Data Analysis and Visualization Project

Tiffany Pang

4/28/2017

Creating a new Category column

In order to analyze and visualize data in the FNDDS 2011-2012 Foods database, Bob Horton’s starter script is used to create the database on the machine from flat files.

```r
data_dir <- "FNDDS_2011"

fortification <- c(`0`="none", `1`="fortified_product", `2`="contains fortified ingredie
nts")

fndds_tables <- list(
    AddFoodDesc = list(
            title="Additional Food Descriptions",
            column_types=c(
                food_code="integer", # foreign key
                seq_num="integer",
                start_date="date",
                end_date="date",
                additional_food_description="text"),
            sep="^"
        ),
    FNDDSNutVal = list(
            title="FNDDS Nutrient Values",
            column_types=c(
                food_code="integer",
                nutrient_code="integer",    # Nutrient Descriptions table
                start_date="date",
                end_date="date",
                nutrient_value="double"
                ),
            sep="^"
        ),
    FNDDSSRLinks = list(
            title="FNDDS-SR Links", # see p34 of fndds_2011_2012_doc.pdf
            column_types=c(
                food_code="integer",
                start_date="date",
                end_date="date",
                seq_num="integer",
                sr_code="integer",
                sr_descripton="text",
                amount="double",
                measure="char[3]",  # lb, oz, g, mg, cup, Tsp, qt, fluid ounce, etc
                portion_code="integer",
                retention_code="integer",
                flag="integer",
                weight="double",
                change_type_to_sr_code="char[1]",   # D=data change; F=food change
                change_type_to_weight="char[1]",
                change_type_to_retn_code="char[1]"
                ),
            sep="^"
        ),
    FoodPortionDesc = list(
            title="Food Portion Descriptions",
            column_types=c(
                portion_code="integer",     # foreign key
                start_date="date",
```

```
            end_date="date",
            portion_description="text",
            change_type="char[1]"
        ),
        sep="^"
    ),
    FoodSubcodeLinks = list(
        title="Food code-subcode links",
        column_types=c(
            food_code="integer",
            subcode="integer",
            start_date="date",
            end_date="date"
            ),
        sep="^"
    ),
    FoodWeights = list(
        title="Food Weights",
        column_types=c(
            food_code="integer",    # foreign key
            subcode="integer",
            seq_num="integer",
            portion_code="integer", # food portion description id
            start_date="date",
            end_date="date",
            portion_weight="double",    # missing values = -9
            change_type="char[1]"   # D=data change, F=food change
            ),
        sep="^"
    ),
    MainFoodDesc = list(
        title="Main Food Descriptions",
        column_types=c(
            food_code="integer",
            start_date="date",
            end_date="date",
            main_food_description="character",
            fortification_id="integer"),
        sep="^"
    ),
    ModDesc = list(
        title="Modifications Descriptons",
        column_types=c(
            modification_code="integer",
            start_date="date",
            end_date="date",
            modification_description="text",
            food_code="integer"

            ),
        sep="^"
    ),
    ModNutVal = list(
        title="Modifications Nutrient Values",
```

```
            column_types=c(
                modification_code="integer",
                nutrient_code="integer",
                start_date="date",
                end_date="date",
                nutrient_value="double"
                ),
            sep="^"
        ),
    MoistNFatAdjust = list(
            title="Moisture & Fat Adjustments", # to account for changes during cooking
            column_types=c(
                food_code="integer",
                start_date="date",
                end_date="date",
                moisture_change="double",
                fat_change="double",
                type_of_fat="integer"   # SR code or food code
                ),
            sep="^"
        ),
    NutDesc = list(
            title="Nutrient Descriptions",
            column_types=c(
                nutrient_code="integer",
                nutrient_description="text",
                tagname="text",
                unit="text",
                decimals="integer"  # decimal places
                ),
            sep="^"
        ),
    SubcodeDesc = list(
            title="Subcode Descriptions",
            column_types=c(
                subcode="integer",  # key; 0=use default gram weights
                start_date="date",
                end_date="date",
                subcode_description="text"
                ),
            sep="^"
        )
)

# flat file to a data frame: call for each table
assign_data_frame <- function(tbl_name){
    tbl <- read.table(
        file.path(data_dir, paste0(tbl_name, ".txt")),
        sep="^",
        quote="~",
        stringsAsFactors=FALSE)
    # drop last (empty) column
    tbl <- tbl[1:(length(tbl)-1)]
    names(tbl) <- names(fndds_tables[[tbl_name]][["column_types"]])
```

```r
    assign(tbl_name, tbl, envir = .GlobalEnv)
}

# flat file to database
fndds2sqlite <- function(data_dir, table_details, sqlite_filename){

    library("RSQLite")
    con <- dbConnect(SQLite(), sqlite_filename)

    for (tbl_name in names(table_details)){
        file_name <- paste0(tbl_name, ".txt")
        assign_data_frame(tbl_name)
        tbl <- get(tbl_name)
        dbWriteTable(con, tbl_name, tbl, row.names = FALSE)
    }

    dbDisconnect(con)
}

fndds2sqlite("FNDDS_2011", fndds_tables, "fndds.sqlite")
library(DBI)

for (tbl in c("FNDDSNutVal", "MainFoodDesc", "NutDesc"))
    assign_data_frame(tbl)

library(dplyr)
library(tidyr)

# Make a simplified selection of foods.
# TO DO: have MainFoodDesc be a tbl sourced from SQLite.
get_selected_foods <- function(){
    # Pull out all "Not Further Specified" foods as a wide selection of reasonably gener
ic items.
    generics <- MainFoodDesc %>%
        filter( grepl(", NFS", main_food_description )) %>%
        filter(!grepl("infant formula", main_food_description, ignore.case = TRUE ) )

    # Raw fruits
    # Berries are covered by "Berries, raw, NFS" and "Berries, frozen, NFS"
    fruits <- MainFoodDesc %>%
        filter( grepl("^6", food_code) ) %>%
        filter( grepl("^([^,\\(]+), raw$", main_food_description) ) %>%
        filter( !grepl("berries", main_food_description) )

    # Raw vegetables
    # Potatoes are covered by "White potato, NFS", "Sweet potato, NFS", etc.
    vegetables <- MainFoodDesc %>%
        filter( grepl("^7", food_code) ) %>%
        filter(!grepl("potato", main_food_description)) %>%
        filter( grepl(", raw$", main_food_description))

    # 4="legumes, nuts, and seeds"
    nuts_and_seeds <- MainFoodDesc %>%
        filter( grepl("^4", food_code) ) %>%
```

```
        mutate( firstWord = strsplit(main_food_description, " ")[[1]][1] )

    # Selected alcoholic beverages
    # All alcoholic beverages: grepl("^93", food_code))
    # "Cocktail, NFS" already gives us "Cocktail"
    alcoholic_beverages <- MainFoodDesc %>%
        filter( main_food_description %in% c("Beer", "Wine, table, red", "Wine, table, w
hite",
            "Whiskey", "Gin", "Rum", "Vodka") )

    # Collect them all into one table
    rbind(generics, fruits, vegetables, alcoholic_beverages) %>%
        select( food_code, main_food_description, fortification_id )  %>%
        filter( nchar(main_food_description) < 20 ) %>%
        mutate( main_food_description = gsub("(, NFS|, raw)", "", main_food_description)
 )

}

foods <- get_selected_foods()   # 163 items
```

The following code is then used to create a new column named Category based on Appendix E. Food/Beverage Coding Scheme, and appended to food_nutrient_df.

```r
library(sqldf)

long_food_nutrients_food_code <-
  sqldf(
    "SELECT f.food_code, nd.nutrient_description, nv.nutrient_value
    FROM foods f
    INNER JOIN FNDDSNutVal nv ON f.food_code = nv.food_code
    INNER JOIN NutDesc nd ON nv.nutrient_code = nd.nutrient_code"
  )

food_code_dataframe <- spread(long_food_nutrients_food_code, food_code, nutrient_value,
 fill=0)
food_code_mat <- t(as.matrix(food_code_dataframe[-1]))
colnames(food_code_mat) <- food_code_dataframe$nutrient_description

food_code <- as.integer(row.names(food_code_mat))
Category <- rep("dairy", length(food_code))
temp_df <- data.frame(food_code, Category, stringsAsFactors=F)

temp_df$Category[temp_df$food_code>=94000000] <- "protein powder"
temp_df$Category[temp_df$food_code<94000000] <- "alcohol"
temp_df$Category[temp_df$food_code<93000000] <- "sugars"
temp_df$Category[temp_df$food_code<90000000] <- "fats"
temp_df$Category[temp_df$food_code<80000000] <- "vegetables"
temp_df$Category[temp_df$food_code<70000000] <- "fruits"
temp_df$Category[temp_df$food_code<60000000] <- "grains"
temp_df$Category[temp_df$food_code<50000000] <- "legumes,nuts,seeds"
temp_df$Category[temp_df$food_code<40000000] <- "eggs"
temp_df$Category[temp_df$food_code<30000000] <- "meat,fish"
temp_df$Category[temp_df$food_code<20000000] <- "dairy"

food_nutrient_df <- as.data.frame(food_code_mat, stringsAsFactors = FALSE)
food_nutrient_df <- cbind(food_nutrient_df, Category=temp_df$Category)
```

Necessary changes are then made to the dataframe for further exploratory data analysis

```r
# remove the first 19 columns of the dataframe
food_df <- food_nutrient_df[, -c(1:19)]

# replace spaces in column names with underscores
library(stringr)
colnames(food_df) <- str_replace_all(colnames(food_df),"[[:punct:]\\s]+","_")
```

The following code was used to find the count of each food category.

```r
as.data.frame(table(food_df$Category))
```

```
##                  Var1 Freq
## 1             alcohol    8
## 2               dairy    4
## 3                fats    4
## 4              fruits   37
## 5              grains   28
## 6  legumes,nuts,seeds    6
## 7           meat,fish   16
## 8      protein powder    1
## 9              sugars    6
## 10         vegetables   53
```

Check the class of all variables.

```
as.data.frame(sapply(food_df, class))
```

```
##                                     sapply(food_df, class)
## Alcohol                                           numeric
## Caffeine                                          numeric
## Calcium                                           numeric
## Carbohydrate                                      numeric
## Carotene_alpha                                    numeric
## Carotene_beta                                     numeric
## Cholesterol                                       numeric
## Choline_total                                     numeric
## Copper                                            numeric
## Cryptoxanthin_beta                                numeric
## Energy                                            numeric
## Fatty_acids_total_monounsaturated                 numeric
## Fatty_acids_total_polyunsaturated                 numeric
## Fatty_acids_total_saturated                       numeric
## Fiber_total_dietary                               numeric
## Folate_DFE                                        numeric
## Folate_food                                       numeric
## Folate_total                                      numeric
## Folic_acid                                        numeric
## Iron                                              numeric
## Lutein_+_zeaxanthin                               numeric
## Lycopene                                          numeric
## Magnesium                                         numeric
## Niacin                                            numeric
## Phosphorus                                        numeric
## Potassium                                         numeric
## Protein                                           numeric
## Retinol                                           numeric
## Riboflavin                                        numeric
## Selenium                                          numeric
## Sodium                                            numeric
## Sugars_total                                      numeric
## Theobromine                                       numeric
## Thiamin                                           numeric
## Total_Fat                                         numeric
## Vitamin_A_RAE                                     numeric
## Vitamin_B_12                                      numeric
## Vitamin_B_12_added                                numeric
## Vitamin_B_6                                       numeric
## Vitamin_C                                         numeric
## Vitamin_D_D2_+_D3_                                numeric
## Vitamin_E_alpha_tocopherol_                       numeric
## Vitamin_E_added                                   numeric
## Vitamin_K_phylloquinone_                          numeric
## Water                                             numeric
## Zinc                                              numeric
## Category                                           factor
```

