# DSApps 2023 @ TAU: Final Project

## Exploratory Data Analysis

Sofia Prahia- 314710567, Noa Shaya – 318455961

2023-08-17

INSTALL REQUIRED PACKAGES

```
if (!require("tidyverse")) {install.packages("tidyverse") }
library(tidyverse)
if (!require("ggplot2")) { install.packages("ggplot2") }
library(ggplot2)
if (!require("stringr")) { install.packages("stringr") }
library(stringr)
if (!require("tidytext")) { install.packages("tidytext") }
library(tidytext)
if (!require("purrr")) { install.packages("purrr") }
library(purrr)
if (!require("dplyr")) { install.packages("dplyr") }
library(dplyr)
if (!require("tidyr")) { install.packages("tidyr") }
library(tidyr)
if (!require("tidymodels")) { install.packages("tidymodels") }
library(tidymodels)
if (!require("naniar")) { install.packages("naniar") }
library(naniar)
if (!require("jpeg")) { install.packages("jpeg") }
library(jpeg)
if (!require("purrr")) { install.packages("purrr") }
library(purrr)
if (!require("jpeg")) { install.packages("jpeg") }
library(jpeg)
if (!require("dplyr")) { install.packages("dplyr") }
library(dplyr)
if (!require("tidyr")) { install.packages("tidyr") }
library(tidyr)
if (!require("ggplot2")) { install.packages("ggplot2") }
library(ggplot2)
```

LOAD THE DATA BASES

```
food_train <- read_csv("data/food_train.csv")

food_test <- read_csv("data/food_test.csv")

nutrients <- read_csv("data/nutrients.csv")
```

```
food_nutrients <- read_csv("data/food_nutrients.csv")
```

LET'S EXPLORE THE DATA !

In this section, we will thoroughly analyze each column, investigating their individual characteristics and presenting thought-provoking questions.

First, let's catch a glimpse of how the data is structured.

```
head(food_train, 3)
```

```
## # A tibble: 3 x 8
##     idx brand          description    ingredients serving_size
serving_size_unit
##   <dbl> <chr>          <chr>          <chr>              <dbl> <chr>
## 1     1 brix chocolate milk chocolate sugar, coc~           28 g
## 2     2 target stores  frosted sugar~ sugar, enr~           38 g
## 3     3 target stores  white frosted~ sugar, enr~           30 g
## # i 2 more variables: household_serving_fulltext <chr>, category
<chr>
```

```
dim (food_train)
```

```
## [1] 31751     8
```

```
head(nutrients, 3)
```

```
## # A tibble: 3 x 3
##   nutrient_id name              unit_name
##         <dbl> <chr>             <chr>
## 1        1002 Nitrogen          G
## 2        1003 Protein           G
## 3        1004 Total lipid (fat) G
```

```
dim(nutrients)
```

```
## [1] 235   3
```

```
head(food_nutrients,3 )
```

```
## # A tibble: 3 x 3
##     idx nutrient_id amount
##   <dbl>       <dbl>  <dbl>
## 1     1        1087 143
## 2     1        1089   5.14
## 3     1        1104   0
```

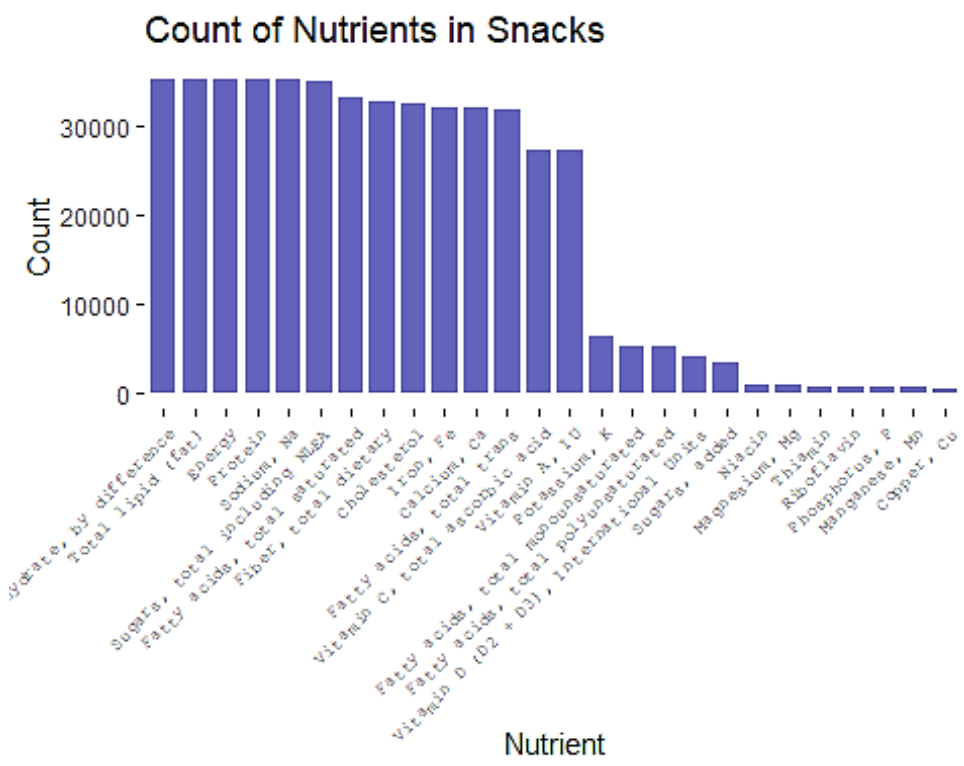
```
dim(food_nutrients)
```

```
## [1] 493054      3
```

