STK2100: Solutions Week 2

Lars H. B. Olsen

21.01.2021

Data Analysis and Data Mining

Exercise 2.3

The linear model (2.14) obtains a $R^2 = 0.6$ for the `auto` data set, see bottom of page 20. We are asked to comment on the change in R^2 for model (2.17) when we apply it on a transformed version of the data.

```
# Read the data from the webpage
auto = read.table("http://azzalini.stat.unipd.it/Book-DM/auto.dat", header = TRUE)
attach(auto)

# Convert fuel into a categorical explanatory variable
fuel1 = factor(fuel, levels = c("gas", "diesel"))

# Define the transformed response variable, consumption.
y = 1/city.distance

# Fit the linear model given in (2.17) and we get R-squared = 0.6381 \approx 0.64
fit_17 = lm(y ~ engine.size + fuel1)
summary(fit_17)$r.squared
```

```
## [1] 0.6380963
```

We get a higher R^2 for model (2.17) than for (2.14), hence, we might be lead to think that (2.17) describes the data better. However, to compare model (2.17) with model (2.14), we need to compute R^2 for (2.17) on the original scale. See page 23. In model (2.17) we are performing a non-linear transformation of our targets, so we expect R^2 to be different.

```
# Use (2.15) to compute R^2 on the original scale
r2_17 =  1 - sum((city.distance - 1/fitted(fit_17))^2) /
  ((length(city.distance)-1)*var(city.distance))
r2_17
```

```
## [1] 0.5614762
```

We now see that R^2 for model (2.17) is lower than that of model (2.14). I.e., (2.14) is still a better fit as it obtained $R^2 = 0.597 \approx 0.6$, which is higher than 0.56.

Exercise 2.7

We are asked to fit an appropriate linear model to predict `highway.distance` for the `auto` data, in two ways:

1. (a): Using variables from chapter: engine.size, fuel, cylinders, curb.weight, city.distance
2. (b): Using variables listed in Appendix B.2

Note: This is en exercise where you should experiment a bit, so there is not one correct solution. This is just one way to do it.

```
# Download the table and attach it
auto = read.table("http://azzalini.stat.unipd.it/Book-DM/auto.dat", header = TRUE)
attach(auto)
```

```
## The following objects are masked from auto (pos = 3):
##
##     aspiration, bodystyle, brand, city.distance, compression.ratio,
##     curb.weight, drive.wheels, engine.location, engine.size, fuel,
##     height, highway.distance, HP, length, n.cylinders, peak.rot,
##     wheel.base, width
```

```
# Create some specific explanatory variables
fuel1 = factor(auto$fuel, levels = c("gas", "diesel"))
cylinders2 = factor(auto$n.cylinders==2)

# Fit the (a) linear model
fit_a = lm(log(highway.distance) ~ log(engine.size) +
             fuel1 + cylinders2 + log(curb.weight))
print(summary(fit_a)) #R-squared = 0.86
```

```
##
## Call:
## lm(formula = log(highway.distance) ~ log(engine.size) + fuel1 +
##     cylinders2 + log(curb.weight))
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.233995 -0.041647  0.001932  0.049527  0.282976
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       8.98644    0.44606  20.146  < 2e-16 ***
## log(engine.size) -0.10507    0.04814  -2.183   0.0302 *
## fuel1diesel       0.26022    0.02076  12.533  < 2e-16 ***
## cylinders2TRUE   -0.33559    0.04858  -6.908 6.54e-11 ***
## log(curb.weight) -0.90791    0.06770 -13.410  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0841 on 198 degrees of freedom
## Multiple R-squared:  0.8636, Adjusted R-squared:  0.8608
## F-statistic: 313.3 on 4 and 198 DF,  p-value: < 2.2e-16
```

