Midterm 1

Louis Bensard

March 3, 2018

Theoretical Portion

Problem 1

The leverage of the i^{th} data point measures the impact of the measured observation y_i on the fitted observation $\hat{y_i}$. The further away a data point is from the mean, the higher its leverage. In other words, if I move my i^{th} data point x_i up and down, the leverage is how this change is going to impact the fitted observation. More specifically, the leverage is the ratio of the change $\Delta\hat{y_i}$ in $\hat{y_i}$ over the change Δy_i in y_i:

$$\frac{\Delta\hat{y_i}}{\Delta y_i}$$

High leverage is not necessarily bad. Indeed, if a point x_i has high leverage, but follows the general trend of the data, it will only strengthen the regression model, which is good. However, if a data point x_i is far away from the mean (high leverage) and does not follow the general trend of the data, then it will have "bad" leverage. That is because a high leverage point have a big influence on the model, so this single high leverage point will make the whole model shift toward itself despite the rest of the data following another trend.

Problem 2

The sample has size $n = 643$. The "testosterone group" (group 1) has size $n_1 = 210$ and the "no testosterone group" (group 2) has size $n_2 = 433$. Let μ_1 and μ_2 be respectively the population mean of the group 1 and group 2. Let's perform the following hypothesis test:

$$\begin{cases} H_0 : \mu_1 = \mu_2 \\ H_a : \mu_1 > \mu_2 \end{cases}$$

If $\bar{x_1}$ and $\bar{x_2}$ are the sample means of group 1 and 2, s_1^2 and s_2^2 are the sample variances of group 1 and 2, then, we have

$$t^* = \frac{\bar{x_1} - \bar{x_2}}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} = \frac{11.2 - 6.6}{\sqrt{\frac{6.1^2}{210} + \frac{4.1^2}{433}}} = 9.897$$

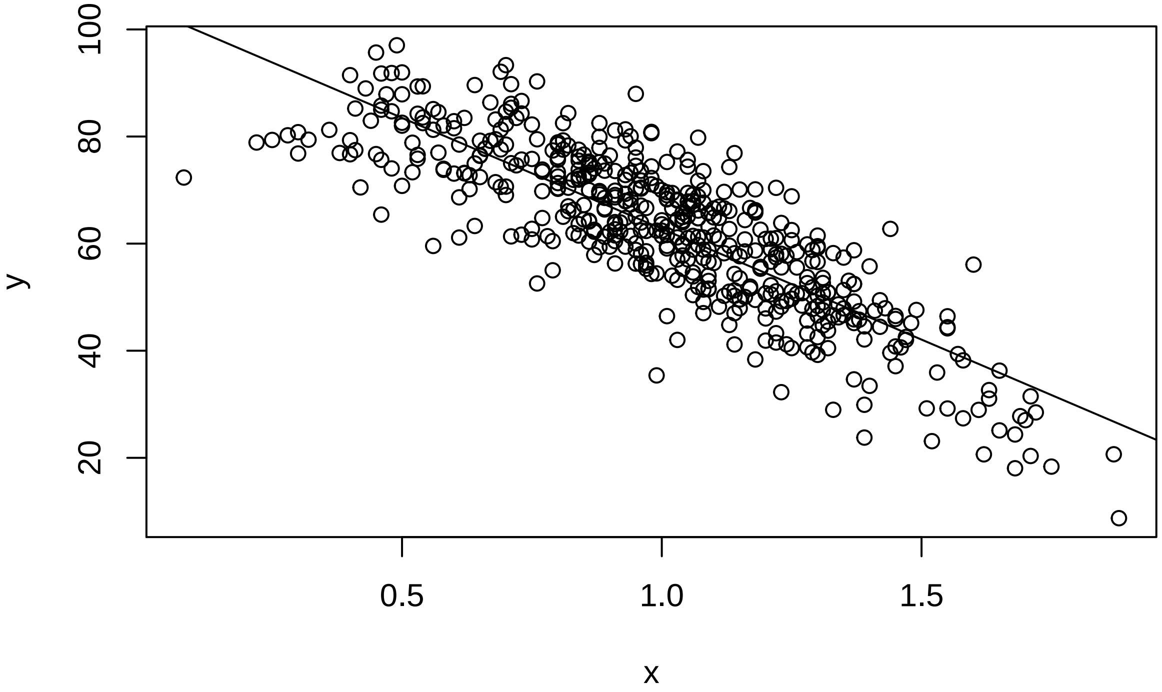
t^* follows a t distribution with $min(209, 433) = 209$ degrees of freedom, therefore, we get $p - value \simeq 0 < 0.05$. As a result, we reject the null hypothesis and thus we conclude that the supplement will help males lose weight.

