ITEC 621 - Homework 1

R, Stats and Regression Refresher

Prof. J. Alberto Espinosa

February 4, 2023

Table of Contents

Overview – Read this carefully

The goal of this homework is to practice with R, R Studio, R Markdown and with simple statistical analysis. Parts of This homework are somewhat similar to HW0 in the KSB-999 R Overview for Business Analytics (Canvas), which you were required to complete on your own. The rest of the homework is about the stats and regression refresher.

** The following applies to all homework assignments in this class, so please read carefully. **

It is not a bad idea to complete this homework (and all homework) in a plain R script first. Once you are satisfied that all your R scripts work properly, you can then open the R Markdown template file **HW1_YourLastName.Rmd**, re-name it using **your actual last**

name and copy over your work to the corresponding code chunk sections in the R Markdown template. If you are comfortable working in R Markdown, you can work directly in the HW template.

Knitting (up to 10 pts.)

You are required to **knit ALL** your homework R Markdown files into a **Word** (preferred), PDF or HTML file. Learning how to prepare analytic reports using the `{knitr}` package (which is what R Markdown uses) is an important learning objective of this course. You are expected to submit your homework in a properly knitted with **business-like** formatting and appearance. **No knitting, inadequate knitting and/or improper formatting of the document will carry point deductions up to 10 points.**

Important Notes about Knitting and Formatting:

- Your R code must be visible in your knitted document. This means that your R Markdown file **MUST** have the attribute `echo = T` in the **{r global_options}** setting below. We need to be able to see your R code to grade your homework. The template provided for the homework usually has the `echo = T` setting, but it is your responsibility to ensure that it is set correctly.

- The knitted file must have a table of contents that include all `Heading 1 (#)` and `Heading 2 (##)` entries. Please review your R Markdown file to ensure that these headings are the only text with # or ## tags. Otherwise, your narrative text will be improperly formatted (with large blue font) and the text will also appear in the table of contents, which is not appropriate for a business document.

- Enter your narrative answers to interpretation questions in the text areas (without # tags), not in the R code chunk. It is OK to enter text in the R code chunks with a # tag, but these should be used to make comments and annotations about your script, not for interpretations. Related to this, please note that comments in R code chunks with the # tag sometimes don't knit well and the text doesn't wrap at the end of the line, preventing me from being able to read all your comments. This happens frequently when knitting to PDF files, but I have seen this problem with Word too. Please be aware that **I cannot give you credit for what I cannot read !!**. It is your responsibility to ensure that I can read all your text. If a line is not wrapping correctly, you can always break it.

- Overall, anything that would not be acceptable to a management or client audience is not acceptable in knitted documents.

Interpretations:

The goal in this course is NOT to make you proficient in R, although you will get a lot of R practice in this class. One important goal is to be able to extract meaningful business insights from your analysis. As such, all **interpretation questions** will be graded rigorously in every homework. Please think through every interpretation question and

respond concisely, but accurately. Your analysis must demonstrate that you understand how to interpret the output of your models.

Submission:

I will always display the solution output in the homework instructions, so that you can compare your results against the solution. In questions involving random sampling, your outputs may differ slightly from the solution. This is OK, but if in doubt, please ask a TA or me. Once done, submit your knitted document in Canvas.

Global Options

R Markdown allows you to set global options that affect the entire knitted document. But you can change these options in specific R code chunks. Specific option settings override the global settings. For example, to show all your code, set the attribute `echo = T` in the **global_options** (the `warning = F` and `message = F` attributes suppress warnings and messages in your knitted file.

Q1. Functions (10 pts.)

Let's refresh some concepts from HW0. Write a function to compute and `return()` the hypotenuse of a squared triangle of sides a and b. The hypotenuse is equal to the square root of the sum of the squares of the two sides (i.e., `round(sqrt(a^2 + b^2), digits = 2)`. Call this function **hyp** and pass arguments a and b (i.e., `function(a, b)`). Notice that the `sqrt()` function is embedded inside the `round()` function to limit the number of decimal points displayed.

