Multiple Regression Project

Kabin Devkota, Nishan Khanal and Udita Bista

4/15/2025

Introduction

This analysis focuses on a data set from Kaggle that describes the nutrition facts for McDonald's Menu. This dataset provides a nutrition analysis of every menu item on the US McDonald's menu, including breakfast, beef burgers, chicken and fish sandwiches, fries, salads, soda, coffee and tea, milkshakes, and desserts. The data for this analysis consists of response variable $y = Calories$ and following explanatory variables:

$Category$

$item$

$ServingSize$

$TotalFat$

$TotalFat(\%DailyValue)$

$SaturatedFat$

$SaturatedFat(\%DailyValue)$

$TransFat$

$Cholesterol$

$Cholesterol(\%DailyValue)$

$Sodium$

$Sodium(\%DailyValue)$

$Carbohydrates$

$Carbohydrates(\%DailyValue)$

$DietaryFiber$

$DietaryFiber(\%DailyValue)$

$Sugars$

$Protein$

$VitaminA(\%DailyValue)$

$VitaminC(\%DailyValue)$

$Calcium(\%DailyValue)$

$Iron(\%DailyValue)$

```
# library(ggplot2)
library(lmtest, pos=4)
```

```
## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(corrplot)
```

```
## corrplot 0.95 loaded
```

```
# reading the data from the csv file which has some values containing comma enclosed by double quotes
mcdonalds = read.csv("Mcdonalds_menu.csv",header=TRUE,quote="\"",sep=",")
head(mcdonalds)
```

```
##    Category                            Item   Serving.Size Calories
## 1 Breakfast                     Egg McMuffin 4.8 oz (136 g)      300
## 2 Breakfast                Egg White Delight 4.8 oz (135 g)      250
## 3 Breakfast                 Sausage McMuffin 3.9 oz (111 g)      370
## 4 Breakfast        Sausage McMuffin with Egg 5.7 oz (161 g)      450
## 5 Breakfast Sausage McMuffin with Egg Whites 5.7 oz (161 g)      400
## 6 Breakfast             Steak & Egg McMuffin 6.5 oz (185 g)      430
##   Calories.from.Fat Total.Fat Total.Fat....Daily.Value. Saturated.Fat
## 1               120        13                        20             5
## 2                70         8                        12             3
## 3               200        23                        35             8
## 4               250        28                        43            10
## 5               210        23                        35             8
## 6               210        23                        36             9
##   Saturated.Fat....Daily.Value. Trans.Fat Cholesterol
## 1                            25         0         260
## 2                            15         0          25
## 3                            42         0          45
## 4                            52         0         285
## 5                            42         0          50
## 6                            46         1         300
##   Cholesterol....Daily.Value. Sodium Sodium....Daily.Value. Carbohydrates
## 1                          87    750                    31            31
## 2                           8    770                    32            30
## 3                          15    780                    33            29
## 4                          95    860                    36            30
## 5                          16    880                    37            30
## 6                         100    960                    40            31
##   Carbohydrates....Daily.Value. Dietary.Fiber Dietary.Fiber....Daily.Value.
## 1                            10             4                            17
## 2                            10             4                            17
## 3                            10             4                            17
## 4                            10             4                            17
## 5                            10             4                            17
## 6                            10             4                            18
##   Sugars Protein Vitamin.A....Daily.Value. Vitamin.C....Daily.Value.
## 1      3      17                        10                         0
## 2      3      18                         6                         0
## 3      2      14                         8                         0
## 4      2      21                        15                         0
```

```
## 5      2      21                          6                          0
## 6      3      26                         15                          2
##   Calcium....Daily.Value. Iron....Daily.Value.
## 1                      25                    15
## 2                      25                     8
## 3                      25                    10
## 4                      30                    15
## 5                      25                    10
## 6                      30                    20
```

```r
# getting the column names
colnames(mcdonalds)
```

```
##  [1] "Category"                      "Item"
##  [3] "Serving.Size"                  "Calories"
##  [5] "Calories.from.Fat"             "Total.Fat"
##  [7] "Total.Fat....Daily.Value."     "Saturated.Fat"
##  [9] "Saturated.Fat....Daily.Value." "Trans.Fat"
## [11] "Cholesterol"                   "Cholesterol....Daily.Value."
## [13] "Sodium"                        "Sodium....Daily.Value."
## [15] "Carbohydrates"                 "Carbohydrates....Daily.Value."
## [17] "Dietary.Fiber"                 "Dietary.Fiber....Daily.Value."
## [19] "Sugars"                        "Protein"
## [21] "Vitamin.A....Daily.Value."     "Vitamin.C....Daily.Value."
## [23] "Calcium....Daily.Value."       "Iron....Daily.Value."
```

```r
# cleaning column names: replace spaces, %, parentheses, etc.
colnames(mcdonalds) <- make.names(colnames(mcdonalds))
```

Out of these 22 explanatory variables, some of the variables like "category", Total.Fat. . . .Daily.Value.","Cholesterol. . . .Daily.Valu etc are redundant.Therefore, we are dropping them from the analysis to avoid multicollinearity and improve model efficiency.

```r
# dropping the columns that is irrelevant or redundant for the analysis
cols_to_drop <- c(
  "Category",
  "Calories.from.Fat",
  "Total.Fat....Daily.Value.",
  "Saturated.Fat....Daily.Value.",
  "Cholesterol....Daily.Value.",
  "Sodium....Daily.Value.",
  "Carbohydrates....Daily.Value.",
  "Dietary.Fiber....Daily.Value."
)
mcdonalds <- mcdonalds[, !(names(mcdonalds) %in% cols_to_drop)]
```