Summary of the data.

```
summary(food_df)
```

```
##     Alcohol          Caffeine          Calcium        Carbohydrate
##  Min.   : 0.000   Min.   : 0.0000   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 0.000   1st Qu.: 0.0000   1st Qu.:  9.00   1st Qu.: 4.91
##  Median : 0.000   Median : 0.0000   Median : 22.00   Median : 9.58
##  Mean   : 1.056   Mean   : 0.1595   Mean   : 54.43   Mean   :19.65
##  3rd Qu.: 0.000   3rd Qu.: 0.0000   3rd Qu.: 51.00   3rd Qu.:19.95
##  Max.   :37.900   Max.   :11.0000   Max.   :950.00   Max.   :99.98
##
##  Carotene_alpha    Carotene_beta      Cholesterol      Choline_total
##  Min.   :   0.00   Min.   :    0.0   Min.   :  0.000   Min.   :  0.00
##  1st Qu.:   0.00   1st Qu.:    0.0   1st Qu.:  0.000   1st Qu.:  6.05
##  Median :   0.00   Median :   29.0   Median :  0.000   Median :  9.80
##  Mean   :  54.98   Mean   :  523.9   Mean   :  8.853   Mean   : 18.66
##  3rd Qu.:   4.00   3rd Qu.:  251.5   3rd Qu.:  0.000   3rd Qu.: 22.00
##  Max.   :3477.00   Max.   :10980.0   Max.   :215.000   Max.   :224.00
##
##      Copper       Cryptoxanthin_beta     Energy
##  Min.   :0.0000   Min.   :   0.00    Min.   : 11.0
##  1st Qu.:0.0420   1st Qu.:   0.00    1st Qu.: 34.0
##  Median :0.0730   Median :   0.00    Median : 68.0
##  Mean   :0.1313   Mean   :  27.87    Mean   :152.8
##  3rd Qu.:0.1500   3rd Qu.:   1.50    3rd Qu.:238.0
##  Max.   :2.2200   Max.   :1447.00    Max.   :886.0
##
##  Fatty_acids_total_monounsaturated Fatty_acids_total_polyunsaturated
##  Min.   : 0.0000                   Min.   : 0.0000
##  1st Qu.: 0.0185                   1st Qu.: 0.0555
##  Median : 0.0860                   Median : 0.1440
##  Mean   : 2.4454                   Mean   : 1.4888
##  3rd Qu.: 1.0440                   3rd Qu.: 0.5410
##  Max.   :40.4390                   Max.   :43.2770
##
##  Fatty_acids_total_saturated Fiber_total_dietary   Folate_DFE
##  Min.   : 0.000              Min.   : 0.000      Min.   :   0.00
##  1st Qu.: 0.026              1st Qu.: 0.600      1st Qu.:   6.00
##  Median : 0.071              Median : 1.700      Median :  18.00
##  Mean   : 1.676              Mean   : 2.221      Mean   :  81.44
##  3rd Qu.: 1.123              3rd Qu.: 2.950      3rd Qu.:  57.00
##  Max.   :51.368              Max.   :11.800      Max.   :1256.00
##
##   Folate_food      Folate_total      Folic_acid          Iron
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:  5.00   1st Qu.:  6.00   1st Qu.:  0.00   1st Qu.: 0.250
##  Median : 14.00   Median : 18.00   Median :  0.00   Median : 0.580
##  Mean   : 23.77   Mean   : 57.69   Mean   : 33.93   Mean   : 2.367
##  3rd Qu.: 28.50   3rd Qu.: 50.50   3rd Qu.:  0.00   3rd Qu.: 1.765
##  Max.   :194.00   Max.   :741.00   Max.   :737.00   Max.   :33.300
##
##  Lutein_+_zeaxanthin    Lycopene        Magnesium          Niacin
##  Min.   :    0.0     Min.   :   0.0   Min.   :  0.00   Min.   : 0.000
##  1st Qu.:    0.0     1st Qu.:   0.0   1st Qu.: 10.00   1st Qu.: 0.252
##  Median :   19.0     Median :   0.0   Median : 15.00   Median : 0.640
##  Mean   :  550.4     Mean   : 227.4   Mean   : 28.42   Mean   : 2.297
```

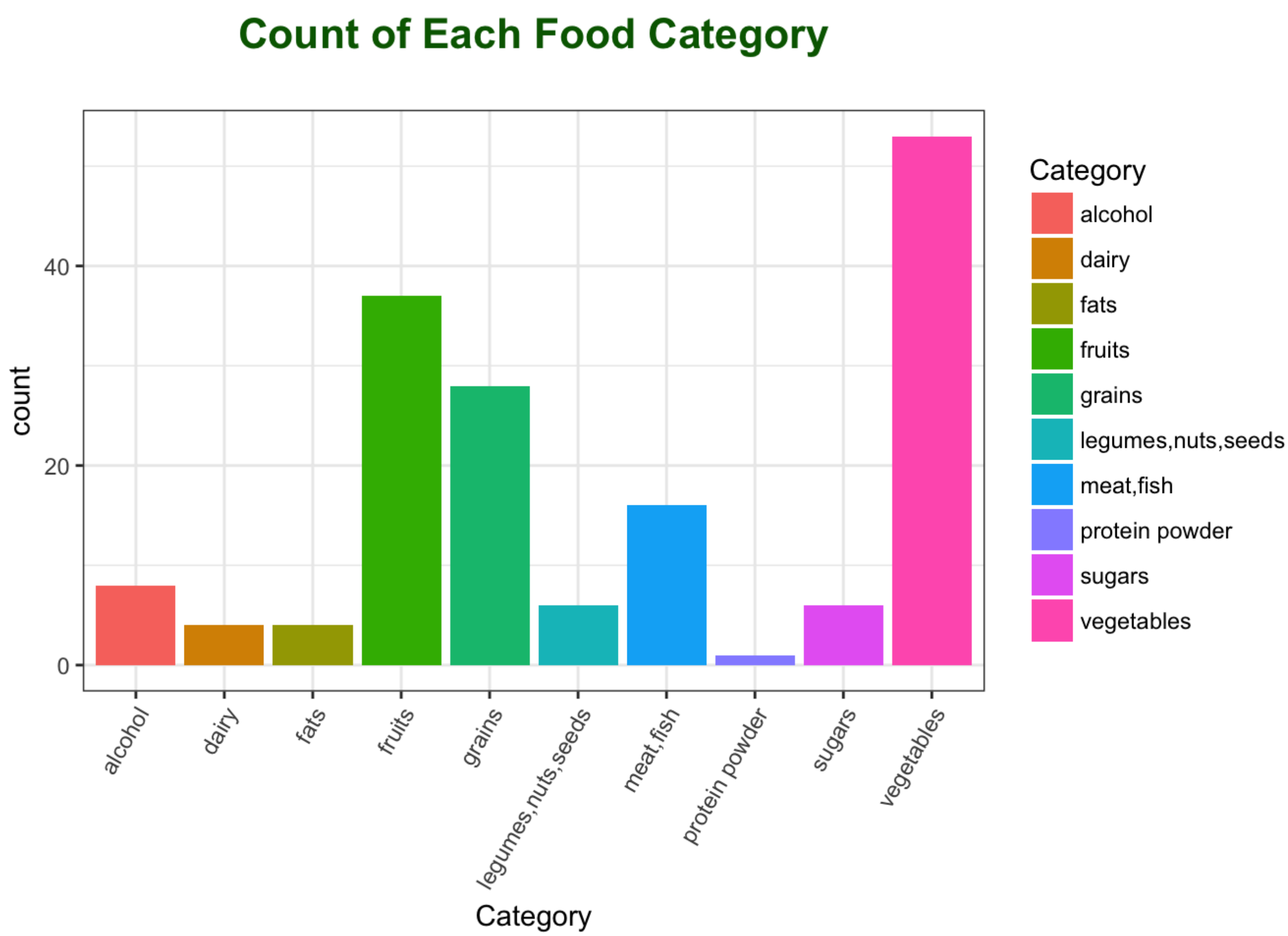
```
##  3rd Qu.:  129.5     3rd Qu.:   0.0   3rd Qu.: 27.00   3rd Qu.: 1.629
##  Max.   :12500.0     Max.   :6312.0   Max.   :279.00   Max.   :28.967
##
##    Phosphorus        Potassium        Protein          Retinol
##  Min.   :   0.00   Min.   :  0.0   Min.   : 0.000   Min.   :   0.00
##  1st Qu.:  18.50   1st Qu.:115.5   1st Qu.: 0.815   1st Qu.:   0.00
##  Median :  41.00   Median :191.0   Median : 1.800   Median :   0.00
##  Mean   :  88.28   Mean   :231.9   Mean   : 4.750   Mean   :  60.64
##  3rd Qu.:  89.00   3rd Qu.:314.0   3rd Qu.: 4.920   3rd Qu.:   0.00
##  Max.   :1321.00   Max.   :762.0   Max.   :78.130   Max.   :1250.00
##
##    Riboflavin       Selenium           Sodium        Sugars_total
##  Min.   :0.000   Min.   :  0.000   Min.   :   0.0   Min.   : 0.000
##  1st Qu.:0.027   1st Qu.:  0.400   1st Qu.:   4.0   1st Qu.: 0.890
##  Median :0.055   Median :  0.900   Median :  38.0   Median : 3.940
##  Mean   :0.198   Mean   :  5.558   Mean   : 204.0   Mean   : 8.040
##  3rd Qu.:0.151   3rd Qu.:  5.600   3rd Qu.: 331.5   3rd Qu.: 9.205
##  Max.   :2.827   Max.   :111.400   Max.   :1737.0   Max.   :99.800
##
##   Theobromine         Thiamin          Total_Fat       Vitamin_A_RAE
##  Min.   : 0.0000   Min.   :0.0000   Min.   :  0.000   Min.   :   0.0
##  1st Qu.: 0.0000   1st Qu.:0.0280   1st Qu.:  0.175   1st Qu.:   0.0
##  Median : 0.0000   Median :0.0520   Median :  0.420   Median :  10.0
##  Mean   : 0.9386   Mean   :0.1622   Mean   :  6.022   Mean   : 107.7
##  3rd Qu.: 0.0000   3rd Qu.:0.1175   3rd Qu.:  3.485   3rd Qu.:  55.0
##  Max.   :83.0000   Max.   :2.0500   Max.   :100.000   Max.   :1250.0
##
##   Vitamin_B_12    Vitamin_B_12_added  Vitamin_B_6       Vitamin_C
##  Min.   :0.0000   Min.   :0.0000     Min.   :0.0000   Min.   :  0.00
##  1st Qu.:0.0000   1st Qu.:0.0000     1st Qu.:0.0435   1st Qu.:  0.10
##  Median :0.0000   Median :0.0000     Median :0.0900   Median :  5.90
##  Mean   :0.5129   Mean   :0.3361     Mean   :0.2546   Mean   : 18.35
##  3rd Qu.:0.0650   3rd Qu.:0.0000     3rd Qu.:0.1945   3rd Qu.: 25.95
##  Max.   :7.1000   Max.   :7.1000     Max.   :2.4910   Max.   :228.30
##
##  Vitamin_D_D2_+_D3_ Vitamin_E_alpha_tocopherol_ Vitamin_E_added
##  Min.   :0.0000     Min.   : 0.000              Min.   : 0.0000
##  1st Qu.:0.0000     1st Qu.: 0.100              1st Qu.: 0.0000
##  Median :0.0000     Median : 0.300              Median : 0.0000
##  Mean   :0.2804     Mean   : 1.064              Mean   : 0.1154
##  3rd Qu.:0.0000     3rd Qu.: 0.785              3rd Qu.: 0.0000
##  Max.   :5.0000     Max.   :23.900              Max.   :11.1800
##
##  Vitamin_K_phylloquinone_     Water            Zinc
##  Min.   :   0.0          Min.   : 0.00   Min.   : 0.000
##  1st Qu.:   0.3          1st Qu.:58.71   1st Qu.: 0.105
##  Median :   2.6          Median :83.07   Median : 0.250
##  Mean   :  46.9          Mean   :67.25   Mean   : 1.116
##  3rd Qu.:  13.2          3rd Qu.:89.70   3rd Qu.: 0.795
##  Max.   :1640.0          Max.   :96.73   Max.   :16.730
##
##                Category
##  vegetables        :53
##  fruits            :37
```

```
##   grains              :28
##   meat,fish           :16
##   alcohol             : 8
##   legumes,nuts,seeds: 6
##   (Other)             :15
```

