DATA SCIENCE AND MACHINE LEARNING (MSC)

DAMA51: Foundations in Computer Science

Academic Year: 2022–2023

#3 Written Assignment	
Submission Deadline	**Wed, 8 Mar 2023, 11:59 PM**
Student Name:	**Kritikos Antonios**

Remarks

The deadline is definitive.

An indicative solution will be posted online along with the returning of the graded assignments.

The assignment is due via the STUDY submission system. **You are expected to turn in a document (.DOC, .ODT, .PDF) and a compressed (.ZIP, .RAR) file containing all your work:**

- **1 document file (this document) with the answers to all the questions, along with the R code and the results of the execution of the code**
- **1 compressed file with 4 R scripts with the code that answers to each one of the problems to the Topics 3 and 5.**

You should not make any changes in the written assignment file other than providing your own answers. You should also type all of your answers into Word and not attach any handwritten notes as pictures into your work otherwise a 5% reduction of your final grade will be applied. Make sure to name all the files (ZIP file, DOC file and R script files) with **your last name first followed by a dash symbol and the names of each component at the end**. For example for the student with last name Aggelou the files should be named as follows: Aggelou-HW4.zip, Aggelou-HW4.doc, Aggelou-Topic3.R, and Aggelou-Topic5.R. The R script files should automatically run with the **source** command and generate the correct results. Also, please include comments before each command to explain the functionality of the command that follows. In the computations, use three decimal places.

Topic	Points	Grades
1. Online Quiz	40	
2. Article review	5	
3. Model Fiiting & Gradient Descent	20	
4. Confusion Matrix and ROC curve	20	
5. Outlier Detection	20	
TOTAL	**105 (max 100)**	**/100**

Topic 1: Online QUIZ

Complete the corresponding online quiz available at:

https://study.eap.gr/mod/quiz/view.php?id=25406

You have one effort and unlimited time to complete the quiz, up to the submission deadline. **(40 points)**

Topic 2: Article Review

The article "Challenges in Deploying Machine Learning: A Survey of Case Studies" (https://dl.acm.org/doi/10.1145/3533378) describes *data poisoning* and *model stealing* as two particularly relevant threats facing organizations which rely on ML projects. Select a business domain that is most familiar or interesting to you and describe a scenario where one of the threats might be considerably more serious than the other.

Note: You should write up your answer to a maximum of 300 words. Any text in excess of 300 words will not be taken into consideration.

(5 points)

Using a scenario which is very controversial in recent times, where a software development business has created a machine learning model for image recognition, which they plan to deploy in an autonomous vehicle, we have a model that has been trained on a large dataset of road images and has shown high accuracy in identifying objects such as cars, pedestrians, and traffic signs. The company wants to ensure that the model's predictions are accurate and reliable in real-world scenarios.

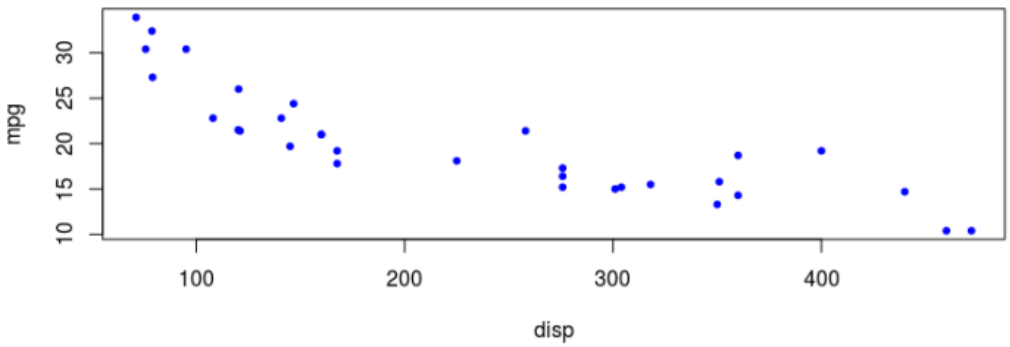
In this case, data poisoning could be a significant threat to the performance of the model. If an attacker can manipulate the training data used to build the model, it could result in the model being biased towards certain types of objects or situations. For instance, an attacker could add images of pedestrians wearing costumes that look like traffic signs, leading the model to misidentify them as traffic signs in the future. This could have serious consequences, such as causing the autonomous vehicle to stop or slow down unexpectedly, leading to accidents.