NUTRIENT DISTRIBUTION:

Q1. Which nutrients are most commonly found in snacks?

```
nutrients_combine <- food_nutrients %>%
  left_join(nutrients, by="nutrient_id")

nutrients_count <- nutrients_combine%>%
  group_by(nutrient_id, name) %>%
  mutate(count = n()) %>%
  select(nutrient_id, name, count) %>%
  distinct()

nutrients_count %>%
  filter(count > 500) %>% #We filter out nutrients with low occurrence
  arrange(desc(count)) %>%
  ggplot(aes(x = reorder(name, -count), y = count)) +
  geom_bar(stat = "identity", fill = "blue4", alpha=0.6, width = 0.8) +
  xlab("Nutrient") +
  ylab("Count") +
  ggtitle("Count of Nutrients in Snacks") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1, family =
"mono", size = 7, color = "black"), panel.background =
element_rect(fill = "white"))
```



Q2: What are the top 14 nutrients that are most commonly found in snacks:

```
nutrients_count %>%
  filter(count > 500) %>%
  arrange(desc(count)) %>%
  head(14)
```

```
## # A tibble: 14 x 3
## # Groups:   nutrient_id, name [14]
##    nutrient_id name                            count
##          <dbl> <chr>                           <int>
##  1        1005 Carbohydrate, by difference    35264
##  2        1004 Total lipid (fat)              35263
##  3        1008 Energy                         35261
##  4        1003 Protein                        35245
##  5        1093 Sodium, Na                     35143
##  6        2000 Sugars, total including NLEA   35113
##  7        1258 Fatty acids, total saturated   33177
##  8        1079 Fiber, total dietary           32764
##  9        1253 Cholesterol                    32605
## 10        1089 Iron, Fe                       32127
## 11        1087 Calcium, Ca                    31991
## 12        1257 Fatty acids, total trans       31809
## 13        1162 Vitamin C, total ascorbic acid 27317
## 14        1104 Vitamin A, IU                  27304
```

Let's determine the categories they are associated with by utilizing the training data.

Q3: Are there any nutrients that are present in only a few snacks? if so let's check if they are related to a specific category

```
few_snacks <- nutrients_count %>% filter(count < 200)

few_snacks_data <- nutrients_combine %>%
  filter(nutrient_id %in% few_snacks$nutrient_id)

few_snacks_data %>%
  left_join(food_train, by = "idx")  %>%
  group_by(category) %>%
  select(category) %>%
  summarize(count = n()) %>%
  distinct()%>%
  arrange(desc(count))
```

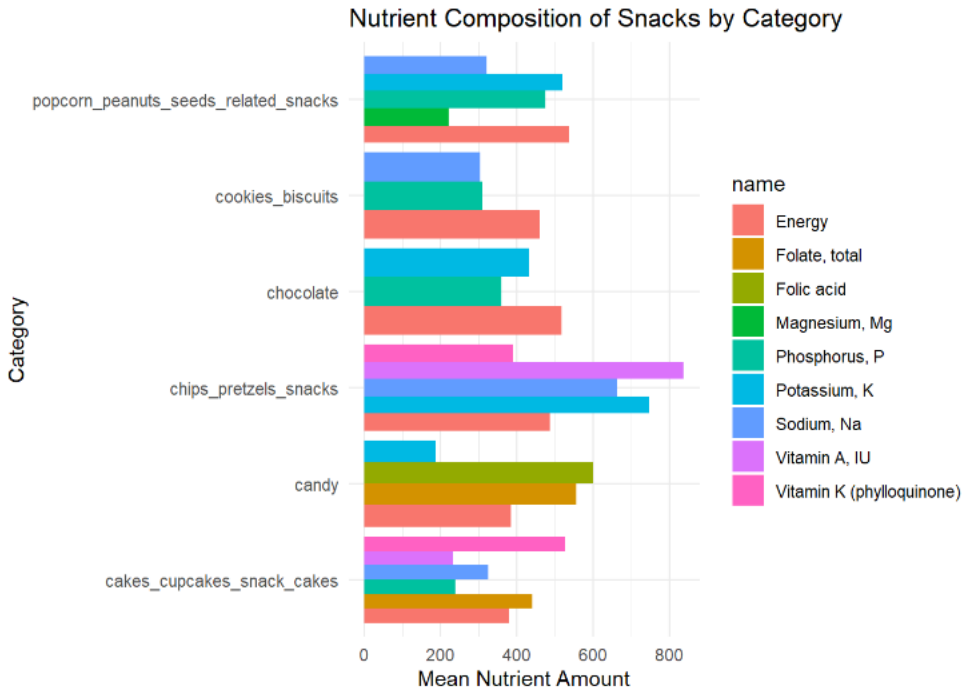
```
## # A tibble: 7 x 2
##   category                              count
##   <chr>                                 <int>
## 1 chips_pretzels_snacks                   243
## 2 popcorn_peanuts_seeds_related_snacks    239
## 3 cookies_biscuits                        102
## 4 <NA>                                     82
```

```
## 5 cakes_cupcakes_snack_cakes                49
## 6 candy                                     32
## 7 chocolate                                 27
```

We can see that most of the nutrients that are present in only a few snacks are related to chips_pretzels_snacks/popcorn_peanuts_seeds_related_snacks categories => meaning to the salty snacks.

