```r
set.seed(42)
```

Arbeitsblatt 5

Aufgabe 1

Erinnerung. Wir nehmen die Gauss-Markov-Annahmen an (Fehlerterme haben Erwartungswert 0, gleiche Varianz und sind unkorreliert) und folglich gilt:

$$
\begin{aligned}
&\mathrm{Cov}(\epsilon_i, \epsilon_j) = 0 \quad \forall i \neq j \\
&\mathrm{E}(\epsilon_i) = \theta \\
&\mathrm{V}(\epsilon_i) = \sigma^2 \\
&Y_i = X_i b + \epsilon_i \\
&\hat{Y}_i = (X\hat{b})_i \\
&\hat{b} = (X^T X)^{-1} X^T Y \\
&\hat{b}_i = \left((X^T X)^{-1} X^T Y\right)_i \\
&H = X(X^T X)^{-1} X^T \\
&H = H \cdot H \\
&H = H^T
\end{aligned}
$$

X_i ist hier eine Zeile der Designmatrix und H_i eine Zeile der Hatmatrix.

Zuerst stellen wir fest, dass $X_i b$ und ϵ_i unabhängig voneinander sind und $X_i b$ varianzlos ist. Dann erhalten wir

$$
\mathrm{V}(Y_i) = \mathrm{V}(X_i b + \epsilon_i) = \mathrm{V}(X_i b) + \mathrm{V}(\epsilon_i) = \sigma^2.
$$

Ferner können wir berechnen

$$
\begin{aligned}
\mathrm{V}(Y_i) &= \mathrm{V}((X\hat{b})_i) = \mathrm{V}((HY)_i) = H_i \mathrm{V}(Y) H_i^T = H_i \mathrm{V}(Xb + \epsilon) H_i^T \\
&= H_i \,\mathrm{diag}(\sigma^2) H_i^T = h_{ii}\sigma^2.
\end{aligned}
$$

Ähnlich können wir berechnen

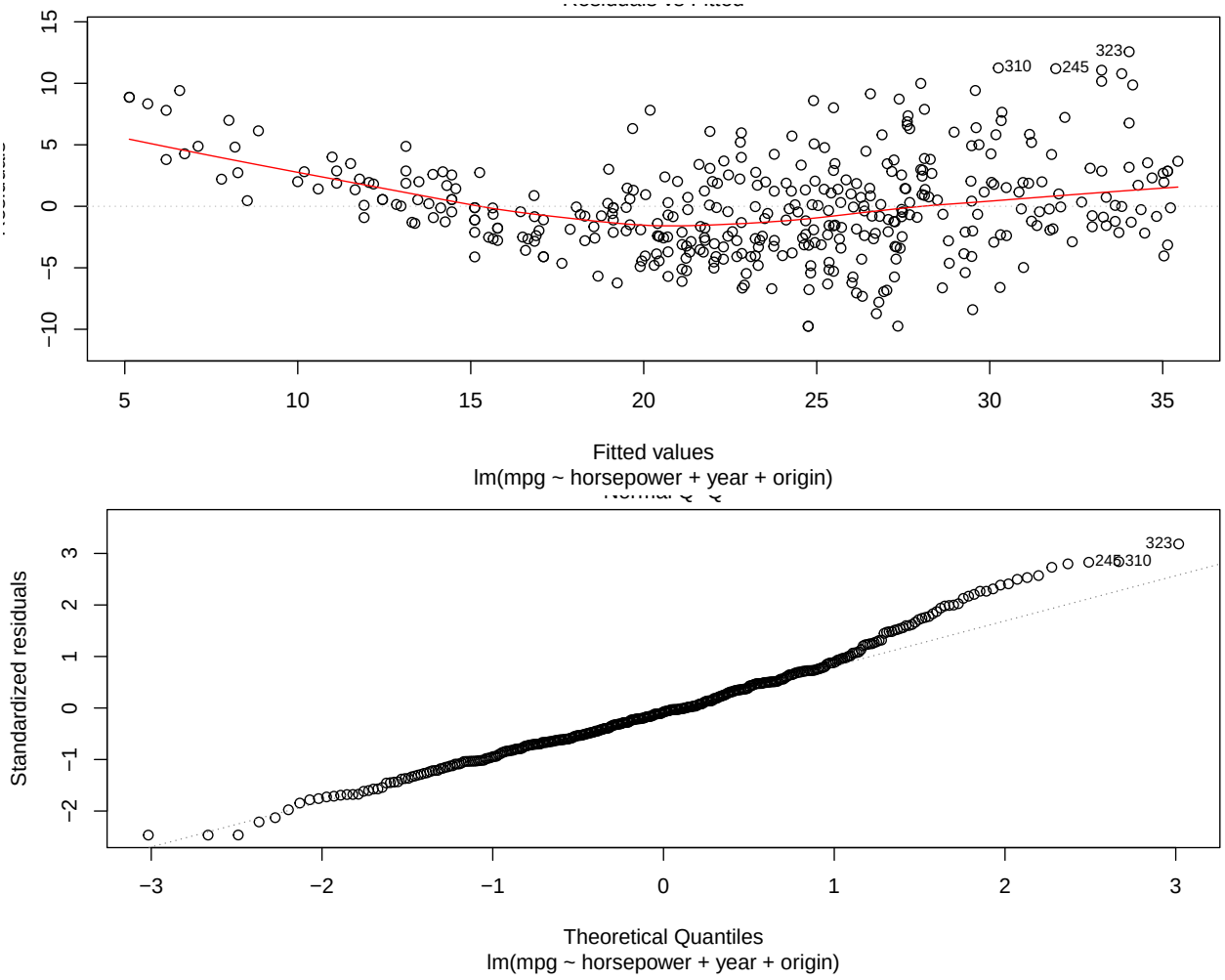
$$
\begin{aligned}
\mathrm{Cov}(Y_i, \hat{Y}_i) &= \mathrm{Cov}(Y_i, H_i Y) \\
&= H_i \,\mathrm{Cov}(x_i b + \epsilon_i, (Xb + \epsilon)_i) H_i^T \\
&= h_{ii}\sigma^2.
\end{aligned}
$$