Applied Portion

Problem 3

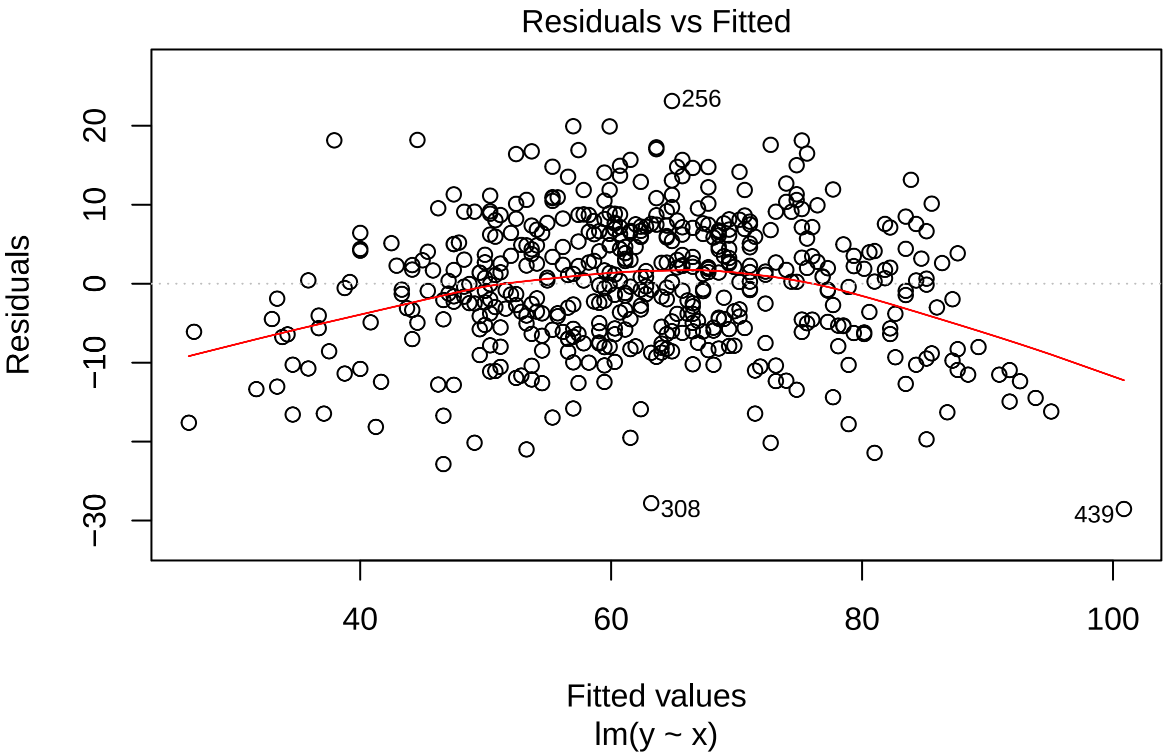
(a)

```r
data = read.csv("C:/Users/Louis/Documents/UPMC/M1/Spring 2018/MATH 535/midterm1/cable.csv")

x = data$calltime
y = data$satisfaction

plot(x,y)
model1 = lm(y~x)
abline(model1)
```



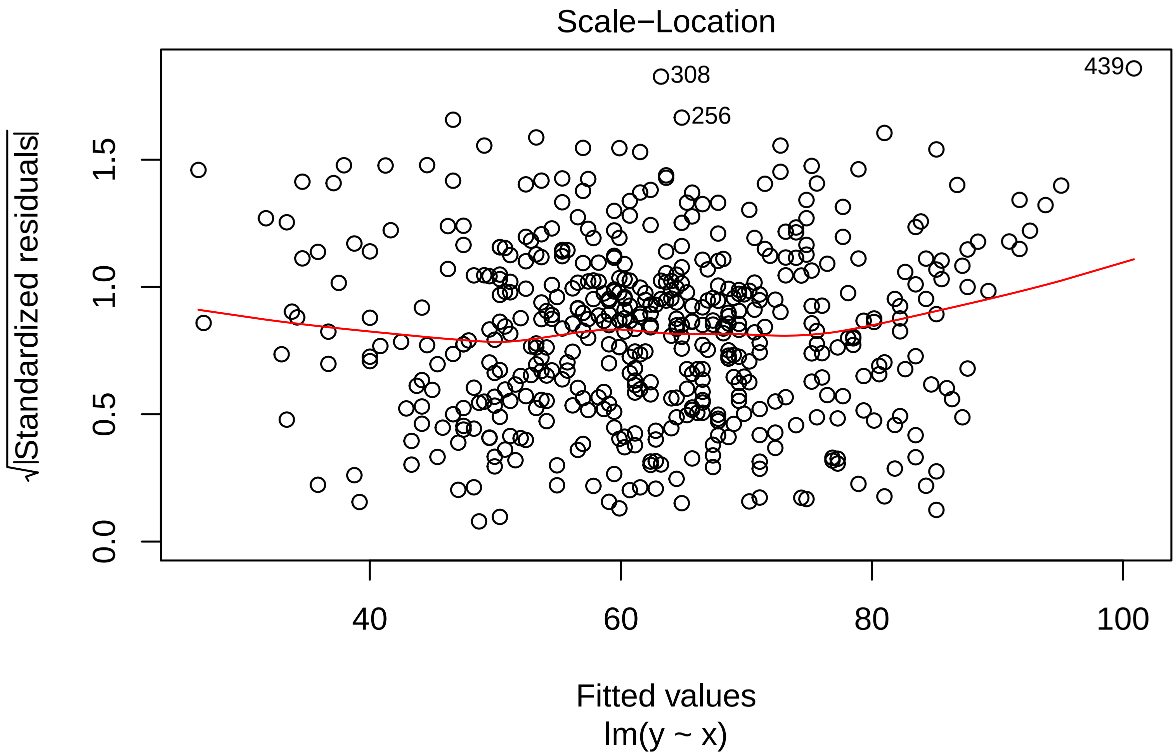
The abline over the plot looks good because it is going through most of the data. It is not enough to judge a model of course but at least we know that our current model is relevant.

```r
plot(model1, which=1)
```



