VARIATION OF SUGAR CONTENT IN STARBUCKS DRINKS

Hidaya Patel

4/18/2022

Introduction

Coffee is one of the most well-known beverages and has become a staple of daily human consumption across the world. According to statistics Canada, 80% of Canadians are coffee lovers. Starbucks is one of the highly favourable cafe companies across the world and in Canada. I, myself, am a huge fan of coffee and Starbucks is one of my favourites which is why I decided to choose its data.

There is so much to coffee than it being beans. Starbucks creates their coffee using fresh beans extracted and exported from various regions around the world. However, a lot of people just drink coffee for the sake of getting that caffeine rush, but very few understand the sugar intake along with it. Many people do not understand the health risks that comes with coffee consumption. Regardless, coffee is more than a refreshment. It's a memory, an anticipation, a lengthy time of inconspicuous delight woven into our existence.

Hypothesis/Question

I am going to regress sugar content against a variety of independent factors to see which nutritional characteristics had the most impact on estimating the quantity of sugar in a Starbucks drinks.

Data Collection

The dataset I chose is recorded by Starbucks and uploaded in Kaggle. The nutritional information for Starbucks' drink menu items is included in this dataset. All nutritional data for beverages is based on a 12-ounce serving size. It has:

- **242 rows**
- **8 variables**

The dependent variable here is the Sugar (in grams) against the independent variables.

Variables

The dataset has 8 variables and below I have given a description of the variables.

Beverage_category :- A particular drink category(Eg: Classic Espresso Drinks, Coffee)

Calories :- It represents the calories count in the drinks

Total Fat (g) :- It represents the Total fat count in the drinks in grams

Trans Fat (g) :- It represents the Trans fat count in the drinks in grams

Saturated Fat (g) :- It represents the Saturated fat count in the drinks in grams

Dietary Fibre (g) :- It represents the Dietry Fibres count in the drinks in grams

Sugars (g) :- It represents the Sugar count in the drinks in grams

Caffeine (mg) :- It represents the Caffeine count in the drinks in milligrams

Data Exploration

```
# Using the required libraries

library(rattle)
```

```
## Loading required package: tibble
```

```
## Loading required package: bitops
```

```
## Rattle: A free graphical interface for data science with R.
## Version 5.5.1 Copyright (c) 2006-2021 Togaware Pty Ltd.
## Type 'rattle()' to shake, rattle, and roll your data.
```

```
library(rpart)
library(readr)
library(caret)
```

```
## Loading required package: ggplot2
```

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

```
# Importing the data
SBUX_Data <- read.csv("/Users/Hidaya/Desktop/Starbucks.csv")

# Dimensions of the data
dim(SBUX_Data)
```

```
## [1] 242   8
```

```
head(SBUX_Data)
```

```
##         Beverage_category Calories Total.Fat..g. Trans.Fat..g.
## 1                  Coffee        3           0.1           0.0
## 2                  Coffee        4           0.1           0.0
## 3                  Coffee        5           0.1           0.0
## 4                  Coffee        5           0.1           0.0
## 5 Classic Espresso Drinks       70           0.1           0.1
## 6 Classic Espresso Drinks      100           3.5           2.0
##   Saturated.Fat..g. Dietary.Fibre..g. Sugars..g. Caffeine..mg.
```

```
## 1              0.0                 0          0           175
## 2              0.0                 0          0           260
## 3              0.0                 0          0           330
## 4              0.0                 0          0           410
## 5              0.0                 0          9            75
## 6              0.1                 0          9            75
```

Checking class of each column in the observations

```
class(SBUX_Data$Calories)
```

```
## [1] "integer"
```

```
class(SBUX_Data$Sugars)
```

```
## [1] "integer"
```

```
class(SBUX_Data$Total_Fat)
```

```
## [1] "NULL"
```

```
class(SBUX_Data$Trans_Fat)
```

```
## [1] "NULL"
```

```
class(SBUX_Data$Saturated_Fat)
```

```
## [1] "NULL"
```

```
class(SBUX_Data$Fibre)
```

```
## [1] "NULL"
```

```
class(SBUX_Data$Caffeine)
```

```
## [1] "character"
```

Since most of the class are Null and one of them is character let's change the classes to numeric.

```
SBUX_Data$Total_Fat<-SBUX_Data$Total.Fat..g.
SBUX_Data$Trans_Fat<-SBUX_Data$Trans.Fat..g.
SBUX_Data$Saturated_Fat<-SBUX_Data$Saturated.Fat..g.
SBUX_Data$Fibre<-SBUX_Data$Dietary.Fibre..g.
SBUX_Data$Sugars<-SBUX_Data$Sugars..g.
SBUX_Data$Caffeine<-SBUX_Data$Caffeine..mg.
```

```
SBUX_Data$Total.Fat..g.<-NULL
SBUX_Data$Trans.Fat..g.<-NULL
SBUX_Data$Saturated.Fat..g.<-NULL
SBUX_Data$Dietary.Fibre..g.<-NULL
SBUX_Data$Sugars..g.<-NULL
SBUX_Data$Caffeine..mg.<-NULL
dim(SBUX_Data)
```

```
## [1] 242   8
```

