Final Project

Sangalli Chiara

The dataset

The data set chosen for this project contains data from the 1985 Ward's Automotive Yearbook and consist of 199 observations of 10 different variables. It was selected from the web site UCI.

Among those variables one is categorical: **fuel_type**, which has two possible outcomes: **diesel** or **gas**.

The remaining 9 variables are all continuous: the **price**, that will be used as the response variable of this analysis and 8 predictors: **length**, **width** and **height** all measured in inches, **curb_weight** whose unit of measures is lb, **engine_size** in cubic centimeters, **horsepower** in hp, and **city_mpg** and **highway_mpg** which respectively measure the fuel consumption in cities and in highways.

All the quantitative covariates have been centered: from each observation has been subtracted the mean of each independent variable. This results in a new set of covariates that have a mean of zero.

The goal

The goal of this project is to rely on a linear regression model to explain the relationship between the dependent variable **price** and the independent ones; in other words the aim of this analysis is to identify which features of cars have a significant impact on the car's price and the magnitude of their impact.

Graphical rapresentation of the data

Graphs a) and b) in figure 1 show the histogram and the boxplot of the response variable **price**: both of them highlight a right-skewed distribution.

From the histogram it's immediate to see that the majority of the observations is condensed between 5 and 10 thousands of dollars; as the price grows, the number of observations decreases, with a major drop after the price gets higher than 20000$. The blue vertical line represents the mean of the distribution and is placed around 13000$.

The same conclusion can be reached by looking at the boxplot, that shows a median value equal to 10000$: it's a lower value than the mean one, and this confirms the asymmetry of the distribution of the response. In the boxplot all the observation with price higher than 30000 dollars are extreme points.

Plot c) in figure 1 represents the relationship between the categorical **fuel_type** and **price**: what can be said by looking at this graph is that the engine type (diesel or gas) will probably have an effect on the price of the car itself: this can be said because the median values for the 2 groups of this variable are quite different.

```r
par(mfrow=c(1,3))
hist(cars$price, main="a) Histogram of Price", xlab="Values", ylab="Frequency", breaks=10,
xlim=c(0,50000), ylim=c(0, 100), col="lightskyblue", border="lightskyblue4")
abline(v=mean(cars$price), col="lightskyblue4", lwd=2)
boxplot(cars$price, main="b) Boxplot of Price", ylab="Values", col="lightskyblue",
border="lightskyblue4")
```

```
plot(cars$fuel_type, cars$price, col="seagreen3", border="seagreen4", main="c)Boxplot of price
     \nfor fuel type")
```

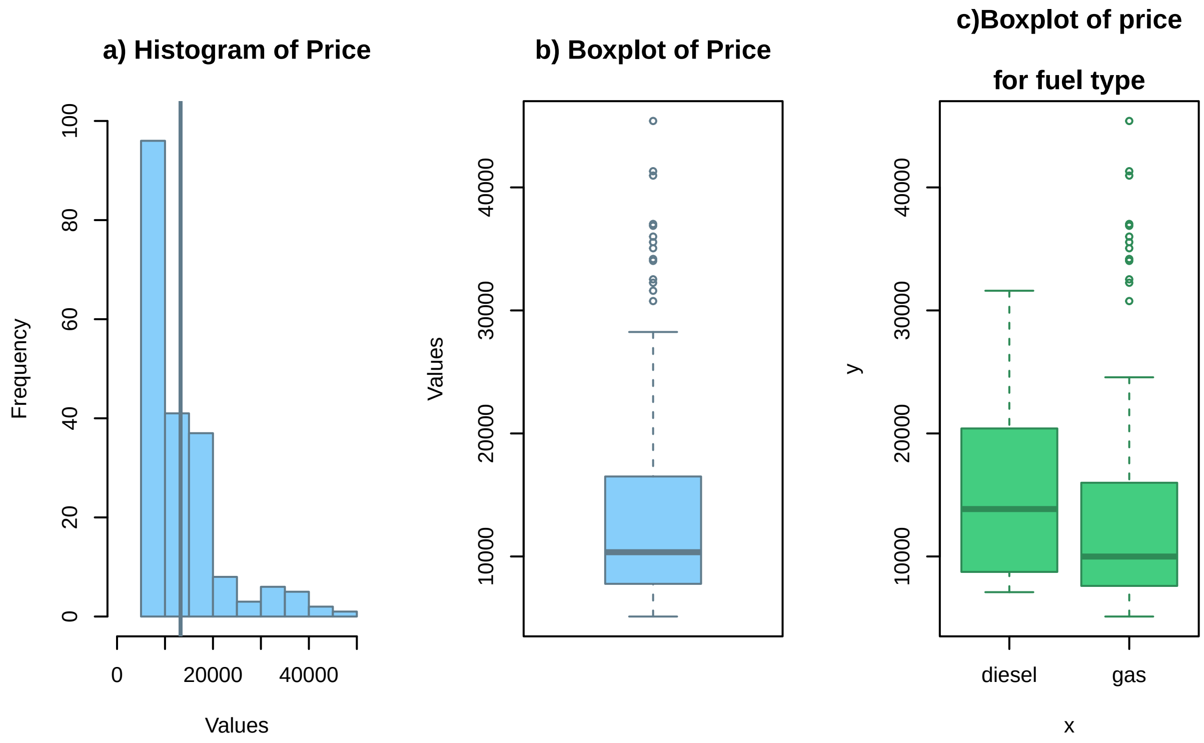

Figure 1: Visualization of price distribution

Figure 2 displays the bivariate relationships between the response **price** and all the quantitative covariates. What can be observed is that there is a linear relationship between some regressors and the price: that's the case for **width**, **curb_weight**, **engine_size** and **horsepower**; in some other cases, like **length**, **city_mpg** and **highway_mpg** the relationship is there, but it's not linear, while between **height** and **price** there is no apparent relationship.

The same figure represents also the bivariate relationships between all the continuous predictors of the model: while some distributions resemble the null plot, meaning that there is not a strong relationship between the variables, others show a clear relationship, linear or of some other type.

