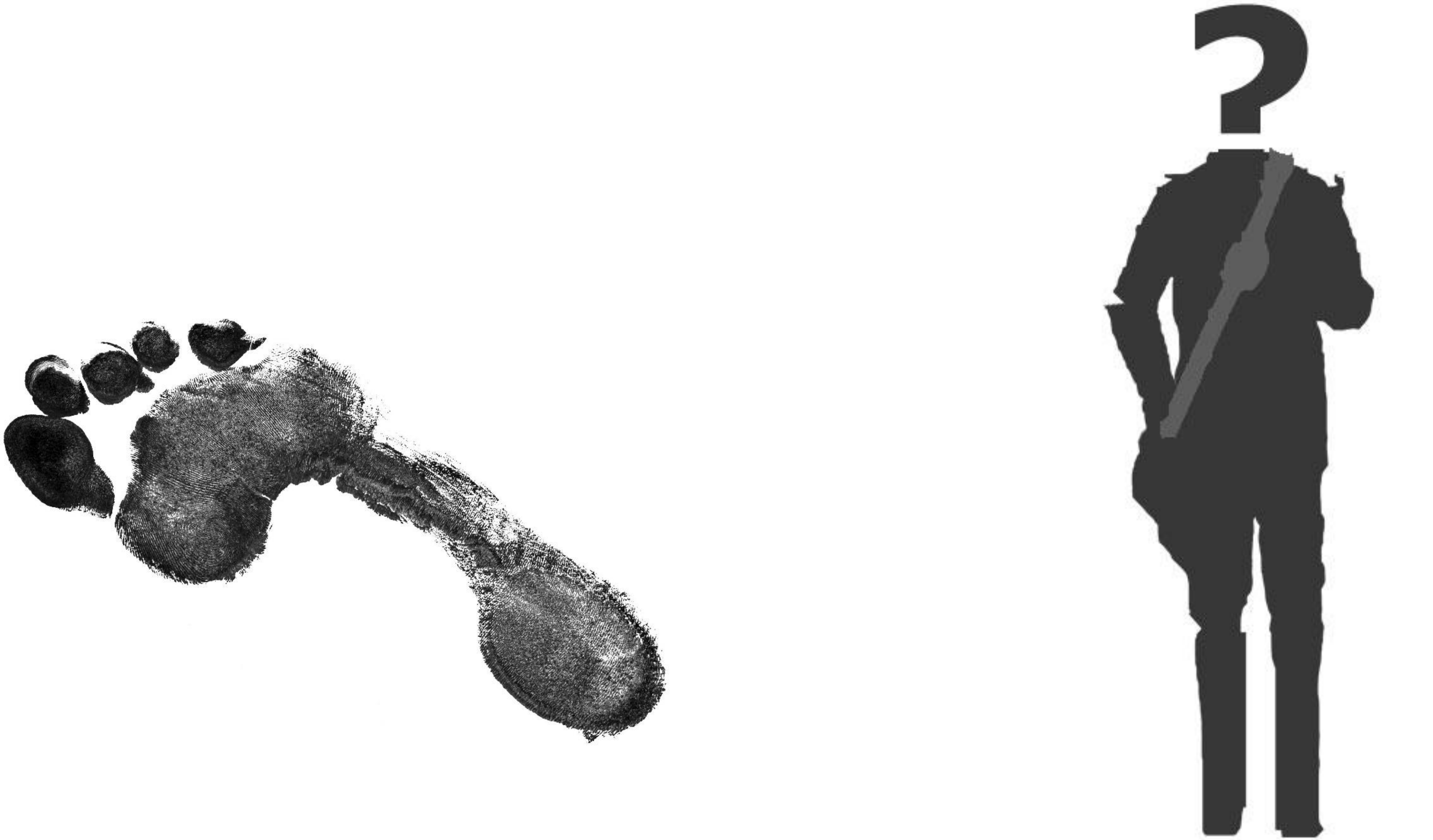
Advanced Analytics with R

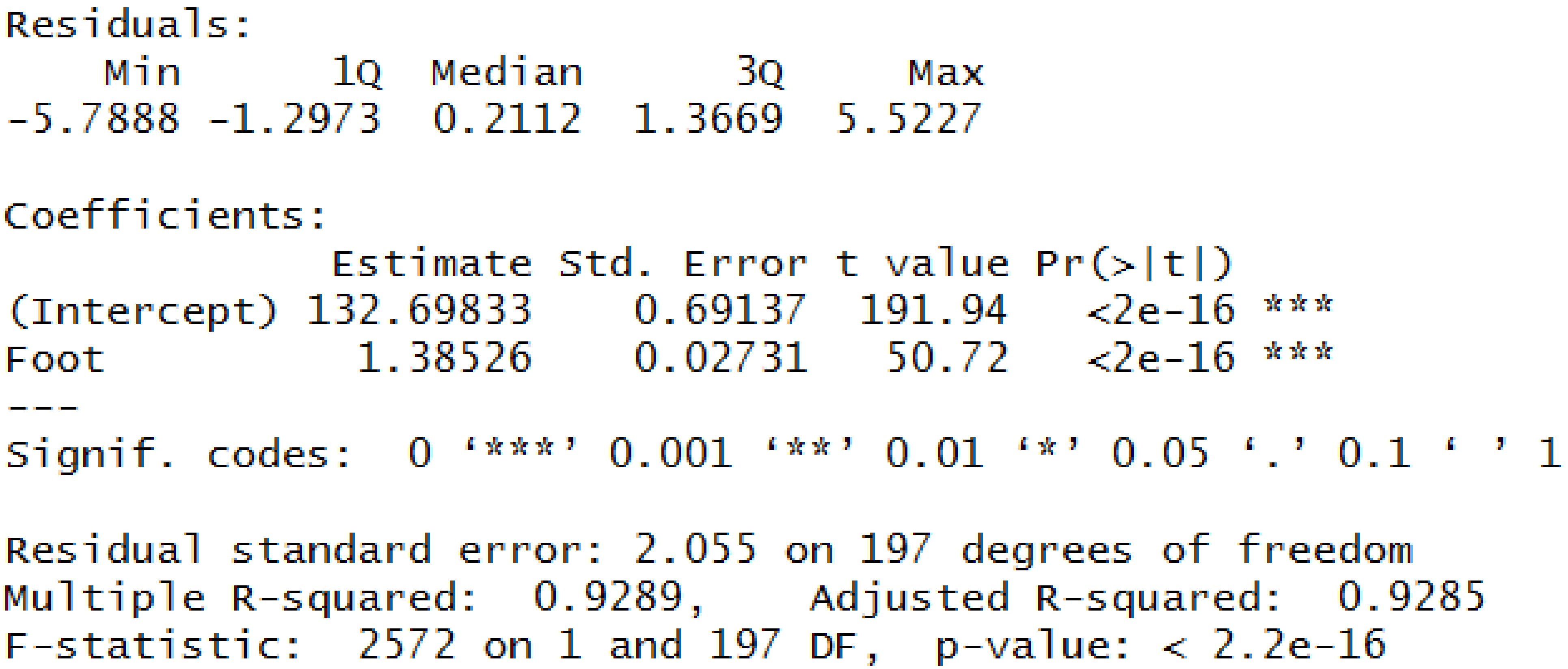
Linear Regression

Simple Linear Regression with R

The night had been warm and very humid. Near closing time a small neighborhood market was robbed and the owner of the market killed. Police arrived on the scene quickly because an alarm had been sounded from the market. The suspect fled on foot through the rear door. At the rear of the store were a grassy area and a sidewalk. When police came through the rear of the store they found a clear set of wet footprints on the sidewalk. The footprints were immediately photographed and measured for size and length of stride. From this evidence alone they were able to determine approximate height, shoe size, any stride deviation and also had the ability to later match a suspect’s shoe tread with the pattern left on the sidewalk.



Summary of Model Fitness



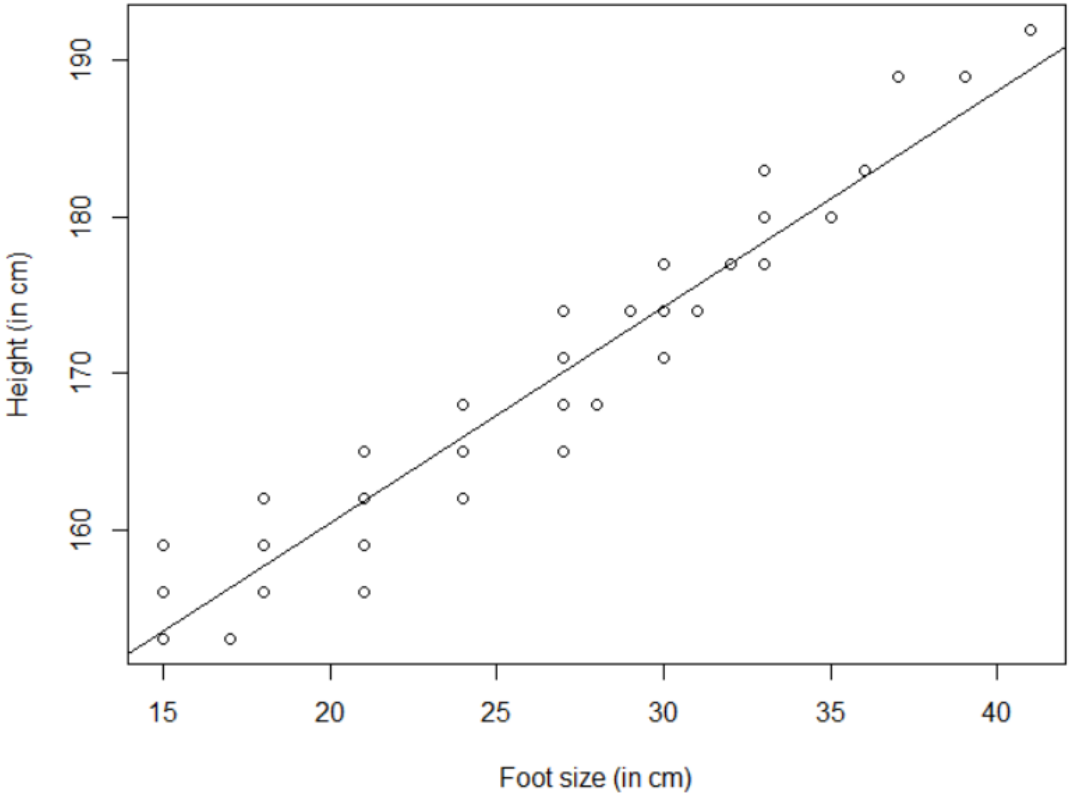
```
Residuals:
    Min      1Q  Median      3Q     Max 
-5.7888 -1.2973  0.2112  1.3669  5.5227 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 132.69833    0.69137  191.94   <2e-16 ***
Foot          1.38526    0.02731   50.72   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.055 on 197 degrees of freedom
Multiple R-squared:  0.9289,    Adjusted R-squared:  0.9285 
F-statistic:  2572 on 1 and 197 DF,  p-value: < 2.2e-16
```

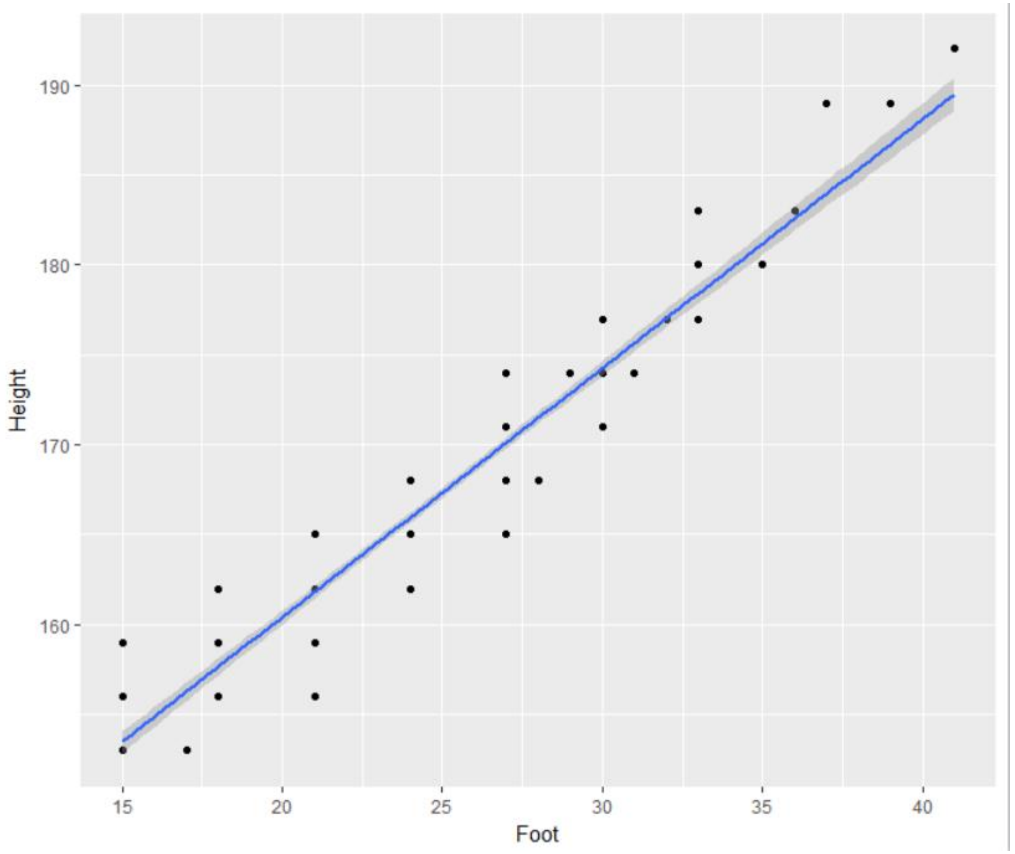
Height = 1.38526 * Foot + 132.69833

Visualize Fitness

```
coefs<-coef(fit)
ggplot(data,aes(x=Foot,y=Height)) +
geom_point() + geom_abline(intercept =
coefs[1],slope=coefs[2],color="red")
```

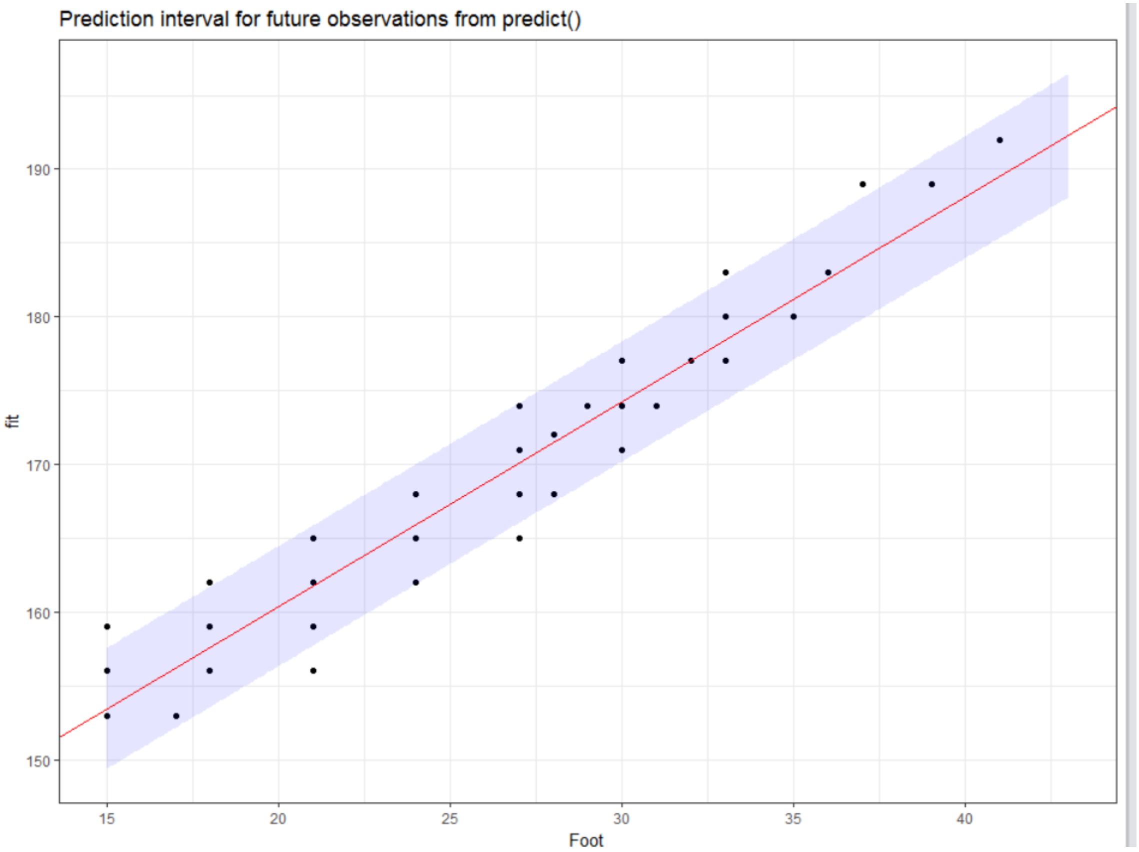
```
ggplot(data,aes(x=Foot, y=Height)) + geom_point() +
geom_smooth(method="lm")
```





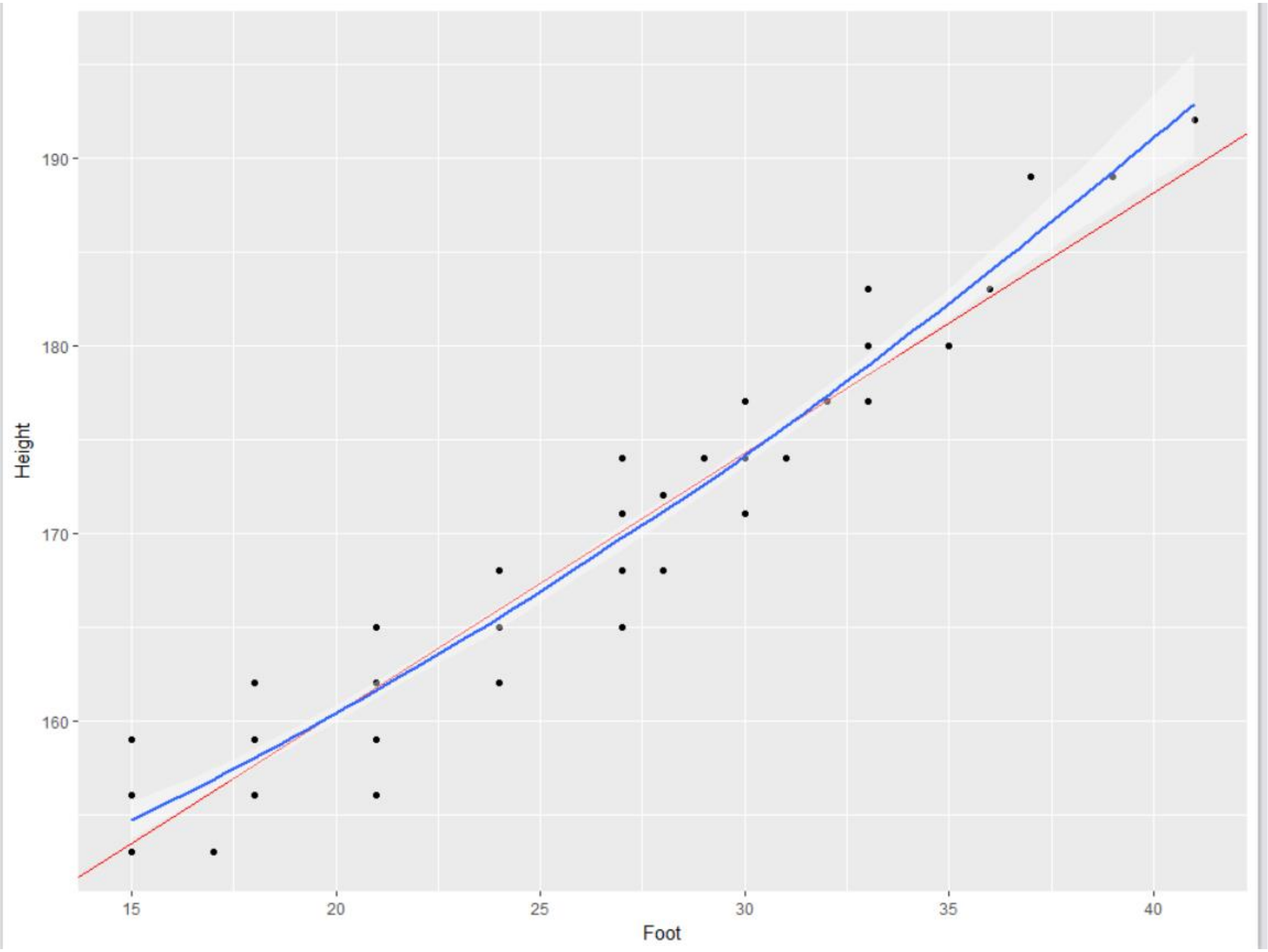
Prediction Interval

```
predx <- data.frame(Foot = seq(from = 15, to = 43, by = 0.1))

 pred.int <- cbind(predx, predict(fit, newdata = predx, interval = "prediction", level = 0.95))

ggplot(pred.int, aes(x = Foot, y = fit)) + theme_bw() +
   ggtitle("Prediction interval for future observations from predict()") +
   geom_point(data = data, aes(x = Foot, y = Height)) +
   geom_ribbon(data = pred.int, aes(ymin = lwr, ymax = upr), stat =
"identity",fill='blue',alpha=0.1) + geom_abline(intercept =
a[1],slope=a[2],color="red")
```



