Exercise 5

Question 1

```
edu_data = read.table("Education5.txt", header=T)
clean_edu_data = edu_data[c('Education','Gender','Wage' )]
summary(clean_edu_data)
```

```
##    Education        Gender         Wage
##  Min.   : 5.00   female:198   Min.   :2047
##  1st Qu.:12.00   male  :299   1st Qu.:4679
##  Median :14.00                Median :5520
##  Mean   :14.26                Mean   :5463
##  3rd Qu.:16.00                3rd Qu.:6319
##  Max.   :22.00                Max.   :8454
```

```
mod <- lm(Wage ~ ., data=clean_edu_data)
summary(mod)
```

```
##
## Call:
## lm(formula = Wage ~ ., data = clean_edu_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -243.473  -76.073    0.354   73.126  280.275
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -569.784     23.100  -24.67   <2e-16 ***
## Education    397.975      1.543  258.00   <2e-16 ***
## Gendermale   597.904      9.173   65.18   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 100.1 on 494 degrees of freedom
## Multiple R-squared:  0.9931, Adjusted R-squared:  0.9931
## F-statistic: 3.544e+04 on 2 and 494 DF,  p-value: < 2.2e-16
```

1. Is there a relationship between the response and predictors?

We test the null hypothesis:

$$H_0 : \beta_1 = \beta_2 = 0$$

versus the alternative: H_a : at least one β_j is non-zero

We perform the hypothesis by computing the F-statistic:

```
l <- anova(mod)
TSS <- sum(l$`Sum Sq`[1:2])
RSS <- l$`Sum Sq`[3]
p <- 2
n <- nrow(clean_edu_data)
F_c <- ((TSS-RSS)/p)/(RSS/(n-p-1))
F_c
```

```
## [1] 35197.19
```

The F-statistic which is shown also on the summary of the multiple linear regresion above is 35197(35440) according to our calculation which is far larger than 1, so it provides a compelling evidence agains the null hypothesis H_0 (no relationship between the predictors and response). So, it means that there exist a relationship between predictors(Education and Gender) and response(Wage).

2. How well does the model fit the data?

Now we want to determine which variables are important for the model, because p=2 is small enough we can perform *variable selection* directly by trying out different models, each containing a different subset of the predictors.

We will consider only three models(we will omit the model with no variables): (1) a model containing only Education variable, (2) a model containing only Gender variable, (3) a model containing Gender and Education variable.

```
mod_1 <- lm(Wage ~ Education, data=clean_edu_data)
mod_2 <- lm(Wage ~ Gender, data=clean_edu_data)
mod_3 <- lm(Wage ~ ., data=clean_edu_data)

RSE_1 <- summary(mod_1)$sigma
RSE_2 <- summary(mod_2)$sigma
RSE_3 <- summary(mod_3)$sigma

R_sq_1 <- summary(mod_1)$r.squared
R_sq_2 <- summary(mod_2)$r.squared
R_sq_3 <- summary(mod_3)$r.squared
```

We can see that the **model (3)** which uses *Education* and *Gender* variable to predict *Wage* has the largest R^2 value so is the best one, $R^2 = 0.9930$ which is very close to 1 and indicates that the model explains a very large portion of the variance in the response variable.

Also if we look at the **RSE** we can clearly see that the model that uses Education and Gender to predict Wage is much more accurate than one that uses only Education (or only Gender).

So, the clear winner is model (3) that uses Education and Gender variable to predict Wage.

Question 2

We remove the variables: **vendor**, **model** and **ERP** which we cannot use to predict **PRP**:

Load the Computer data set from the file ComputerData.txt.

1. Remove the variable that you cannot use to predict performances.

```
computerData <- read.table('ComputerData.txt', header = T)
computerData <- computerData[c("MYCT","MMIN","MMAX","CACH","CGMIN","CHMAX","PRP")]
```

