Predicting calories from the quantity of nutrients in the food using Multiple Linear Regression

Mohsin Asif

February 20, 2018

Introduction:

In this project we are analyzing nutritional data to predict amount of calories in food. In our data set we have nutritional variables as covariates such as fat, sugar, carbohydrates, protein etc. we use these variables to predict response variable calories.

In the first part, we performed data cleaning by changing variable names and removing null values from our dataset. We then build linear regression model and performed model adequacy checking to make model more robust. Once we established that our regression is valid, we looked for multicollinearity in our dataset. We checked for multicollinearity in our model and it was very high and caused large Variance Inflation Factor(VIF). We did ANOVA test to check which variables are significant to get a clue on what will reduce multicollinearity in our model. We dropped some variables to reduce VIF to an acceptable level. Based on that we selected our final model.

Data Exploration and Data Cleaning:

Firstly we will read the data file and load the CAR library

install.packages(“car”) library(car)

```
library(car)

## Warning: package 'car' was built under R version 3.4.4

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.4.4

calories<-read.csv("C:/Users/Mohsin Asif/Box Sync/MS IS/Spring Semester/Flex
1/DAM/Final Project/Final Project Calories/calories.csv")

str(calories)

## 'data.frame':    126 obs. of  11 variables:
##  $ Fast.Food.Restaurant: Factor w/ 12 levels "Burger King",..: 8 8 8 8 8 8
8 8 8 8 ...
##  $ Type                : Factor w/ 6 levels "Breaded Chicken Sandwich",..:
2 2 2 2 2 2 6 1 5 3 ...
##  $ Serving.Size..g.    : int  98 113 211 202 270 283 257 213 200 65 ...
##  $ Calories            : int  240 290 530 520 720 750 530 510 350 190 ...
##  $ Total.Fat..g.       : num  8 11 27 26 40 43 15 22 9 12 ...
##  $ Saturated.Fat..g.   : num  3 5 10 12 15 19 10 3.5 2 2 ...
##  $ Trans.Fat..g.       : num  0 0.5 1 1.5 1.5 2.5 1 0 0 0 ...
##  $ Sodium..mg.         : int  480 680 960 1100 1470 1280 160 990 820 360
...
##  $ Carbs..g.           : num  32 33 47 41 51 42 86 55 42 12 ...
##  $ Sugars..g.          : num  6 7 9 10 14 10 63 10 8 0 ...
##  $ Protein..g.         : num  12 15 24 30 39 48 11 24 28 9 ...
```

We have some hard to handle variable names so we will replace them with names that are easy to read and work with.

```
names(calories)

##  [1] "Fast.Food.Restaurant" "Type"                 "Serving.Size..g."
##  [4] "Calories"             "Total.Fat..g."        "Saturated.Fat..g."
##  [7] "Trans.Fat..g."        "Sodium..mg."          "Carbs..g."
## [10] "Sugars..g."           "Protein..g."

names(calories)=c(
  "Fast Food Restaurant",
  "Type",
  "ServingSize",
  "Calories",
  "TotalFat",
  "SaturatedFat",
  "TransFat",
  "Sodium",
  "Carbs",
  "Sugars",
  "Protein"
)

str(calories)

## 'data.frame':    126 obs. of  11 variables:
##  $ Fast Food Restaurant: Factor w/ 12 levels "Burger King",..: 8 8 8 8 8 8
8 8 8 8 ...
##  $ Type                : Factor w/ 6 levels "Breaded Chicken Sandwich",..:
2 2 2 2 2 2 6 1 5 3 ...
##  $ ServingSize         : int  98 113 211 202 270 283 257 213 200 65 ...
##  $ Calories            : int  240 290 530 520 720 750 530 510 350 190 ...
##  $ TotalFat            : num  8 11 27 26 40 43 15 22 9 12 ...
##  $ SaturatedFat        : num  3 5 10 12 15 19 10 3.5 2 2 ...
##  $ TransFat            : num  0 0.5 1 1.5 1.5 2.5 1 0 0 0 ...
##  $ Sodium              : int  480 680 960 1100 1470 1280 160 990 820 360
...
##  $ Carbs               : num  32 33 47 41 51 42 86 55 42 12 ...
##  $ Sugars              : num  6 7 9 10 14 10 63 10 8 0 ...
##  $ Protein             : num  12 15 24 30 39 48 11 24 28 9 ...
```

This looks better!