The most striking case appears to be the link between **city_mpg** and **highway_mpg** which resembles almost perfectly a straight line: since both these variables offer a measure of the fuel consumed by a car it's logical that the two of them have a strong correlation, that will be addressed later.

```
plot(cars[,-1], col="seagreen3", pch=18)
```

Fit of the linear model

```
M1 <- lm(price ~ .^2, data = cars)
```

The first model fitted, **M1**, contains the 9 predictors and all the possible interactions among them: the total number of estimated coefficients is 46. Due to the computational problems that arise when dealing with a model with so much parameters to estimate and since a lot of the interactions are not significant, only the

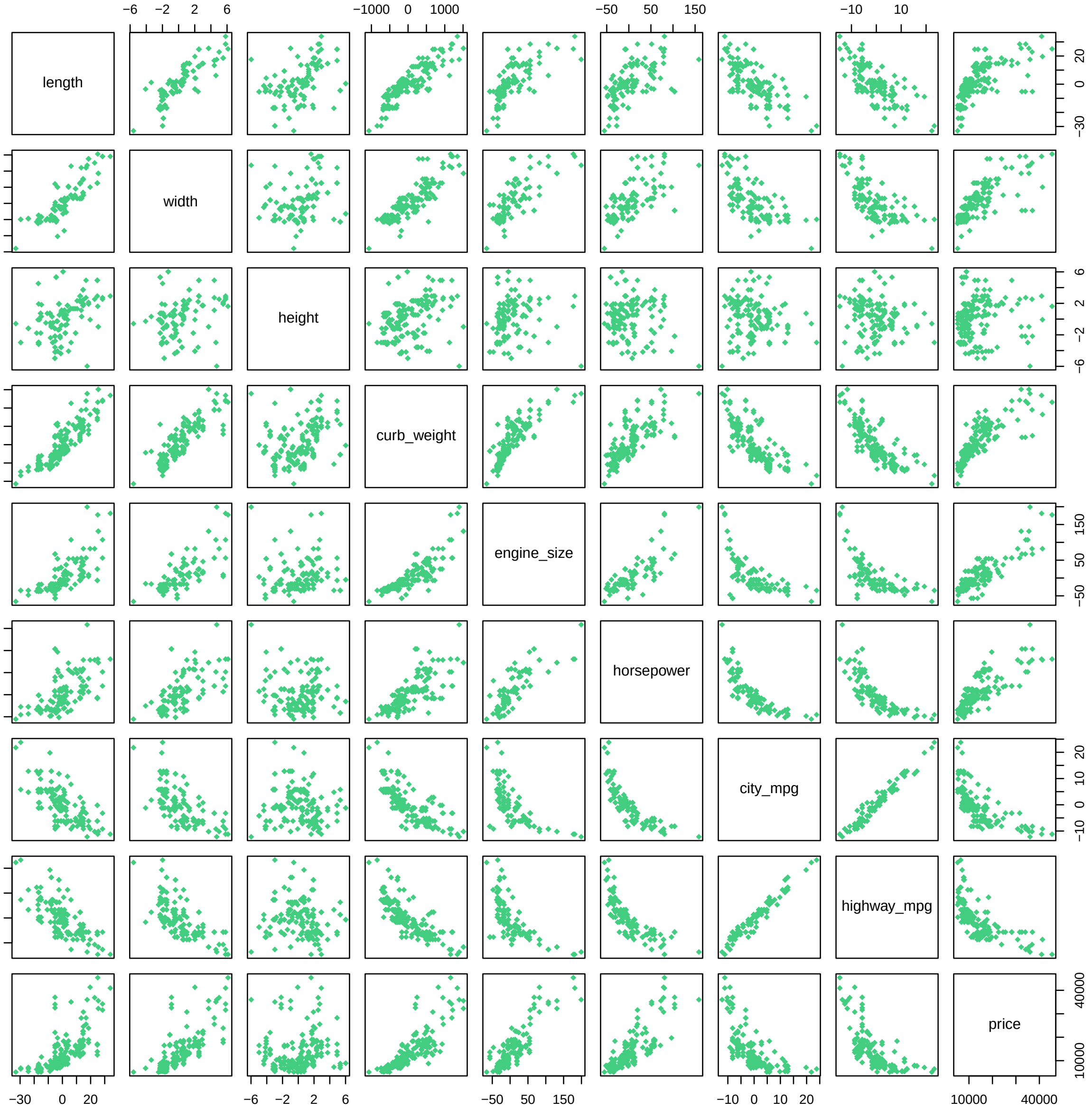

Figure 2: Plot of predictors vs eachothers

ones with a significance level below 0.001, written in table 1, are kept in the fit: this second model is called **M2**,

Table 1: Signficative interactions

engine_size:city_mpg	engine_size:highway_mpg
42	43

M2 contains 11 predictors (the 9 variables present in the dataset and the 2 most relevant interactions) and from this model a subset selection is performed: this process aims to identify the set of predictors that are most relevant for predicting the outcome of the response for each possible number of parameters (in this case, from 1 to 11).

```
M2 <- lm(price ~ .+engine_size*city_mpg+engine_size*highway_mpg, data = cars)
M2_ss <- regsubsets(price ~ .+engine_size*city_mpg+engine_size*highway_mpg, nvmax = 11, data = cars)
```

Choice of the best model

Once the best model for each number of parameters are known, the following step is to determine which is the optimal number of regressor that must be used to fit the model: this decision can be taken by using different criteria: the BIC, the AIC, the Cp Mallow's and the adjusted R”2 are among them and they are represented in figure 3.

```
par(mfrow=c(1,4))
# BIC
plot(summary(M2_ss)$bic, type="b", pch=19, xlab="Number of predictors", ylab="", main="Drop in BIC")
abline(v=which.min(summary(M2_ss)$bic),col = 2, lty=2)
# AIC
p <- 11
n <- nrow(cars)
AIC = matrix(NA, p, 1)
for(j in 1:p) {
  AIC[j] = summary(M2_ss)$bic[j] - (j+2)*log(n) + 2*(j+1)  }
plot(AIC, type="b", pch=19, xlab="Number of predictors", ylab="", main="Drop in AIC")
abline(v=which.min(AIC),col = 2, lty=2)
# Cp
plot(summary(M2_ss)$cp, type="b", pch=19, xlab="Number of predictors", ylab="", main="Mallow's Cp")
abline(v=which.min(summary(M2_ss)$cp), col=2, lty=2)
# R2
plot(summary(M2_ss)$adjr2, type="b", pch=19, xlab="Number of predictors", ylab="", main="Adjusted R^2")
abline(v=which.max(summary(M2_ss)$adjr2), col=2, lty=2)
```

These 4 graphs show different results: according to the BIC criterion the optimal number of parameters is 5, while for both the AIC and the Mallow's Cp the best solution is to consider 8 covariates and the adjusted *Rˆ2* suggests to fit a model with 9 regressors.

