Exercise 5

for Data Analysis

Valentino Lazarevic - 1223211

23.04.2024

Setup

```
setwd("/Users/valentino/Documents/Studium/Semester 4/Datenanalyse/Exercises/exercise5")

Diamonds = read.csv("Diamonds.csv")
library(robustHD)

data(TopGear)
library(mblm)
library(robustbase)
```

Exercise 1 - 5

Use the data set TopGear, included in the R package robustHD. The data set contains information on various characteristics of cars featured on the website of the BBC television show Top Gear. There are 297 observations, but some have missing values. We want to predict the logarithm of Price from the variable MPG. Plot log-Price versus MPG and draw into the plot the regression lines of the following methods:

1. LS-regression (lm())

2. Tukey line (line())

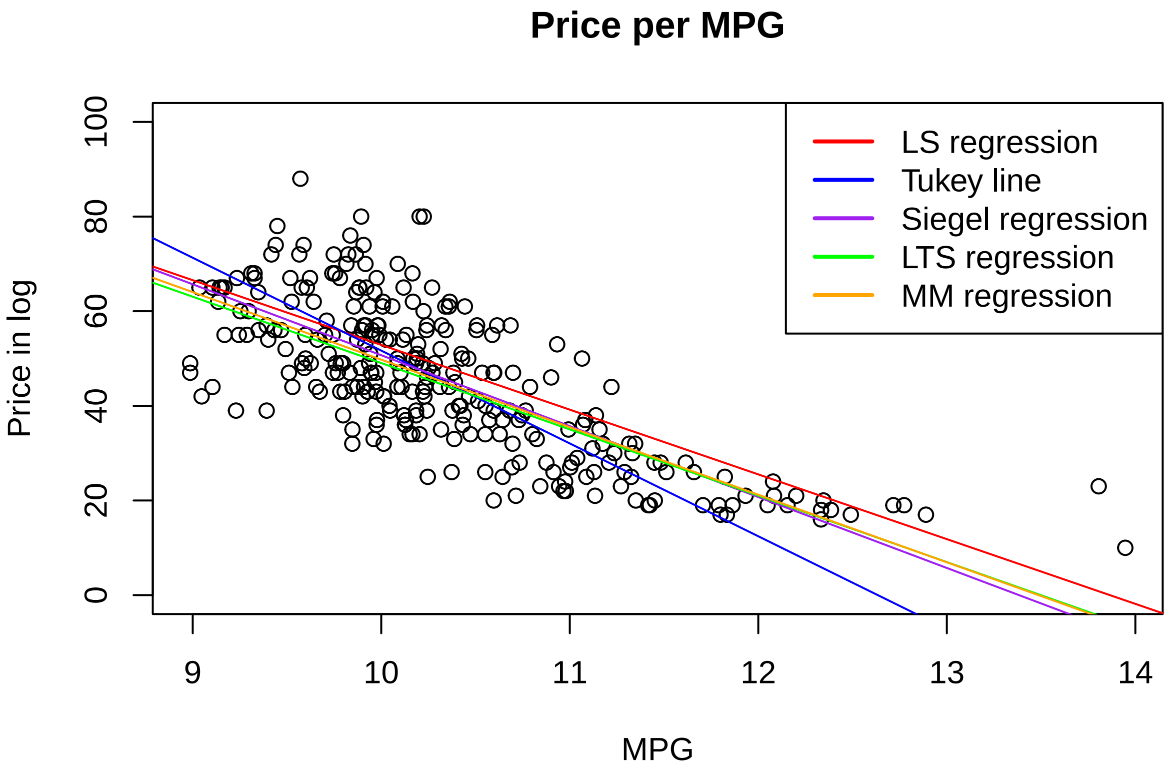
3. Siegel regression (mblm() from the package mblm)

4. LTS regression (ltsReg() from the package robustbase)

5. MM regression (lmrob() from the package robustbase)

Draw briefly general conclusions of the results.