Fitting trend

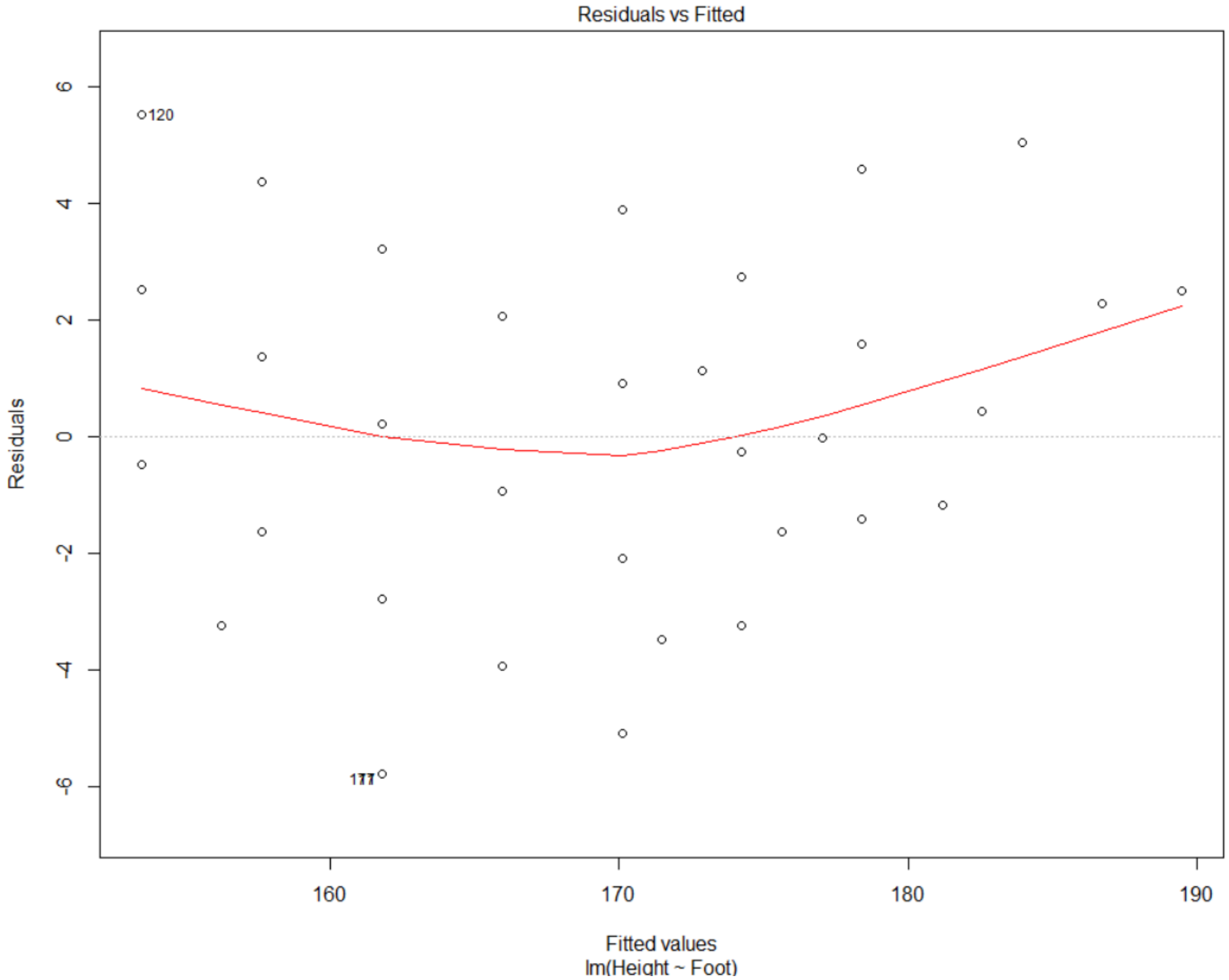
```
ggplot(data,aes(x=Foot,y=Height)) + geom_point() +
geom_abline(intercept = a[1],slope=a[2],color="red")+ geom_smooth(fill='NA')
```



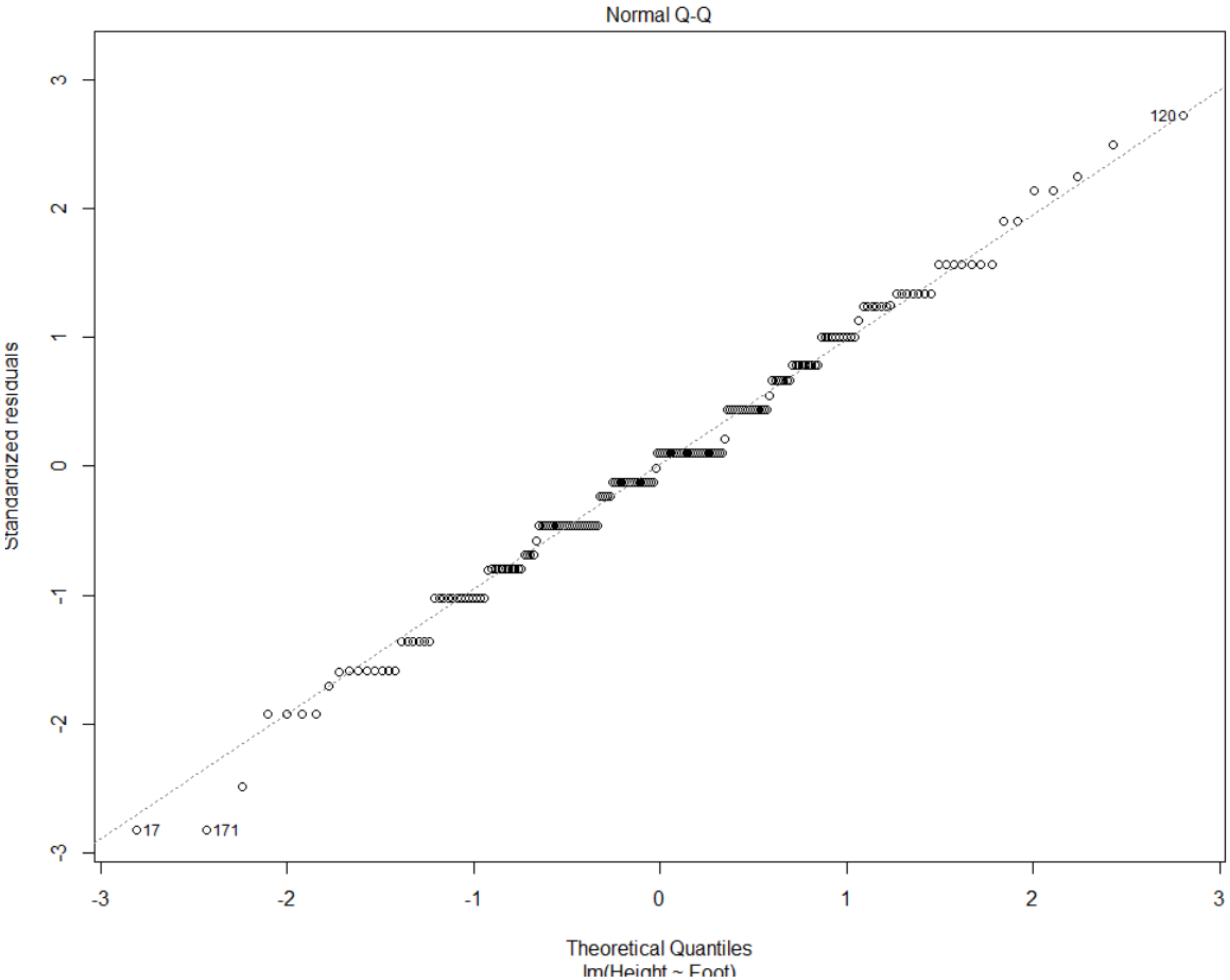
Assumptions of Linear Regression

- Linearity
- Normality of error
- No auto-correlation (Independence of errors)
- Homoscedasticity (equal variance)

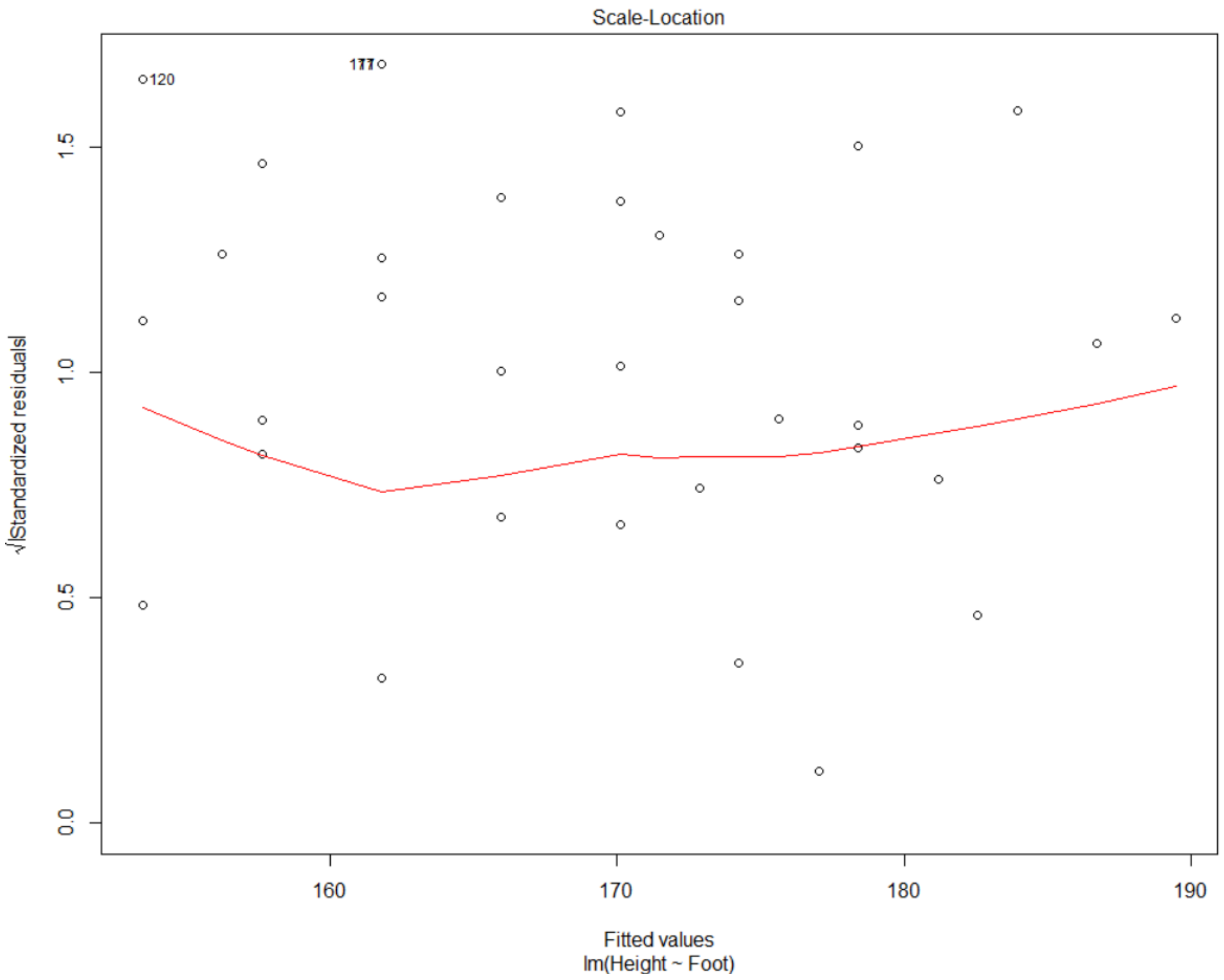
Plot(fit)


Residuals vs Fitted
Residuals
6
4
2
0
-2
-4
-6
120
177
160
170
180
190
Fitted values
lm(Height ~ Foot)

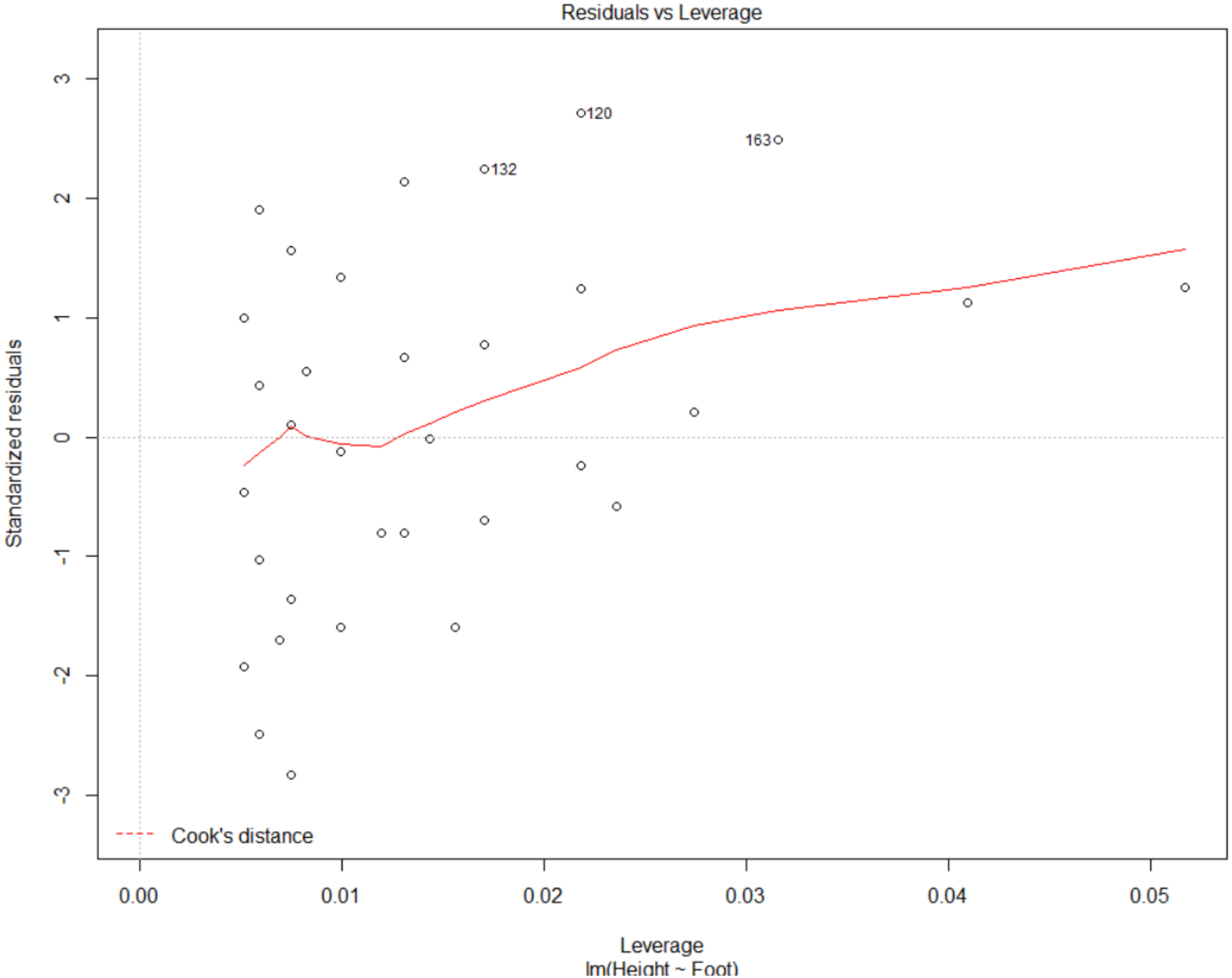
Plot(fit)


Normal Q-Q
Standardized residuals
3
2
1
0
-1
-2
-3
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Height ~ Foot)
120
17
171

Plot(fit)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
160
170
180
190
Fitted values
lm(Height ~ Foot)
120
177

Plot(fit)


Residuals vs Leverage
Standardized residuals
3
2
1
0
-1
-2
-3
120
163
132
Cook's distance
0.00
0.01
0.02
0.03
0.04
0.05
Leverage
lm(Height ~ Foot)

Leverage

- **Leverage** is a measure of how far away the independent variable values of an observation are from those of the other observations.
- The leverage of an observation measures its ability to move the regression model all by itself by simply moving in the y-direction. The leverage measures the amount by which the predicted value would change if the observation was shifted one unit in the y-direction.
- The leverage always takes values between 0 and 1. A point with zero leverage has no effect on the regression model. If a point has leverage equal to 1 the line must follow the point perfectly.

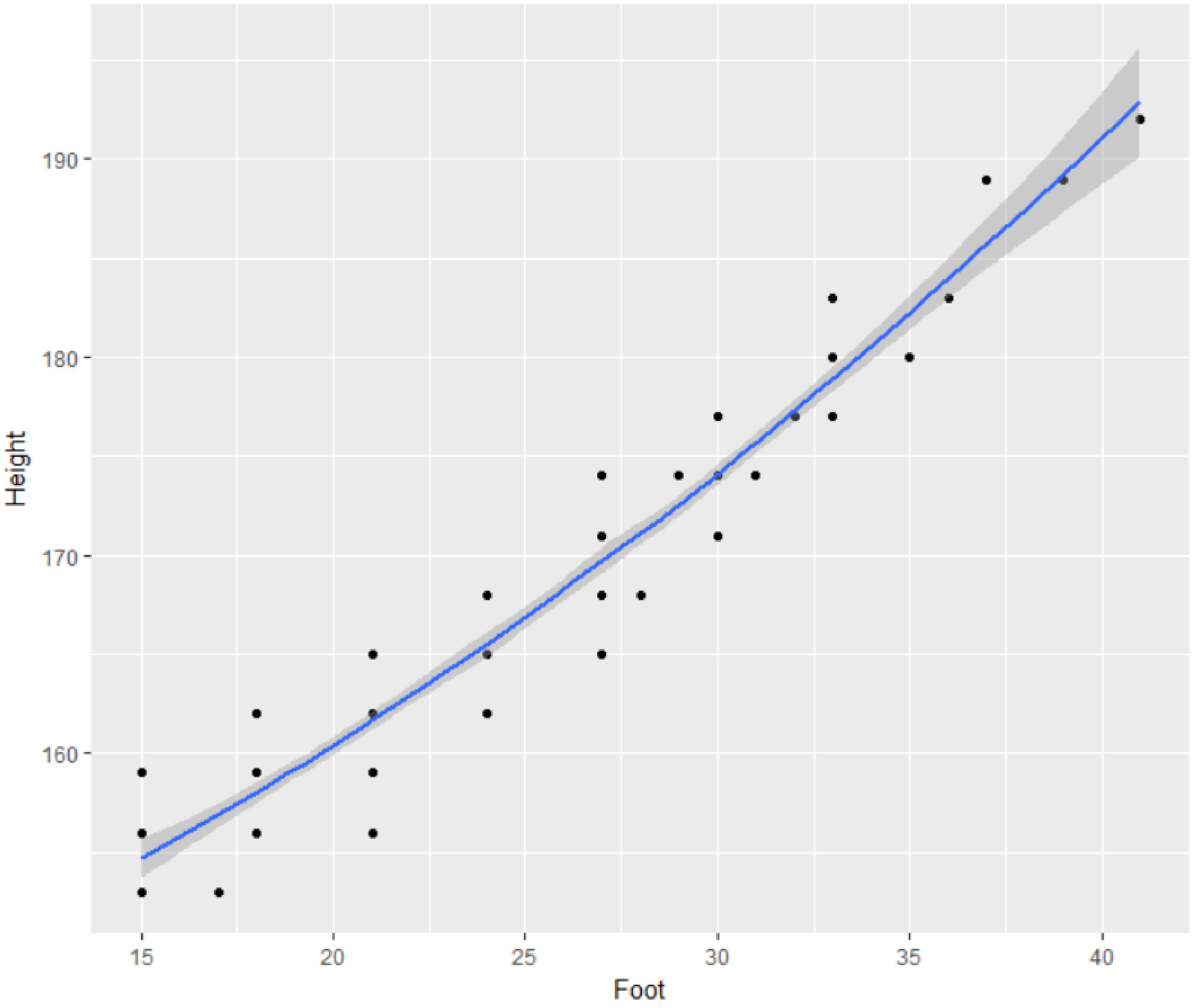
Cook’s Distance

- One measure of the leverage of a data point is *Cook’s distance*, an estimate of how much predicted (y) values would change if the model were re-estimated with that point eliminated from the data.