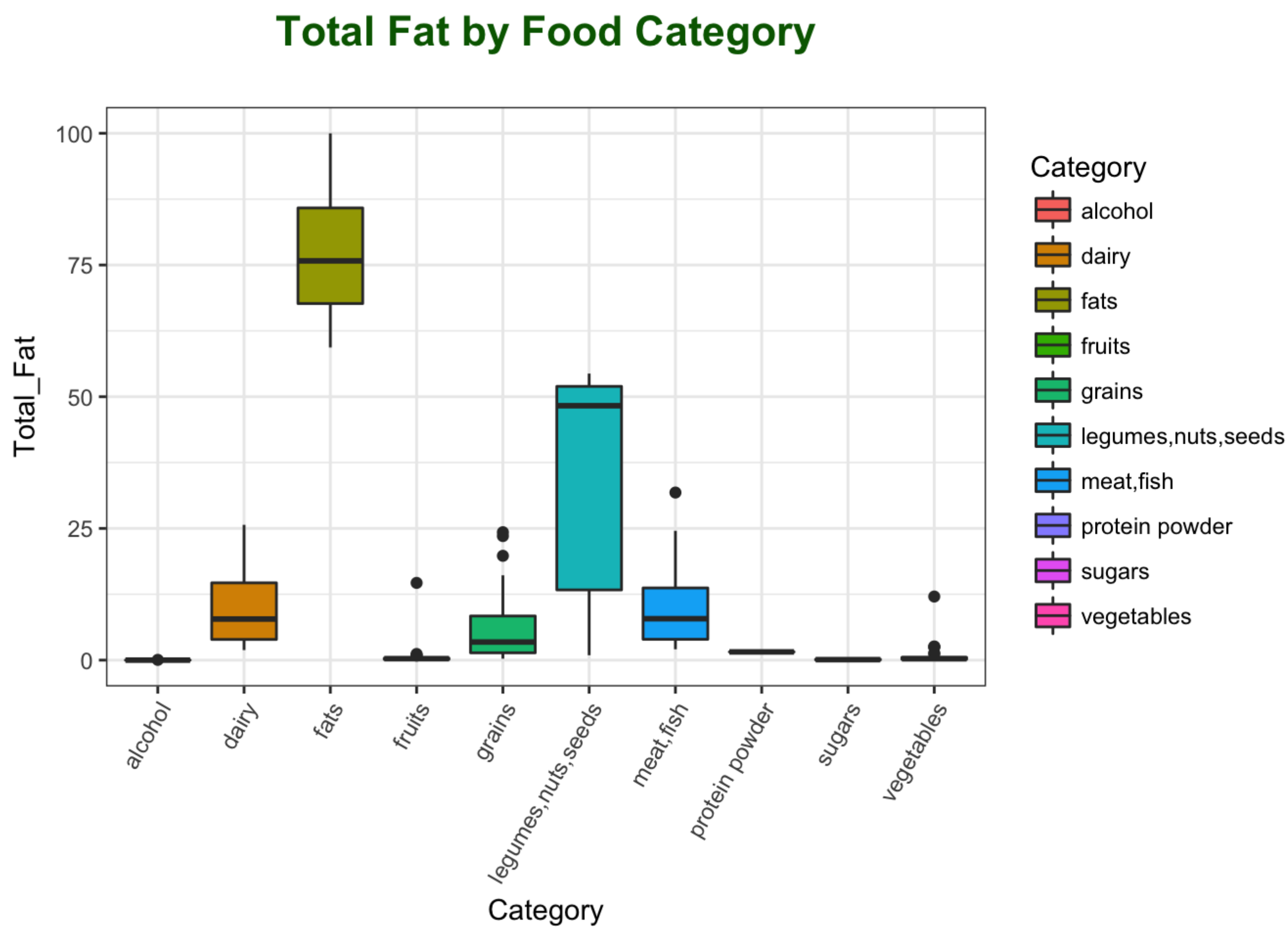
Data Visualization with ggplot

```
library(ggplot2)

# Visualize the count of each Food Category
g <- ggplot(food_df, aes(x=Category))
g + geom_bar(aes(fill = Category)) +
  labs(title = "Count of Each Food Category\n") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
  theme(plot.title=element_text(hjust=0.5, size=16, face="bold", color="darkgreen"))
```

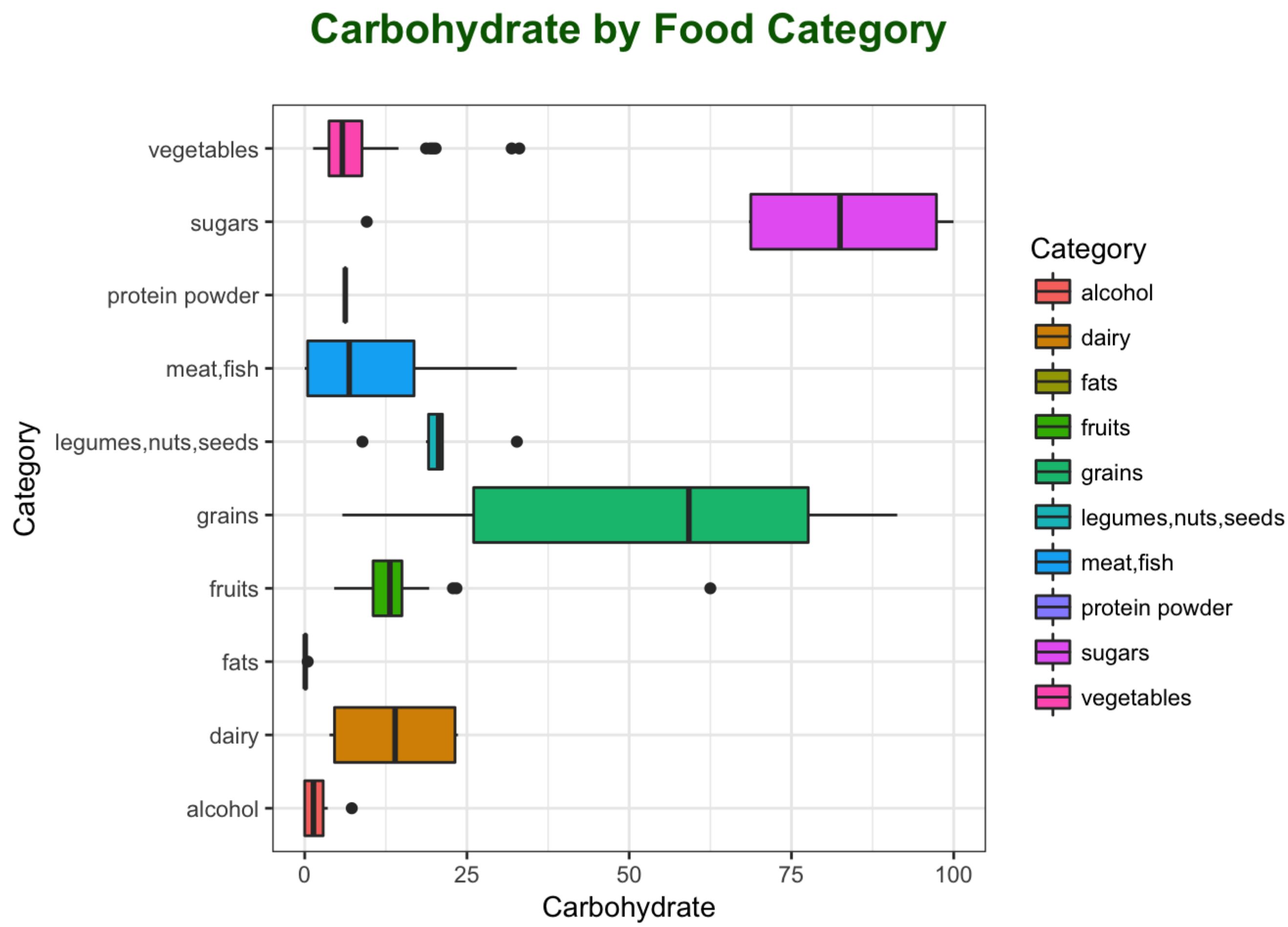


```
# Boxplot of Total Fat by Food Category
ggplot(food_df, aes(x=Category, y=Total_Fat)) + # categorical variable on x-axis
  geom_boxplot(aes(fill = Category)) +
  labs(title = "Total Fat by Food Category\n") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
  theme(plot.title=element_text(hjust=0.5, size=16, face="bold", color="darkgreen"))
```



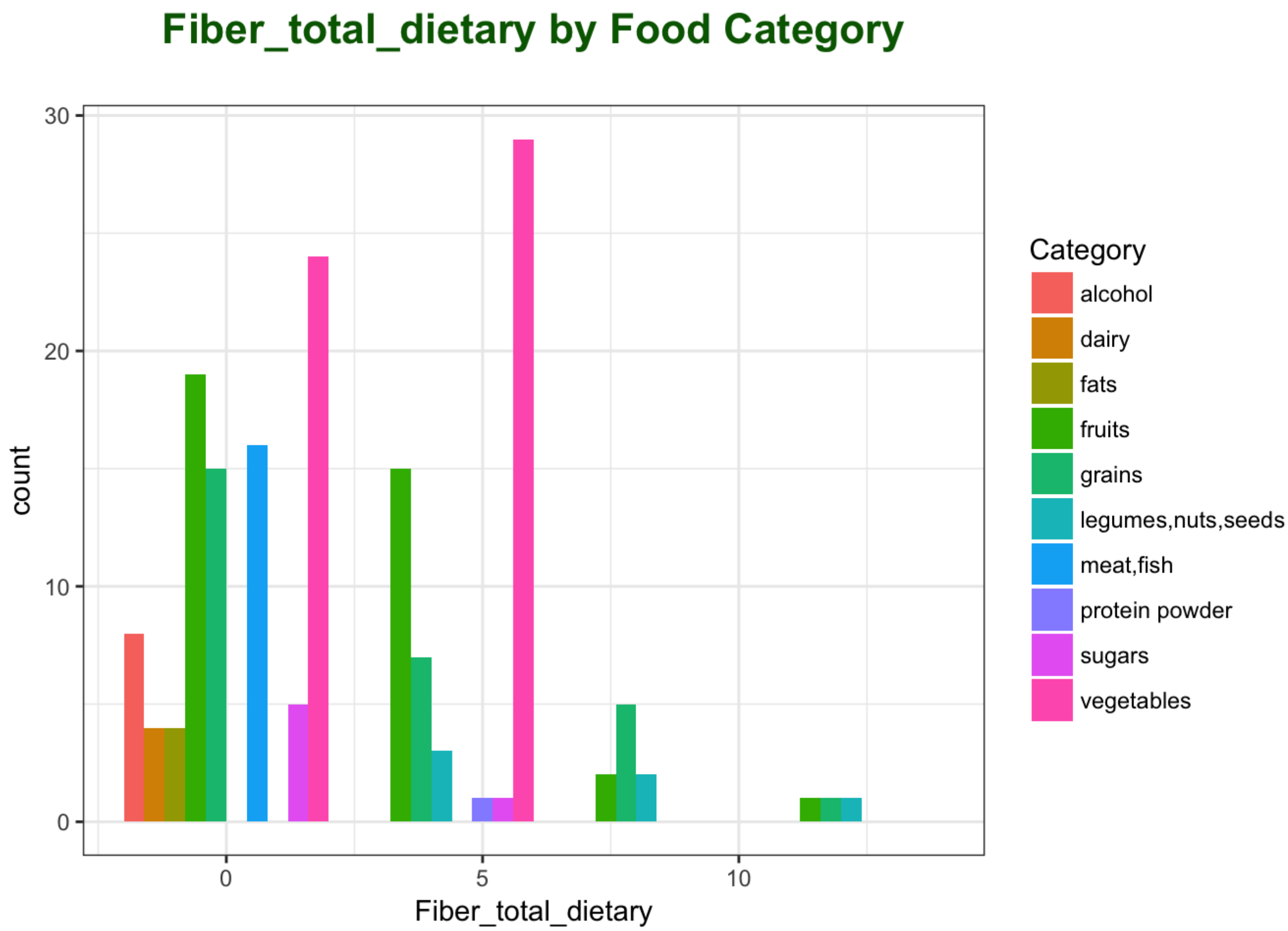
The category 'fats' contains the highest total_fat, followed by 'legumes,nuts,seeds'. 'dairy' and 'meat,fish' categories contain the same amount of total_fat.

```
# Boxplot of Carbohydrate by Food Category
ggplot(food_df, aes(x=Category, y=Carbohydrate)) + # categorical variable on x-axis
  geom_boxplot(aes(fill = Category)) + coord_flip() +
  labs(title = "Carbohydrate by Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=16, face="bold", color="darkgreen"))
```