Verifying the class again

```
class(SBUX_Data$Calories)
```

```
## [1] "integer"
```

```
class(SBUX_Data$Sugars)
```

```
## [1] "integer"
```

```
class(SBUX_Data$Total_Fat)
```

```
## [1] "character"
```

```
class(SBUX_Data$Trans_Fat)
```

```
## [1] "numeric"
```

```
class(SBUX_Data$Saturated_Fat)
```

```
## [1] "numeric"
```

```
class(SBUX_Data$Fibre)
```

```
## [1] "integer"
```

```
class(SBUX_Data$Caffeine)
```

```
## [1] "character"
```

Now changing the character classes.

```
SBUX_Data$Total_Fat <- as.numeric(SBUX_Data$Total_Fat)
```

```
## Warning: NAs introduced by coercion
```

```
SBUX_Data$Caffeine <- as.numeric(SBUX_Data$Caffeine)
```

```
## Warning: NAs introduced by coercion
```

```
class(SBUX_Data$Total_Fat)
```

```
## [1] "numeric"
```

```
class(SBUX_Data$Caffeine)
```

```
## [1] "numeric"
```

So far, there was success for all typecasts except for the character classes. To notice here is that there is coercion, So checking the observation now which has NA.

```
which(is.na(SBUX_Data$Total_Fat))
```

```
## [1] 238
```

```
which(is.na(SBUX_Data$Caffeine))
```

```
##  [1] 103 104 105 106 131 132 133 134 135 136 137 138 139 140 141 142 159 168 169
## [20] 170 171 172 173
```

Now, I will change the observation from NA to 0.

```
SBUX_Data$Total_Fat[is.na(SBUX_Data$Total_Fat)] <- 0
SBUX_Data$Caffeine[is.na(SBUX_Data$Caffeine)] <- 0
```

Final Check !

```
class(SBUX_Data$Total_Fat)
```

```
## [1] "numeric"
```

```
class(SBUX_Data$Caffeine)
```