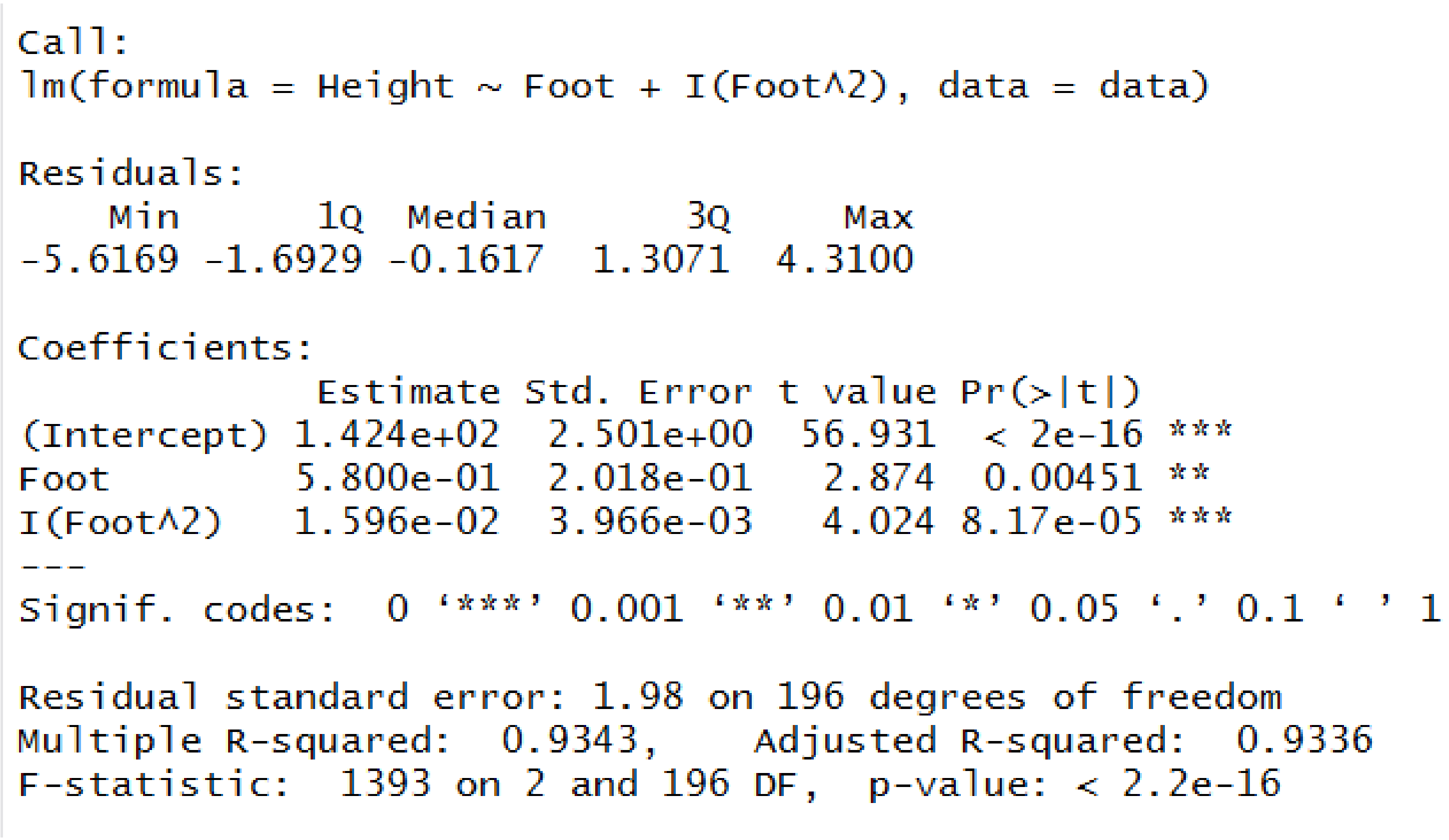
Outliers

- Outliers normally have high residual (indicating different pattern) and high leverage (indicating undue influence).

Can the model be improved?



Quadratic Model



```
Call:
lm(formula = Height ~ Foot + I(Foot^2), data = data)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.6169 -1.6929 -0.1617  1.3071  4.3100 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) 1.424e+02  2.501e+00  56.931  < 2e-16 ***
Foot        5.800e-01  2.018e-01   2.874  0.00451 ** 
I(Foot^2)   1.596e-02  3.966e-03   4.024 8.17e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.98 on 196 degrees of freedom
Multiple R-squared:  0.9343,	Adjusted R-squared:  0.9336 
F-statistic:  1393 on 2 and 196 DF,  p-value: < 2.2e-16
```

Height = 0.58 * Foot + 0.01596 * Foot^2 + 142.4

Multiple Regression with R

Linear Regression Procedure

- Inspect the data to make sure it is clean and has the structure you expect.
- Check the distributions of the variables to make sure they are not highly skewed. If one is skewed, consider transforming it.
- Examine the bivariate scatterplots and correlation matrix to see whether there are any extremely correlated variables. If so, omit some variables or consider transforming them if needed.
- If you wish to estimate coefficients on a consistent scale, standardize the data with scale().
- After fitting a model, check the residual quantiles in the output. The residuals show how well the model accounts for the individual observations.
- Check the standard model plots using plot(), which will help you judge whether a linear model is appropriate or whether there is nonlinearity, and will identify potential outliers in the data.
- Try several models and compare them for overall interpretability and model fit by inspecting the residuals' spread and overall R^2.
- Report the confidence intervals of the estimates with your interpretation and recommendations.

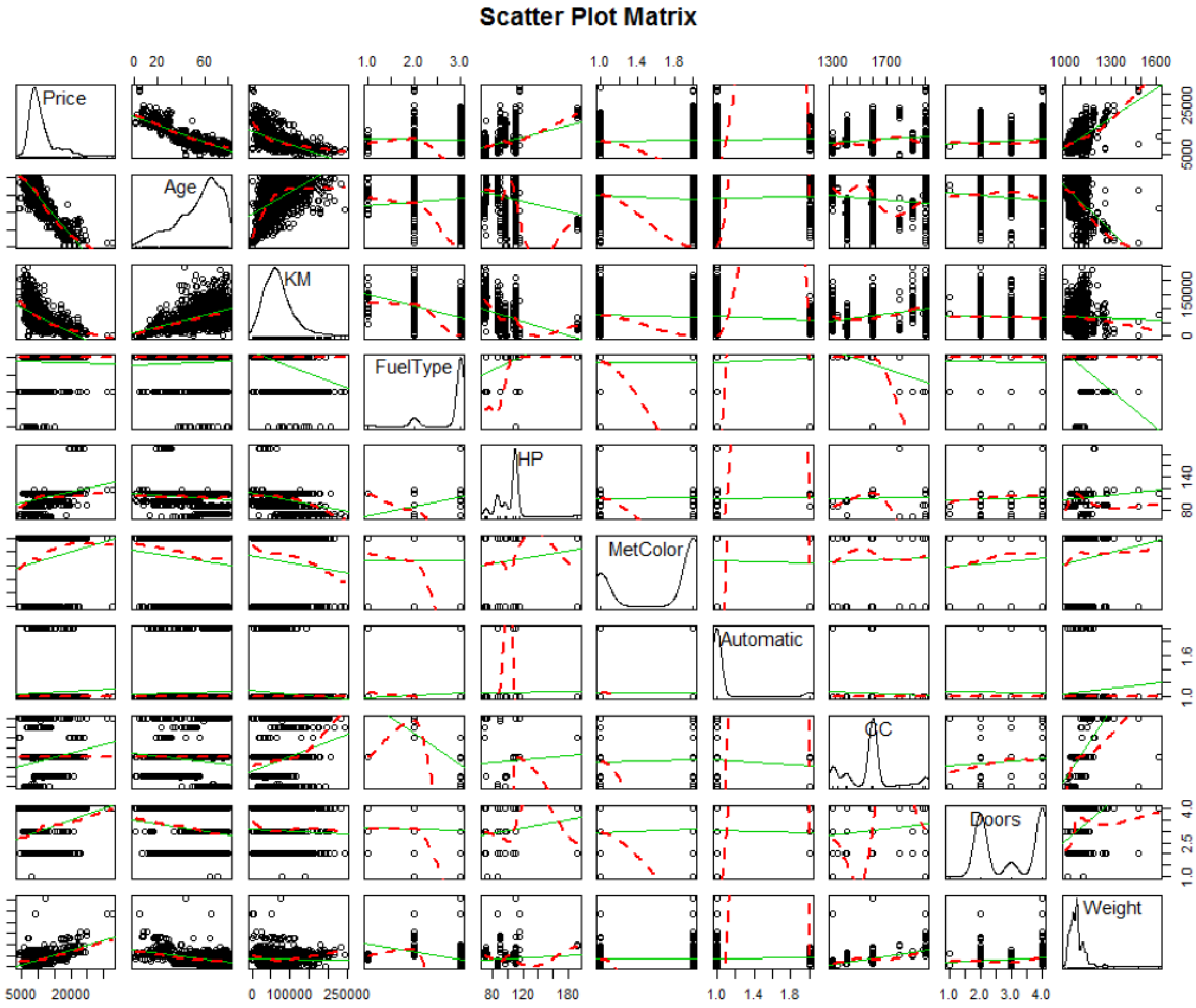
lm() format

Operator	Usage	Example
~	Separates dependent variable on the left from the independent variable(s) on the right.	y ~ x means the model is about relationship between y and x where y is dependent variable and x is independent variable.
+	Links several independent variables	y ~ x1+x2+x3 means there are three independent variables x1, x2, and x3.
:	Indicates an interaction between independent variables.	y~x1+x2+x1:x2 means we want to predict y using x1, x2 and their interaction.
*	Denotes factor crossing.	y ~ x1*x2 is equivalent to y ~ x1 + x2 + x1:x2; y ~ x1*x2*x3 is equivalent to y ~ x1 + x2 + x3 + x1:x2 + x1:x3 + x2:x3 + x1:x2:x3
^	Indicates crossing to a certain degree.	(a+b+c)^2 is identical to (a+b+c)*(a+b+c)
.	All variables in the data except the dependent variable.	If the data has four variables y, x1, x2, and x3, then y ~ . is identical to y ~ x1 + x2 + x3.
-	Removes the specific terms from the list	a*b - a is identical to b + a:b
-1	Suppress he intercept.	y ~ x -1 fits a regression of y on x and forces the line through the origin.
I()	The operator in I() will be treated as common arithmetic operator rather than special operator as listed above.	y ~ x1 + I((x1+x2)^2) models y against x1 and (x1+x2)^2

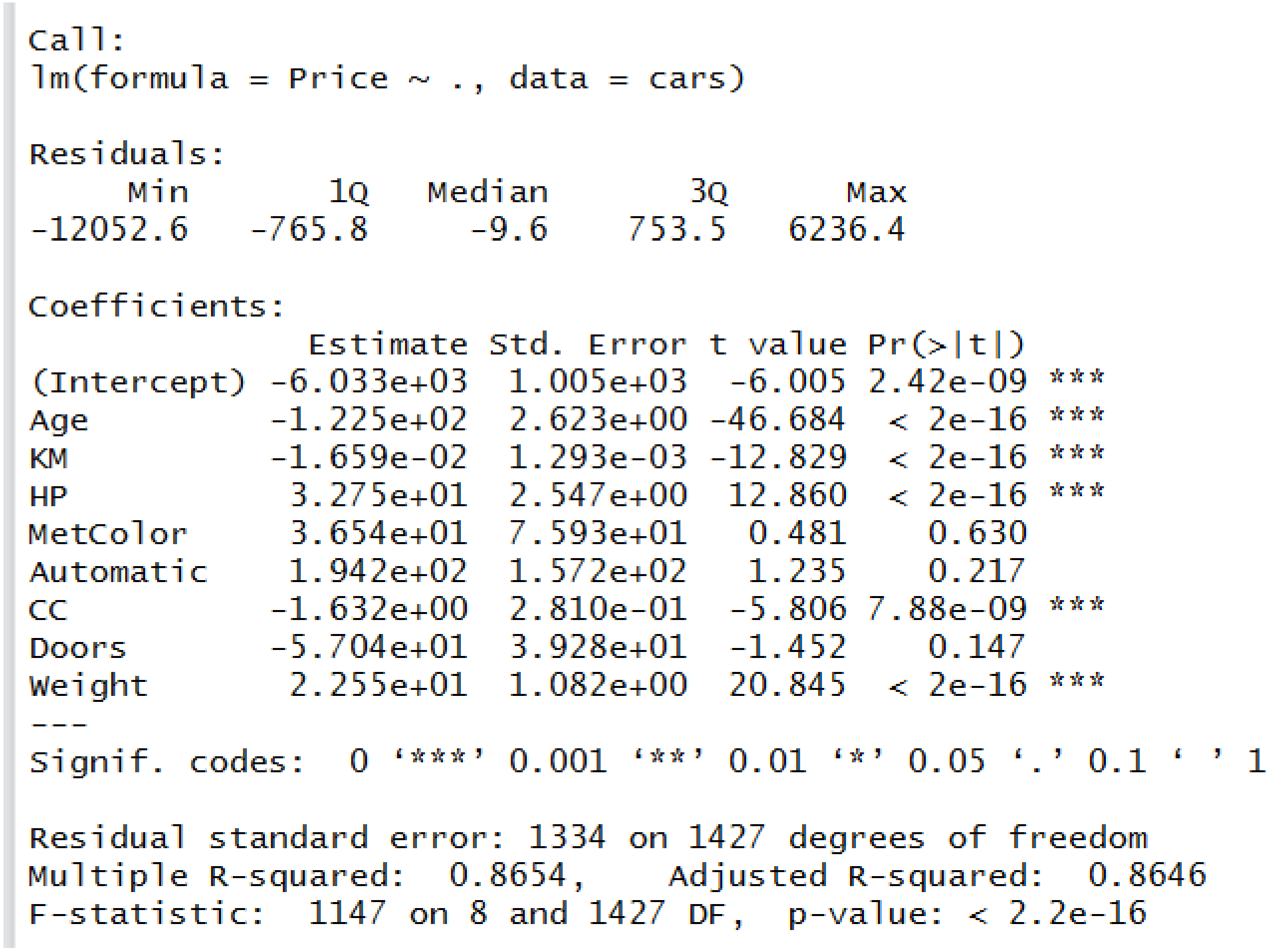
Functions on lm result

Function	Usage
summary()	Displays detailed information of the fitted model. This is normally the first one called after fitting.
coefficients()	Lists all the coefficients of the fitted model.
confint()	Provides confidence interval, 95% by default, for the fitted model.
residuals()	Lists the residual values in a fitted model.
fitted()	Lists the fitted values of the model
anova()	Generates an ANOVA table for a fitted model, or an ANOVA table comparing two or more fitted models.
vcov()	Lists the covariance matrix for model parameters.
AIC()	Prints Akaike's information criterion
plot()	Generates diagnostic plots for evaluating the fit of a model.
predict()	Uses a fitted model to predict response values for a new dataset.

What matters in second hand car price?

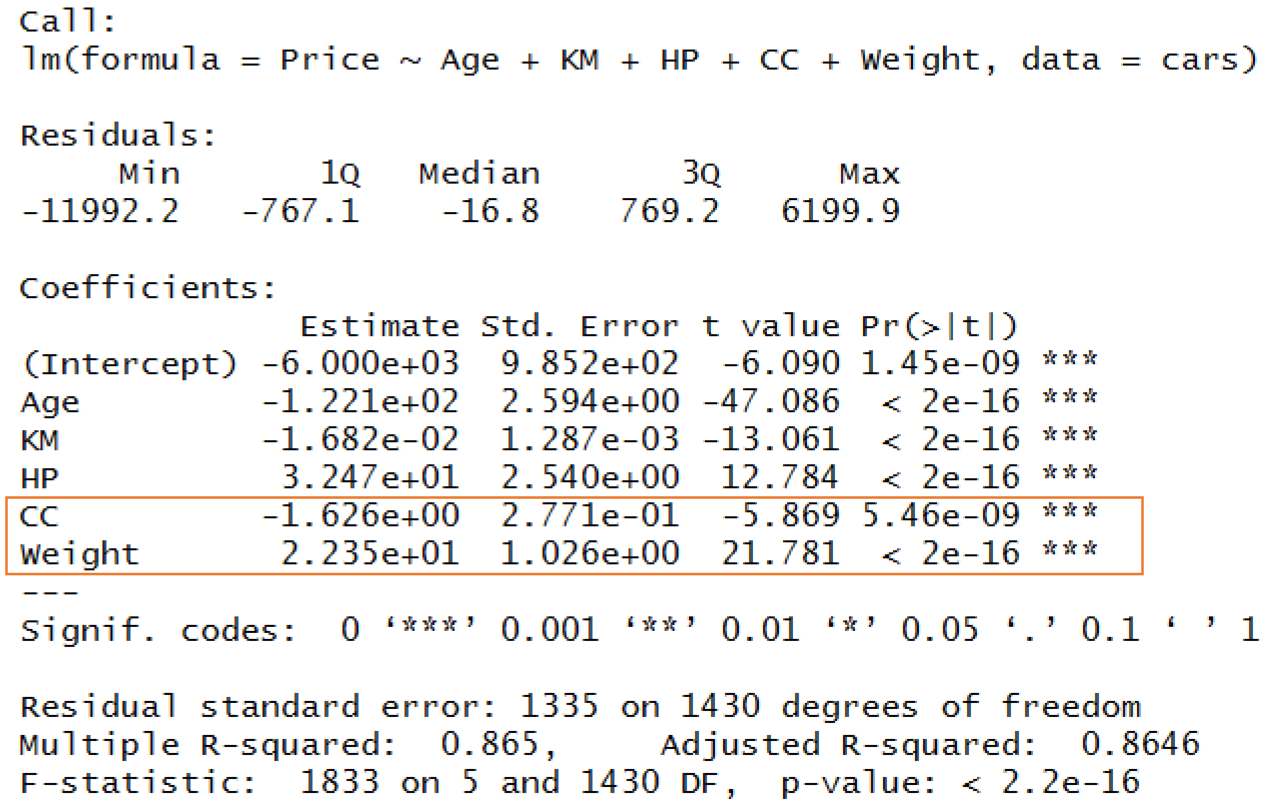

Scatter Plot Matrix
Price
Age
KM
FuelType
HP
MetColor
Automatic
CC
Doors
Weight
0 20 60
1.0 2.0 3.0
1.0 1.4 1.8
1300 1700
1000 1300 1600
5000 25000
80 40 0
0 150000
3.0 2.0 1.0
80 140
1.0 1.6
1.0 1.6
1300 1800
1.0 2.5 4.0
1000 1400
5000 20000
0 100000 250000
80 120 180
1.0 1.4 1.8
1.0 2.0 3.0 4.0

