DSApps 2022 @ TAU: Final Project

Part 1 - Exploratory Data Analysis

Sergei Dakov

load packages

```
library(tidyverse)
library(ggplot2)
library(ggmosaic)
```

read in data

```
nutrients <-read.csv(file="data/nutrients.csv")
food_nutrients <- read.csv(file="data/food_nutrients.csv")
train <- read.csv(file="data/food_train.csv")
test <- read.csv(file="data/food_test.csv")
```

glimpse data

```
glimpse(food_nutrients)
```

```
## Rows: 493,054
## Columns: 3
## $ idx         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2~
## $ nutrient_id <int> 1087, 1089, 1104, 1162, 1003, 1004, 1005, 1008, 2000, 1079~
## $ amount      <dbl> 143.00, 5.14, 0.00, 0.00, 7.14, 35.71, 53.57, 536.00, 42.8~
```

```
glimpse(nutrients)
```

```
## Rows: 235
## Columns: 3
## $ nutrient_id <int> 1002, 1003, 1004, 1005, 1007, 1008, 1009, 1010, 1011, 1012~
## $ name        <chr> "Nitrogen", "Protein", "Total lipid (fat)", "Carbohydrate,~
## $ unit_name   <chr> "G", "G", "G", "G", "G", "KCAL", "G", "G", "G", "G", "G", ~
```

```
glimpse(train)
```

```
## Rows: 31,751
## Columns: 8
## $ idx                       <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, ~
## $ brand                     <chr> "brix chocolate", "target stores", "target ~
## $ description               <chr> "milk chocolate", "frosted sugar cookies", ~
## $ ingredients               <chr> "sugar, cocoa butter, whole milk, chocolate~
```

```
## $ serving_size               <dbl> 28.0, 38.0, 30.0, 40.0, 40.0, 36.0, 28.0, 2~
## $ serving_size_unit          <chr> "g", "g", "g", "g", "g", "g", "g", "g", "g"~
## $ household_serving_fulltext <chr> "1 onz", "1 cookie", "2 cookies", "5 pieces~
## $ category                   <chr> "chocolate", "cookies_biscuits", "cookies_b~
```

shorter names for the categories

```
short_names <- c("Cakes","Candy","Chips","Chocolate","Cookies","Nuts")
```

the pizza is a salad conundrum - vitamins

In recent years, the US has been actively attempting to make the diets of its residents, especially children healthier and cut down on heavily produced snack. One debate that came out of it is whether a pizza could be qualified as a salad as it contains a large amount of vegetable ingredients. in a similar vein since many of the snacks do contain many ingredients from fruit and vegetables (some, such as chips or popcorn are predominately out of ingredients from a vegetable origin) to test that comparison, we will use some of the main reasons for importance of fruit and vegetables - vitamins.

find which vitamins are available, and get their id

```
vitamins <- nutrients %>% filter(str_detect(name,"Vitamin"))  %>% filter(!str_detect(unit_name,"IU")) %>
```

find how many foodstuffs contain these vitamins

```
food_nutrients_wider <- food_nutrients %>% pivot_wider(names_from = nutrient_id,values_from = amount,val
#food_nutrients_vitamin <- food_nutrients_wider %>% select(c(idx,as.symbol(vitamins)))
food_nutrients %>% filter(nutrient_id%in%vitamins) %>% group_by(nutrient_id) %>% summarise(n=n()) %>% pu
nutrients %>% filter(nutrient_id %in% vit_present) %>% pull(name) -> vit_names
vit_present <- as.character(vit_present)
vit_contain <- food_nutrients_wider %>% select(idx,all_of(vit_present)) %>%
    rename_at(vars(vit_present),~vit_names)
```

```
## Note: Using an external vector in selections is ambiguous.
## i Use ‘all_of(vit_present)‘ instead of ‘vit_present‘ to silence this message.
## i See <https://tidyselect.r-lib.org/reference/faq-external-vector.html>.
## This message is displayed once per session.
```

get how many snacks in each category have vitamins

```
pull_indexes <- function(column) {
  vit_contain %>% filter(!!as.symbol(column) > 0) %>% pull(idx) -> res
  return(res)
  }
snacks_list <- lapply(vit_names, pull_indexes)
total_group <- train %>% group_by(category) %>% summarise(n=n())
```

prepare the data - some vitamins appear under different labels in different products, those are unified