Q4: Are there any specific categories that tend to have higher or lower levels of certain nutrients?

```
merged_food_data <- food_nutrients %>%
  left_join(nutrients, by = "nutrient_id") %>%
  inner_join(food_train, by = "idx")

nutrient_stats <- merged_food_data %>%
  group_by(category, name) %>%
  summarize(mean_amount = mean(amount))

#We will group nutrients that are similar or belong to the same
nutrient group because there are a lot of nutrients. Then, we will
filter only those nutrients with a mean amount greater than 180:

nutrient_stats <- nutrient_stats %>%
  mutate(name = ifelse(str_starts(name, "Fatty acids"), "Fatty acids",
ifelse(str_starts(name, "Carbohydrate"), "Carbohydrate",
ifelse(str_starts(name, "Fiber"),"Fiber", name)))) %>%
  filter(mean_amount>=180)

ggplot(nutrient_stats, aes(x = category, y = mean_amount, fill = name))
+
  geom_bar(stat = "identity", position = "dodge") +
  xlab("Category") +
  ylab("Mean Nutrient Amount") +
  ggtitle("Nutrient Composition of Snacks by Category") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))+
  theme_minimal() +
  coord_flip()
```




We observed that Folate is more commonly found in the Candy and Cakes, Cupcakes, Snack Cakes categories. The Cakes, Cupcakes, Snack Cakes categories also contain a significant amount of Vitamin K. In the Candy category, there is a high concentration of Folic Acid. The Chips, Pretzels & Snacks category stands out with substantial amounts of Potassium, Sodium, and Vitamin A. The energy content across all categories ranges from around 400 to 500. Additionally, the Cookies & Biscuits and Popcorn, Peanuts, Seeds & Related Snacks categories contain approximately 400 units of Potassium and Phosphorus.

BRANDS:

Q1: what are the 10 top brands:

```
food_train %>%
  group_by(brand) %>%
  summarize(count = n()) %>%
  select(brand, count) %>%
  distinct() %>%
  arrange(desc(count)) %>%
  head(10)
```

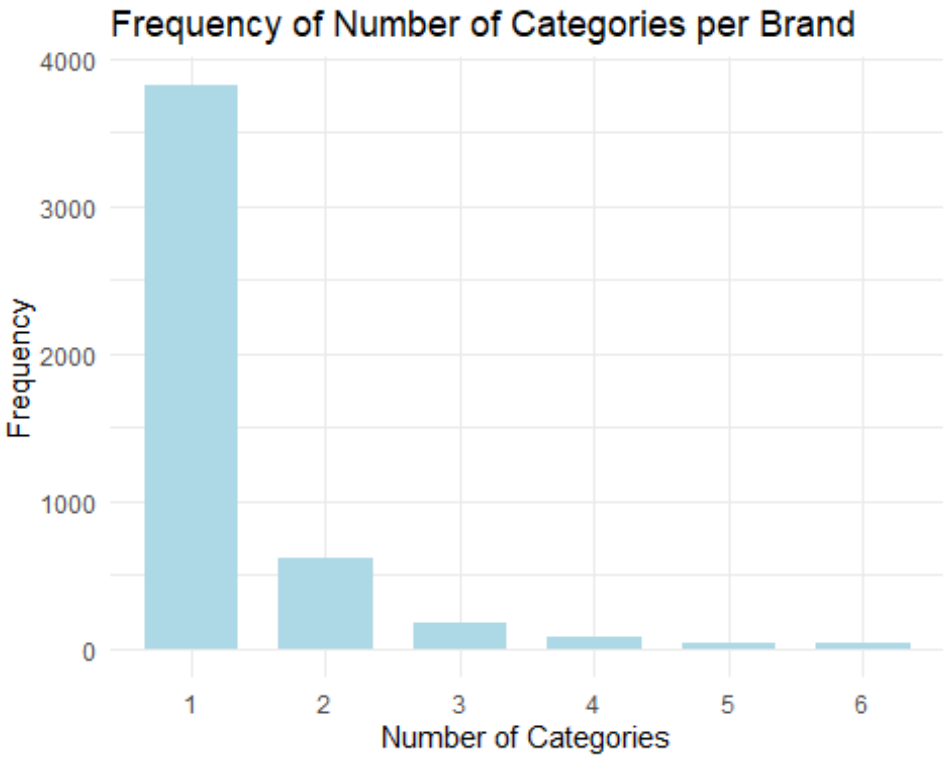
```
## # A tibble: 10 x 2
##    brand                    count
##    <chr>                    <int>
##  1 wal-mart stores, inc.      579
##  2 target stores              540
##  3 ferrara candy company      506
##  4 not a branded item         467
##  5 meijer, inc.               463
```

```
##  6 cvs pharmacy, inc.       342
##  7 the kroger co.           340
##  8 walgreens co.            336
##  9 topco associates, inc.   320
## 10 ahold usa, inc.          291
```

Q2: What is the number of different categories that are common for a brand?