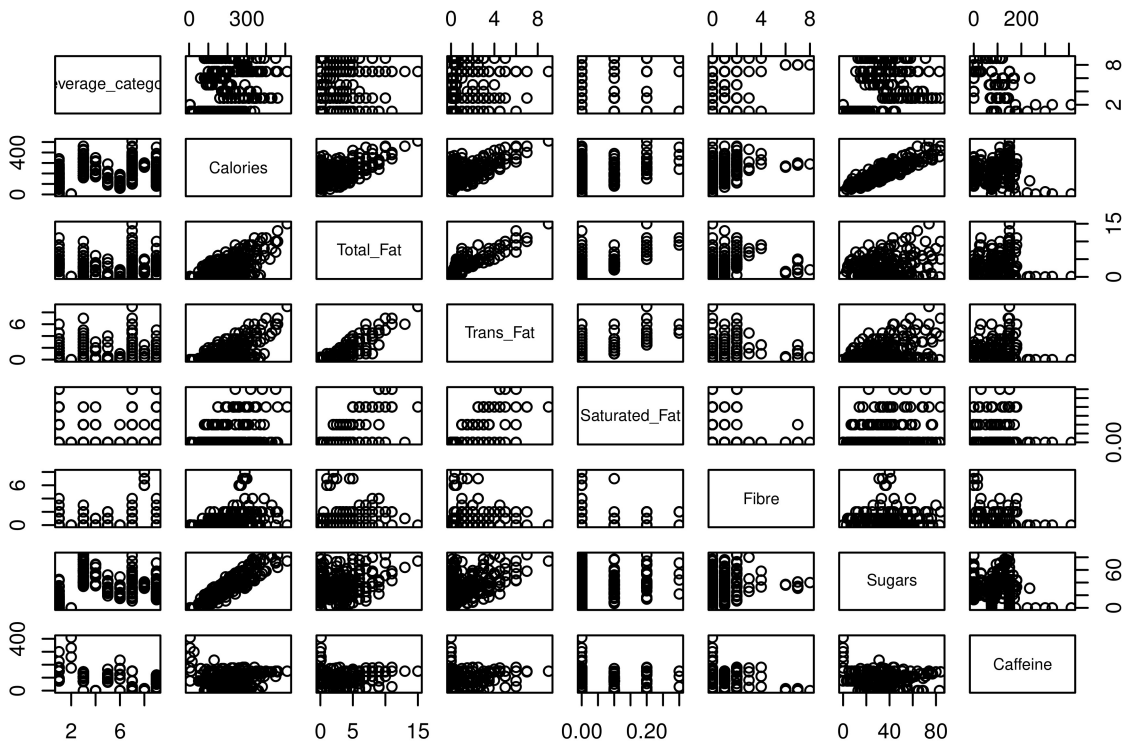
```
## [1] "numeric"
```

Wohooo! Since we have sorted all the classes now lets just check if there are any missing observations!

```
SBUX_Data <- SBUX_Data[-c(103, 104, 105,106), ]
dim(SBUX_Data)
```

```
## [1] 238   8
```

```
plot(SBUX_Data)
```



From the plot we notice that there is potential correlation between a few variables. Lets check out how strong the correlation is!

```
cor(SBUX_Data$Calories, SBUX_Data$Sugars)
```

```
## [1] 0.9049966
```

```
cor(SBUX_Data$Calories, SBUX_Data$Total_Fat)
```

```
## [1] 0.6185434
```

```
cor(SBUX_Data$Total_Fat, SBUX_Data$Sugars)
```

```
## [1] 0.3047575
```

```
cor(SBUX_Data$Total_Fat, SBUX_Data$Trans_Fat)
```

```
## [1] 0.8897008
```

```r
cor(SBUX_Data$Calories, SBUX_Data$Fibre)
```

```
## [1] 0.3790694
```

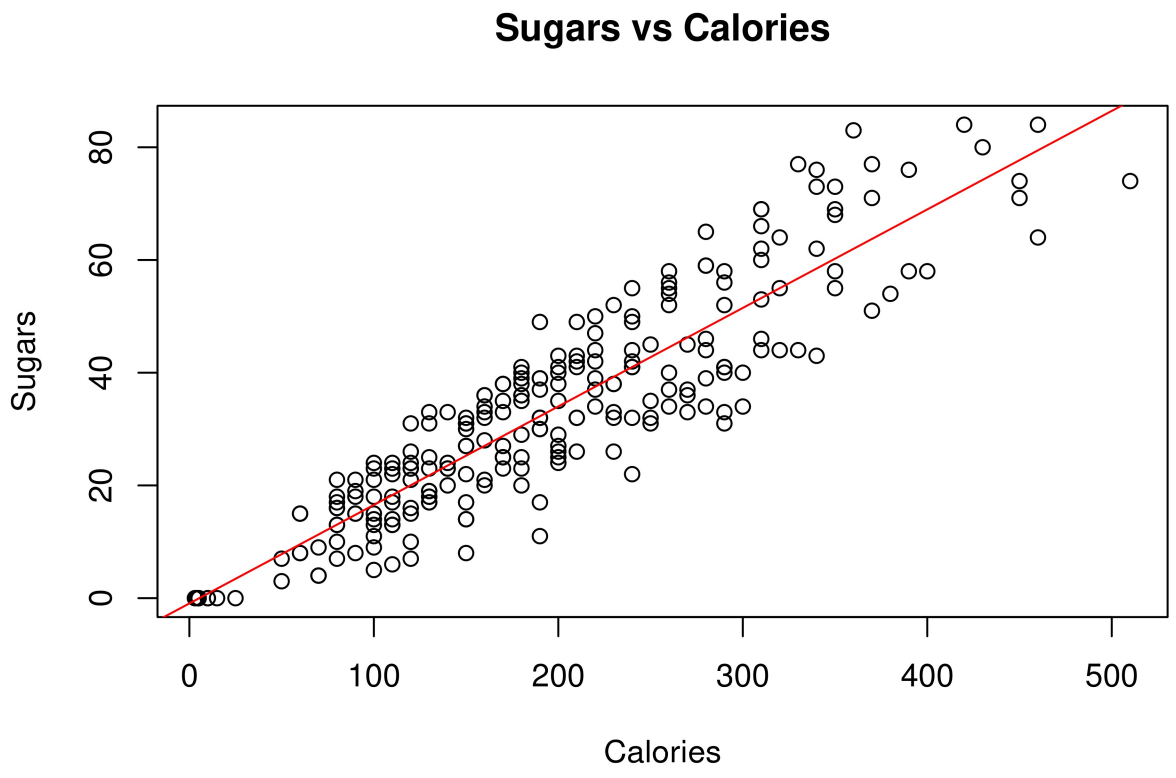
```r
cor(SBUX_Data$Trans_Fat, SBUX_Data$Caffeine)
```

```
## [1] 0.1744247
```

There is strong correlation between sugar and calories, total fat and trans fat. Now lets plot the correlation.