V1.0 – All Numeric



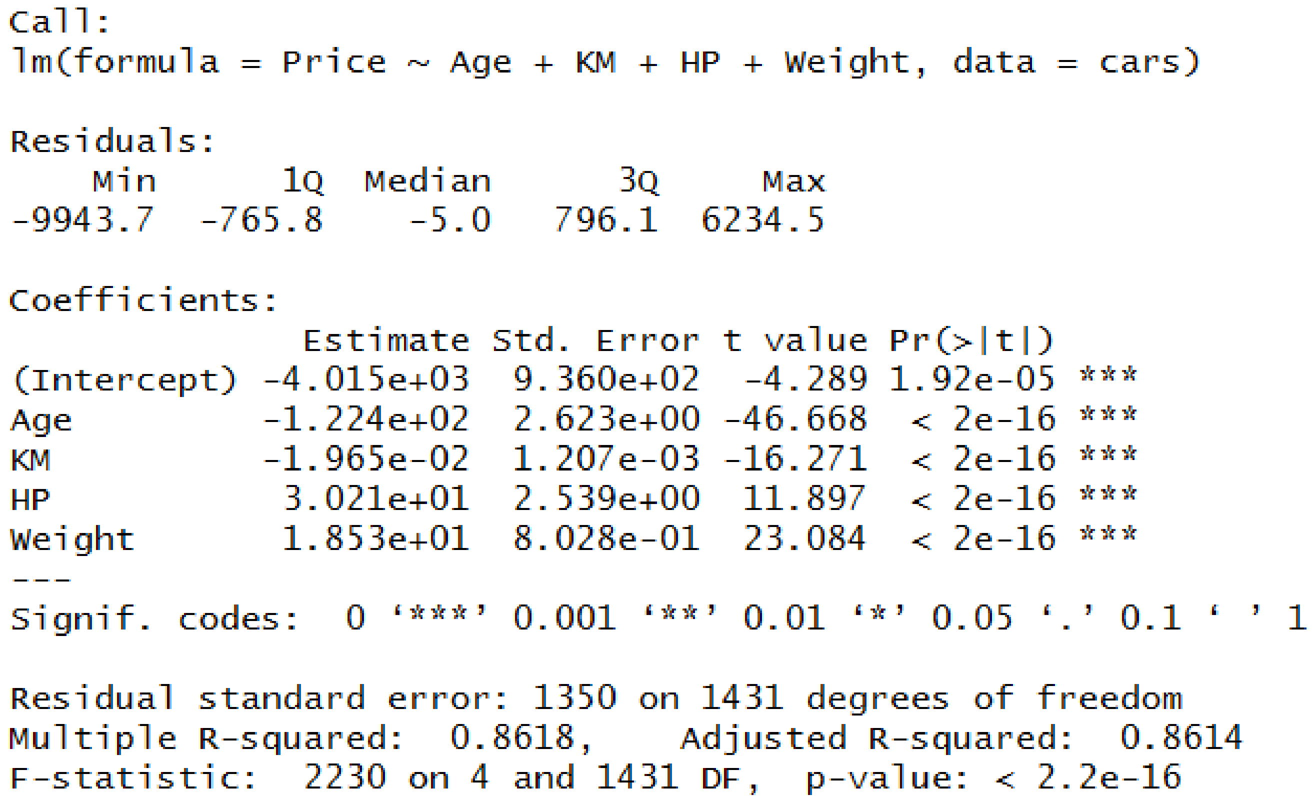
```
Call:
lm(formula = Price ~ ., data = cars)

Residuals:
     Min       1Q   Median       3Q      Max 
-12052.6   -765.8     -9.6    753.5   6236.4 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.033e+03  1.005e+03  -6.005 2.42e-09 ***
Age         -1.225e+02  2.623e+00 -46.684  < 2e-16 ***
KM          -1.659e-02  1.293e-03 -12.829  < 2e-16 ***
HP           3.275e+01  2.547e+00  12.860  < 2e-16 ***
MetColor     3.654e+01  7.593e+01   0.481    0.630    
Automatic    1.942e+02  1.572e+02   1.235    0.217    
CC          -1.632e+00  2.810e-01  -5.806 7.88e-09 ***
Doors       -5.704e+01  3.928e+01  -1.452    0.147    
Weight       2.255e+01  1.082e+00  20.845  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1334 on 1427 degrees of freedom
Multiple R-squared:  0.8654,	Adjusted R-squared:  0.8646 
F-statistic:  1147 on 8 and 1427 DF,  p-value: < 2.2e-16
```

V1.1 – Remove insignificant factors



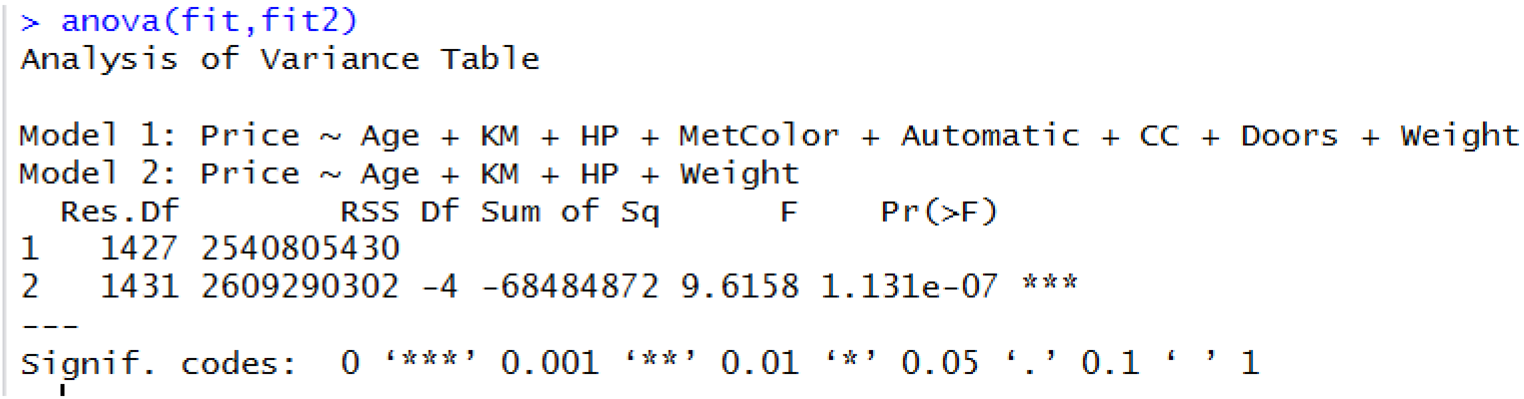
```
Call:
lm(formula = Price ~ Age + KM + HP + CC + Weight, data = cars)

Residuals:
     Min       1Q   Median       3Q      Max 
-11992.2   -767.1    -16.8    769.2   6199.9 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.000e+03  9.852e+02  -6.090 1.45e-09 ***
Age         -1.221e+02  2.594e+00 -47.086  < 2e-16 ***
KM          -1.682e-02  1.287e-03 -13.061  < 2e-16 ***
HP           3.247e+01  2.540e+00  12.784  < 2e-16 ***
CC          -1.626e+00  2.771e-01  -5.869 5.46e-09 ***
Weight       2.235e+01  1.026e+00  21.781  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1335 on 1430 degrees of freedom
Multiple R-squared:  0.865,	Adjusted R-squared:  0.8646 
F-statistic:  1833 on 5 and 1430 DF,  p-value: < 2.2e-16
```

V1.2 – Remove highly correlated factors



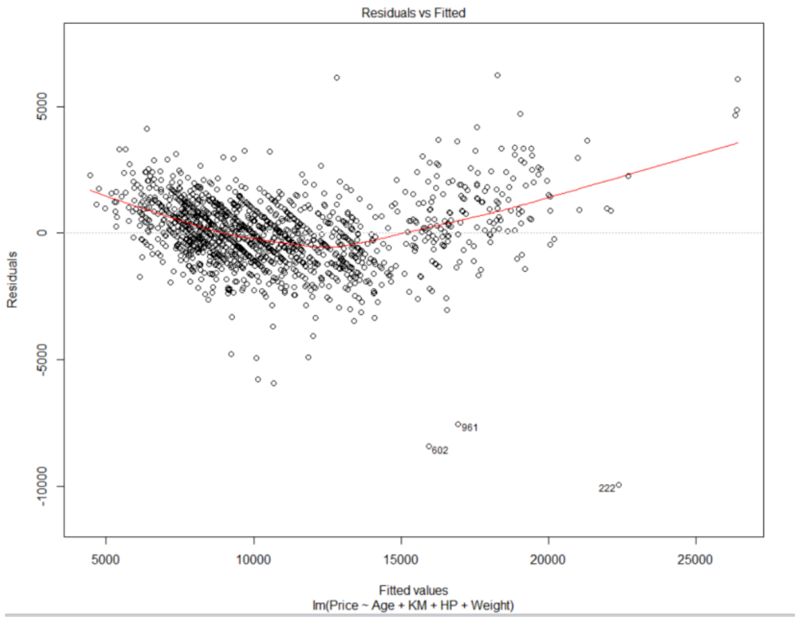
```
Call:
lm(formula = Price ~ Age + KM + HP + Weight, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-9943.7  -765.8    -5.0   796.1  6234.5 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.015e+03  9.360e+02  -4.289 1.92e-05 ***
Age         -1.224e+02  2.623e+00 -46.668  < 2e-16 ***
KM          -1.965e-02  1.207e-03 -16.271  < 2e-16 ***
HP           3.021e+01  2.539e+00  11.897  < 2e-16 ***
Weight       1.853e+01  8.028e-01  23.084  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1350 on 1431 degrees of freedom
Multiple R-squared:  0.8618,    Adjusted R-squared:  0.8614 
F-statistic:  2230 on 4 and 1431 DF,  p-value: < 2.2e-16
```

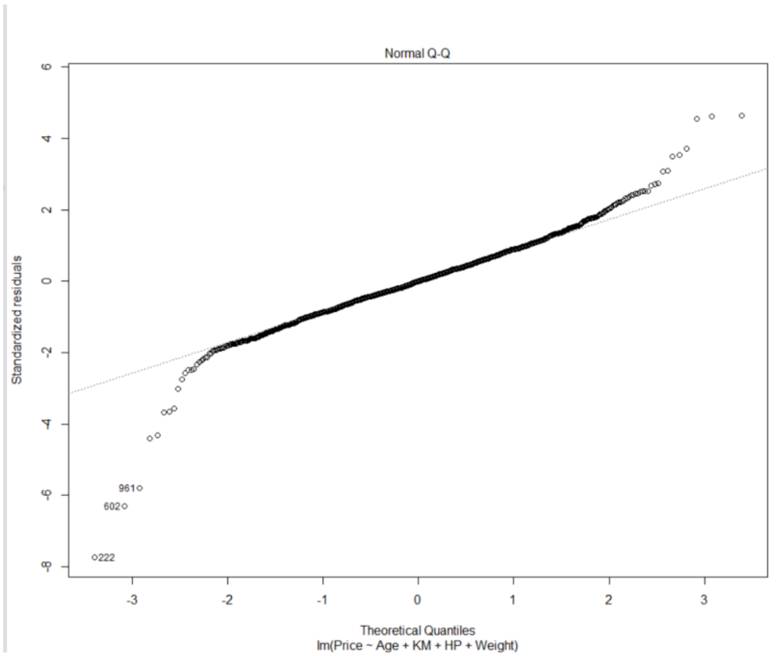
Comparing Models

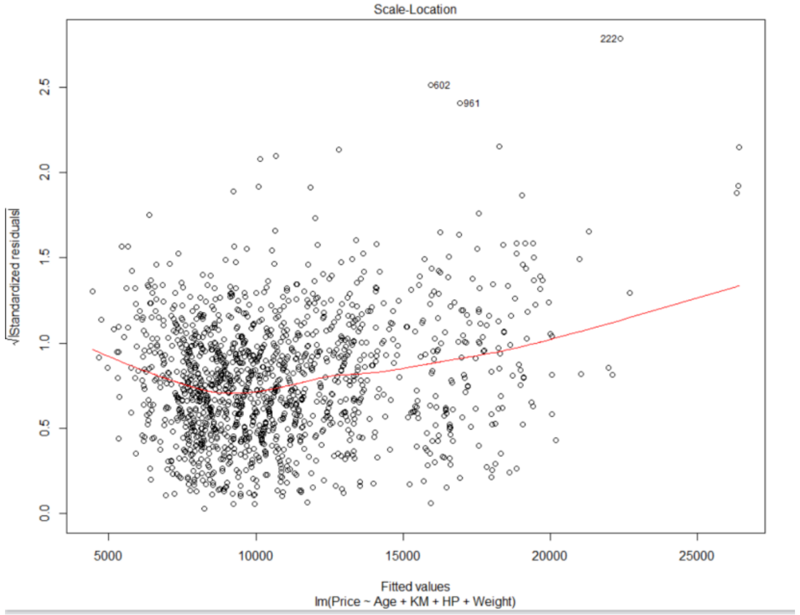


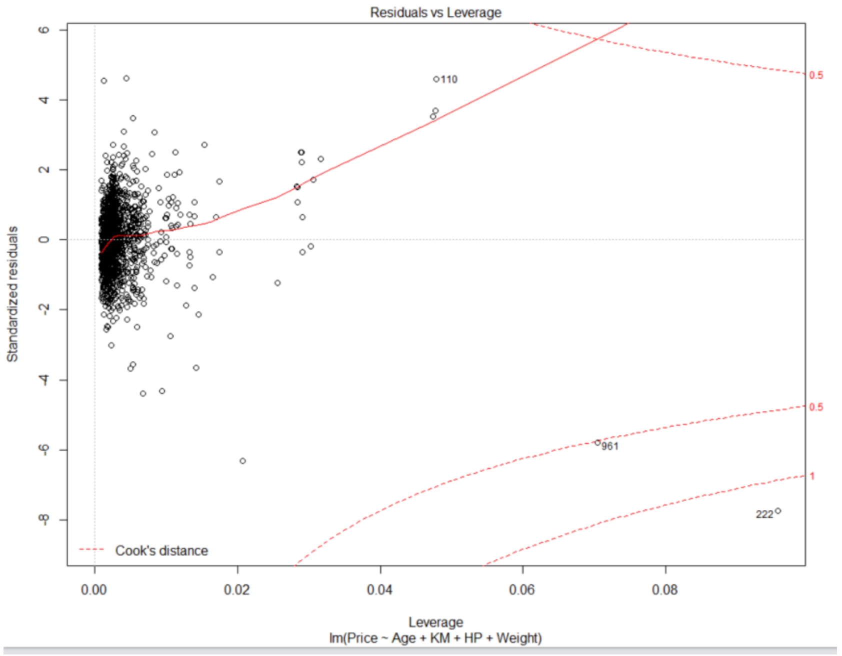
```
> anova(fit,fit2)
Analysis of Variance Table

Model 1: Price ~ Age + KM + HP + MetColor + Automatic + CC + Doors + Weight
Model 2: Price ~ Age + KM + HP + Weight
  Res.Df        RSS Df Sum of Sq      F    Pr(>F)    
1   1427 2540805430                                   
2   1431 2609290302 -4 -68484872 9.6158 1.131e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Regression Analysis





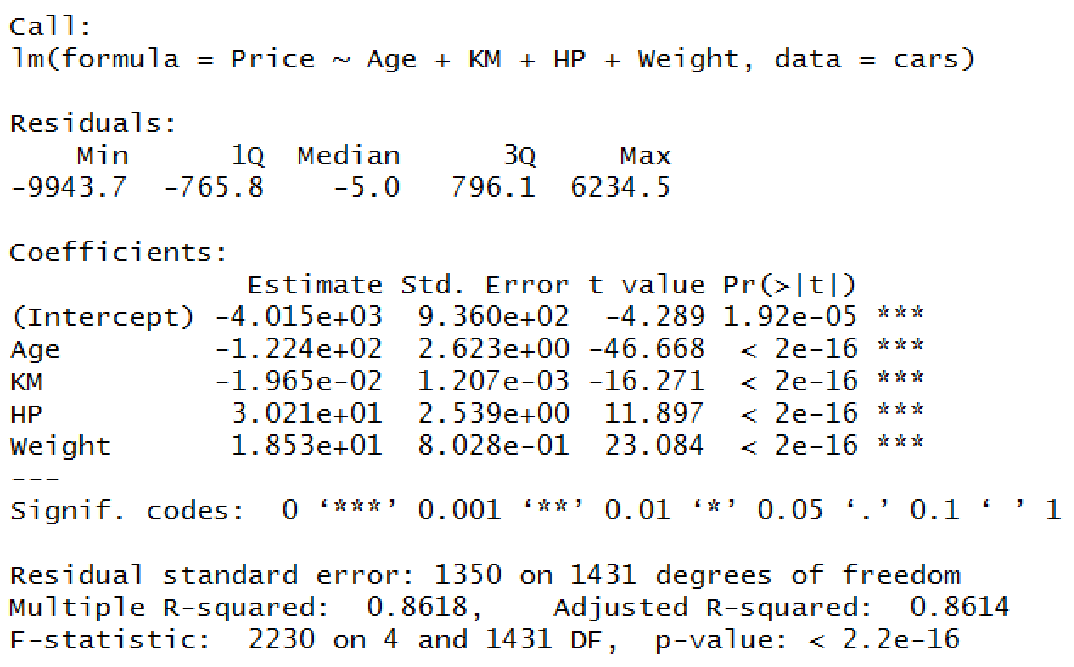




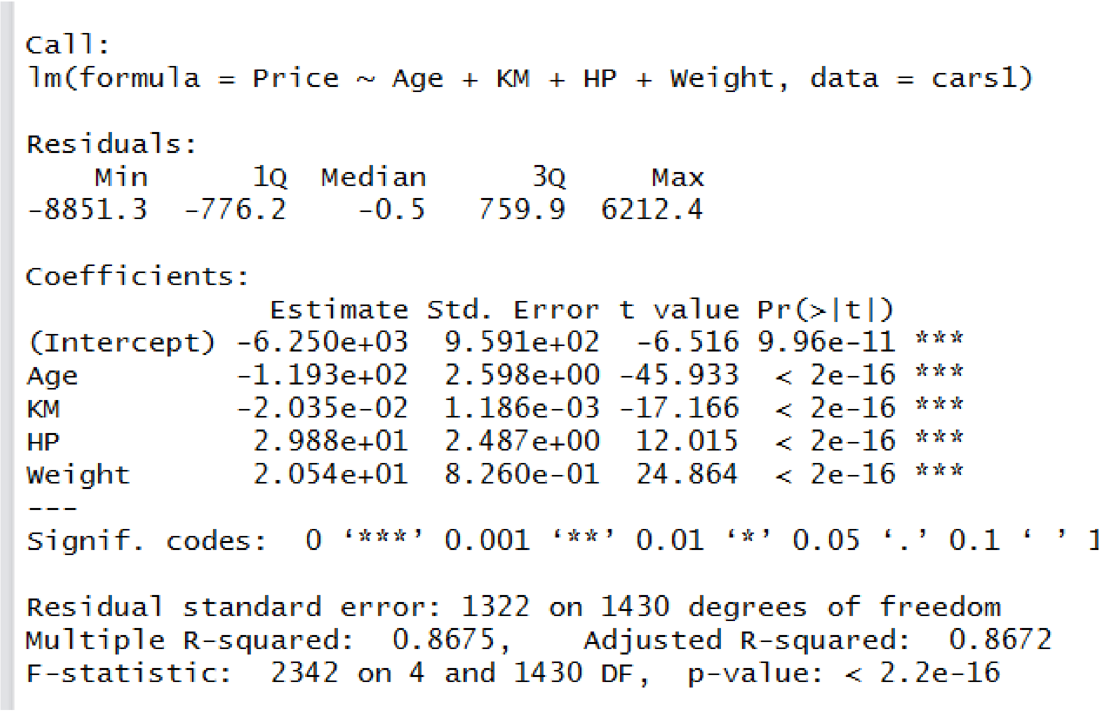
Problematic Points

- **Outliers**: an outlier is defined as an observation that has a large residual. In other words, the observed value for the point is very different from that predicted by the regression model.
- **Leverage points**: A leverage point is defined as an observation that has a value of x that is far away from the mean of x.
- **Influential observations**: An influential observation is defined as an observation that changes the slope of the line. Thus, influential points have a large influence on the fit of the model. One method to find influential points is to compare the fit of the model with and without each observation.
- **Outliers may or may not be influential points**. Influential outliers are of the greatest concern. They should never be disregarded. Careful scrutiny of the original data may reveal an error in data entry that can be corrected. If they remain excluded from the final fitted model, they must be noted in the final report or paper.

Remove point 222 to see the impact

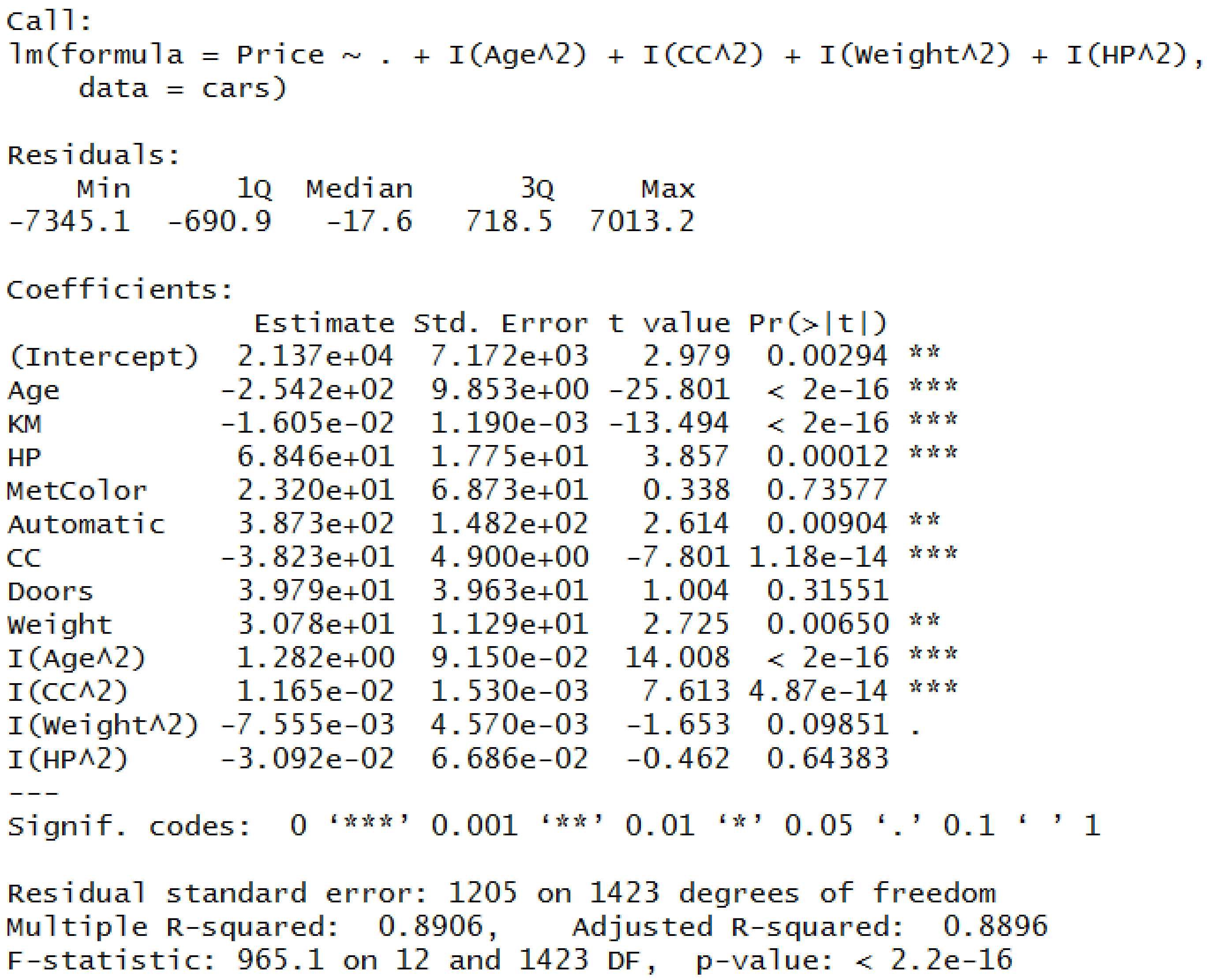