The plan is to analyze how many distinct categories each brand has and visualize the frequency of occurrence for each number of categories. This will provide insights into the commonality of category diversity among brands.

```
brand_category_counts <- food_train %>%
  group_by(brand) %>%
  summarize(num_categories = n_distinct(category))

category_frequency <- brand_category_counts %>%
  count(num_categories)


ggplot(category_frequency, aes(x = as.factor(num_categories), y = n)) +
  geom_bar(stat = "identity", fill =  "lightblue", width = 0.7) +
  xlab("Number of Categories") +
  ylab("Frequency") +
  ggtitle("Frequency of Number of Categories per Brand") +
  theme_minimal() +
  scale_x_discrete(labels =
as.character(category_frequency$num_categories))
```




We can observe that most brands are associated with only one of the six categories. This suggests that the brand column could be helpful in predicting the category to which a product belongs. By leveraging the brand information, we can enhance our prediction accuracy and effectively classify products into their respective categories.

However, it is important to note that the database contain a wide variety of brands.

Q3: what is the number of brands in the train set? :

```
#number of brands in the train set:
food_train %>%
  distinct(brand) %>%
  n_distinct()

## [1] 4783

#The data reveals a wide variety of brands, suggesting that the
diversity within the brand information could potentially offer valuable
insights for our analysis.
```

Q4: Which brand dominates each category?

```
brand_category_counts <- food_train %>%
  group_by(category, brand) %>%
  summarize(count = n()) %>%
  ungroup()

#Now, let's identify the dominant brand for each category
```

```
brand_category_counts %>%
  group_by(category) %>%
  filter(count == max(count))
```

```
## # A tibble: 6 x 3
## # Groups:   category [6]
##   category                            brand                          count
##   <chr>                               <chr>                          <int>
## 1 cakes_cupcakes_snack_cakes          wal-mart stores, inc.            235
## 2 candy                               ferrara candy company            475
## 3 chips_pretzels_snacks               utz quality foods, inc.          146
## 4 chocolate                           lindt & sprungli (schweiz) ag    166
## 5 cookies_biscuits                    nabisco biscuit company          140
## 6 popcorn_peanuts_seeds_related_snacks meijer, inc.                    208
```

INGREDIENT ANALYSIS:

Q1: What are the most common ingredients found in snacks?

```
ingredient_counts <- food_train %>%
  mutate(ingredients = strsplit(ingredients, ", ")) %>%
  unnest(ingredients) %>%
  group_by(ingredients) %>%
  summarize(count = n()) %>%
  arrange(desc(count))

#Top 10 most common ingredients in snacks, along with their counts:

ingredient_counts %>%
  head(10)
```

```
## # A tibble: 10 x 2
##    ingredients  count
##    <chr>        <int>
##  1 sugar        21375
##  2 salt         18215
##  3 corn syrup    9968
##  4 water         8074
##  5 soy lecithin  8007
##  6 cocoa butter  7409
##  7 citric acid   7043
##  8 niacin        6779
##  9 riboflavin    5980
## 10 reduced iron  5808
```

Q2: What are the top 5 ingredients for each category?

```
food_train %>%
  mutate(ingredients = strsplit(ingredients, ", ")) %>%
  unnest(ingredients) %>%
  group_by(category, ingredients) %>%
  select(category, ingredients) %>%
  summarize(count = n()) %>%
```

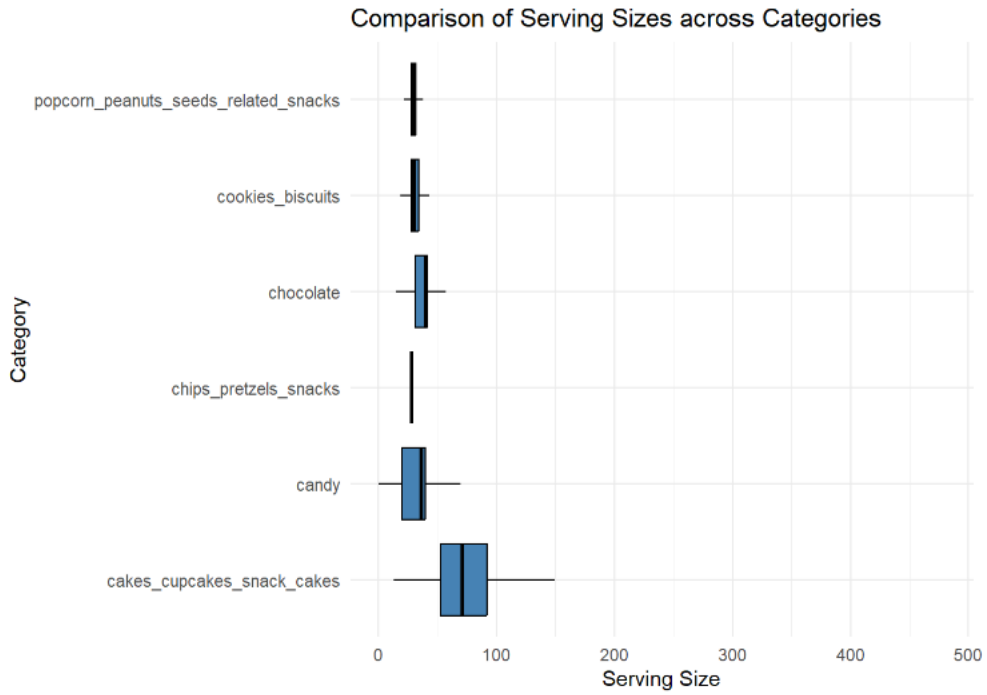
```
  distinct() %>%
  arrange(desc(count)) %>%
  slice_head(n = 5)
```

```
## # A tibble: 30 x 3
## # Groups:   category [6]
##    category                    ingredients  count
##    <chr>                       <chr>        <int>
##  1 cakes_cupcakes_snack_cakes salt          5202
##  2 cakes_cupcakes_snack_cakes water         4253
##  3 cakes_cupcakes_snack_cakes sugar         4002
##  4 cakes_cupcakes_snack_cakes niacin        2796
##  5 cakes_cupcakes_snack_cakes soy lecithin  2694
##  6 candy                      sugar         6031
##  7 candy                      corn syrup    4822
##  8 candy                      citric acid   3209
##  9 candy                      gelatin       1693
## 10 candy                      salt          1669
## # i 20 more rows
```