However all the 3 plots that suggest a number of parameters higher than 5, display a common trend: in fact from 5 on, all the 3 curves become flat, meaning that the drop in AIC, the Cp and the adjusted R^2 do not change much when using 5 parameters or more. The conclusion is that a model with 5 regressors is the best choice in terms of both goodness of fit and complexity.

Another way to determine the optimal number of parameters to use for the fit of the model is the cross validation error; however this criterion is mostly used to estimate the goodness of predictions made by a

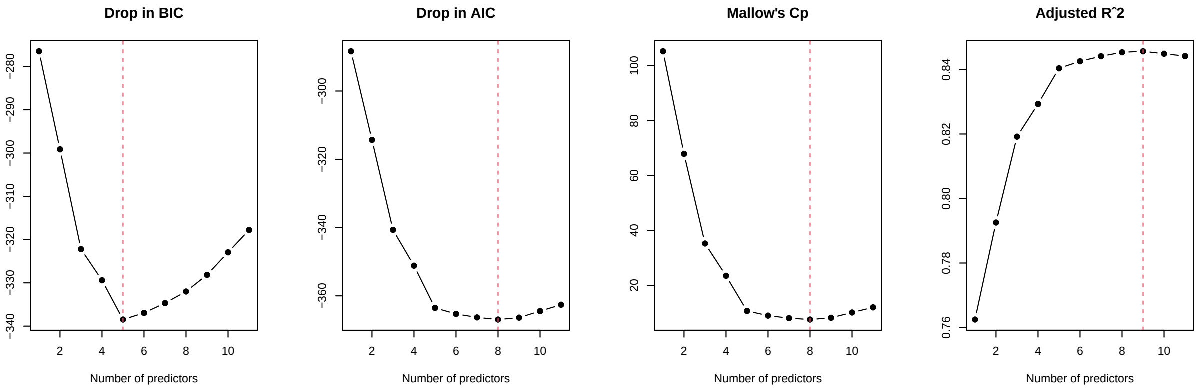

Figure 3: Best model according to different crietria

model: since prediction is not the goal of this analysis, the previous 4 criteria are more suited to determine the optimal number of regressors.

That being said, if the CV errors are computed it turns out that also according to this criterion the optimal choice is to fit the model with 5 parameters.

The 5 parameters that must be used to fit the model collected in table 2:

Table 2: Optimal covariates to fit a model with 5 regressors

(Intercept)	fuel_typegas	width	horsepower	city_mpg	engine_size:city_mpg
17204.47	-5996.318	711.3255	53.18973	-505.1925	-8.216229

Since the interaction between **engine_size** and **city_mpg** is relevant, the hierarchical principle states that the 2 variables that form the interaction must be considered in the model: this means that **engine_size** must be added to the model, that in total will be made up of 6 variables; the best model selected is the following:

```
M3 <- lm(price ~ fuel_type + width + horsepower + city_mpg + engine_size + engine_size*city_mpg,
data = cars)
```

Table 3: Estimated coefficients from M3

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	16929.757335	907.462731	18.656146	0.0000000
fuel_typegas	-5462.750984	1077.028092	-5.072060	0.0000009
width	645.993861	189.207721	3.414204	0.0007804
horsepower	47.396384	14.821276	3.197861	0.0016195
city_mpg	-457.036474	99.211270	-4.606699	0.0000074
engine_size	20.331701	17.296117	1.175507	0.2412477
city_mpg:engine_size	-7.039222	1.337455	-5.263146	0.0000004

In table 3 are represented the estimated coefficients with their standard errors, t-values and p-values. **engine_size** was added for the hierarchical principle, but it's not significant, as one can see from its high p-value.

Collinearity issues

The graph in figure 3 showed a linear relationship between **city_mpg** and **highway_mpg**: to solve this problem the latter variable can be dropped from the model.

```
cars <- cars[, -9]
```

To study the correlation between the variables selected to fit the model it's possible to represent for each of them their **VIF**. The choice of representing the **VIF** instead of the correlation matrix was made because the **VIF** takes into account the correlation between one regressor and all the other ones in the dataset, while a correlation matrix focuses only on the bivariate correlations.

```
VIF = vif(M3)
```

```
## there are higher-order terms (interactions) in this model
## consider setting type = 'predictor'; see ?vif
```

```
plot(VIF, pch=16, ylim=c(1,11), ylab="Vif values", main="Variance Inflation plot", xlim=c(0,8))
abline(h = 10, col = 'orangered1', lty = 2, lwd = 2)
text(x = VIF, labels = names(VIF), cex = 0.8, pos = 3)
```