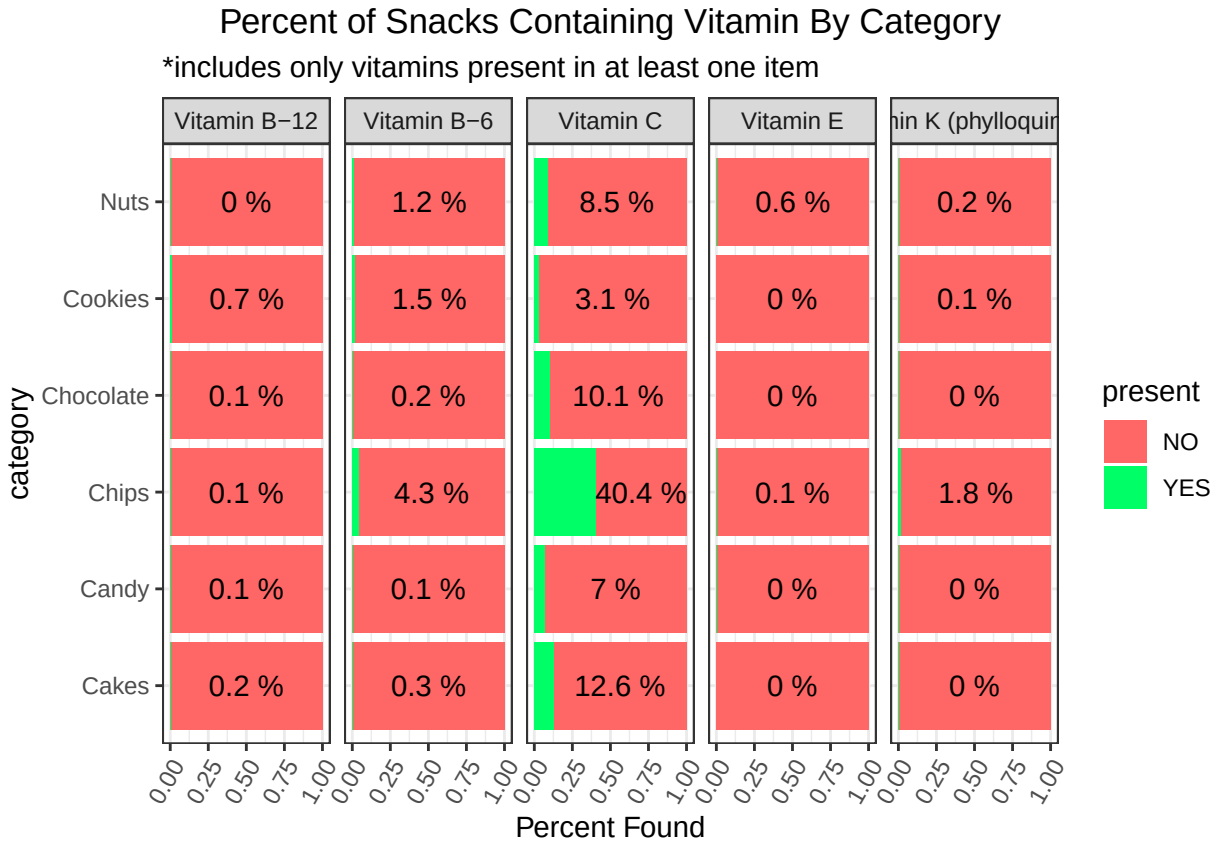
```
vit_contain <- vit_contain %>%
  mutate(across(where(is.numeric),~replace_na(.,0))) %>%
  mutate(`Vitamin E`=`Vitamin E`+`Vitamin E (alpha-tocopherol)`)%>%
  select(-`Vitamin E (alpha-tocopherol)`)%>%
  rename('Vitamin C'=`Vitamin C, total ascorbic acid`) %>%
  mutate(across(starts_with("Vitamin"),~ifelse(.x>0,TRUE,FALSE)))

plot_vit <- vit_contain %>% pivot_longer(starts_with("Vitamin"),values_to = "present")  %>% left_join(ti
plot_vit %>% group_by(category,name) %>% summarize(percent = mean(present))-> label_vec
```

```
## `summarise()` has grouped output by 'category'. You can override using the
## `.groups` argument.
```

plot the distribution of the different vitamins across the different categories of snacks (only vitamins where at least one instance is present)

```
  plot_vit %>% ggplot() +
  facet_grid(~name)+
  geom_bar(aes(x=category,fill=present),position = "fill")+
  geom_text(data=label_vec, aes(x=category,y=0.5,label=paste(round(label_vec$percent,3)*100,"%")),size =
  scale_x_discrete(labels=short_names) +
  coord_flip() +
  scale_fill_manual(values=c("#FF6666","#00FF66"),labels=c("NO","YES")) +
  theme_bw()+
  labs(y="Percent Found",title = "Percent of Snacks Containing Vitamin By Category",subtitle = "*includ
    theme(plot.title = element_text(hjust = 0.5),axis.text.x = element_text(angle = 60, hjust=1))
```



we can see from the plot that in general snacks are not a good source of vitamins except for chips and vitamin c (which makes sense as potatoes are known to be rich in vitamin c)

Stuck on a Desert Island - Nutritional Balance

The FDA proposes recommendations for the daily intake of many nutrients, we will focus on the major ones:Calories, Protein, Fat, Fibers, Sodium, Cholesterol, Trans Fat, Lipid Fat, and Sugars. the daily intake recommendations are based on a 2000 calorie per day diet. using these guidelines we will look at the general trends of nutritional balance among the categories as well as help answer the question "if you were to be on a desert island and could only have one food item, what would it be?", i.e. which of the snacks is the most nutritionally balanced (in the above categories at least)

find the nutritional indexes of the required nutrients

```
nut_index <- c(1008,1004,1257,1258,1079,1253,1003,1093,2000)
nut_index <- sort(nut_index)
```

get their names as they appear in the table, and clean them up

```
nut_names <- nutrients %>% filter(nutrient_id %in% nut_index) %>% pull(name) %>% str_replace_all(" ","_
```

recommended daily values (as per the FDA)

```
daily_values <- c(50,78,2000,28,2300,300,2,20,90)
names(daily_values) <- nut_names
```