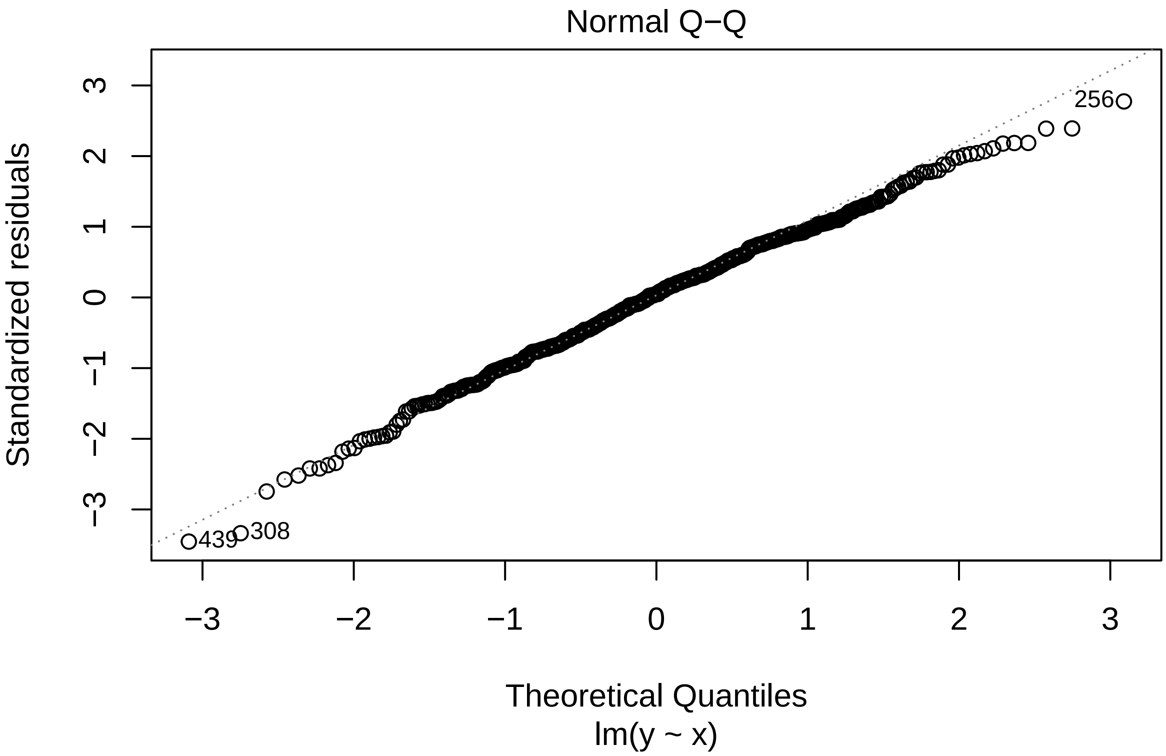
The Residuals vs Fitted plot looks good enough. Indeed, a good Residual vs Fitted plot should display points randomly spread along the 0-horizontal axis. If this is the case, we consider the first assumption valid (i.e. $E(y|x) = x\beta$). Here, even though we can observe a slight curve, the errors seems randomly enough speard around 0. Therefore, we consider the first assumption valid enough and thus we can keep the model analysis going.

```r
plot(model1, which=3)
```



The Scale-Location plot looks good as well. Indeed, a good Scale-Loation plot should display points randomly spread along the 1-horizontal line. If this is the case, we consider the Constant Variance assumption valid. Here, due to the slight parabola in the first plot, we can observe a slight "w" shape. But again, the errors seems randomly enough speard around 1. Therefore, we consider the Constant Variance assumption valid enough and thus we can keep the model analysis going.

```r
plot(model1, which=2)
```



The Normal Q-Q plot looks ok. Indeed, if all the points are on a straight line, this means that the Normality assumption is valid. Therefore, even though the tails are a little off, since almost all the points are aligned here, I would say that the Normality assumption is valid enough.

```
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -28.5323  -5.6875   0.3991   6.2116  23.1199
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  104.185      1.278   81.52   <2e-16 ***
## x            -41.405      1.216  -34.04   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.343 on 498 degrees of freedom
## Multiple R-squared:  0.6994, Adjusted R-squared:  0.6988
## F-statistic:  1159 on 1 and 498 DF,  p-value: < 2.2e-16
```

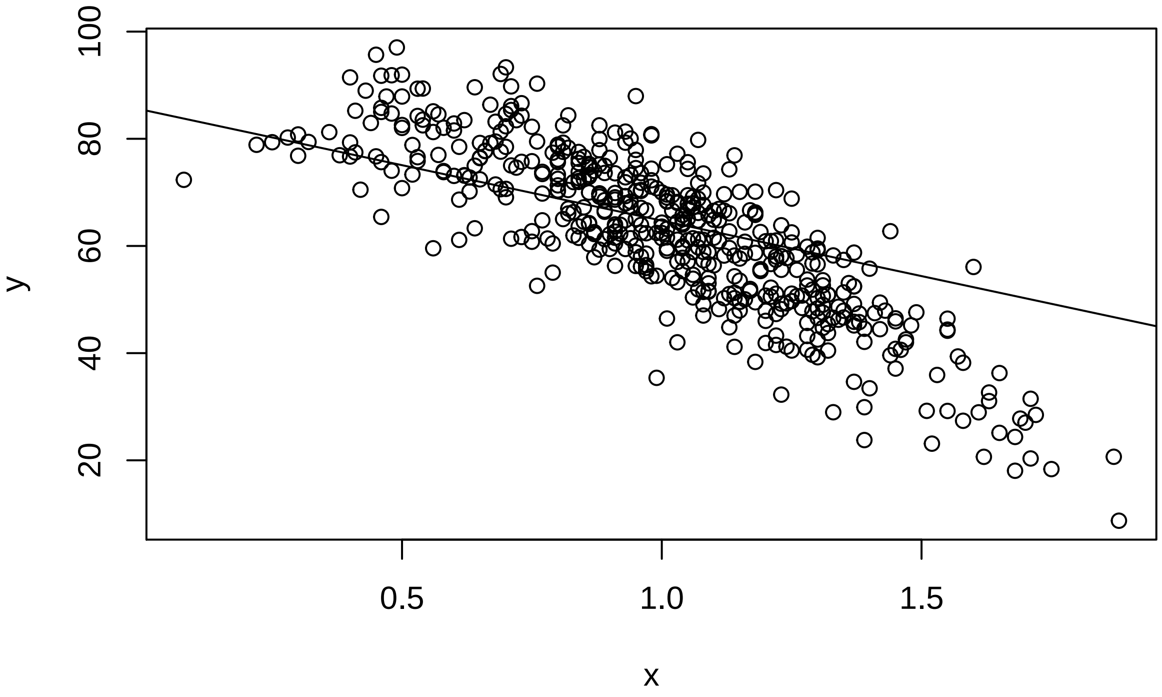
The r^2 value of 70% is not bad either. Indeed, the higher the r^2 value, the better the model fit the data. We cannot judge a model only through the r^2 value, but this 70% is an decent value and does not rule out our model.