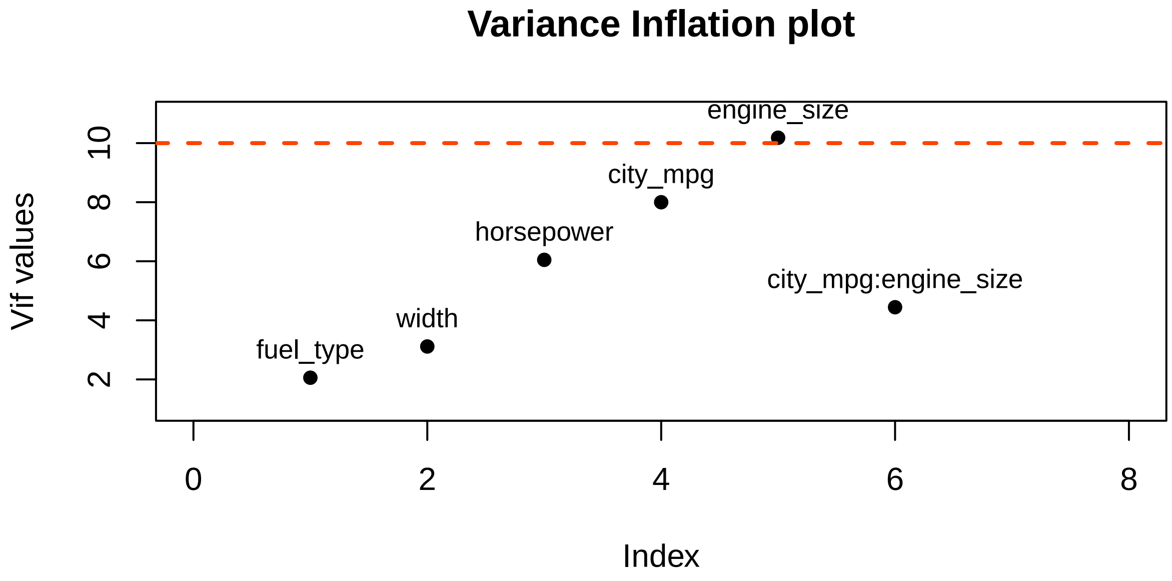

Figure 4: VIF of the selected variables

The points in figure 4 represent the **VIF** for each variable used in the fitting: the only problematic variable is the **engine_size** which has a **VIF** equal to 10.181676, just a little bit higher than the threshold; this predictor was included in the model to respect the hierarchical principle.

In this case the interaction is between **city_mpg** and **engine_size**: while both the interaction and the first variable are significant (their p-value is below 0.05, as showed in table 3), the same can't be said for the second one: for this reason **engine_size** is kept in the model despite its high **VIF**, because removing it would mean remove a significative interaction.

Diagnostics analysis and Model improving

Linearity

The first step of the diagnostic analysis is to verify how the residuals plotted versus the coefficients and the fitted values are distributed: the results are showed in figure 5.

```
residualPlots(M3, test=F)
```

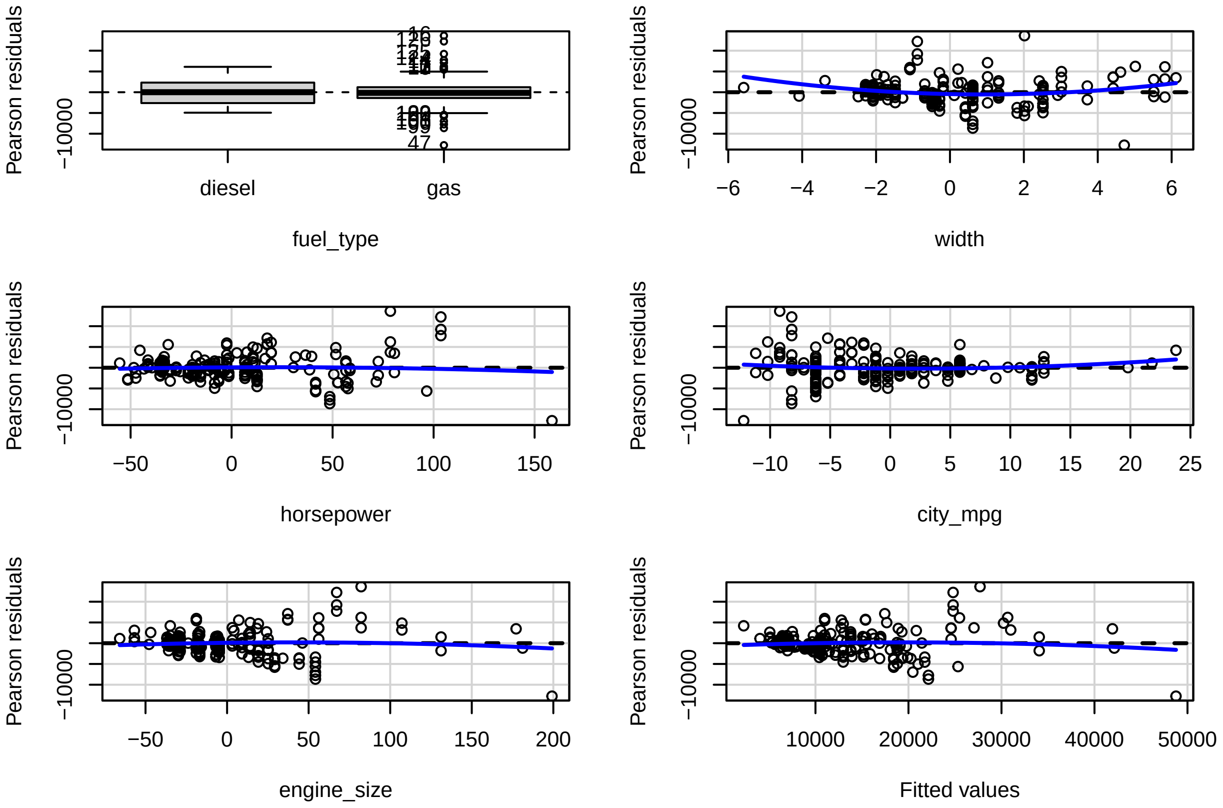

Figure 5: Residuals vs regressors and fitted values

Figure 5 shows that there are not particular problems concerning the linearity assumption: the only variable where the Pearsons residuals follows a non perfectly linear trend is **width**. A possible solution would be to transform these variable: however both the logarithmic and the square root transformation were note feasible due to the fact that the variable is centered (they produces NaN values); since the non linearity is not very marked, the variable can be left unchanged.

Omoschedasticity

The variance of residuals, which should be constant, is clearly heterogeneous: the trend that emerges from plot a) in figure 6 is a right-opening megaphone.

To solve this problem, a logarithmic transformation is applied to the response **price**: plot c) in figure 6 shows the residuals versus the fitted values of the model with **log(price)** as the response, and now the assumption of omoschedasticity is valid.

Normality

To check whether the residuals are normally distributed, one can rely on the QQ-plot (plot b in figure 6) and on the Shapiro-Wilk test: both this tools shows that this property is not satisfied, since form the graph it's clear that the residuals distribution has long tails and the p-value obtained from the test is close to 0; this problem can be addressed by implementing a non parametric test.

