CMTH 642 Data Analytics: Advanced Methods

Assignment 1 (10%)

Tusaif Azmat

CMTH642-DJ0 500660278

---

library(rmarkdown) render("C:/Users/Zanara/Documents/Ryerson/Winter2022/CMTH642/CMTH642_winter 2022/A1/A1/CMTH 642 Assignment 1.Rmd", output_format = "word_document") #### 1. Read the csv files in the folder. (4 points)

```
df_usda_macro <- read.csv("USDA_Macronutrients.csv")
df_usda_micro <- read.csv("USDA_Micronutrients.csv")
```

2. Merge the data frames using the variable "ID". Name the Merged Data Frame "USDA". (4 points)

```
USDA <- merge(df_usda_macro, df_usda_micro, by='ID')
summary(USDA)
```

```
##        ID         Description           Calories        Protein       
TotalFat
##  Min.   : 1001   Length:7057        Min.   :  0.0   Min.   : 0.00   Min.   
:  0.00
##  1st Qu.: 8387   Class :character   1st Qu.: 85.0   1st Qu.: 2.29   1st 
Qu.:  0.72
##  Median :13293   Mode  :character   Median :181.0   Median : 8.20   Median 
:  4.37
##  Mean   :14258                      Mean   :219.7   Mean   :11.71   Mean   
: 10.32
##  3rd Qu.:18336                      3rd Qu.:331.0   3rd Qu.:20.43   3rd 
Qu.: 12.70
##  Max.   :93600                      Max.   :902.0   Max.   :88.32   Max.   
:100.00
##                                                                             
##   Carbohydrate       Sodium           Cholesterol          Sugar       
##  Min.   :  0.00   Length:7057        Min.   :   0.00   Min.   : 0.000  
##  1st Qu.:  0.00   Class :character   1st Qu.:   0.00   1st Qu.: 0.000  
##  Median :  7.13   Mode  :character   Median :   3.00   Median : 1.395  
##  Mean   : 20.70                      Mean   :  41.55   Mean   : 8.257  
##  3rd Qu.: 28.17                      3rd Qu.:  69.00   3rd Qu.: 7.875  
##  Max.   :100.00                      Max.   :3100.00   Max.   :99.800  
##                                      NA's   :287       NA's   :1909    
##     Calcium              Iron          Potassium            VitaminC       
##  Min.   :   0.00   Min.   :  0.000   Length:7057        Min.   :   0.000  
##  1st Qu.:   9.00   1st Qu.:  0.520   Class :character   1st Qu.:   0.000
##  Median :  19.00   Median :  1.330   Mode  :character   Median :   0.000
##  Mean   :  73.53   Mean   :  2.828                      Mean   :   9.436
##  3rd Qu.:  56.00   3rd Qu.:  2.620                      3rd Qu.:   3.100
##  Max.   :7364.00   Max.   :123.600                      Max.   :2400.000
##  NA's   :135       NA's   :122                          NA's   :331
##     VitaminE          VitaminD
##  Min.   :  0.000   Min.   :  0.0000
##  1st Qu.:  0.120   1st Qu.:  0.0000
##  Median :  0.270   Median :  0.0000
##  Mean   :  1.488   Mean   :  0.5769
##  3rd Qu.:  0.710   3rd Qu.:  0.1000
##  Max.   :149.400   Max.   :250.0000
##  NA's   :2719      NA's   :2833
```

3. Check the datatypes of the attributes. Delete the commas in the Sodium and Potasium records. Assign Sodium and Potasium as numeric data types. (4 points)

```
sapply(USDA, class)

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

USDA$Sodium <- gsub(',','',USDA$Sodium)
USDA$Potassium <- gsub(',','', USDA$Potassium)

USDA$Sodium = as.numeric(USDA$Sodium)
USDA$Potassium = as.numeric(USDA$Potassium)
```

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

```
USDA.nacount <- apply(USDA,1, function(x) sum(is.na(x)))

USDATrim <- USDA[USDA.nacount <= 4,]
nrow(USDATrim)

## [1] 6887
```