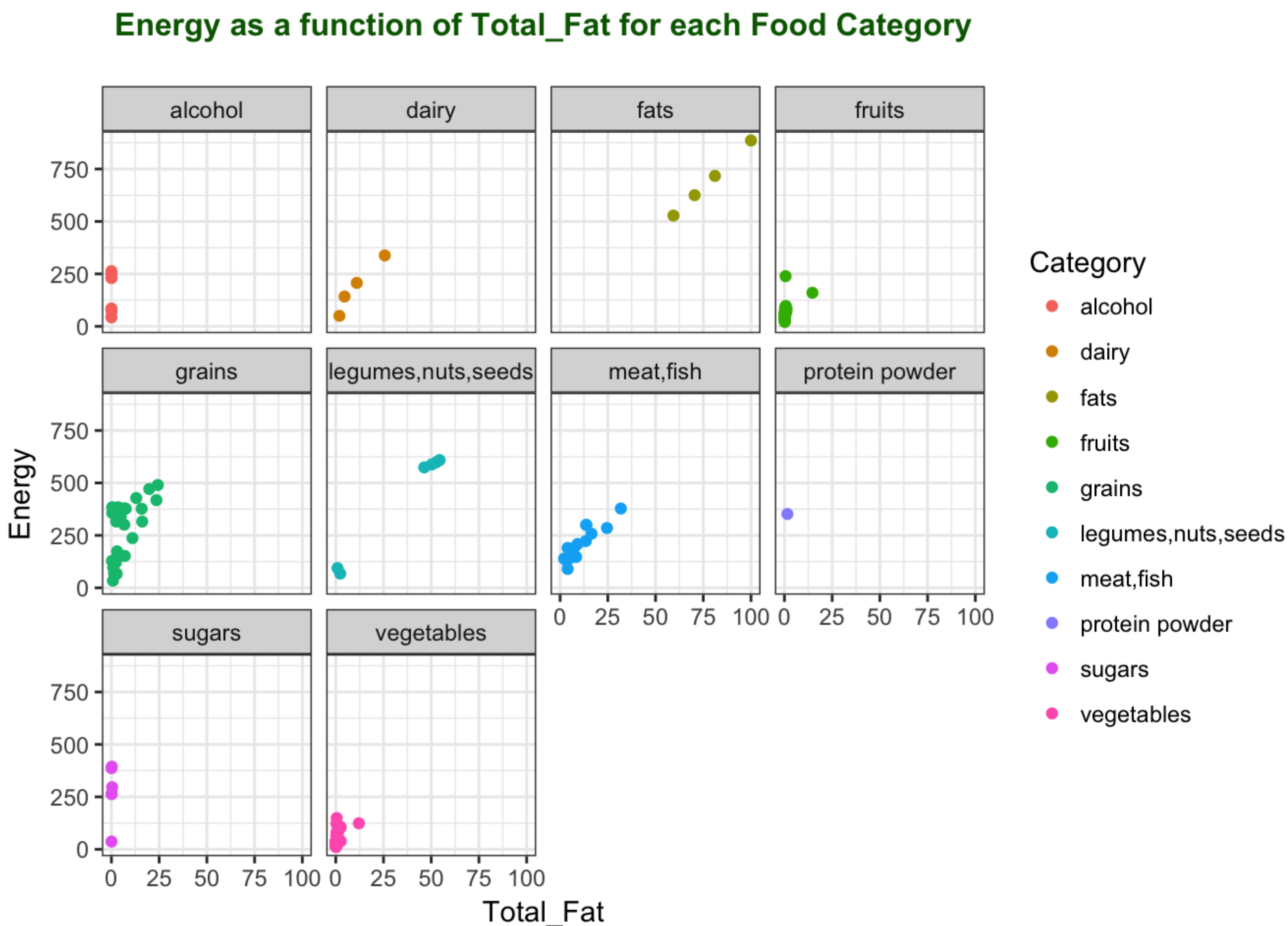
Sugars contain the highest amount of carbohydrate, followed by grains.

```
# Interleaved histogram of fiber_total_dietary by Food Category
ggplot(food_df, aes(x=Fiber_total_dietary, fill=Category)) +
  geom_histogram(position = "dodge", binwidth = 4) +
  labs(title = "Fiber_total_dietary by Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=16, face="bold", color="darkgreen"))
```



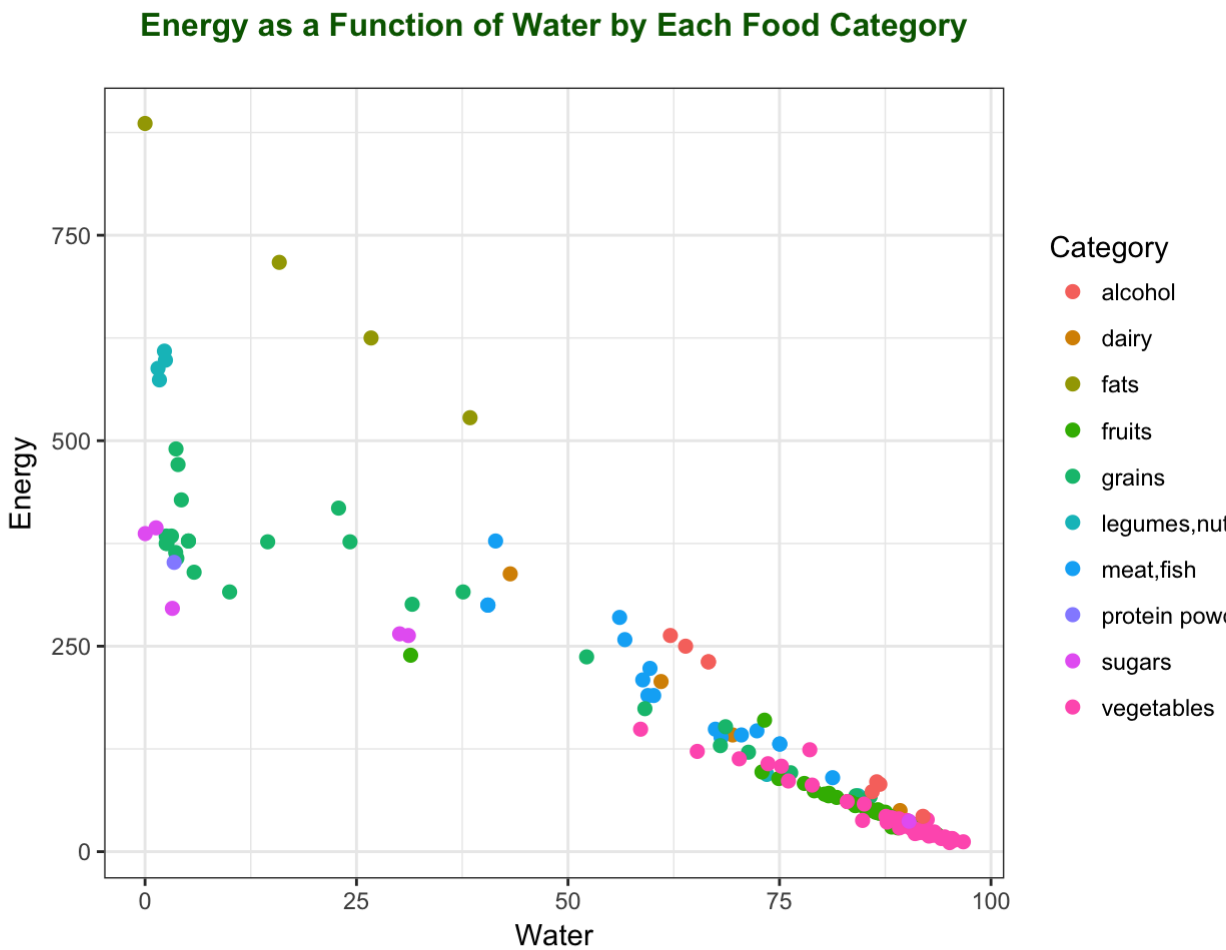
Grains, legumes, nuts, seeds and fruits contain the highest total dietary fiber and alcohol, diary and fats contain no fiber.

```
# Energy as a function of Total_Fat for each Food Category
ggplot(food_df, aes(x = Total_Fat, y = Energy, colour = Category)) +
  geom_point() +
  facet_wrap( ~ Category) +
  labs(title="Energy as a function of Total_Fat for each Food Category\n", x="Total_Fat"
, y="Energy") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=12, face="bold", color="darkgreen"))
```



Fats category has the highest amount of total_fat and energy, whereas vegetables category has low total_fat and also low energy.

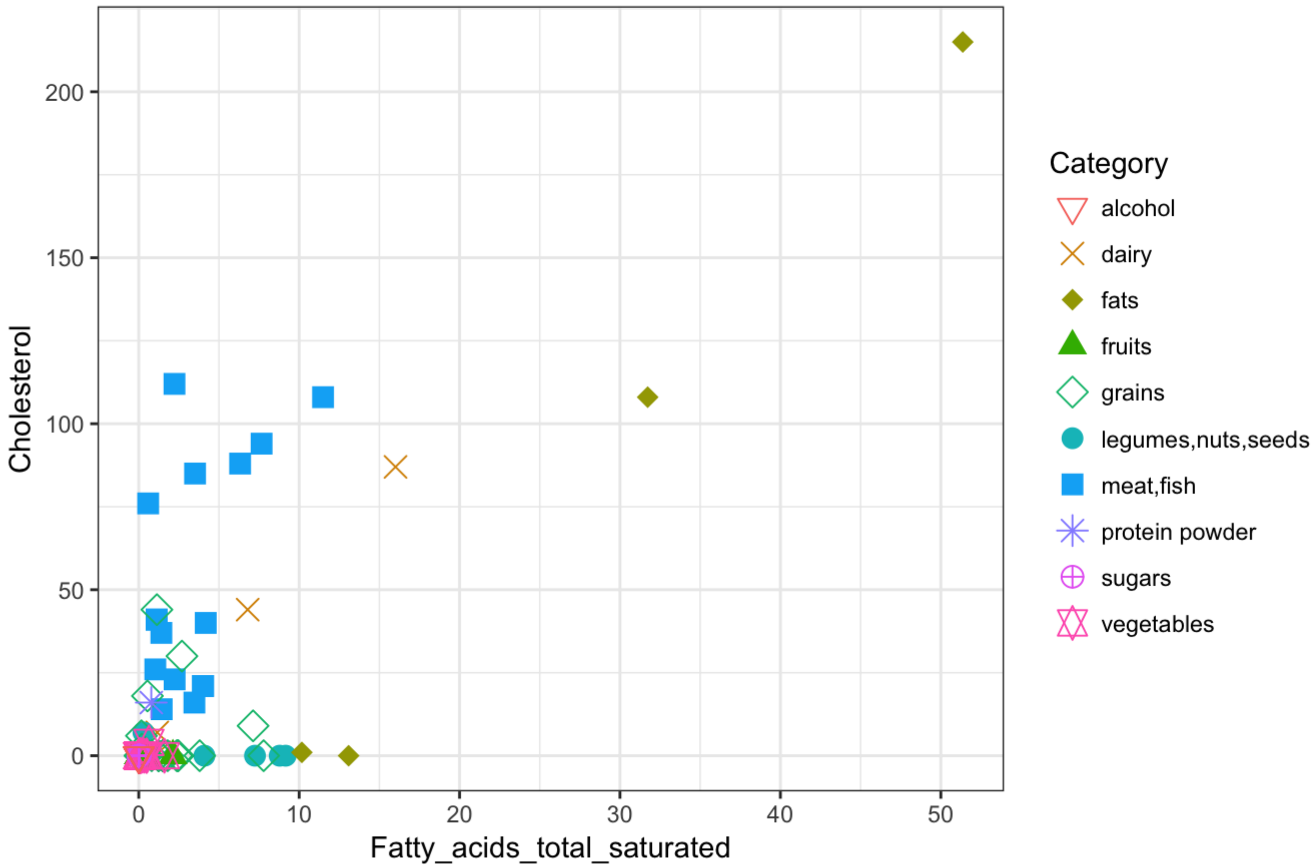
```
# Scatterplot of Energy as a Function of Water by Each Food Category
ggplot(food_df, aes(x=Water,y=Energy)) +
  geom_point(aes(fill=Category, color=Category), size=2) +
  labs(title="Energy as a Function of Water by Each Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=12, face="bold", color="darkgreen"))
```



As the water content of the food goes up, it contains lower energy. Water and energy are inversely correlated. Again, here vegetables have the highest water content and provide lowest energy.