Now, we are converting the % Daily Value columns to absolute values. The daily values are based on a 2000 calorie diet, and the conversion is done using the following formula:

$$\text{Absolute Value} = \left(\frac{\text{Percentage Daily Value}}{100}\right) \times \text{Daily Value}$$

where the daily values are as follows:

```r
# Daily values (units must match dataset)
daily_values <- c(
  "Vitamin.A" = 900,    # mcg RAE
  "Vitamin.C" = 90,     # mg
```

```
  "Calcium" = 1300,      # mg
  "Iron" = 18            # mg
)

# Map of columns to their associated nutrients
conversion_map <- list(
  "Vitamin.A....Daily.Value." = "Vitamin.A",
  "Vitamin.C....Daily.Value." = "Vitamin.C",
  "Calcium....Daily.Value." = "Calcium",
  "Iron....Daily.Value." = "Iron"
)

# For each %DV column, calculate the absolute value and overwrite the same column with absolute value
for (dv_col in names(conversion_map)) {
  nutrient <- conversion_map[[dv_col]]

  if (dv_col %in% names(mcdonalds)) {
    # Create a new column name or overwrite the existing one
    new_col <- nutrient  # Replace the DV column with nutrient name only
    mcdonalds[[new_col]] <- (mcdonalds[[dv_col]] / 100) * daily_values[[nutrient]]
  }
}
```

The values in the serving size column contain both numbers and units (e.g., "15 oz"). In this step, we are extracting only the numeric part and discarding the unit, so "15 oz" becomes just 15.

```
# Extract numeric part before "oz" or "fl oz" from the Serving Size column
mcdonalds$Serving.Size.Oz <- as.numeric(sub("([0-9.]+)\\s*(fl\\s*)?oz.*", "\\1", mcdonalds$Serving.Size
```

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

```
head(mcdonalds)
```

```
##                              Item   Serving.Size Calories Total.Fat
## 1                    Egg McMuffin 4.8 oz (136 g)      300        13
## 2                Egg White Delight 4.8 oz (135 g)      250         8
## 3                 Sausage McMuffin 3.9 oz (111 g)      370        23
## 4        Sausage McMuffin with Egg 5.7 oz (161 g)      450        28
## 5 Sausage McMuffin with Egg Whites 5.7 oz (161 g)      400        23
## 6             Steak & Egg McMuffin 6.5 oz (185 g)      430        23
##   Saturated.Fat Trans.Fat Cholesterol Sodium Carbohydrates Dietary.Fiber Sugars
## 1             5         0         260    750            31             4      3
## 2             3         0          25    770            30             4      3
## 3             8         0          45    780            29             4      2
## 4            10         0         285    860            30             4      2
## 5             8         0          50    880            30             4      2
## 6             9         1         300    960            31             4      3
##   Protein Vitamin.A....Daily.Value. Vitamin.C....Daily.Value.
## 1      17                        10                         0
## 2      18                         6                         0
## 3      14                         8                         0
## 4      21                        15                         0
## 5      21                         6                         0
## 6      26                        15                         2
##   Calcium....Daily.Value. Iron....Daily.Value. Vitamin.A Vitamin.C Calcium Iron
```

```
## 1                      25                   15        90       0.0     325 2.70
## 2                      25                    8        54       0.0     325 1.44
## 3                      25                   10        72       0.0     325 1.80
## 4                      30                   15       135       0.0     390 2.70
## 5                      25                   10        54       0.0     325 1.80
## 6                      30                   20       135       1.8     390 3.60
##   Serving.Size.Oz
## 1             4.8
## 2             4.8
## 3             3.9
## 4             5.7
## 5             5.7
## 6             6.5
```

```
# dropping the data points with null values that would be a hindrance for the analysis
mcdonalds <- na.omit(mcdonalds)

head(mcdonalds)
```

```
##                             Item   Serving.Size Calories Total.Fat
## 1                    Egg McMuffin 4.8 oz (136 g)      300        13
## 2                Egg White Delight 4.8 oz (135 g)      250         8
## 3                 Sausage McMuffin 3.9 oz (111 g)      370        23
## 4        Sausage McMuffin with Egg 5.7 oz (161 g)      450        28
## 5 Sausage McMuffin with Egg Whites 5.7 oz (161 g)      400        23
## 6             Steak & Egg McMuffin 6.5 oz (185 g)      430        23
##   Saturated.Fat Trans.Fat Cholesterol Sodium Carbohydrates Dietary.Fiber Sugars
## 1             5         0         260    750            31             4      3
## 2             3         0          25    770            30             4      3
## 3             8         0          45    780            29             4      2
## 4            10         0         285    860            30             4      2
## 5             8         0          50    880            30             4      2
## 6             9         1         300    960            31             4      3
##   Protein Vitamin.A....Daily.Value. Vitamin.C....Daily.Value.
## 1      17                        10                         0
## 2      18                         6                         0
## 3      14                         8                         0
## 4      21                        15                         0
## 5      21                         6                         0
## 6      26                        15                         2
##   Calcium....Daily.Value. Iron....Daily.Value. Vitamin.A Vitamin.C Calcium Iron
## 1                      25                   15        90       0.0     325 2.70
## 2                      25                    8        54       0.0     325 1.44
## 3                      25                   10        72       0.0     325 1.80
## 4                      30                   15       135       0.0     390 2.70
## 5                      25                   10        54       0.0     325 1.80
## 6                      30                   20       135       1.8     390 3.60
##   Serving.Size.Oz
## 1             4.8
## 2             4.8
## 3             3.9
## 4             5.7
## 5             5.7
## 6             6.5
```

```r
# Identify all columns to normalize (exclude Item and Serving Size Oz)
cols_to_normalize <- setdiff(names(mcdonalds), c("Item", "Serving.Size", "Serving.Size.Oz"))

# Convert values to per oz
mcdonalds[cols_to_normalize] <- lapply(mcdonalds[cols_to_normalize], function(col) col / mcdonalds$Servi

# dropping the remaining columns containing daily value percentages and serving size
cols_to_drop <- c(
  "Vitamin.A....Daily.Value.",
  "Vitamin.C....Daily.Value.",
  "Calcium....Daily.Value.",
  "Iron....Daily.Value.",
  "Serving.Size"
)
mcdonalds <- mcdonalds[, !(names(mcdonalds) %in% cols_to_drop)]


# saving all the column names except "Item"
col_names <- setdiff(names(mcdonalds), "Item")
```

```r
col_names
```

```
##  [1] "Calories"        "Total.Fat"       "Saturated.Fat"   "Trans.Fat"
##  [5] "Cholesterol"     "Sodium"          "Carbohydrates"   "Dietary.Fiber"
##  [9] "Sugars"          "Protein"         "Vitamin.A"       "Vitamin.C"
## [13] "Calcium"         "Iron"            "Serving.Size.Oz"
```