Here is a short description of variables in our dataset: 1. FastFoodRest: which has the name of restaurant e.g. McDonalds, Wendy, Sonic etc. 2. Type: has the types of restaurants e.g. burger, MilkShake, Grilled Chicken etc. 3. ServingSize: contains the serving size in grams 4. Calories: has the number of calories per Serving Size 5. TotalFat: sum of saturated, monounsaturated and polyunsaturated fats in grams 6. SaturatedFat: saturated fat content in grams 7. TransFat: Trans fatty acids in grams which is unhealthy 8. Sodium_mg: Sodium content in milligrams 9. Protein: Protein content in grams

We have two categorical variable columns i.e. Fast Food Restaurant and Type. We do not need this in our analysis so will subset our data and exclude these columns.

```
calories<-calories[,3:11]

str(calories)

## 'data.frame':    126 obs. of  9 variables:
##  $ ServingSize : int  98 113 211 202 270 283 257 213 200 65 ...
##  $ Calories    : int  240 290 530 520 720 750 530 510 350 190 ...
##  $ TotalFat    : num  8 11 27 26 40 43 15 22 9 12 ...
##  $ SaturatedFat: num  3 5 10 12 15 19 10 3.5 2 2 ...
##  $ TransFat    : num  0 0.5 1 1.5 1.5 2.5 1 0 0 0 ...
##  $ Sodium      : int  480 680 960 1100 1470 1280 160 990 820 360 ...
##  $ Carbs       : num  32 33 47 41 51 42 86 55 42 12 ...
##  $ Sugars      : num  6 7 9 10 14 10 63 10 8 0 ...
##  $ Protein     : num  12 15 24 30 39 48 11 24 28 9 ...
```

Now the structure of data looks good. We have a total of 11 variables with 126 observations.

We will now check for missing values in the data.

```
any(is.na(calories))==TRUE

## [1] TRUE

which(is.na(calories))

##  [1] 592 593 594 595 596 597 598 599 600 601 602 603
```

We can see that data we have 11 missing values. This is a pretty big chunk of data to be removed from data without affecting results. So instead of removing, we will replace these values with column sum.

```
calories$TransFat[which(is.na(calories$TransFat))]<-
mean(calories$TransFat,na.rm = TRUE)
```

Checking again to ensure we do not have any missing values in our data.

```
any(is.na(calories$TransFat))==TRUE
```

```
## [1] FALSE
```

```
any(is.na(calories))==TRUE
```

```
## [1] FALSE
```

Checking the number of rows, first five rows of data frame, and summary of data.
nrow(calories)

```
head(calories, 5)
```

```
##   ServingSize Calories TotalFat SaturatedFat TransFat Sodium Carbs Sugars
## 1          98      240        8            3      0.0    480    32      6
## 2         113      290       11            5      0.5    680    33      7
## 3         211      530       27           10      1.0    960    47      9
## 4         202      520       26           12      1.5   1100    41     10
## 5         270      720       40           15      1.5   1470    51     14
##   Protein
## 1      12
## 2      15
## 3      24
## 4      30
## 5      39
```

```
summary(calories)
```