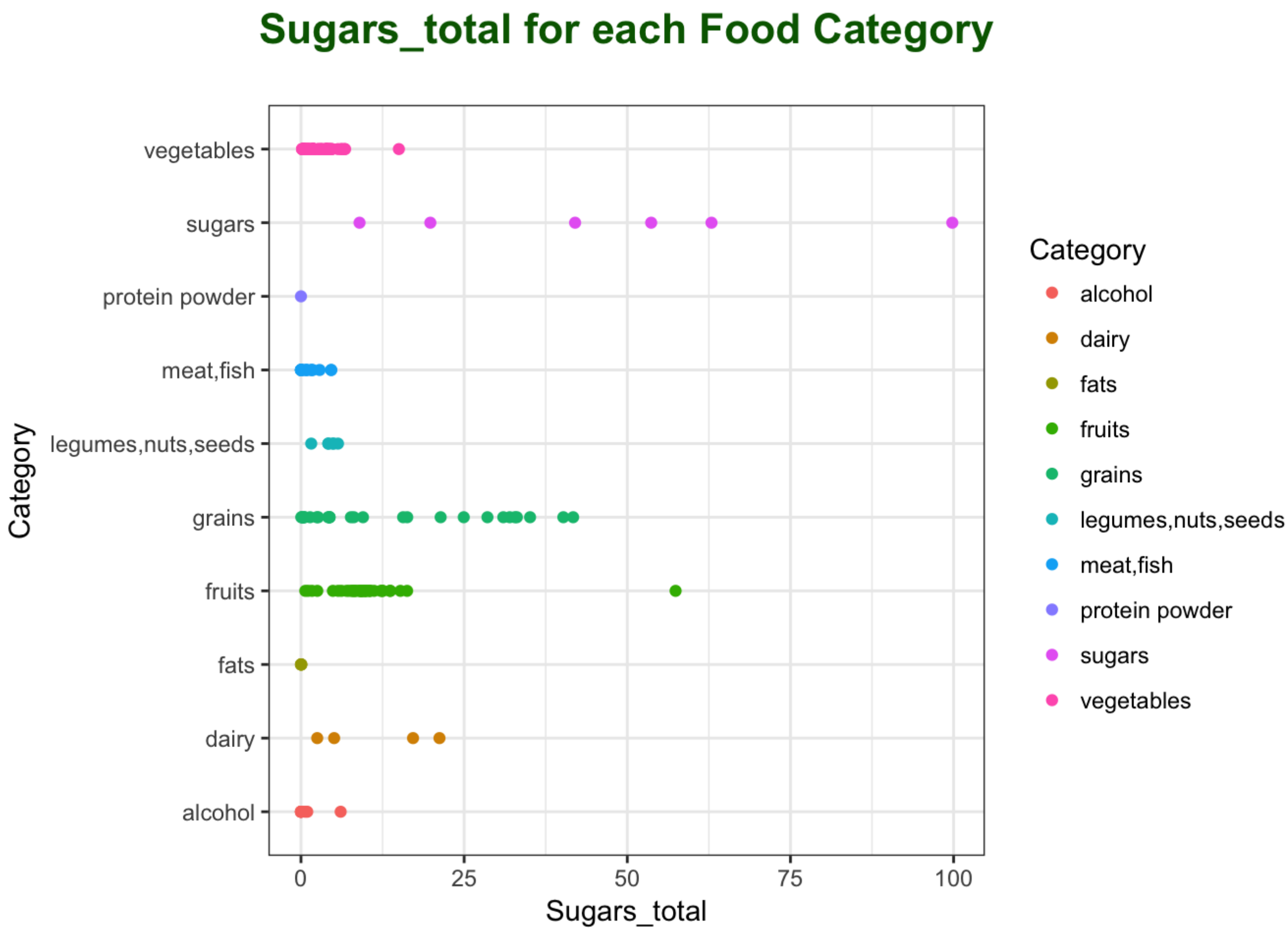
```
# Scatterplot of Cholesterol as a function of Fatty_acids_total_saturated
ggplot(food_df, aes(x=Fatty_acids_total_saturated, y=Cholesterol)) +
  geom_point(aes(shape=Category, color=Category), size=3.5) +
  scale_shape_manual(values=c(6, 4, 18, 17, 5, 16, 15, 8, 10, 11)) +
  labs(title="Scatterplot of Cholesterol as a function of Fatty_acids_total_saturated fo
r each Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=12, face="bold", color="darkgreen"))
```

t of Cholesterol as a function of Fatty_acids_total_saturated for each Food Category



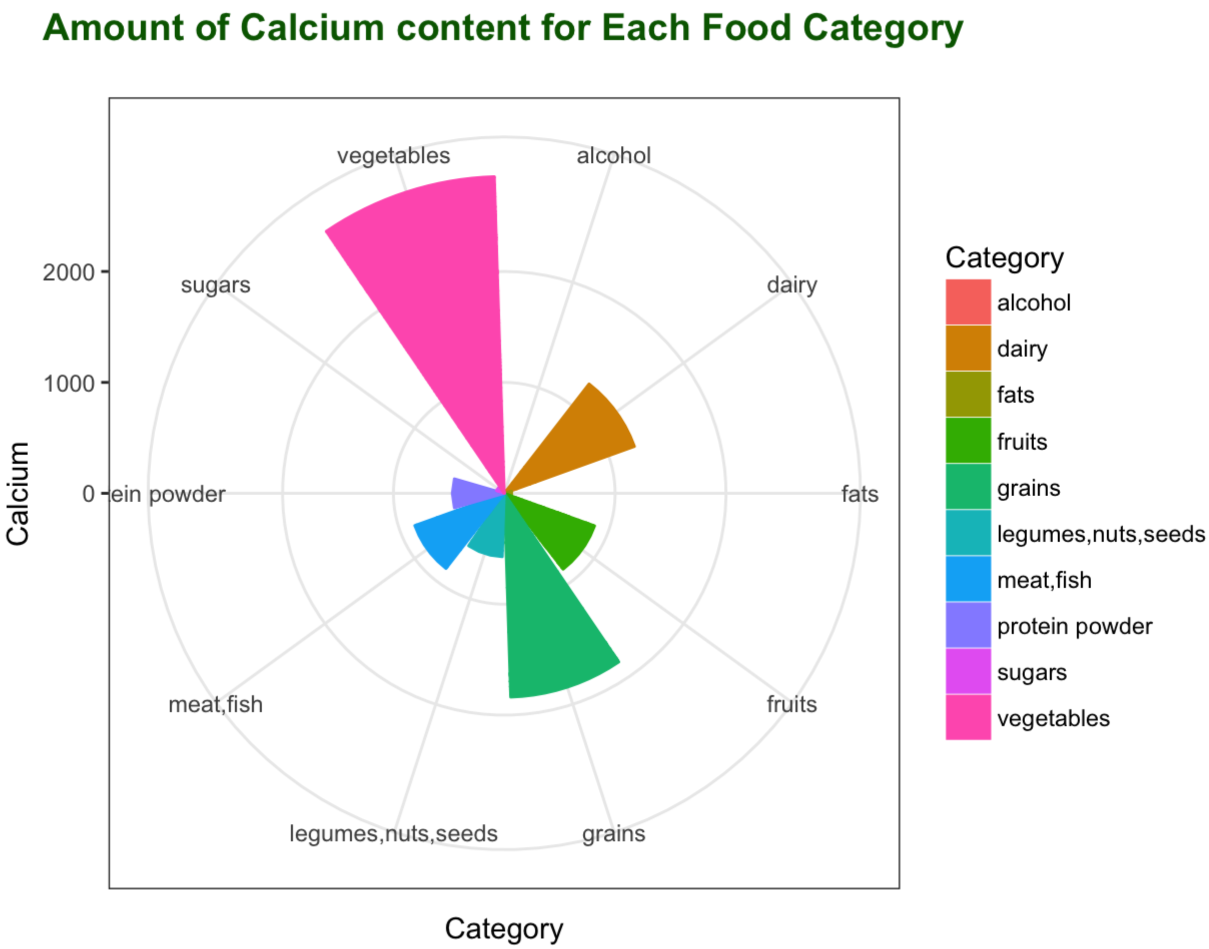
***Fats* category has the highest total saturated fatty acids and cholesterol. Meat and fish have second highest cholesterol, but dairy has the second highest total saturated fatty acids.**

```
# Sugars_total for each Food Category
ggplot(food_df, aes(x=Sugars_total, y=Category)) +
  geom_point(aes(color=Category)) +
  labs(title="Sugars_total for each Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=16, face="bold", color="darkgreen"))
```



Sugars category has the highest total sugars, followed by fruits. Protein powder and fats have no sugar content.

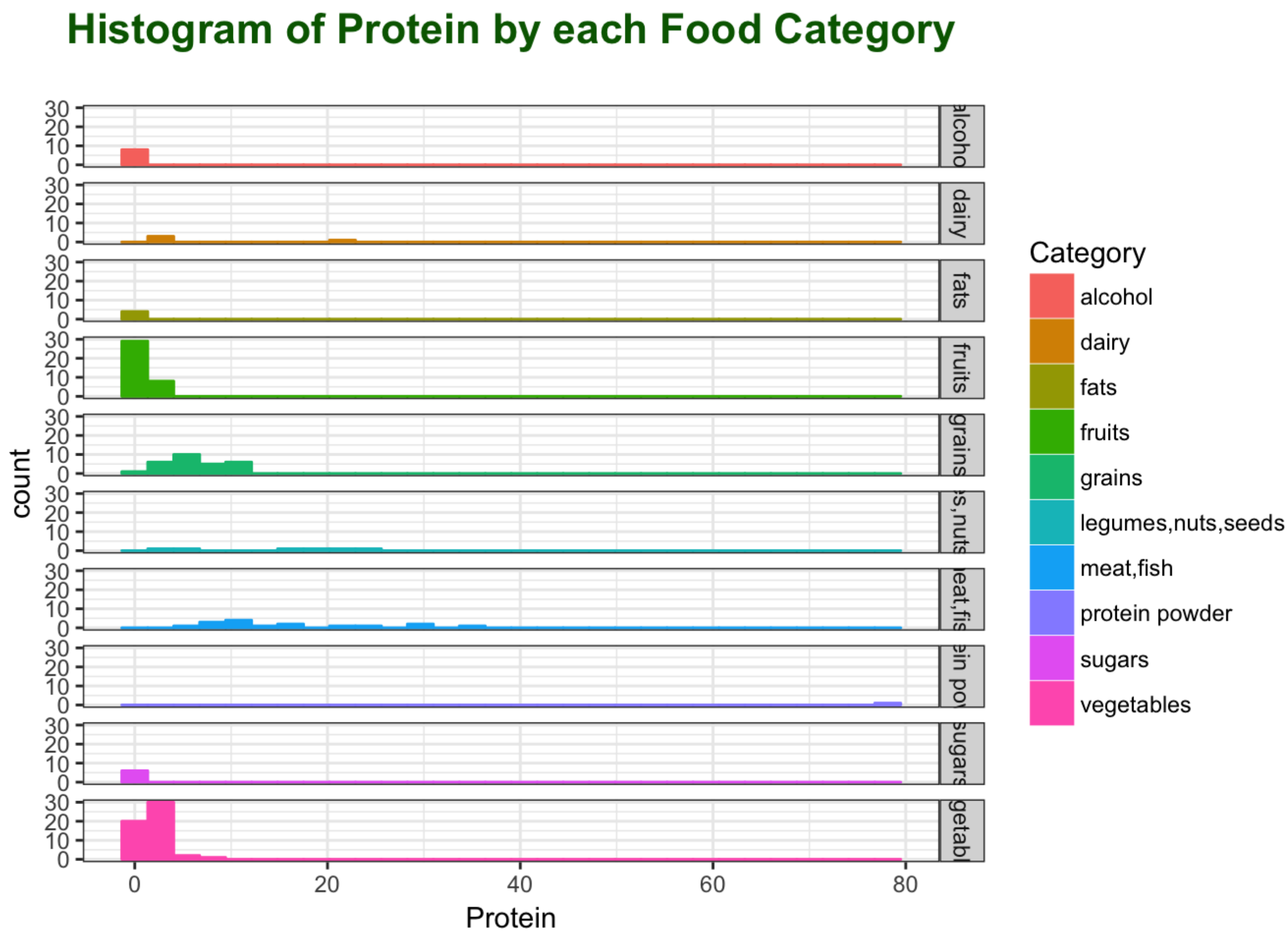
```
# Amount of Calcium content for Each Food Category
ggplot(food_df, aes(x=Category, y=Calcium))+
  geom_bar(stat="identity", aes(fill=Category, color=Category)) +
  coord_polar(theta = "x", direction=1 ) +
  labs(title="Amount of Calcium content for Each Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=14, face="bold", color="darkgreen"))
```



Vegetables contains the most calcium, followed by grains and dairy. Alcohol, fats and sugars do not seem to contain any calcium.

```
# Histogram of Protein by each Food Category
ggplot(food_df, aes(Protein))+
  geom_histogram(aes(color=Category, fill=Category)) +
  facet_grid(Category ~ .) +
  labs(title="Histogram of Protein by each Food Category\n") +
  theme_bw() +
  theme(plot.title=element_text(hjust=0.5, size=16, face="bold", color="darkgreen"))
```

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



Protein powder contains the highest protein amount, followed by meat, fish and legumes, nuts, seeds. Alcohol, fats and sugars do not contain any protein.