Conclusion: For all the reasons listed aboven I would say that the current linear model is a decent model for describing the customer satisfaction base on time spent waiting in queue. Indeed, none of the above can rule out the current linear model. However, because of the curvature of the Residual vs Fitted plot, the Scale-Location plot, the tails of the QQ plot and the only decent r^2 value, this model can probably be improved.

(b)

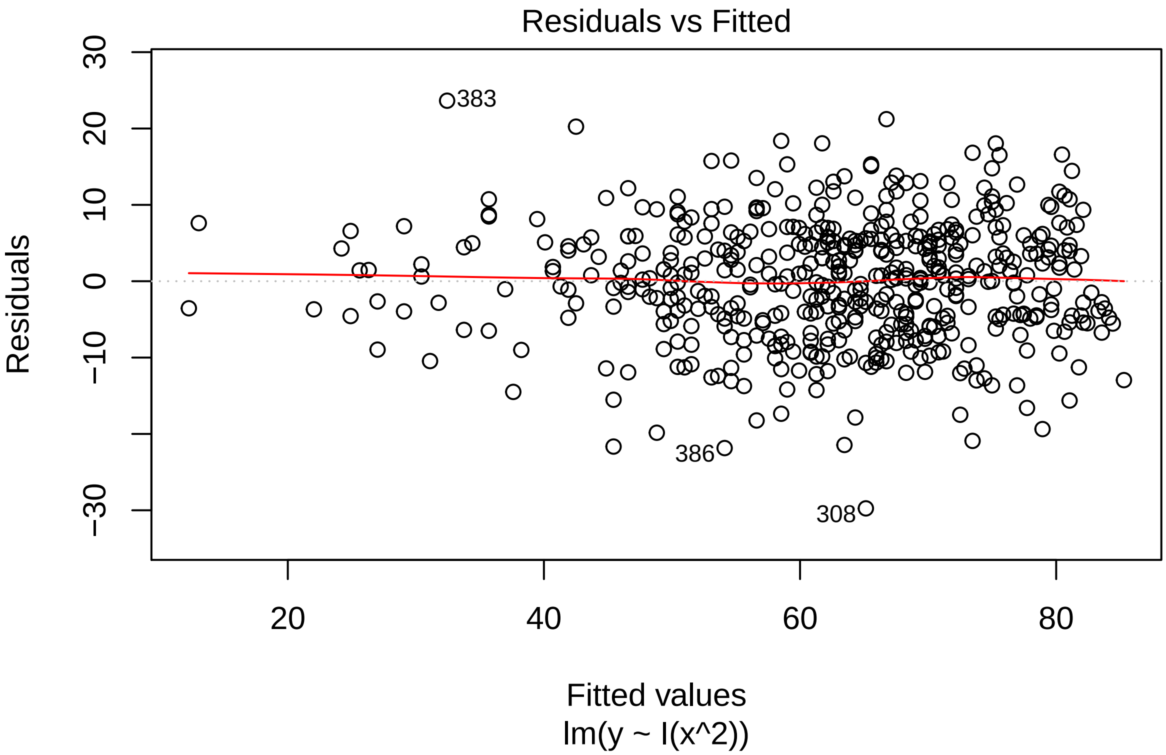
The slight parabola on the previous Residual vs Fitted plot lead me to think that we are possibly trying to fit a quadratic data inside a linear model. Moreover, if we look at the real life situation where the data is coming from, it would make more sense that this data has a quadratic shape. Indeed, when the calltime is decent (say less than 1h) I would think that customers would be satisfied accordingly (So sort of linear for < 1h). But as soon as the calltime gets larger than 1h for instance, customers would probably be very unsatisfied and will overly rate their unsatisfaction by giving a very low satisfaction percentage (not linear anymore for >1h). I would never base a whole model on this latter analysis but it goes along what the plots in part (a) depicted, therefore a quadratic model of the form $\hat{y} = \beta_1 x^2 + \beta_0$ is worth being considered and further analysed, as in the following:

```r
plot(x,y)
model2 = lm(y~I(x^2))
abline(model2)
```