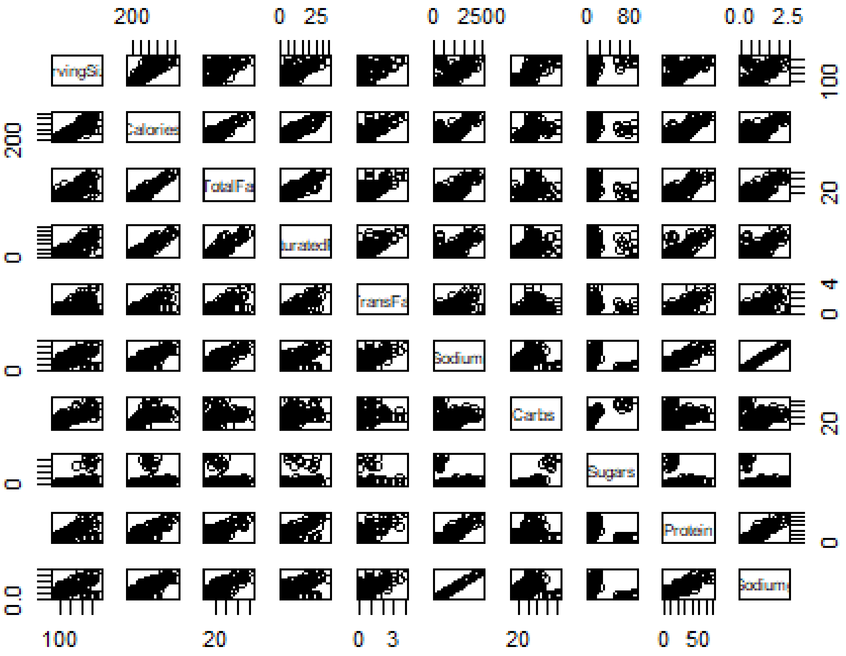
```
##   ServingSize       Calories         TotalFat      SaturatedFat
##  Min.   : 44.0   Min.   : 130.0   Min.   : 3.50   Min.   : 1.00
##  1st Qu.:126.5   1st Qu.: 330.0   1st Qu.:14.18   1st Qu.: 3.50
##  Median :217.5   Median : 515.0   Median :22.50   Median : 7.75
##  Mean   :224.3   Mean   : 532.5   Mean   :28.54   Mean   :10.15
##  3rd Qu.:315.2   3rd Qu.: 670.0   3rd Qu.:39.50   3rd Qu.:15.00
##  Max.   :467.0   Max.   :1240.0   Max.   :87.00   Max.   :35.00
##     TransFat          Sodium           Carbs            Sugars
##  Min.   :0.0000   Min.   :  50.0   Min.   :  6.00   Min.   : 0.00
##  1st Qu.:0.0000   1st Qu.: 569.2   1st Qu.: 33.00   1st Qu.: 3.00
##  Median :0.6605   Median : 930.0   Median : 42.50   Median : 7.00
##  Mean   :0.8211   Mean   : 973.7   Mean   : 44.57   Mean   :13.28
##  3rd Qu.:1.3750   3rd Qu.:1285.2   3rd Qu.: 54.00   3rd Qu.:11.00
##  Max.   :4.0000   Max.   :2460.0   Max.   :106.00   Max.   :93.00
##     Protein
##  Min.   : 2.00
##  1st Qu.:13.00
##  Median :23.00
##  Mean   :24.85
##  3rd Qu.:34.00
##  Max.   :69.00
```

We have sodim is milligrams while other variables are in grams so lets convert Sodium from milligrams to grams to make it more consistent with rest of the data

```
calories$Sodiumg=calories$Sodium/1000

head(calories)

##   ServingSize Calories TotalFat SaturatedFat TransFat Sodium Carbs Sugars
## 1          98      240        8            3      0.0    480    32      6
## 2         113      290       11            5      0.5    680    33      7
## 3         211      530       27           10      1.0    960    47      9
## 4         202      520       26           12      1.5   1100    41     10
## 5         270      720       40           15      1.5   1470    51     14
## 6         283      750       43           19      2.5   1280    42     10
##   Protein Sodiumg
## 1      12    0.48
## 2      15    0.68
## 3      24    0.96
## 4      30    1.10
## 5      39    1.47
## 6      48    1.28
```

We will now check the correlation between various variables.

```
pairs(calories)
```



As we can see that Calories have positive correlation with almost all the variables except sugars. On the other hand, some covariates are highly correlated to other covariates which suggests that there might be multicollinearity in our model.

Model Building:

```
model1<-
lm(Calories~ServingSize+TotalFat+SaturatedFat+TransFat+Sodiumg+Carbs+Sugars+P
rotein, data=calories)

summary(model1)

##
## Call:
## lm(formula = Calories ~ ServingSize + TotalFat + SaturatedFat +
##     TransFat + Sodiumg + Carbs + Sugars + Protein, data = calories)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -128.045   -7.378    0.193    8.252   91.041
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.21972    7.07287   0.314 0.754204
## ServingSize   0.11264    0.07256   1.552 0.123277
## TotalFat      8.30463    0.42666  19.464  < 2e-16 ***
## SaturatedFat -0.16102    1.00365  -0.160 0.872818
## TransFat     16.90850    4.72411   3.579 0.000503 ***
## Sodiumg      20.21584   10.16560   1.989 0.049074 *
## Carbs         3.59735    0.26051  13.809  < 2e-16 ***
## Sugars        0.03908    0.28486   0.137 0.891124
## Protein       3.02440    0.43272   6.989 1.82e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.43 on 117 degrees of freedom
## Multiple R-squared:  0.9911, Adjusted R-squared:  0.9905
## F-statistic:  1632 on 8 and 117 DF,  p-value: < 2.2e-16

vif(model1)

##  ServingSize     TotalFat SaturatedFat     TransFat      Sodiumg
##    12.866636    12.683006    13.364630     3.915993     5.928043
##        Carbs       Sugars      Protein
##     5.985313     7.509111     9.224646
```

