Week 12 Live Session - Multiple OLS Regression Inference

w203 Instructional Team

Announcements

Useful functions in R:

model coefficients: coefficients(fit)

predicted values: fitted(fit)

residuals: residuals(fit)

heteroskedasticity-robust covariance matrix for model parameters: vcovHC(fit)

heteroskedasticity-robust standard errors and hypothesis tests: coeftest(fit, vcov = vcovHC)

CIs for model parameters (at 95%): confint(fit, level=0.95) (Warning: not robust to heteroskedasticity)

For heteroskedasticity-robust confidence intervals, get the variance of each coefficient from vcovHC, take the square root to get the standard error, get the proper t critical values from qt, and construct manually.

Variance of OLS Estimators

Recall the expression for the variance of each OLS slope coefficient:

$$var(\hat{\beta}_j) = \frac{\sigma^2}{SST_j(1 - R_j^2)}$$

For each component of this equation, explain (1) what it means, and (2) why it moves the standard error of β_j up or down.

1. σ^2
2. $\frac{1}{SST_j}$
3. $\frac{1}{1-R_j^2}$

Component 3 has a special name: the Variance Inflation Factor. You can find the variance inflation factor for each variable in a linear model using the vif function in the car package. Interpreting VIFs depends very much on context, but a VIF of 10 would usually be considered very high.

To get the variance of each coefficient in R, we would typically get the diagonal elements of the robust covariance matrix, diag(vcovHC(model))

To get the standard error of a coefficient, take the square root of the variance.

R Exercise

In this analysis, we will use the mtcars dataset which is a dataset that was extracted from the 1974 Motor Trend US magazine, and comprises fuel consumption and 10 aspects of automobile design and performance for 32 automobiles (1973-74 models). The dataset is automatically available when you start R. For more information about the dataset, use the R command: help(mtcars)

Some useful libraries for multivariate ols regression:

```
library(car)
library(lmtest)
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
library(sandwich)
```

Q1.1: Using the mtcars data, run a multiple linear regression to find the effect of displacement (disp), gross horsepower (hp), weight (wt), and rear axle ratio (drat) on the miles per gallon (mpg).

```
  model <- lm(mpg~disp+hp+wt+drat, data=mtcars)
```

Q1.2: For each of the 6 CLM assumptions, assess whether the assumption holds. Where possible, demonstrate multiple ways of assessing an assumption. When an assumption appears violated, state what steps you would take in response.

a. Linear population model
b. Random Sampling
c. No perfect multicollinearity
d. Zero-conditional mean
e. Homoskedasticity
f. Normality of Errors

- checking assumptions as follows

a. We don't have to check the linear population model, because we haven't constrained the error term, so there's nothing to check at this point.

b. To check random sampling, we need background knowledge of how the data was collected. Unfortunately, the description in R does not explain much about how the cars in the dataset were selected.