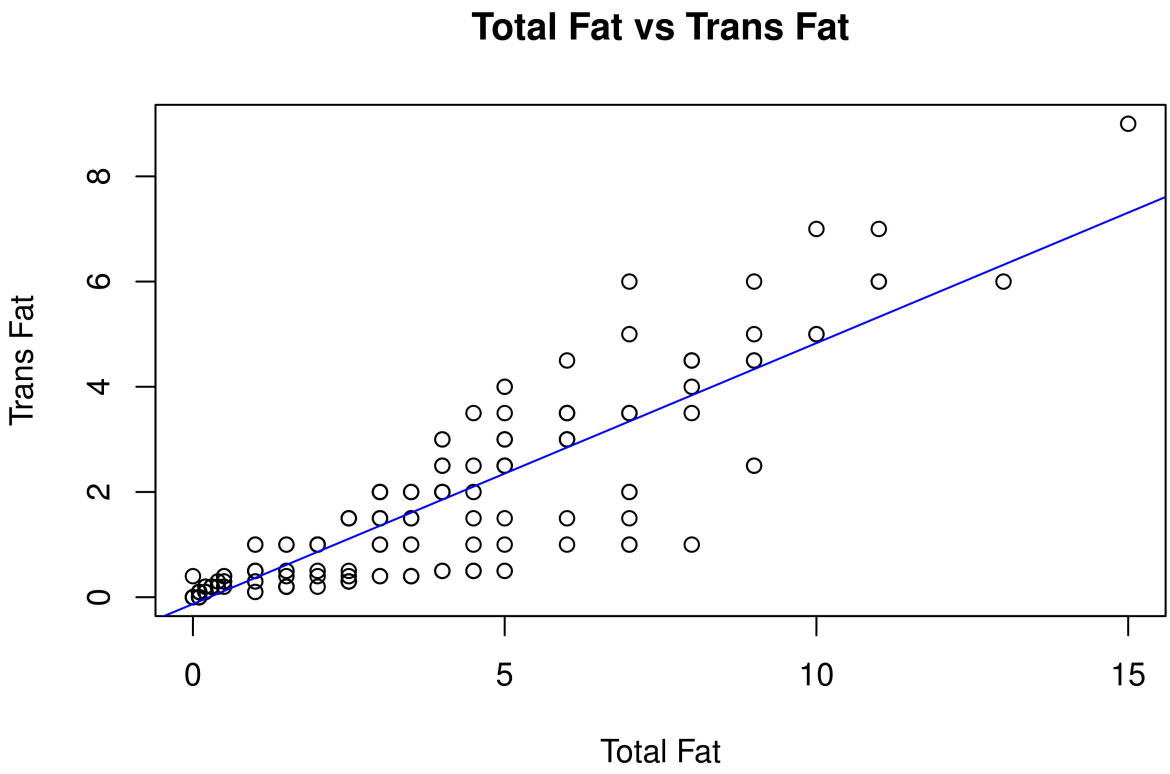
Sugars vs Calories

```r
plot(SBUX_Data$Calories,SBUX_Data$Sugars, xlab = "Calories", ylab = "Sugars", main = "Sugars vs Calories
abline(lm(Sugars~Calories, SBUX_Data), col = "red")
```



Trans fat vs Total fat

```r
plot(SBUX_Data$Total_Fat,SBUX_Data$Trans_Fat, xlab = "Total Fat", ylab = "Trans Fat", main = "Total Fat
abline(lm(Trans_Fat~Total_Fat, SBUX_Data), col = "blue")
```



CHECK LINEAR MODEL

Now, we are looking to see what model best explains the variations in sugar content for items on the Starbucks Drink Menu.

So the assumptions will be,

1. Check for constant variance of error
2. Normality
3. Independence of errors

Let's check if our model satisfies these assumptions!

```
# Assumption 1, first we will check for interaction
L1 <- lm(Sugars~Caffeine+Total_Fat+Fibre+Calories+Saturated_Fat+Trans_Fat, SBUX_Data)
summary(L1)
```

```
##
## Call:
## lm(formula = Sugars ~ Caffeine + Total_Fat + Fibre + Calories +
##     Saturated_Fat + Trans_Fat, data = SBUX_Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -12.2574  -2.6223   0.2108   2.8195  10.3564
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.047029   0.724979  -2.824  0.00516 **
## Caffeine      -0.010740   0.004190  -2.563  0.01100 *
## Total_Fat     -2.893982   0.208603 -13.873  < 2e-16 ***
## Fibre         -2.556878   0.216549 -11.807  < 2e-16 ***
## Calories       0.234364   0.003778  62.027  < 2e-16 ***
## Saturated_Fat -4.485272   5.466108  -0.821  0.41274
## Trans_Fat      0.770558   0.417900   1.844  0.06648 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.078 on 231 degrees of freedom
## Multiple R-squared:  0.957,  Adjusted R-squared:  0.9559
## F-statistic: 857.6 on 6 and 231 DF,  p-value: < 2.2e-16
```