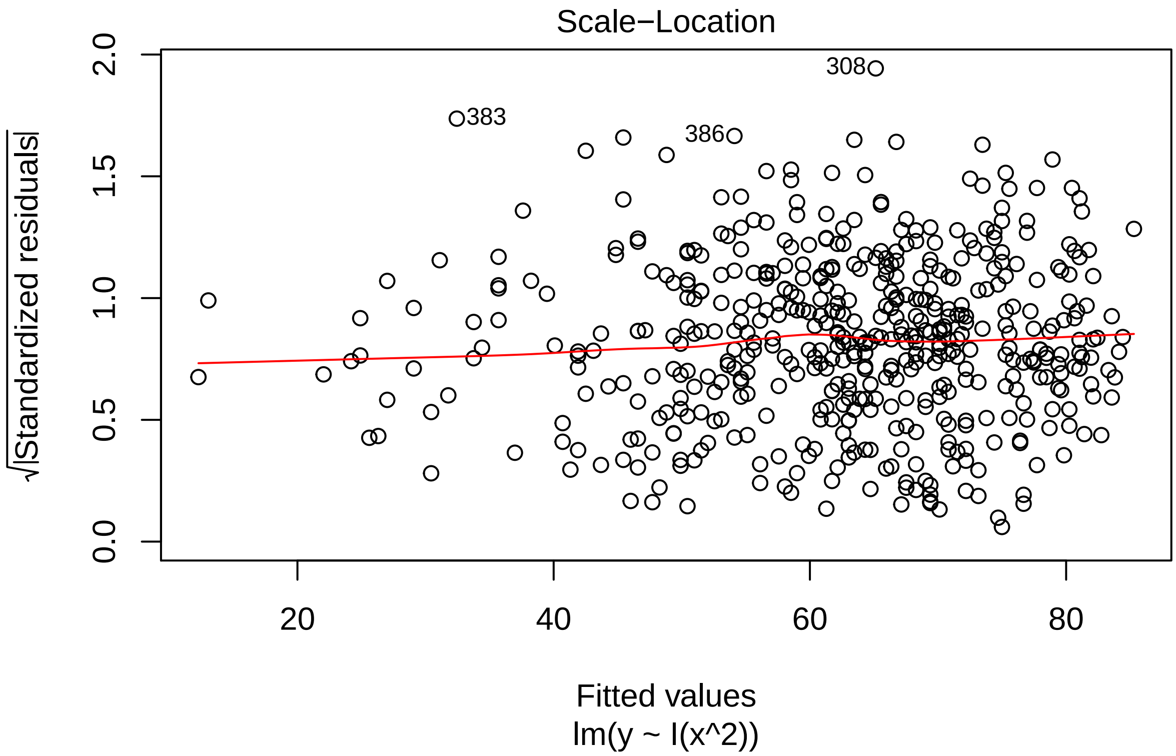
The abline over the plot looks good because it is going through most of the data. It is not enough to judge a model of course but at least we know that our current model is relevant.

```
plot(model2, which=1)
```



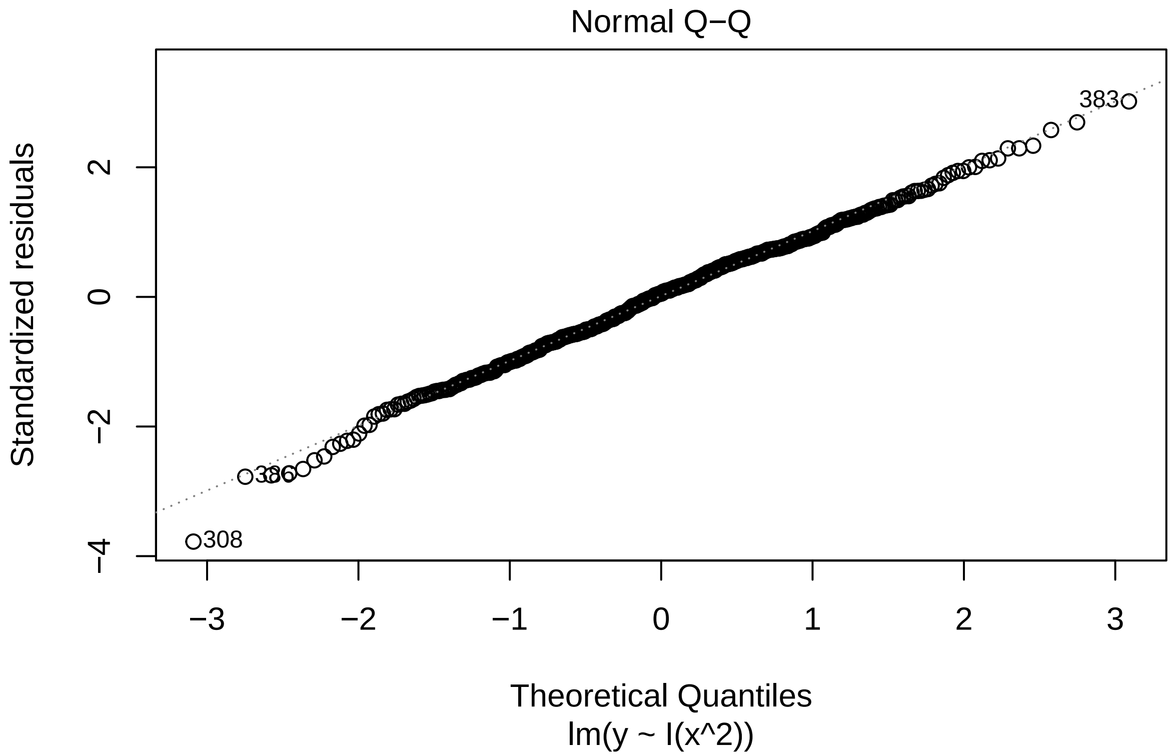
This time, the Residuals vs Fitted plot looks really good. No curvature, the errors are spread equally around the 0-horizontal axis. Therefore the first assumption is strongly valid.

```
plot(model2, which=3)
```



Similarly, the Scale-Location plot looks really good here, it displays errors randomly spread around the 1-horizontal axis. Therefore, the Constant Variance assumption is strongly valid.

```
plot(model2, which=2)
```



The Normal Q-Q plot is here more aligned that in the first model, the tails got aligned with the other points, this is very good. Therefore the Normality assumption is here strongly valid.

```r
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ I(x^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.750  -5.200   0.406   5.432  23.640
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  85.4251     0.7137  119.69   <2e-16 ***
## I(x^2)      -20.6974     0.5621  -36.82   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.887 on 498 degrees of freedom
## Multiple R-squared:  0.7313, Adjusted R-squared:  0.7308
## F-statistic:  1356 on 1 and 498 DF,  p-value: < 2.2e-16
```

The r^2 value went up to 73% here. I was excpecting a higher r^2 value but it is still a good value that is higher than the previous 70%.