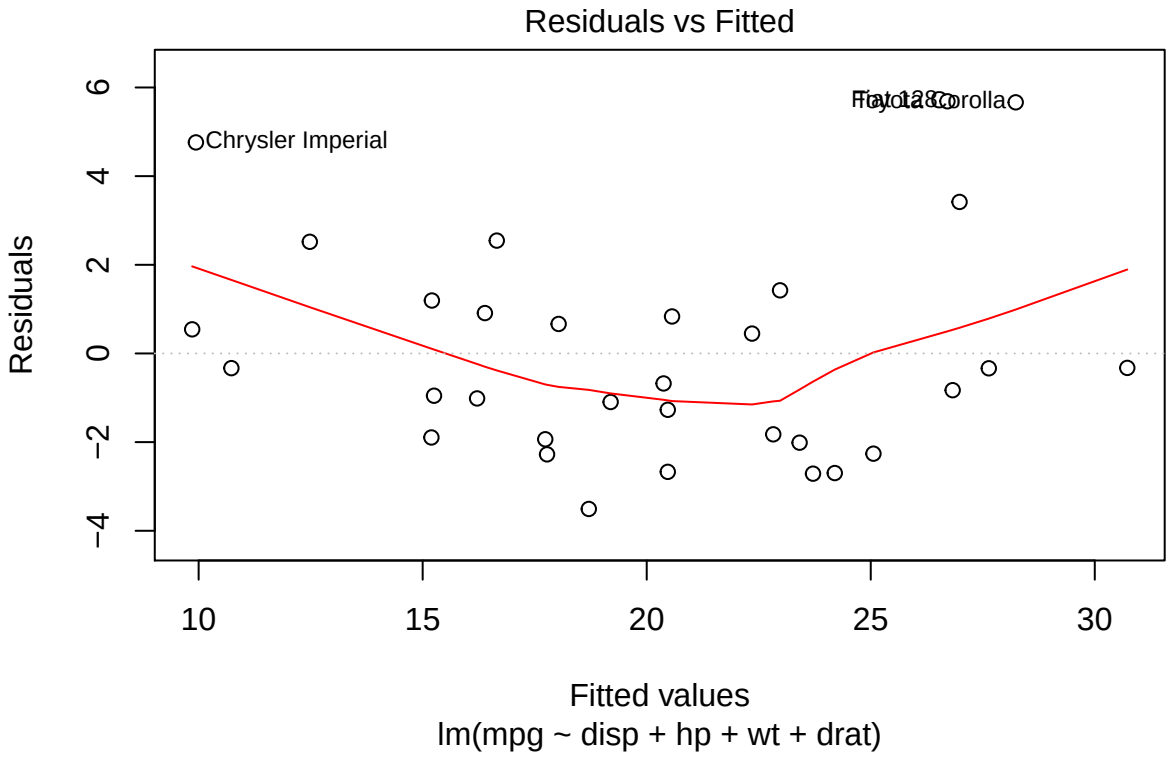
```
?mtcars
```

In general, we may be concerned about possible problems with independence. For example, car makers might not want to create cars that are too similar to each other, to avoid cannibalizing sales. They may pursue a more complicated strategy of designing a car for each of several market segments. Finally, other car manufacturers may imitate particularly successful models. For all these reasons, knowing the data on one car may inform the data we expect for another car. If there are a large number of car manufacturers, we may expect these clustering (and also anti-clustering) effects to be small.

c. No need to explicitly check for perfect collinearity, because R will alert us if this rare condition happens.

We start looking at the diagnostic plots:

```
plot(model, which = 1)
```



Notice the clear deviation from zero conditional mean, indicated by the parabolic shape. This means that our coefficients will be biased.