5. For records with missing values for Sugar, Vitamin E and Vitamin D, replace missing values with mean value for the respective variable. (4 points)

```
USDA$Sugar[is.na(USDA$Sugar)] = mean(USDA$Sugar[!is.na(USDA$Sugar)])

USDA$VitaminE[is.na(USDA$VitaminE)] =
```

```r
mean(USDA$VitaminE[!is.na(USDA$VitaminE)])

USDA$VitaminD[is.na(USDA$VitaminD)] =
mean(USDA$VitaminD[!is.na(USDA$VitaminD)])
```

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? (5 points)

```r
USDAclean = USDA[complete.cases(USDA),]
# 6310 records remain
cat(nrow(USDAclean), " records remain in a the data frame.")

## 6310  records remain in a the data frame.
```

7. Which food has the highest sodium level? (5 points)

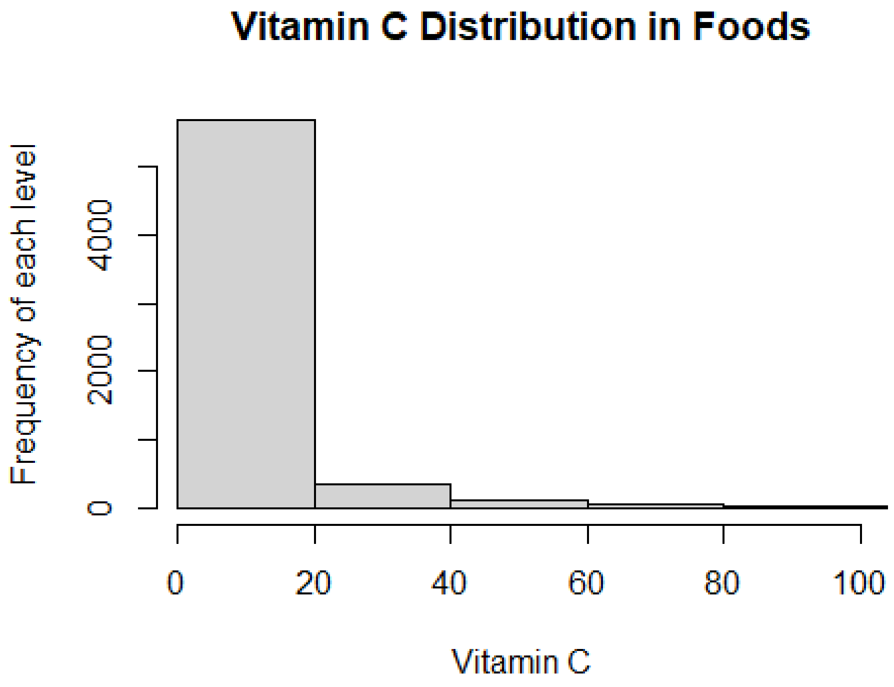
```r
USDAclean$Description[which.max(USDAclean$Sodium)]

## [1] "SALT,TABLE"

# Table Salt with ID 2047 has the highest Sodium of 38758
```