After the completion of the data preprocessing part, our dataset has been narrowed down to one response variable *calories* and fourteen explanatory variables:

1. x1 = $TotalFat$
2. x2 = $SaturatedFat$
3. x3 = $TransFat$
4. x4 = $Cholestrol$
5. x5 = $Sodium$
6. x6 = $Carbohydrates$
7. x7 = $DietaryFiber$
8. x8 = $Sugars$
9. x9 = $Protein$
10. x10 = $VitaminA$
11. x11 = $VitaminC$
12. x12 = $Calcium$
13. x13 = $Iron$
14. x14 = $Serving.size.oz$

```r
# creating new names: y for Calories, x1, x2, ... for the rest
new_names <- c("y", paste0("x", seq_along(col_names[-which(col_names == "Calories")])))

# creating a named vector to rename the columns
```

```r
name_map <- setNames(new_names, c("Calories", col_names[col_names != "Calories"]))

# renaming the columns
names(mcdonalds)[names(mcdonalds) %in% names(name_map)] <- name_map[names(mcdonalds)[names(mcdonalds) %

# saving all the column names except "Item"
col_names <- setdiff(names(mcdonalds), "Item")
```

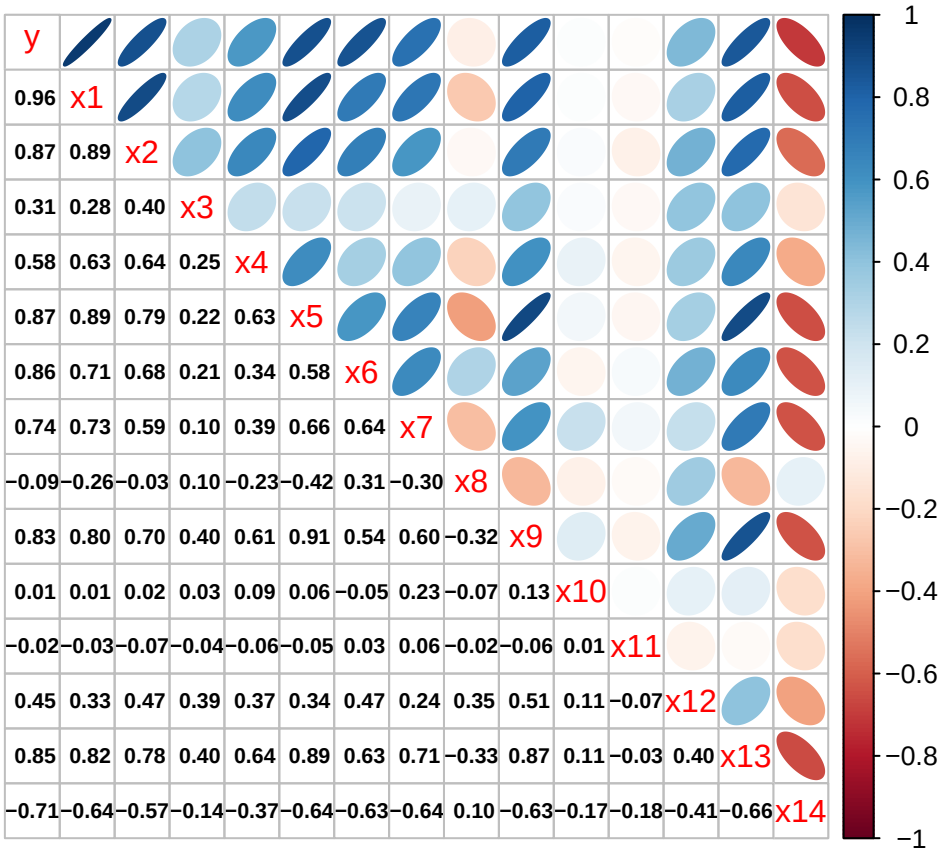
Correlation Coefficients

Since we are looking at linear relationships between the outcome variable (calories) with each explanatory variable, it may be of interest to determine the correlation coefficients between the outcome variable with each explanatory variable.

```r
corr_matrix <- cor(mcdonalds[,col_names], use="everything")
round(corr_matrix, 3)
```

```
##          y     x1     x2     x3     x4     x5     x6     x7     x8     x9
## y    1.000  0.960  0.873  0.311  0.579  0.872  0.862  0.745 -0.088  0.826
## x1   0.960  1.000  0.894  0.285  0.628  0.890  0.709  0.726 -0.263  0.802
## x2   0.873  0.894  1.000  0.402  0.640  0.792  0.681  0.588 -0.032  0.705
## x3   0.311  0.285  0.402  1.000  0.250  0.225  0.212  0.096  0.102  0.396
## x4   0.579  0.628  0.640  0.250  1.000  0.626  0.336  0.394 -0.229  0.607
## x5   0.872  0.890  0.792  0.225  0.626  1.000  0.581  0.664 -0.416  0.906
## x6   0.862  0.709  0.681  0.212  0.336  0.581  1.000  0.638  0.310  0.538
## x7   0.745  0.726  0.588  0.096  0.394  0.664  0.638  1.000 -0.303  0.599
## x8  -0.088 -0.263 -0.032  0.102 -0.229 -0.416  0.310 -0.303  1.000 -0.323
## x9   0.826  0.802  0.705  0.396  0.607  0.906  0.538  0.599 -0.323  1.000
## x10  0.012  0.010  0.022  0.027  0.092  0.056 -0.050  0.229 -0.074  0.134
## x11 -0.019 -0.035 -0.071 -0.040 -0.056 -0.047  0.033  0.057 -0.022 -0.063
## x12  0.448  0.326  0.470  0.393  0.367  0.339  0.470  0.237  0.351  0.510
## x13  0.848  0.823  0.779  0.402  0.642  0.890  0.631  0.709 -0.325  0.868
## x14 -0.705 -0.642 -0.570 -0.143 -0.373 -0.643 -0.631 -0.636  0.102 -0.633
##        x10    x11    x12    x13    x14
## y    0.012 -0.019  0.448  0.848 -0.705
## x1   0.010 -0.035  0.326  0.823 -0.642
## x2   0.022 -0.071  0.470  0.779 -0.570
## x3   0.027 -0.040  0.393  0.402 -0.143
## x4   0.092 -0.056  0.367  0.642 -0.373
## x5   0.056 -0.047  0.339  0.890 -0.643
## x6  -0.050  0.033  0.470  0.631 -0.631
## x7   0.229  0.057  0.237  0.709 -0.636
## x8  -0.074 -0.022  0.351 -0.325  0.102
## x9   0.134 -0.063  0.510  0.868 -0.633
## x10  1.000  0.013  0.107  0.114 -0.172
## x11  0.013  1.000 -0.067 -0.025 -0.176
## x12  0.107 -0.067  1.000  0.403 -0.406
## x13  0.114 -0.025  0.403  1.000 -0.657
## x14 -0.172 -0.176 -0.406 -0.657  1.000
```