2. Try to build the best multiple linear regression model to predict performances, you can use all the variables.

Our implementation of *backward variable selection* without using **step** function:

```
#start with all the descriptors of computer performance in the lm
summary.lm <- summary(lm(PRP ~ ., data=computerData))
```

```
vars <- names(computerData)
vars <- vars[which(vars != "PRP")]

tol <- 0.01

#while there are still variables with p-values larger than tol
while(length(which(summary.lm$coefficients[,4] > tol)) != 0){

  #coefficients table
  coef <- summary.lm$coefficients

  #find the variable with the largest p-value
  var_with_max_p.value <- rownames(coef)[apply(coef,2,which.max)][4]

  #remove the variable with the larger value
  vars <- vars[which(vars != var_with_max_p.value)]

  #computer the model with the updated variables
  lm2 <- lm(computerData$PRP~., data = computerData[vars])
  summary.lm <- summary(lm2)
}
summary.lm
```

```
##
## Call:
## lm(formula = computerData$PRP ~ ., data = computerData[vars])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -193.37  -24.95    5.76   26.64  389.66
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.608e+01  8.007e+00  -7.003 3.59e-11 ***
## MYCT         4.911e-02  1.746e-02   2.813   0.0054 **
## MMIN         1.518e-02  1.788e-03   8.490 4.34e-15 ***
## MMAX         5.562e-03  6.396e-04   8.695 1.18e-15 ***
## CACH         6.298e-01  1.344e-01   4.687 5.07e-06 ***
## CHMAX        1.460e+00  2.076e-01   7.031 3.06e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.86 on 203 degrees of freedom
## Multiple R-squared:  0.8648, Adjusted R-squared:  0.8615
## F-statistic: 259.7 on 5 and 203 DF,  p-value: < 2.2e-16
```

Variable selection using step function:

```
null_computer.lm <- lm(PRP ~ 1, data=computerData)
full_computer.lm <- lm(PRP ~ ., data=computerData)

fwd_search <- step(null_computer.lm, scope=list(lower=null_computer.lm,
                                upper=full_computer.lm),direction="forward", trace=F)

bckwd_search <- step(full_computer.lm, data=computerData, direction="backward", trace = F)
```

```r
#summary(fwd_search)

summary(bckwd_search)
```

```
##
## Call:
## lm(formula = PRP ~ MYCT + MMIN + MMAX + CACH + CHMAX, data = computerData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -193.37  -24.95    5.76   26.64  389.66
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.608e+01  8.007e+00  -7.003 3.59e-11 ***
## MYCT         4.911e-02  1.746e-02   2.813   0.0054 **
## MMIN         1.518e-02  1.788e-03   8.490 4.34e-15 ***
## MMAX         5.562e-03  6.396e-04   8.695 1.18e-15 ***
## CACH         6.298e-01  1.344e-01   4.687 5.07e-06 ***
## CHMAX        1.460e+00  2.076e-01   7.031 3.06e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 59.86 on 203 degrees of freedom
## Multiple R-squared:  0.8648, Adjusted R-squared:  0.8615
## F-statistic: 259.7 on 5 and 203 DF,  p-value: < 2.2e-16
```

4. Explain the strategy you applied.

We have used **Backward and Forward selection** for variable selection of the mode

We have computed the *Backward selection* ourselves without using the R *step* function: we start with all variables in the model and then we remove the variable with largest *p-value*(the least stat. significant), we continue to do so until we have reached the condition that all remaining variables have a p-value below some threshold(e.g < 0.1 confidence level 99%) and the same logic would work also for the *forward selection approach*; here we start with *null model* and add it the variable that results in the lowest RSS.

We also have used the **step** function for *variable selection* which uses a different measure to judge the quality of the model: *Akaike information criterion (AIC)*, the logic part is the same as described above, we can clearly see that in the trace of the *forward selection* above (it continuesly adds the variable with the smallest AIC value and stops when the previous step is smaller then the next one).

3. Does you model explain something?

5. Interpret the important values of the output of the final model you obtained with R.

The model is:

$$\mathbf{lm(PRP \sim MYCT + MMIN + MMAX + CACH + CHMAX, data = computerData)}$$

Our final model includes 5 variables in linear regression model, $R^2 = 0.8648$ value which means that our model explains a large portion of variance in the response variable. Adding the variable that was excluded from the model **CGMIN** leads to just a tiny increase in R^2, this is due to the fact that adding another variable to the least squares equations allow us to fit the training data more accurately but it also increases the chance of overfitting and poor prediction performance on the new data.