For this model with limited number of explanatory variables, we get $R^2 = 0.86$.

```
# Fit the (b) linear model
fit_b = lm(log(highway.distance) ~ log(engine.size) + fuel1 + cylinders2 +
             log(curb.weight) + factor(brand) + factor(aspiration) + factor(bodystyle) +
             factor(drive.wheels) + factor(engine.location) + wheel.base +
             length + width + height + compression.ratio + log(HP) + peak.rot)
print(summary(fit_b)) #R-squared = 0.92
```

```
##
## Call:
```

```
## lm(formula = log(highway.distance) ~ log(engine.size) + fuel1 +
##     cylinders2 + log(curb.weight) + factor(brand) + factor(aspiration) +
##     factor(bodystyle) + factor(drive.wheels) + factor(engine.location) +
##     wheel.base + length + width + height + compression.ratio +
##     log(HP) + peak.rot)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.216410 -0.028696  0.000688  0.038577  0.242530
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  6.331e+00  8.263e-01   7.662 1.55e-12 ***
## log(engine.size)            -2.094e-01  1.026e-01  -2.040  0.04294 *
## fuel1diesel                 -1.172e-01  1.784e-01  -0.657  0.51193
## cylinders2TRUE              -3.318e-01  6.717e-02  -4.940 1.92e-06 ***
## log(curb.weight)            -5.066e-01  1.407e-01  -3.601  0.00042 ***
## factor(brand)audi           -2.856e-02  6.479e-02  -0.441  0.65991
## factor(brand)bmw             5.921e-02  5.736e-02   1.032  0.30355
## factor(brand)chevrolet       1.699e-01  6.693e-02   2.539  0.01206 *
## factor(brand)dodge           7.743e-02  6.092e-02   1.271  0.20557
## factor(brand)honda           8.311e-02  5.916e-02   1.405  0.16197
## factor(brand)isuzu           8.873e-02  6.219e-02   1.427  0.15560
## factor(brand)jaguar          1.539e-02  7.644e-02   0.201  0.84068
## factor(brand)mazda           4.452e-02  5.504e-02   0.809  0.41978
## factor(brand)mercedes-gas   -7.063e-02  6.901e-02  -1.024  0.30756
## factor(brand)mercury         5.633e-02  8.921e-02   0.631  0.52860
## factor(brand)mitsubishi      9.179e-02  5.874e-02   1.563  0.12009
## factor(brand)nissan          7.446e-02  5.367e-02   1.387  0.16723
## factor(brand)peugeot         2.388e-02  6.621e-02   0.361  0.71883
## factor(brand)plymouth        9.745e-02  6.042e-02   1.613  0.10872
## factor(brand)porsche         1.645e-01  7.743e-02   2.124  0.03519 *
## factor(brand)saab            5.951e-02  6.355e-02   0.937  0.35039
## factor(brand)subaru         -5.116e-04  5.575e-02  -0.009  0.99269
## factor(brand)toyota          6.663e-02  5.033e-02   1.324  0.18736
## factor(brand)volkswagen      3.831e-02  5.563e-02   0.689  0.49199
## factor(brand)volvo           7.985e-02  6.167e-02   1.295  0.19726
## factor(aspiration)turbo     -5.599e-02  3.060e-02  -1.830  0.06914 .
## factor(bodystyle)hardtop     2.362e-02  4.326e-02   0.546  0.58584
## factor(bodystyle)hatchback   1.916e-02  3.822e-02   0.501  0.61677
## factor(bodystyle)sedan       3.000e-02  4.031e-02   0.744  0.45784
## factor(bodystyle)wagon       2.637e-02  4.537e-02   0.581  0.56181
## factor(drive.wheels)fwd      5.738e-02  3.339e-02   1.718  0.08765 .
## factor(drive.wheels)rwd      5.717e-02  4.040e-02   1.415  0.15896
## factor(engine.location)rear -5.384e-02  7.918e-02  -0.680  0.49749
## wheel.base                  -1.480e-03  1.256e-03  -1.178  0.24036
## length                      -3.635e-04  7.007e-04  -0.519  0.60460
## width                        5.096e-03  3.170e-03   1.607  0.10989
## height                       9.779e-05  2.020e-03   0.048  0.96146
## compression.ratio            2.479e-02  1.257e-02   1.972  0.05025 .
## log(HP)                     -8.855e-02  8.052e-02  -1.100  0.27311
## peak.rot                    -7.099e-05  2.328e-05  -3.049  0.00268 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.06914 on 163 degrees of freedom
## Multiple R-squared:  0.9241, Adjusted R-squared:  0.9059
## F-statistic: 50.87 on 39 and 163 DF,  p-value: < 2.2e-16
```

For the other linear model, with many explanatory variables, we get $R^2 = 0.92$.

We see that the model (b) is better than model (a), also with respect to the adjusted R^2. However, due to the large number of explanatory variables, model (b) is not that easy to interpret and there are many non-significant variables.

Exercise 2.8

The idea is to gradually estimate the least square solution (2.7). From (2.22), we have that $\hat{\beta} = W^{-1}u$, where W and u are given on page 31. On the same page they introduce $W_{(j)}$ and $u_{(j)}$. From these, we can use (2.22) to create an estimate $\hat{\beta}_{(j)}$ based on the first j observations. Here we would need to use e.g. Cholesky decomposition as described on page 32. We then need to compute the deviance $D(\hat{\beta}_{(j)})$, given in (2.10), which uses y_i and $\hat{y}_i = x_i^t\hat{\beta}_{(j)}$, for $i = 1, 2, \ldots, j$. Note the that we only use the j first responses, and not up to n. We can then use $\hat{\beta}_{(j)}$ in (2.11) to obtain $s^2_{(j)}$, which is an estimate of σ^2. After this, we get the estimated standard errors of the components of $\hat{\beta}_{(j)}$ from (2.12). More precisely, by taking the sqaure root of the diagonal elements of $\widehat{\text{var}}(\hat{\beta}_{(j)}) = s^2_{(j)}W^{-1}_{(j)}$.

Introduction to Statistical Learning

Exercise 3.7.3

a)

Which answer is correct, and why?

1. i. For a fixed value of `IQ` and `GPA`, males earn more on average than females.
2. ii. For a fixed value of `IQ` and `GPA`, females earn more on average than males.
3. iii. For a fixed value of `IQ` and `GPA`, males earn more on average than females provided that the `GPA` is high enough.
4. iv. For a fixed value of `IQ` and `GPA`, females earn more on average than males provided that the `GPA` is high enough.