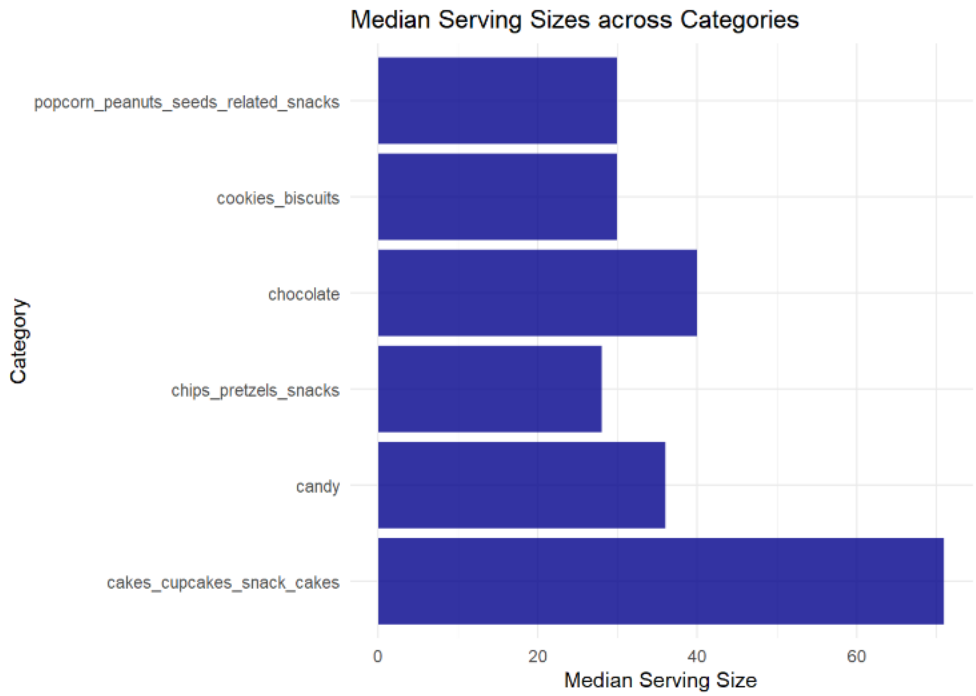
SERVING SIZE:

Q1: Are there any notable differences in serving sizes between categories?

```
# Calculate the median serving size for each category
category_serving_sizes <- food_train %>%
  group_by(category) %>%
  summarize(median_serving_size = median(serving_size))

# Create a boxplot to compare serving sizes across categories
ggplot(food_train, aes(x = category, y = serving_size)) +
  geom_boxplot(fill = "steelblue", color = "black", outlier.shape = NA)
+
  xlab("Category") +
  ylab("Serving Size") +
  ggtitle("Comparison of Serving Sizes across Categories") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  coord_cartesian(ylim = c(0, 170)) +
  theme_minimal()+
  coord_flip()
```




```r
#A plot to compare median serving sizes across categories
ggplot(category_serving_sizes, aes(x = category, y =
median_serving_size)) +
  geom_bar(stat = "identity", fill = "blue4", alpha = 0.8) +
  xlab("Category") +
  ylab("Median Serving Size") +
  ggtitle("Median Serving Sizes across Categories") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme_minimal() +
  coord_flip()
```