```
Call:
lm(formula = Price ~ Age + KM + HP + Weight, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-9943.7  -765.8    -5.0   796.1  6234.5 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -4.015e+03  9.360e+02  -4.289 1.92e-05 ***
Age         -1.224e+02  2.623e+00 -46.668  < 2e-16 ***
KM          -1.965e-02  1.207e-03 -16.271  < 2e-16 ***
HP           3.021e+01  2.539e+00  11.897  < 2e-16 ***
Weight       1.853e+01  8.028e-01  23.084  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1350 on 1431 degrees of freedom
Multiple R-squared:  0.8618,	Adjusted R-squared:  0.8614 
F-statistic:  2230 on 4 and 1431 DF,  p-value: < 2.2e-16
```



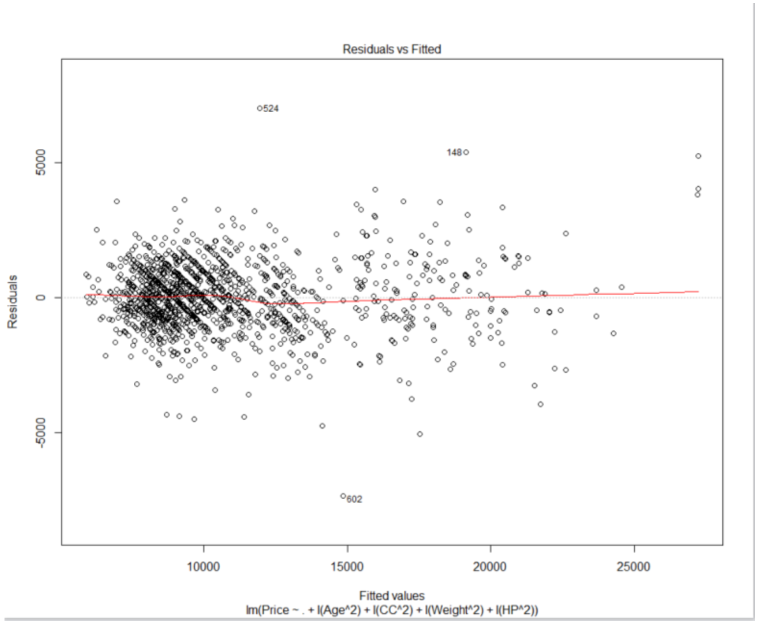
```
Call:
lm(formula = Price ~ Age + KM + HP + Weight, data = cars1)

Residuals:
    Min      1Q  Median      3Q     Max 
-8851.3  -776.2    -0.5   759.9  6212.4 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -6.250e+03  9.591e+02  -6.516 9.96e-11 ***
Age         -1.193e+02  2.598e+00 -45.933  < 2e-16 ***
KM          -2.035e-02  1.186e-03 -17.166  < 2e-16 ***
HP           2.988e+01  2.487e+00  12.015  < 2e-16 ***
Weight       2.054e+01  8.260e-01  24.864  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1322 on 1430 degrees of freedom
Multiple R-squared:  0.8675,	Adjusted R-squared:  0.8672 
F-statistic:  2342 on 4 and 1430 DF,  p-value: < 2.2e-16
```

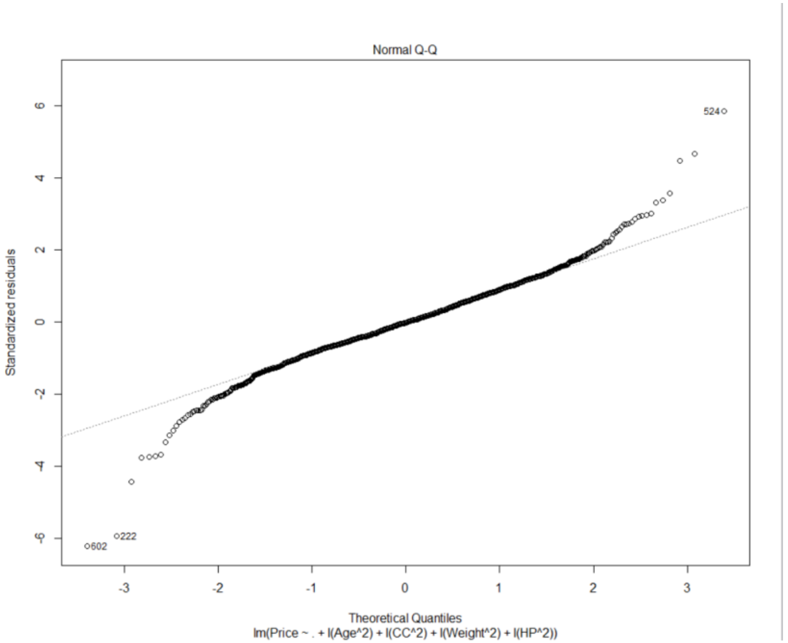
V2 – Handling Non-linearity

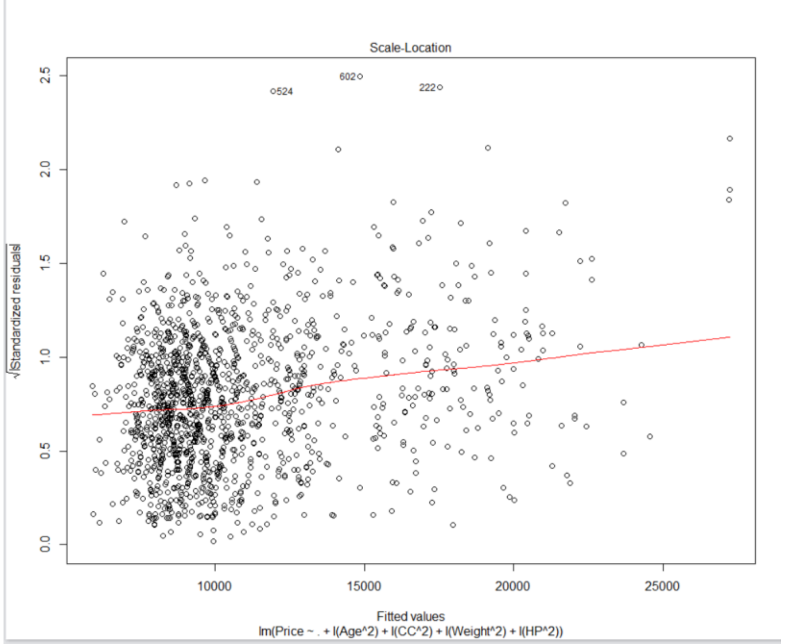


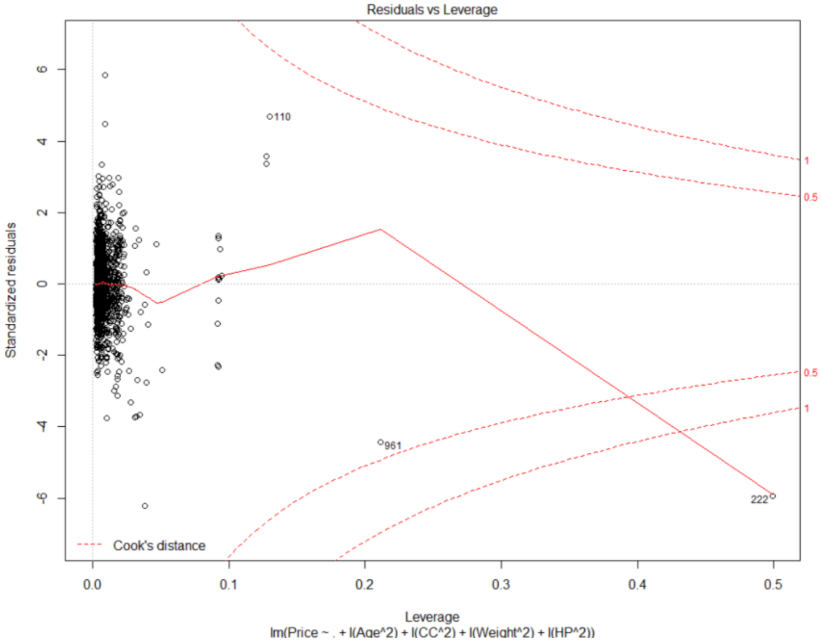
```
Call:
lm(formula = Price ~ . + I(Age^2) + I(CC^2) + I(Weight^2) + I(HP^2), 
    data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-7345.1  -690.9   -17.6   718.5  7013.2 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.137e+04  7.172e+03   2.979  0.00294 ** 
Age         -2.542e+02  9.853e+00 -25.801  < 2e-16 ***
KM          -1.605e-02  1.190e-03 -13.494  < 2e-16 ***
HP           6.846e+01  1.775e+01   3.857  0.00012 ***
MetColor     2.320e+01  6.873e+01   0.338  0.73577    
Automatic    3.873e+02  1.482e+02   2.614  0.00904 ** 
CC          -3.823e+01  4.900e+00  -7.801 1.18e-14 ***
Doors        3.979e+01  3.963e+01   1.004  0.31551    
Weight       3.078e+01  1.129e+01   2.725  0.00650 ** 
I(Age^2)     1.282e+00  9.150e-02  14.008  < 2e-16 ***
I(CC^2)      1.165e-02  1.530e-03   7.613 4.87e-14 ***
I(Weight^2) -7.555e-03  4.570e-03  -1.653  0.09851 .  
I(HP^2)     -3.092e-02  6.686e-02  -0.462  0.64383    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1205 on 1423 degrees of freedom
Multiple R-squared:  0.8906,	Adjusted R-squared:  0.8896 
F-statistic: 965.1 on 12 and 1423 DF,  p-value: < 2.2e-16
```

Plot(fit2.0)

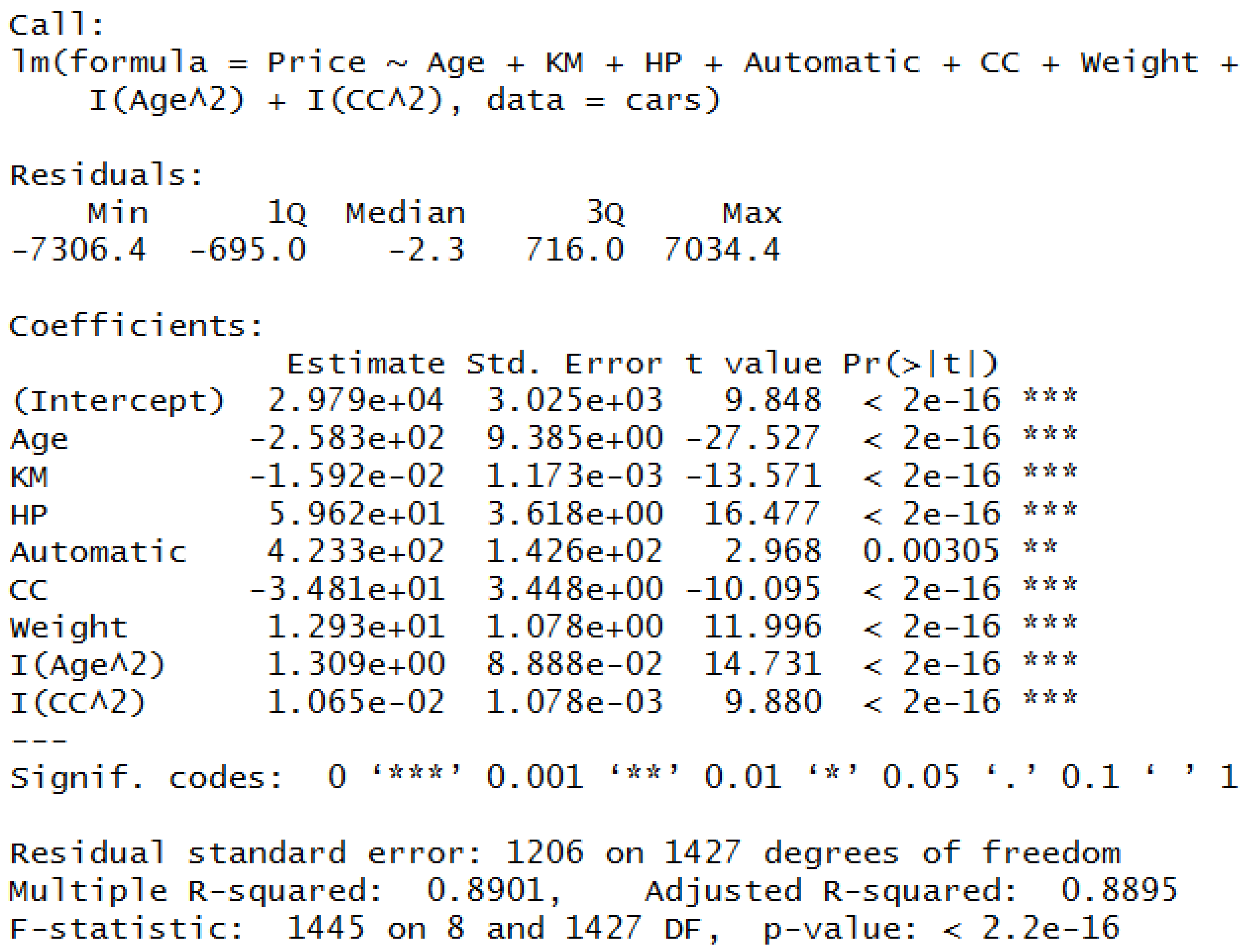

Residuals vs Fitted
Residuals
5000
0
-5000
524
148
602
10000
15000
20000
25000
Fitted values
lm(Price ~ . + I(Age^2) + I(CC^2) + I(Weight^2) + I(HP^2))


Normal Q-Q
Standardized residuals
6
4
2
0
-2
-4
-6
524
222
602
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Price ~ . + I(Age^2) + I(CC^2) + I(Weight^2) + I(HP^2))


Scale-Location
√|Standardized residuals|
2.5
2.0
1.5
1.0
0.5
0.0
524
602
222
10000
15000
20000
25000
Fitted values
lm(Price ~ . + I(Age^2) + I(CC^2) + I(Weight^2) + I(HP^2))

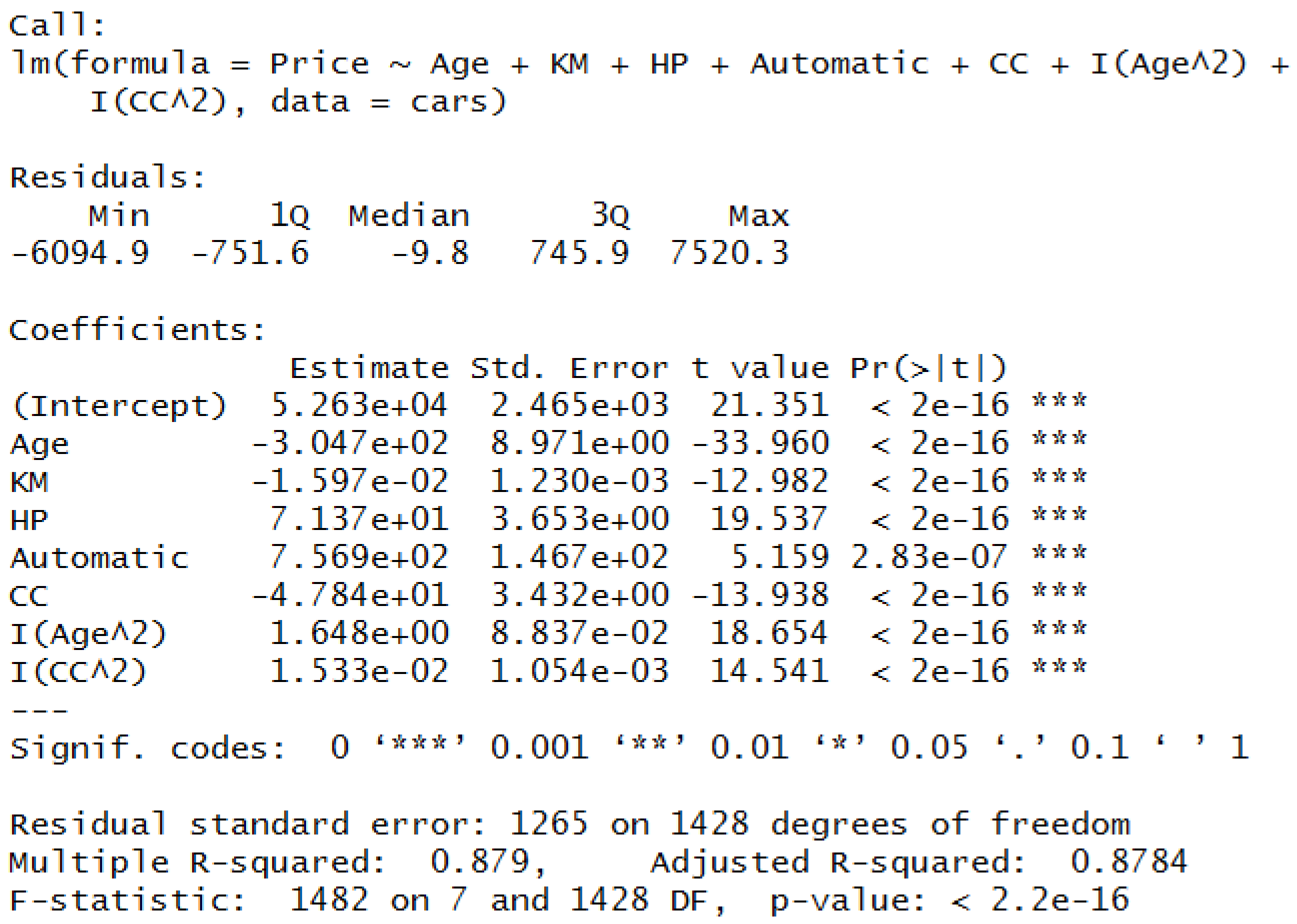

Residuals vs Leverage
Standardized residuals
6
4
2
0
-2
-4
-6
110
961
222
1
0.5
0.5
1
Cook's distance
0.0
0.1
0.2
0.3
0.4
0.5
Leverage
lm(Price ~ . + I(Age^2) + I(CC^2) + I(Weight^2) + I(HP^2))

V2.1 – Remove insignificant factors