The least square line is given by

$$\hat{y} = 50 + 20\texttt{GPA} + 0.07\texttt{IQ} + 35\texttt{Gender} + 0.01\texttt{GPA} \times \texttt{IQ} - 10\texttt{GPA} \times \texttt{Gender}.$$

For males we get $\hat{y} = 50+20\texttt{GPA}+0.07\texttt{IQ}+0.01\texttt{GPA}\times\texttt{IQ}$, and for the females $\hat{y} = 85+10\texttt{GPA}+0.07\texttt{IQ}+0.01\texttt{GPA}\times\texttt{IQ}$. So the starting salary for males is higher than for females on average if and only if $50+20\texttt{GPA} \geq 85+10\texttt{GPA}$, which is equivalent to $\texttt{GPA} \geq 3.5$. Therefore iii. is the right answer.

b)

We are asked to predict the salary of a female with `IQ` of 110 and a `GPA` of 4.0. Just insert these values into the least square line for females above, and we get $\hat{y} = 85 + 40 + 7.7 + 4.4 = 137.1$, which gives us a starting salary of 137100$.

c)

False. To verify if the `GPA/IQ` has an impact on the quality of the model we need to test the hypothesis $H_0 : \hat{\beta}_4 = 0$ and look at the p-value associated with the t- or the F-statistic to draw a conclusion.

Exercise 3.7.7

It is claimed in the text that in the case of simple linear regression of Y onto X, the R^2 statistic (3.17) is equal to the square of the correlation between X and Y (3.18). We will now show it, and for simplicity, we assume that $\bar{x} = \bar{y} = 0$.

This means that

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{j=1}^{n}(x_j - \bar{x})^2} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{j=1}^{n} x_j^2}.$$

Recall that $\hat{y}_i = \hat{\beta}_1 x_i$. We have the following equalities

$$\begin{aligned}
R^2 &= 1 - \frac{RSS}{TSS} \\
&= 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{j=1}^{n} y_j^2} \\
&= 1 - \frac{\sum_{i=1}^{n}\left(y_i - \left(\frac{\sum_{j=1}^{n} x_j y_j}{\sum_{j=1}^{n} x_j^2}\right) x_i\right)^2}{\sum_{j=1}^{n} y_j^2} \\
&= \frac{\sum_j y_j^2 - (\sum_i y_i^2 - 2\sum_i y_i(\sum_j x_j y_j / \sum_j x_j^2)x_i + \sum_i(\sum_j x_j y_j / \sum_j x_j^2)^2 x_i^2)}{\sum_j y_j^2} \\
&= \frac{2(\sum_i x_i y_i)^2 / \sum_j x_j^2 - (\sum_i x_i y_i)^2 / \sum_j x_j^2}{\sum_j y_j^2} \\
&= \frac{(\sum_i x_i y_i)^2}{\sum_j x_j^2 \sum_j y_j^2} \\
&= \mathrm{Cor}(X, Y)^2.
\end{aligned}$$

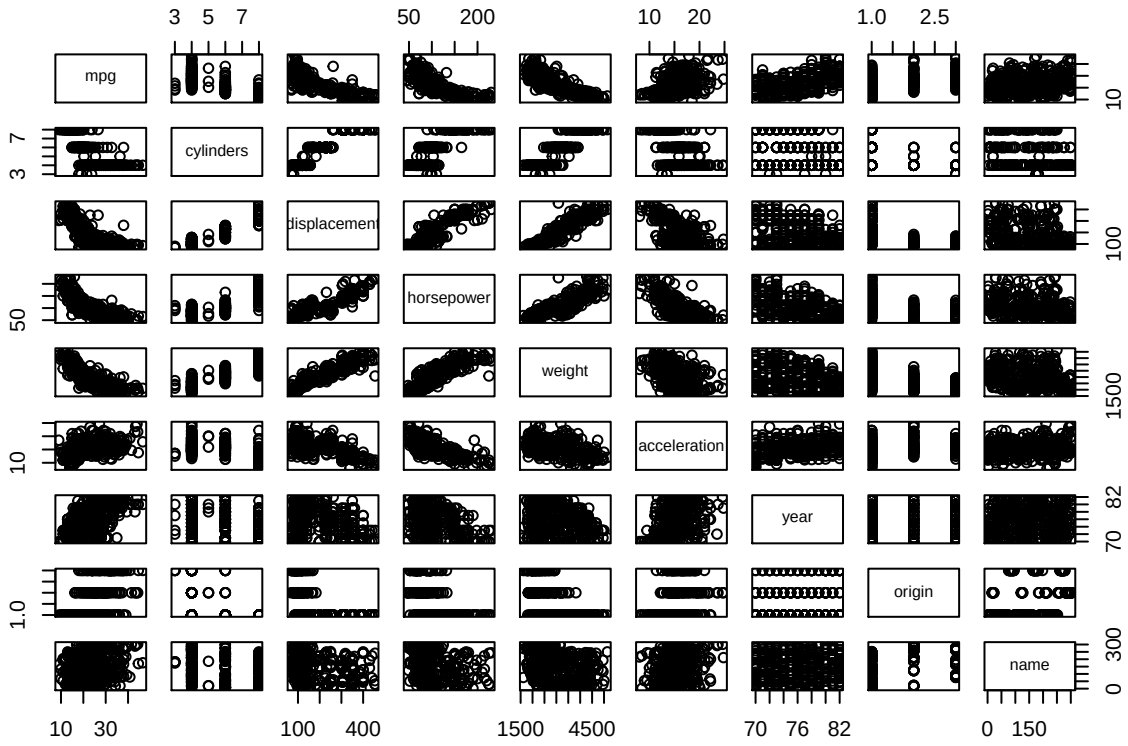
That is, we have showed that (3.17) is the squared of (3.18).