find the quantities of said nutrients in the data

```
nut_quantities <- food_nutrients_wider %>% select(idx,as.character(nut_index))
snack_name_id <- train %>% unite(long_name,c('brand','description')) %>% select(idx,long_name,category)
```

calculate the nutrients against daily values (as percents), note calories is not present as it is being used as a benchmark (2000 calorie diet)

```
nut_quantities <-  merge(nut_quantities,snack_name_id,'idx','idx') %>%
  rename_at(vars(as.character(nut_index)),~nut_names) %>%
  filter(Energy!=0) %>%
  mutate(quantity = 2000/Energy) %>%
  mutate(across(all_of(nut_names),~.x*quantity)) %>%
  mutate(protein_vs_dv = (Protein-daily_values[1])/daily_values[1],
         fat_vs_dv = (Total_lipid_fat - daily_values[2])/daily_values[2],
         fiber_vs_dv = (Fiber_total_dietary-daily_values[4])/daily_values[4],
         sodium_vs_dv = (Sodium_Na - daily_values[5])/daily_values[5],
         cholesterol_vs_dv = (Cholesterol - daily_values[6])/daily_values[6],
         trans_vs_dv = (Fatty_acids_total_trans - daily_values[7])/daily_values[7],
         saturated_vs_dv = (Fatty_acids_total_saturated - daily_values[8])/daily_values[8],
         sugars_vs_dv = (Sugars_total_including_NLEA - daily_values[9])/daily_values[9]) %>%
  filter(!if_all(ends_with("dv"),~.x==(-1)))
```

calculate mean square distance from ideal (0% difference from daily value)

```
nut_vs_dv <- nut_quantities %>% select(ends_with("dv")) %>% mutate(across(everything(),~.x^2))
nut_vs_dv %>% mutate(MSE = rowMeans(nut_vs_dv)) %>% pull(MSE) -> mse_vec
nut_quantities <- nut_quantities %>% cbind(MSE=mse_vec)
```

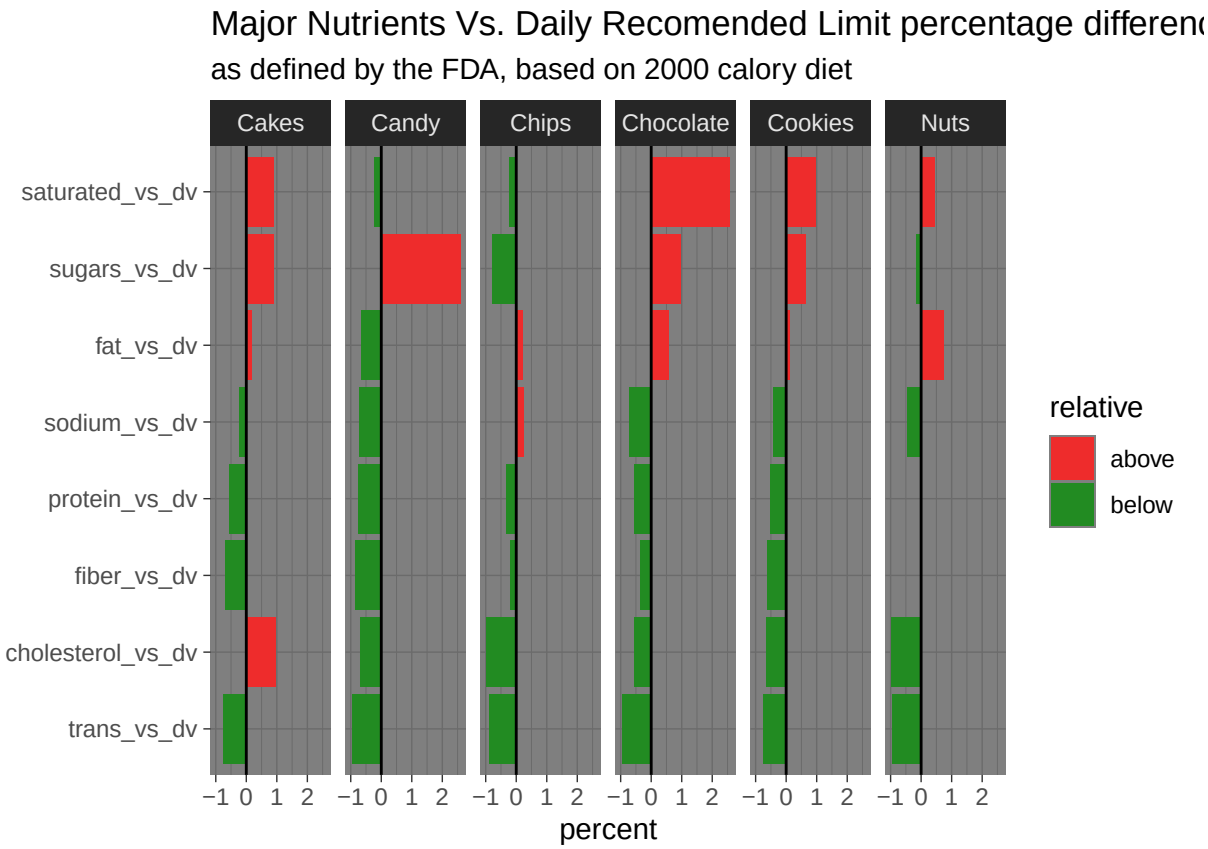
note the best balanced snacks, also since the daily values serve as recommended upper limits, get the snacks that violate these limits the least

```
best_dv <- nut_quantities %>% arrange(MSE) %>% head(5) %>% select(long_name,ends_with("dv"),MSE,category
under_cap <- nut_quantities %>% mutate(overall =  rowSums(across(ends_with("dv")))) %>% arrange(overall)
```