SQL Queries

Food_code column is added to the dataframe, so it can be used for joining with other tables for SQL queries.

```
df <- cbind(food_df, food_code=temp_df$food_code)
```

The following SQL queries are written to understand and explore the dataset.

```
library(sqldf)
#1
sqldf("SELECT Category, count(*) AS counts FROM df GROUP BY Category")
```

```
##              Category counts
## 1             alcohol      8
## 2               dairy      4
## 3                fats      4
## 4              fruits     37
## 5              grains     28
## 6  legumes,nuts,seeds      6
## 7          meat,fish     16
## 8      protein powder      1
## 9              sugars      6
## 10         vegetables     53
```

```
#2
# number of food high in Calcium
sqldf("SELECT Category, count(*) as counts FROM df WHERE Calcium > 100
      GROUP BY Category")
```

```
##             Category counts
## 1              dairy      3
## 2             grains      4
## 3 legumes,nuts,seeds      2
## 4          meat,fish      4
## 5     protein powder      1
## 6         vegetables      9
```

```
#3
# Percentage of each food category in regards to all food categories
sqldf(
  "SELECT Category, ROUND((cast(sub_query.counts as float)/sub_query.total)*100, 2) AS
  Percentage_of_Each_Category
  FROM (SELECT Category, count(*) AS counts,
  (SELECT count(*) FROM df) AS total FROM df GROUP BY Category) AS sub_query
  ORDER BY Percentage_of_Each_Category DESC")
```

```
##              Category Percentage_of_Each_Category
## 1          vegetables                       32.52
## 2              fruits                       22.70
## 3              grains                       17.18
## 4           meat,fish                        9.82
## 5             alcohol                        4.91
## 6  legumes,nuts,seeds                        3.68
## 7              sugars                        3.68
## 8               dairy                        2.45
## 9                fats                        2.45
## 10     protein powder                        0.61
```

```r
#4
# Averages of the nutrients
sqldf(
  "SELECT Category, AVG(Fatty_acids_total_monounsaturated) AS Avg_monounsat,
  Avg(Fatty_acids_total_polyunsaturated) AS Avg_polyunsat,
  Avg(Fatty_acids_total_saturated) AS Avg_sat
  FROM df
  GROUP BY Category")
```

```
##              Category Avg_monounsat Avg_polyunsat    Avg_sat
## 1             alcohol     0.0003750    0.00112500  0.0005000
## 2               dairy     3.3515000    0.36850000  6.3035000
## 3                fats    25.8405000   21.14275000 26.5865000
## 4              fruits     0.3227838    0.14345946  0.1084595
## 5              grains     2.5535000    2.20717857  1.5891429
## 6  legumes,nuts,seeds    19.3080000    8.54266667  5.0283333
## 7           meat,fish     4.2552500    1.70062500  3.4866250
## 8      protein powder     0.1580000    0.29900000  0.7810000
## 9              sugars     0.0240000    0.04816667  0.0140000
## 10         vegetables     0.2671132    0.19726415  0.1198113
```

```r
sqldf(
  "SELECT Category, AVG(Calcium), AVG(Magnesium), AVG(Iron), AVG(Phosphorus),
  AVG(Potassium), AVG(Sodium), AVG(Zinc)
  FROM df
  GROUP BY Category")
```

```
##              Category AVG(Calcium) AVG(Magnesium) AVG(Iron)
## 1             alcohol      3.00000       3.625000 0.1275000
## 2               dairy    312.00000      18.000000 0.5100000
## 3                fats     16.75000       1.500000 0.0300000
## 4              fruits     23.40541      14.513514 0.4032432
## 5              grains     65.64286      39.000000 9.2978571
## 6  legumes,nuts,seeds     95.33333     166.666667 3.0800000
## 7           meat,fish     53.62500      19.687500 1.8225000
## 8      protein powder    469.00000     195.000000 1.1300000
## 9              sugars     12.33333       2.666667 0.1033333
## 10         vegetables     53.88679      25.849057 1.0935849
##    AVG(Phosphorus) AVG(Potassium) AVG(Sodium) AVG(Zinc)
## 1        10.000000       30.75000    2.750000 0.0575000
## 2       213.750000      168.00000  273.000000 1.0500000
## 3        15.000000       20.25000  499.250000 0.0325000
## 4        23.405405      209.51351    4.081081 0.1305405
## 5       145.821429      193.32143  420.035714 2.9385714
## 6       324.833333      509.33333  439.833333 3.6200000
## 7       165.562500      262.18750  662.625000 2.5537500
## 8      1321.000000      500.00000  156.000000 6.1800000
## 9         4.833333       19.16667   24.666667 0.1766667
## 10       47.132075      297.49057   88.377358 0.3813208
```

```
sqldf(
  "SELECT Category, AVG(Energy), AVG(Folate_total), AVG(Sugars_total),
  AVG(Total_Fat), AVG(Water)
  FROM df
  GROUP BY Category")
```

```
##              Category AVG(Energy) AVG(Folate_total) AVG(Sugars_total)
## 1             alcohol   157.25000           1.12500          0.957500
## 2               dairy   184.25000           7.25000         11.490000
## 3                fats   689.00000           1.50000          0.022500
## 4              fruits    59.86486          17.54054         10.206757
## 5              grains   284.78571         205.21429         14.408214
## 6  legumes,nuts,seeds   421.83333          61.66667          4.248333
## 7           meat,fish   203.87500          23.87500          1.239375
## 8      protein powder   352.00000          33.00000          0.000000
## 9              sugars   273.66667           0.00000         47.858333
## 10         vegetables    40.98113          41.13208          2.703774
##    AVG(Total_Fat) AVG(Water)
## 1       0.0062500   76.30750
## 2      10.7975000   65.72000
## 3      77.7300000   20.25250
## 4       0.7083784   83.58297
## 5       6.7828571   33.18821
## 6      34.4600000   27.62500
## 7      10.3487500   61.44375
## 8       1.5600000    3.44000
## 9       0.1283333   26.01500
## 10      0.6924528   88.23679
```

```
#5
# Max, min and avgerages of some nutrients
sqldf(
  "SELECT Category, MAX(Cholesterol), MIN(Cholesterol), AVG(Cholesterol)
  FROM df
  GROUP BY Category
  ORDER BY Category")
```

```
##              Category MAX(Cholesterol) MIN(Cholesterol) AVG(Cholesterol)
## 1             alcohol                0                0        0.0000000
## 2               dairy               87                1       34.7500000
## 3                fats              215                0       81.0000000
## 4              fruits                0                0        0.0000000
## 5              grains               44                0        3.8928571
## 6  legumes,nuts,seeds                7                0        1.3333333
## 7           meat,fish              112               14       52.6875000
## 8      protein powder               16               16       16.0000000
## 9              sugars                0                0        0.0000000
## 10         vegetables                4                0        0.0754717
```

```
sqldf(
  "SELECT Category, MAX(Protein), MIN(Protein), AVG(Protein)
  FROM df
  GROUP BY Category
  ORDER BY MAX(Protein)")
```

```
##              Category MAX(Protein) MIN(Protein) AVG(Protein)
## 1              sugars         0.07         0.00     0.015000
## 2             alcohol         0.46         0.00     0.085000
## 3                fats         0.85         0.00     0.472500
## 4              fruits         2.80         0.11     1.002973
## 5          vegetables         6.84         0.59     2.058868
## 6              grains        12.09         0.92     5.982500
## 7               dairy        22.72         2.09     7.897500
## 8  legumes,nuts,seeds        24.55         3.08    14.698333
## 9           meat,fish        36.08         5.23    16.244375
## 10     protein powder        78.13        78.13    78.130000
```

```
sqldf(
  "SELECT Category, MAX(Fiber_total_dietary), MIN(Fiber_total_dietary),
  AVG(Fiber_total_dietary)
  FROM df
  GROUP BY Category
  ORDER BY AVG(Fiber_total_dietary) DESC")
```

```
##              Category MAX(Fiber_total_dietary) MIN(Fiber_total_dietary)
## 1  legumes,nuts,seeds                     10.9                      3.0
## 2              grains                     11.8                      0.4
## 3      protein powder                      3.1                      3.1
## 4              fruits                     10.4                      0.3
## 5          vegetables                      5.1                      0.3
## 6           meat,fish                      1.7                      0.0
## 7              sugars                      2.4                      0.0
## 8               dairy                      0.7                      0.0
## 9             alcohol                      0.1                      0.0
## 10               fats                      0.0                      0.0
##    AVG(Fiber_total_dietary)
## 1                  6.000000
## 2                  3.342857
## 3                  3.100000
## 4                  2.618919
## 5                  2.269811
## 6                  0.562500
## 7                  0.400000
## 8                  0.175000
## 9                  0.012500
## 10                 0.000000
```

```
#6
# Sum of some food nutrients
sqldf(
  "SELECT Category, SUM(Total_Fat), SUM(Cholesterol), SUM(Carbohydrate),
  SUM(Protein), SUM(Sugars_total)
  FROM df
  GROUP BY Category")
```

```
##              Category SUM(Total_Fat) SUM(Cholesterol) SUM(Carbohydrate)
## 1             alcohol           0.05                0             16.11
## 2               dairy          43.19              139             55.28
## 3                fats         310.92              324              0.62
## 4              fruits          26.21                0            523.69
## 5              grains         189.92              109           1463.82
## 6  legumes,nuts,seeds         206.76                8            122.83
## 7           meat,fish         165.58              843            156.30
## 8      protein powder           1.56               16              6.25
## 9              sugars           0.77                0            440.96
## 10         vegetables          36.70                4            416.80
##    SUM(Protein) SUM(Sugars_total)
## 1          0.68              7.66
## 2         31.59             45.96
## 3          1.89              0.09
## 4         37.11            377.65
## 5        167.51            403.43
## 6         88.19             25.49
## 7        259.91             19.83
## 8         78.13              0.00
## 9          0.09            287.15
## 10       109.12            143.30
```

```
#7
# Distinct nutrtient descriptions
sqldf("SELECT DISTINCT(nutrient_description)
      FROM NutDesc
      LIMIT 10")
```

```
##    nutrient_description
## 1               Protein
## 2             Total Fat
## 3          Carbohydrate
## 4                Energy
## 5               Alcohol
## 6                 Water
## 7              Caffeine
## 8           Theobromine
## 9         Sugars, total
## 10 Fiber, total dietary
```

```r
#8
# Main and additional food description
head(
  sqldf(
    "SELECT f.main_food_description, afd.additional_food_description
    FROM foods f
    INNER JOIN AddFoodDesc afd ON f.food_code = afd.food_code")
)
```

```
##   main_food_description additional_food_description
## 1             Ice cream             NS as to flavor
## 2          Venison/deer                    elk, NFS
## 3               Bologna              German bologna
## 4               Bologna                       fried
## 5               Bologna                 ham bologna
## 6                Salami                cotto salami
```

```r
#9
# Additional food description that contain words with ES
sqldf(
  "SELECT f.main_food_description, afd.additional_food_description
  FROM foods f
    INNER JOIN AddFoodDesc afd ON f.food_code = afd.food_code
    WHERE additional_food_description LIKE '%ES%'
    ORDER BY main_food_description")
```

```
##   main_food_description      additional_food_description
## 1                Chives                      chives, NFS
## 2              Cilantro                 coriander leaves
## 3              Cilantro                  Chinese parsley
## 4           Corn flakes store brands (See also Toasties)
## 5           Rice, fried                     Chinese rice
## 6             Tangerine          mandarin orange, fresh
## 7              Tomatoes        plum and Italian tomatoes
```

```r
#10
# Food with the highest Cholesterol and Total Fat
head(
  sqldf(
    "SELECT f.main_food_description, nd.nutrient_description, nv.nutrient_value
    FROM foods f
      INNER JOIN FNDDSNutVal nv ON f.food_code = nv.food_code