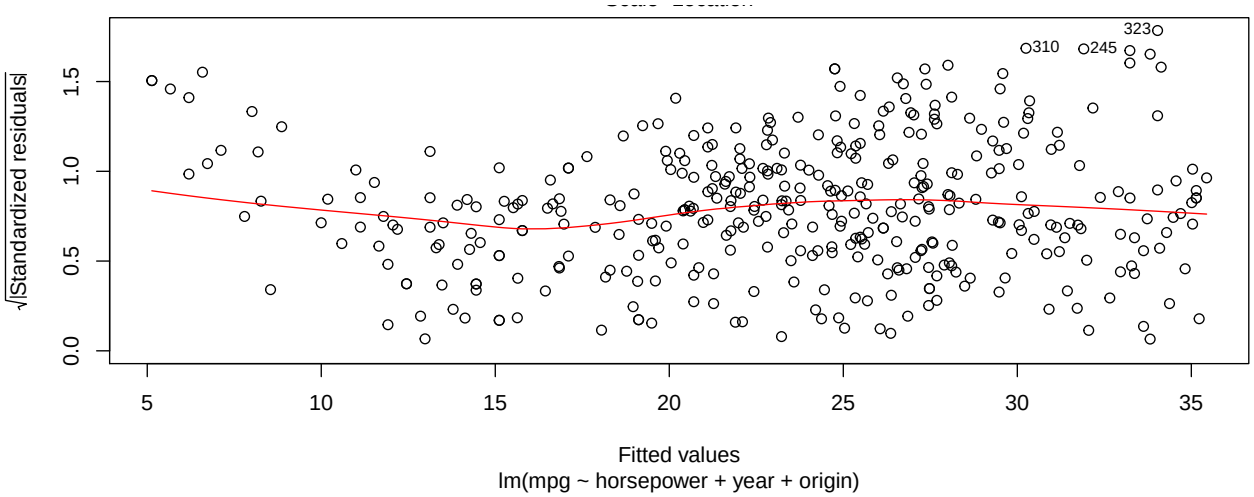
Und schließ durch Einsetzen erhalten wir