In the next line after the function definition, store a value of **7** in a variable named **a** and **10** in **b**. Then, use the function `paste()` to output this result: `"The hypotenuse of a triangle with sides", a, "and", b, "is", hyp(a, b)`. **Technical note:** in some R routines (such as in functions), `paste()` will compute a value, but will not display a result (and sometimes it will). To display your results, enclose everything in the `paste()` function within the `print()` function.

A technical tip: A common mistake is to have an incorrect number of closing parentheses in a formula with functions. In the example above you have 3 opening parentheses, one for `print(` one for `paste(` and one for `hyp(`. This means that you need to have 3 closing parentheses somewhere to close each of the 3 functions.

```
## [1] "The hypotenuse of a triangle with sides 7 and 10 is 12.21"
```

Q2. Data Work (10 pts.)

Note: The **PizzaCal** data set contains the grams of moisture, protein, etc. per slice of pizza. The pizzas are categorized by brand and whether the brand is imported (1) or domestic (0).

2.1 Read the **PizzaCal.csv** data table into a data frame named **Pizza** (tip: use the `read.table()` function with `header = T`, `row.names = 1` and `sep = ","`. Display the **first 6 rows** (use the `head()` function) of this data set.

```
##       brand import  mois  prot   fat  ash sodium  carb cal
## 14001     D      0 47.17 22.29 21.30 4.08   0.74  5.16 302
## 14002     D      0 49.16 27.99 17.49 3.29   0.39  2.07 278
## 14003     A      1 30.49 21.28 41.65 4.82   1.64  1.76 467
## 14004     B      0 52.68 14.38 25.72 3.26   0.93  3.96 305
## 14005     H      0 33.05  7.34 15.78 1.34   0.42 42.49 341
## 14006     H      0 35.55  7.32 16.40 1.76   0.36 38.97 333
```

2.2 Then, display the object class for the **Pizza** data frame and for the vectors **cal** (i.e., `Pizza$cal`), **fat** and **brand**.

```
## [1] "data.frame"

## [1] "integer"

## [1] "numeric"

## [1] "character"
```

2.3 Then create a matrix called **Pizza.mat** that contains only the quantitative variables in the data set (i.e., mois through cal - tip: the index `[, 3:9]` will select all rows and columns 3 to 9 in the matrix). Display the class of the **Pizza.mat** object and then list the first 6 rows of this matrix, just to ensure you did the right thing.

```
## [1] "matrix" "array"

##        mois  prot   fat  ash sodium  carb cal
## 14001 47.17 22.29 21.30 4.08   0.74  5.16 302
## 14002 49.16 27.99 17.49 3.29   0.39  2.07 278
## 14003 30.49 21.28 41.65 4.82   1.64  1.76 467
## 14004 52.68 14.38 25.72 3.26   0.93  3.96 305
## 14005 33.05  7.34 15.78 1.34   0.42 42.49 341
## 14006 35.55  7.32 16.40 1.76   0.36 38.97 333
```

Q3. Descriptive Statistics (10 pts.)

Let's analyze the data quantitatively. First get a `summary()` of the **Pizza** data frame and inspect the frequencies. Then load the **{psych}** library and display the descriptive statistics for the data set using the `describe()` function, but only for the quantitative variables in columns 3 to 9. Since this function provides many descriptive stats, let's limit the display to a few important statistics in columns 1 to 9. To do this, add the index `[3:9, 1:9]` after the `describe()` function.