plot the average behaviour for each category

```
nut_quantities <- nut_quantities %>% group_by(category) %>%
  summarise(across(ends_with("dv"),~mean(.x)))

names(short_names) <- nut_quantities$category
nut_quantities %>% pivot_longer(cols=(-1), names_to = "nutrient") %>% mutate(relative = ifelse(value<=0
  ggplot() +
    facet_grid(~category,labeller = as_labeller(short_names),space = "free") +
  geom_bar(stat = "identity",aes(x=reorder(nutrient,value),y=value,fill=relative))+
  coord_flip() +
    geom_hline(aes(yintercept = 0))+
    scale_fill_manual(values=c("firebrick2","forestgreen")) +
    labs(y="percent",x="",title="Major Nutrients Vs. Daily Recomended Limit percentage difference",subti
    theme_dark()
```



we can see that most snacks do not hit the cap on sodium, fiber and trans fat. also, candy for example contains very few nutrients besides sugar on average

next, let's look at the top 5 most balance items and their category, as well as the distribution

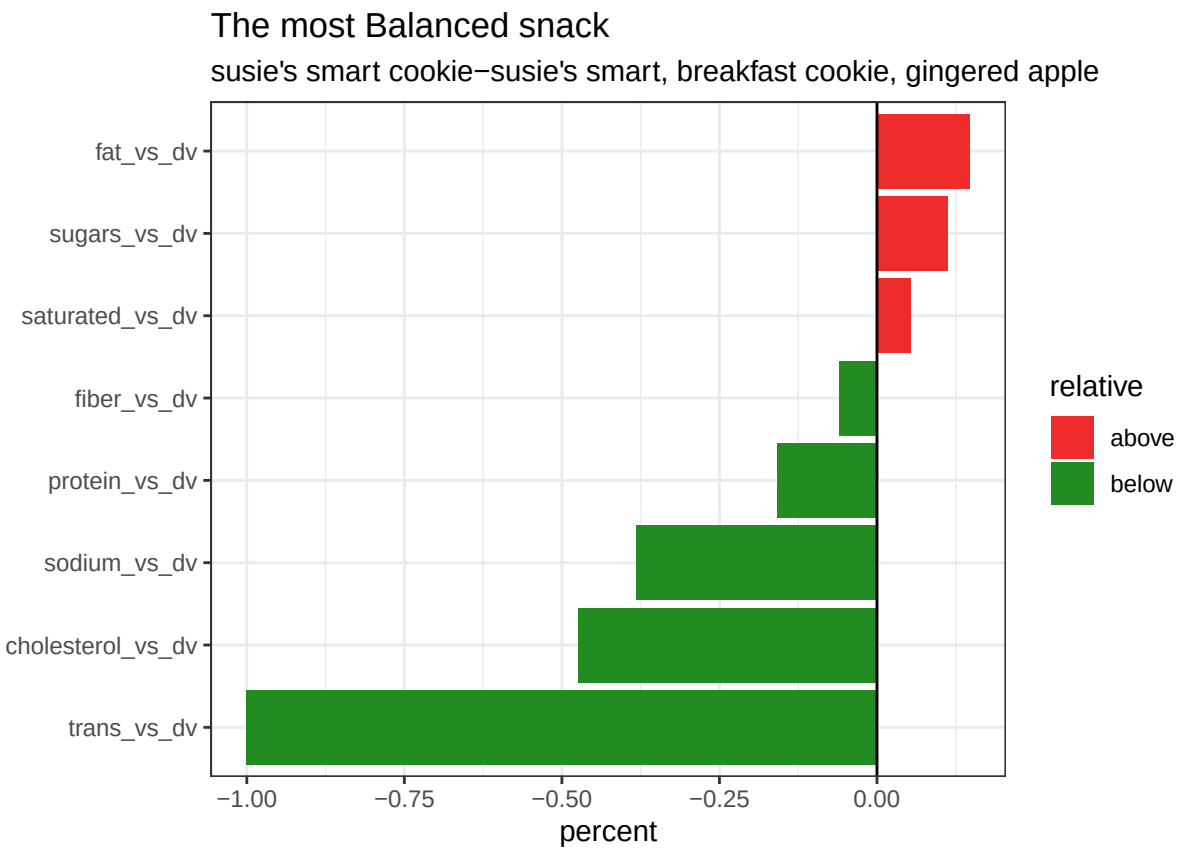
```
print(best_dv %>% select(long_name,category))
```

```
##                                                                long_name
## 1 susie's smart cookie_susie's smart, breakfast cookie, gingered apple
## 2                                 aryzta, llc_cinnamon crumb loaf cake
## 3                 stella d'oro biscuit co inc_breakfast treats cookies
## 4        back to nature foods company, llc_snack well's, biscuit thins
## 5                buttercup bakeries inc_danish coffeecake cream cheese
##                     category
## 1           cookies_biscuits
## 2 cakes_cupcakes_snack_cakes
## 3           cookies_biscuits
## 4           cookies_biscuits
## 5 cakes_cupcakes_snack_cakes
```