The standard Least Squares regression is heavily influenced by outlier leverage points, while robust methods like Tukey and MM-regression align more closely. Siegel and LTS regression exhibit flatter slopes, possibly due to a greater density of data points at lower prices for Siegel, and LTS filtering out both leverage and vertical outliers. The plot depicts a negative linear relationship between combined fuel consumption (MPG) and the logarithm of the list price in UK pounds, indicating that higher-priced vehicles tend to have lower fuel consumption.

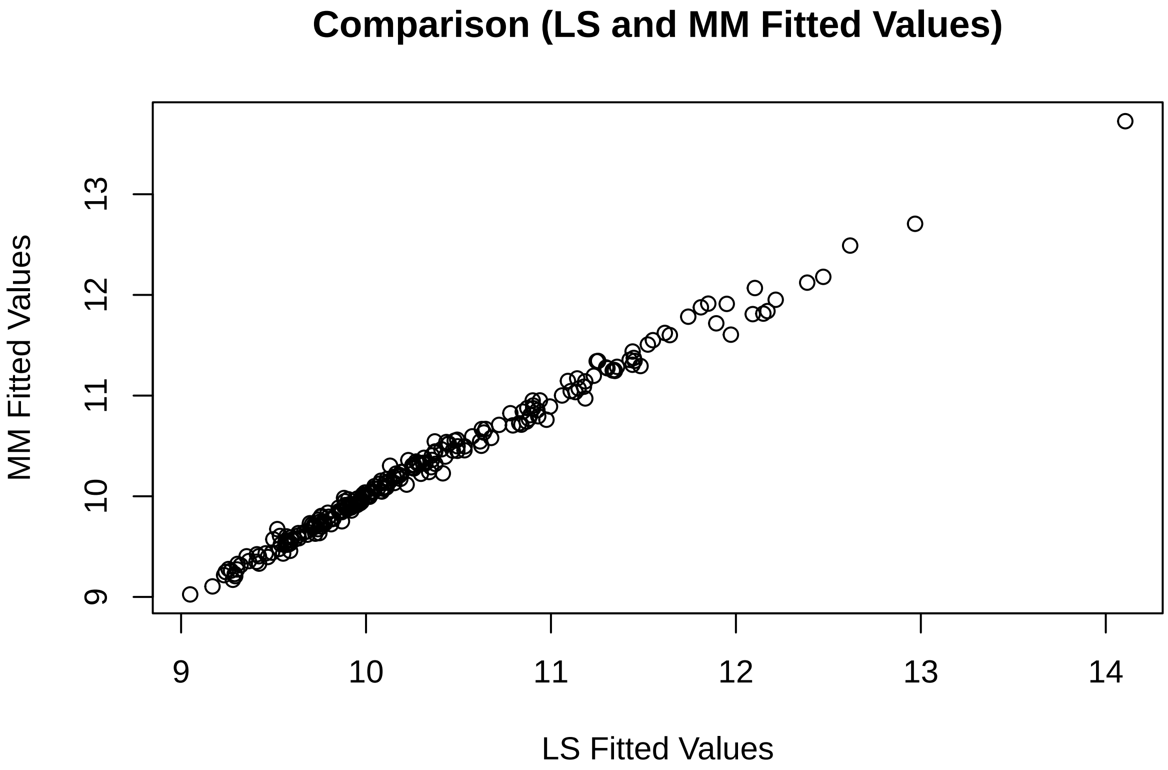


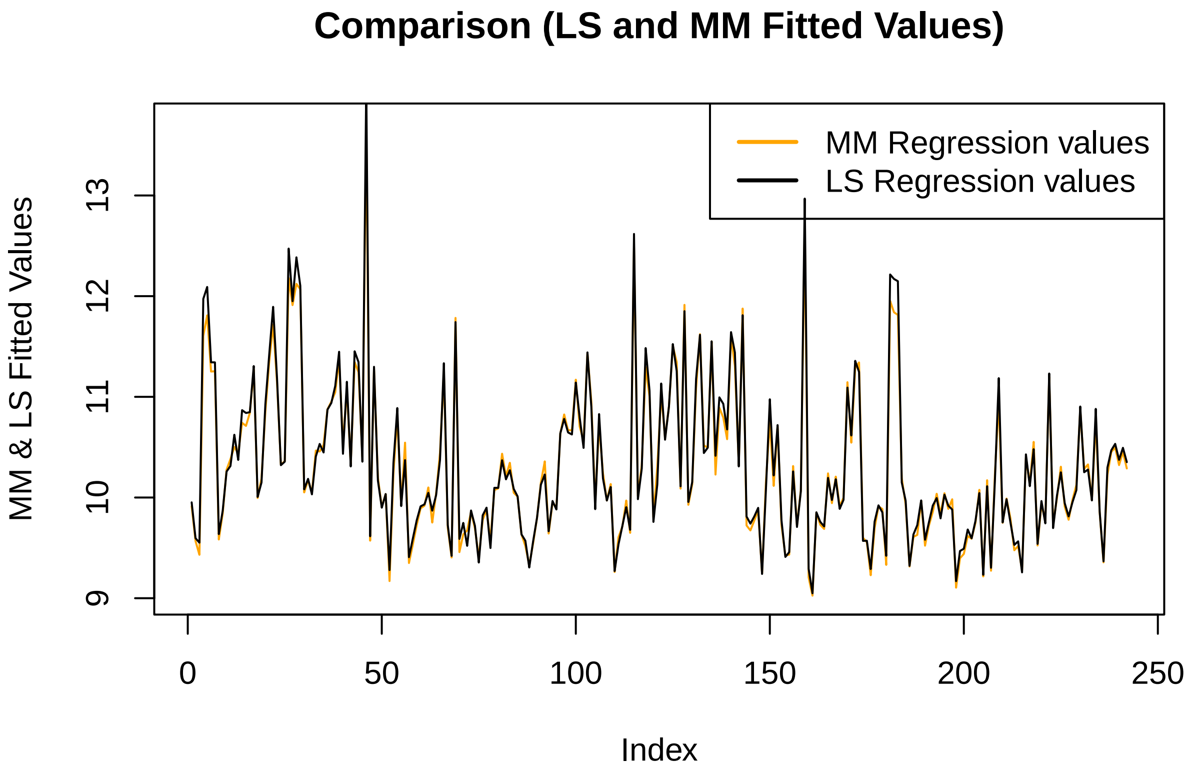
Exercise 6

Now use all variables of the TopGear data set which are measured on a continuous scale to predict log-Price. First you might want to exclude observations with missings us- ing na.omit(). For estimating the regression parameters, use LS-regression and MM regression.

6. Compare the fitted values y^i from both estimators. What do you conclude?

The fitted values from the LS and MM regression models generally follow the same trend, but at higher values, there's a noticeable deviation towards LS fitted values, indicating LS estimates bigger values for larger data points compared to MM regression. This discrepancy is more pronounced considering the fitted values are log-scaled prices, emphasizing larger price differences between regression models at higher price intervals. Despite this, a strong linear relationship between the fitted values from MM and LS regression models is evident in the scatter plot, suggesting they produce very similar results for the logarithm of price in the dataset.





Exercise 7

7. Compare the outcome of the inference tables (i.e. the p-values in the summary() output). What do you conclude?

The LS-regression highlights Cylinders, MPG, and Weight as significant features, whereas MM-regression focuses on a smaller set, including BHP, MPG, Weight, and Width, omitting Cylinders. Cylinders are particularly interesting because they are significant for LS but not for MM, possibly due to high cylinder cars acting as leverage points that skew LS results but have little impact on robust methods like MM. Both outputs agree on strong predictors like BHP, MPG, and Weight, while some variables show significance in only one model, indicating potential model sensitivity or contextual influences. Other predictors, such as Displacement and Length, demonstrate no consistent effect across models, suggesting they are weak or inconsistent predictors.