Exercise 3.7.9

a)

Produce a scatter plot matrix which includes all the variables in the data.

```
library("ISLR")
attach(Auto)
pairs(Auto)
```



b)

Compute the matrix of correlations between the variables using the function `cor()`.

```
names(Auto)
```

```
## [1] "mpg"          "cylinders"    "displacement" "horsepower"   "weight"
## [6] "acceleration" "year"         "origin"       "name"
```

```
cor(Auto[1:8])
```

```
##                     mpg  cylinders displacement horsepower     weight
## mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
## cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
## displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
## horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

c)

Use the `lm()` function to perform a multiple linear regression with `mpg` as the response and all other variables except `name` as the predictors. Use the `summary()` function to print the results.

```
lm.fit1 = lm(mpg~.-name, data=Auto)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

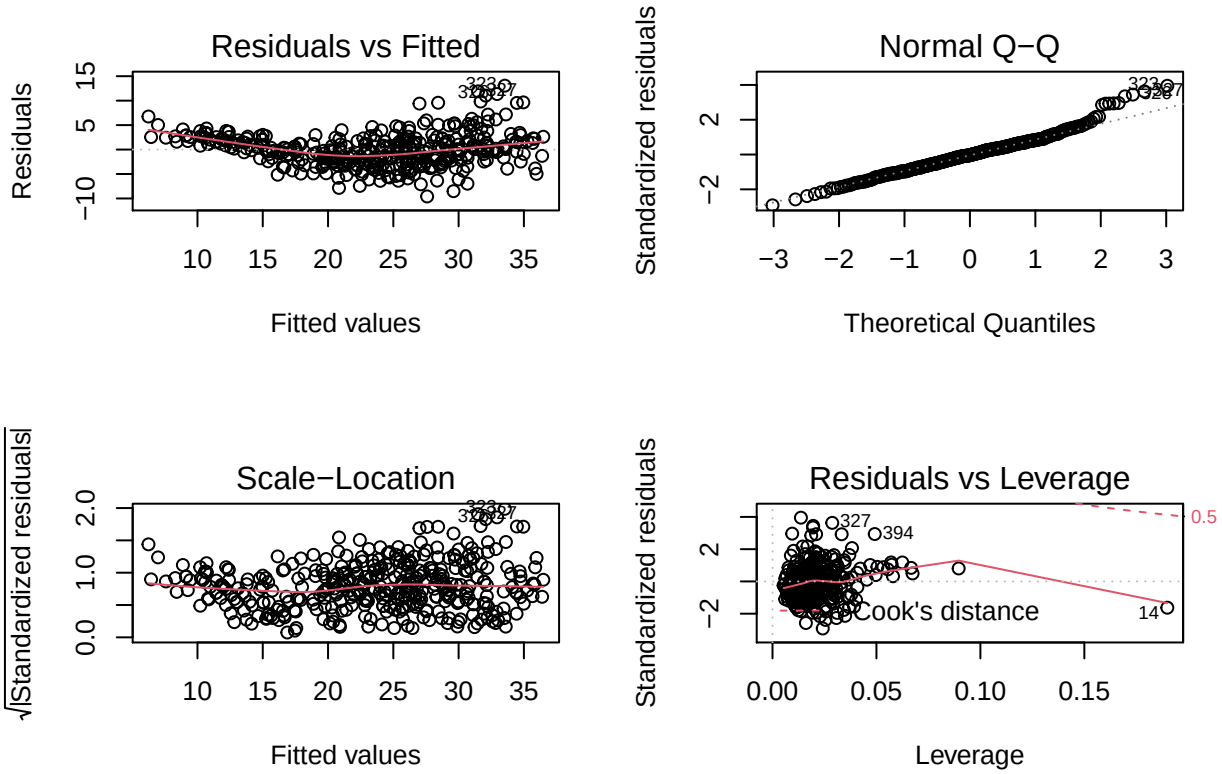
i. Yes, there is a relationship between the predictors and the response by testing the null hypothesis of whether all the regression coefficients are zero. The F-statistic is far from 1 (with a small p-value), indicating evidence against the null hypothesis.

ii. Looking at the p-values associated with each predictor's t-statistic, we see that `displacement`, `weight`, `year`, and `origin` have a statistically significant relationship, while `cylinders`, `horsepower`, and `acceleration` do not.

iii. The regression coefficient for `year`, 0.7508, suggests that for every one year, `mpg` increases by the coefficient. In other words, cars become more fuel efficient every year by almost 1 mpg / year.

d)

```
par(mfrow=c(2,2))
plot(lm.fit1)
```



The fit does not appear to be accurate because there is a discernible curve pattern to the residuals plots. From the leverage plot, point 14 appears to have high leverage, although not a high magnitude residual. There are possible outliers as seen in the plot of standardized residuals because there are data with a value greater than 3.

e)