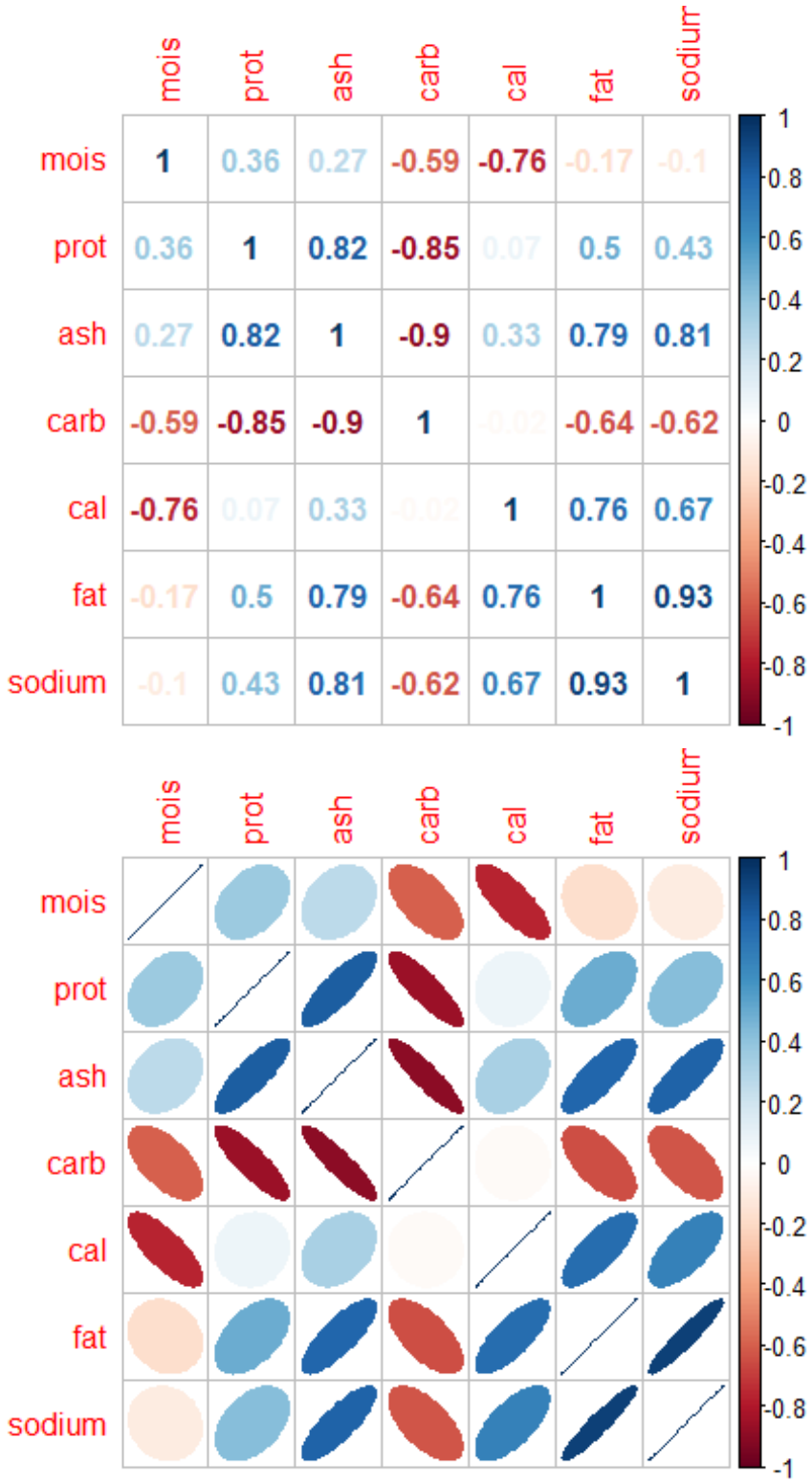
```
##     brand               import            mois            prot      
##  Length:300         Min.   :0.0000   Min.   :25.00   Min.   : 6.98  
##  Class :character   1st Qu.:0.0000   1st Qu.:30.90   1st Qu.: 8.06  
##  Mode  :character   Median :0.0000   Median :43.30   Median :10.44  
```

```
##                    Mean   :0.2933   Mean   :40.90   Mean   :13.37
##                    3rd Qu.:1.0000   3rd Qu.:49.12   3rd Qu.:20.02
##                    Max.   :1.0000   Max.   :57.22   Max.   :28.48
##       fat              ash             sodium            carb
##  Min.   : 4.38   Min.   :1.170   Min.   :0.2500   Min.   : 0.510
##  1st Qu.:14.77   1st Qu.:1.450   1st Qu.:0.4500   1st Qu.: 3.467
##  Median :17.14   Median :2.225   Median :0.4900   Median :23.245
##  Mean   :20.23   Mean   :2.633   Mean   :0.6694   Mean   :22.865
##  3rd Qu.:21.43   3rd Qu.:3.592   3rd Qu.:0.7025   3rd Qu.:41.337
##  Max.   :47.20   Max.   :5.430   Max.   :1.7900   Max.   :48.640
##       cal
##  Min.   :218.0
##  1st Qu.:291.0
##  Median :321.5
##  Mean   :327.1
##  3rd Qu.:352.0
##  Max.   :508.0

##        vars   n   mean    sd median trimmed   mad    min    max
## mois      3 300  40.90  9.55  43.30   40.83 12.24  25.00  57.22
## prot      4 300  13.37  6.43  10.44   12.54  3.99   6.98  28.48
## fat       5 300  20.23  8.98  17.13   18.55  4.71   4.38  47.20
## ash       6 300   2.63  1.27   2.22    2.49  1.21   1.17   5.43
## sodium    7 300   0.67  0.37   0.49    0.58  0.12   0.25   1.79
## carb      8 300  22.86 18.03  23.24   22.56 28.50   0.51  48.64
## cal       9 300 327.10 62.00 321.50  319.69 45.22 218.00 508.00
```