```
corrplot.mixed(corr_matrix, lower.col = "black", number.cex = .7, upper = "ellipse")
```



Multiple Regression

At this point we have interest in building a model for calories using some combination of the explanatory variables. Using multiple regression one initially build a model with all of the possible explanatory variables. Below is some R output for this Multiple Linear Regression (MLR) analysis.

General Form for a Multiple Regression Model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_{14} X_{14}$$

```
model.1 <- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x12+x13, data=mcdonalds)

summary(model.1)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11 + x12 + x13, data = mcdonalds)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.4175 -0.2747  0.0243  0.2977  2.5216
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept) -0.0535084  0.0855611  -0.625  0.53231
## x1           8.8817894  0.0961551  92.369  < 2e-16 ***
## x2           0.3572917  0.2037190   1.754  0.08072 .
## x3           2.0976351  1.1667932   1.798  0.07346 .
## x4          -0.0128552  0.0050975  -2.522  0.01232 *
## x5           0.0016771  0.0020662   0.812  0.41778
## x6           4.1577971  0.0460569  90.275  < 2e-16 ***
## x7          -1.0142317  0.3272456  -3.099  0.00217 **
## x8          -0.1780114  0.0581046  -3.064  0.00243 **
## x9           3.7992028  0.0931406  40.790  < 2e-16 ***
## x10          0.0028241  0.0017583   1.606  0.10954
## x11         -0.0040799  0.0048318  -0.844  0.39929
## x12         -0.0004547  0.0035534  -0.128  0.89828
## x13         -0.1625323  0.5564523  -0.292  0.77047
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5893 on 242 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 4.913e+04 on 13 and 242 DF,  p-value: < 2.2e-16
```

Equation of the Model with all of the Explanatory Variables

(Note: This is referred to at the Full Model)

$$Y = -0.054 + 8.882X_1 + 0.357X_2 + 2.098X_3 - 0.013X_4 + \cdots - 0.163X_{13}$$

Coefficient of Determination

Interpretation: **99.96** % of the variability in calories is accounted for in this model, (i.e., is accounted for in the model between calories and the thirteen explanatory variables).

Test for the Significance of the Model:

Ho: None of the explanatory variables is a linear predictor of calories (i.e., the model is not significant or is not useful in predicting the response)

Ha: At least one of the explanatory variables is a significant linear predictor of calories (i.e., the model is significant or at least some portion of the model is useful in predicting the response)

Test statistic: F* = 4.563e+04

P-value: $<$ 2.2e-16

Conclusion: Reject Ho in favor of Ha. There is sufficient evidence to conclude that at least one of the explanatory variables is a significant linear predictor of calories (i.e., the model is significant or at least some portion of the model is useful in predicting the response)

Further Analysis

Since at least one of the independent variables is significant, we do further analysis to determine which one(s) is/are significant.