      INNER JOIN NutDesc nd ON nv.nutrient_code = nd.nutrient_code
      WHERE nutrient_description IN ('Cholesterol', 'Total Fat')
      ORDER BY nutrient_value DESC")
)
```

```
##   main_food_description nutrient_description nutrient_value
## 1                 Butter          Cholesterol            215
## 2           Venison/deer          Cholesterol            112
## 3                 Salami          Cholesterol            108
## 4              Table fat          Cholesterol            108
## 5          Vegetable oil            Total Fat            100
## 6                Bologna          Cholesterol             94
```

```
#11
# food with the max and avg nutrient_value in any one nutrient description,
#  ordered by max nutrient_value
sqldf(
  "SELECT f.main_food_description, d.Category, nd.nutrient_description,
  AVG(nv.nutrient_value), MAX(nv.nutrient_value)
  FROM FNDDSNutVal nv
    INNER JOIN foods f ON f.food_code = nv.food_code
    INNER JOIN df d ON f.food_code = d.food_code
    INNER JOIN NutDesc nd ON nv.nutrient_code = nd.nutrient_code
    GROUP BY main_food_description
    ORDER BY MAX(nv.nutrient_value) DESC
    LIMIT 10")
```

```
##    main_food_description   Category nutrient_description
## 1                  Cress vegetables  Lutein + zeaxanthin
## 2                Spinach vegetables  Lutein + zeaxanthin
## 3                  Chard vegetables  Lutein + zeaxanthin
## 4           Sweet potato vegetables       Carotene, beta
## 5              Radicchio vegetables  Lutein + zeaxanthin
## 6                Carrots vegetables       Carotene, beta
## 7          Raw vegetable vegetables       Carotene, beta
## 8                  Salsa vegetables             Lycopene
## 9             Watercress vegetables  Lutein + zeaxanthin
## 10                 Basil vegetables  Lutein + zeaxanthin
##    AVG(nv.nutrient_value) MAX(nv.nutrient_value)
## 1                289.5841                  12500
## 2                313.8656                  12198
## 3                258.6790                  11000
## 4                203.3713                  10980
## 5                151.2046                   8832
## 6                208.5909                   8285
## 7                208.5909                   8285
## 8                124.1804                   6312
## 9                136.2379                   5767
## 10               161.1887                   5650
```

```r
#12
# High Energy, low cholesterol and low sugar
sqldf(
  "SELECT f.main_food_description, d.Category, d.Energy, d.Cholesterol, d.Sugars_total
  FROM foods f
    INNER JOIN df d ON f.food_code = d.food_code
    WHERE d.Energy >= 400 AND d.Cholesterol < 200 AND d.Sugars_total < 15
    ORDER BY d.Energy DESC")
```

```
##   main_food_description           Category Energy Cholesterol Sugars_total
## 1         Vegetable oil               fats    886           0         0.00
## 2             Table fat               fats    625         108         0.03
## 3           Mixed nuts legumes,nuts,seeds    609           0         4.20
## 4              Almonds legumes,nuts,seeds    598           0         4.86
## 5              Peanuts legumes,nuts,seeds    588           0         4.18
## 6          Cashew nuts legumes,nuts,seeds    574           0         5.01
## 7            Margarine               fats    528           1         0.00
```

```r
# High carb low fat food
sqldf(
  "SELECT DISTINCT f.main_food_description, d.Category, d.Carbohydrate, d.Total_Fat
  FROM foods f
    INNER JOIN df d ON f.food_code = d.food_code
    WHERE Carbohydrate > 50 AND Total_Fat < 50
    GROUP BY main_food_description
    ORDER BY Carbohydrate DESC")
```

```
##    main_food_description Category Carbohydrate Total_Fat
## 1                  Sugar   sugars        99.98      0.00
## 2                  Candy   sugars        98.00      0.20
## 3            Chewing gum   sugars        95.37      0.37
## 4           Frosted rice   grains        91.30      0.40
## 5            Fruit Rings   grains        88.00      3.40
## 6            Rice Flakes   grains        86.22      1.26
## 7            Chex cereal   grains        84.50      1.40
## 8            Corn flakes   grains        84.10      0.40
## 9               Pretzels   grains        79.97      3.47
## 10        Mueslix cereal   grains        77.80      4.90
## 11           Raisin bran   grains        77.52      2.46
## 12                Cereal   grains        73.23      6.73
## 13            Oat cereal   grains        73.23      6.73
## 14         Breakfast bar   grains        72.90      7.50
## 15               Granola   grains        69.76     12.99
## 16         Pancake syrup   sugars        69.60      0.10
## 17                 Syrup   sugars        68.45      0.08
## 18                Cookie   grains        65.76     24.28
## 19           Granola bar   grains        64.40     19.80
## 20              Tamarind   fruits        62.50      0.60
## 21                Muffin   grains        53.98     15.85
## 22              Tortilla   grains        51.21      6.70
```

```
#13
# Food highest in Energy
sqldf("SELECT f.main_food_description, afd.additional_food_description,
  d.Category, nd.nutrient_description, nv.nutrient_value
    FROM foods f
      INNER JOIN FNDDSNutVal nv ON f.food_code = nv.food_code
      INNER JOIN NutDesc nd ON nv.nutrient_code = nd.nutrient_code
      INNER JOIN AddFoodDesc afd ON f.food_code = afd.food_code
      INNER JOIN df d ON f.food_code = d.food_code
      WHERE nutrient_description = 'Energy'
      ORDER BY nutrient_value DESC
      LIMIT 5")
```

```
##   main_food_description        additional_food_description Category
## 1        Vegetable oil                             oil, NFS     fats
## 2          Granola bar             New Trail Granola Bars   grains
## 3          Granola bar   Sunbelt Granola Bar, all flavors   grains
## 4          Granola bar              with chocolate chips   grains
## 5          Granola bar with oats, sugar, raisins, coconut   grains
##   nutrient_description nutrient_value
## 1               Energy            886
## 2               Energy            471
## 3               Energy            471
## 4               Energy            471
## 5               Energy            471
```

```
# Food highest in Vitamin C
sqldf("SELECT f.main_food_description, afd.additional_food_description,
  d.Category, nd.nutrient_description, nv.nutrient_value
    FROM foods f
      INNER JOIN FNDDSNutVal nv ON f.food_code = nv.food_code
      INNER JOIN NutDesc nd ON nv.nutrient_code = nd.nutrient_code
      INNER JOIN AddFoodDesc afd ON f.food_code = afd.food_code
      INNER JOIN df d ON f.food_code = d.food_code
      WHERE nutrient_description = 'Vitamin C'
      ORDER BY nutrient_value DESC
      LIMIT 5")
```

```
##   main_food_description           additional_food_description   Category
## 1                Pepper     sweet pepper, raw, NS as to color vegetables
## 2            Kiwi fruit                             kiwifruit     fruits
## 3        Pepper, banana                 Hungarian wax pepper vegetables
## 4        Pepper, banana yellow peppers, NS as to sweet or hot vegetables
## 5                Lychee                                frozen     fruits
##   nutrient_description nutrient_value
## 1            Vitamin C           97.0
## 2            Vitamin C           92.7
## 3            Vitamin C           82.7
## 4            Vitamin C           82.7
## 5            Vitamin C           71.5
```

```
# Food highest in Sugars, total
sqldf("SELECT f.main_food_description, afd.additional_food_description,
  d.Category, nd.nutrient_description, nv.nutrient_value
    FROM foods f
      INNER JOIN FNDDSNutVal nv ON f.food_code = nv.food_code
      INNER JOIN NutDesc nd ON nv.nutrient_code = nd.nutrient_code
      INNER JOIN AddFoodDesc afd ON f.food_code = afd.food_code
      INNER JOIN df d ON f.food_code = d.food_code
      WHERE nutrient_description = 'Sugars, total'
      ORDER BY nutrient_value DESC
      LIMIT 5")
```

```
##   main_food_description      additional_food_description Category
## 1           Fruit Rings              Frosted Fruit Rings   grains
## 2           Fruit Rings                     store brands   grains
## 3           Raisin bran                     store brands   grains
## 4           Granola bar           New Trail Granola Bars   grains
## 5           Granola bar Sunbelt Granola Bar, all flavors   grains
##   nutrient_description nutrient_value
## 1        Sugars, total          41.70
## 2        Sugars, total          41.70
## 3        Sugars, total          31.98
## 4        Sugars, total          28.57
## 5        Sugars, total          28.57
```

```
#14
# Max monounsat, polyunsat, and sat in 'fats', 'legumes,nuts,seeds', 'grains' catagorie
s,
#  ordered by max_sat descending
sqldf(
  "SELECT sub_query.main_food_description, sub_query.Category, sub_query.Max_monounsat,
  sub_query.Max_polyunsat, sub_query.Max_sat
  FROM
    (SELECT f.main_food_description, d.Category, MAX(d.Fatty_acids_total_monounsaturate
d)
    AS Max_monounsat, MAX(d.Fatty_acids_total_polyunsaturated) AS Max_polyunsat,
    MAX(d.Fatty_acids_total_saturated) AS Max_sat
    FROM foods f
    INNER JOIN df d ON f.food_code = d.food_code
    GROUP BY main_food_description) AS sub_query
  WHERE Category IN ('fats', 'legumes,nuts,seeds', 'grains')
  ORDER BY Max_sat DESC
  LIMIT 5")
```

```
##   main_food_description           Category Max_monounsat Max_polyunsat
## 1                Butter               fats        21.021         3.043
## 2             Table fat               fats        20.184        14.760
## 3         Vegetable oil               fats        40.439        43.277
## 4             Margarine               fats        21.718        23.491
## 5           Cashew nuts legumes,nuts,seeds        27.317         7.836
##   Max_sat
## 1  51.368
## 2  31.727
## 3  13.083
## 4  10.168
## 5   9.157
```

```
#15
# Total nutrient value for each food description
sqldf("SELECT f.main_food_description, d.Category,
  SUM(nv.nutrient_value) AS total_nutrient_value
    FROM foods f
      INNER JOIN FNDDSNutVal nv ON f.food_code = nv.food_code
      INNER JOIN df d ON f.food_code = d.food_code
      GROUP BY main_food_description
      ORDER BY total_nutrient_value DESC
      LIMIT 10")
```

```
##    main_food_description   Category total_nutrient_value
## 1                Spinach vegetables            20401.261
## 2                  Cress vegetables            18822.967
## 3                  Chard vegetables            16814.138
## 4                Parsley vegetables            14282.706
## 5                Carrots vegetables            13558.406
## 6          Raw vegetable vegetables            13558.406
## 7           Sweet potato vegetables            13219.134
## 8                  Basil vegetables            10477.268
## 9              Radicchio vegetables             9828.299
## 10              Collards vegetables             9144.666
```

```
#16
# Total nutrient value for each food category
sqldf("SELECT d.Category, SUM(nv.nutrient_value) AS total_nutrient_value
    FROM df d
      INNER JOIN FNDDSNutVal nv ON d.food_code = nv.food_code
      GROUP BY Category
      ORDER BY total_nutrient_value DESC")
```

```
##              Category total_nutrient_value
## 1          vegetables           246011.547
## 2              grains            76045.294
## 3              fruits            47166.830
## 4           meat,fish            31054.898
## 5  legumes,nuts,seeds            18286.211
## 6                fats            12220.581
## 7               dairy             6578.268
## 8      protein powder             3467.775
## 9              sugars             2926.704
## 10            alcohol             2532.843
```

The following SQL queries are used for subsetting different food category datasets.

```
dairy_df <- sqldf("SELECT * FROM df WHERE Category = 'dairy'")
grains_df <- sqldf("SELECT * FROM df WHERE Category = 'grains'")
meat_df <- sqldf("SELECT * FROM df WHERE Category = 'meat,fish'")
nuts_df <- sqldf("SELECT * FROM df WHERE Category = 'legumes,nuts,seeds'")
fats_df <- sqldf("SELECT * FROM df WHERE Category = 'fats'")
fruit_veg_df <- sqldf("SELECT * FROM df WHERE Category IN ('fruits', 'vegetables')")
```

The SQL queries help us understand the ranges of nutritional content among various food categories, and also which categories/food are high in certain nutritional values. Next, some t-tests are run to infer difference in mean nutrient value for the population of foods from which this sample is drawn.