Here we see that saturated fat, Caffeine and trans fat are not quite good predictors, so lets leave them out.

```
L2 <- lm(Sugars~Total_Fat+Fibre+Calories, SBUX_Data)
summary(L2)
```

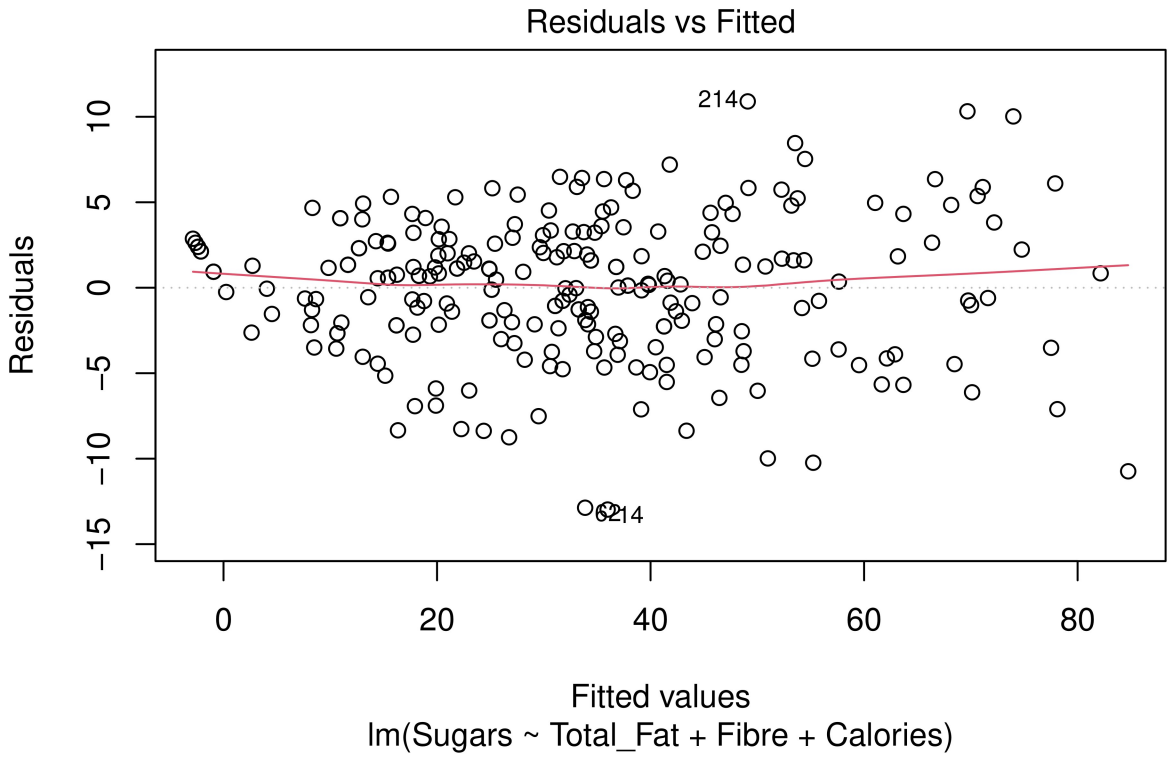
```
##
## Call:
## lm(formula = Sugars ~ Total_Fat + Fibre + Calories, data = SBUX_Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -12.9759  -2.5025   0.3833   2.6280  10.8994
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.309724   0.596727  -5.546 7.86e-08 ***
## Total_Fat   -2.683616   0.115726 -23.189  < 2e-16 ***
## Fibre       -2.541946   0.199387 -12.749  < 2e-16 ***
## Calories     0.237406   0.003551  66.853  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.13 on 234 degrees of freedom
## Multiple R-squared:  0.9554, Adjusted R-squared:  0.9548
## F-statistic:  1669 on 3 and 234 DF,  p-value: < 2.2e-16
```