Test for an Individual Predictor in this Model:

Ho: With x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, and x13 in the model, x12 is not a linear predictor of y

Ha: With x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, and x13 in the model, x12 is a significant linear predictor of y

Test statistic: t* = -0.128

P-value: 0.89828

Conclusion: Fail to reject Ho. There is insufficient evidence to conclude that with x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, and x13 in the model, x12 is not a linear predictor of y

Notice that x12 has the largest p-value and thus is the least significant. We could remove it from the model and rerun the analysis. Then we could test for significance of another independent variable. We could continue this process until only significant variables are left. This method for identifying the best model is referred to as **Backward Selection**.

Some selected output for the **Backward Selection** procedure:

```
model.2 <- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11+x13, data=mcdonalds)

summary(model.2)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11 + x13, data = mcdonalds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.41818 -0.27335  0.02511  0.29646  2.51582
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.054183   0.085225  -0.636  0.52553
## x1           8.885830   0.090639  98.036  < 2e-16 ***
## x2           0.349597   0.194248   1.800  0.07314 .
## x3           2.096475   1.164394   1.800  0.07302 .
## x4          -0.012981   0.004992  -2.600  0.00989 **
## x5           0.001710   0.002046   0.836  0.40398
## x6           4.157373   0.045844  90.685  < 2e-16 ***
## x7          -1.020775   0.322571  -3.165  0.00175 **
## x8          -0.179371   0.057009  -3.146  0.00186 **
## x9           3.793077   0.079735  47.571  < 2e-16 ***
## x10          0.002833   0.001753   1.616  0.10739
## x11         -0.004076   0.004822  -0.845  0.39878
## x13         -0.153017   0.550345  -0.278  0.78122
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5881 on 243 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 5.344e+04 on 12 and 243 DF,  p-value: < 2.2e-16
```

```
model.3 <- lm(y~x1+x2+x3+x4+x5+x6+x7+x8+x9+x10+x11, data=mcdonalds)

summary(model.3)
```

```
##
## Call:
```

```
## lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7 + x8 + x9 +
##     x10 + x11, data = mcdonalds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.46208 -0.27072  0.02635  0.29811  2.53002
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.052104   0.084736  -0.615  0.53920
## x1           8.896960   0.081167 109.614  < 2e-16 ***
## x2           0.331463   0.182625   1.815  0.07075 .
## x3           1.978195   1.081850   1.829  0.06869 .
## x4          -0.013358   0.004796  -2.785  0.00577 **
## x5           0.001576   0.001984   0.794  0.42779
## x6           4.151463   0.040542 102.400  < 2e-16 ***
## x7          -1.033354   0.318778  -3.242  0.00135 **
## x8          -0.172700   0.051617  -3.346  0.00095 ***
## x9           3.787788   0.077285  49.010  < 2e-16 ***
## x10          0.002797   0.001745   1.603  0.11026
## x11         -0.004043   0.004811  -0.840  0.40158
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.587 on 244 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 5.852e+04 on 11 and 244 DF,  p-value: < 2.2e-16
```

```
model.4 <- lm(y~x1+x2+x3+x4+x6+x7+x8+x9+x10+x11, data=mcdonalds)

summary(model.4)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x6 + x7 + x8 + x9 + x10 +
##     x11, data = mcdonalds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.51832 -0.26681  0.01471  0.30730  2.67349
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.048483   0.084549  -0.573 0.566880
## x1           8.894925   0.081065 109.726  < 2e-16 ***
## x2           0.401547   0.159776   2.513 0.012607 *
## x3           1.598667   0.969883   1.648 0.100571
## x4          -0.013501   0.004789  -2.819 0.005204 **
## x6           4.163518   0.037566 110.833  < 2e-16 ***
## x7          -1.093311   0.309479  -3.533 0.000491 ***
## x8          -0.195280   0.043051  -4.536 8.98e-06 ***
## x9           3.834571   0.050001  76.690  < 2e-16 ***
## x10          0.002764   0.001743   1.586 0.114117
## x11         -0.004098   0.004807  -0.852 0.394789
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5865 on 245 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 6.447e+04 on 10 and 245 DF,  p-value: < 2.2e-16
```

```
model.5 <- lm(y~x1+x2+x3+x4+x6+x7+x8+x9+x10, data=mcdonalds)

summary(model.5)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x6 + x7 + x8 + x9 + x10,
##     data = mcdonalds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.50768 -0.26522  0.01231  0.30185  2.69375
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.055384   0.084114  -0.658 0.510875
## x1           8.896820   0.080990 109.852  < 2e-16 ***
## x2           0.407671   0.159526   2.556 0.011206 *
## x3           1.558451   0.968196   1.610 0.108758
## x4          -0.013557   0.004785  -2.833 0.004994 **
## x6           4.159962   0.037313 111.490  < 2e-16 ***
## x7          -1.098810   0.309240  -3.553 0.000456 ***
## x8          -0.191996   0.042855  -4.480 1.14e-05 ***
## x9           3.838566   0.049753  77.152  < 2e-16 ***
## x10          0.002725   0.001742   1.564 0.119018
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5862 on 246 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 7.171e+04 on 9 and 246 DF,  p-value: < 2.2e-16
```

```
model.6 <- lm(y~x1+x2+x3+x4+x6+x7+x8+x9, data=mcdonalds)

summary(model.6)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x3 + x4 + x6 + x7 + x8 + x9, data = mcdonalds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.48958 -0.25429  0.01117  0.28908  2.62973
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.035606   0.083402  -0.427  0.66981
## x1           8.885881   0.080923 109.807  < 2e-16 ***
## x2           0.410993   0.159978   2.569  0.01079 *
```

```
## x3           1.493471   0.970135   1.539  0.12498
## x4          -0.013135   0.004792  -2.741  0.00657 **
## x6           4.143245   0.035854 115.559  < 2e-16 ***
## x7          -0.891481   0.280217  -3.181  0.00165 **
## x8          -0.176626   0.041835  -4.222 3.41e-05 ***
## x9           3.856634   0.048536  79.459  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5879 on 247 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 8.02e+04 on 8 and 247 DF,  p-value: < 2.2e-16
```

```
model.7 <- lm(y~x1+x2+x4+x6+x7+x8+x9, data=mcdonalds)

summary(model.7)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2 + x4 + x6 + x7 + x8 + x9, data = mcdonalds)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.53827 -0.27597  0.02063  0.29637  2.71109
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.049023   0.083174  -0.589  0.55612
## x1           8.872330   0.080665 109.990  < 2e-16 ***
## x2           0.471516   0.155500   3.032  0.00268 **
## x4          -0.013661   0.004793  -2.850  0.00473 **
## x6           4.136878   0.035713 115.837  < 2e-16 ***
## x7          -0.909683   0.280740  -3.240  0.00136 **
## x8          -0.167353   0.041514  -4.031 7.39e-05 ***
## x9           3.883821   0.045334  85.672  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5895 on 248 degrees of freedom
## Multiple R-squared:  0.9996, Adjusted R-squared:  0.9996
## F-statistic: 9.115e+04 on 7 and 248 DF,  p-value: < 2.2e-16
```