ServingSize, SaturatedFat, Sodium_g and Sugars are not good covariates as there p values are > 0.05 thus we cannot reject the null hypothesis. Which implies in this model there is not a linear relationship between Calories and ServingSize, SaturatedFat, Sodium_g and Sugars. The VIF is also greater than 10 for some variable which is less than ideal.

In order to overcome this, we build a new model model2 by dropping ServingSize, Sugars, SaturatedFat, Sodiumg.

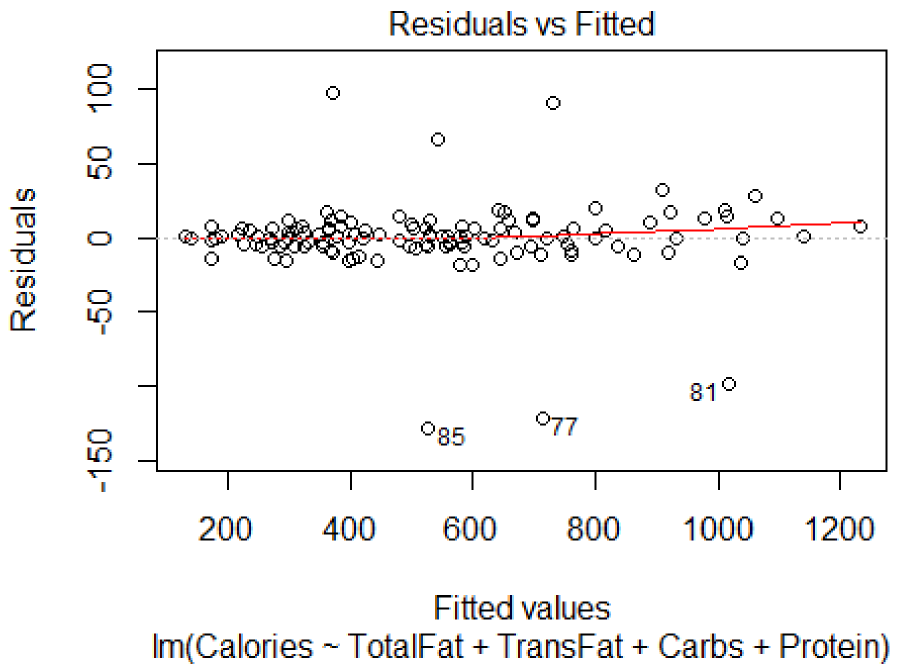
```
model2<-lm(Calories~TotalFat+TransFat+Carbs+Protein, data=calories)

summary(model2)

##
## Call:
## lm(formula = Calories ~ TotalFat + TransFat + Carbs + Protein,
##     data = calories)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -128.035   -5.701    0.413    6.769   97.178
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.2550     6.2136   0.524 0.601346
## TotalFat      8.5760     0.2360  36.342  < 2e-16 ***
## TransFat     13.6531     3.8088   3.585 0.000488 ***
## Carbs         3.9648     0.1132  35.033  < 2e-16 ***
## Protein       3.8833     0.2575  15.079  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.68 on 121 degrees of freedom
## Multiple R-squared:  0.9906, Adjusted R-squared:  0.9903
## F-statistic:  3197 on 4 and 121 DF,  p-value: < 2.2e-16

vif(model2)

## TotalFat TransFat    Carbs  Protein
## 3.801581 2.494282 1.106829 3.201417
```