Although our

$$R^2$$

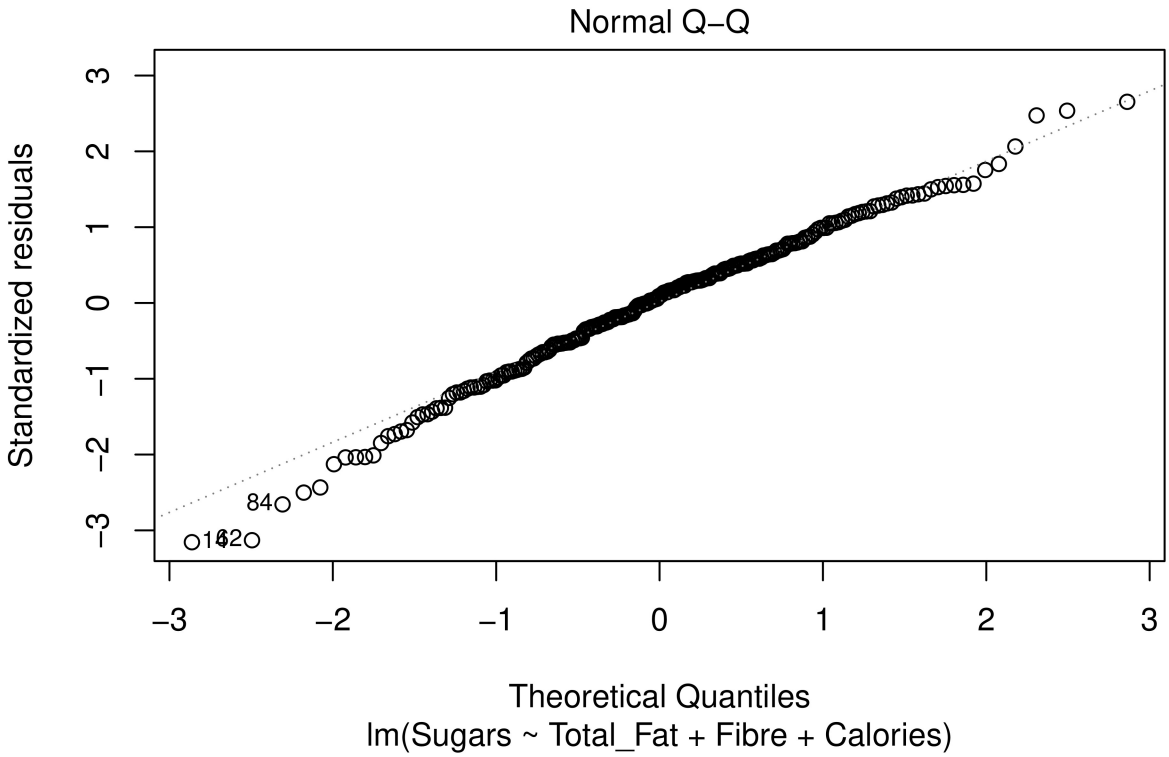
value reduced very little but we know that saturated fat and trans fat were not quite useful in our model! Now looking at the diagnostics to make sure our assumptions are met.

```
plot(L2, which = 1)
```



Our first assumption of constant variance is met as there is no definite pattern and the variance looks constant (although it is never constant in real life!).

```
plot(L2, which = 2)
```



Our second assumption of normality is also met as almost all the values fall on the line. There are a few outliers but that can be ignored for the assumption.

```
plot(L2, which = 3)
```