Conclusion: The quadratic model was worth being considered because it improved every plots that looked only decent in part (a) and increased r^2. Therefore, model 2 is more appropriate for statistical inference here.

A quick run of AIC and BIC on model 1 and model 2 will confirm this statement:

```
AIC(model1)
```

```
## [1] 3544.394
```

```
AIC(model2)
```

```
## [1] 3488.189
```

```
library(stats4)
BIC(model1)
```

```
## [1] 3557.037
```

```
BIC(model2)
```

```
## [1] 3500.833
```

$AIC(model1) = 3544.39 > 3488.19 = AIC(model2)$ and $BIC(model1) = 3557.04 > 3500.83 = AIC(model2)$. Therefore, the Akaike and the Bayesian Information Criterion are both minimized by model 2. Thus model 2 is indeed more appropriate.

(c)

```
predict(model2,newdata=data.frame(x=2), interval="confidence")
```

```
##        fit        lwr    upr
## 1 2.635497 -0.6374061 5.9084
```

Therefore a 95% confidence interval for the average satisfaction index score of individuals who have to wait 2 hours in the queue is [0,5.9084] (no negative percentages).

(d)

I do have reservations about using your confidence interval from part (c):

```
summary(data)
```

```
##     calltime       satisfaction
##  Min.   :0.0800   Min.   : 8.73
##  1st Qu.:0.8175   1st Qu.:52.40
##  Median :1.0100   Median :63.75
##  Mean   :1.0048   Mean   :62.58
##  3rd Qu.:1.2200   3rd Qu.:73.61
##  Max.   :1.8800   Max.   :97.04
```

In part (c), we are building a confidence interval for calltime = 2 hours, but the summary above shows that we don't even have any data with a calltime more than 1.88 hours. Moreover, $Q3 + 1.5 * IQR = 1.22 + 1.5 * (1.22 - 0.818) = 1.823$, therefore, every point above 1.823 is considered an outlier of the data. As a result, in part (c), we are building a confidence interval at $x = 2$ from a model that descibes a trend of data whose $x = 2$ is an outlier from. So, I would not have any reservation if the confidence interval was for a calltime of 1 hour for instance, but for 2 hours, it seems that the actual accuracy of that interval is limited.

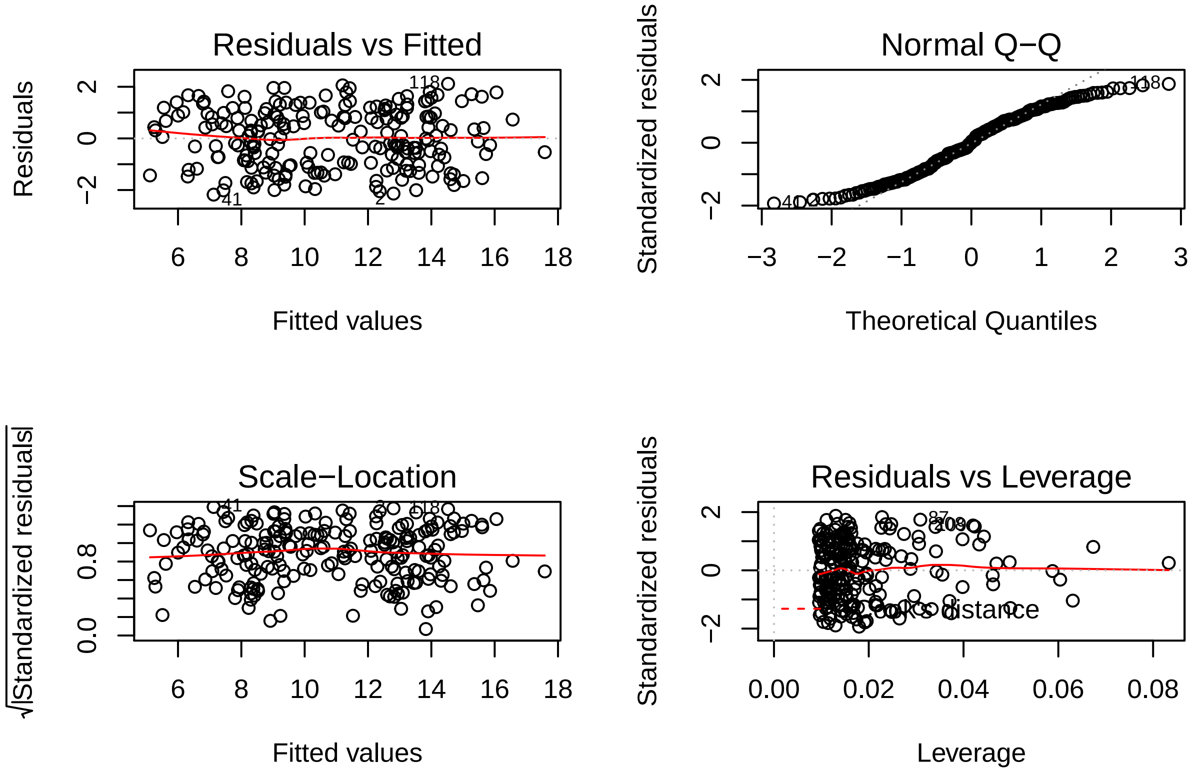
Problem 4

(a)

```r
data = read.csv("C:/Users/Louis/Documents/UPMC/M1/Spring 2018/MATH 535/midterm1/Albumin.csv")

x_ope = data$Operation.Length
x_diab = data$Diabetes
x_bmi = data$BMI
y = data$Albumin
```