First, we check to see if we have a large sample:

```
nrow(mtcars)
```

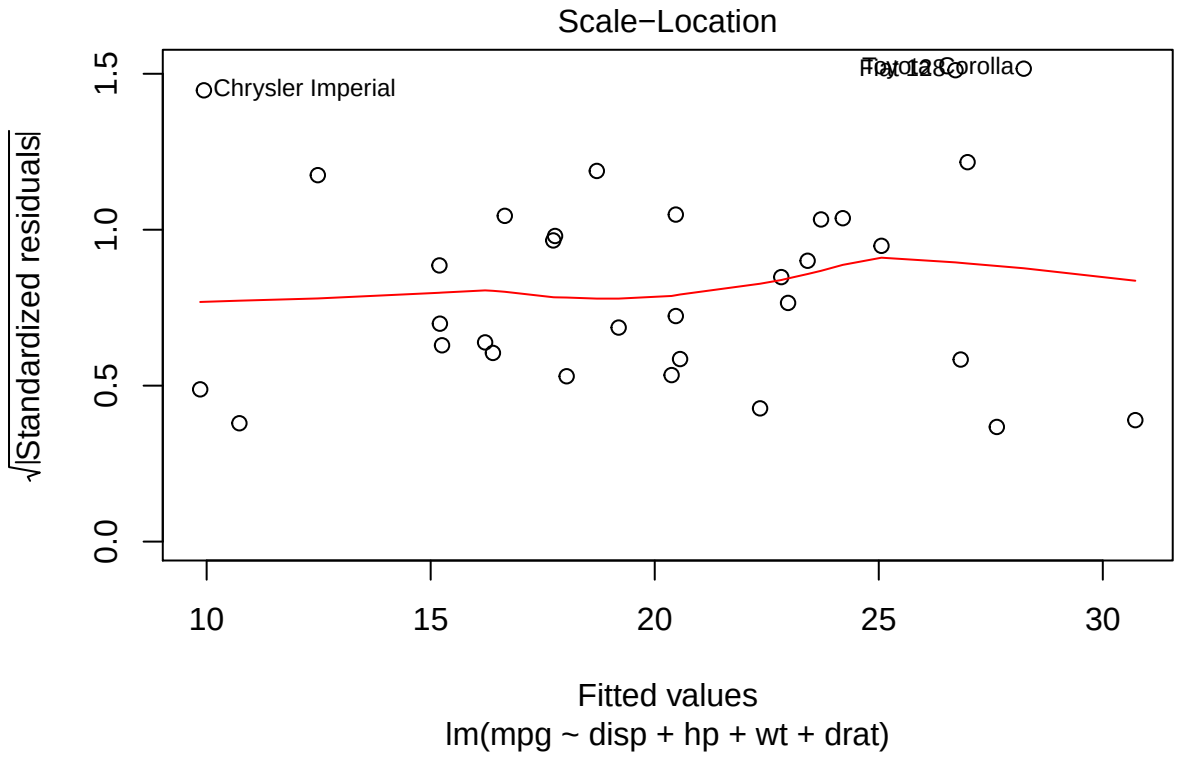
```
## [1] 32
```

n=32, so we can try to establish consistency. However, this still requires an assumption of exogeneity, $cov(x, u) = 0$, which we cannot test using our diagnostic plots. Instead we have to use our background knowledge and ask whether there are omitted variables hidden in u, which may be correlated with our x's. In this case, there are some possible omitted variables we should worry about. For example, does the car have a setting for more efficient gear selection, or an exhuast system tuned for fuel efficiency, both of which may be more common with smaller engines.

If we think we can't meet exogeneity, one option is to give up on building a causal model. If we only want to describe the association between mpg and our other variables, CLT 1-3 are sufficient for consistency.

Our residuals versus fitted values plot doesn't seem to indicate heteroskedasticity - the band seems to have even thickness. However, with this number of data points, it can be hard to tell. The scale location plot gives us another way to assess this assumption:

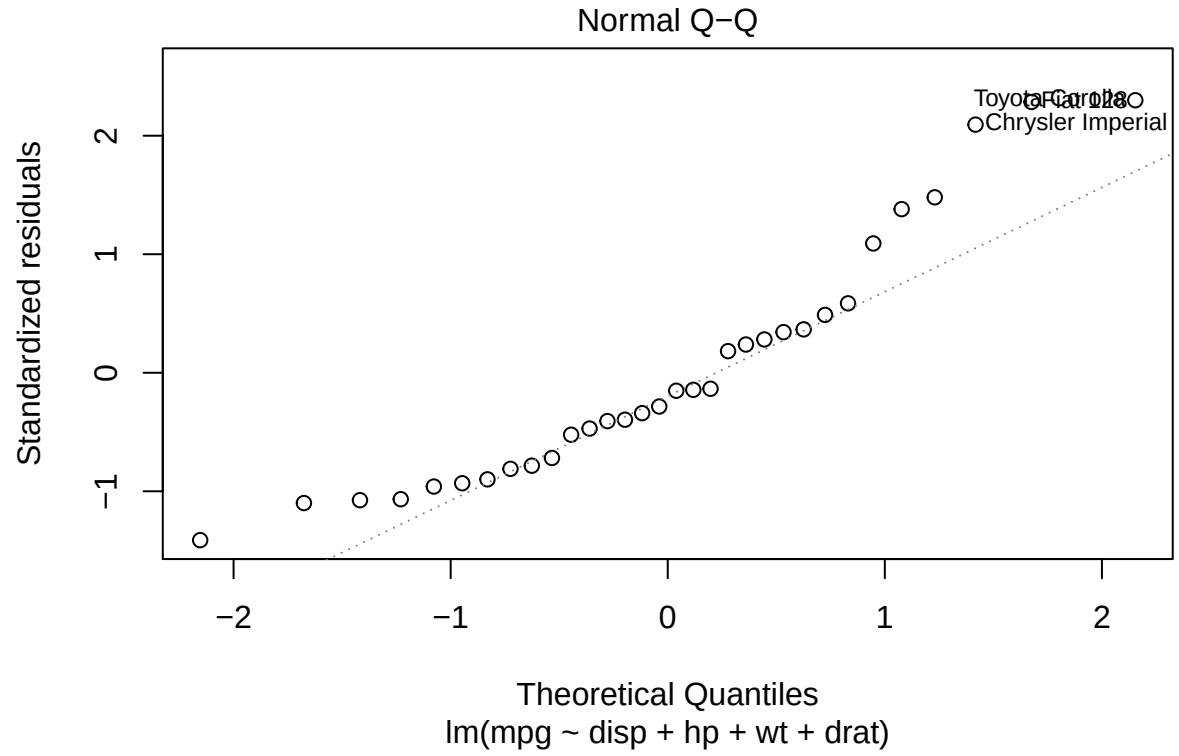
```
plot(model, which = 3)
```



The flat red line also suggests homoskedasticity. Despite this evidence, we will proceed with robust standard errors, because that's good conservative practice.

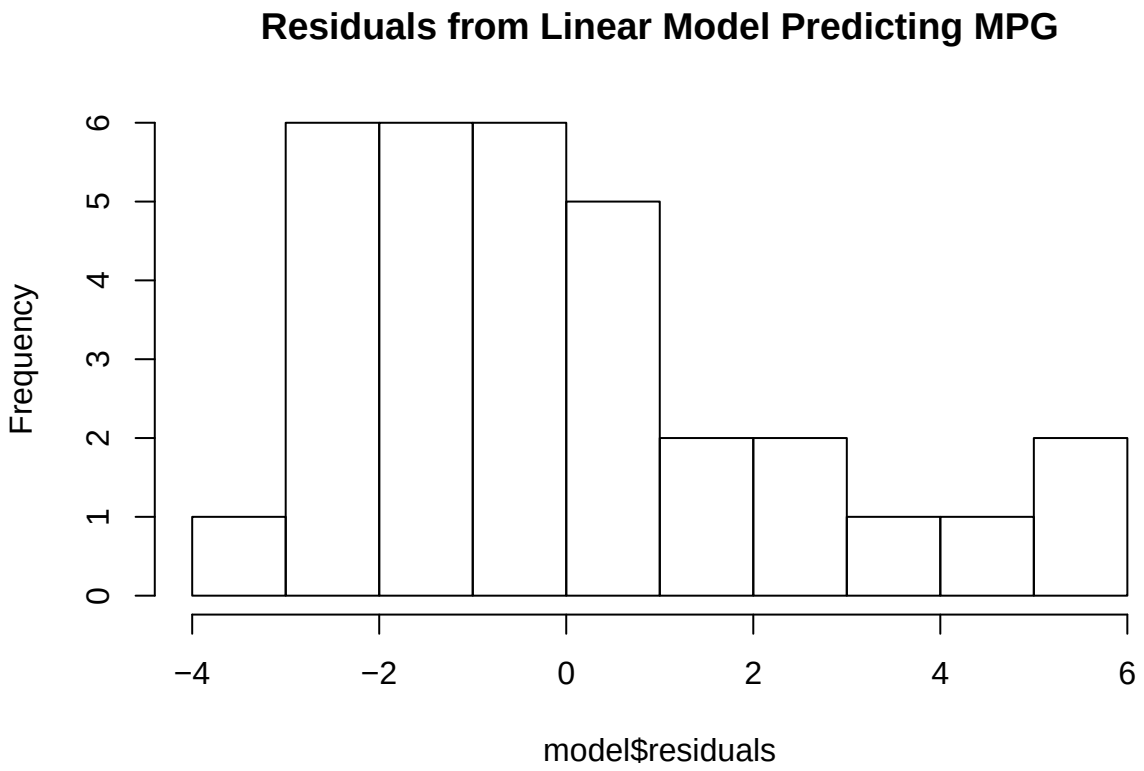
To check normality of errors, we can look at the qqplot that's part of R's standard diagnostics:

```
plot(model, which = 2)
```