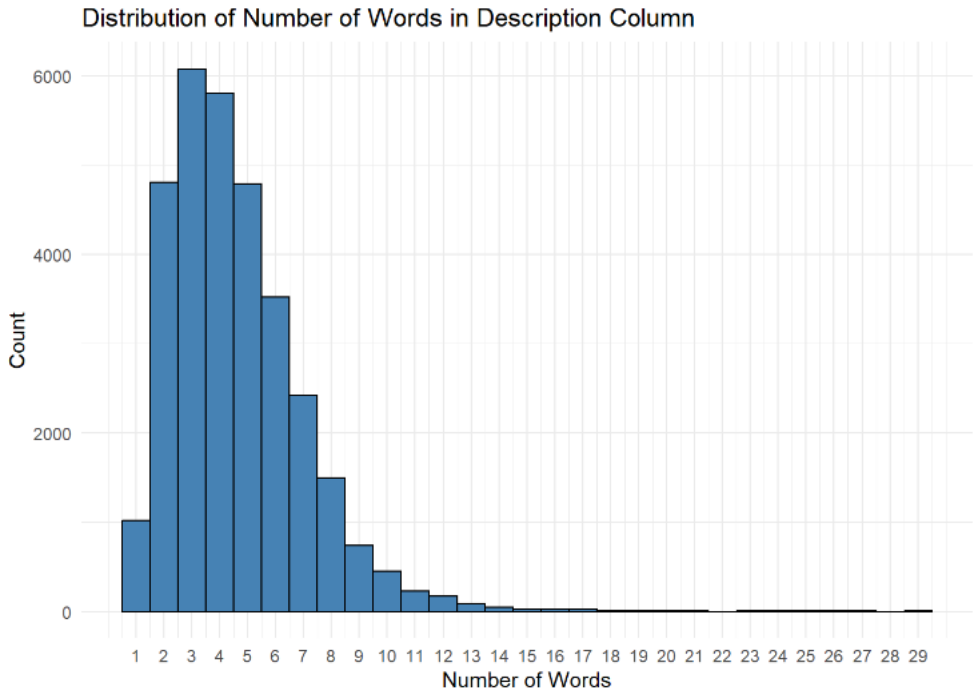



We can infer that the serving size for the Cakes, Cupcakes, Snack Cakes category is notably larger, with an average of around 70. In contrast, the serving sizes of the other categories have a median range of 30 to 40, indicating comparatively smaller portion sizes. Moreover by looking at the boxplot, we can see that the serving size range for Chips, Pretzels & Snacks, Cookies & Biscuits, and Popcorn, Peanuts, Seeds & Related Snacks is very narrow. This suggests that products within these categories generally have consistent serving sizes, providing a predictable portion for consumers.

DESCRIPTION:

Q1. What is the distribution of the number of words in the description column?

```
# Calculate the number of words in the description column
word_count <- food_train %>% mutate(words = str_count(description,
"\\S+"))

word_count%>%
  ggplot(aes(x = word_count$words)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "black") +
  labs(x = "Number of Words", y = "Count") +
  ggtitle("Distribution of Number of Words in Description Column") +
  scale_x_continuous(breaks = seq(min(word_count$words),
max(word_count$words), 1)) +
  theme(axis.text.x = element_text(angle = 0, hjust = 1, size = 8)) +
  theme_minimal()
```




We observe that the majority of products have descriptions containing approximately 2-6 words and the mean is around 4-5.This information could be useful in the future if we choose to rearrange the description column and use it to our prediction (as we will do later)

Q2: most common words used across snacks

```
# Tokenize the description column into individual words
word_freq_top_20 <- food_train %>%
  mutate(description = str_to_lower(description)) %>%
  unnest_tokens(word, description, token = "words") %>%
  count(word, sort = TRUE) %>%
  head(20)

ggplot(word_freq_top_20, aes(x = reorder(word, n), y = n)) +
  geom_bar(stat = "identity", fill = "steelblue", alpha = 0.8,  width =
0.8) +
  labs(x = "Word", y = "Frequency", title = paste("Top", 20, "Most
Common Words in Descriptions")) +
  theme(axis.text.x = element_text(angle = 45,  size=5)) +
  theme_minimal() +
  coord_flip()
```

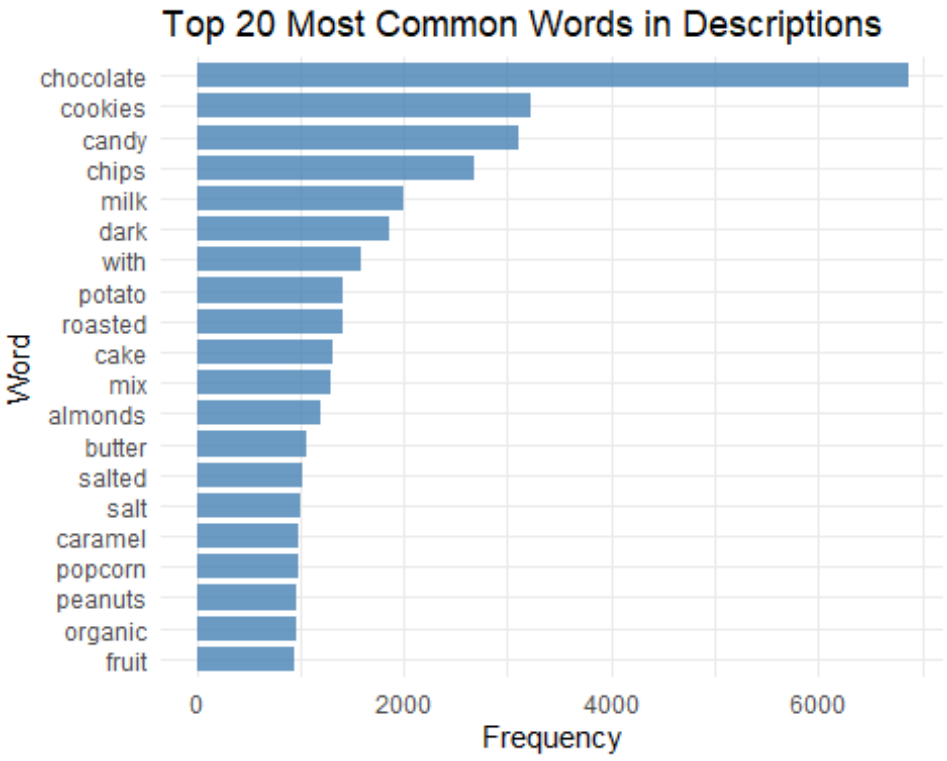



IMAGE ANALYSIS

Q1: What is the most common color in each category?

```
#We will calculate the average RGB in each category.
get_rgb <- function(path_image){
  bytes <- readBin(path_image, "raw", n = 2)
  if (identical(bytes, as.raw(c(0xFF, 0xD8)))){
    image <- readJPEG(path_image)
    avg_red <- mean(image[,,1])
    avg_green <- mean(image[,,2])
    avg_blue <- mean(image[,,3])
    return (c(avg_red, avg_green,avg_blue ))
  }
}
class_paths <- c(
  cakes_cupcakes_snack_cakes =
"C:/Users/noash/OneDrive/project/data/images_final/train/cakes_cupcakes
_snack_cakes",
  candy =
"C:/Users/noash/OneDrive/project/data/images_final/train/candy",
  chips_pretzels_snacks =
"C:/Users/noash/OneDrive/project/data/images_final/train/chips_pretzels
_snacks",
  chocolate =
"C:/Users/noash/OneDrive/project/data/images_final/train/chocolate",
  cookies_biscuits =
```