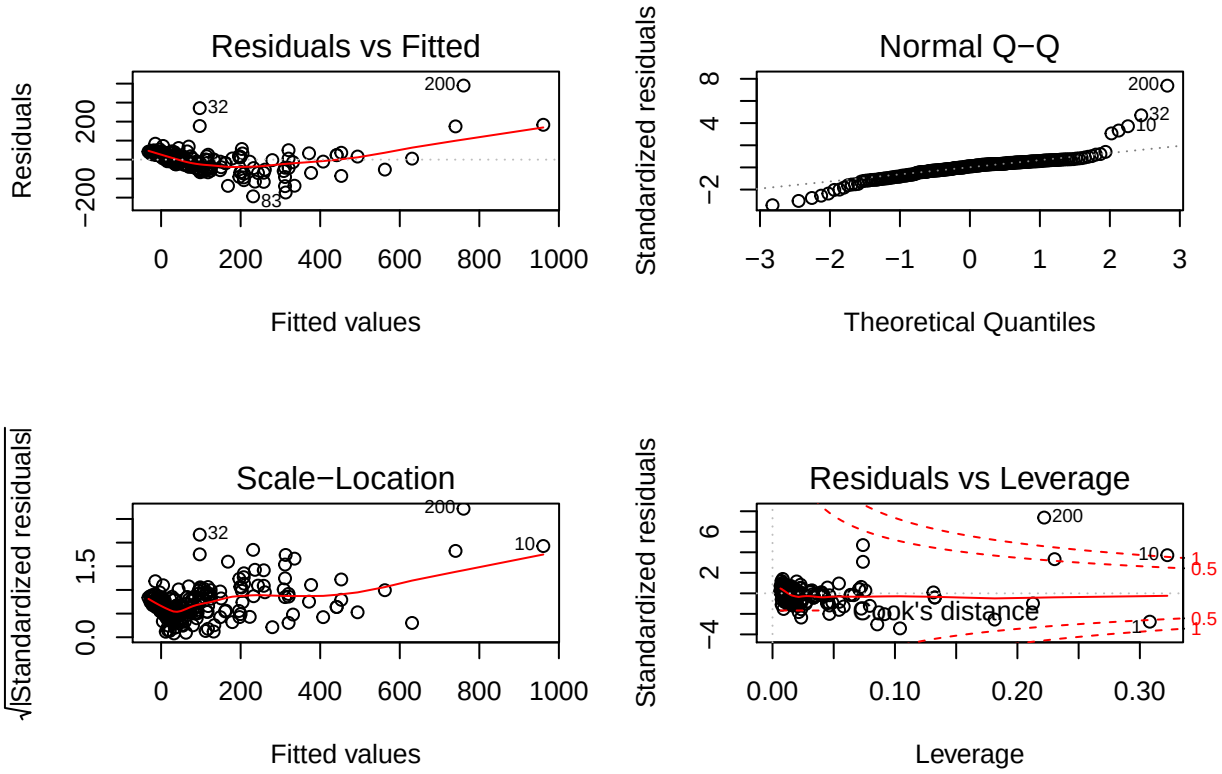
F-statistic: 259.7 on 5 and 203 DF, p-value: < 2.2e-16, since F-statistic is far larger than 1, and p-value associated with the F-statistic is essentially zero, so we have extremely strong evidence against the

null hypothesis H_0.

RSE: 59.86 without CGMIN, **RSE=59.99** with CGMIN, so the model that does not use CGMIN to predict PRP is more accurate.

Question 3

```
computer.lm.model <- lm(PRP ~ MYCT + MMIN + MMAX + CACH + CHMAX, data = computerData)
par(mfrow=c(2,2))
plot(computer.lm.model)
```



Is there outliers or predictions that are hard to predict?

Yes, in the graph **Residuals vs Fitted** we can see that there are 3 data-points which are very far away from our model, so our model does not capture them, so there are 'outliers' (data points 32,83,200) compared to our model(or extreme values/cases).

The graph also shows if residuals have non-linear patterns, we see that we have a small degree of non-linearity on our graph but because we don't have any distinct pattern that is a good indication we don't have non-linear relationships between the predictors and response and that non-linearity degree may be because of non-normalization.

The other two plots (Normal Q-Q and Scale-Location) are not very important for our disscusion.

The fourth plot **Residuals vs Leverage** helps us to find influential cases if any. Not all 'outliers' are influential in linear regression analysis. Even though data have extreme values, they might not be influential to determine a regression line. That means, the results wouldn't be much different if we either include or exclude them from analysis and that is the case in our data we see that the regression line is not really affected by the extreme values.

Question 4

Load the Computer data set from the file ComputerData.txt.