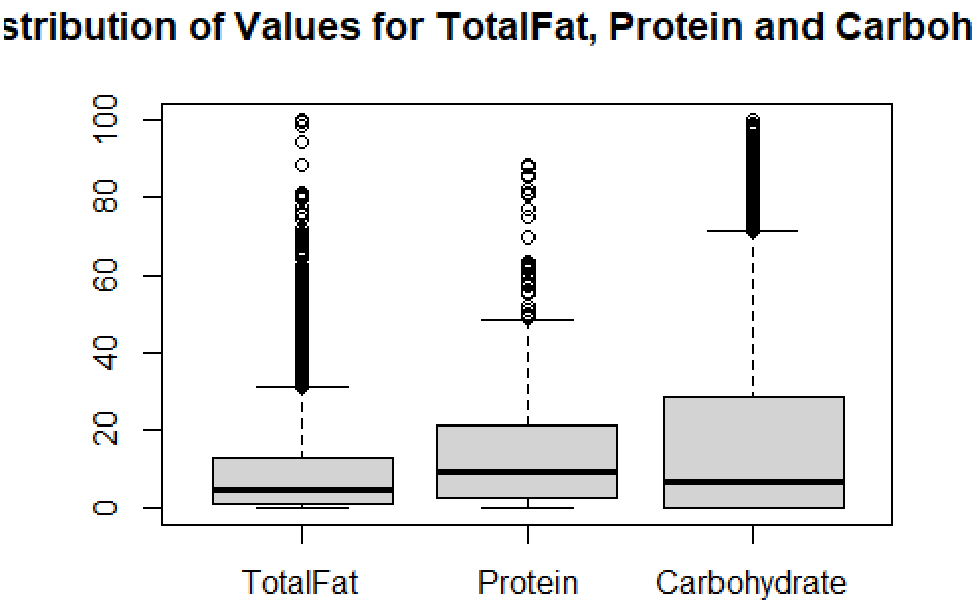
8. Create a histogram of Vitamin C distribution in foods. (5 points)

```r
hist(USDAclean$VitaminC, xlim = c(1, 100), breaks = 100, xlab = "Vitamin C",
ylab= "Frequency of each level", main="Vitamin C Distribution in Foods")
```



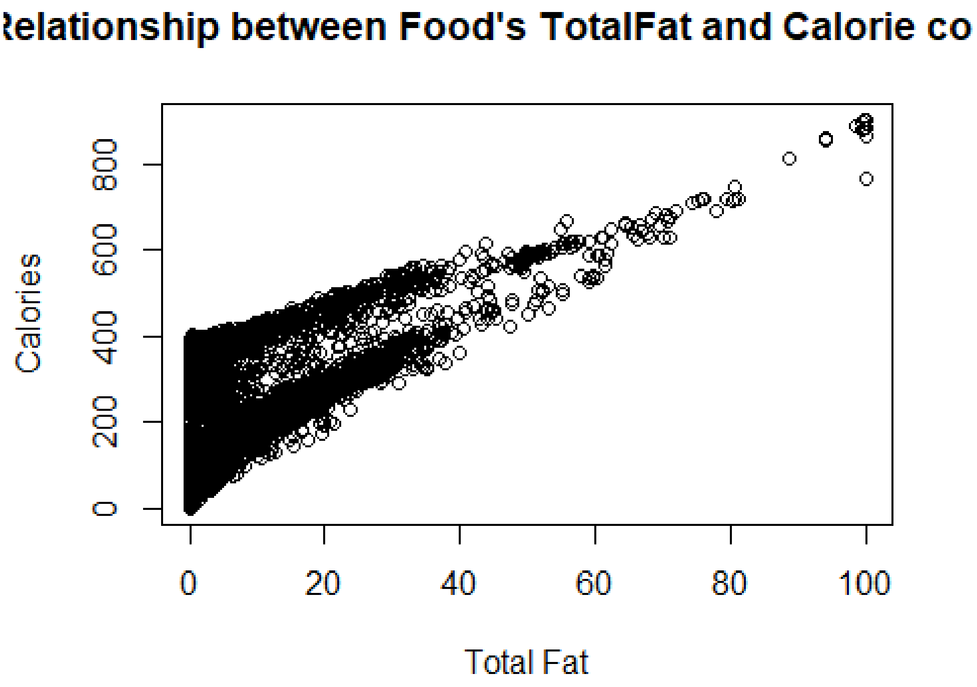
9. Create one boxplot to illustrate the distribution of values for TotalFat, Protein and Carbohydrate. (5 points)

```
TPC <- list(USDAclean$TotalFat, USDAclean$Protein, USDAclean$Carbohydrate)
names(TPC) <- c("TotalFat", "Protein", "Carbohydrate")
boxplot(TPC, main="Distribution of Values for TotalFat, Protein and
Carbohydrate")
```



10. Create a scatterplot to illustrate the relationship between a food's TotalFat content and its Calorie content. (5 points)

```
plot(USDAclean$Calories~USDAclean$TotalFat, main="Relationship between Food's
TotalFat and Calorie content", ylab="Calories", xlab="Total Fat")
```