Final Model using Backward Selection:

$$Y = -0.049 + 8.872X_1 + 0.472X_2 - 0.014X_4 + 4.137X_6 - 0.909X_7 - 0.167X_8 + 3.884X_9$$

Coefficient of Determination

(Assessing the fit of the model) 99.96 % of the variability in the calories is accounted for in this multiple linear regression model.

Prediction:

For a sample that has a total fat (x1) of 20, a saturated fat (x2) of 11, a cholestrol (x4) of 35, a carbohydrates (x6) of 36, a dietary fiber (x7) of 2 , a sugar (x8) of 3 and a protein (x9) of 20, we predict the calories (y) to be 406.386.

Estimation:

For samples that have a total fat (x1) of 20, a saturated fat (x2) of 11, a cholestrol (x4) of 35, a carbohydrates (x6) of 36, a dietary fiber (x7) of 2 , a sugar (x8) of 3 and a protein (x9) of 20, we predict the average calories (y) to be 406.386.

Interpretation of Partial Slopes (B-weights or Coefficients):

For a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), and a fixed sugar (x8), as the protein increases by 1, the calories increases by 3.884.

For a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), and a fixed protein (x9), as the sugar increases by 1, the calories decreases by 0.167.

For a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed protein (x9), and a fixed sugar (x8), as the dietary fiber increases by 1, the calories decreases by 0.909.

For a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed dietary fiber (x7), a fixed protein (x9), and a fixed sugar (x8), as the carbohydrates increases by 1, the calories increases by 4.137.

For a fixed total fat (x1), a fixed saturated fat (x2), a fixed carbohydrates (x6), a fixed dietary fiber (x7), a fixed protein (x9), and a fixed sugar (x8), as the cholestrol increases by 1, the calories decreases by 0.014.

For a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), a fixed protein (x9), and a fixed sugar (x8), as the total fat increases by 1, the calories increases by 8.872.

For a fixed total fat (x1), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), and a fixed sugar (x8), and a fixed protein (x9) as the saturated fat increases by 1, the calories increases by 0.472.

Confidence Intervals for the Coefficients

```
confint(model.7, level=0.95)
```

```
##                    2.5 %       97.5 %
## (Intercept) -0.21284084  0.114793940
## x1           8.71345478  9.031205670
## x2           0.16524752  0.777784778
## x4          -0.02310119 -0.004221547
## x6           4.06653886  4.207217726
## x7          -1.46262105 -0.356745606
## x8          -0.24911721 -0.085588355
## x9           3.79453345  3.973109217
```

We are 95% confident that for a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), and a fixed sugar (x8), as the protein increases by 1, the calories increases between 3.795 and 3.973

We are 95% confident that for a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), and a fixed protein (x9), as the sugar increases by 1, the calories decreases between -0.249 and -0.086

We are 95% confident that for a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), and a fixed sugar (x8), as the dietary fiber increases by 1, the calories increases between -1.463 and -0.357

We are 95% confident that for a fixed total fat (x1), a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed protein (x9), a fixed dietary fiber (x7), and a fixed sugar (x8), as the carbohydrates increases by 1, the calories increases between 4.067 and 4.207

We are 95% confident that for a fixed total fat (x1), a fixed saturated fat (x2), a fixed carbohydrates (x6), a fixed dietary fiber (x7), a fixed protein (x9), and a fixed sugar (x8), as the cholestrol increases by 1, the calories decreases between -0.023 and -0.004

We are 95% confident that for a fixed total fat (x1), a fixed carbohydrates (x6), a fixed dietary fiber (x7), a fixed protein (x9), and a fixed sugar (x8), as the saturated fat increases by 1, the calories increases between 0.165 and 0.778

We are 95% confident that for a fixed saturated fat (x2), a fixed cholestrol (x4), a fixed carbohydrates (x6), a fixed dietary fiber (x7), a fixed protein (x9), and a fixed sugar (x8), as the total fat increases by 1, the calories increases between 8.713 and 9.031

Residual Analysis

This is checking the assumptions that need to be satisfied before it is appropriate to perform inference from a multiple regression model.