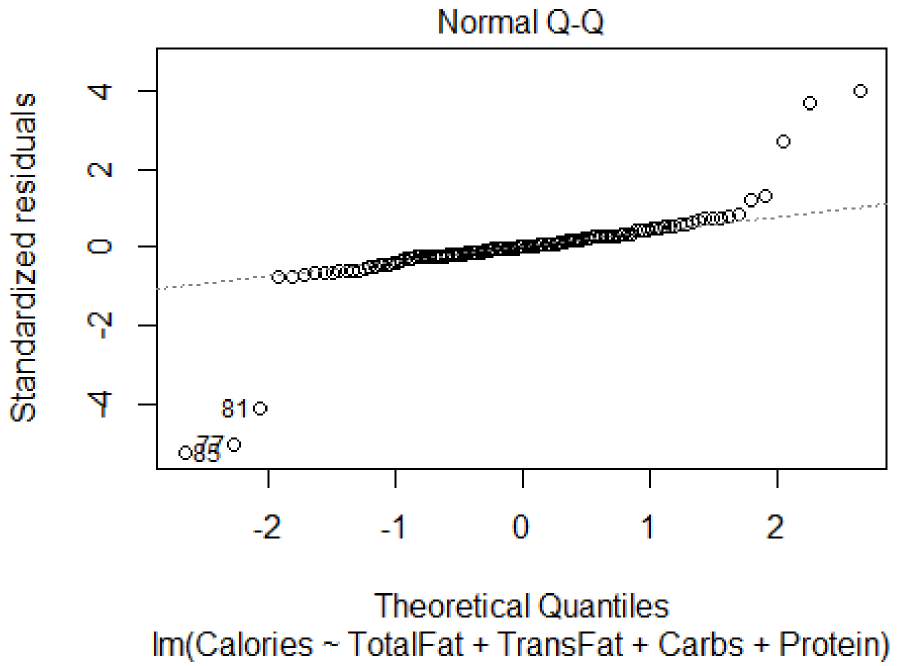
Looking at summary statistics of our new model we can see all the covariates are having a significant P and t value to express Calories linearly. VIF values are also less than 10 which is good.

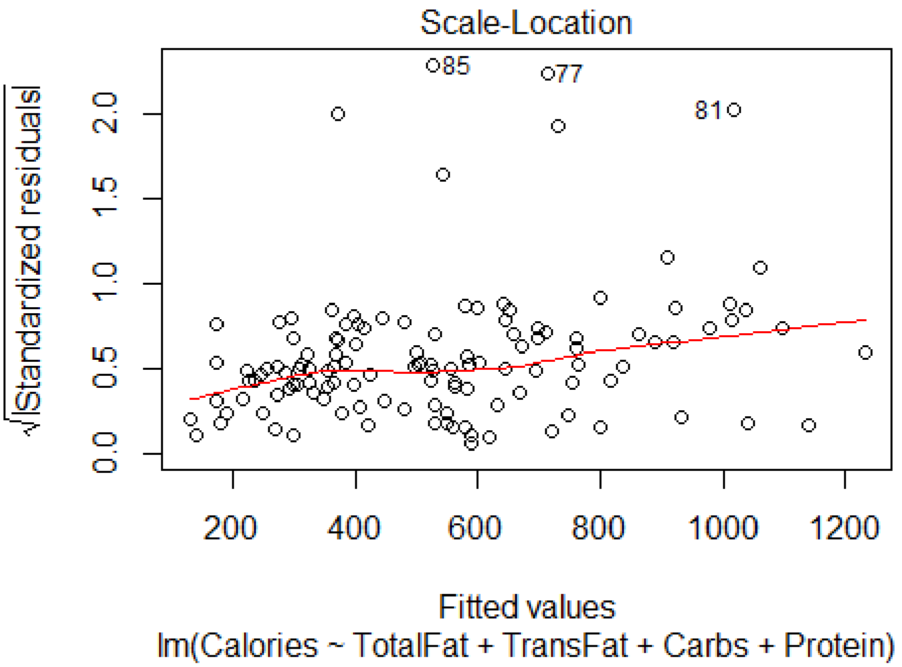
Model Adequacy Checking:

