CMTH 642 Data Analytics: Advanced Methods

Assignment 1 (10%)

Tasdeed Aziz

Section:DHB & ID:500945638

Due:23th May,Time:11:30PM

```
setwd("/Users/ayon/Desktop/CMTH642/Dataset")
macro <- read.csv("USDA_Macronutrients.csv", header= T)
micro <- read.csv("USDA_Micronutrients.csv", header = T)
head(macro, n = 5)
```

1. Read the csv files in the folder. (3 points)

```
##     ID                                       Description Calories Protein
## 1 2047                                      SALT,TABLE        0       0
## 2 2048                                    VINEGAR,CIDER       21       0
## 3 2053                                 VINEGAR,DISTILLED       18       0
## 4 2073          CAMPBELL SOUP CO,PACE,DRY TACO SEAS MIX      188       0
## 5 6597 CAMPBELL SOUP COMPANY,PACE,CHIPOTLE CHUNKY SALSA       25       0
##   TotalFat Carbohydrate
## 1        0         0.00
## 2        0         0.93
## 3        0         0.04
## 4        0        56.29
## 5        0         6.25
```

```
head(micro, n = 5)
```

```
##      ID Sodium Cholesterol Sugar Calcium  Iron Potassium VitaminC VitaminE
## 1  4038      0           0  0.00       0  0.00         0      0.0   149.40
## 2  8504    813          NA 17.17      45 67.67       630    239.7    80.46
## 3 25021    386           0 16.90     886 14.20       412     68.0    64.25
## 4  8590    242           0 14.30      47  8.70       296     89.0    58.96
## 5  4532      0           0  0.00       0  0.00         0      0.0    47.20
##   VitaminD
## 1      0.0
## 2       NA
## 3      3.1
## 4      0.0
## 5       NA
```

```
USDA <- merge(macro,micro, by="ID")
head(USDA, n=10)
```

2. Merge the data frames using the variable "ID". Name the Merged Data Frame "USDA". (6 points)

```
##      ID              Description Calories Protein TotalFat Carbohydrate Sodium
## 1  1001         BUTTER,WITH SALT      717    0.85    81.11         0.06    714
## 2  1002 BUTTER,WHIPPED,WITH SALT      717    0.85    81.11         0.06    827
## 3  1003     BUTTER OIL,ANHYDROUS      876    0.28    99.48         0.00      2
## 4  1004             CHEESE,BLUE      353   21.40    28.74         2.34  1,395
## 5  1005            CHEESE,BRICK      371   23.24    29.68         2.79    560
## 6  1006             CHEESE,BRIE      334   20.75    27.68         0.45    629
## 7  1007         CHEESE,CAMEMBERT      300   19.80    24.26         0.46    842
## 8  1008           CHEESE,CARAWAY      376   25.18    29.20         3.06    690
## 9  1009           CHEESE,CHEDDAR      403   24.90    33.14         1.28    621
## 10 1010          CHEESE,CHESHIRE      387   23.37    30.60         4.78    700
##    Cholesterol Sugar Calcium Iron Potassium VitaminC VitaminE VitaminD
## 1          215  0.06      24 0.02        24        0     2.32      1.5
## 2          219  0.06      24 0.16        26        0     2.32      1.5
## 3          256  0.00       4 0.00         5        0     2.80      1.8
## 4           75  0.50     528 0.31       256        0     0.25      0.5
## 5           94  0.51     674 0.43       136        0     0.26      0.5
## 6          100  0.45     184 0.50       152        0     0.24      0.5
## 7           72  0.46     388 0.33       187        0     0.21      0.4
## 8           93    NA     673 0.64        93        0       NA       NA
## 9          105  0.52     721 0.68        98        0     0.29      0.6
## 10         103    NA     643 0.21        95        0       NA       NA
```

```
sapply(USDA,class)
```

3. Check the datatypes of the attributes. Delete the commas in the Sodium and Potasium records. Assign Sodium and Potasium as numeric data types. (6 points)

```
##          ID  Description     Calories      Protein     TotalFat Carbohydrate
##    "integer"     "factor"    "integer"    "numeric"    "numeric"    "numeric"
##       Sodium  Cholesterol        Sugar      Calcium         Iron    Potassium
##     "factor"    "integer"    "numeric"    "integer"    "numeric"     "factor"
##     VitaminC     VitaminE     VitaminD
##    "numeric"    "numeric"    "numeric"
```

```
USDA$Sodium <- as.numeric(gsub(",","",USDA$Sodium))
USDA$Potassium <- as.numeric(gsub(",","",USDA$Potassium))
#sapply(USDA,class)
```

```r
#Total observation
nrow(USDA)
```