T-Tests

```
#1
t.test(dairy_df$Protein, meat_df$Protein)
```

```
##
##  Welch Two Sample t-test
##
## data:  dairy_df$Protein and meat_df$Protein
## t = -1.5257, df = 4.4307, p-value = 0.1949
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -22.971161   6.277411
## sample estimates:
## mean of x mean of y
##   7.89750  16.24437
```

p-value is greater than 0.05, fail to reject null hypothesis. There is no difference in mean nutrient value for these two categories.

```
#2
t.test(nuts_df$Total_Fat, grains_df$Total_Fat)
```

```
##
##  Welch Two Sample t-test
##
## data:  nuts_df$Total_Fat and grains_df$Total_Fat
## t = 2.6285, df = 5.1683, p-value = 0.04513
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##   0.8732843 54.4810014
## sample estimates:
## mean of x mean of y
## 34.460000  6.782857
```

p-value is less than 0.05, reject null hypothesis. There is a difference in mean nutrient value for these two categories.

```
#3
t.test(nuts_df$Energy, fats_df$Energy)
```

```
##
##  Welch Two Sample t-test
##
## data:  nuts_df$Energy and fats_df$Energy
## t = -2.0224, df = 7.9386, p-value = 0.07804
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -572.2058   37.8725
## sample estimates:
## mean of x mean of y
##  421.8333  689.0000
```

p-value is greater than 0.05, fail to reject null hypothesis. There is no difference in mean nutrient value for these two categories.

```
#4
t.test(fruit_veg_df$Fiber_total_dietary, grains_df$Fiber_total_dietary)
```

```
##
##  Welch Two Sample t-test
##
## data:  fruit_veg_df$Fiber_total_dietary and grains_df$Fiber_total_dietary
## t = -1.4071, df = 30.733, p-value = 0.1694
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -2.2772864  0.4182388
## sample estimates:
## mean of x mean of y
##  2.413333  3.342857
```

p-value is greater than 0.05, fail to reject null hypothesis. There is no difference in mean nutrient value for these two categories.

T-tests reveal that some food catogories contain similar mean value of nutritional contents.

Correlations

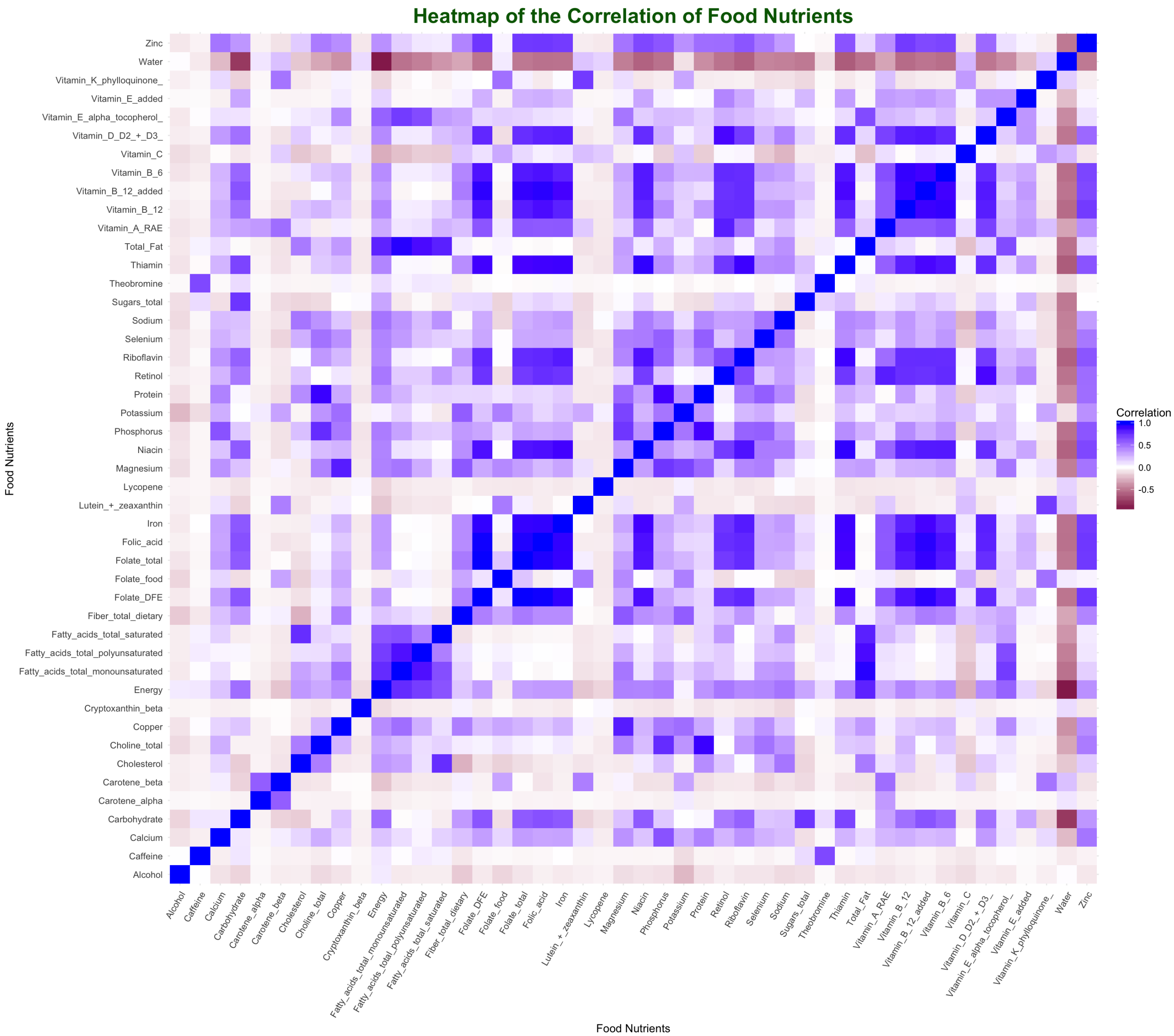
```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

```
library(scales)
# calculate correlation matrix
correlationMatrix <- cor(df[, -c(47, 48)])
# melt it into the long format
foodMelt <- melt(correlationMatrix, varnames=c("x", "y"), value.name="Correlation")
# order it according to the correlation
foodMelt <- foodMelt[order(foodMelt$Correlation), ]
```

```
# plot of correlation heatmap
ggplot(foodMelt, aes(x=x, y=y)) +
  geom_tile(aes(fill = Correlation)) +
  scale_fill_gradient2(low = muted("deeppink4"), mid = "white",
                       high = "blue")+
  labs(title="Heatmap of the Correlation of Food Nutrients") +
  theme_minimal()+
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
  theme(plot.title=element_text(hjust=0.5, size=20, face="bold",
                                color="darkgreen")) +
  xlab("Food Nutrients") +
  ylab("Food Nutrients")
```



The following code is then used to find the highly correlated variables (correlation > 0.5).

```
library(mlbench)
library(caret)
```

```
## Loading required package: lattice
```

```
## Warning in as.POSIXlt.POSIXct(Sys.time()): unknown timezone 'zone/tz/2017c.
## 1.0/zoneinfo/America/Los_Angeles'
```

```
highlyCorrelated <- findCorrelation(correlationMatrix, cutoff=0.5, names = TRUE)
highlyCorrelated
```

```
##  [1] "Water"
##  [2] "Niacin"
##  [3] "Thiamin"
##  [4] "Vitamin_B_12"
##  [5] "Vitamin_B_6"
##  [6] "Folate_total"
##  [7] "Riboflavin"
##  [8] "Folate_DFE"
##  [9] "Folic_acid"
## [10] "Vitamin_B_12_added"
## [11] "Iron"
## [12] "Energy"
## [13] "Zinc"
## [14] "Vitamin_D_D2_+_D3_"
## [15] "Retinol"
## [16] "Phosphorus"
## [17] "Vitamin_A_RAE"
## [18] "Magnesium"
## [19] "Selenium"
## [20] "Protein"
## [21] "Fiber_total_dietary"
## [22] "Copper"
## [23] "Fatty_acids_total_monounsaturated"
## [24] "Total_Fat"
## [25] "Vitamin_E_alpha_tocopherol_"
## [26] "Fatty_acids_total_saturated"
## [27] "Carotene_beta"
## [28] "Sugars_total"
## [29] "Folate_food"
## [30] "Vitamin_K_phylloquinone_"
## [31] "Theobromine"
```

Logistic Regression

```
# remove food_code column
fruit_veg_df$food_code <- NULL
# set fruits as 0 and vegetables as 1
fruit_veg_df$Category <- ifelse(fruit_veg_df$Category=='fruits', 0, 1)

fit_main <- glm(Category ~ ., family = binomial(), data = fruit_veg_df)
```

Summary(fit_main):
Null deviance: 1.2191e+02 on 89 degrees of freedom Residual deviance: 3.3005e-09 on 51 degrees of freedom AIC: 78

```
# Fit a regression model for the null model: Category as a function of the intercept onl
y.
fit_null <- glm(Category ~ 1, family = binomial(), data = fruit_veg_df)
```

Summary(fit_null): Null deviance: 121.91 on 89 degrees of freedom Residual deviance: 121.91 on 89 degrees of freedom AIC: 123.91

Next, step function is used for variable selection. The step function iterates through possible models, and return the optimal model with the lowest AIC.

The optimal model (AIC = 14) returned from the step function is then fitted as fit_final.

```
# final model returned from step function
fit_final <- glm(Category ~ Sugars_total + Sodium + Thiamin +
                 Fatty_acids_total_monounsaturated + Choline_total + Vitamin_B_6,
                 family = binomial(), data = fruit_veg_df)
summary(fit_final)
```

```
##
## Call:
## glm(formula = Category ~ Sugars_total + Sodium + Thiamin + Fatty_acids_total_monounsa
turated +
##     Choline_total + Vitamin_B_6, family = binomial(), data = fruit_veg_df)
##
## Deviance Residuals:
##        Min          1Q      Median          3Q         Max
## -2.673e-04  -2.100e-08   2.100e-08   2.100e-08   2.405e-04
##
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)                       -408.354  50551.764  -0.008    0.994
## Sugars_total                       -49.510   6455.383  -0.008    0.994
## Sodium                               3.331    950.858   0.004    0.997
## Thiamin                           3719.734 542663.358   0.007    0.995
## Fatty_acids_total_monounsaturated  -396.264 104764.010  -0.004    0.997
## Choline_total                       57.530   7162.763   0.008    0.994
## Vitamin_B_6                        881.080 129863.049   0.007    0.995
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.2191e+02  on 89  degrees of freedom
## Residual deviance: 1.9737e-07  on 83  degrees of freedom
## AIC: 14
##
## Number of Fisher Scoring iterations: 25
```

```r
# Written by Andy Field
logisticPseudoR2s <- function(LogModel) {
  dev <- LogModel$deviance
  nullDev <- LogModel$null.deviance
  modelN <-  length(LogModel$fitted.values)
  R.l <-  1 -  dev / nullDev
  R.cs <- 1- exp ( -(nullDev - dev) / modelN)
  R.n <- R.cs / ( 1 - ( exp (-(nullDev / modelN))))
  cat("Pseudo R^2 for logistic regression\n")
  cat("Hosmer and Lemeshow R^2  ", round(R.l, 3), "\n")
  cat("Cox and Snell R^2        ", round(R.cs, 3), "\n")
  cat("Nagelkerke R^2           ", round(R.n, 3),    "\n")
}
logisticPseudoR2s(fit_final)
```

```
## Pseudo R^2 for logistic regression
## Hosmer and Lemeshow R^2   1
## Cox and Snell R^2         0.742
## Nagelkerke R^2            1
```

```r
# Anova Test
a_mcv <- anova(fit_final)
a_mcv
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Category
##
## Terms added sequentially (first to last)
##
##
##                                 Df Deviance Resid. Df Resid. Dev
## NULL                                               89    121.907
## Sugars_total                     1   57.003        88     64.904
## Sodium                           1   19.690        87     45.214
## Thiamin                          1    8.958        86     36.256
## Fatty_acids_total_monounsaturated  1    6.691        85     29.565
## Choline_total                    1   12.712        84     16.853
## Vitamin_B_6                      1   16.853        83      0.000
```