From the correlation matrix, we obtained the two highest correlated pairs and use them in picking interaction effects.

```
lm.fit2 = lm(mpg ~ cylinders*displacement + displacement*weight)
summary(lm.fit2) #R-squared = 0.73
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * displacement + displacement *
##     weight)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -13.2934  -2.5184  -0.3476   1.8399  17.7723
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                 5.262e+01  2.237e+00  23.519  < 2e-16 ***
## cylinders                   7.606e-01  7.669e-01   0.992    0.322
## displacement               -7.351e-02  1.669e-02  -4.403 1.38e-05 ***
## weight                     -9.888e-03  1.329e-03  -7.438 6.69e-13 ***
## cylinders:displacement     -2.986e-03  3.426e-03  -0.872    0.384
## displacement:weight         2.128e-05  5.002e-06   4.254 2.64e-05 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.103 on 386 degrees of freedom
## Multiple R-squared:  0.7272, Adjusted R-squared:  0.7237
## F-statistic: 205.8 on 5 and 386 DF,  p-value: < 2.2e-16
```

From the p-values, we can see that the interaction between `displacement` and `weight` is statistically significant, while the interaction between `cylinders` and `displacement` is not.

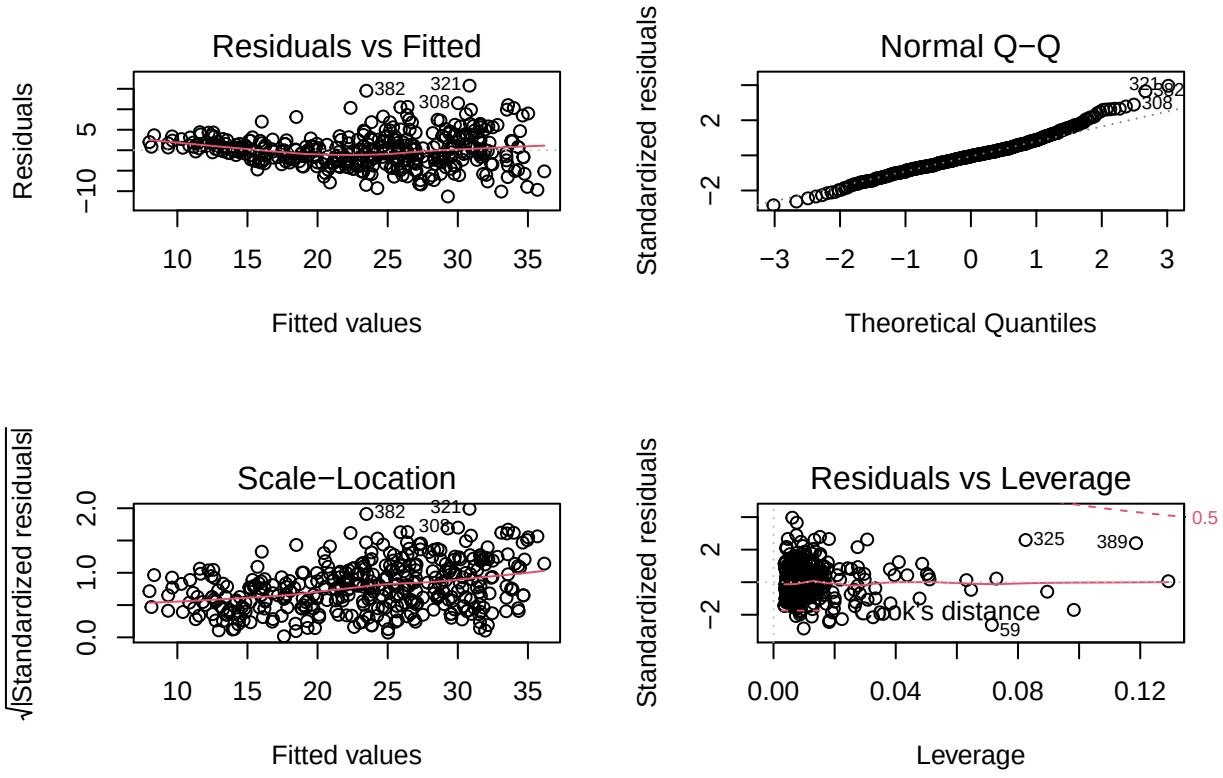
f)

Try a few diffferent transformations of the variables. This is one of many possible solutions.

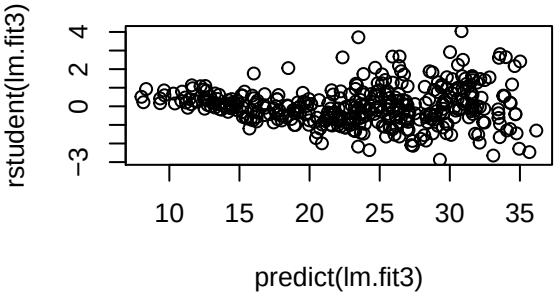
```
lm.fit3 = lm(mpg ~ log(weight) + sqrt(horsepower) + acceleration + I(acceleration^2))
summary(lm.fit3) #R-squared = 0.72
```

```
##
## Call:
## lm(formula = mpg ~ log(weight) + sqrt(horsepower) + acceleration +
##     I(acceleration^2))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.2932  -2.5082  -0.2237   2.0237  15.7650
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       178.30303   10.80451  16.503  < 2e-16 ***
## log(weight)       -14.74259    1.73994  -8.473 5.06e-16 ***
## sqrt(horsepower)   -1.85192    0.36005  -5.144 4.29e-07 ***
## acceleration       -2.19890    0.63903  -3.441 0.000643 ***
## I(acceleration^2)   0.06139    0.01857   3.305 0.001037 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.99 on 387 degrees of freedom
## Multiple R-squared:  0.7414, Adjusted R-squared:  0.7387
## F-statistic: 277.3 on 4 and 387 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.fit3)
```




```
plot(predict(lm.fit3), rstudent(lm.fit3))
```