On the other hand, model stealing may not be as serious of a threat in this scenario. Model stealing involves an attacker obtaining the machine learning model by reverse engineering it from a deployed system or by stealing the model during transit. While this could lead to intellectual property theft, it may not necessarily affect the performance of the model in the same way that data poisoning would. Additionally, there are techniques such as model encryption that can be used to minimize the risk of model stealing.

In conclusion, in the context of a software development business creating a machine learning model for autonomous vehicle image recognition, data poisoning poses a more significant threat than model stealing. It is essential for organizations to be aware of the different threats facing their ML projects and take measures to mitigate them. This could involve monitoring the training data for anomalies, using model verification techniques, and implementing security measures such as data encryption and access control.

Topic 3: Model Fitting and Gradient Descent

We will use the *mtcars* data set. We want to investigate the relationship between miles per gallon (*mpg*) and engine displacement volume (*disp*) of the cars. A simple scatterplot is an excellent visual tool to assess the linearity between two variables. Below is a scatterplot of these two variables.



a. Does the plot imply that a linear relationship between disp and mpg might exist? **(2 Points)**

Yes it seems to be a linear relationship between disp and mpg, although we cannot be absolutely certain about that relationship unless we run some calculations and take a look at the summary of our model.

b. Use R to build a linear model for *mpg* (response variable) and *disp* (predictor variable). Print the corresponding slope and intercept. **(4 points)**

slope: -0.04122

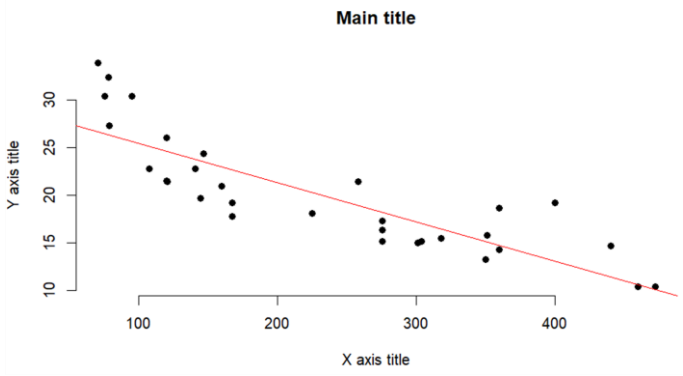
intercept: 29.59985

R code: lm(mpg ~ disp, mtcars)

```
Call:
lm(formula = y ~ x, data = mtcars)

Residuals:
     Min       1Q   Median       3Q      Max
-4.8922 -2.2022 -0.9631  1.6272  7.2305

Coefficients:
             Estimate Std. Error t value
(Intercept) 29.599855   1.229720  24.070
x           -0.041215   0.004712  -8.747
            Pr(>|t|)
(Intercept)  < 2e-16 ***
x           9.38e-10 ***
---
Signif. codes:
  0 '***' 0.001 '**' 0.01 '*' 0.05 '.'
  0.1 ' ' 1

Residual standard error: 3.251 on 30 degrees of freedom
Multiple R-squared:  0.7183,    Adjusted R-squared:  0.709
F-statistic: 76.51 on 1 and 30 DF,  p-value: 9.38e-10
```

c. Use R to draw the regression line on top of the scatter plot presented above. Then, calculate the sum of squared errors (SSE). Hint: You have to draw the scatter plot first, in order to draw the regression line. **(4 points)**

```
plot(x, y, main = "Main title",
     xlab = "X axis title", ylab = "Y axis title",
     pch = 19, frame = FALSE)
abline(model, col = "red")
```



SSE = : sum((fitted(model) - x)^2)

Result: 1935942

d. Use the first four rows of mtcars in order to calculate by hand the first and second iteration of the gradient descent algorithm. You will use *disp* and *mpg*, **min-max normalized**, as the predictor and response variable, respectively.