4. Remove records (rows) with missing values in more than 4 attributes (columns). How many records remain in the data frame? (6 points)

```
## [1] 7057
```

```r
##Remove the missing values
USDA.rm <- USDA[rowSums(is.na(USDA)) < 5,]
##Remaining records/observation
nrow(USDA.rm)
```

```
## [1] 6887
```

```r
sugar <- round(mean(USDA.rm$Sugar,na.rm = T),digits = 2)
vitaE <- round(mean(USDA.rm$VitaminE,na.rm = T),digits = 2)
vitaD <- round(mean(USDA.rm$VitaminD,na.rm = T),digits = 2)

s <- is.na(USDA.rm$Sugar)
e <- is.na(USDA.rm$VitaminE)
d <- is.na(USDA.rm$VitaminD)

USDA.rm$Sugar[s] = sugar
USDA.rm$VitaminE[e] = vitaE
USDA.rm$VitaminD[d] = vitaD

#Random column check to see the missing values are replace
#View(USDA.rm$Sugar[21])
#View(USDA.rm$VitaminE[10])
#View(USDA.rm$VitaminD[8])
```

5. For records with missing values for Sugar, Vitamin E and Vitamin D, replace missing values with mean value for the respective variable. (6 points)

```r
USDAclean <- na.omit(USDA.rm)
nrow(USDAclean)
```

6. With a single line of code, remove all remaining records with missing values. Name the new Data Frame "USDAclean". How many records remain in the data frame? (6 points)

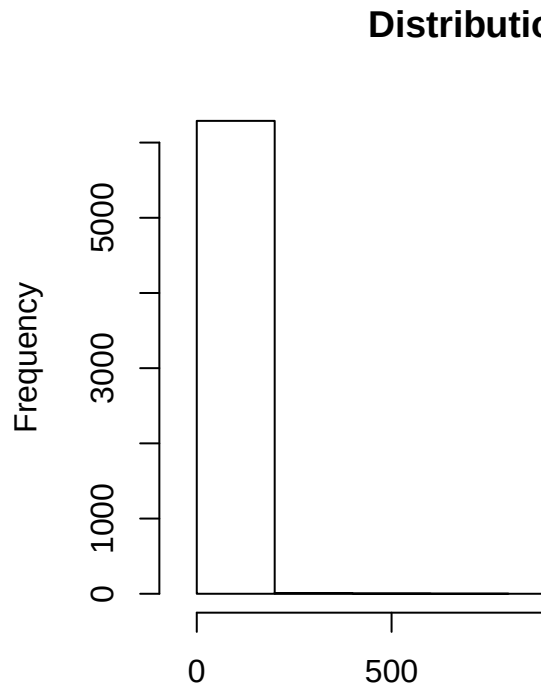
```
## [1] 6310
```

```
USDAclean$Description[which.max(USDAclean$Sodium)]
```

7. Which food has the highest sodium level? (6 points)

```
## [1] SALT,TABLE
## 7053 Levels: ABALONE,MIXED SPECIES,RAW ABALONE,MXD SP,CKD,FRIED ... ZWIEBACK
```

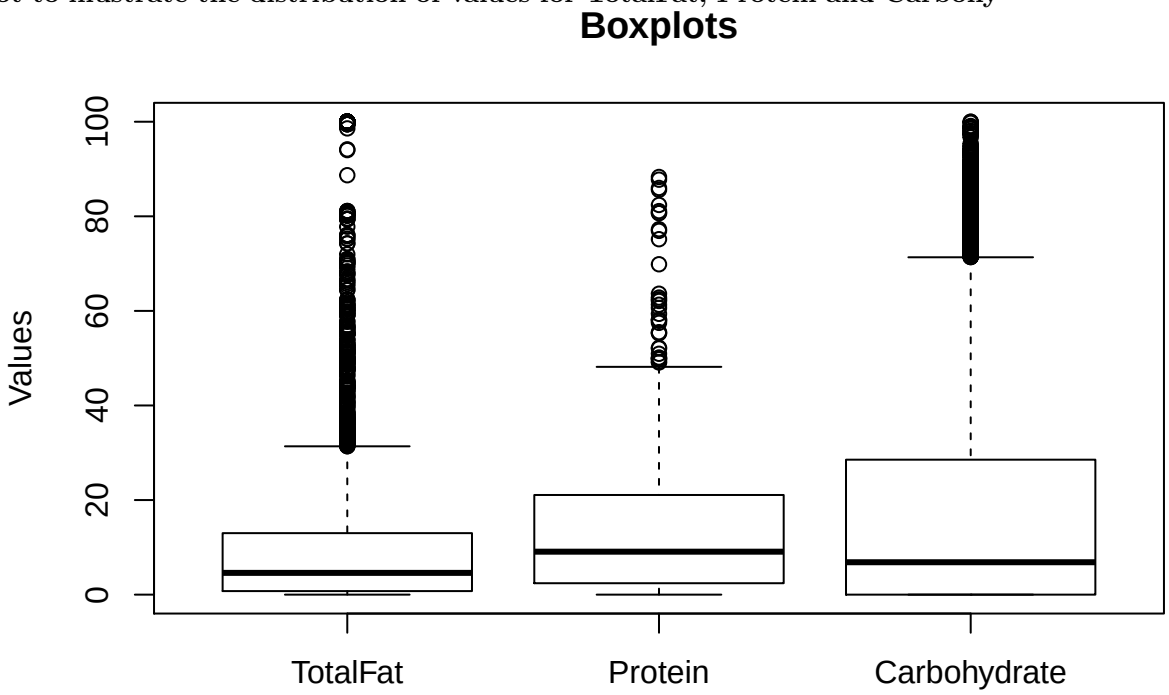
```
hist(x=USDAclean$VitaminC, xlab ='VitaminC', main = 'Distribution of VitaminC in Foods')
```