We can also visually look at the residuals directly:

```
hist(model$residuals, breaks = 10, main = "Residuals from Linear Model Predicting MPG")
```



Both methods suggest we have a rightward skew. However, we have a large sample size, so the CLT tells us that our estimators will have a normal sampling distribution. Our look at the histogram confirms that we aren't in a situation with an extreme skew, so n=30 should be sufficient for the CLT.

Q1.3: In addition to the above, assess to what extent (imperfect) multicollinearity is affecting your inference.

```
vif(model)
```

```
##     disp       hp       wt     drat
## 8.209402 2.894373 5.096601 2.279547
```

```
vif(model) > 4
```

```
##  disp    hp    wt  drat
##  TRUE FALSE  TRUE FALSE
```

We note that displacement and weight are both closely predicted from the other variables. However, we would not take any automatic action here. Instead, we keep this in mind when we interpret our coefficients, and if these variables are insignificant, we may then consider changing our model.

Q1.4 Interpret your slope coefficients, and note which ones are significantly different from zero. Whether or not you detected heteroskedasticity above, be conservative in this step and use robust standard errors.

```
 coeftest(model, vcov = vcovHC)
```

```
##
## t test of coefficients:
##
##               Estimate Std. Error t value  Pr(>|t|)
## (Intercept) 29.1487376  7.0402909  4.1403 0.0003051 ***
```

```
## disp         0.0038152  0.0109347  0.3489 0.7298630
## hp          -0.0347835  0.0167887 -2.0718 0.0479579 *
## wt          -3.4796675  1.2644920 -2.7518 0.0104582 *
## drat         1.7680488  1.2465829  1.4183 0.1675383
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We notice that only hp and wt are statistically significant. In particular, every increase of 1 horsepower is associated with 0.034 less miles per gallon, holding weight, displacement, and rear axel ratio constant. From another perspective, it would take $1/0.034 = 29$ more horsepower to result in an efficiency reduction of 1 mpg.

Each 1000 lb increase in weight is associated with 3.47 less miles per gallon, holding other variables constant. This seems like a lot, but notice that most cars only weigh a few thousand pounds total:

```
summary(mtcars$wt)
```

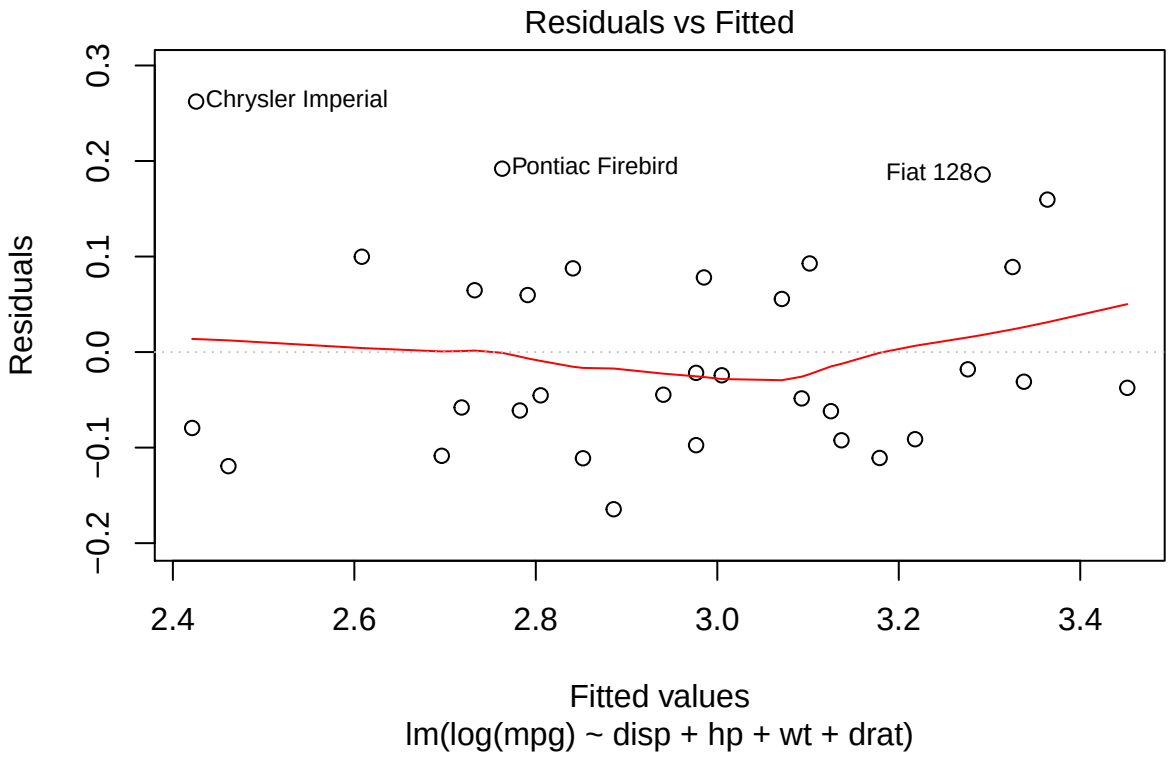
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.513   2.581   3.325   3.217   3.610   5.424
```