After the logarithmic transformation was applied, from plot d) it's clear that the problem is partially solved: now one of the two tails is gone and the remaining one is shorter than before; however the Shapiro test show a p-value that, despite being higher than the previous one, is still below the threshold.

```r
# model before the transformation:
res3 <- residuals(M3)
par(pty="s",mfrow=c(2,2))
plot(fitted.values(M3), res3, pch=16, col="chocolate1", xlab="Fitted values", ylab="Residuals",
main="a) Residuals vs fitted values")
abline(h=0, col="chocolate3", lwd=2)
qqnorm(res3, pch=16,
main="b) QQplot of the residuals", col="darkgoldenrod1")
qqline(res3, lwd=2, col="darkgoldenrod3")
# transformation of the response variable:
cars$log_price <- log(cars$price)
M4 <- lm(log_price ~ fuel_type + width + horsepower + city_mpg + engine_size + engine_size*city_mpg,
data = cars)
res4 <- residuals(M4)
# model after the transformation:
plot(fitted.values(M4), res4, pch=16, col="chocolate1", xlab="Fitted values", ylab="Residuals",
     main="c) Residuals vs fitted values\n with log(price)")
abline(h=0, col="chocolate3", lwd=2)
qqnorm(res4, pch=16, main="d) QQplot of the residuals\n with log(price)", col="darkgoldenrod1")
qqline(res4, lwd=2, col="darkgoldenrod3")
```

After applying the log transformation to the response, the dataset has changed ad so did the relationship between the response (that now is **log_price**) and the regressors: for this reason the subset selection must be re-done.

After repeating the same process done before, the conclusion reached is that now the best model is the one with the 5 regressors contained in table 4:

Table 4: Optimal coefficients selected with the subset selection done on the model with log(price) as response

(Intercept)	fuel_typegas	width	curb_weight	horsepower	city_mpg
9.565227	-0.237118	0.0373044	0.0002924	0.0053883	-0.0124892

The number of parameters is the same as before, but the variables considered are different: the interaction is no more included, so there is no reason to keep **engine_size** either.

```r
M5 <- lm(log_price ~ fuel_type + width + curb_weight + horsepower + city_mpg, data = cars)
```

As figure 7 shows, **M5**, the model fitted with **log(price)** as response, satisfies both the omoschedasticity and the normality assumptions; also the linearity property, which is not represented here, is still valid. In figure 7 also the VIF plot was added, to show that in this model multicollinearity is no longer a problem, since all the variables have a **VIF** lower than 10.