8. Create a histogram of Vitamin C distribution in foods. (6 points)

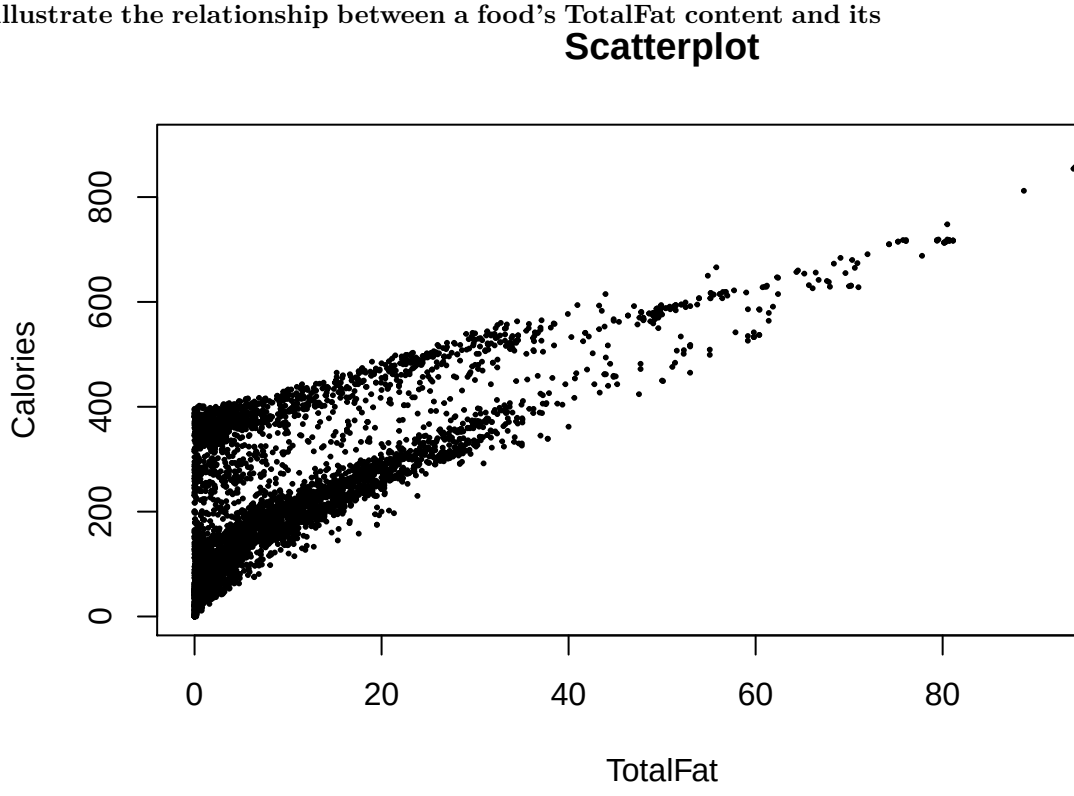
```
boxplot(USDAclean$TotalFat,USDAclean$Protein,USDAclean$Carbohydrate, names = c('TotalFat','Protein','Ca
```

9. Create a boxplot to illustrate the distribution of values for TotalFat, Protein and Carbohydrate. (6 points)




```
plot(x=USDAclean$TotalFat, y=USDAclean$Calories, xlab = 'TotalFat', ylab = 'Calories', main = 'Scatterpl
```