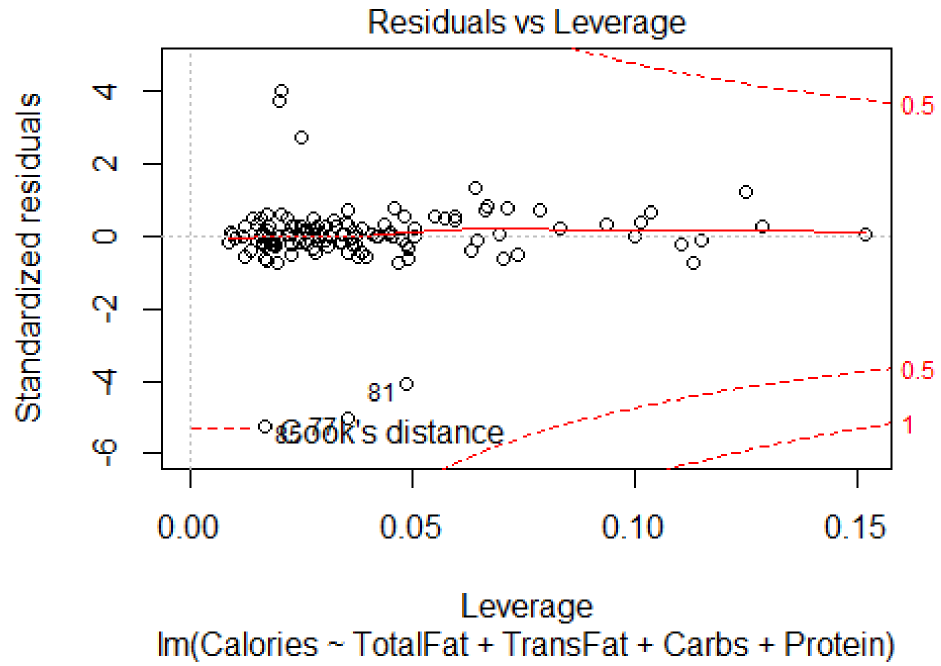
Lets check if our model meets all normality assumptions and actually valid model

```
plot(model2)
```


Residuals vs Fitted
Residuals
100
50
0
-50
-150
81
85
77
200
400
600
800
1000
1200
Fitted values
lm(Calories ~ TotalFat + TransFat + Carbs + Protein)


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
81
77
85
-2
-1
0
1
2
Theoretical Quantiles
lm(Calories ~ TotalFat + TransFat + Carbs + Protein)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
85
77
81
200
400
600
800
1000
1200
Fitted values
lm(Calories ~ TotalFat + TransFat + Carbs + Protein)


Residuals vs Leverage
Standardized residuals
4
2
0
-2
-4
-6
0.5
0.5
1
81
77
85
Cook's distance
0.00
0.05
0.10
0.15
Leverage
lm(Calories ~ TotalFat + TransFat + Carbs + Protein)

Equal variance assumption:

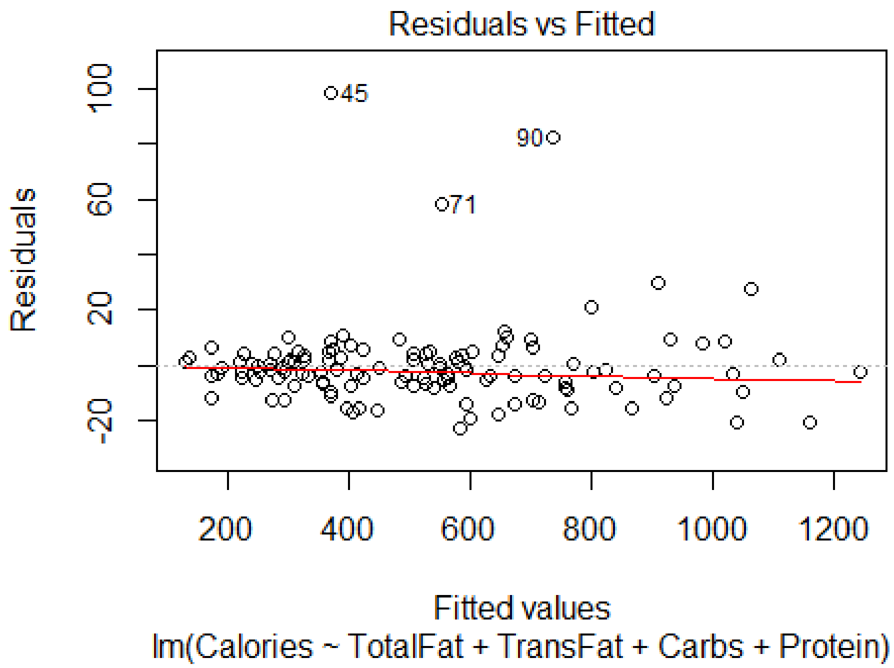
Above in the first plot we have Residual vs Fitted Values , we dont see any pattern on the red line Thus residuals are linearly distributed over fitted values and we can say approximately that variance is equal. #Normality assumption: QQ plot is fairly linear except few outliers. Standardized residuals mostly follow the fitted model line. Thus, meeting our normality assumption