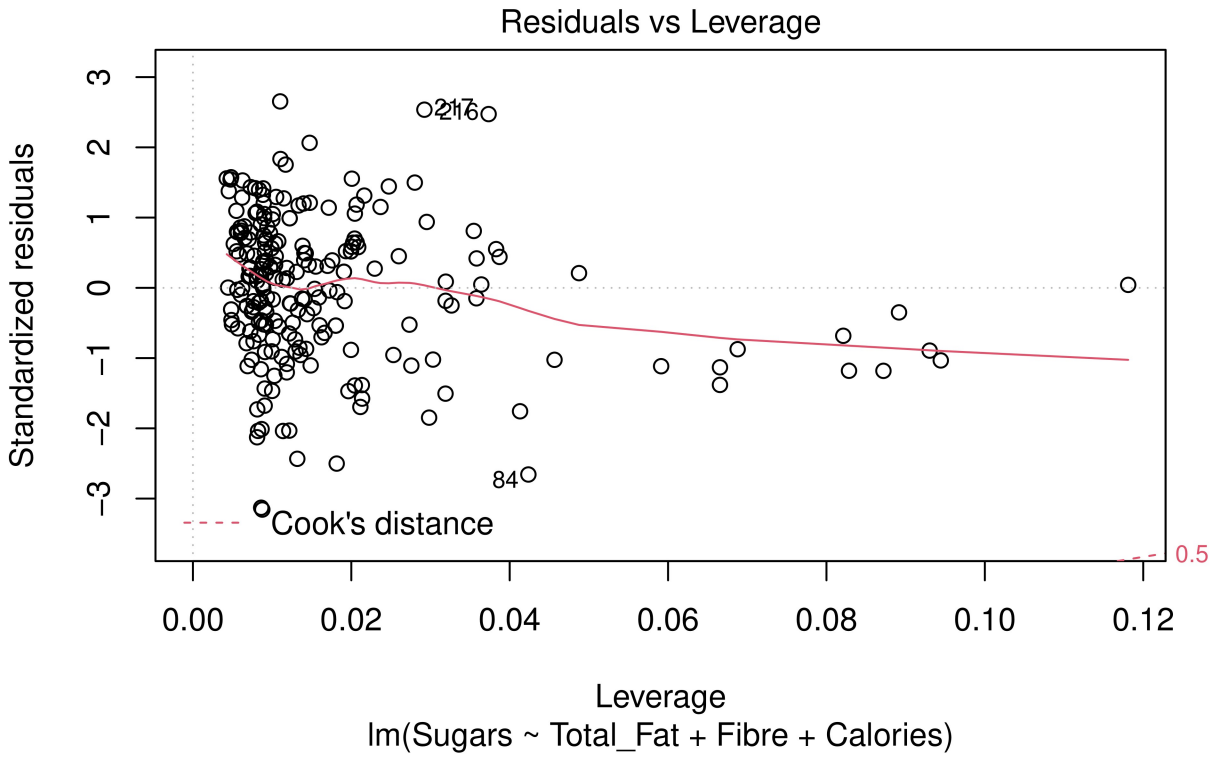
To note here that observation 62, 214 and 14 are the outliers! Let's check why are they the outliers!

```
print(SBUX_Data[c(62,214,14),])
```

```
##                       Beverage_category Calories Total_Fat Trans_Fat
## 62              Classic Espresso Drinks      160       0.3       0.2
## 218 FrappuccinoÃ® Light Blended Coffee       90       0.1       0.0
## 14              Classic Espresso Drinks      170       0.4       0.3
##     Saturated_Fat Fibre Sugars Caffeine
## 62              0     0     21      150
## 218             0     0     19       70
## 14              0     0     23      150
```

As we saw since these observations have zero fbre and high calories which is why they were the outliers. Since these outliers are not bad I will not remove them from the observation!

```
plot(L2, which = 5)
```



There is no point with high leverage but there are points with low leverage which are 84, 216, 217. There is no chance of an influential point right here! Saved us some extra work!

Checking for possible interactions

Although our model looks pretty good now but is there anyway we can improvise it! Interactions might save us but who knows!

Looking at the plots earlier we can deduce that there is a possibility for a lot of interactions, but will they be quite useful?

```
# Possible interactions between independent variables Fibre, Total fat and Calories

inter <- lm(Sugars~Calories*Total_Fat+Fibre*Calories+Fibre*Total_Fat, SBUX_Data)
summary(inter)
```

```
##
## Call:
## lm(formula = Sugars ~ Calories * Total_Fat + Fibre * Calories +
##     Fibre * Total_Fat, data = SBUX_Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.3333  -2.3496   0.2438   2.5012  10.3613
##
```

```
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -3.0530548  0.7620215  -4.007 8.31e-05 ***
## Calories            0.2386271  0.0043529  54.820  < 2e-16 ***
## Total_Fat          -2.9754720  0.2844736 -10.460  < 2e-16 ***
## Fibre              -2.9774693  1.0591596  -2.811  0.00536 **
## Calories:Total_Fat  0.0001463  0.0008403   0.174  0.86190
## Calories:Fibre     -0.0013136  0.0039823  -0.330  0.74180
## Total_Fat:Fibre     0.2556159  0.0822717   3.107  0.00213 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.06 on 231 degrees of freedom
## Multiple R-squared:  0.9574, Adjusted R-squared:  0.9563
## F-statistic: 865.4 on 6 and 231 DF,  p-value: < 2.2e-16
```

There is a slight interaction between total fat and Fibre. But I assume it is not very useful for the model as there is no large dependency on each other! Lets plot the interaction.

```
library(sjmisc)
```

```
##
## Attaching package: 'sjmisc'

## The following object is masked from 'package:tibble':
##
##     add_case
```