```
Call:
lm(formula = Price ~ Age + KM + HP + Automatic + CC + Weight + 
    I(Age^2) + I(CC^2), data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-7306.4  -695.0    -2.3   716.0  7034.4 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  2.979e+04  3.025e+03   9.848  < 2e-16 ***
Age         -2.583e+02  9.385e+00 -27.527  < 2e-16 ***
KM          -1.592e-02  1.173e-03 -13.571  < 2e-16 ***
HP           5.962e+01  3.618e+00  16.477  < 2e-16 ***
Automatic    4.233e+02  1.426e+02   2.968  0.00305 ** 
CC          -3.481e+01  3.448e+00 -10.095  < 2e-16 ***
Weight       1.293e+01  1.078e+00  11.996  < 2e-16 ***
I(Age^2)     1.309e+00  8.888e-02  14.731  < 2e-16 ***
I(CC^2)      1.065e-02  1.078e-03   9.880  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1206 on 1427 degrees of freedom
Multiple R-squared:  0.8901,	Adjusted R-squared:  0.8895 
F-statistic:  1445 on 8 and 1427 DF,  p-value: < 2.2e-16
```

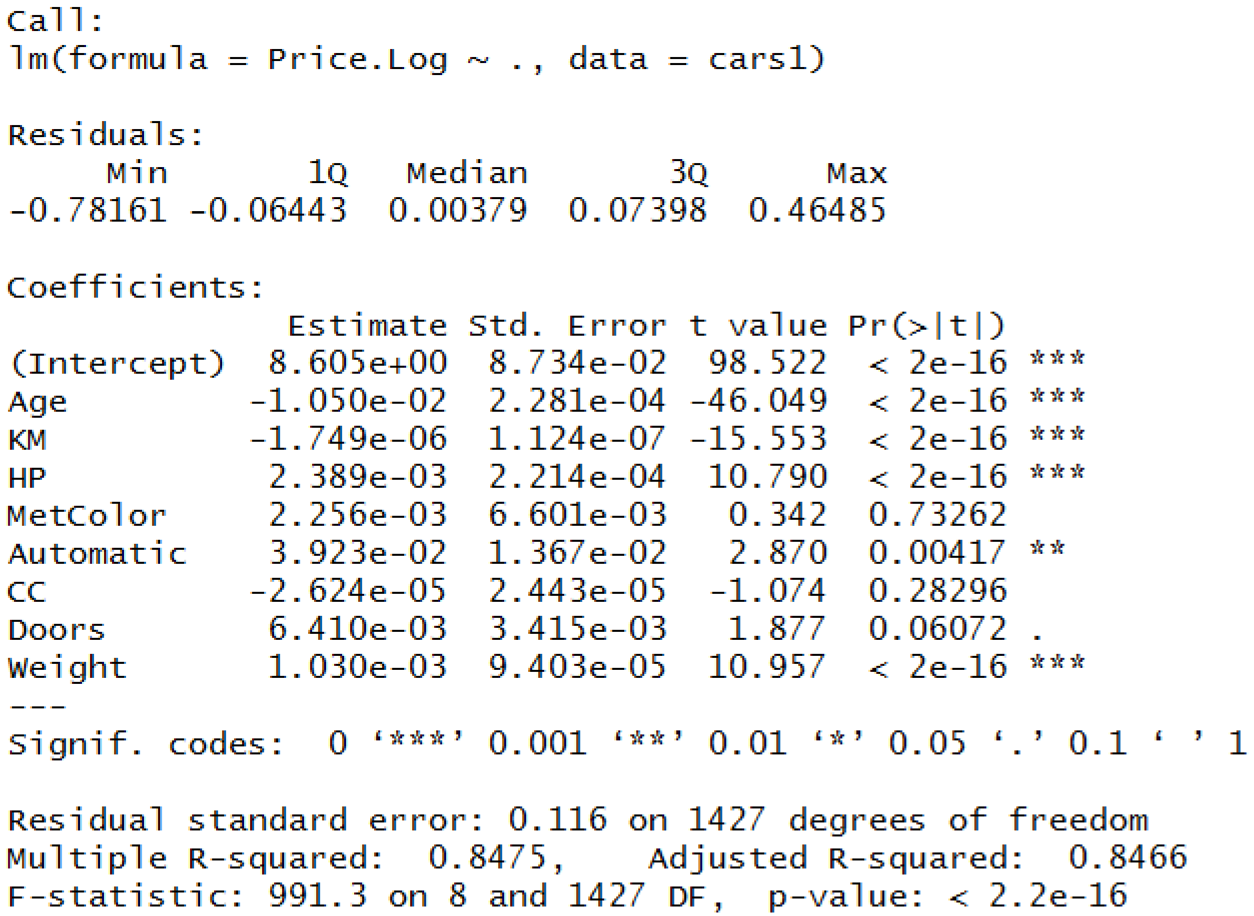
V2.2 – Remove highly correlated factors



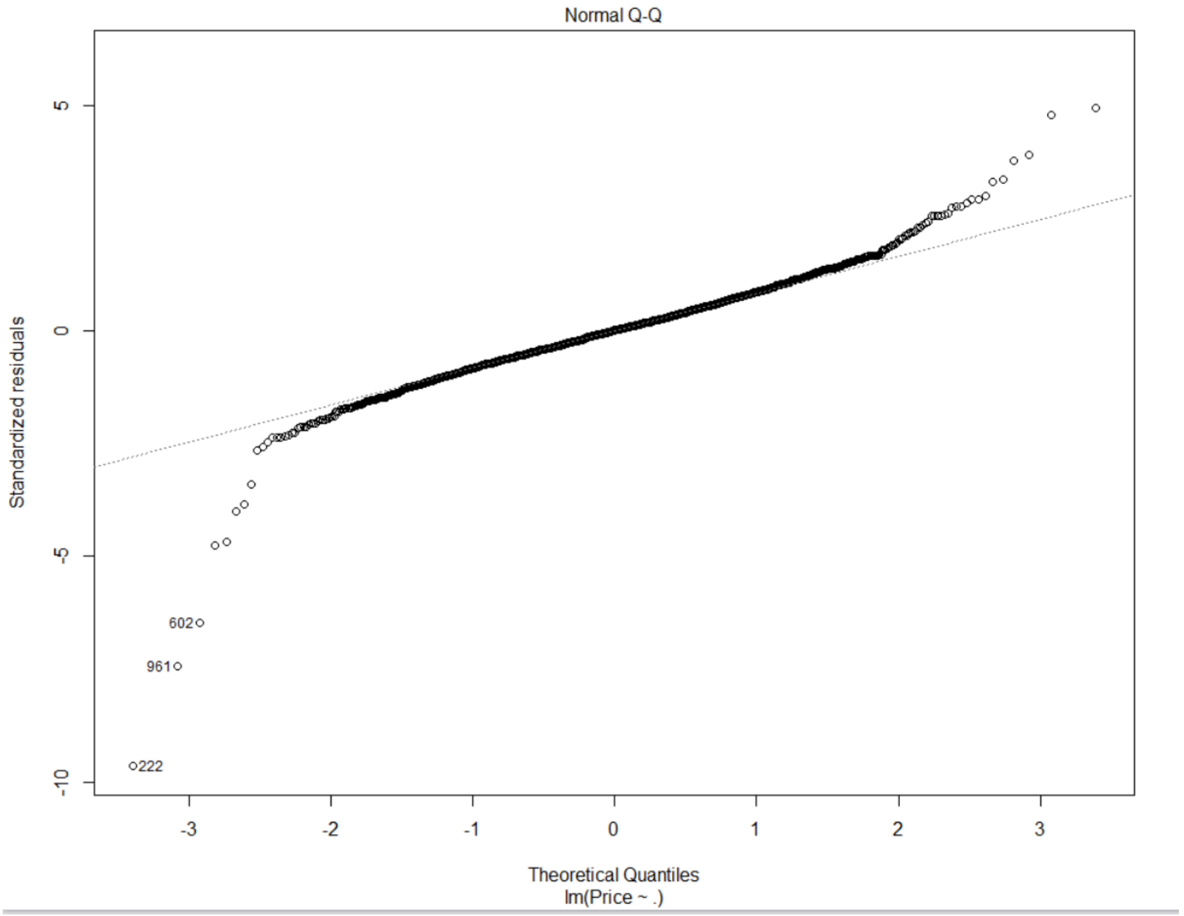
```
Call:
lm(formula = Price ~ Age + KM + HP + Automatic + CC + I(Age^2) + 
    I(CC^2), data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-6094.9  -751.6    -9.8   745.9  7520.3 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  5.263e+04  2.465e+03  21.351  < 2e-16 ***
Age         -3.047e+02  8.971e+00 -33.960  < 2e-16 ***
KM          -1.597e-02  1.230e-03 -12.982  < 2e-16 ***
HP           7.137e+01  3.653e+00  19.537  < 2e-16 ***
Automatic    7.569e+02  1.467e+02   5.159 2.83e-07 ***
CC          -4.784e+01  3.432e+00 -13.938  < 2e-16 ***
I(Age^2)     1.648e+00  8.837e-02  18.654  < 2e-16 ***
I(CC^2)      1.533e-02  1.054e-03  14.541  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1265 on 1428 degrees of freedom
Multiple R-squared:  0.879,	Adjusted R-squared:  0.8784 
F-statistic:  1482 on 7 and 1428 DF,  p-value: < 2.2e-16
```

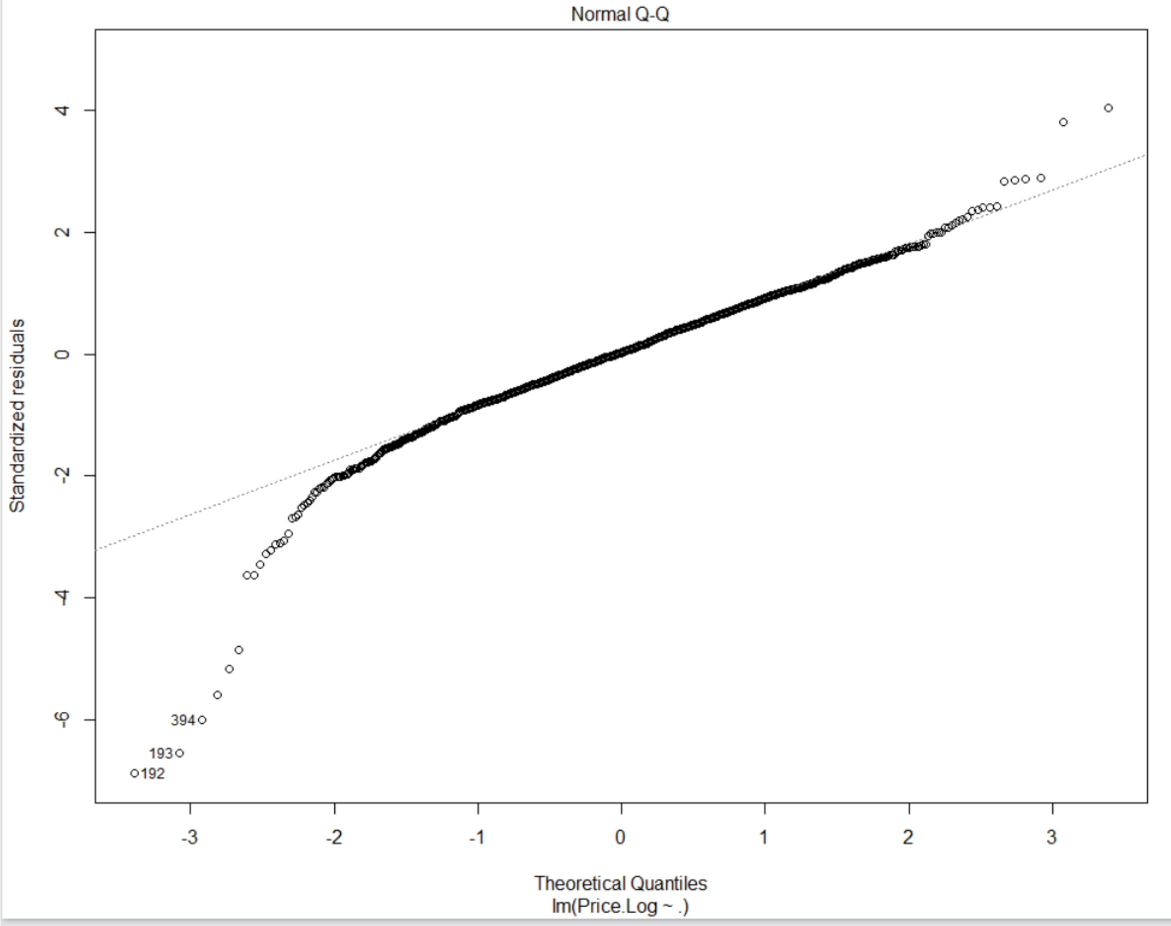
V3 – Handling Normality

```
cars1 <- cars
cars1$Price.Log <- log(cars1$Price)
cars1$Price <- NULL
```



```
Call:
lm(formula = Price.Log ~ ., data = cars1)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.78161 -0.06443  0.00379  0.07398  0.46485 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  8.605e+00  8.734e-02  98.522  < 2e-16 ***
Age         -1.050e-02  2.281e-04 -46.049  < 2e-16 ***
KM          -1.749e-06  1.124e-07 -15.553  < 2e-16 ***
HP           2.389e-03  2.214e-04  10.790  < 2e-16 ***
MetColor     2.256e-03  6.601e-03   0.342  0.73262    
Automatic    3.923e-02  1.367e-02   2.870  0.00417 ** 
CC          -2.624e-05  2.443e-05  -1.074  0.28296    
Doors        6.410e-03  3.415e-03   1.877  0.06072 .  
Weight       1.030e-03  9.403e-05  10.957  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.116 on 1427 degrees of freedom
Multiple R-squared:  0.8475,	Adjusted R-squared:  0.8466 
F-statistic: 991.3 on 8 and 1427 DF,  p-value: < 2.2e-16
```

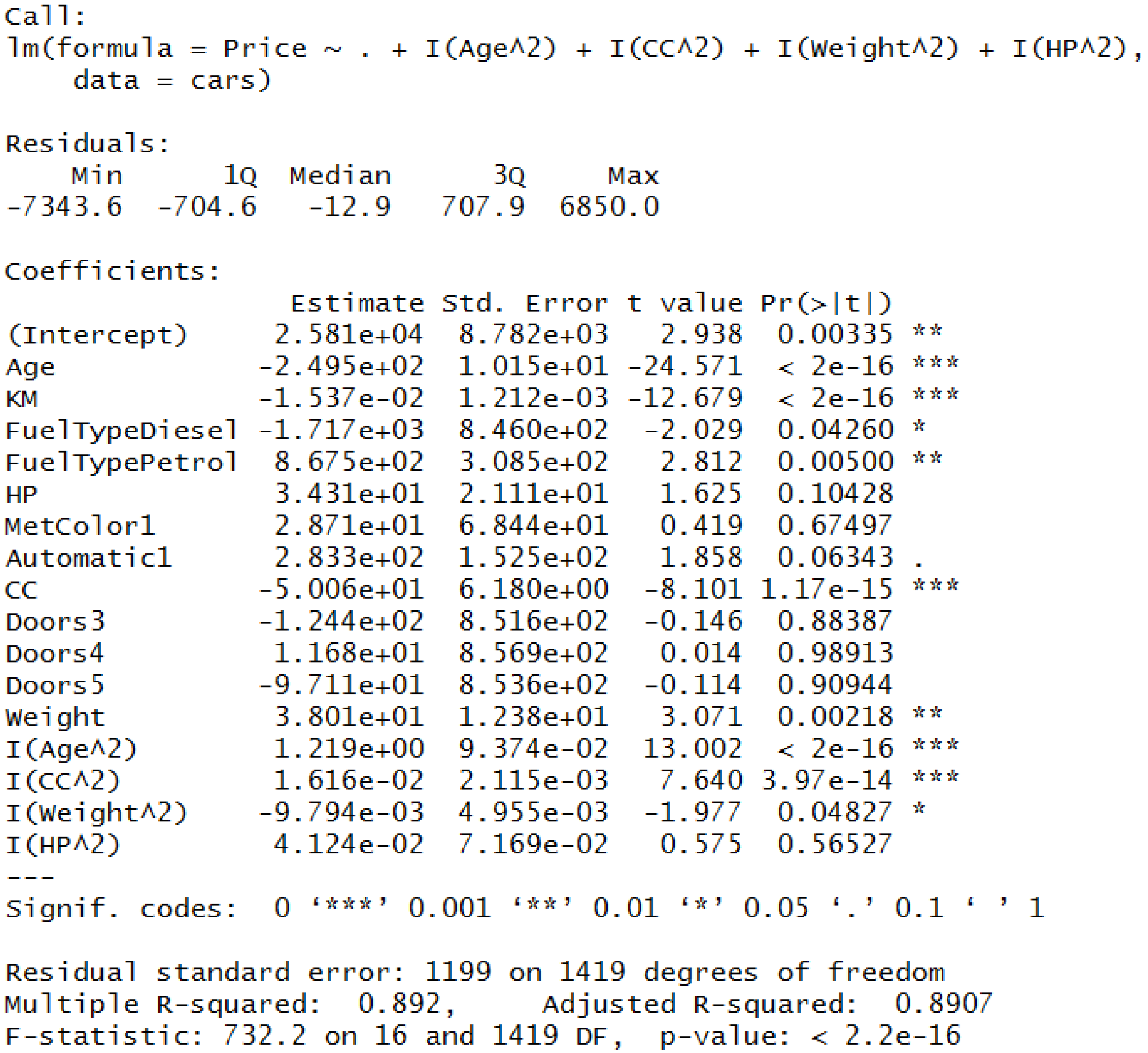

Normal Q-Q
Standardized residuals
5
0
-5
-10
602
961
222
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Price ~ .)


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-6
394
193
192
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Price.Log ~ .)

V4 – Handling Categorical Data

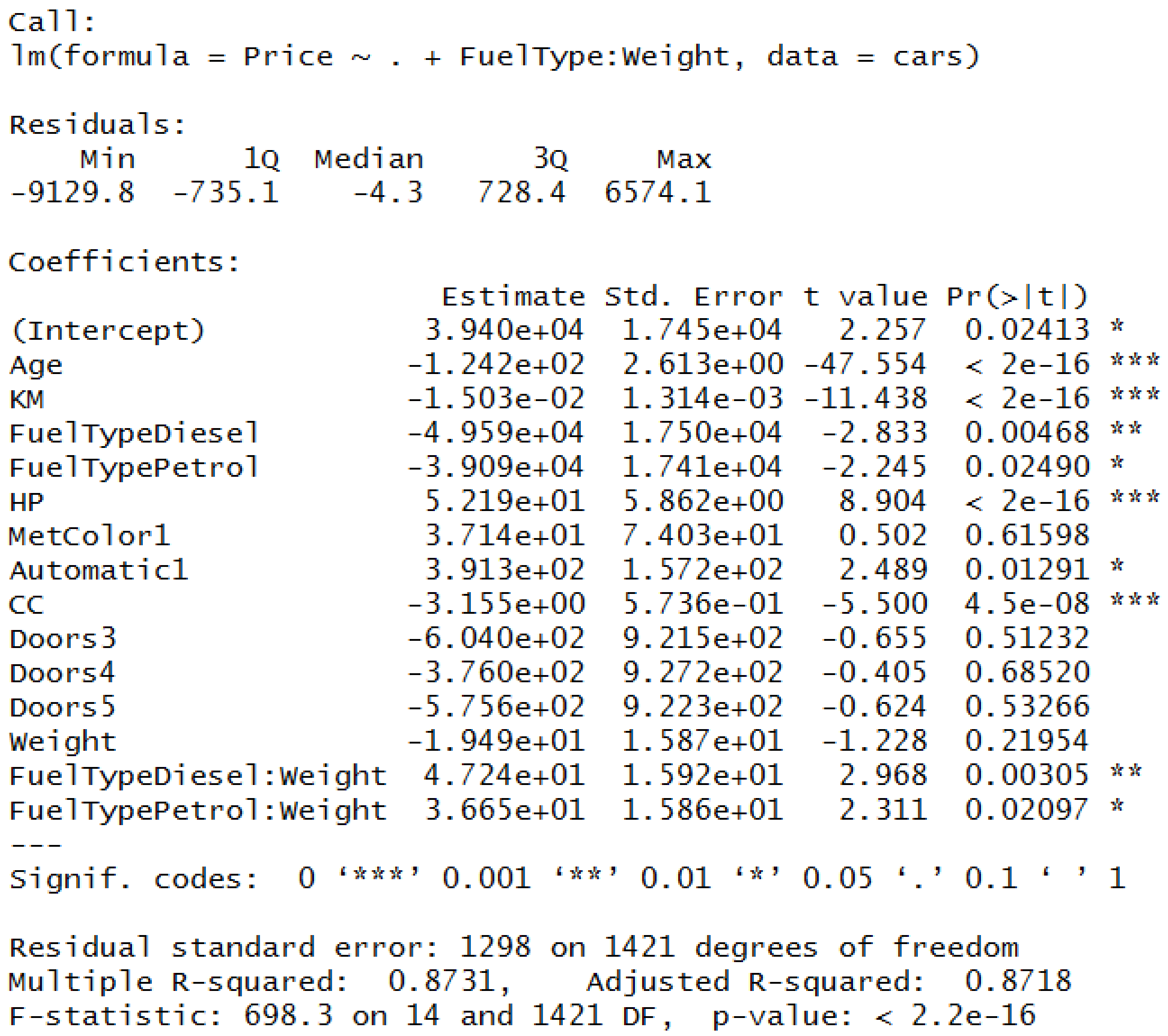
```
cars<-read.csv("ToyotaCorolla.csv")
cars$FuelType <- as.factor(cars$FuelType)
cars$MetColor <- as.factor(cars$MetColor)
cars$Automatic <- as.factor(cars$Automatic)
cars$Doors <- as.factor(cars$Doors)
```

V4 – Model summary