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? (5 points)

```
USDAclean$HighSodium = 0
USDAclean$HighSodium[USDAclean$Sodium > mean(USDAclean$Sodium)] = 1

USDAclean$HighCalories = 0
USDAclean$HighCalories[USDAclean$Calories > mean(USDAclean$Calories)] = 1

USDAclean$HighProtein = 0
USDAclean$HighProtein[USDAclean$Protein > mean(USDAclean$Protein)] = 1

USDAclean$HighSugar = 0
USDAclean$HighSugar[USDAclean$Sugar > mean(USDAclean$Sugar)] = 1

USDAclean$HighFat = 0
USDAclean$HighFat[USDAclean$TotalFat > mean(USDAclean$TotalFat)] = 1

cat(sum(apply(USDAclean[c("HighSodium", "HighFat")], 1, function(x) sum(x) ==
2)), "foods have both high sodium and high fat.")

## 644 foods have both high sodium and high fat.

# 644 foods high sodium and high fat.
```

12. Calculate the average amount of iron, for high and low protein foods. (5 points)

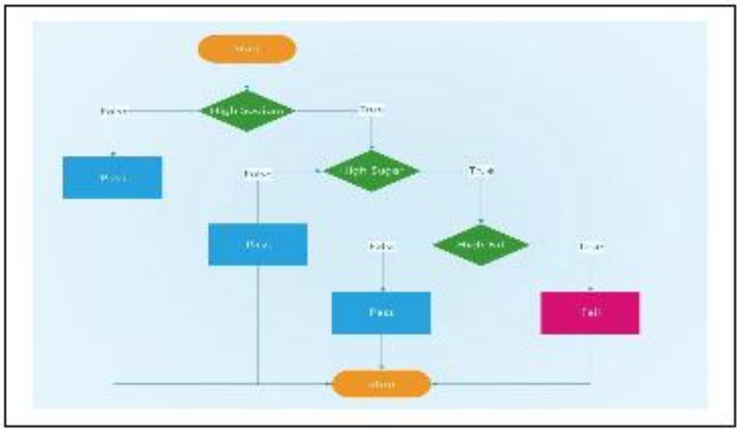
```
MeanProteinIron <- aggregate(USDAclean$Iron,list(USDAclean$HighProtein),FUN =
mean)
colnames(MeanProteinIron) <- c("low(0)/high(1) protein","AVG amount")
head(MeanProteinIron)

##   low(0)/high(1) protein AVG amount
## 1                      0   2.696634
## 2                      1   3.069541

# 3.069541 Iron for High Protein
# 2.696634 Iron for Low Protein
```

13. Create a function for a "HealthCheck" program to detect unhealthy foods. Use the algorithm flowchart below as a basis. (5 points)

```
require(jpeg)
img<-readJPEG("HealthCheck.jpg")
plot(1:4, ty = 'n', ann = F, xaxt = 'n', yaxt = 'n')
rasterImage(img,1,1,4,4)
```



```
HealthCheck <- function (sodium,sugar,fat) {
  ifelse (sodium==0, "Pass", ifelse (sugar==0, "Pass", ifelse (fat==0,
"Pass", "Fail")))
}
```

14. Add a new variable called HealthCheck to the data frame using the output of the function. (5 points)

```
USDAclean$HealthCheck = HealthCheck(USDAclean$HighSodium,
USDAclean$HighSugar, USDAclean$HighFat)
```

15. How many foods in the USDAclean data frame fail the HealthCheck? (5 points)

```
sum(USDAclean$HealthCheck == "Fail",na.rm = TRUE)

## [1] 237

# 237 food fail Health check
```