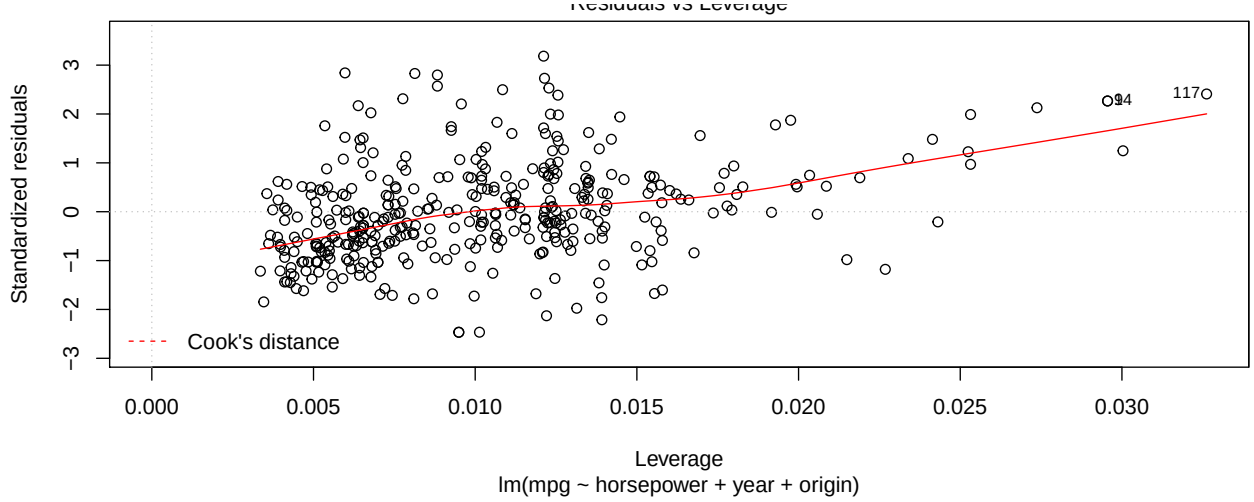
$$\begin{aligned}\mathrm{Cor}(Y_i, \hat{Y}_i) &= \frac{\mathrm{Cov}(Y_i, \hat{Y}_i)}{\sqrt{\mathrm{V}(Y_i)\mathrm{V}(\hat{Y}_i)}} \\ &= \frac{h_{ii}\sigma^2}{\sqrt{\sigma^2 h_{ii}\sigma^2}} \\ &= \sqrt{h_{ii}}.\end{aligned}$$

Aufgabe 2

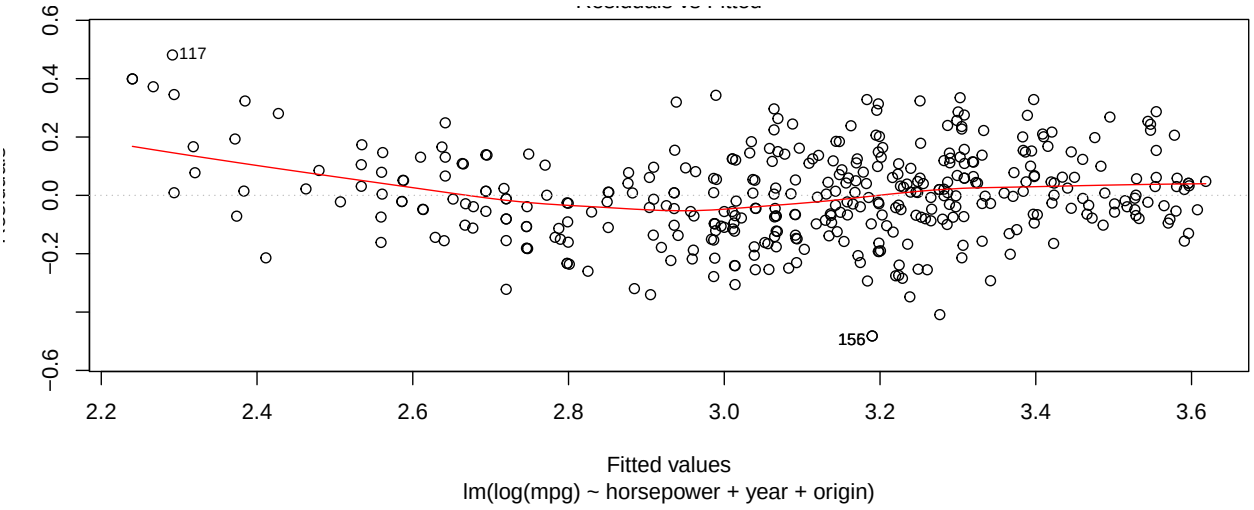
```
library("ISLR")
model_mpg <- lm(mpg~horsepower+year+origin, data = Auto)
plot(model_mpg)
```