1. Remove the variable that you cannot use to predict performances.

```
cars <- read.table('Cars2Data.txt', header = T)
cars <- cars[-9]
cars <- na.omit(cars)
cars$origin <- as.factor(cars$origin)
summary(cars)
```

```
##       mpg          cylinders      displacement     horsepower
##  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
##  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
##  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
##  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
##  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
##  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
##      weight      acceleration        year       origin
##  Min.   :1613   Min.   : 8.00   Min.   :70.00   1:245
##  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   2: 68
##  Median :2804   Median :15.50   Median :76.00   3: 79
##  Mean   :2978   Mean   :15.54   Mean   :75.98
##  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00
##  Max.   :5140   Max.   :24.80   Max.   :82.00
```

2. Try to build the best multiple linear regression model to predict performances, you can use all the variables.

```
null_cars.lm <- lm(mpg ~ 1, data=cars)
full_cars.lm <- lm(mpg ~ ., data=cars)

fwd_search <- step(null_cars.lm, scope=list(lower=null_cars.lm,
                                upper=full_cars.lm),direction="forward", trace=F)
bckwd_search <- step(full_cars.lm, data=cars, direction="backward", trace = F)

summary(bckwd_search)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     year + origin, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.1754 -2.1139 -0.0863  1.9711 13.4207
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.633e+01  4.219e+00  -3.871 0.000127 ***
## cylinders    -5.028e-01  3.207e-01  -1.568 0.117742
## displacement  2.337e-02  7.613e-03   3.070 0.002292 **
## horsepower   -2.500e-02  1.078e-02  -2.320 0.020855 *
## weight       -6.460e-03  5.763e-04 -11.209  < 2e-16 ***
## year          7.739e-01  5.161e-02  14.994  < 2e-16 ***
```

```
## origin2       2.635e+00  5.661e-01   4.654 4.50e-06 ***
## origin3       2.857e+00  5.525e-01   5.172 3.74e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.305 on 384 degrees of freedom
## Multiple R-squared:  0.8239, Adjusted R-squared:  0.8207
## F-statistic: 256.7 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
#start with all the descriptors of computer performance in the lm
summary.lm <- summary(lm(mpg ~ ., data=cars))

vars <- names(cars)
vars <- vars[which(vars != "mpg")]
tol <- 0.01

#while there are still variables with p-values larger than tol
while(length(which(summary.lm$coefficients[,4] > tol)) != 0){

  #coefficients table
  coef <- summary.lm$coefficients

  #find the variable with the largest p-value
  var_with_max_p.value <- rownames(coef)[apply(coef,2,which.max)][4]

  #remove the variable with the larger value
  vars <- vars[which(vars != var_with_max_p.value)]

  #computer the model with the updated variables
  lm2 <- lm(cars$mpg ~ ., data = cars[vars])
  summary.lm <- summary(lm2)
}
summary.lm
```

```
##
## Call:
## lm(formula = cars$mpg ~ ., data = cars[vars])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.6025 -2.1132 -0.0206  1.7617 13.5261
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.831e+01  4.017e+00  -4.557 6.96e-06 ***
## weight      -5.887e-03  2.599e-04 -22.647  < 2e-16 ***
## year         7.698e-01  4.867e-02  15.818  < 2e-16 ***
## origin2      1.976e+00  5.180e-01   3.815 0.000158 ***
## origin3      2.215e+00  5.188e-01   4.268 2.48e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.337 on 387 degrees of freedom
## Multiple R-squared:  0.819,  Adjusted R-squared:  0.8172
## F-statistic: 437.9 on 4 and 387 DF,  p-value: < 2.2e-16
```

4. Explain the strategy you applied.

We have used **Backward and Forward selection** for variable selection of the mode

We have used the **step** function for *variable selection* which uses *Akaike information criterion (AIC)* measure to judge the quality of the model, the logic part is the same and we can clearly see in the trace of the *forward selection* how it continuesly adds the variable with the smallest AIC value and stops when the previous step is smaller then the next one.

The model that we gain with this solution is:

lm(mpg ~ cylinders+displacement+horsepower+weight+year+origin, data = cars)

When we tried our solution for *forward selection* we got a different solution for the model and actually looking at some parameters it looks that our solution might be better(the reason for this difference might be because of different quality measurements that we use *AIC vs p-value* and also the tolerance that I have used), for example the p-value for 'cylinders' is 0.117742(also 'horsepower' has a large p-value), **F-statistic** is higher in our model(437 vs 256), and we have similar **RSE** (3.337 vs 3.305) and R^2 (0.819 vs 0.8239), so it means that our model might be more robust to overfitting.