the most balanced snacks are baked goods, gakes or cookies. this makes sense as this type of goods tends to have the largest ammount of varied ingredients needed in manufacture

```
best_dv %>% head(1)  %>% pivot_longer(cols = ends_with("dv"),names_to = "nutrient") %>% mutate(relative
  ggplot() +
  geom_bar(stat = "identity",aes(x=reorder(nutrient,value),y=value,fill=relative))+
```

```
  coord_flip() +
  geom_hline(aes(yintercept = 0))+
  scale_fill_manual(values=c("firebrick2","forestgreen")) +
  labs(x="",y="percent",title="The most Balanced snack",subtitle = str_replace(best_dv$long_name[1],"_"
  theme_bw()
```



the thing to not here is that this product (gingered apple cookies) is mostly below the daily values, especially on trans fat and chholesterol, and while still not perfectly balanced probably

The Serving Dilemma

as concern with healthier eating grows, people became more aware and selective of the foods they consume. To aid them in this endeavor manufacturers now must put on the package a list of nutritional values in a clear manner. in this field two ways of thought have appeared, one favored in Europe labels the nutrients per 100 grams of product, the other favoured in the US is using a serving size. the proponents of the serving size method argue that using a unit of mass is complicated as it requires the consumer to posses measuring equipment thus is non-intuitive to the common user, while a common serving size such as “one package” is easier to quickly parse. on the downside, there s concern that manufacturers would tweak their presented serving size to present a more favorable image of their product. in his part we will explore how consistent and clear are serving sizes and labels

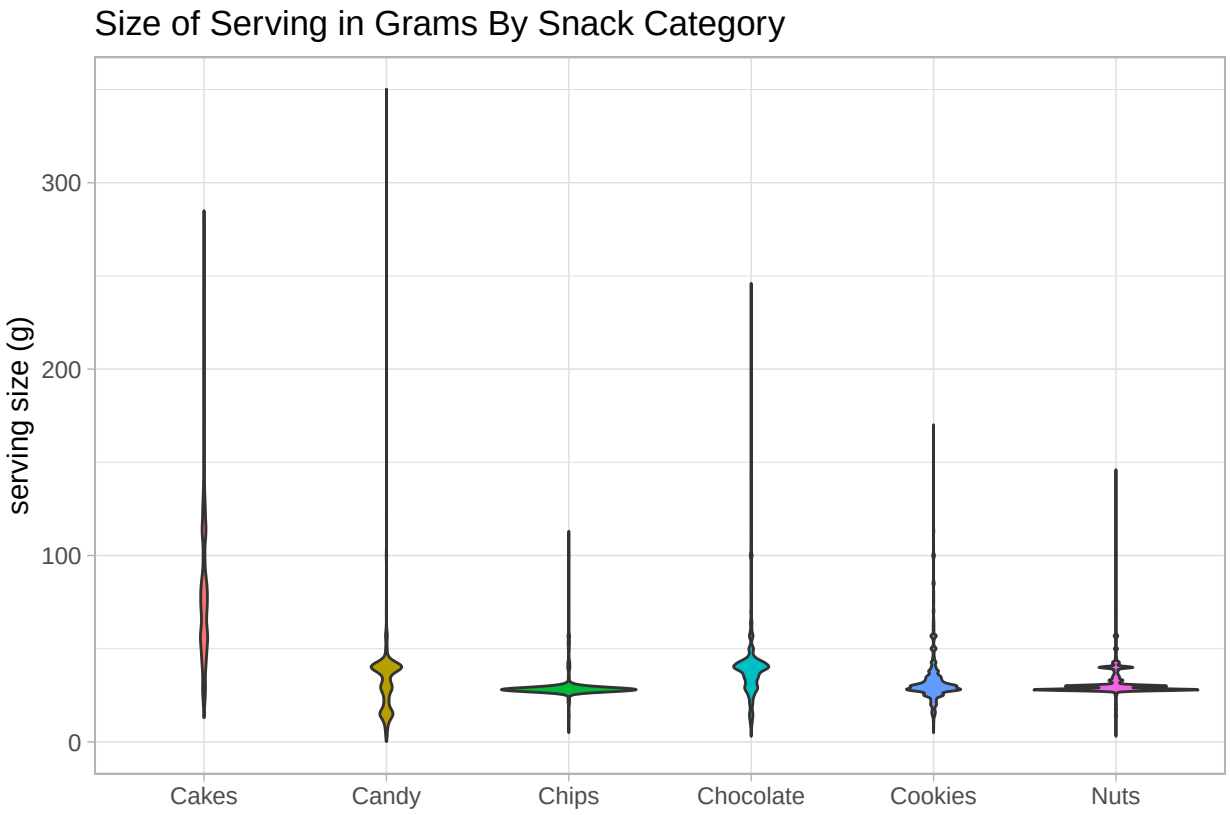
first copare the size of serving to 100g

```
snack_per_100 <-
  train %>% select(!ingredients) %>%filter(serving_size_unit=="g") %>% mutate(servings_per_100 = 100/se
```