Q4. Correlation Analysis (10 pts.)

4.1 Then create a correlation object named **Pizza.cor** using the `cor()` function. Then load the **{corrplot}** library and feed this **Pizza.cor** object into the `corrplot()` function. Add the parameter "`order = hclust`" to group the cluster the variables by correlation strength, and the parameters `method = number` to display correlation values. Then run the same `corrplot()` function, but this time use `method = ellipse` to get a graphical display.



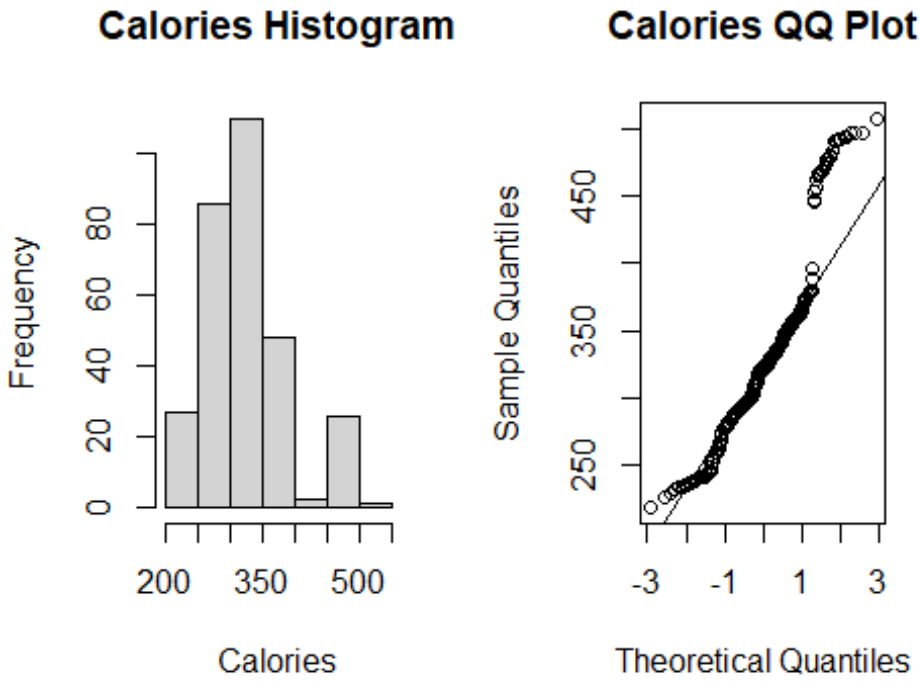
4.2 Based on the correlation results above, suggest two desirable predictors to include in a regression model with **cal** as the outcome variable. Provide a brief rationale for your selection.