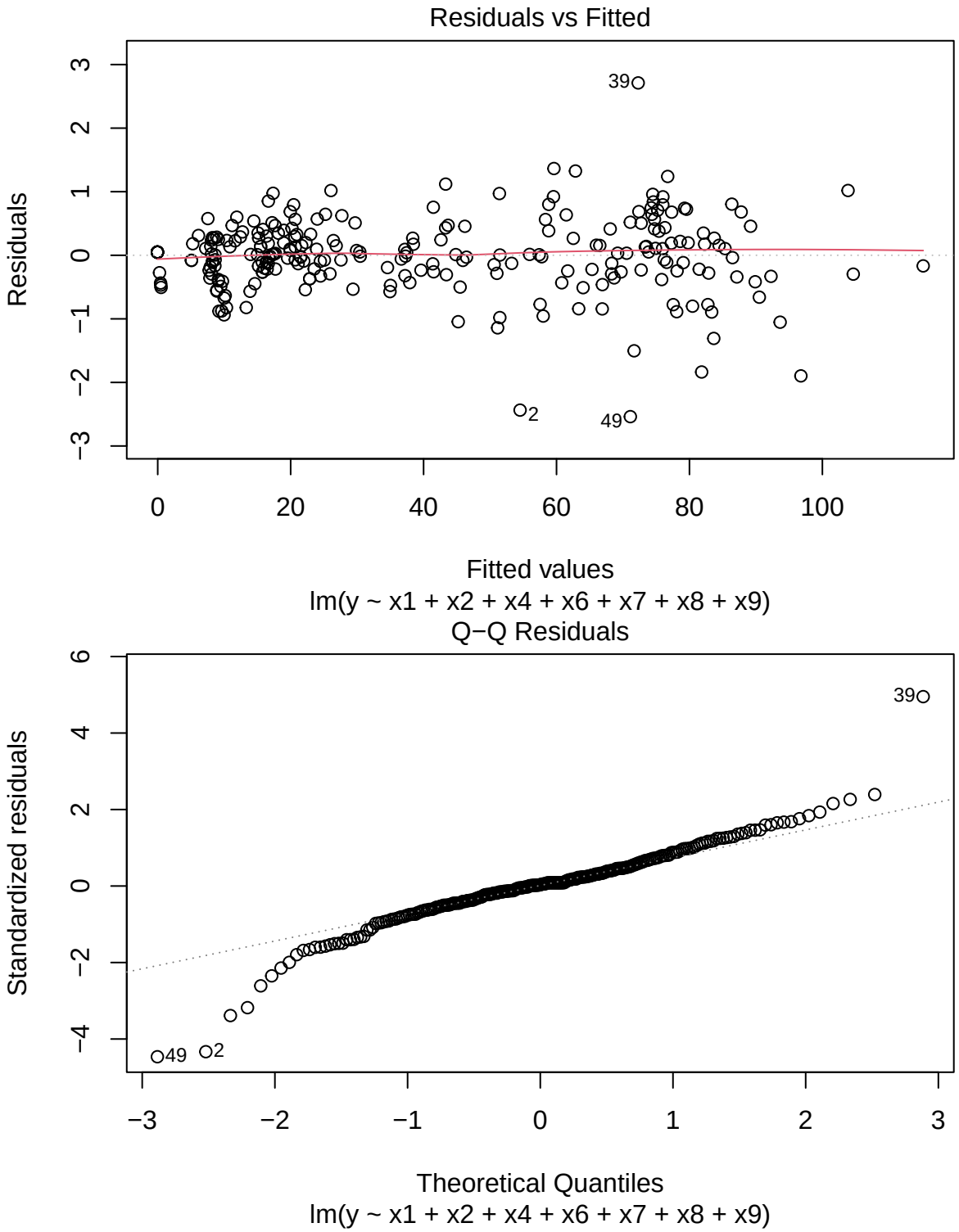
1. The random errors are independent of each other.
2. The random errors are normally distributed
3. The random errors have constant variance (homoscedasticity)