variability of serving sizes per category

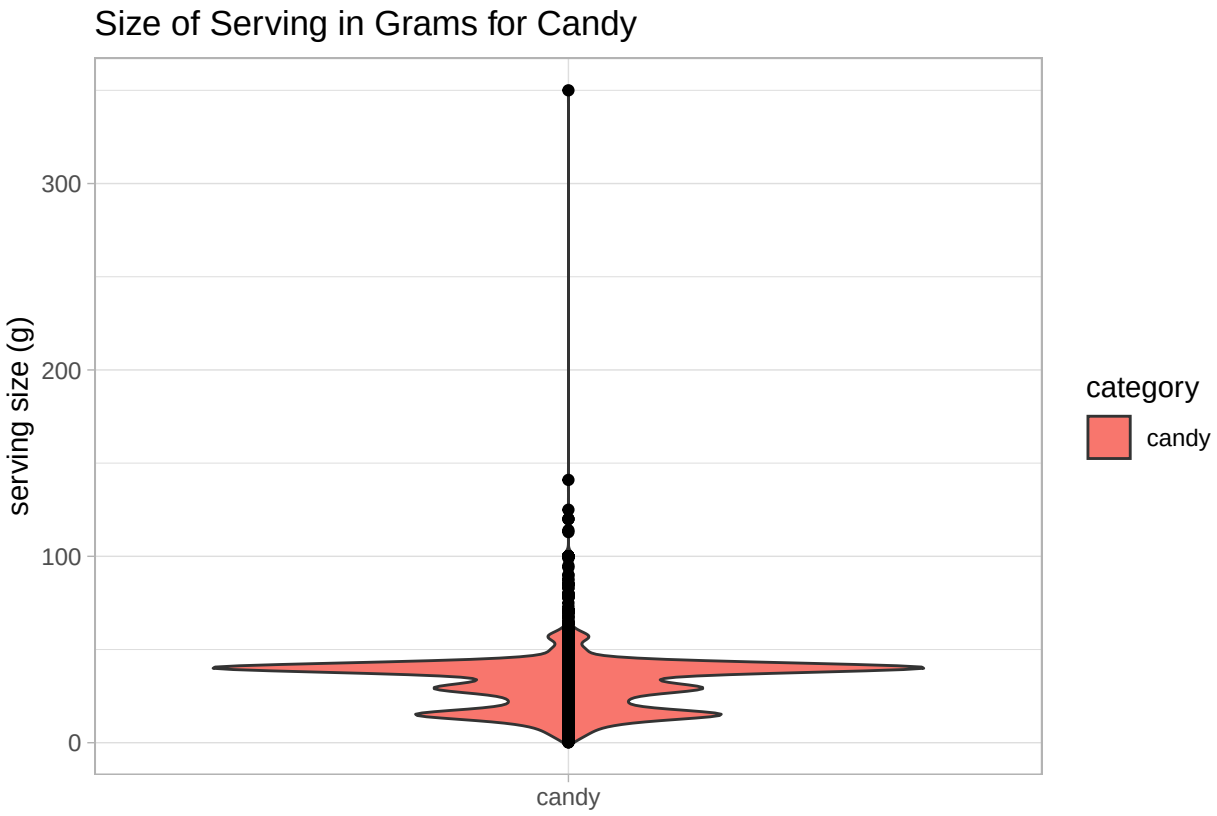
```
average_serving <- snack_per_100 %>% select(serving_size,category) %>% group_by(category) %>% summarise_

snack_per_100 %>% select(category,serving_size) %>% ggplot() +
  geom_violin(aes(x=category,y=serving_size,fill=category),show.legend = FALSE) +
  theme_light() +
  scale_x_discrete(labels=short_names)+
  labs(x="",y="serving size (g)",title="Size of Serving in Grams By Snack Category")
```



while the servings of nuts and chips are overall mostly consistent, cakes and candy vary much more lets focus on the biggest spread - candy

```
snack_per_100 %>% select(category,serving_size) %>% filter(category=="candy") %>% ggplot() +
  geom_violin(aes(x=category,y=serving_size,fill=category)) +
  geom_point(aes(x=category,y=serving_size))+
  theme_light() +
  #scale_x_discrete(labels=short_names)+
  labs(x="",y="serving size (g)",title="Size of Serving in Grams for Candy")
```



we can see that this spread is caused by a small number of outliers that stretch the range significantly

Let us now focus on the accessibility of serving sizes, we define several infractions, that is things that make serving sizes harder to quickly and intuitively parse, and compare the different categories of snacks in those areas the infractions are: 1) uncertain values - giving the serving size as a rough estimate muddles the calculations of nutritional values 2) using units of measurement - th main argumet for serving sizes against the "per 100g" system is the fact it does not require extra measuring equipment to assess 3)complicated numbers - it is easy to keep track of a portion of one cookie or two rolls, it becomes much harder to mentally track more tricky numbers, e.g. 17 pretzels 4) fractions - counting up is easier than dividing, it is also much harder to partition food into exact fractions

```
infraction_counter <- function(x) {
  y <- parse_number(x)
  #1 Uncertain values - using ranges or approximations when defining serving size
  uncertain_strings <- c("-","~","about","aprox")
  uncertain_strings <- str_c(uncertain_strings,collapse = "|")
  inf_unclear <- ifelse(str_detect(x,uncertain_strings),1,0)
  #2 measurement units - Using units that require a measuring device to parse
  measurement_strings <- c("grm","onz","oz","cup","sq.")
  measurement_strings <- str_c(measurement_strings,collapse = "|")
  inf_measurement <- ifelse(str_detect(x,measurement_strings),1,0)
  #3 complicated numbers - Using numbers that are harder to track
  easy_numbers <- c(1,2,3,0.5)
  inf_hard_numbers <- ifelse(y %in% easy_numbers,0,1)
  #4 fractions - tracking and portioning whole numbers is much easier than fractions, especially comple
  fraction_strings <- c("\\.\\d","\\d/\\d","half")
  fraction_strings <- str_c(fraction_strings,collapse = "|")
```