Hint: $a, b,$ and y_p are the intercept, slope, and vector of predictions of *mpg*, respectively. Use learning rate equal to `0.001`. **(10 points)**

1. Provide the min-max normalized values of *disp* and *mpg* in the following table. **(2 points)**

	disp	*disp normalized*	*mpg*	*mpg normalized*
1.	160	0.22175106	21.000	0.4510638
2.	160	0.22175106	21.000	0.4510638
3.	108	0.09204290	22.800	0.5276596
4.	258	0.46620105	21.400	0.4680851

2. Provide the results and calculations of the first iteration in the following spaces. **(3 points)**

	a	b	y_p	$SSE = \sum(\frac{1}{2}(y - y_p)^2)$ **
1.	0.650	0.350	0.4941382 0.4941382 0.4098279 0.6530307	7.233736

** $\frac{1}{2}$ *is only used for facilitating computations when you will use the derivatives.*

Calculations:

```
gradientDesc <- function(x, y, learn_rate, n, max_iter) {
  a <- 0.650
  b <- 0.350
  yhat <- a * x + b
  SSE <- sum((y - yhat) ^ 2)
  yhat <- a * x + b
  SSE_new <- sum((y - yhat) ^ 2)
  a_new <- a - learn_rate * ((1 / n) * (sum((yhat - y) * x)))
  b_new <- b - learn_rate * ((1 / n) * (sum(yhat - y)))
  a <- a_new
  b <- b_new
  yhat <- a * x + b
  SSE_new <- sum((y - yhat) ^ 2)
}

disp_norm <- norm_scale$disp
mpg_norm <- norm_scale$mpg
gradientDesc(disp_norm, mpg_norm, 0.001, 4, 2)
```

3. Provide the results and calculations of the second iteration in the following spaces. **(5 points)**

	a	b	y_p	$SSE = \sum(\frac{1}{2}(y - y_p)^2)$ **
1.	0.648372	0.3484285	0.4922057 0.4922057 0.4081066 0.6507002	7.192943
	** $\frac{1}{2}$ *is only used for facilitating computations when you will use the derivatives.*			

Calculations:

```
gradientDesc <- function(x, y, learn_rate, n, max_iter) {
  a <- 0.650
  b <- 0.350
  yhat <- a * x + b
  SSE <- sum((y - yhat) ^ 2)
  yhat <- a * x + b
  SSE_new <- sum((y - yhat) ^ 2)
  a_new <- a - learn_rate * ((1 / n) * (sum((yhat - y) * x)))
  b_new <- b - learn_rate * ((1 / n) * (sum(yhat - y)))
  a <- a_new
  b <- b_new
  yhat <- a * x + b
  SSE_new <- sum((y - yhat) ^ 2)
}

disp_norm <- norm_scale$disp
mpg_norm <- norm_scale$mpg
gradientDesc(disp_norm, mpg_norm, 0.001, 4, 2)
```

Topic 4: Confusion Matrix & ROC curve

A diagnostic test is performed on a number of individuals that aims to identify whether these individuals have been infected or not by COVID-19. The following table quotes the results of the test and the true state of each individual. The result of each test is the probability of each tested individual to be infected. **(20 points)**

	result	true state
1.	0.51	N
2.	0.67	N
3.	0.88	I
4.	0.34	N
5.	0.22	N
6.	0.01	N
7.	0.71	I
8.	0.23	N

Infected (I), Non-infected (N)

a. Set the threshold to 0.50 and build the corresponding confusion matrix. An individual whose result is above or equal to the threshold is classified as Infected. **(8 points)**

Predicted \ Reference	Infected	Non-infected
Infected	2	0
Non-infected	2	4

b. Calculate the sensitivity and specificity of the test based on the confusion matrix of the previous topic. **(4 points)**

Sensitivity : 1.0000

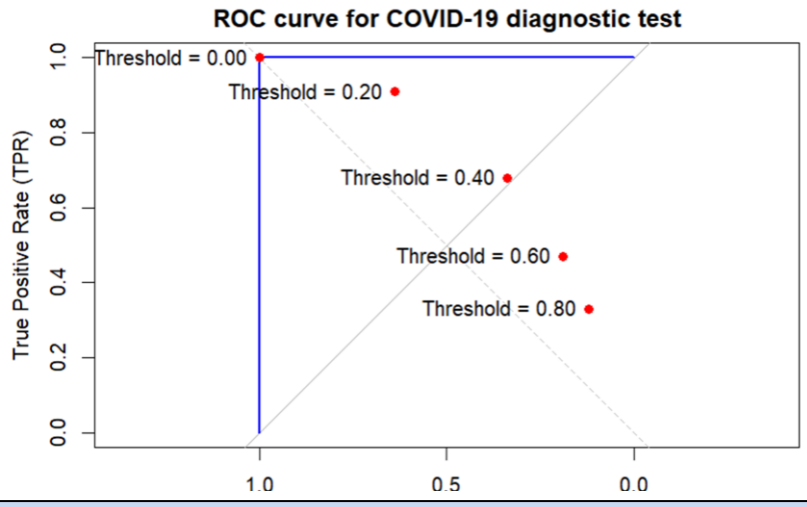
Specificity : 0.5000

c. Draw the ROC curve of the following table that displays the True Positive Rates (TPR) and False Positive Rates (FPR) for a COVID-19 diagnostic test. **(2 points)**