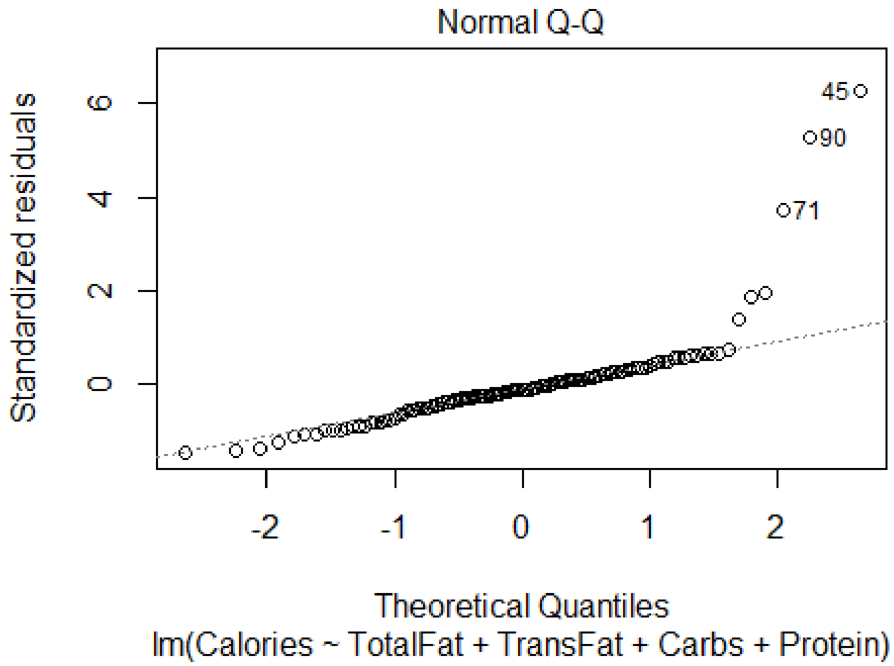
From the above graphs we can see there are few outliers in our model . To fix those let's remove them and build another model.

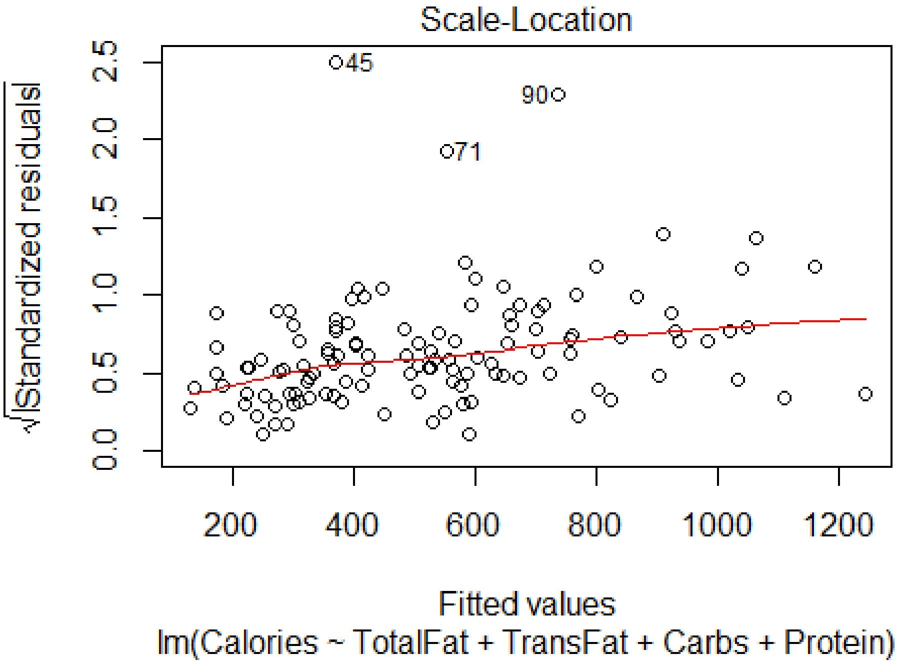
```
calories<-calories[-c(77,81,85),]

model2<-lm(Calories~TotalFat+TransFat+Carbs+Protein, data=calories)
```