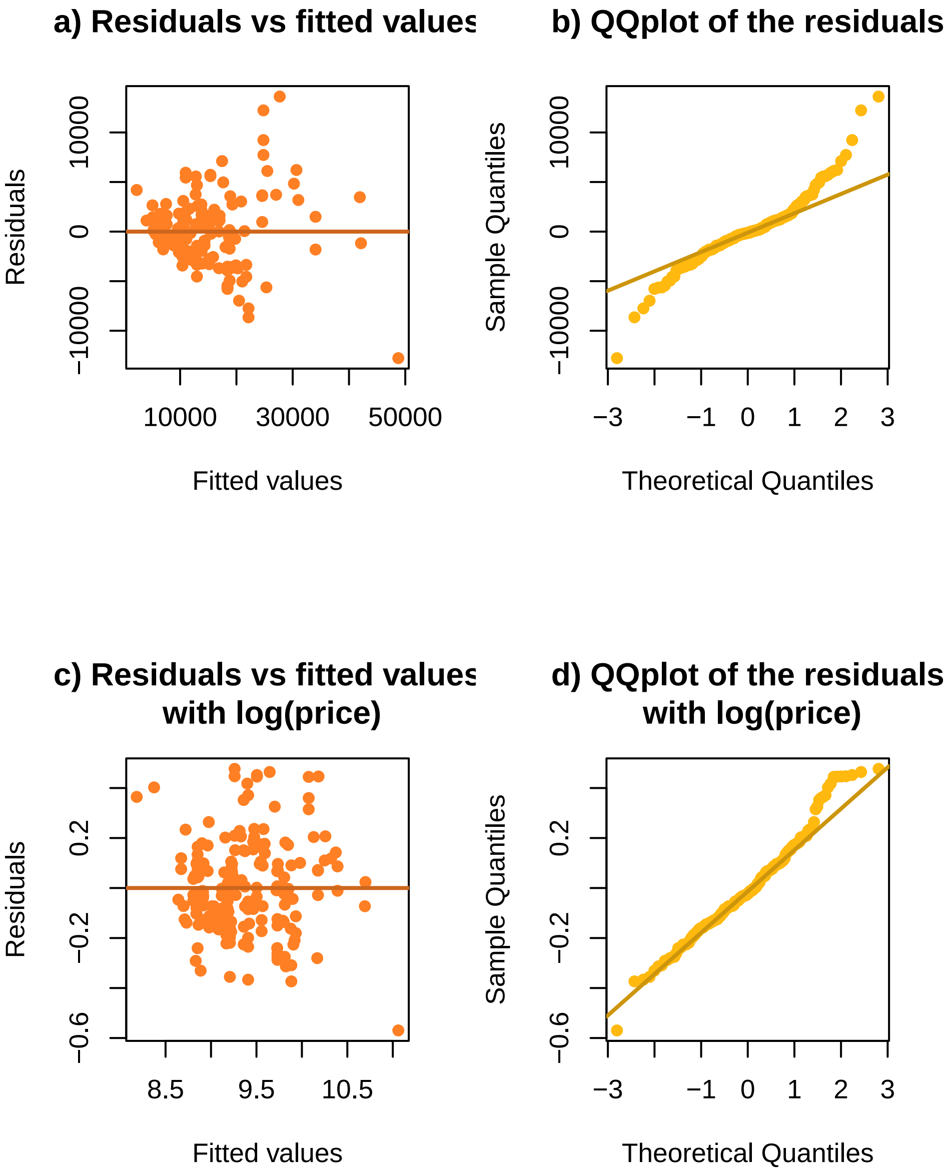

Figure 6: Residuals rapresentations

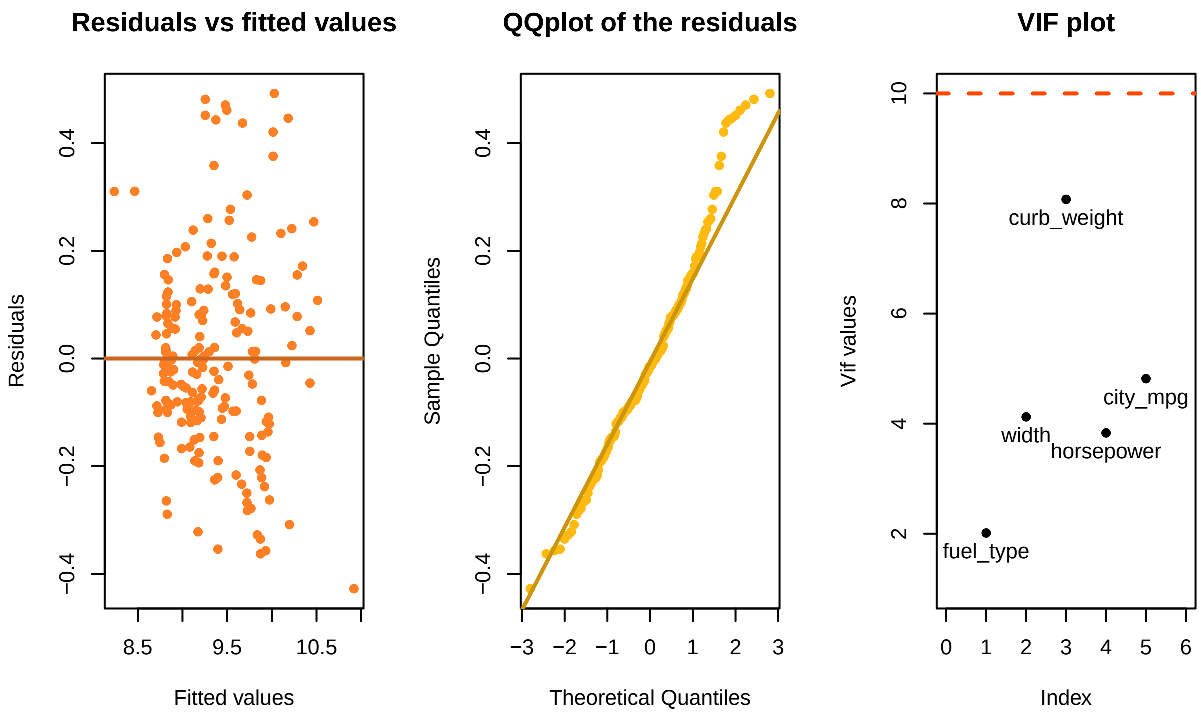

Figure 7: Diagnostics analysis for the model with log(price)

Outliers, high leverage points and influential points

To plot the outliers, the standardized residual **r__standard** must be taken into account: a point is regarded as outliers if its studentized residual is higher than 3 or lower than -3.

A a point is considered to be a leverage point if its leverage **hat** it's higher than 2(p+1)/2: for this dataset the threshold is equal to $2 * (5 + 1)/199 = 0.06$.

Finally, a point is influential if its Cook's Distance **cook** it's higher than 0.5

```
r_standard <- rstandard(M5)
hat <- hatvalues(M5)
cook <- cooks.distance(M5)
```

```
par(mfrow=c(1,3))

plot(r_standard, ylim=c(-4,4), ylab="Studentized residuals", main="Outliers", col= "turquoise2", pch=16
abline(h=3, lty=2, col="red3", lwd=2)
abline(h=-3, lty=2, col="red3", lwd=2)

plot(hat, ylab="Leverages", main="High leverage points", col= "turquoise2", pch=16)
abline(h=12/nrow(cars), lty=2, col="red3", lwd=2)
high_lev <- hat > 0.15
x_hl <- which(high_lev)
y_hl <- (hat)[high_lev]
text(x_hl, y_hl, labels = x_hl, cex = 1, pos = 2)

cook <- cooks.distance(M5)
```

```
plot(cook, ylab="Cook's Distance", main="Influential points", col= "turquoise2", pch=16, ylim=c(0,0.55)
abline(h=0.5, lty=2, col="red3", lwd=2)
ip <- cook > 0.10
x_ip <- which(ip)
y_ip <- (cook)[ip]
text(x_ip, y_ip, labels = x_ip, cex = 1, pos = 4)
```