The plan here is to start by analysing two models. The first one will be the model with all three predictors (Operation length, Diabetes and BMI). The second model will only contain the predictors Operation length and Diabetes. We will compare both models so we can have a idea of how significant BMI is.

```r
model1 = lm(y~x_ope+x_diab+x_bmi)

par(mfrow=c(2,2))
plot(model1)
```



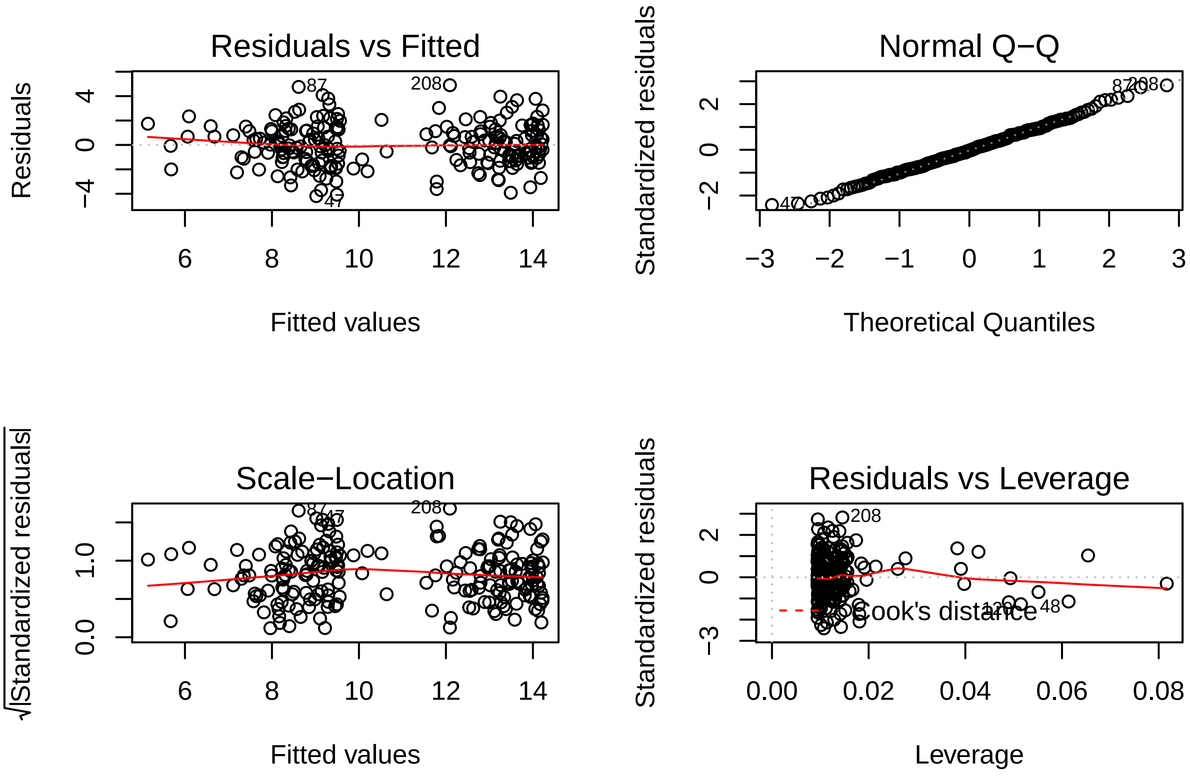
- Residuals vs Fitted plot looks good. Points randomly aligned around 0-horizontal line. 1st assumption valid.
- Scale-Location plot looks good. Points randomly aligned around 1-horizontal line. 2nd assumption valid.
- Normal Q-Q plot does not look good. Normality assumption clearly broken.

```
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x_ope + x_diab + x_bmi)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.17850 -0.95855  0.02268  0.92349  2.11312
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 23.05098    0.52873   43.60   <2e-16 ***
## x_ope       -0.66924    0.04714  -14.20   <2e-16 ***
## x_diab      -4.88531    0.15681  -31.16   <2e-16 ***
## x_bmi       -0.27755    0.01645  -16.87   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.137 on 208 degrees of freedom
## Multiple R-squared:  0.8628, Adjusted R-squared:  0.8608
## F-statistic:   436 on 3 and 208 DF,  p-value: < 2.2e-16
```

High r^2 value of 86% is very good, the model fits the data well.

```
model2 = lm(y~x_ope+x_diab)

par(mfrow=c(2,2))
plot(model2)
```



- Residuals vs Fitted plot looks good. Points randomly aligned around 0-horizontal line. 1st assumption valid.
- Scale-Location plot looks good. Points randomly aligned around 1-horizontal line. 2nd assumption valid.
- Normal Q-Q plot looks good. Normality assumption is valid here.

```
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x_ope + x_diab)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.1896 -1.2386 -0.0555  1.1555  4.8924
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 14.54661    0.24518  59.330  < 2e-16 ***
## x_ope       -0.58479    0.07197  -8.126 3.85e-14 ***
## x_diab      -4.68991    0.24010 -19.533  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.746 on 209 degrees of freedom
## Multiple R-squared:  0.675,  Adjusted R-squared:  0.6719
## F-statistic: 217.1 on 2 and 209 DF,  p-value: < 2.2e-16
```

The r^2 value dropped to 67%. That is a big drop just by removing BMI from the equation. This hints me to think that BMI might be a statistically significant predictor of Albumin.