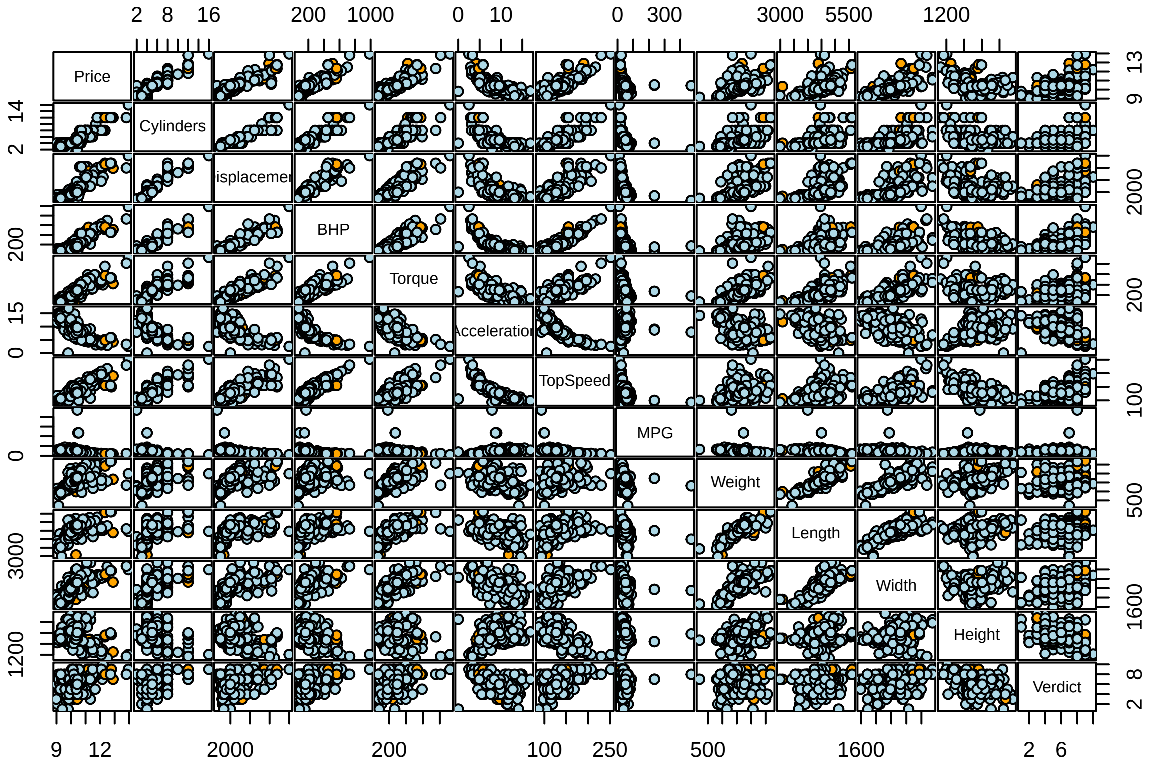
```
##
## Call:
## lm(formula = log(Price) ~ ., data = topgear[, sapply(topgear,
##     is.numeric)])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.70184 -0.12510 -0.02330  0.09727  0.91641
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.438e+00  6.324e-01  10.180  < 2e-16 ***
## Cylinders     1.061e-01  2.256e-02   4.703 4.44e-06 ***
## Displacement -2.132e-05  4.033e-05  -0.529 0.597531    
## BHP           1.296e-03  4.717e-04   2.748 0.006467 ** 
## Torque        9.577e-04  2.997e-04   3.196 0.001592 ** 
## Acceleration  7.595e-03  1.073e-02   0.708 0.479881    
## TopSpeed      5.849e-03  2.176e-03   2.688 0.007722 ** 
## MPG           1.996e-03  5.446e-04   3.665 0.000308 ***
## Weight        3.693e-04  1.064e-04   3.472 0.000617 ***
## Length       -4.712e-05  8.132e-05  -0.579 0.562856    
## Width         1.043e-03  3.843e-04   2.713 0.007166 ** 
## Height       -3.176e-04  2.055e-04  -1.545 0.123625    
## Verdict       2.658e-02  1.093e-02   2.431 0.015828 *  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Residual standard error: 0.2334 on 229 degrees of freedom
## Multiple R-squared:  0.924,  Adjusted R-squared:   0.92 
## F-statistic: 232.1 on 12 and 229 DF,  p-value: < 2.2e-16
```

```
## 
## Call:
## lmrob(formula = log(Price) ~ ., data = topgear[, sapply(topgear, is.numeric)])
##  \--> method = "MM"
## Residuals:
##       Min        1Q    Median        3Q       Max 
## -0.682519 -0.095928  0.002613  0.103035  1.283968 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   5.957e+00  4.721e-01  12.618  < 2e-16 ***
## Cylinders     1.519e-02  3.489e-02   0.436 0.663579    
## Displacement  3.129e-07  5.525e-05   0.006 0.995487    
## BHP           2.395e-03  4.492e-04   5.332 2.32e-07 ***
## Torque        4.974e-04  3.063e-04   1.624 0.105821    
## Acceleration  2.082e-02  8.621e-03   2.415 0.016503 *  
## TopSpeed      5.676e-03  2.548e-03   2.227 0.026897 *  
## MPG           1.947e-03  3.693e-04   5.271 3.14e-07 ***
## Weight        3.887e-04  9.738e-05   3.991 8.86e-05 ***
## Length        3.080e-05  9.876e-05   0.312 0.755400    
## Width         1.299e-03  3.467e-04   3.746 0.000227 ***
## Height       -4.352e-04  1.843e-04  -2.362 0.019026 *  
## Verdict       2.398e-02  1.062e-02   2.258 0.024915 *  
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
## 
## Robust residual standard error: 0.156 
## Multiple R-squared:  0.9523, Adjusted R-squared:  0.9498 
## Convergence in 22 IRWLS iterations
## 
## Robustness weights: 
##  7 observations c(3,4,136,150,159,182,183)
##   are outliers with |weight| = 0 ( < 0.00041); 
##  20 weights are ~= 1. The remaining 215 ones are summarized as
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
## 0.01649 0.86400 0.95890 0.89060 0.98660 0.99880
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##         1.548e+00         5.000e-01         4.685e+00         1.000e-07
##           rel.tol         scale.tol         solve.tol          zero.tol
##         1.000e-07         1.000e-10         1.000e-07         1.000e-10
##       eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
##         4.132e-04         1.454e-08         5.000e-01         5.000e-01
##      nResample         max.it       best.r.s       k.fast.s          k.max
##            500             50              2              1            200
##    maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##            200              0           1000              0           2000
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)
```