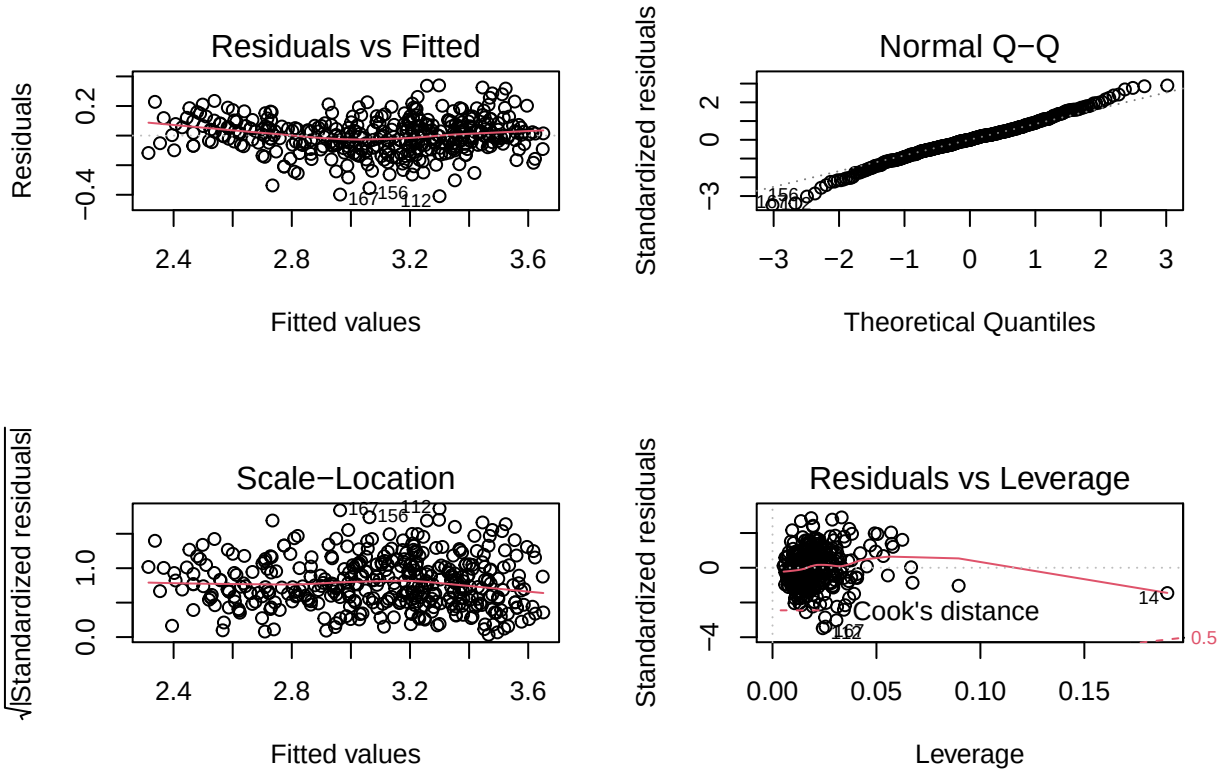
Apparently, from the p-values, the log(`weight`), $\sqrt{\texttt{horsepower}}$, and $\texttt{acceleration}^2$ all have statistical significance of some sort. The residuals plot has less of a discernible pattern than the plot of all linear regression terms. The studentized residuals displays potential outliers (>3). The leverage plot indicates more than three points with high leverage. However, 2 problems are observed from the above plots: 1) the residuals vs fitted plot indicates heteroskedasticity (unconstant variance over mean) in the model. 2) The Q-Q plot indicates somewhat unnormality of the residuals. So, a better transformation need to be applied to our model. From the correlation matrix in 9a., `displacement`, `horsepower` and `weight` show a similar nonlinear pattern against our response `mpg`. This nonlinear pattern is very close to a log form. So in the next attempt, we use log(`mpg`) as our response variable.

```
lm.fit2=lm(log(mpg) ~ cylinders + displacement + horsepower + weight +
             acceleration + year + origin, data=Auto)
summary(lm.fit2) # R-squared = 0.88
```