16. Visualize the correlation among Calories, Protein, Total Fat, Carbohydrate, Sodium and Cholesterol. (5 points)

```
cor(USDAclean[3:8])

##                Calories      Protein     TotalFat Carbohydrate
Sodium
## Calories     1.00000000  0.122122537  0.804495022   0.42460618
0.032321026
## Protein      0.12212254  1.000000000  0.057035611  -0.30471117 -
0.003489485
## TotalFat     0.80449502  0.057035611  1.000000000  -0.12434291
0.002916089
## Carbohydrate 0.42460618 -0.304711167 -0.124342914   1.00000000
0.046838692
## Sodium       0.03232103 -0.003489485  0.002916089   0.04683869
1.000000000
## Cholesterol  0.02391933  0.269854840  0.093289601  -0.21937986 -
0.017774863
##              Cholesterol
## Calories      0.02391933
## Protein       0.26985484
## TotalFat      0.09328960
## Carbohydrate -0.21937986
## Sodium       -0.01777486
## Cholesterol   1.00000000
```

17. Is the correlation between Calories and Total Fat statistically significant? Why? (5 points)

```
cor.test(USDAclean$Calories,USDAclean$TotalFat)

##
##  Pearson's product-moment correlation
##
## data:  USDAclean$Calories and USDAclean$TotalFat
## t = 107.58, df = 6308, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.7956139 0.8130305
## sample estimates:
```

```
##      cor
## 0.804495

#Yes, The correlation between Calories and Total Fat is statistically
significant.As described below, the p-value, 2.2e-16, is less than the
confidence level of 0.05.The relationship would be described as a high
positive relationship.
```

18. Create a Linear Regression Model, using Calories as the dependent variable Protein, Total Fat, Carbohydrate, Sodium and Cholesterol as the independent variables. (4 points)

```
lm_USDA <- lm(USDAclean$Calories ~ USDAclean$Protein + USDAclean$TotalFat+
               USDAclean$Carbohydrate + USDAclean$Sodium +
USDAclean$Cholesterol)
summary(lm_USDA)

## 
## Call:
## lm(formula = USDAclean$Calories ~ USDAclean$Protein + USDAclean$TotalFat + 
##     USDAclean$Carbohydrate + USDAclean$Sodium + USDAclean$Cholesterol)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -191.087   -3.832    0.426    5.147  291.011 
## 
## Coefficients:
##                         Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            3.9882753  0.4832629   8.253  < 2e-16 ***
## USDAclean$Protein      3.9891994  0.0233550 170.807  < 2e-16 ***
## USDAclean$TotalFat     8.7716980  0.0143291 612.158  < 2e-16 ***
## USDAclean$Carbohydrate 3.7432001  0.0091404 409.522  < 2e-16 ***
## USDAclean$Sodium       0.0003383  0.0002189   1.545    0.122    
## USDAclean$Cholesterol  0.0110138  0.0019861   5.545 3.05e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 18.92 on 6304 degrees of freedom
## Multiple R-squared:  0.9877, Adjusted R-squared:  0.9877 
## F-statistic: 1.009e+05 on 5 and 6304 DF,  p-value: < 2.2e-16

# y = 3.9882753 + 3.9891994(Protein) + 8.7716980(TotalFat) +
3.7432001(Carbohydrate) + 0.0003383(Sodium) + 0.0110138(Cholesterol)
```

19. Which independent variable is the least significant? Why? (4 points)

```
lm_Anova_USDA <- anova(lm_USDA)

lm_Anova_USDA

## Analysis of Variance Table
## 
## Response: USDAclean$Calories
##                           Df    Sum Sq   Mean Sq    F value    Pr(>F)
## USDAclean$Protein         1   2728899   2728899 7.6197e+03 < 2.2e-16 ***
## USDAclean$TotalFat        1 116762840 116762840 3.2603e+05 < 2.2e-16 ***
## USDAclean$Carbohydrate    1  61215495  61215495 1.7093e+05 < 2.2e-16 ***
## USDAclean$Sodium          1       789       789 2.2031e+00    0.1378    
## USDAclean$Cholesterol     1     11014     11014 3.0753e+01  3.05e-08 ***
## Residuals              6304   2257685       358                         
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

# Sodium is the least significant variable. We can interpret this from the
analysis of Variance Table. The p-value for sodium is 0.1378, which is not
very significant, especially compared to the p-values of the other variables,
which are all much smaller, less than 2e-16.
```