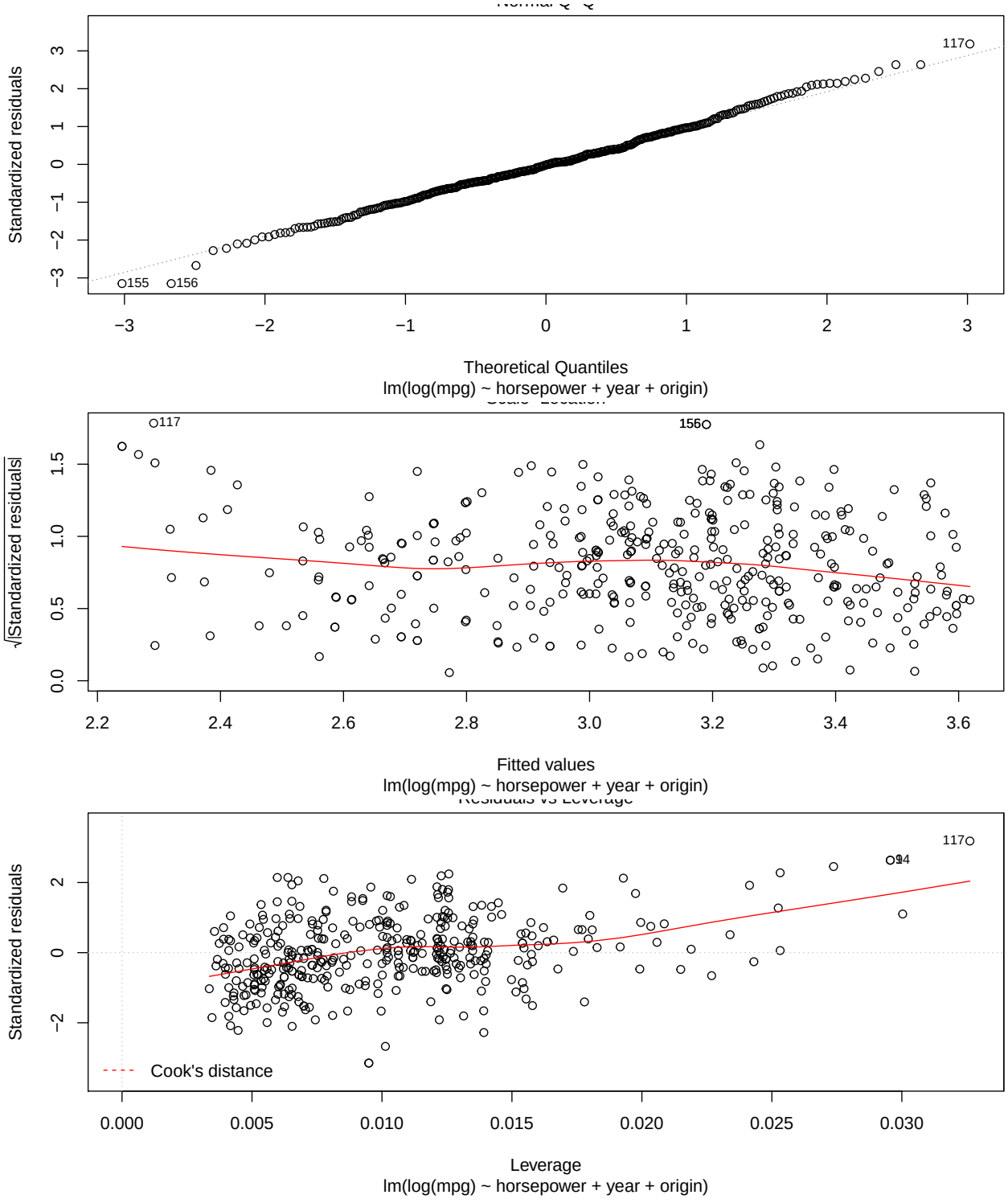







```r
model_log_mpg <- lm(log(mpg)~horsepower+year+origin, data = Auto)
plot(model_log_mpg)
```





a)