10. Create a scatterplot to illustrate the relationship between a food's TotalFat content and its Calorie content. (6 points)



```
USDAclean$HighSodium <- ifelse(mean(USDAclean$Sodium)<USDAclean$Sodium,1,0)
USDAclean$HighCalories <- ifelse(mean(USDAclean$Calories)<USDAclean$Calories,1,0)
USDAclean$HighProtein <-ifelse(mean(USDAclean$Protein)<USDAclean$Protein,1,0)
USDAclean$HighSugar <-ifelse(mean(USDAclean$Sugar)<USDAclean$Sugar,1,0)
USDAclean$HighFat  <- ifelse(mean(USDAclean$TotalFat)<USDAclean$TotalFat,1,0)
#head(USDAclean$HighFat)
table(USDAclean$HighSodium, USDAclean$HighFat)
```

11. Add a variable to the data frame that takes value 1 if the food has higher sodium than average, 0 otherwise. Call this variable HighSodium. Do the same for High Calories, High Protein, High Sugar, and High Fat. How many foods have both high sodium and high fat? (8 points)

```
##
##        0    1
##   0 3233 1250
##   1 1183  644
```

```
#644 foods have both high sodium and hig fat
```

```
aggregate(USDAclean$Iron,list(USDAclean$HighProtein), FUN=mean)
```

12. Calculate the average amount of iron, for high and low protein foods. (8 points)

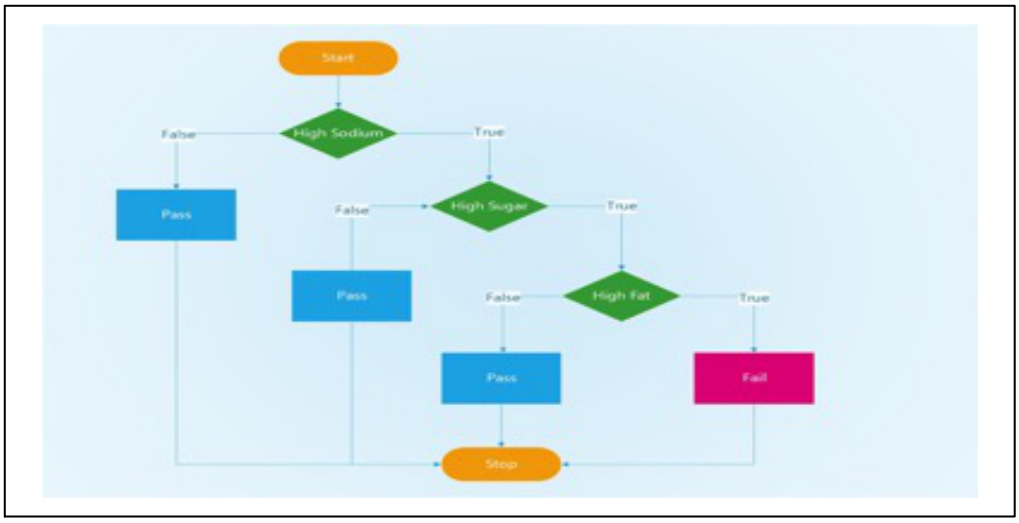
```
##   Group.1        x
## 1       0 2.696634
## 2       1 3.069541
```

```
require(jpeg)
```

13. Create a script for a "HealthCheck" program to detect unhealthy foods. Use the algorithm flowchart below as a basis for this script. (8 points)

```
## Loading required package: jpeg
```

```
img<-readJPEG("/Users/ayon/Desktop/CMTH642/Dataset/HealthCheck.jpg")
plot(1:4, ty = 'n', ann = F, xaxt = 'n', yaxt = 'n')
rasterImage(img,1,1,4,4)
```




```
healthcheck <- function(HighSodium,HighSugar,HighFat){
  ifelse(HighSodium==0,'Pass',ifelse(HighSugar==0,'Pass',ifelse(HighFat==0,'Pass','Fail')))
}
```

```
USDAclean$HealthCheck <- healthcheck(USDAclean$HighSodium,USDAclean$HighSugar,USDAclean$HighFat)
head(USDAclean$HealthCheck, n=10)
```

14. Add a new variable called HealthCheck to the data frame using the output of the function. (8 points)

```
##  [1] "Pass" "Pass" "Pass" "Pass" "Pass" "Pass" "Pass" "Fail" "Pass" "Fail"
```

```
sum(USDAclean$HealthCheck=='Fail')
```

15. How many foods in the USDAclean data frame fail the HealthCheck? (8 points)

```
## [1] 237
```

```
#237 foods fail the HealthCheck in the dataframe USDAclean
```

```
USDAclean_Aziz <- USDAclean
```

16. Save your final data frame as "USDAclean_ [your last name]." (3 points) This is the end of Assignment 1

Ceni Babaoglu, PhD