If we increase the tolerance to 0.05 (95% confidence level) than the model that we gain is:

lm(mpg ~ displacement + horsepower + weight + year + origin, data = cars)

but again we would have only a tiny increase in R^2 value and a drop in F-statistic.

The model that we gain with our solution(without using step function) is:

lm(mpg ~ weight + year + origin, data = cars)

and this is the model that we're goint to use for the other part of the question, we also we will plot both of them and we will clearly see that this model actually might be better.

3. Does you model explain something?

5. Interpret the important values of the output of the final model you obtained with R.

The model is:

lm(mpg ~ weight + year + origin, data = cars)

Our final model includes 3 variables in linear regression model, R^2 value is 0.8190 which means that our model explains a large portion of variance in the response variable. Adding the variables that were excluded from the model **acceleration, cylinders, displacement, horsepower** leads to just a tiny increase in R^2 value(0.8239), this is due to the fact that adding another variable to the least squares equations allow us to fit the training data more accurately but it also increases the chance of overfitting and poor prediction performance on the new data.

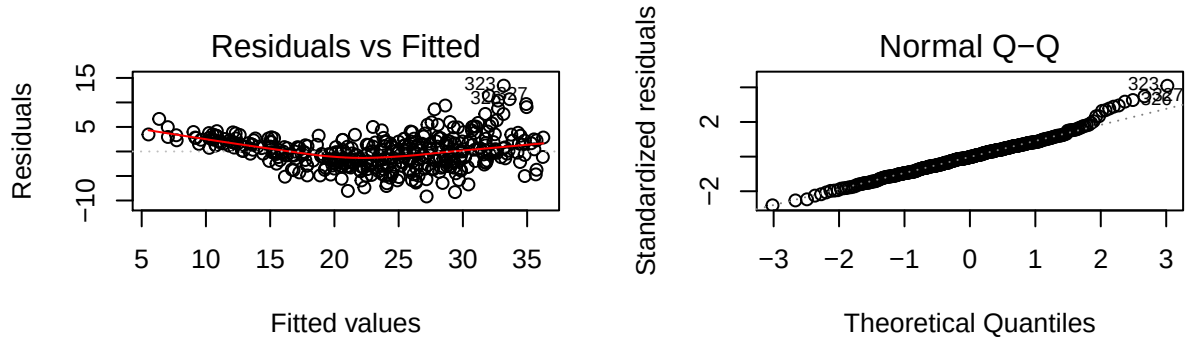
F-statistic: 437.9 on 4 and 387 DF, p-value: < 2.2e-16, since F-statistic is far larger than 1, and p-value associated with the F-statistic is essentially zero, so we have extremely strong evidence against the null hypothesis H_0 (F-statistic = 256 in LM model gained with 'step' func)

RSE = 3.337 a slight increase compared to that of the LM model of step function(**RSE = 3.305**).

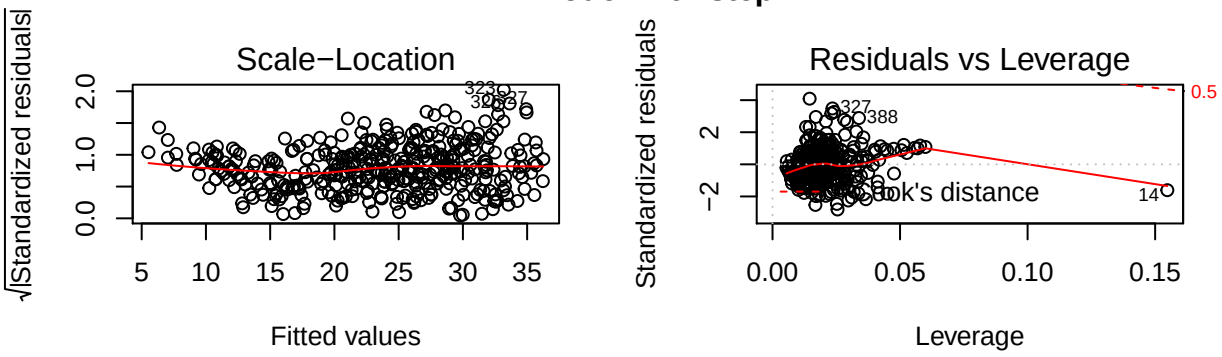
Plot your model graphically on a graph with all the data.