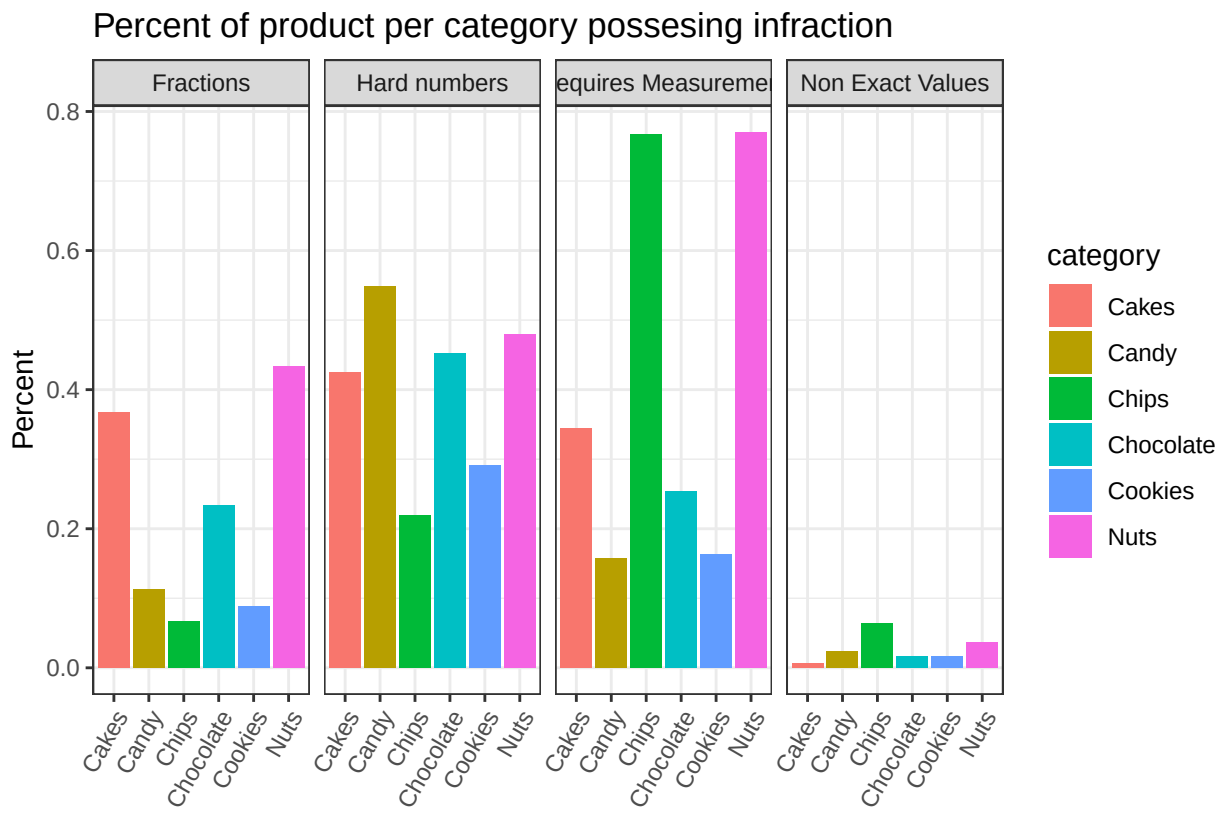
```
  inf_fractions <- ifelse(str_detect(x,fraction_strings),1,0)

  infraction <- cbind(inf_unclear,inf_measurement,inf_hard_numbers,inf_fractions)
  infraction
}
```

show how common each infraction is in the different categories

```
vis_infraction <- train %>% filter(!is.na(household_serving_fulltext))
vis_infraction <- vis_infraction %>% cbind(infraction_counter(vis_infraction$household_serving_fulltext)

infractions <- c("inf_fractions_percent"="Fractions",
                 "inf_hard_numbers_percent" = "Hard numbers",
                 "inf_measurement_percent" ="Requires Measurement",
                 "inf_unclear_percent"="Non Exact Values")
vis_infraction %>% select(category,starts_with("inf")) %>% group_by(category) %>% summarise(across(star
  pivot_longer(cols=(-1),names_to = "infraction") %>%
  ggplot() +
  facet_grid(~infraction,labeller = as_labeller(infractions)) +
  geom_bar(stat="identity",aes(x=category,y=value,fill=category)) +
  scale_x_discrete(labels=short_names)+
  scale_fill_discrete(labels=short_names)+
  labs(title = "Percent of product per category possesing infraction",y="Percent",x="")+
  theme_bw()+
  theme(axis.text.x = element_text(angle = 60, hjust=1))
```



