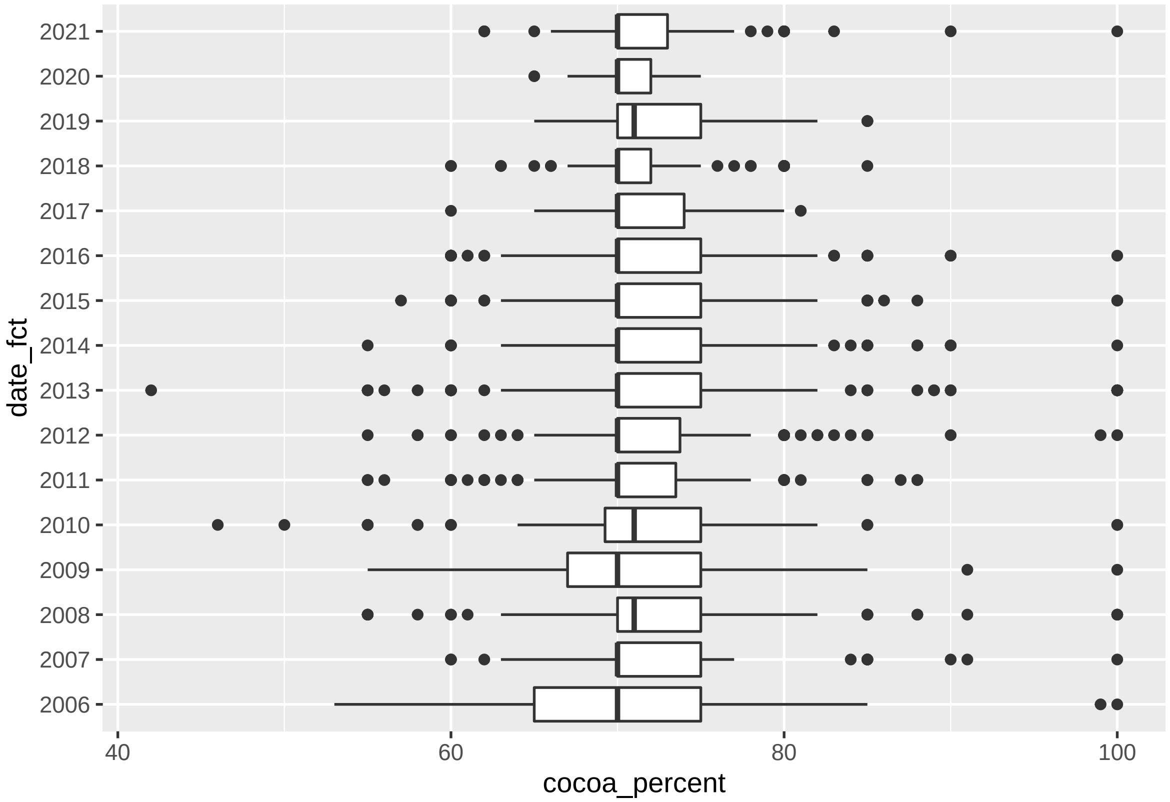


```
# statistics of review dates

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

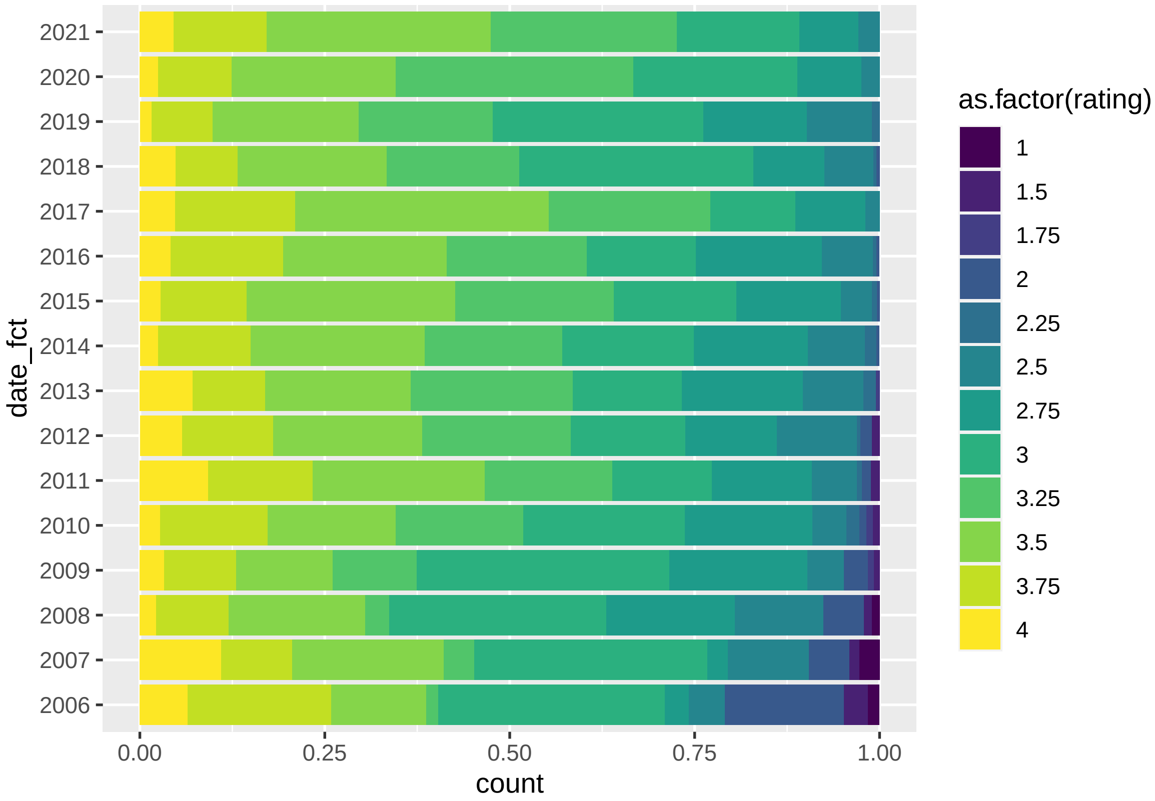
```
# A tibble: 1 x 3
   mean median    sd
  <dbl>  <dbl> <dbl>
1 2014.   2015  3.97
```

```
#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"
  ))
```

Data

The data dictionary can be found here.