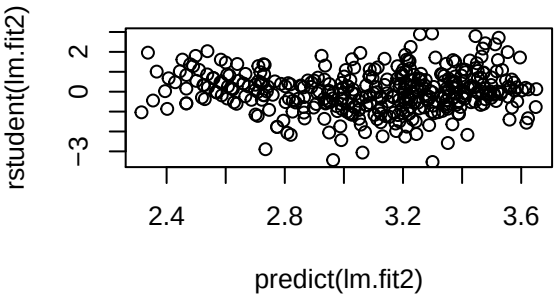
```
##
## Call:
## lm(formula = log(mpg) ~ cylinders + displacement + horsepower +
##     weight + acceleration + year + origin, data = Auto)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.40955 -0.06533  0.00079  0.06785  0.33925
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.751e+00  1.662e-01  10.533  < 2e-16 ***
## cylinders    -2.795e-02  1.157e-02  -2.415  0.01619 *
## displacement  6.362e-04  2.690e-04   2.365  0.01852 *
## horsepower   -1.475e-03  4.935e-04  -2.989  0.00298 **
## weight       -2.551e-04  2.334e-05 -10.931  < 2e-16 ***
## acceleration -1.348e-03  3.538e-03  -0.381  0.70339
## year          2.958e-02  1.824e-03  16.211  < 2e-16 ***
## origin        4.071e-02  9.955e-03   4.089 5.28e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1191 on 384 degrees of freedom
## Multiple R-squared:  0.8795, Adjusted R-squared:  0.8773
## F-statistic: 400.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.fit2)
```




```
plot(predict(lm.fit2),rstudent(lm.fit2))
```



The outputs show that log transform of `mpg` yield better model fitting (better R^2, normality of residuals).

Exercise 3.7.10

a)

Fit a multiple regression model to predict `Sales` using `Price`, `Urban` and `US`.

```
# Get the data and attach it
# install.packages("ISLR")
library(ISLR)
attach(Carseats)

# Fit the linear model
lm.fit = lm(Sales ~ Price + Urban + US)
summary(lm.fit) # R-squared: 0.24
```

```
##
## Call:
## lm(formula = Sales ~ Price + Urban + US)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9206 -1.6220 -0.0564  1.5786  7.0581
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.043469   0.651012  20.036  < 2e-16 ***
## Price       -0.054459   0.005242 -10.389  < 2e-16 ***
## UrbanYes    -0.021916   0.271650  -0.081    0.936
## USYes        1.200573   0.259042   4.635 4.86e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.472 on 396 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2335
## F-statistic: 41.52 on 3 and 396 DF,  p-value: < 2.2e-16
```

b)

`Price`: The linear regression suggests a relationship between `Price` and `Sales` given the low p-value of the t-statistic. The coefficient states a negative relationship between `Price` and `Sales`: as `Price` increases, `Sales` decreases (while the other predictors are kept fixed).

`UrbanYes`: The linear regression suggests that there is not a relationship between the location of the store and the number of `Sales` based on the high p-value of the t-statistic.

`USYes`: The linear regression suggests there is a relationship between whether the store is in the US or not and the amount of `Sales`. The coefficient states a positive relationship between `USYes` and `Sales`: if the store is in the US, the `Sales` will increase by approximately 1201 units (while the other predictors are kept fixed).

c)

The model may be written as $\texttt{Sales} = 13.04 + -0.05\texttt{Price} + -0.02\texttt{UrbanYes} + 1.20\texttt{USYes} + \epsilon$. with $\texttt{Urban} = 1$ if the store is in an urban location and 0 if not, and $\texttt{US} = 1$ if the store is in the US and 0 if not.

d)

We can reject the null hypothesis $H_0 : \beta_j = 0$ for predictors `Price` and `USYes`, based on their p-values. We can also reject the hypothesis related to the F-statistic.

e)

Fit a new linear model of only significant predictors.

```
# Fit the new linear model
lm.fit2 = lm(Sales ~ Price + US)
summary(lm.fit2) #R-squared = 0.24
```

```
##
## Call:
## lm(formula = Sales ~ Price + US)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9269 -1.6286 -0.0574  1.5766  7.0515
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 13.03079    0.63098  20.652  < 2e-16 ***
## Price       -0.05448    0.00523 -10.416  < 2e-16 ***
## USYes        1.19964    0.25846   4.641 4.71e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.469 on 397 degrees of freedom
## Multiple R-squared:  0.2393, Adjusted R-squared:  0.2354
## F-statistic: 62.43 on 2 and 397 DF,  p-value: < 2.2e-16
```

f)

Based on the RSE and R^2 of the linear regressions, they both fit the data similarly, with linear regression from e) fitting the data slightly better. Essentially, about 23.93% of the variability is explained by the model in e).

g)

Using the model from e), we obtain 95% confidence intervals for the coefficient(s) witht the following code:

```
# Compute confidence intervals
confint(lm.fit2)
```