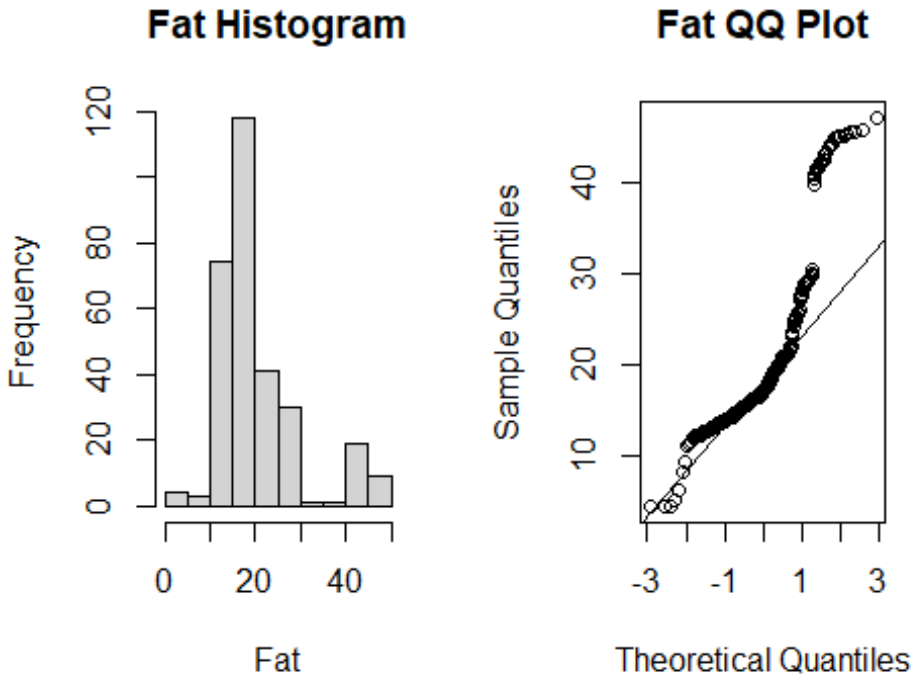
Q5. Descriptive Analytics: Normality (10 pts.)

5.1 Divide the graph output to 1 row and 2 columns (`par(mfrow = c(1, 2))`). Then draw a histogram for the **cal** variable. Title your diagram **"Calories Histogram"** and label the x axis **"Calories"**.

Then draw a **QQ Plot** to inspect the normality of this variable (tip: this is a 2 step process; first draw the QQ Plot with the `qqnorm()` function and give it a main title of `"Calories QQ Plot"`), then draw the QQ Plot line with the function `qqline()`.

Also, draw a histogram and a QQ Plot for the **fat** variable. Title the histogram **Fat Histogram"** and label the x axis **Fat**. Title the QQ Plot **Fat QQ Plot"**. Then reset the graph output to 1 row and 1 column.