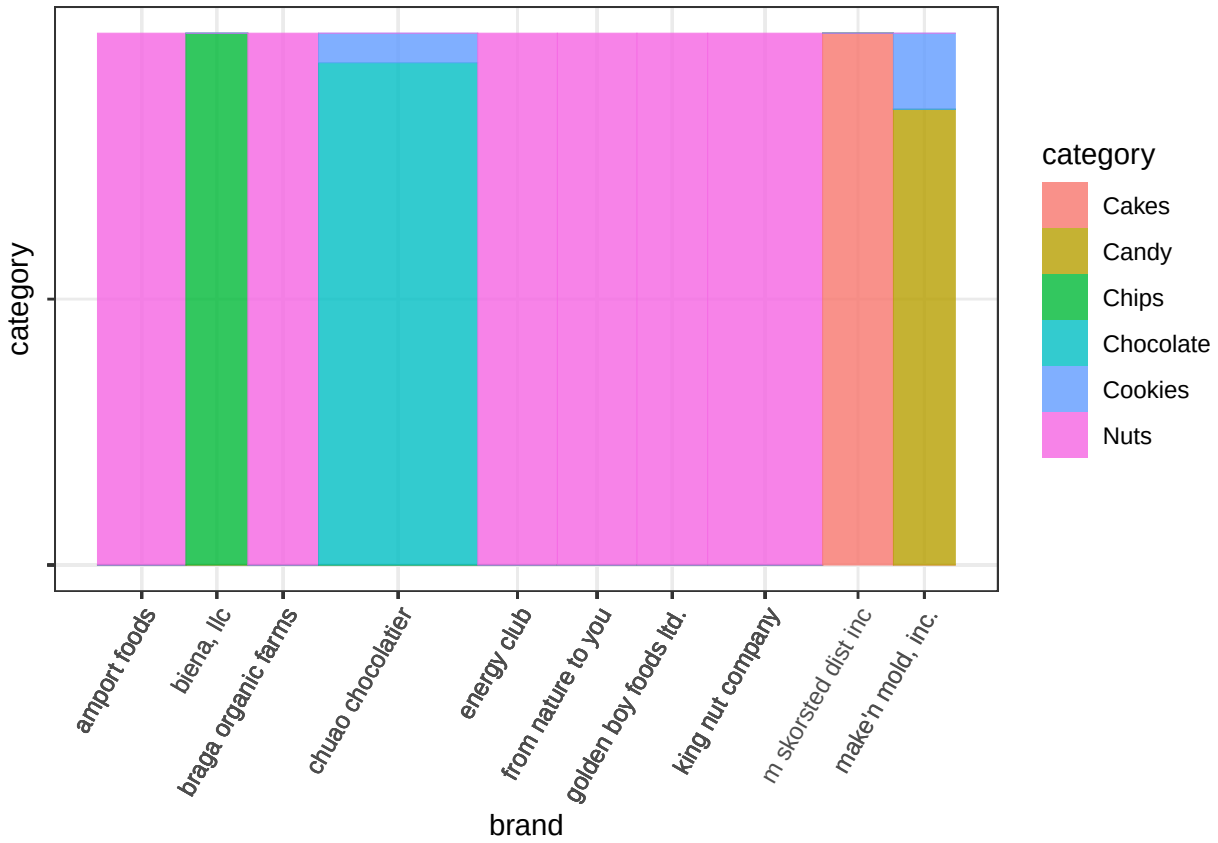
we can see some trends occuring in this plot: - cakes tend to have fractions and hard numbers, likley reffering to slices of a larger whole cake -chips and nuts most use measurement, they weight could possibly correlate to the size of a package, unfortunatley that information is not available to confirm this suspicion -most small piece foods tend to have non easy numbers per serving making it harder ot calculate Let's now focus on the manufacturers side: which manufacturers commit the most infractions on average (only manufacturers with at least 5 products included are counted), and what categories do the manufacture

```
large_manufacturers <- vis_infraction %>% group_by(brand) %>% summarise(n=n()) %>% filter(n>5) %>% pull
problematic_brands <- vis_infraction %>% filter(brand %in% large_manufacturers) %>%  mutate(total_infra
print(problematic_brands)
```

```
## # A tibble: 10 x 3
##    brand                 average products
##    <chr>                   <dbl>    <int>
##  1 braga organic farms      3.62        8
##  2 amport foods             3          10
##  3 biena, llc               3           7
##  4 chuao chocolatier        3          18
##  5 energy club              3           9
##  6 from nature to you       3           9
##  7 golden boy foods ltd.    3           8
##  8 king nut company         3          13
##  9 m skorsted dist inc      3           8
## 10 make'n mold, inc.        3           7
```

```
vis_infraction %>% select(brand,category) %>% filter(brand %in% problematic_brands$brand) %>%
  ggplot() +
  geom_mosaic(aes(x=product(category,brand),fill=category),offset = 0)+
  scale_fill_discrete(labels=short_names) +
  scale_y_productlist(labels = rep("",6)) +
  theme_bw()+
  theme(axis.text.x = element_text(angle = 60, hjust=1))
```

```
## Warning: `unite_()` was deprecated in tidyr 1.2.0.
## Please use `unite()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```



we can see that it is mostly nut companies that have the highest average infraction cut, possible because nuts are the least "manufactured" good and thus is hardest to control the nutrition value precisely