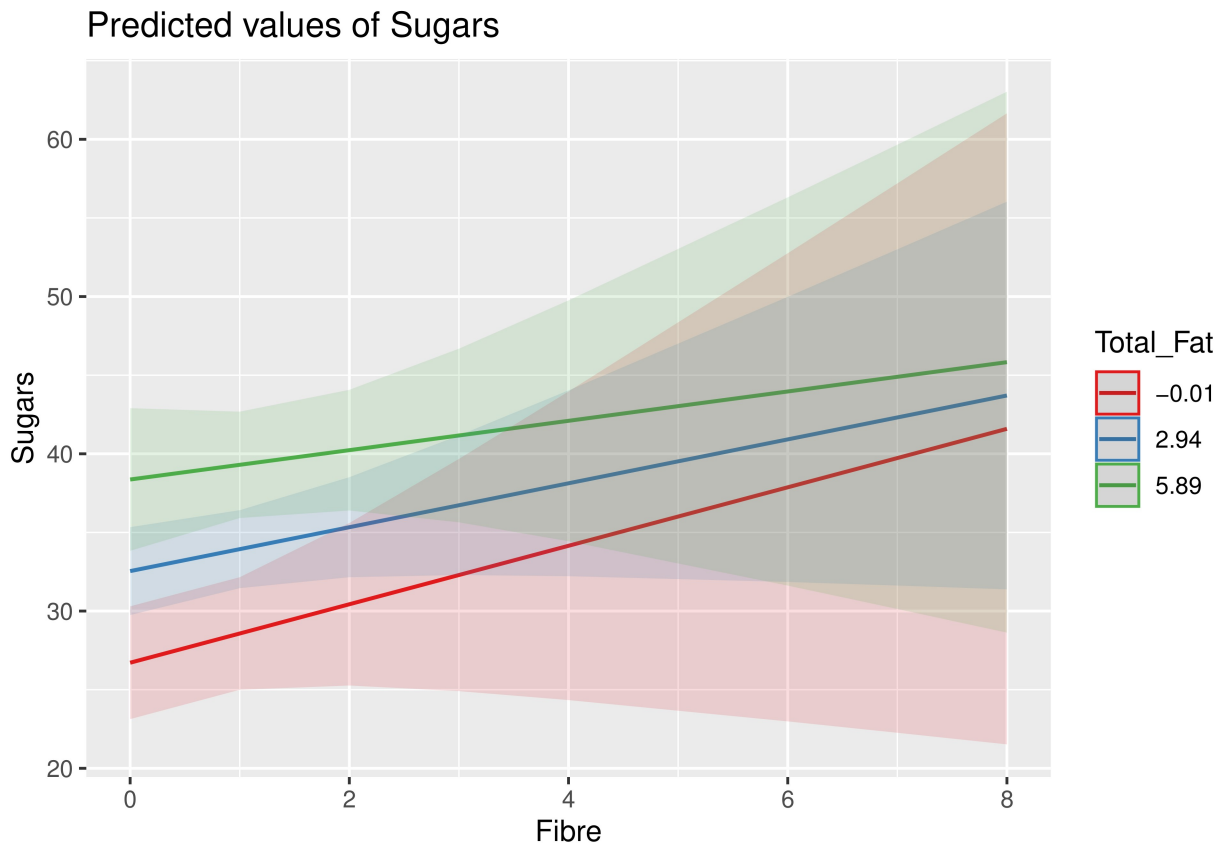
```
library(sjPlot)
```

```
## Registered S3 methods overwritten by 'parameters':
##   method                          from
##   format.parameters_distribution datawizard
##   predict.kmeans                 rattle

## #refugeeswelcome
```

```
fit <- lm(Sugars ~ Fibre * Total_Fat, data = SBUX_Data)

plot_model(fit, type = "pred", terms = c("Fibre", "Total_Fat"))
```



Here we can note that the interaction does exist but it is not very useful as the dependency of Fibre on Total_Fat is not quite large.

So I will not be including any interactions in the model fit. In addition there is no requirement for transformation as there is no issue with the dataset and our assumptions of linearity have been satisfied!

Then finally we have our model fit which has 3 predictor variables Total Fat, Fibre and Calories. But one last thing, in model 2 the intercept is -3.3 which does not make sense as sugar variation cannot be negative! So lets look at the model without the intercept.

```
L2_intercept <- lm(Sugars~Fibre+Calories+Total_Fat-1, SBUX_Data)
summary(L2_intercept)
```

```
##
## Call:
## lm(formula = Sugars ~ Fibre + Calories + Total_Fat - 1, data = SBUX_Data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.8094  -3.3362  -0.8819   1.8568  12.6292
##
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## Fibre     -2.448666   0.210883  -11.61   <2e-16 ***
## Calories   0.222444   0.002451   90.74   <2e-16 ***
## Total_Fat -2.605619   0.121926  -21.37   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.383 on 235 degrees of freedom
## Multiple R-squared:  0.9873, Adjusted R-squared:  0.9872
## F-statistic:  6111 on 3 and 235 DF,  p-value: < 2.2e-16
```

Now this looks like a good model! Therefore the model 2 is the best fit for the data to represent the variation of Sugar content in Starbucks drinks!

$$Sugars = 0.22 * \text{Calories} - 2.45 * \text{Fibre} - 2.6 * \text{Total Fat}$$