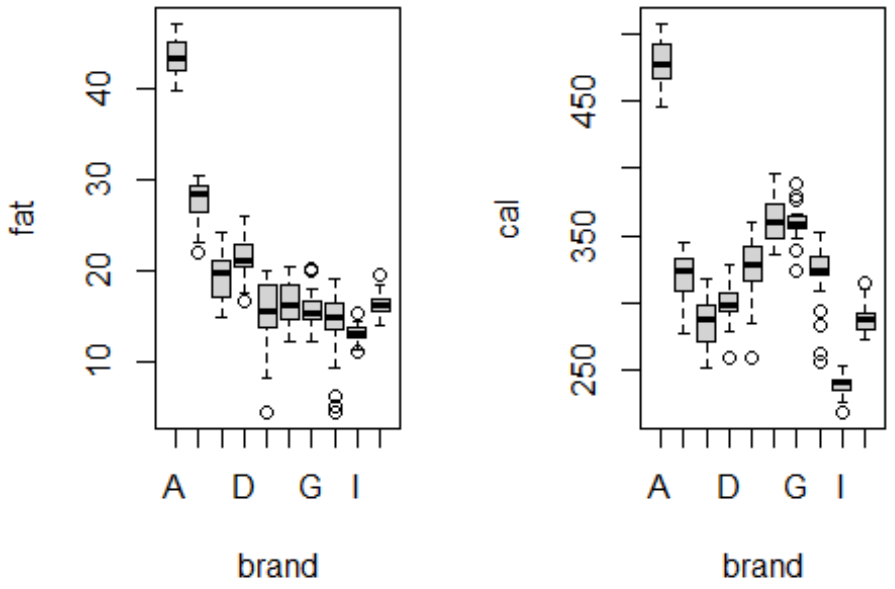




5.2 Briefly answer: Do calories and fat appear to be normally distributed? Why or why not.

Q6. Descriptive Analytics: Boxplots and ANOV (10 pts.)

6.1 Divide the graph output to 1 row and 2 columns. Then draw 2 boxplots one for **fat** by **brand** and another for **cal** by **brand**. Then reset the graph output back to 1 row and 1 column.



6.2 Then conduct two **ANOVA** tests using the `aov()` function, one to evaluate if **fat** varies **by brand** and another to evaluate if **calories** vary **by brand**. Store the results of the first **ANOVA** test in an object named **aov.fat** [not aov.brand] and the second one named **aov.cal**. Then display the summary of each of these objects, but write the function `cat("\n")` in between the two summaries to separate the displays with a blank line. **Technical note:** the `cat()` function concatenates and prints strings and `"\n"` is the code for a new line.

```
##              Df Sum Sq Mean Sq F value Pr(>F)
## brand         9  22338    2482   411.4 <2e-16 ***
## Residuals   290   1750       6
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

##              Df  Sum Sq Mean Sq F value Pr(>F)
## brand         9 1068297  118700     424 <2e-16 ***
## Residuals   290   81186     280
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

6.3 Briefly answer: Does fat vary by brand? And, do calories vary by brand? Briefly explain why or why not. Please refer to **both**, the visual boxplot and the quantitative ANOVA output.

Q7. Simple Linear Regression Model (10 pts.)

7.1 Fit a **simple** linear regression model object with the `lm()` function to predict **calories** using **fat** as the only predictor. Store your linear model results in an object named **fit.simple**. Then display the `summary()` results.

```
##
## Call:
## lm(formula = cal ~ fat, data = Pizza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -60.089 -36.161  -8.474  32.265  83.369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 220.2558     5.7067   38.60   <2e-16 ***
## fat           5.2816     0.2579   20.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 40.03 on 298 degrees of freedom
## Multiple R-squared:  0.5846, Adjusted R-squared:  0.5832
## F-statistic: 419.3 on 1 and 298 DF,  p-value: < 2.2e-16
```

7.1 If **fat** is in grams per slice and **cal** is in calories per slice, provide a brief **interpretation** of both, the **significance** and **effect** of fat on cal.

Q8. Linear Regression Model with a Binary Predictor (10 pts.)

8.1 Now fit a larger linear regression model, same as above, but add **import** as a predictor. Name the resulting linear model **fit.dummy**. Then display the `summary()` results.