We will do a comparison between the two models: the one we get using the *step* function and the one we get using our method for *backward selection*.

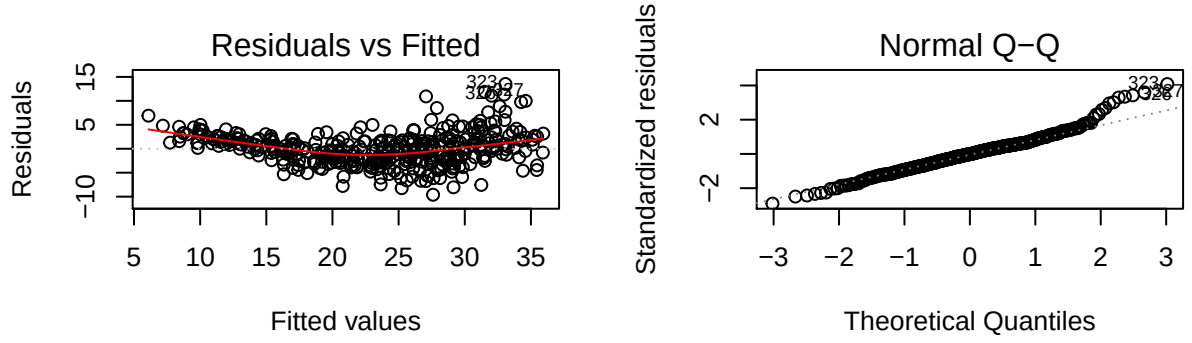
```
cars.lm <- lm(mpg ~ cylinders + displacement + horsepower + weight +  year + origin,
              data = cars)
par(mfrow=c(2,2))
plot(cars.lm)
title("LM model with step", line = -15, outer=TRUE)
```



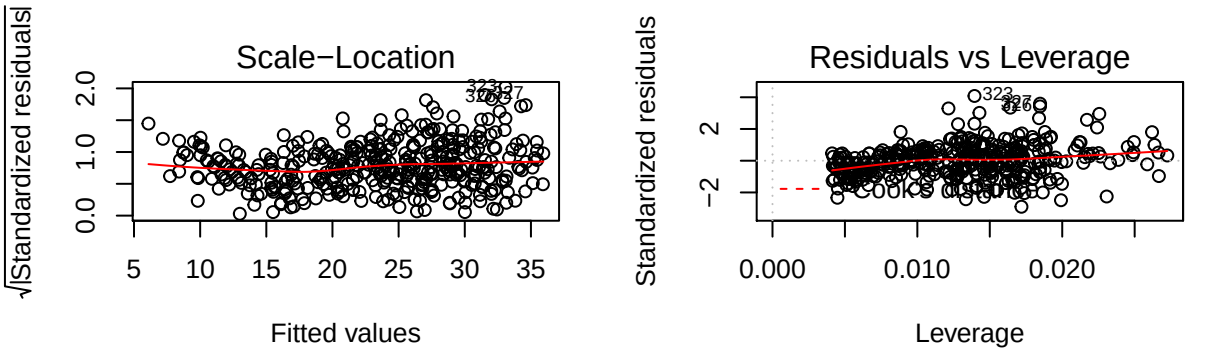
LM model with step




```
cars_lm2 <- lm(mpg ~ weight + year + origin, data = cars)
par(mfrow=c(2,2))
plot(cars_lm2)
title("LM model with our method", line = -15, outer=TRUE)
```



LM model with our method



Is there outliers or predictions that are hard to predict?

Yes, in the graph **Residuals vs Fitted** we can see that there are 2 data-points which are far away from our model, so our model does not capture them, so we would say that there are some extreme values (data points 323, 327) compared to our model that are hard to predict.

The graph also shows that we have a small degree of non-linearity on our model but because we don't have any distinct patterns that is a good indication we don't have non-linear relationships between the predictors and response; that small non-linearity degree may be because of non-normalization.

The other two plots (**Normal Q-Q and Scale-Location**) are not very important for our disscusion.

The fourth plot **Residuals vs Leverage** helps us to find influential cases if any in our model. Even though data have extreme values, they might not be influential to determine a regression line.

This is the graph that shows a very meaningful difference between the two models, we can see that in the plot of the **LM model gained by the step** function the datapoint 14 influences our regression line and also the pattern is somewhat strange with all the points clustered, but in the plot of the **LM model gained with our method** we don't have any real influential cases which proves the point that our model is more robust.