Exercise 8

8. From MM regression we obtain weights in [0, 1], corresponding to the outlyingness of the observations (list element $rweights). Visualize this information e.g. by two colors (small/larger weight) in a plot of the data frame (=scatterplot matrix) to investigate the reason for the outlyingness.

The data analysis reveals challenges with extreme values and non-linear relationships. Extreme values in columns like MPG and Weight may be outliers or influential observations, while non-linear relationships, such as between Acceleration and BHP, pose difficulties for linear MM regression models. Moreover, scatterplots indicate varying data spread across the range of variables like Height. The scatterplot matrix suggests inconclusive reasons for extreme outliers, particularly in the extremely high price segment, where other qualities may not justify the price.

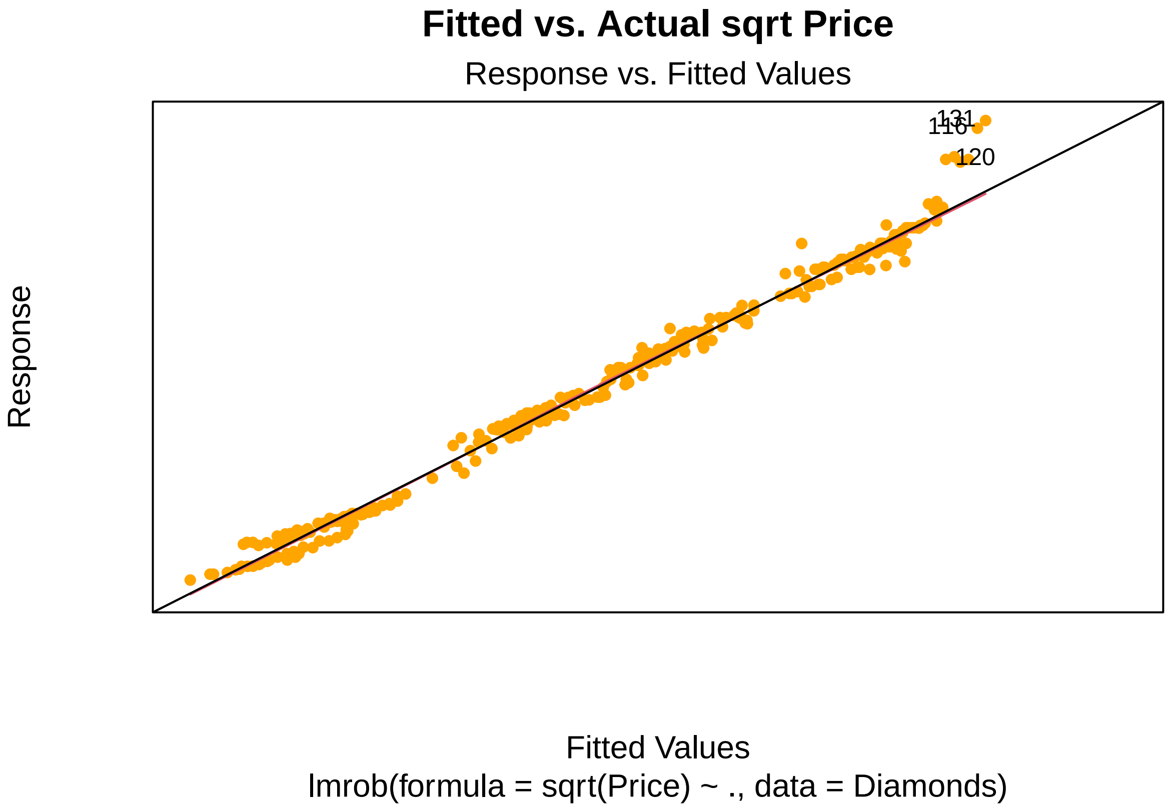


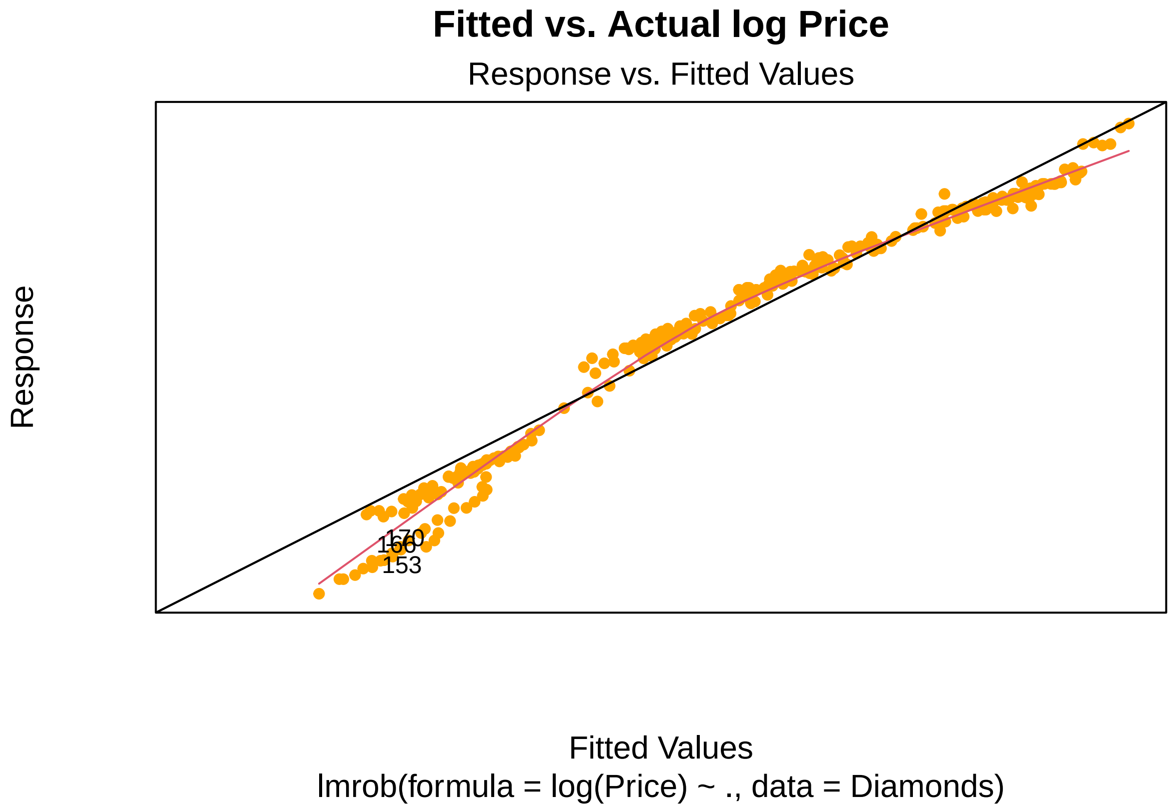
Exercise 9

Use again the data set Diamonds from last exercise and perform MM regression of Price (after sqrt- or log-transformation) on the remaining variables.

9. Compare the fitted values with those of the response. What do you conclude?

The sqrt-transformed plot showcases MM-regression's fitted values closely tracking the response data, with a notable deviation appearing only at certain price points. Conversely, the log-transformed plot illustrates a more pronounced discrepancy, particularly with lower prices being overestimated and higher prices underestimated by MM-regression. This underscores the substantial impact of transformation methods on estimation accuracy, with sqrt-transformed prices showing greater promise. This comparison highlights the nuanced relationship between transformation techniques and the success of regression estimation.





Exercise 10

10. Interpret the output of the inference table.

It's evident that all factors, except Lab, play a significant role in predicting the price.