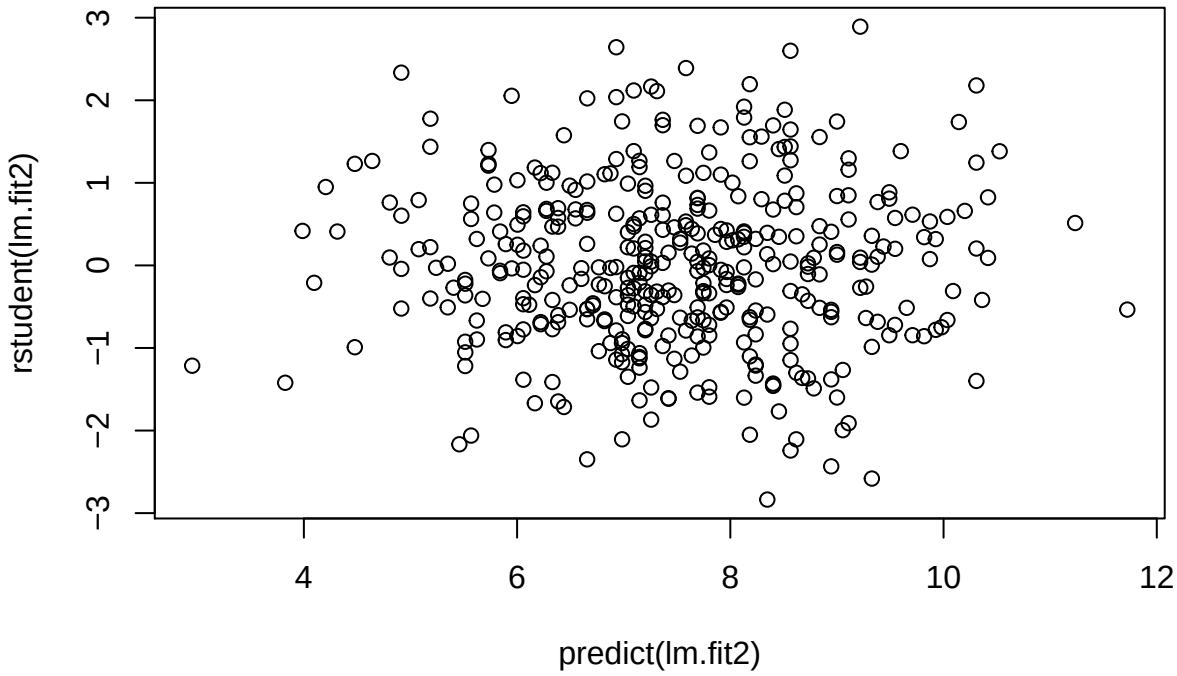
```
##                   2.5 %      97.5 %
```

```
## (Intercept) 11.79032020 14.27126531
## Price       -0.06475984 -0.04419543
## USYes        0.69151957  1.70776632
```

h)

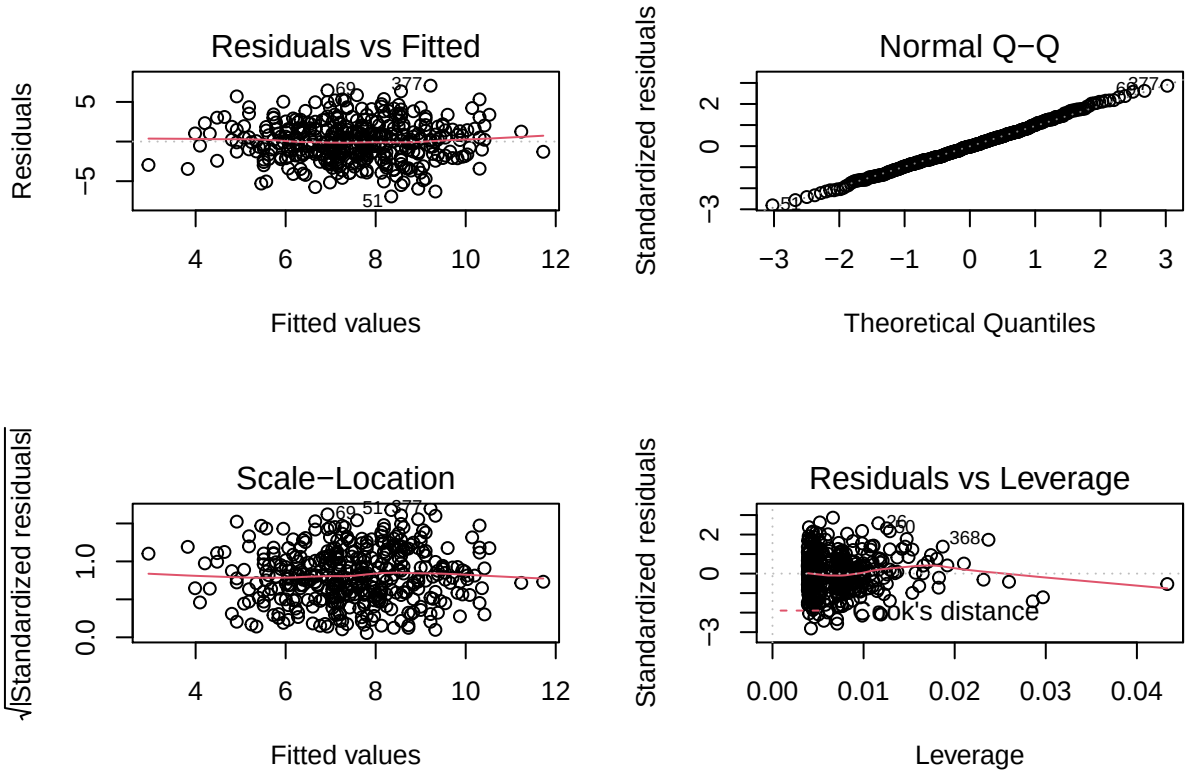
Check if there is evidence of outliers or high leverage observations in the model from e):

```
par(mfrow=c(1,1))
plot(predict(lm.fit2), rstudent(lm.fit2))
```



All studentized residuals appear to be bounded by -3 to 3, so no potential outliers are suggested from the linear regression. (Some people use -2 and 2, then we do have outliers.)

```
par(mfrow=c(2,2))
plot(lm.fit2)
```



I would say no, but this can be discussed.