Wenn wir uns die Grafik “Residuals vs Fitted” ansehen, sehen wir, dass die Daten kein offensichtliches, eindeutiges Muster haben. Obwohl die Kurven leicht gewölbt sind, haben sie gleichmäßige Residuen um die horizontale Linie herum verteilt, ohne ein ausgeprägtes Muster. Dies ist ein guter Hinweis darauf, dass es sich nicht um eine nichtlineare Beziehung handelt.

Die Residuen sollten normalverteilt sein, und der Q-Q Plot ist ein gutes Instrument, um dies zu überprüfen. Für beide Modelle zeigt der Q-Q-Plot eine ziemlich gute Anpassung auf die Linie. Nach oben hin weichen die Punkte für das mpg-Modell deutlicher von der Linie ab, als im log(mpg)-Modell.

Die "Scale-Location"-Plots testen die lineare Regressionsannahme der gleichen Varianz (Homoscedastizität), d.h. dass die Residuen die gleiche Varianz entlang der Regressionslinie aufweisen.

In unseren beiden Modellen sind die Residuen relativ gut über und unter einer ziemlich horizontalen Linie verteilt.

Die "Residuals vs Leverage"-Plots können verwendet werden, um einflussreiche Datenpunkte im Datensatz zu finden. Ein einflussreicher Datenpunkt ist einer, der, wenn er entfernt wird, das Modell beeinflusst, so dass seine Einbeziehung oder sein Ausschluss in Betracht gezogen werden sollte.

Ein einflussreicher Datenpunkt kann ein Ausreißer sein oder auch nicht, und der Zweck dieser Grafik ist es, Fälle zu identifizieren, die einen hohen Einfluss auf das Modell haben. Ausreißer neigen dazu, Leverage und damit Einfluss auf das Modell auszuüben.

Ein einflussreicher Fall erscheint oben rechts oder unten links in der Grafik innerhalb einer roten Linie, die Cook's Distance markiert.

In beiden Modellen gibt es keine offensichtlichen Ausreißer (laut Cook's Distance).

Abschließend würden wir das log(mpg)-Modell bevorzugen, obwohl es kaum Unterschiede in den Residuen gibt. Alleine der Q-Q-Plot des log(mpg)-Modells deutet auf eine bessere Anpassung der Residuen auf die Normalverteilung hin als im mpg-Modell.

b)

Der Punkt 116 ist in beiden Modellen des Öfteren markiert. Schauen wir uns diesen näher an:

```
pt_116 <- Auto[116,]
summary <- summary(Auto)
c(pt_116)
#> $mpg
#> [1] 16
#>
#> $cylinders
#> [1] 8
#>
#> $displacement
#> [1] 400
#>
#> $horsepower
#> [1] 230
#>
#> $weight
#> [1] 4278
#>
#> $acceleration
#> [1] 9.5
#>
#> $year
#> [1] 73
#>
#> $origin
#> [1] 1
#>
```

```
#> $name
#> [1] pontiac grand prix
#> 304 Levels: amc ambassador brougham ... vw rabbit custom
summary
#>       mpg          cylinders      displacement     horsepower
#>  Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0
#>  1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0
#>  Median :22.75   Median :4.000   Median :151.0   Median : 93.5
#>  Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5
#>  3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0
#>  Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0
#>
#>      weight      acceleration        year           origin
#>  Min.   :1613   Min.   : 8.00   Min.   :70.00   Min.   :1.000
#>  1st Qu.:2225   1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000
#>  Median :2804   Median :15.50   Median :76.00   Median :1.000
#>  Mean   :2978   Mean   :15.54   Mean   :75.98   Mean   :1.577
#>  3rd Qu.:3615   3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
#>  Max.   :5140   Max.   :24.80   Max.   :82.00   Max.   :3.000
#>
#>                  name
#>  amc matador       :  5
#>  ford pinto        :  5
#>  toyota corolla    :  5
#>  amc gremlin       :  4
#>  amc hornet        :  4
#>  chevrolet chevette:  4
#>  (Other)           :365
```