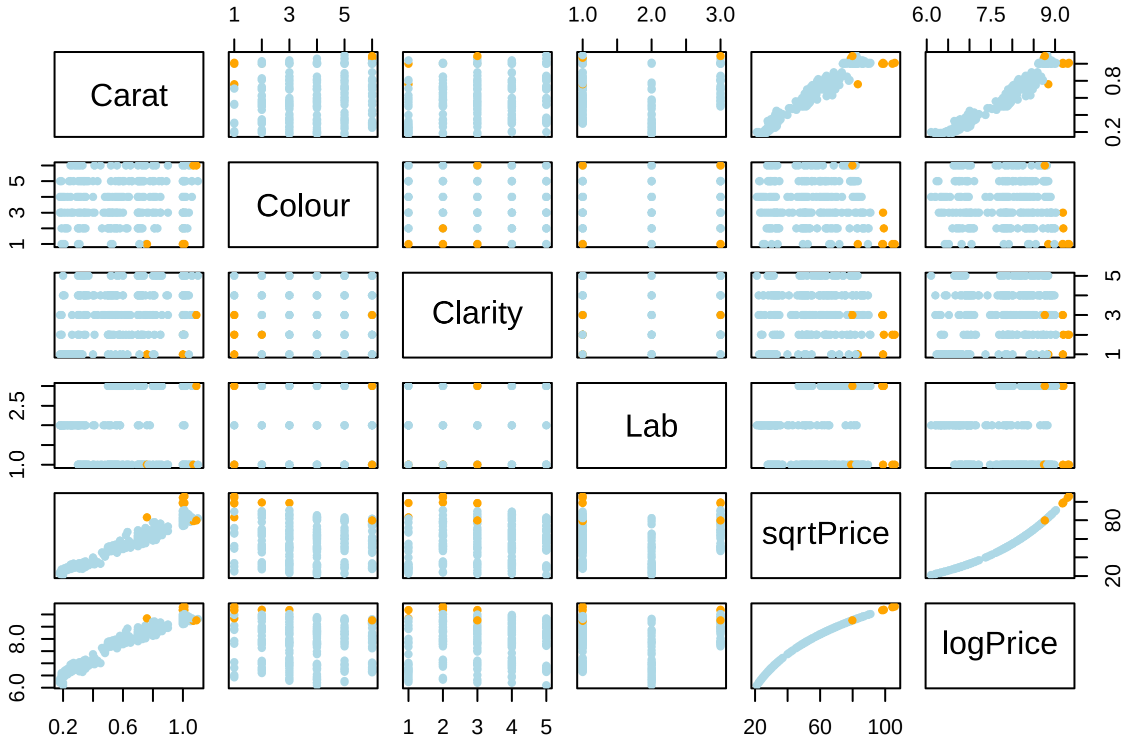
```
##
## Call:
## lmrob(formula = sqrt(Price) ~ ., data = Diamonds)
##  \--> method = "MM"
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.4258 -1.1631  0.2161  1.0950 13.0619
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22.06163    0.54344   40.60   <2e-16 ***
## Carat       75.94077    0.49997  151.89   <2e-16 ***
## Colour      -2.39981    0.09102  -26.36   <2e-16 ***
## Clarity     -1.84422    0.11442  -16.12   <2e-16 ***
## Lab          0.12109    0.11528    1.05    0.294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 1.663
## Multiple R-squared:  0.9935, Adjusted R-squared:  0.9934
## Convergence in 10 IRWLS iterations
##
## Robustness weights:
##  5 observations c(116,120,131,278,279)
##   are outliers with |weight| <= 5.6e-05 ( < 0.00032);
##  16 weights are ~= 1. The remaining 287 ones are summarized as
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.004385 0.892500 0.953200 0.912800 0.987900 0.999000
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##         1.548e+00         5.000e-01         4.685e+00         1.000e-07
##           rel.tol         scale.tol         solve.tol          zero.tol
##         1.000e-07         1.000e-10         1.000e-07         1.000e-10
##       eps.outlier             eps.x warn.limit.reject warn.limit.meanrw
##         3.247e-04         1.091e-11         5.000e-01         5.000e-01
##      nResample         max.it       best.r.s       k.fast.s          k.max
##            500             50              2              1            200
##    maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##            200              0           1000              0           2000
##                   psi           subsampling                   cov
##            "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                  "SM"
## seed : int(0)


##
## Call:
## lmrob(formula = log(Price) ~ ., data = Diamonds)
##  \--> method = "MM"
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48555 -0.12400 -0.01835  0.10573  0.23250
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.716728   0.102053  65.816  < 2e-16 ***
## Carat        2.835298   0.088285  32.115  < 2e-16 ***
## Colour      -0.100301   0.007082 -14.163  < 2e-16 ***
## Clarity     -0.070444   0.010330  -6.819 4.96e-11 ***
## Lab          0.003542   0.011543   0.307    0.759
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Robust residual standard error: 0.09554