20. Create a new model by using only the significant independent variables. (4 points)

```
lm_USDA_new <- lm(USDAclean$Calories ~ USDAclean$Protein +
USDAclean$TotalFat+
               USDAclean$Carbohydrate + USDAclean$Cholesterol)
summary(lm_USDA_new)

## 
## Call:
## lm(formula = USDAclean$Calories ~ USDAclean$Protein + USDAclean$TotalFat + 
##     USDAclean$Carbohydrate + USDAclean$Cholesterol)
## 
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -191.220   -3.787    0.464    5.104  290.922 
## 
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)    
## (Intercept)            4.077907   0.479822   8.499  < 2e-16 ***
## USDAclean$Protein      3.989679   0.023355 170.824  < 2e-16 ***
## USDAclean$TotalFat     8.771904   0.014330 612.131  < 2e-16 ***
## USDAclean$Carbohydrate 3.743859   0.009131 409.996  < 2e-16 ***
## USDAclean$Cholesterol  0.010980   0.001986   5.528 3.36e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 18.93 on 6305 degrees of freedom
## Multiple R-squared:  0.9877, Adjusted R-squared:  0.9876 
## F-statistic: 1.261e+05 on 4 and 6305 DF,  p-value: < 2.2e-16

lm_Anova_USDA_new <- anova(lm_USDA_new)

lm_Anova_USDA_new

## Analysis of Variance Table
## 
## Response: USDAclean$Calories
##                          Df    Sum Sq   Mean Sq    F value    Pr(>F)    
## USDAclean$Protein         1   2728899   2728899   7618.067 < 2.2e-16 ***
## USDAclean$TotalFat        1 116762840 116762840 325958.246 < 2.2e-16 ***
## USDAclean$Carbohydrate    1  61215495  61215495 170890.802 < 2.2e-16 ***
## USDAclean$Cholesterol     1     10947     10947     30.561 3.365e-08 ***
## Residuals              6305   2258540       358
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

21. A new product is just produced with the following data: Protein=0.1, TotalFat=37, Carbohydrate=400, Cholesterol=75, Sugar=NA, Calcium=35, Iron=NA, Potassium=35, VitaminC=10, VitaminE=NA, VitaminD=NA. Based on the new model you created, what is the predicted value for Calories? (4 points)

```
New_Product <- data.frame(Protein=0.1, TotalFat=37, Carbohydrate=400,
Sodium=440, Cholesterol=75, Sugar=NA, Calcium=35, Iron=NA, Potassium=35,
VitaminC=10, VitaminE=NA, VitaminD=NA)



Predicted_Calories_value <- 3.9882753 + 3.9891994*New_Product$Protein +
8.7716980*New_Product$TotalFat + 3.7432001*New_Product$Carbohydrate +
0.0003383*New_Product$Sodium + 0.0110138*New_Product$Cholesterol

Predicted_Calories_value

## [1] 1827.195

#The predicted value would be 1827.195
```

22. If the Carbohydrate amount increases from 400 to 40000 (10000% increase), how much change will occur on Calories in percent? Explain why? (4 points)

```
Predicted_Calories_Increased <- 3.9882753 + 3.9891994*New_Product$Protein +
8.7716980*New_Product$TotalFat + 3.7432001*New_Product$Carbohydrate +
0.0003383*44440 + 0.0110138*New_Product$Cholesterol

Predicted_Calories_Increased

## [1] 1842.08

Change_in_Calories <- (44440-440)*0.0003383
Percentage_of_Change <- (Change_in_Calories/Predicted_Calories_value)* 100
Percentage_of_Change

## [1] 0.8146476

# If the value of Sodium increased from 440 to 44440, the value of Calories
would change by 14.8852.
# This represents a 0.81% change in the value of Calories from when Sodium
was equal to 440.
# To get this result we multiply the difference in the Sodium value from
before to after (44440-440 = 44000) by the coefficient for Sodium from the
model, which is 0.0003383.
```

```
# We use this value in our regression calculation. The coefficient describes
the change in the dependent variable for each unit of change in the Sodium
variable.
```

This is the end of Assignment 1

Ceni Babaoglu, PhD