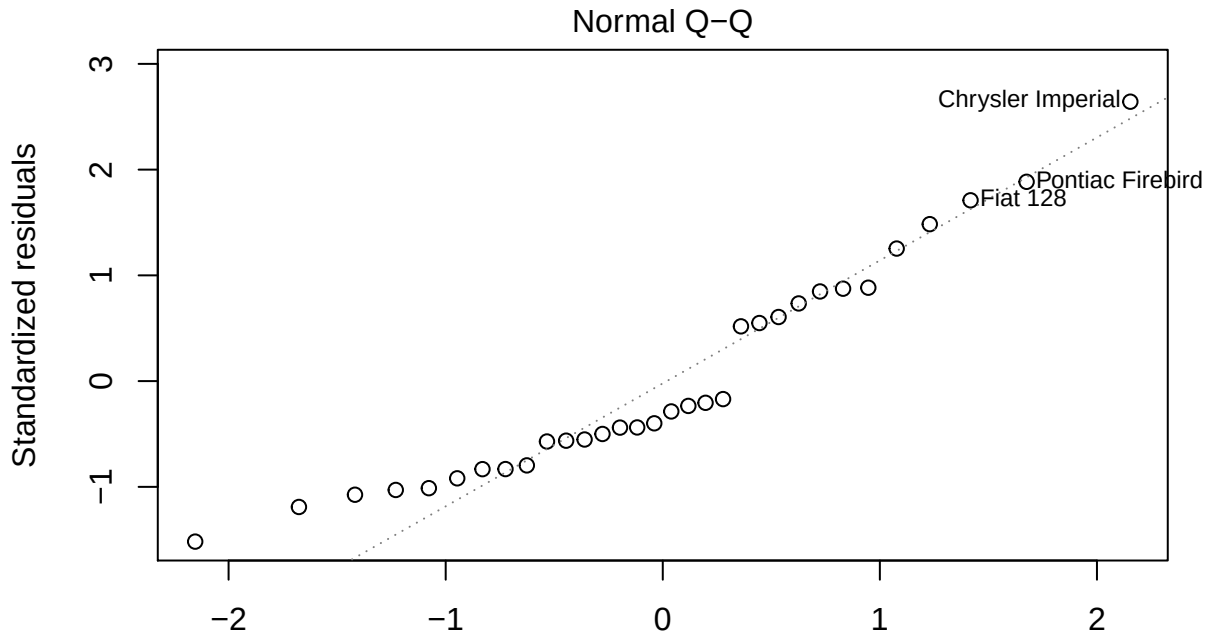
Alternate specification

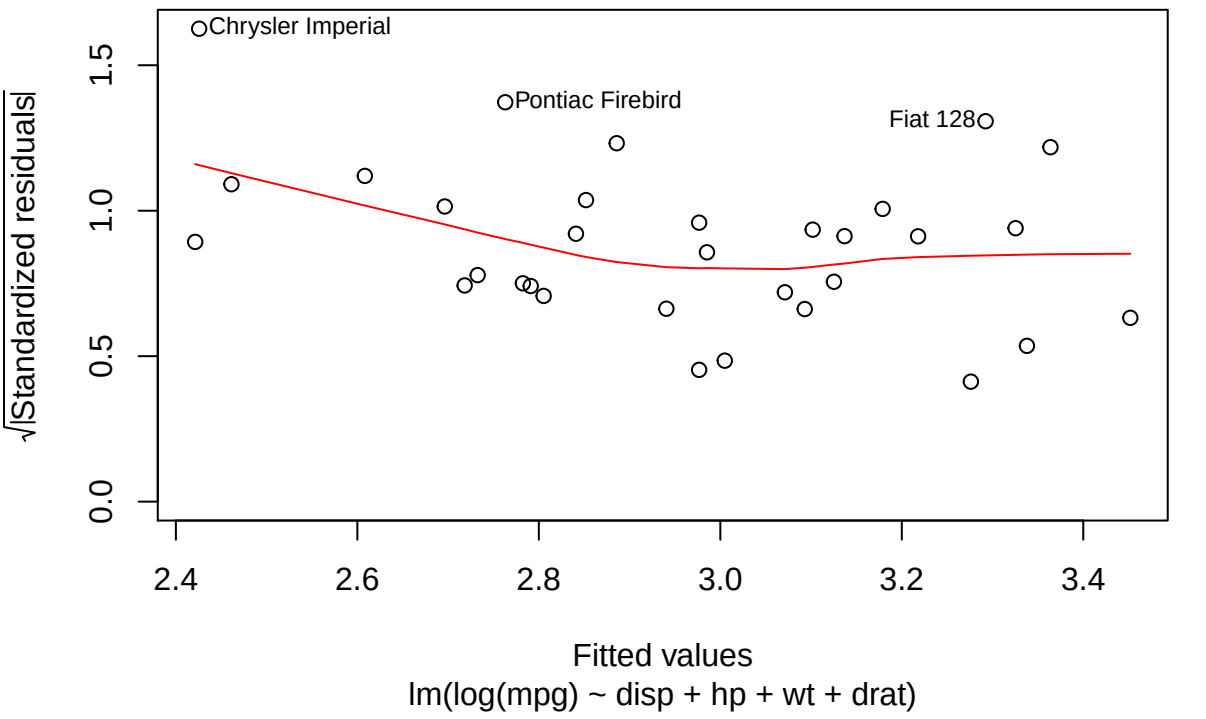
Next, alter your previous model by taking the log of mpg. Fit the model as before and examine your diagnostic plots.