```
Call:
lm(formula = Price ~ . + I(Age^2) + I(CC^2) + I(Weight^2) + I(HP^2),
    data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-7343.6  -704.6   -12.9   707.9  6850.0 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)     2.581e+04  8.782e+03   2.938  0.00335 ** 
Age            -2.495e+02  1.015e+01 -24.571  < 2e-16 ***
KM             -1.537e-02  1.212e-03 -12.679  < 2e-16 ***
FuelTypeDiesel -1.717e+03  8.460e+02  -2.029  0.04260 *  
FuelTypePetrol  8.675e+02  3.085e+02   2.812  0.00500 ** 
HP              3.431e+01  2.111e+01   1.625  0.10428    
MetColor1       2.871e+01  6.844e+01   0.419  0.67497    
Automatic1      2.833e+02  1.525e+02   1.858  0.06343 .  
CC             -5.006e+01  6.180e+00  -8.101 1.17e-15 ***
Doors3         -1.244e+02  8.516e+02  -0.146  0.88387    
Doors4          1.168e+01  8.569e+02   0.014  0.98913    
Doors5         -9.711e+01  8.536e+02  -0.114  0.90944    
Weight          3.801e+01  1.238e+01   3.071  0.00218 ** 
I(Age^2)        1.219e+00  9.374e-02  13.002  < 2e-16 ***
I(CC^2)         1.616e-02  2.115e-03   7.640 3.97e-14 ***
I(Weight^2)    -9.794e-03  4.955e-03  -1.977  0.04827 *  
I(HP^2)         4.124e-02  7.169e-02   0.575  0.56527    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1199 on 1419 degrees of freedom
Multiple R-squared:  0.892,	Adjusted R-squared:  0.8907 
F-statistic: 732.2 on 16 and 1419 DF,  p-value: < 2.2e-16
```

V5 – Interaction between factors



```
Call:
lm(formula = Price ~ . + FuelType:Weight, data = cars)

Residuals:
    Min      1Q  Median      3Q     Max 
-9129.8  -735.1    -4.3   728.4  6574.1 

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)    
(Intercept)            3.940e+04  1.745e+04   2.257  0.02413 *  
Age                   -1.242e+02  2.613e+00 -47.554  < 2e-16 ***
KM                    -1.503e-02  1.314e-03 -11.438  < 2e-16 ***
FuelTypeDiesel        -4.959e+04  1.750e+04  -2.833  0.00468 ** 
FuelTypePetrol        -3.909e+04  1.741e+04  -2.245  0.02490 *  
HP                     5.219e+01  5.862e+00   8.904  < 2e-16 ***
MetColor1              3.714e+01  7.403e+01   0.502  0.61598    
Automatic1             3.913e+02  1.572e+02   2.489  0.01291 *  
CC                    -3.155e+00  5.736e-01  -5.500  4.5e-08 ***
Doors3                -6.040e+02  9.215e+02  -0.655  0.51232    
Doors4                -3.760e+02  9.272e+02  -0.405  0.68520    
Doors5                -5.756e+02  9.223e+02  -0.624  0.53266    
Weight                -1.949e+01  1.587e+01  -1.228  0.21954    
FuelTypeDiesel:Weight  4.724e+01  1.592e+01   2.968  0.00305 ** 
FuelTypePetrol:Weight  3.665e+01  1.586e+01   2.311  0.02097 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1298 on 1421 degrees of freedom
Multiple R-squared:  0.8731,	Adjusted R-squared:  0.8718 
F-statistic: 698.3 on 14 and 1421 DF,  p-value: < 2.2e-16
```

Advance Methods

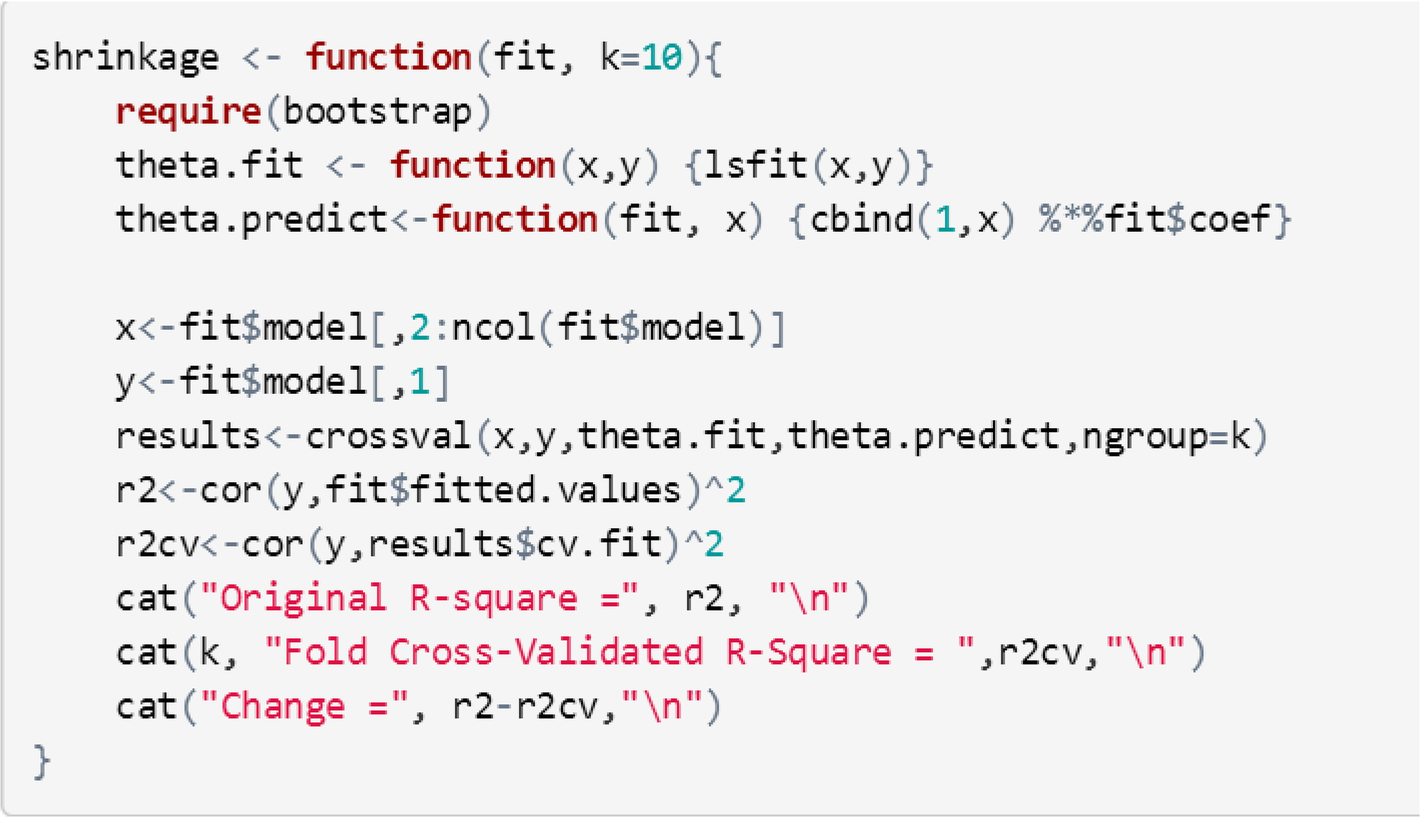
Variable Selection

- library(MASS)
- stepAIC(fit4.0, direction = "backward")

Cross Validation

- In Cross-Validation, a portion of the data is selected as the training sample, and a portion is selected as the hold-out sample. A regression model is developed on the training sample, and then applied to the hold-out sample to validate its performance.
- In *k-fold cross-validation*, the sample is divided into *k* subsamples. Each of the *k* subsamples serves as a hold-out group, and the combined observations from the remaining *k-1* subsamples serve as the training group. The performance for the *k* prediction equations applied to the *k*-hold out samples is recorded and then averaged.

Cross Validation

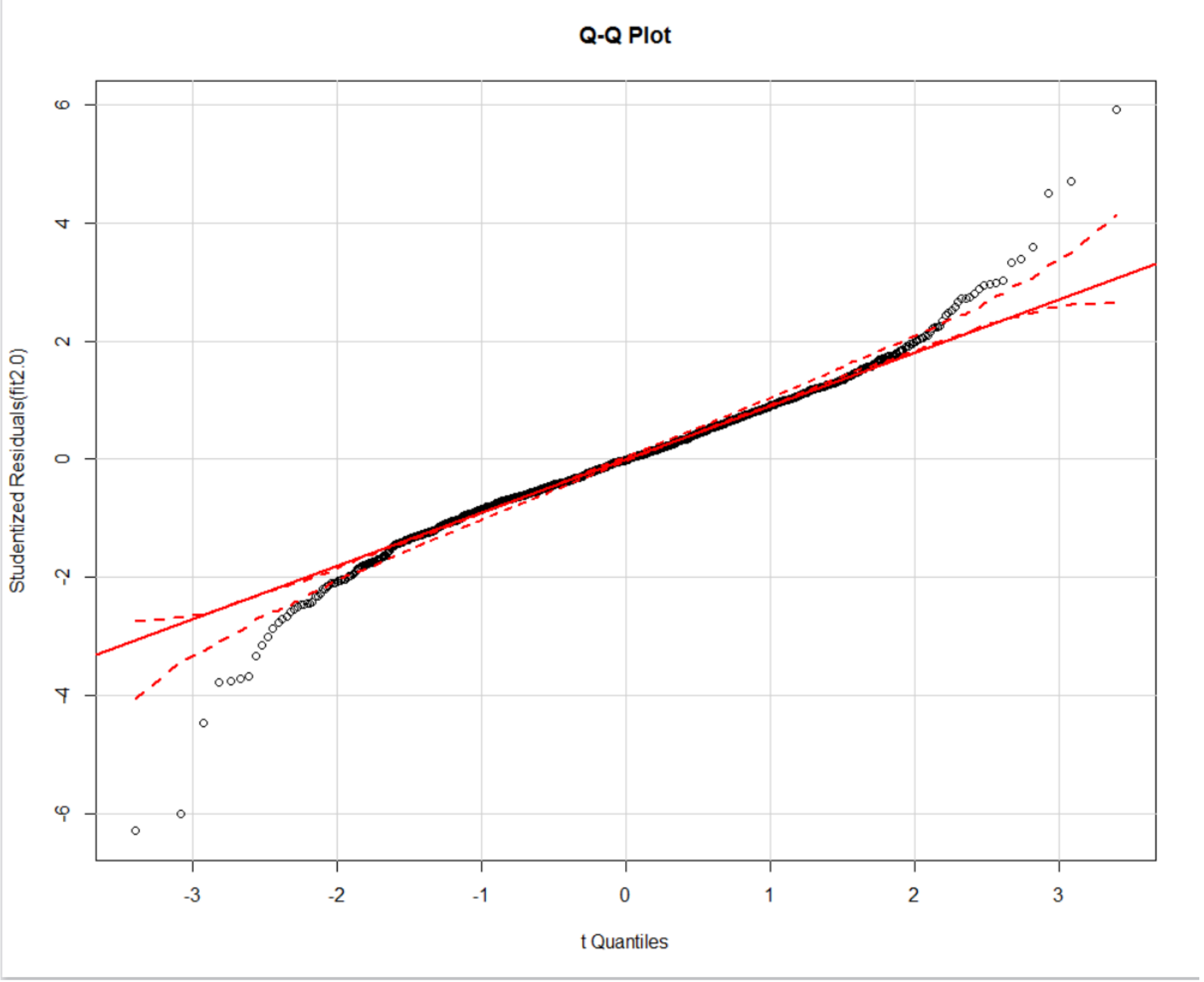


```
shrinkage <- function(fit, k=10){
    require(bootstrap)
    theta.fit <- function(x,y) {lsfit(x,y)}
    theta.predict<-function(fit, x) {cbind(1,x) %*%fit$coef}

    x<-fit$model[,2:ncol(fit$model)]
    y<-fit$model[,1]
    results<-crossval(x,y,theta.fit,theta.predict,ngroup=k)
    r2<-cor(y,fit$fitted.values)^2
    r2cv<-cor(y,results$cv.fit)^2
    cat("Original R-square =", r2, "\n")
    cat(k, "Fold Cross-Validated R-Square = ",r2cv,"\n")
    cat("Change =", r2-r2cv,"\n")
}
```

car Package

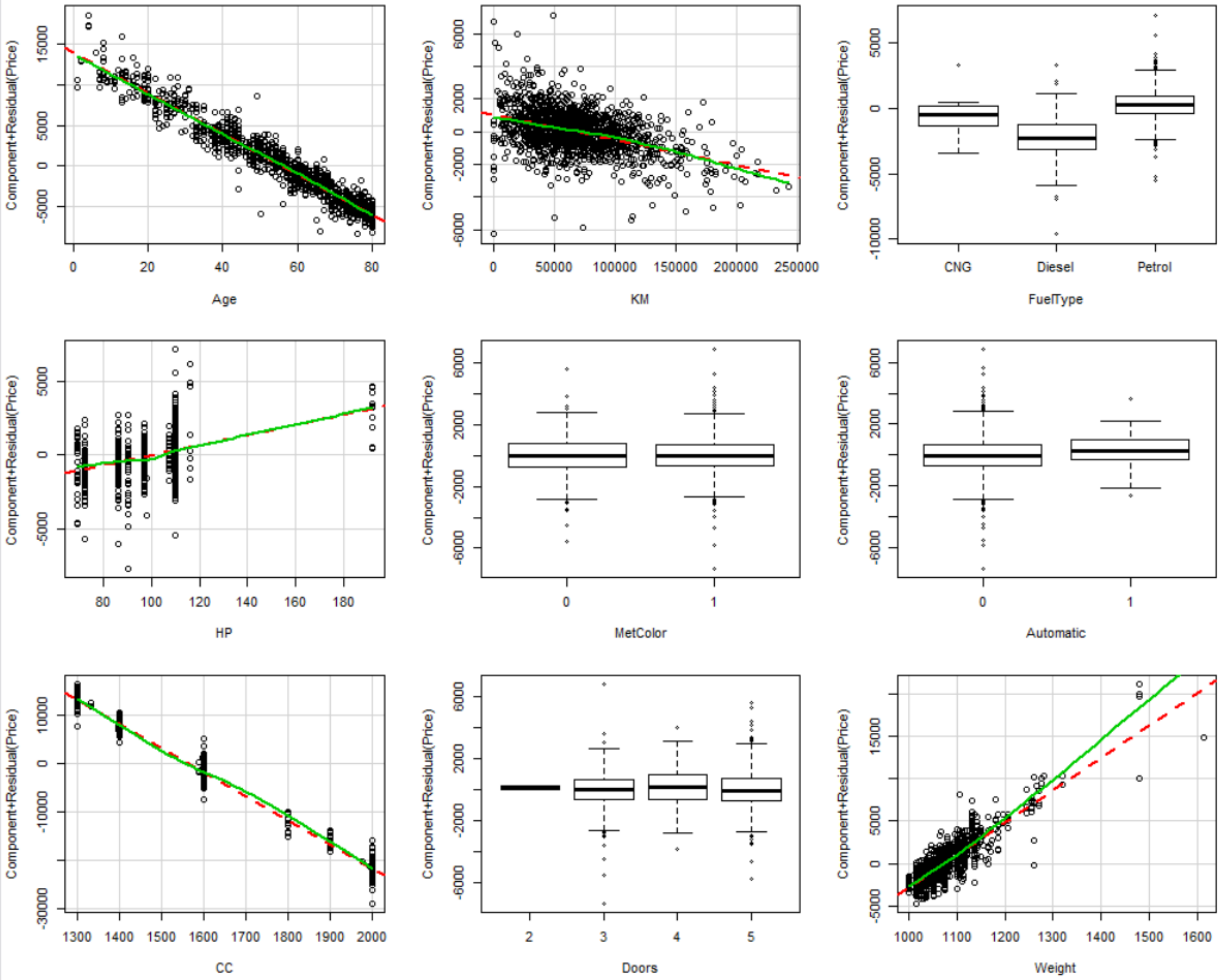
qqPlot – Normality Test

```
library(car)
qqPlot(fit2,label=cars$Price, id.method = "identity",simulate = TRUE, main="Q-Q Plot")
```



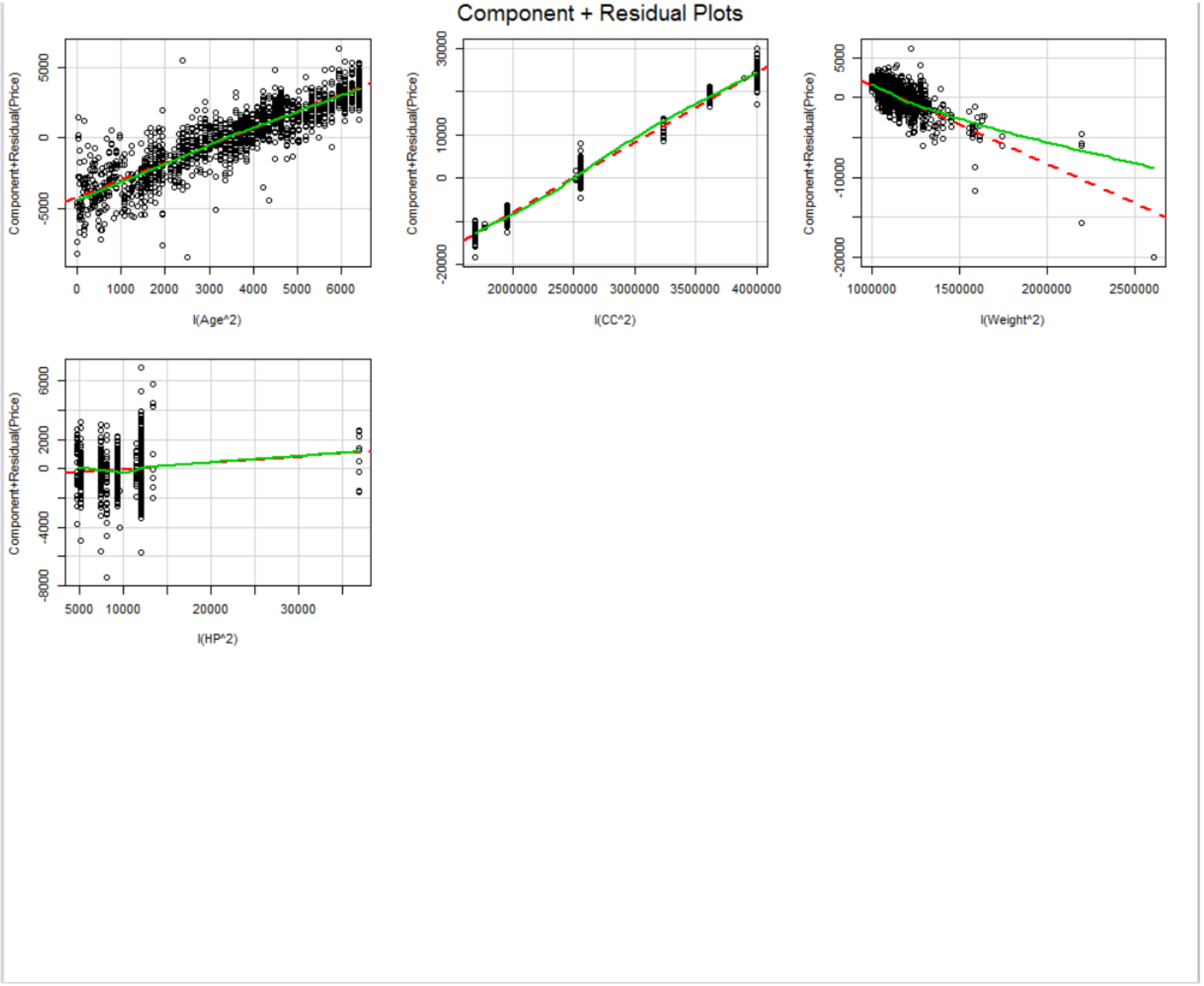
crPlots – Linearity Test