Even though the model without BMI gained in Normality, the big 19% drop of r^2 lead me to think that the model would still be better with BMI inside and that BMI is indeed significant. To make sure, I am going to run AIC and BIC on model 1 and model 2 to see which model minimizes the loss of information.

```r
AIC(model1)
```

```
## [1] 662.1476
```

```r
AIC(model2)
```

```
## [1] 842.9616
```

```r
library(stats4)
BIC(model1)
```

```
## [1] 678.9305
```

```r
BIC(model2)
```

```
## [1] 856.3879
```

$AIC(model1) = 662.15 < 842.96 = AIC(model2)$ and $BIC(model1) = 678.93 < 856.39 = BIC(model2)$. Therefore, the Akaike and the Bayesian Information Criterion are both minimized by model 1.

Conclusion: Both Akaike and Bayesian Information Criterion are minimized by model 1 that contains the BMI predictor, therefore, according to these criteria, model 1 is better than model 2. Moreover, The big drop in r^2 by removing the BMI predictor (despite the gain in Normality) is another indicator that BMI is a non-negligible component of the model. As a result, I conclude that the Body Mass Index is a statistically significant predictor of Albumin in the context of the other covariates.

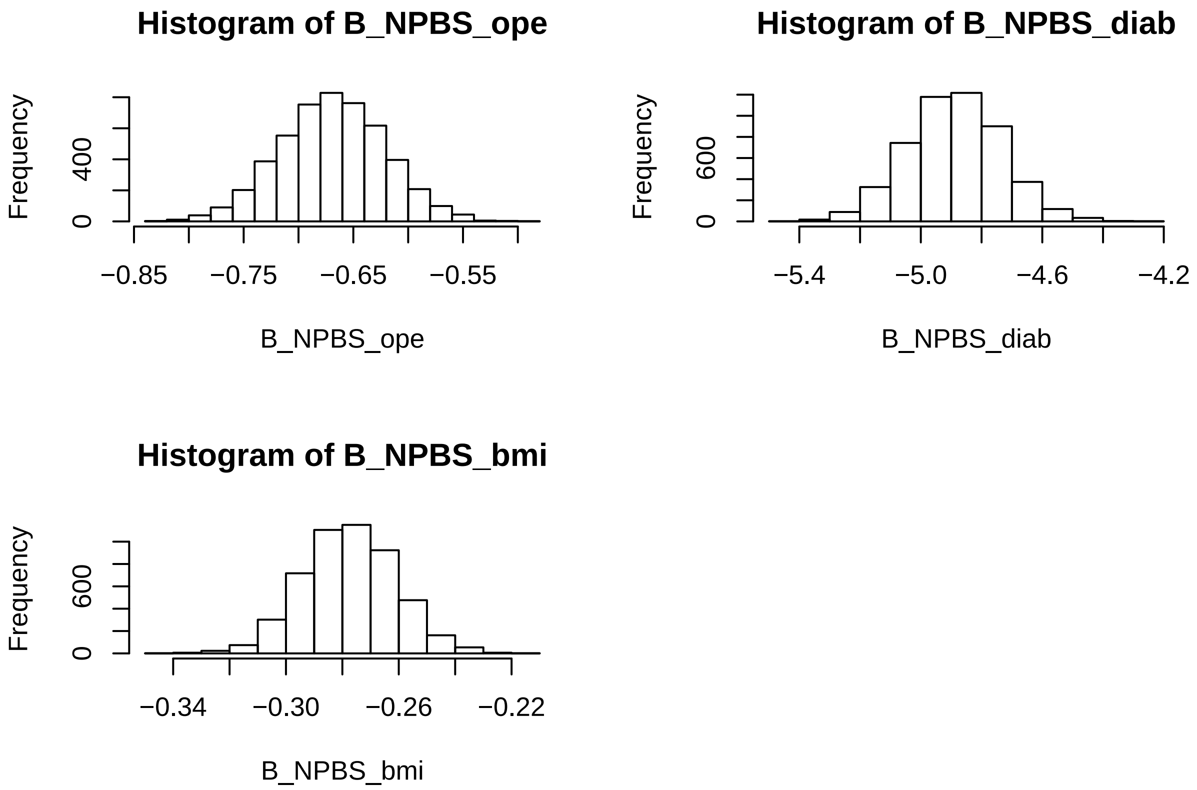
Bonus:

The Normality assumption can be fixed by boostrapping all three predictors, as follows:

```r
n = length(x_ope)
X = data.frame(x_ope,x_diab, x_bmi)
residuals = as.numeric(model1$res)

B_NPBS_ope = rep(0,5000)
B_NPBS_diab = rep(0,5000)
B_NPBS_bmi = rep(0,5000)

    for(i in 1:5000){

        new_x = X[sample(1:n,n,replace=T),]
        fit_y = predict(model1, new_x)

        new_y = fit_y + sample(residuals, n, replace=T)

        B_NPBS_ope[i] = lm(new_y~new_x[,1]+new_x[,2]+new_x[,3])$coef[2]
        B_NPBS_diab[i] = lm(new_y~new_x[,1]+new_x[,2]+new_x[,3])$coef[3]
        B_NPBS_bmi[i] = lm(new_y~new_x[,1]+new_x[,2]+new_x[,3])$coef[4]
    }
```

Here are the histograms of the residuals after boostrapping:

```
par(mfrow=c(2,2))
hist(B_NPBS_ope)
hist(B_NPBS_diab)
hist(B_NPBS_bmi)
```



We can clearly see a normal distribution there, the boostrapping worked.