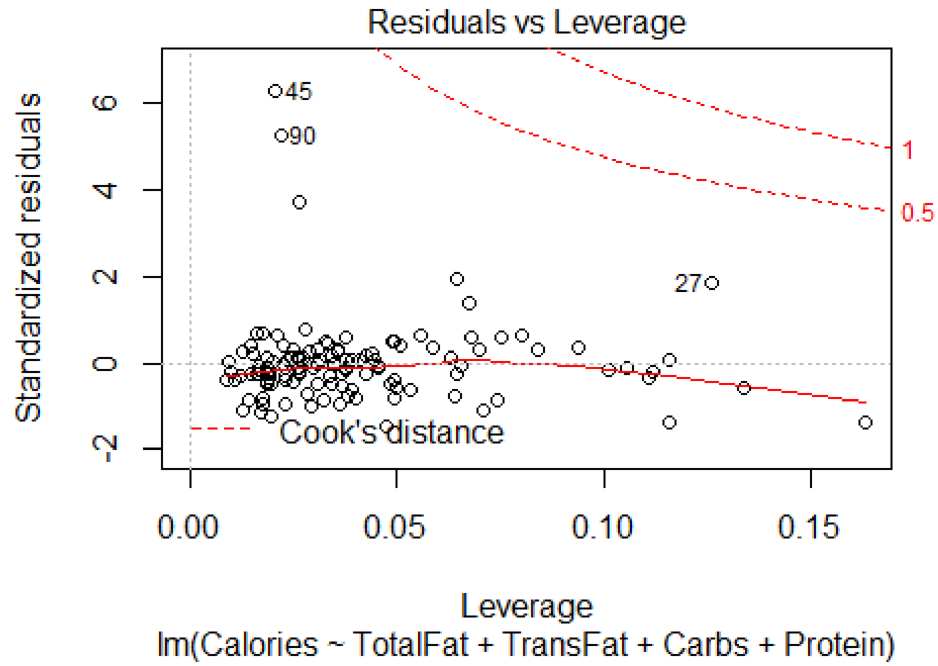
Checking model adequecy again

```
plot(model2)
```


Residuals vs Fitted
Residuals
100
60
20
-20
45
90
71
200
400
600
800
1000
1200
Fitted values
lm(Calories ~ TotalFat + TransFat + Carbs + Protein)


Normal Q-Q
Standardized residuals
6
4
2
0
45
90
71
-2
-1
0
1
2
Theoretical Quantiles
lm(Calories ~ TotalFat + TransFat + Carbs + Protein)





The model looks fairly good meeting all the assumption of Linearity between response and regressor, Normality of error distribution, Independence of errors i.e. non-correlation, and equal variance of errors

Looking at the VIF values of our model we can say that we do not have multicollinearity problem. Looking at the correlations between the variables earlier, we could see that there will be high multicollinearity but by dropping variables in model in the early phase, we got rid of multicollinearity. All of the values are below 10 so we are good.

Final Model:

$\text{Calories}=3.25+8.57\,TotalFat+13.65\,TransFat+3.96\,Carbs+3.88\,Protein$

Conclusion and Interpretation:

After building our final model, we can say that while determining calories in a product nutrients such as total fat, trans fat, carbohydrates, and protein are most significant variables that largely explain the variation in calories. Keeping all variables fixed, a unit increase in total fat in a food, increases calories by 8.57 on average. Similarly, trans fat causes 13.65 unit increase on average for every one unit increase. Lastly, carbs and proteins, cause calories to increase by 3.96 and 3.88 units on average for every one unit increase keeping all other variables fixed.

Data Source:

Kaggle.com