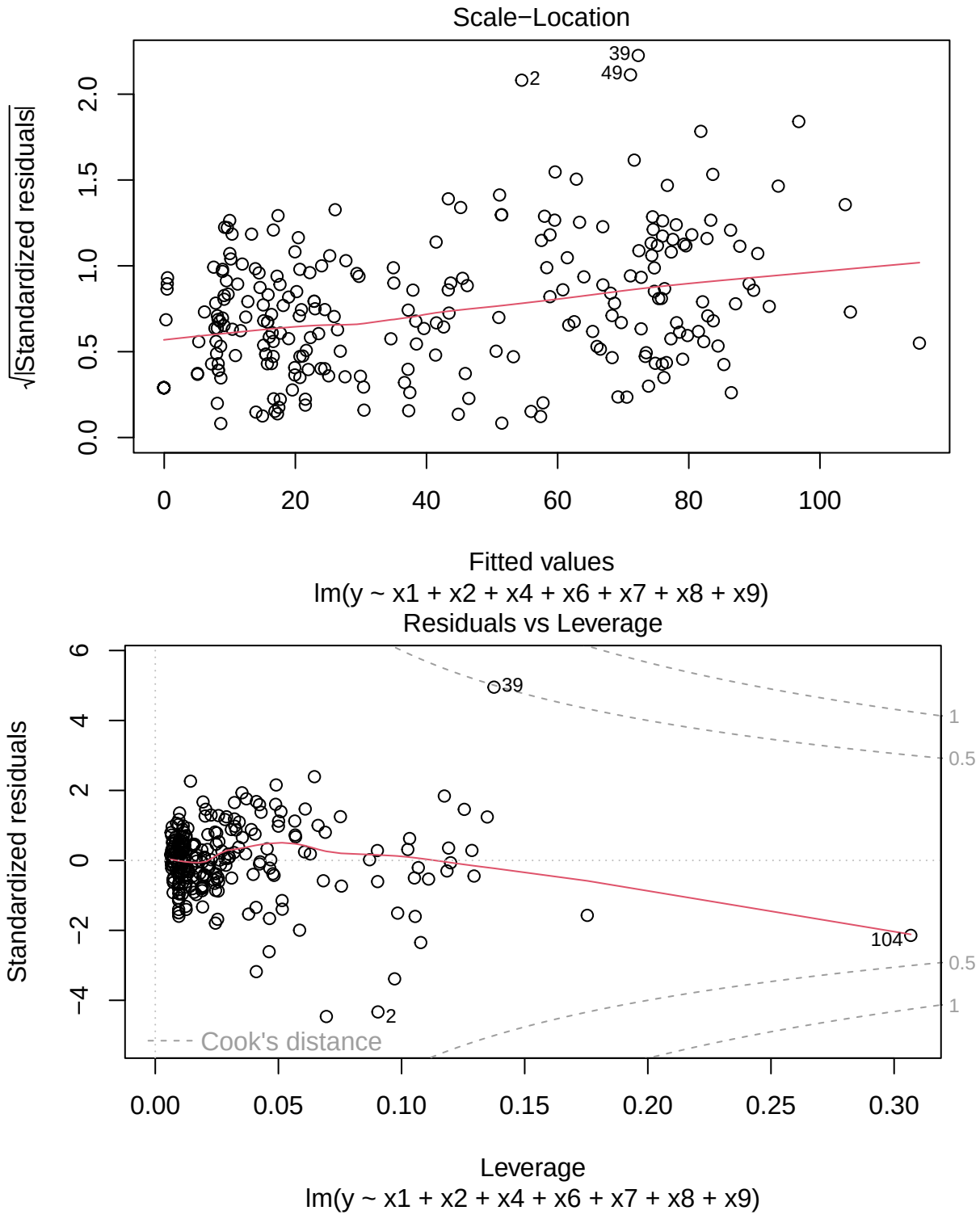
(Note: There is a lot more to residual analysis than just these things, but to keep things focused on the inferential aspects of linear regression we will just do this quick check.)

```
bptest(y~x1+x2+x4+x6+x7+x8+x9, varformula = ~ fitted.values(model.7), studentize=TRUE, data=mcdonalds)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  y ~ x1 + x2 + x4 + x6 + x7 + x8 + x9
## BP = 16.973, df = 1, p-value = 3.791e-05
```

```
#oldpar <- par(oma=c(0,0,3,0), mfrow=c(2,2))
plot(model.7)
```

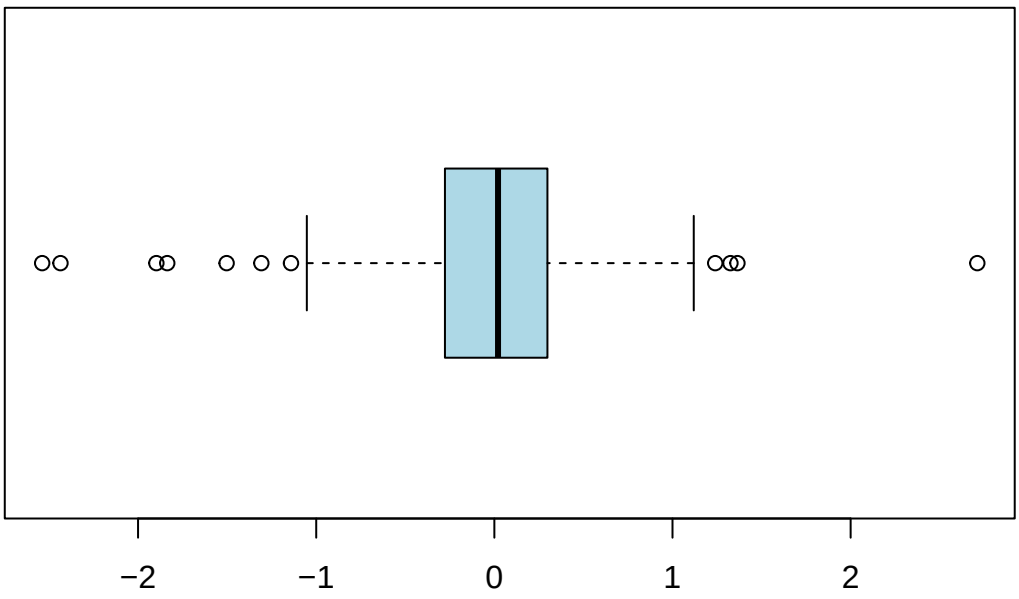

Residuals vs Fitted
Residuals
Fitted values
lm(y ~ x1 + x2 + x4 + x6 + x7 + x8 + x9)
Q−Q Residuals
Standardized residuals
Theoretical Quantiles
lm(y ~ x1 + x2 + x4 + x6 + x7 + x8 + x9)




```
#par(oldpar)
```

```
# Examination of the distribution of the residuals

boxplot(model.7$residuals, col="lightblue", horizontal = TRUE)
```




```
shapiro.test(model.7$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  model.7$residuals
## W = 0.94438, p-value = 2.781e-08
```

Ho: The residuals follow a normal distribution

Ha: The residuals do not follow a normal distribution

Test statistic: W* = 0.94438

P-value: 2.781e-08

Conclusion (at the .05 level): Reject Ho in favor of Ha. There is sufficient evidence to conclude that the residuals do not follow a normal distribution.

In multiple linear regression, one key assumption is that the residuals (errors) should be normally distributed. However, in our case, residuals do not follow a normal distribution. This suggests that our model may not be a good fit for the Macdonald's menu data. We could explore alternative regression techniques or include interaction terms in the model to better capture the relationships in the data.