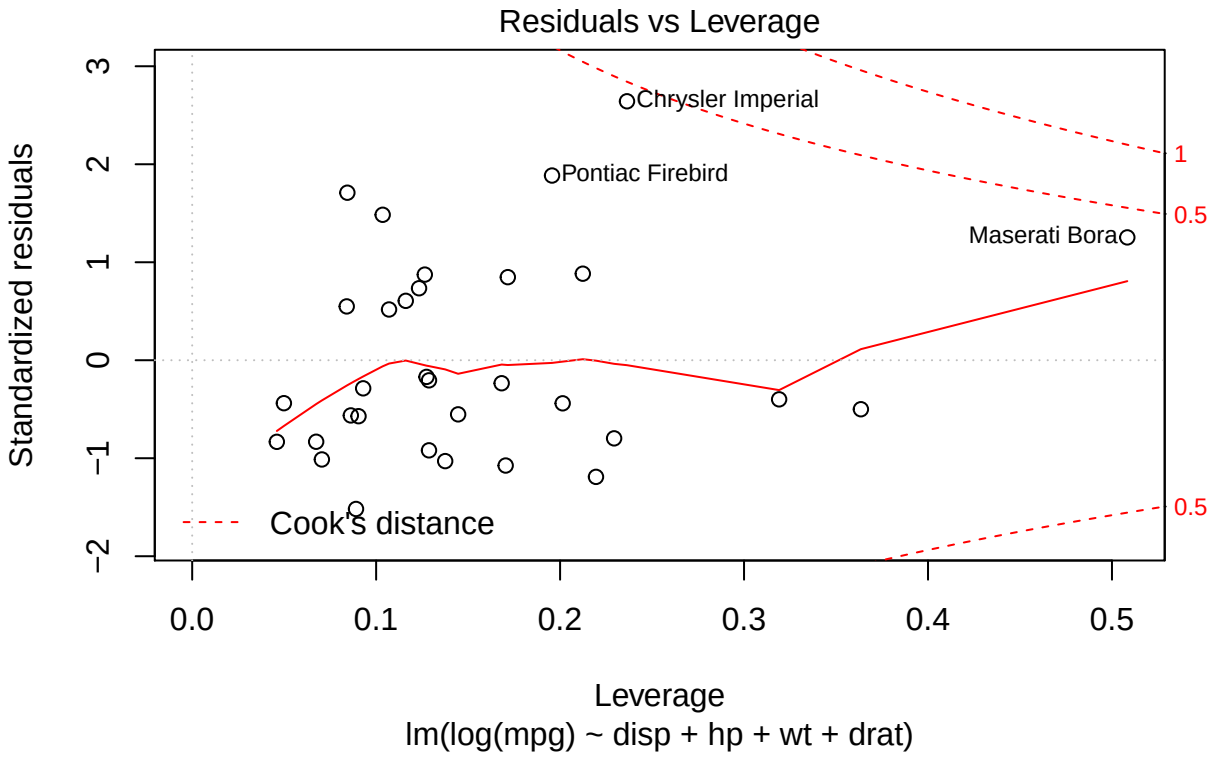
Q1.5 How does the log transform affect which CLM assumptions hold.

```
model2 <- lm(log(mpg)~disp+hp+wt+drat, data=mtcars)
plot(model2)
```




Normal Q−Q
Standardized residuals
Chrysler Imperial
Pontiac Firebird
Fiat 128
−2
−1
0
1
2
3
Theoretical Quantiles
lm(log(mpg) ~ disp + hp + wt + drat)


Scale−Location
√|Standardized residuals|
Chrysler Imperial
Pontiac Firebird
Fiat 128
0.0
0.5
1.0
1.5
2.4
2.6
2.8
3.0
3.2
3.4
Fitted values
lm(log(mpg) ~ disp + hp + wt + drat)



Q1.6 Which model has a better fit.

Q1.7 (As time allows) Report the results of both models in a nicely formatted regression table.

More about Multicollinearity

A common problem with multivariate regression is collinearity. If two or more predictor variables are highly correlated, and they are both entered into a regression model, it increases the standard error of each one and you get very unstable estimates of the slope. We usually assess the collinearity by variance inflation factor (VIF).

Ways to Detect Multicollinearity

We begin by regressing a particular independent variable on all other independent variables.

1. As the squared correlation (r2) increases toward 1.0, the magnitude of potential problems associated with multicollinearity increases correspondingly.
2. Tolerance (1-R2) One minus the squared multiple correlation of a given IV from other Ivs in the equation. Tolerance values of 0.10 or less Indicate that there may be serious multicollinearity.
3. The Variance Inflation Factor [VIF=1/(1-R2)] VIF Is the reciprocal of the Tolerance. Any VIF of 10 or more provides evidence of serious multicollinearity.
4. Condition Number (k) The square root of the ratio of the largest eigenvalue to the smallest eigenvalue. k of 30 or larger indicate that there may be serious multicollinearity.