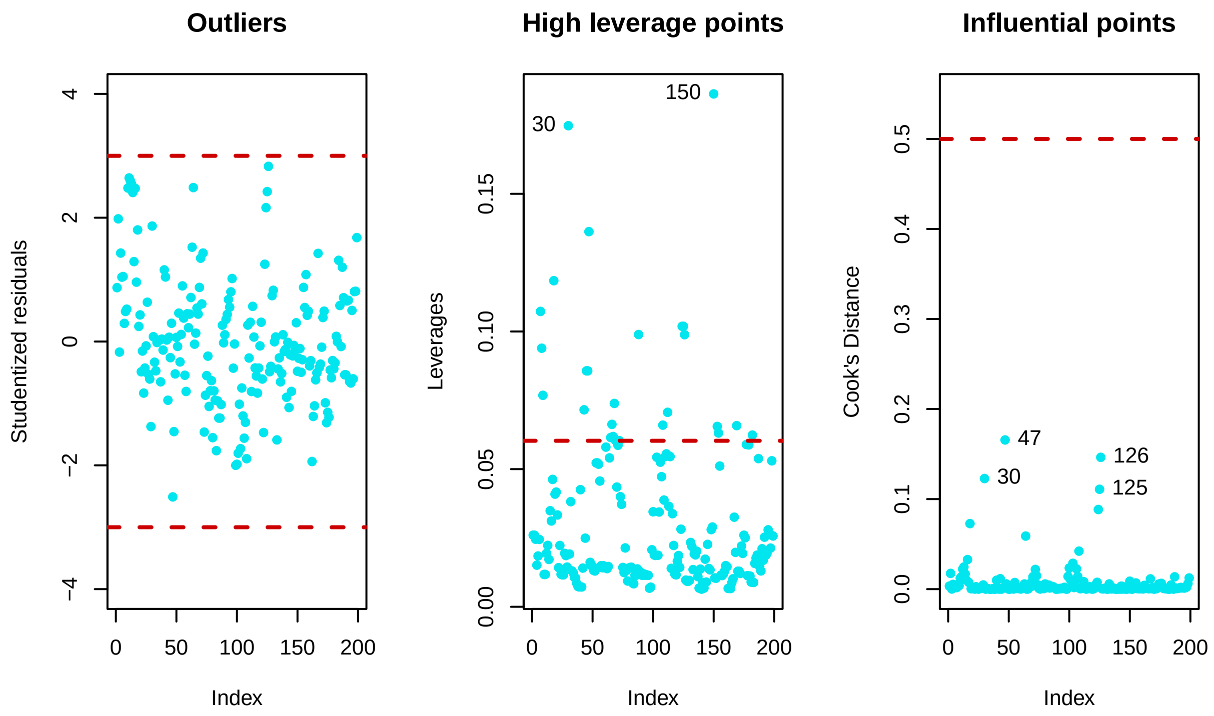

Figure 8: Outliers, high leverage points and influential points

From the graphs in figure 8 emerge that, while not a single outlier is present, several points have a leverage higher than 0.6; however none of them is an influential point, because the Cook's distance of all the points is below the threshold 0.5.

The points with a Cook's Distance higher than 0.1 are the 47th, the 126th, the 30th and the 125th: all those observations not only are below the threshold, but also do not have any significant impact on the model, since the fit without each one of them is the same as the fit with them. For this reason all the points can be kept in the model. The 30th observation is also one of the two points with the highest leverage; the other one is the 150th and also for this data point the model does not change when it's not included in the fit, so there is no reason to remove it.

At this point **M5** is a model with the transformet response that contains the best subset of regressors and respects all the assumptions: its output is the following:

```
summary(M5)
```

```
##
## Call:
## lm(formula = log_price ~ fuel_type + width + curb_weight + horsepower +
##     city_mpg, data = cars)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -0.42744 -0.10949 -0.01439  0.09801  0.49223 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   9.565e+00  5.663e-02 168.913  < 2e-16 ***
## fuel_typegas -2.371e-01  6.128e-02  -3.870 0.000149 ***
## width         3.730e-02  1.252e-02   2.979 0.003268 ** 
## curb_weight   2.924e-04  7.118e-05   4.108 5.88e-05 ***
## horsepower    5.388e-03  6.787e-04   7.939 1.63e-13 ***
## city_mpg     -1.249e-02  4.430e-03  -2.820 0.005311 ** 
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 0.1832 on 193 degrees of freedom
## Multiple R-squared:  0.8714, Adjusted R-squared:  0.8681 
## F-statistic: 261.5 on 5 and 193 DF,  p-value: < 2.2e-16
```

Interpreation of the coefficients

The coefficients estimated by **M5** can be found in table 6:

Table 5: Coefficients table from M5

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	9.5652269	0.0566280	168.913406	0.0000000
fuel_typegas	-0.2371180	0.0612768	-3.869623	0.0001490
width	0.0373044	0.0125243	2.978568	0.0032677
curb_weight	0.0002924	0.0000712	4.108423	0.0000588
horsepower	0.0053883	0.0006787	7.939216	0.0000000
city_mpg	-0.0124892	0.0044295	-2.819547	0.0053110

$\hat{\beta}_0 = 9.5652269$: the intercept represents the estimated value of **log_price**, the log-transformed response variable, when all the quantitative predictor are at their mean values (because they have all been centered) and when the engine type of the car is diesel.

To interpret the intercept in terms of **price**, one must exponentiate the value estimated: $e^{\hat{\beta}_0} = 14260.19$. The expected value of a car's price with a diesel engine when all the other regressors are equal to their mean value is 14260.19 dollars.

$\hat{\beta}_1 = -0.2371180$: this coefficient represent the difference in the **log(price)** between a car with a diesel engine and a car with a a gas one, when all the other predictors are equal to their mean value. When the car's engine is a gas one, the expected value of the car's price is $e^{\hat{\beta}_0} * e^{\hat{\beta}_1} = 11249.84$: so the expected value of a gas car when all the other predictors are held constant is 11249.84 dollars, almost 3000 less than a diesel one; the fact that the price decreases when **fuel_type = gas** is expressed by the negative sign of the coefficient.

All the other coefficients relative to quantitative variables can be interpreted in the same way: they represent the expected change in **log_price** for a one-unit increase in the predictor, holding all other predictors constant. This is because a one-unit increase in the predictor is associated with a proportional change in the response variable rather than an absolute change.

To interpret the coefficient with respect to the **price** one must take the exponend of the value from the output and subtract 1: for example, $\hat{\beta}_4 = 0.0053883$ is the coefficient associated to **horsepower**:

```r
exp(0.0053883)-1
```

```
## [1] 0.005402843
```

This means that a one-unit increase in the **horsepower** is associated with an expected 0.54% increase in the **price**, holding all the other predictors constant.

All the coefficients, whit the exception of the one related to **city_mpg** have positive sign: this means that, as the regressors grows of one unit, the **log(price)** will increase.

Table 6 contains the 95% confidence intervals of the estimated β:

```r
kable(t(confint(M5)))
```

	(Intercept)	fuel_typegas	width	curb_weight	horsepower	city_mpg
2.5 %	9.453538	-0.3579761	0.0126024	0.0001520	0.0040497	-0.0212257
97.5 %	9.676916	-0.1162599	0.0620065	0.0004328	0.0067269	-0.0037527

A graphical representation of the coefficients and their confidence intervals obtained with the function **predictorEffect()** is represented in figure 9:

```r
e1.ols <- predictorEffect("fuel_type", M5, main =" ")
e2.ols <- predictorEffect("width", M5, main =" ")
e3.ols <- predictorEffect("curb_weight", M5, main =" ")
e4.ols <- predictorEffect("horsepower", M5, main =" ")
e5.ols <- predictorEffect("city_mpg", M5, main =" ")
grid.arrange(plot(e1.ols, main =" "), plot(e2.ols, main=""), plot(e3.ols, main =" "),
             plot(e4.ols, main =" "), plot(e5.ols, main =" "), ncol=3)
```