```
##
## Call:
## lm(formula = cal ~ import + fat, data = Pizza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -50.582 -13.590  -4.148  10.972  57.155
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 225.0086     3.5304   63.73   <2e-16 ***
## import       73.6061     3.3438   22.01   <2e-16 ***
## fat           3.9793     0.1699   23.42   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.72 on 297 degrees of freedom
## Multiple R-squared:  0.8421, Adjusted R-squared:  0.8411
## F-statistic: 792.1 on 2 and 297 DF,  p-value: < 2.2e-16
```

8.2 Provide a brief **interpretation** of both, the **significance** and **effect** of **import** on **cal**.

Q9. Multivariate Linear Model (10 pts.)

9.1 Now fit a larger linear regression model to predict calories, using import, fat, carb and mois as predictors. Name the resulting linear model **fit.full**. Then display the `summary()` results.

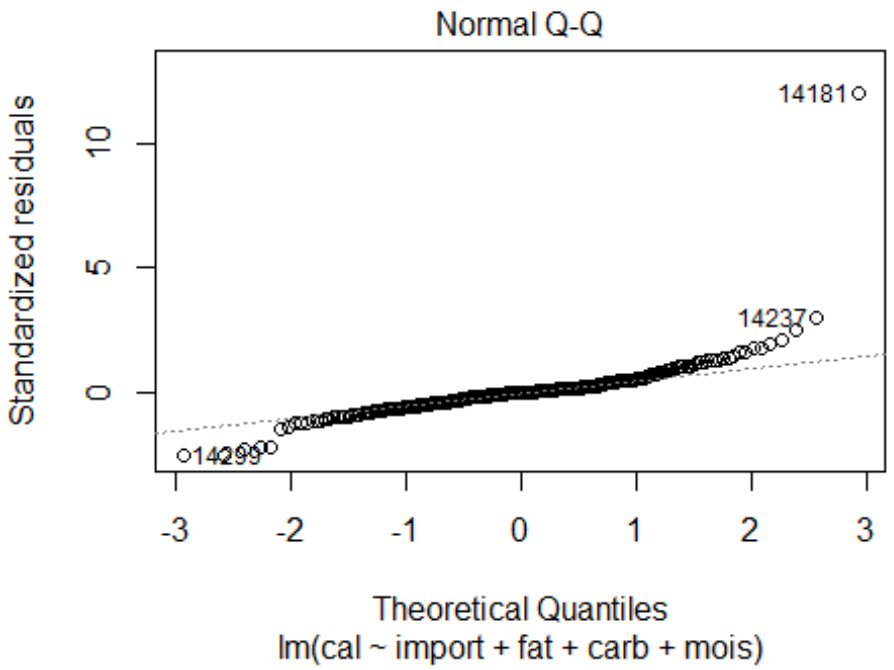
```
##
## Call:
## lm(formula = cal ~ import + fat + carb + mois, data = Pizza)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5803 -0.9505 -0.0796  0.5382 26.1168
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 371.69573    2.90280  128.05   <2e-16 ***
## import        0.06278    0.48398    0.13    0.897
## fat           5.03099    0.04002  125.72   <2e-16 ***
## carb          0.34040    0.02435   13.98   <2e-16 ***
## mois         -3.76920    0.04083  -92.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.183 on 295 degrees of freedom
## Multiple R-squared:  0.9988, Adjusted R-squared:  0.9988
## F-statistic: 6.02e+04 on 4 and 295 DF,  p-value: < 2.2e-16
```

9.2 Then provide a brief **interpretation** of both, the **significance** and **effect** of both, **import** and **fat** on **cal**.

9.3 Briefly answer: The **import** predictor was significant in the model in 8.1 above, but it is no longer significant in this multivariate model. Which result do you believe more, the one in 8.1 or this one? Briefly explain why.

Q10. Residual Plots and Model Evaluation (10 pts.)

10.1 Let's inspect the results and provide some final storytelling. First, `plot()` the **fit.full** object. This function yields 4 residual plots, but for now, we are only interested in the second residual plot, so add the attribute `which = 2`, which renders the QQ Plot of the residuals.



10.2 Then conduct an **ANOVA** test to compare all 3 models together, **fit.simple, fit.dummy and fit.full.

```
## Analysis of Variance Table
##
## Model 1: cal ~ fat
## Model 2: cal ~ import + fat
## Model 3: cal ~ import + fat + carb + mois
##   Res.Df    RSS Df Sum of Sq     F    Pr(>F)
## 1    298 477538
## 2    297 181469  1    296069 62101 < 2.2e-16 ***
## 3    295   1406  2    180063 18884 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

10.3 Which of the three models is preferred? Briefly explain why.