```
crPlots(fit3.0)
```

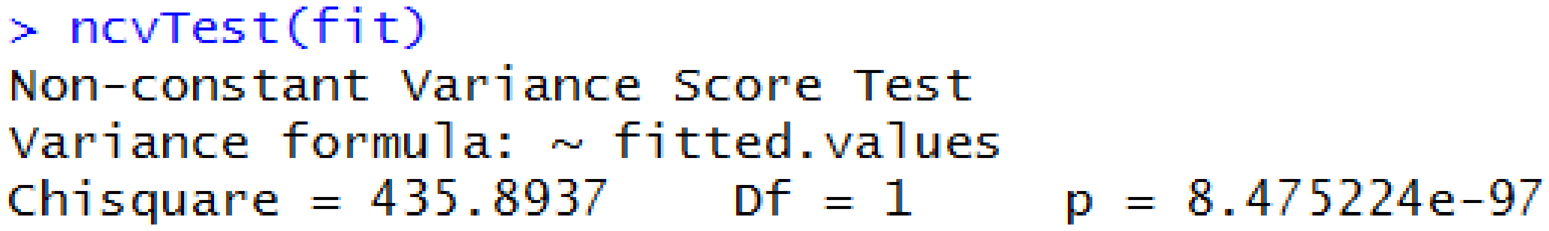


crPlots

```
crPlots(fit3.0)
```



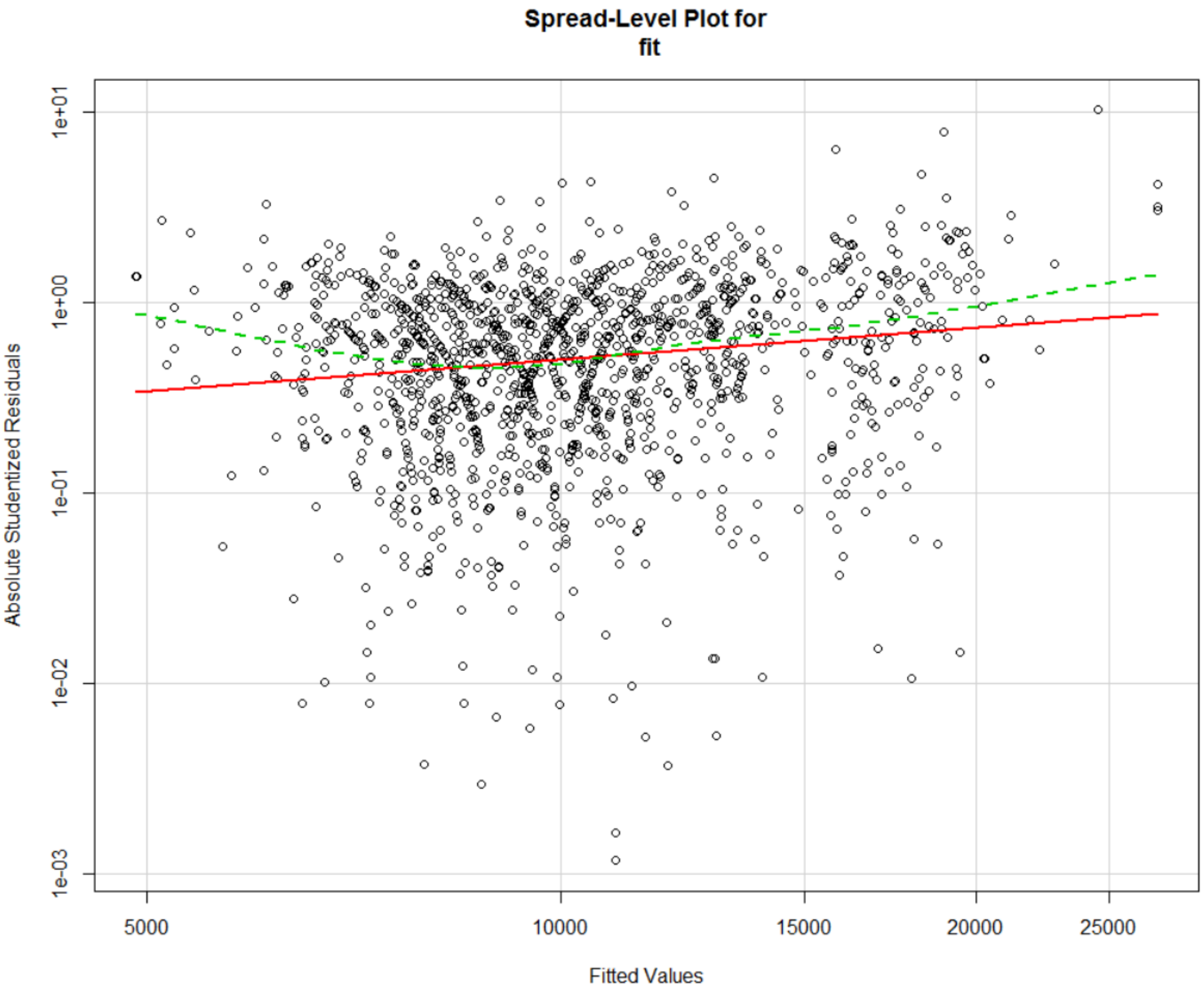
ncvTest – Homoscedasticity test



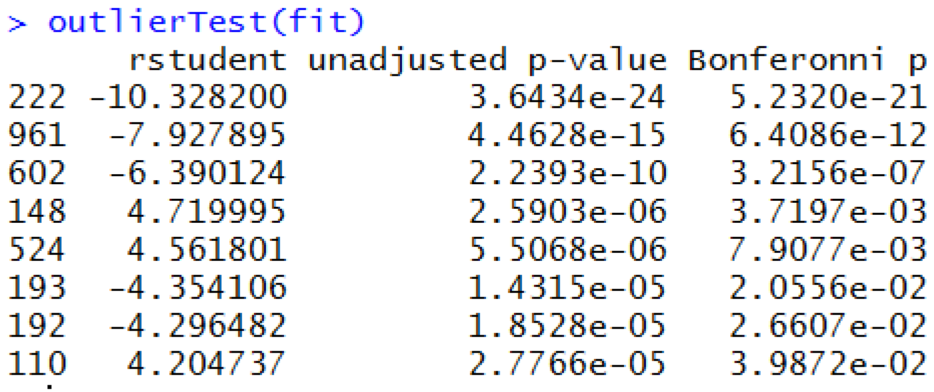
```
> ncvTest(fit)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 435.8937    Df = 1     p = 8.475224e-97
```

spreadLevelPlot – Homoscedasticity Test

```
> spreadLevelPlot(fit)

Suggested power transformation:  0.442026
```



outlierTest

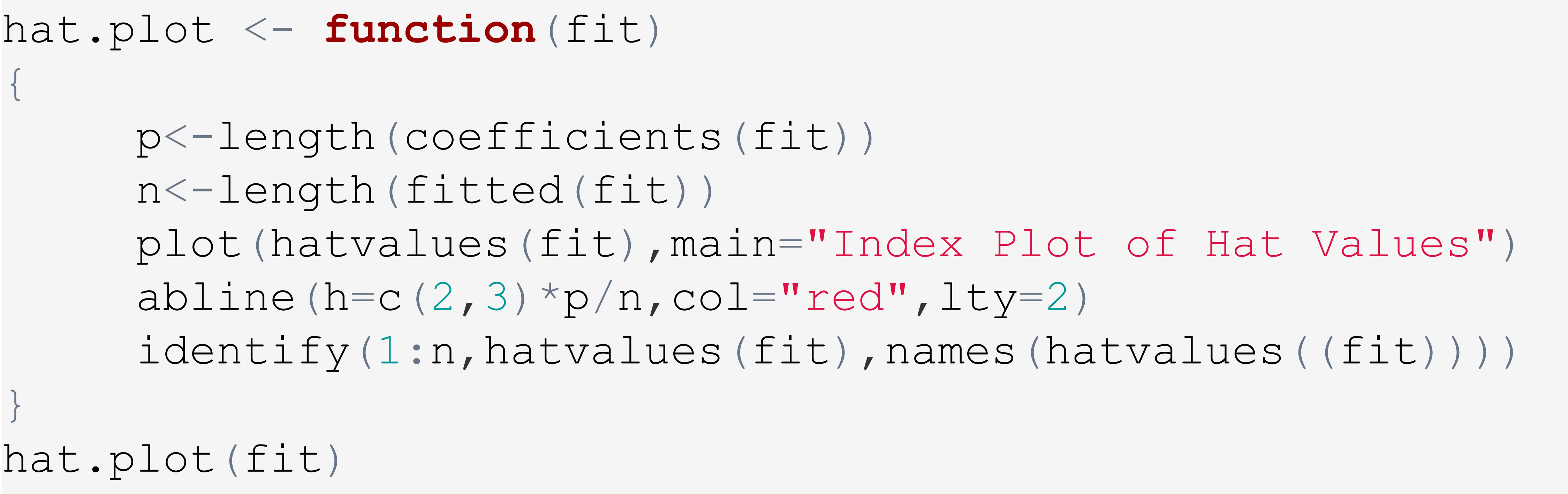


```
> outlierTest(fit)
      rstudent unadjusted p-value Bonferonni p
222 -10.328200         3.6434e-24   5.2320e-21
961  -7.927895         4.4628e-15   6.4086e-12
602  -6.390124         2.2393e-10   3.2156e-07
148   4.719995         2.5903e-06   3.7197e-03
524   4.561801         5.5068e-06   7.9077e-03
193  -4.354106         1.4315e-05   2.0556e-02
192  -4.296482         1.8528e-05   2.6607e-02
110   4.204737         2.7766e-05   3.9872e-02
```

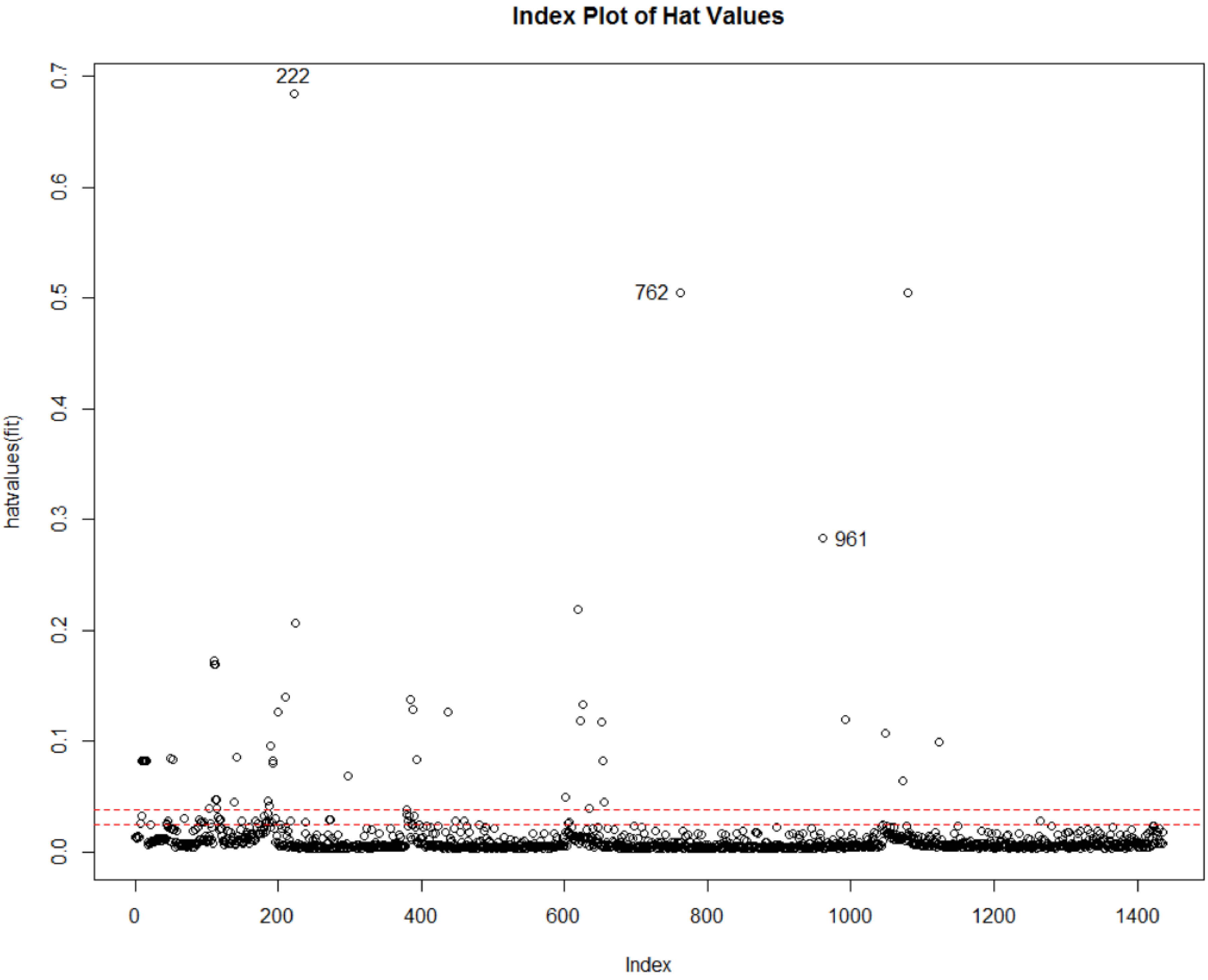
High Leverage Points

- Observations with high leverage are identified through the *hat statistic*.
- For a given dataset, the average hat value is *p/n*, where *p* is the number of parameters estimated in the model (including the intercept) and *n* is the sample size.
- An observation with a hat value greater than 2 or 3 times the average hat value should be examined.

How to identify High Leverage Points?



```
hat.plot <- function(fit)
{
     p<-length(coefficients(fit))
     n<-length(fitted(fit))
     plot(hatvalues(fit),main="Index Plot of Hat Values")
     abline(h=c(2,3)*p/n,col="red",lty=2)
     identify(1:n,hatvalues(fit),names(hatvalues((fit))))
}
hat.plot(fit)
```


Index Plot of Hat Values
hatvalues(fit)
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
222
762
961
0
200
400
600
800
1000
1200
1400
Index

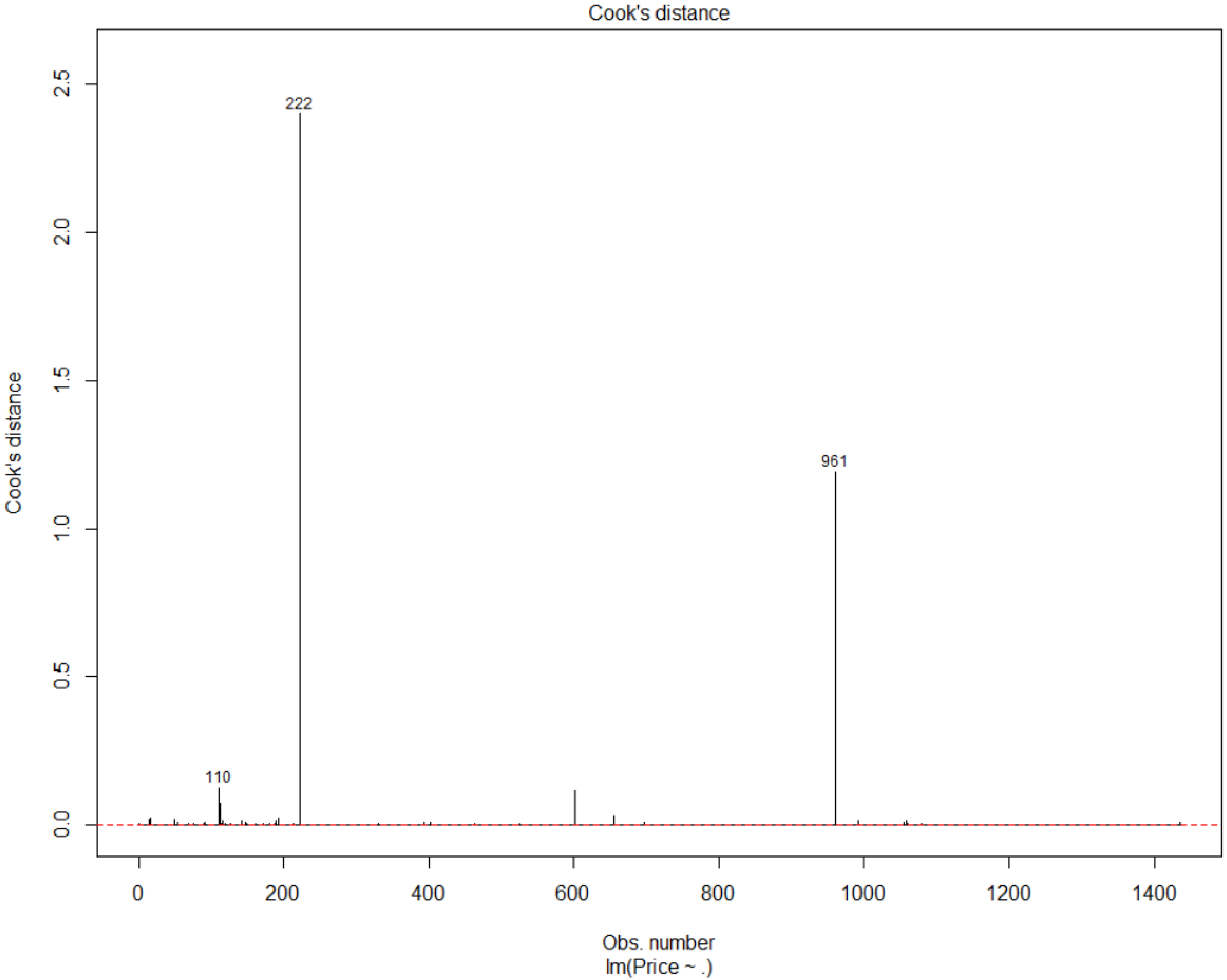
Influential Points

- *Influential* observations have a disproportionate impact on the values of the model parameters, in other words, with or without these observations change your model significantly.

- Cook's D values greater than *4/(n-k-1)*, where *n* is the sample size and *k* is the number of predictor variables, indicate influential observations.

How to identify influential points?

```r
cutoff <- 4/(nrow(cars)-length(fit$coefficients)-1)
plot(fit,which=4,cook.levels = cutoff)
abline(h=cutoff, lty=2,col="red")
```


Cook's distance
Cook's distance
0.0
0.5
1.0
1.5
2.0
2.5
222
961
110
0
200
400
600
800
1000
1200
1400
Obs. number
lm(Price ~ .)