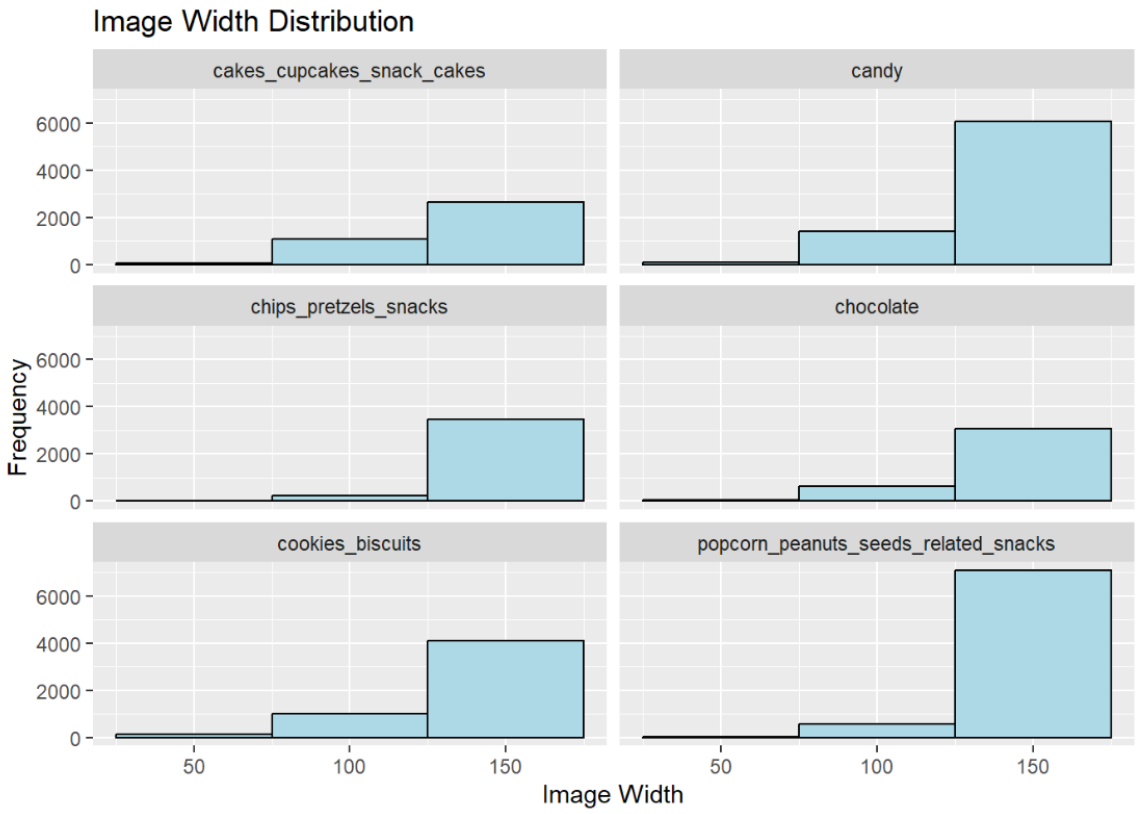
```
"C:/Users/noash/OneDrive/project/data/images_final/train/cookies_biscuits",
  popcorn_peanuts_seeds_related_snacks = "C:/Users/noash/OneDrive/project/data/images_final/train/popcorn_peanuts_seeds_related_snacks"
)

get_avg_rgb <- function(file_path){
  image_files <- list.files(file_path, pattern = ".jpg", full.names = TRUE)
  avg_rgbs_image_files <- lapply(image_files, get_rgb)
  overall_avg_rgb <- colMeans(do.call(rbind, avg_rgbs_image_files))
  return (overall_avg_rgb)
}
result_list <- map(class_paths, get_avg_rgb)

result_df <- data.frame(
  Category = names(class_paths),
  do.call(rbind, result_list)
)

new_column_names <- c("Category","Red", "Blue", "Green")
colnames(result_df) <- new_column_names
result_df <- result_df %>% select (Red, Blue, Green)
print(result_df*255)
```

```
##                                          Red     Blue    Green
## cakes_cupcakes_snack_cakes           186.3374 169.6247 157.6951
## candy                                196.0022 177.9274 167.4210
## chips_pretzels_snacks                192.3329 178.7419 161.7630
## chocolate                            189.5774 174.0342 166.4742
## cookies_biscuits                     194.3371 178.3446 165.9980
## popcorn_peanuts_seeds_related_snacks 191.1477 178.7284 167.9601
```