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## Convergence in 31 IRWLS iterations
##
## Robustness weights:
##  2 observations c(166,170) are outliers with |weight| = 0 ( < 0.00032);
##  13 weights are ~= 1. The remaining 293 ones are summarized as
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.001549 0.754100 0.859700 0.791100 0.935300 0.998700
## Algorithmic parameters:
##        tuning.chi                bb        tuning.psi        refine.tol
##          1.548e+00          5.000e-01          4.685e+00          1.000e-07
##            rel.tol          scale.tol          solve.tol           zero.tol
##          1.000e-07          1.000e-10          1.000e-07          1.000e-10
##        eps.outlier              eps.x warn.limit.reject warn.limit.meanrw
##          3.247e-04          1.091e-11          5.000e-01          5.000e-01
##       nResample          max.it       best.r.s       k.fast.s          k.max
##             500              50              2              1            200
##     maxit.scale      trace.lev            mts     compute.rd fast.s.large.n
##             200               0           1000              0           2000
##                    psi           subsampling                   cov
##             "bisquare"         "nonsingular"         ".vcov.avar1"
## compute.outlier.stats
##                   "SM"
## seed : int(0)
```

Exercise 11

11. Use the robustness weights as color information in the scatterplot matrix of the data frame to investigate the reason for the outlyingness.

The analysis involves examining 16 plots depicting the relationship between four variables and log-transformed and sqrt-transformed prices. While carat exhibits a linear relationship with both transformed prices, outliers occur mainly at high carat and high prices. Although Lab may not influence regression, attention shifts to color and clarity, where outliers appear across all levels but mainly at high prices. Outliers in the dataset are concentrated in the high-price segment, likely serving as detrimental leverage points, particularly evident in diamonds with high clarity and color ratings.


Carat
Colour
Clarity
Lab
sqrtPrice
logPrice