Hier wird deutlich, dass der Punkt 116 ein Extremfall in unserem Datensatz ist.

c)

```
library(leaps)
#> Error in library(leaps): there is no package called 'leaps'
regfit.full = regsubsets(log(mpg)~cylinders+displacement+horsepower+weight+acceleration+year+origin, da
#> Error in regsubsets(log(mpg) ~ cylinders + displacement + horsepower + : could not find function "re
reg.summary = summary(regfit.full)
#> Error in summary(regfit.full): object 'regfit.full' not found
```

```
plot(reg.summary$bic, xlab="Number of Variables", ylab="BIC", type="l")
#> Error in plot(reg.summary$bic, xlab = "Number of Variables", ylab = "BIC", : object 'reg.summary' no
plot(reg.summary$cp, xlab="Number of Variables", ylab="AIC", type="l")
#> Error in plot(reg.summary$cp, xlab = "Number of Variables", ylab = "AIC", : object 'reg.summary' not
plot(reg.summary$rsq, xlab="Number of Variables", ylab="RSq", type="l")
#> Error in plot(reg.summary$rsq, xlab = "Number of Variables", ylab = "RSq", : object 'reg.summary' no
```

BIC:

- BIC wird für das Modell mit nur 4 Kovariaten minimal.

AIC:

- Nach AIC ist die Entscheidungsfindung nicht ganz so eindeutig. Auch hier ist bei 4 Kovariaten der AIC sehr gering, jedoch steigt der AIC bei wachsender Anzahl an Kovariaten nicht mehr an, sinkt aber auch nicht deutlich weiter.

R^2 :

- Ein hohes R^2 spricht für ein gutes Modell. Allerdings gibt es, anders als bei AIC und BIC, keinen Strafterm, welcher die Anzahl an Kovariaten "bestraft". Deshalb steigt das R^2 mit Anzahl dr Kovariaten (leicht) an.

Wir würden uns für ein Modell mit 4 Kovariaten entscheiden, da dort BIC und AIC sehr klein sind und R^2 groß ist.

```
coef(regfit.full, 4)
#> Error in coef(regfit.full, 4): object 'regfit.full' not found
```

d)

```
library(car)
#> Error in library(car): there is no package called 'car'

model_all_log <- lm(log(mpg)~cylinders+displacement+horsepower+weight+acceleration+year+origin, data=Au

model_4_log <- lm(log(mpg)~horsepower+weight+year+origin, data=Auto)
vif(model_all_log)
#> Error in vif(model_all_log): could not find function "vif"
cat("\n")
vif(model_4_log)
#> Error in vif(model_4_log): could not find function "vif"
```

Schaut man sich die VIF-Werte an, wird deutlich, dass im Modell mit allen Kovariaten definitiv Multikollinearität ein Problem darstellt.

Im Modell, dass wir durch die Subset-Selektion gefunden haben, sind die VIF-Werte geringer. "Pi-Mal-Daumen" ist die Faustregel:

- 1 = unkorreliert
- zwischen 1 und 5 = moderat korreliert
- größer 5 = stark korreliert

Die VIF-Werte für horsepower und weight liegen beide bei circa 5. Nimmt man nun die Kovariate mit dem höchsten VIF-Wert aus dem Modell heraus, ergeben sich folgende VIF-Werte für das neue Modell:

```
model_3_log <- lm(log(mpg)~horsepower+year+origin, data=Auto)
vif(model_3_log)
#> Error in vif(model_3_log): could not find function "vif"
```

Eindeutig wird der Einfluss von weight durch horsepower bereits gut "erklärt". Die VIF-Werte liegen nun im annehmbaren Bereich. Das Modell mit horsepower, year und origin scheint das robustere zu sein.