Threshold	TPR	FPR
0.80	0.33	0.12
0.60	0.47	0.19
0.40	0.68	0.34
0.20	0.91	0.64
0.00	1.00	1.00

Curve:

```
library(pROC)
actual <- c("N", "N", "I", "N", "N", "N", "I", "N")
roc_curve <- roc(actual, result)
plot(roc_curve, col = "blue", lwd = 2, main = "ROC curve for COVID-19 diagnostic test",
    xlab = "False Positive Rate (FPR)", ylab = "True Positive Rate (TPR)")
abline(a = 0, b = 1, lty = 2, col = "gray")
points(x = 0.12, y = 0.33, col = "red", pch = 19)
points(x = 0.19, y = 0.47, col = "red", pch = 19)
points(x = 0.34, y = 0.68, col = "red", pch = 19)
points(x = 0.64, y = 0.91, col = "red", pch = 19)
points(x = 1, y = 1, col = "red", pch = 19)
text(x = 0.12, y = 0.33, labels = "Threshold = 0.80", pos = 2)
text(x = 0.19, y = 0.47, labels = "Threshold = 0.60", pos = 2)
text(x = 0.34, y = 0.68, labels = "Threshold = 0.40", pos = 2)
text(x = 0.64, y = 0.91, labels = "Threshold = 0.20", pos = 2)
text(x = 1, y = 1, labels = "Threshold = 0.00", pos = 2)
```



d. What is the trade-off between TPR and FPR by relaxing the threshold in the previous topic? **(2 point)**

The trade-off between TPR and FPR by relaxing the threshold is that TPR will increase while FPR will also increase. By setting a lower threshold, the test will classify more individuals as Infected, increasing the TPR. However, this also means that some Non-infected individuals will be incorrectly classified as Infected, increasing the FPR.

e. Given an AUC = 0.89, what is the probability p that the diagnostic test used, ranks a random positive case higher than a random negative case? **(2 points)**

p = 0.78
Code: auc <- 0.89
p <- (auc - 0.5) / (1 - 0.5)
p

f. Assume that the total number of individuals tested is 200. Assume, also, that the number of infected and non-infected individuals is 40 and 160, respectively. What will be the number of these individuals that are expected to be classified as Infected, given a random classifier biased to the Non-infected class with probability equal to 0.60? **(2 points)**

```
total <- 200
infected <- 40
non_infected <- 160
prob <- 0.6
predicted <- ifelse(runif(total) < prob, "N", "I")
expected_infected <- sum(predicted == "I") * (infected / total)
expected_infected
```

Truly Infected individuals classified as Infected = 16 (round of 16.2 which was the result, it varies from run to run)
Truly Non-infected individuals classified as Infected = 24

Topic 5: Outlier Detection

You will work on this topic using the fastfood dataset in `library(openintro)` which contains 515 observations on nutritional facts regarding the products of 8 different fastfood chains. You are requested to provide your R code and the result of its execution in every answer box. **(20 points)**

a. Print a summary of the dataset and indicate the minimum and maximum values of attribute "`calcium`" (**2 points**)

Answer:

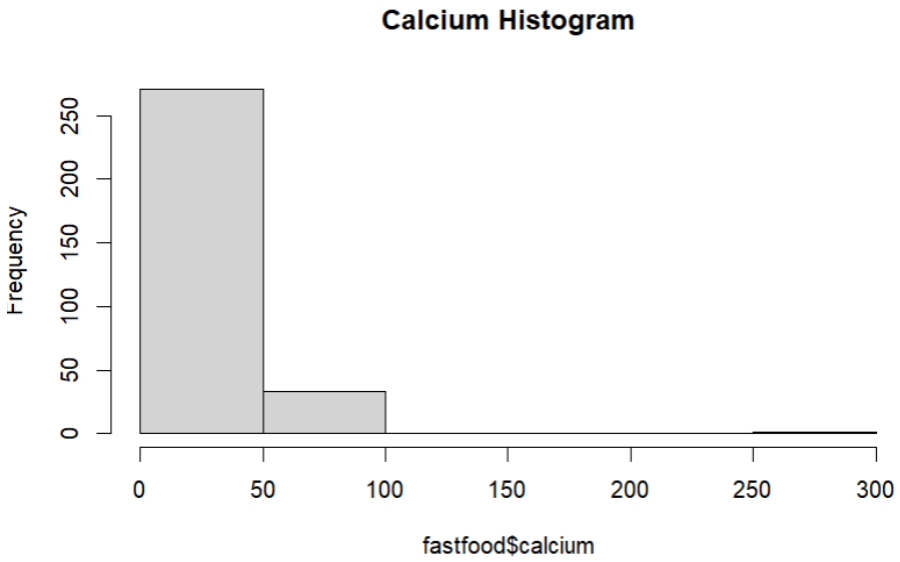
```
library(openintro)
data(fastfood)
summary(fastfood)
```

```
 restaurant            item              calories        cal_fat         total_fat
Length:515         Length:515         Min.   :  20.0   Min.   :   0.0   Min.   :  0.00
Class :character   Class :character   1st Qu.: 330.0   1st Qu.: 120.0   1st Qu.: 14.00
Mode  :character   Mode  :character   Median : 490.0   Median : 210.0   Median : 23.00
                                      Mean   : 530.9   Mean   : 238.8   Mean   : 26.59
                                      3rd Qu.: 690.0   3rd Qu.: 310.0   3rd Qu.: 35.00
                                      Max.   :2430.0   Max.   :1270.0   Max.   :141.00

   sat_fat         trans_fat      cholesterol        sodium       total_carb
Min.   : 0.000   Min.   :0.000   Min.   :  0.00   Min.   :  15   Min.   :  0.00
1st Qu.: 4.000   1st Qu.:0.000   1st Qu.: 35.00   1st Qu.: 800   1st Qu.: 28.50
Median : 7.000   Median :0.000   Median : 60.00   Median :1110   Median : 44.00
Mean   : 8.153   Mean   :0.465   Mean   : 72.46   Mean   :1247   Mean   : 45.66
3rd Qu.:11.000   3rd Qu.:1.000   3rd Qu.: 95.00   3rd Qu.:1550   3rd Qu.: 57.00
Max.   :47.000   Max.   :8.000   Max.   :805.00   Max.   :6080   Max.   :156.00

    fiber            sugar           protein          vit_a            vit_c
Min.   : 0.000   Min.   : 0.000   Min.   :  1.00   Min.   :  0.00   Min.   :  0.00
1st Qu.: 2.000   1st Qu.: 3.000   1st Qu.: 16.00   1st Qu.:  4.00   1st Qu.:  4.00
Median : 3.000   Median : 6.000   Median : 24.50   Median : 10.00   Median : 10.00
Mean   : 4.137   Mean   : 7.262   Mean   : 27.89   Mean   : 18.86   Mean   : 20.17
3rd Qu.: 5.000   3rd Qu.: 9.000   3rd Qu.: 36.00   3rd Qu.: 20.00   3rd Qu.: 30.00
Max.   :17.000   Max.   :87.000   Max.   :186.00   Max.   :180.00   Max.   :400.00
NA's   :12                        NA's   :1        NA's   :214      NA's   :210
   calcium           salad
Min.   :  0.00   Length:515
1st Qu.:  8.00   Class :character
Median : 20.00   Mode  :character
Mean   : 24.85
3rd Qu.: 30.00
Max.   :290.00
NA's   :210
```

b. Create a histogram for the attribute "`calcium`". Do you detect any potential outliers? (**2 points**)

Answer:

hist(fastfood$calcium, main="Calcium Histogram")