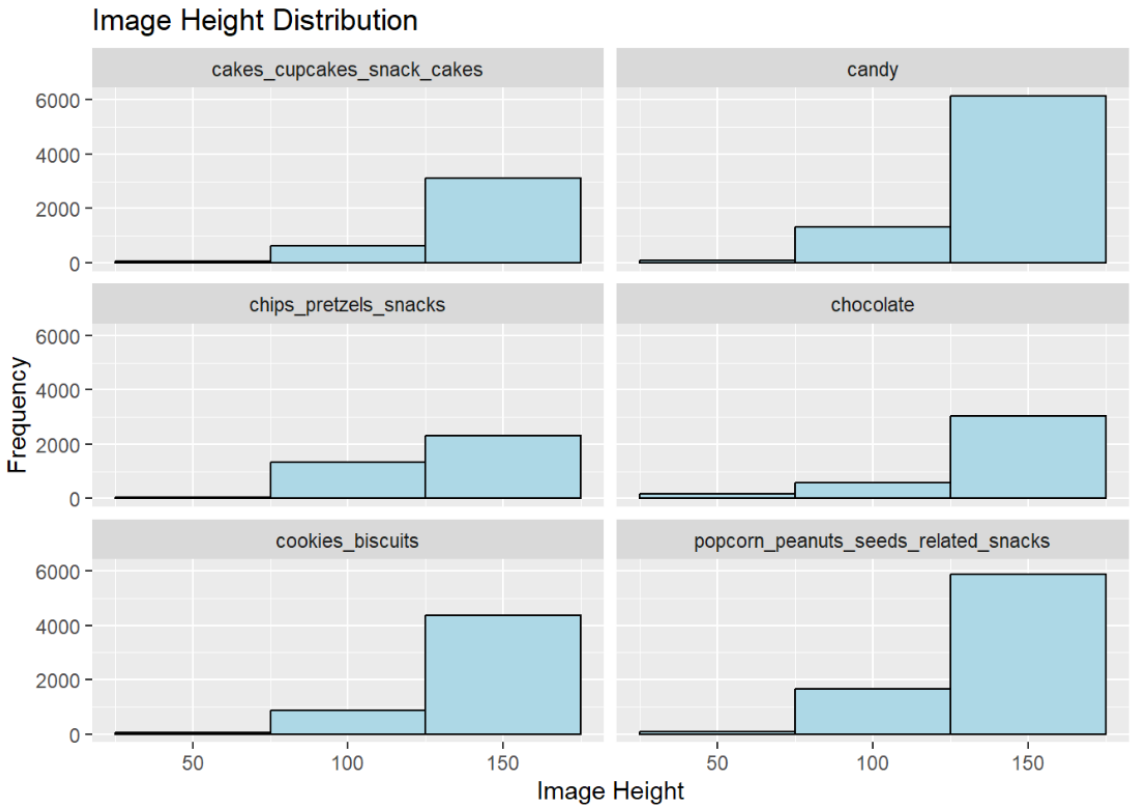
Q2: Analyze and visualize the distribution of image sizes within different categories of snacks

```
process_folder <- function(folder_path) {
  image_paths <- list.files(folder_path, full.names = TRUE)

  image_data <- lapply(image_paths, function(image_path) {
    tryCatch({
      img <- readJPEG(image_path)
      dimensions <- dim(img)[1:2]
      return(data.frame(path = image_path, width = dimensions[1], height = dimensions[2]))
    }, error = function(e) {
      return(NULL)
    })
  }) %>%
```

```
    bind_rows()

  image_data <- image_data[!is.na(image_data$path), ]

  return(image_data)
}

all_image_data <- lapply(class_paths, process_folder)

combined_image_data <- bind_rows(all_image_data, .id = "Category")

ggplot(combined_image_data, aes(x = width)) +
  geom_histogram(binwidth = 50, fill = "lightblue", color = "black") +
  labs(title = "Image Width Distribution", x = "Image Width", y =
"Frequency") +
  facet_wrap(~Category, ncol = 2)
```




```
ggplot(combined_image_data, aes(x = height)) +
  geom_histogram(binwidth = 50, fill = "lightblue", color = "black") +
  labs(title = "Image Height Distribution", x = "Image Height", y =
"Frequency") +
  facet_wrap(~Category, ncol = 2)
```




```
summary(combined_image_data)
```

```
##    Category            path               width           height     
##  Length:31707       Length:31707       Min.   : 38.0   Min.   : 30.0  
##  Class :character   Class :character   1st Qu.:140.0   1st Qu.:140.0  
##  Mode  :character   Mode  :character   Median :140.0   Median :140.0  
##                                        Mean   :133.4   Mean   :135.2  
##                                        3rd Qu.:140.0   3rd Qu.:140.0  
##                                        Max.   :140.0   Max.   :162.0
```

SUMMARY

At the beginning, we explored the data. We wanted to know the dimension of the data bases, what kinds of information it held (numeric or text), the distribution of each column values, and the characteristics of the columns.

Nutrients: We aimed to understand the popular nutrients and the categories they are associated with.

Brands: Our focus was on determining the count of distinct brands and identifying the dominant brand within each category. Additionally, we examined the relationship between each brand and its connection to one or more categories. Interestingly, our findings revealed that the majority of brands (approximately 80%) are associated with just one category. Consequently, leveraging brand-related information to predict the category appears to be a sensible approach.

Ingredients: We examined the common ones to see if they might indicate a specific category.

Serving Size: The serving size could indicate the type of snack. We observed significant differences in serving sizes among certain categories. For instance, "cakes_cupcakes_snack_cakes" notably has a larger serving size compared to other categories.

Description: We've recognized the importance of the description column. It contains important keywords that assist the model in predicting the category accurately.

Image processing: We wanted to explore the colors in the images and see what is the most common color in each category. We also checked the image sizes within different categories of items.