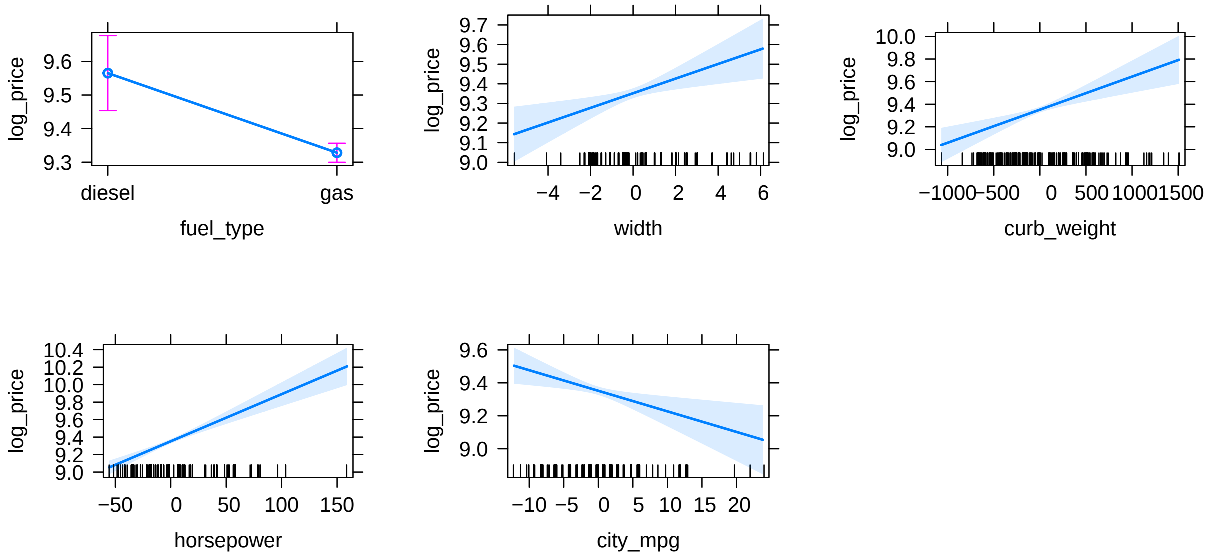

Figure 9: Effect plots for the coefficients

For **fuel_type**, the categorical predictor, the confidence interval is represented with the vertical lines in correspondence to the two points (the one corresponding to diesel is the value of $\hat{\beta}_0$, while the other in correspondence of gas is equal to $\hat{\beta}_0 + \hat{\beta}_1$): it's immediate to notice that the confidence interval related to the intercept is bigger.

For the continuous regressors the line has a slope equal to the correspondent $\hat{\beta}$ and the confidence interval for each point is represented by the light-blue area surrounding the line.

Tests of the coefficients

Table 5 displays for each regressor the p-value relative to the t-test computed to determine whether the estimated coefficient is equal to 0 (H_0) or different from 0 (H_1): since all of them are below the threshold of 0.05, for all the predictors there is evidence against the null hypothesis, so the conclusion is that all the estimated coefficient are different than 0.

Testing a group of predictors

M6 is a model fitted without the 2 variables with the smallest estimated coefficients, **horsepower** and **curb_weights**: due to their small $\hat{\beta}$, a unit change of these 2 regressors won't have a large impact on the response, so the idea is to fit a model that contains only the variables with large $\hat{\beta}$ and see if it's different to the previously fitted **M5**

```
M6 <- lm(log_price ~ fuel_type + city_mpg + width, data = cars)
anova(M5, M6)
```

```
## Analysis of Variance Table
##
## Model 1: log_price ~ fuel_type + width + curb_weight + horsepower + city_mpg
## Model 2: log_price ~ fuel_type + city_mpg + width
##   Res.Df     RSS Df Sum of Sq      F    Pr(>F)
## 1    193  6.4794
## 2    195 10.6542 -2   -4.1748 62.178 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value of the ANOVA test it's almost 0: this suggest that there is evidence against the null hypothesis (according to which the full and nested model are equivalent); in this case the alternative H_1 must be accepted and the conclusion reached is that **M5** provides a better fit to the data than the nested **M6**, so all the 5 variables, also the one with very little $\hat{\beta}$ must be kept in the model.

Goodness of fit

An indication of the goodness of fit of the model is offered by the R^2, the ratio between the variablity explained by the regression model and the total variability of the response.

```
summary(M5)$r.squared
```

```
## [1] 0.8713966
```

The 87.14% of the variability of $log(price)$ is explained by **M5**

Prdiction

```
new_obs <- data.frame("fuel_type"="diesel", "width"=2.7286945, "curb_weight"=503.969849,
"horsepower"=31.5910671, "city_mpg"=7.2950201)
```

new__obs contains information regarding a car which was not included in the dataset **cars** used to fit the model: one can predict the price of this car using the estimated coefficients of **M5** and the values of the predictors from **new__obs**. The fit value in table 6 is the **log(price)** estimated by the model, and the other two values represent the lower and the upper extreme of the 95% confidence interval.

```
y_fitted <- predict(M5, newdata = new_obs, level = 0.95, interval="prediction")
```

Table 7: Preicted value of log(price) and its confidence interval

fit	lwr	upr
9.893508	9.515075	10.27194

The fitted price of the car according to **M5** is equal to e^9.893508 = 19801.40$ dollars.

Simulation of data points

```
n <- 199
set.seed(7)
beta <- coefficients(M5)
X <- model.matrix(M5)
y <- X %*% beta + rnorm(n, 0, sigma(M5))
```

The vector **y** contains 199 fitted values of **log(price)**, predicted using the model **M5**. To check how far are the predictions made by the model from the actual data points contained in the dataset, the mean square error can be computed as follows:

```
MSE <- mean((cars$log_price - y)^2)
MSE
```

```
## [1] 0.05472827
```

MSE offers a measure of the average squared difference between the actual and simulated data points: in this case the model provides a relatively good fit to the data, because, on average, the difference between actual and simulated points is relatively small.