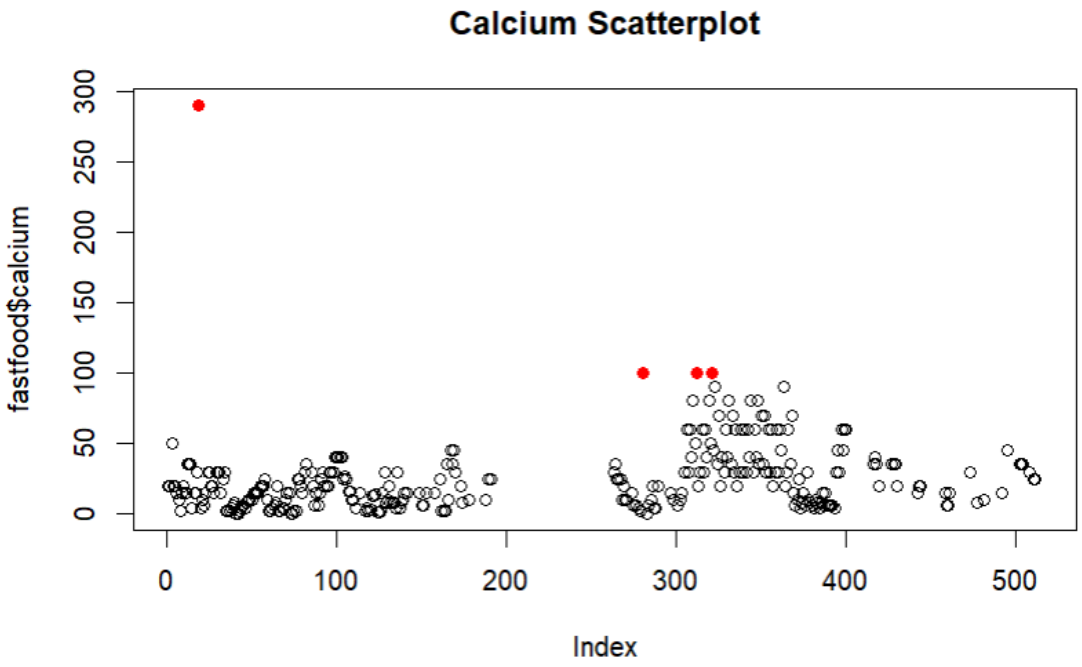


Minimum value of attribute "calcium" is 0 and maximum value is 400.

c. Now, create a scatterplot for the attribute "`calcium`". On the scatterplot, indicate any points you consider to be outliers (for example, by drawing arrowed lines pointing to the outliers, or by drawing small circles around the outliers). (**3 points**)

Answer:

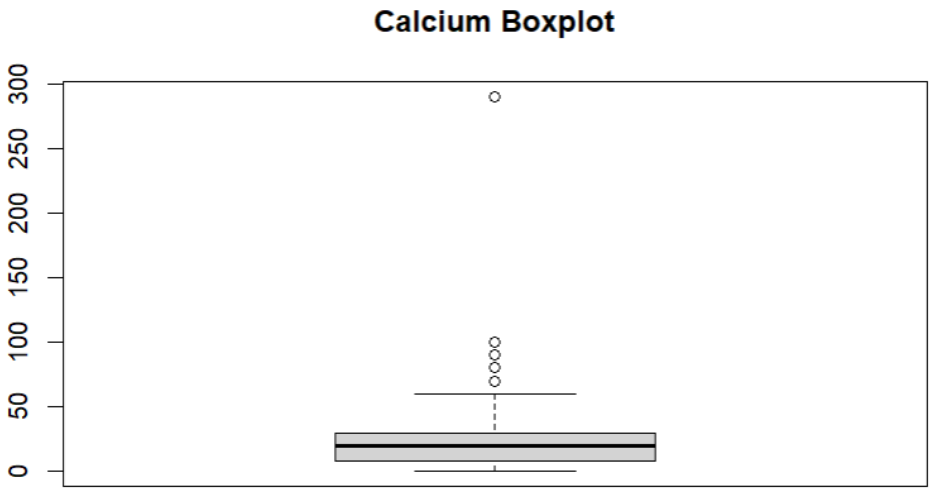
```
plot(fastfood$calcium, main = "Calcium Scatterplot")
outliers <- (replace(fastfood$calcium, is.na(fastfood$calcium), 0)) >= 100
points(which(outliers), fastfood$calcium[outliers], pch = 19, col = "red")
```



d. Finally, create a boxplot for the attribute "`calcium`". On the boxplot, indicate any points you consider to be outliers (for example, by drawing arrowed lines pointing to the outliers, or by drawing small circles around the outliers). (**3 points**)

Answer:

```
boxplot(fastfood$calcium, main = "Calcium Boxplot")
```



We can clearly see the outlier on the top-middle of the boxplot.

e. Identify all the outliers of the attribute "`calcium`", by performing a Grubbs test (use only the `grubbs.test` function). Hint: Start by identifying the first outlier, replace it in the data with NA and then proceed with the identification of the next one and so on. Insert your answers in the table provided. Do not forget to provide your R code as a separate file. (**10 points**)

Answer

Outlier	Index of outlier