Week 6 Regression Report Sample

For this sample report, I'll use a portion of `R`'s built-in data set `mtcars`. I created a data file with five of the variables from that set for the purposes of this sample report.

Question

Can we predict the mpg of a car from its engine displacement, horsepower, weight, and number of gears?

Understanding the Data

We load the data set.

```r
library(tidyverse)
```

```r
myData <- read_csv("carmpg.csv")
```

```
Parsed with column specification:
cols(
  mpg = col_double(),
  disp = col_double(),
  hp = col_double(),
  wt = col_double(),
  gear = col_double()
)
```

```r
attach(myData)
```

```
The following object is masked from package:ggplot2:

    mpg
```

```r
head(myData)
```

```
# A tibble: 6 x 5
    mpg  disp    hp    wt  gear
  <dbl> <dbl> <dbl> <dbl> <dbl>
1  21     160   110  2.62     4
2  21     160   110  2.88     4
3  22.8   108    93  2.32     4
4  21.4   258   110  3.22     3
5  18.7   360   175  3.44     3
6  18.1   225   105  3.46     3
```

```r
summary(myData)
```

```
      mpg             disp             hp              wt       
 Min.   :10.40   Min.   : 71.1   Min.   : 52.0   Min.   :1.513  
 1st Qu.:15.43   1st Qu.:120.8   1st Qu.: 96.5   1st Qu.:2.581  
 Median :19.20   Median :196.3   Median :123.0   Median :3.325  
 Mean   :20.09   Mean   :230.7   Mean   :146.7   Mean   :3.217  
 3rd Qu.:22.80   3rd Qu.:326.0   3rd Qu.:180.0   3rd Qu.:3.610  
 Max.   :33.90   Max.   :472.0   Max.   :335.0   Max.   :5.424  
      gear      
```

```
 Min.   :3.000  
 1st Qu.:3.000  
 Median :4.000  
 Mean   :3.688  
 3rd Qu.:4.000  
 Max.   :5.000  
```

The variable `gear` only has three values which seems to suggest that we should treat it as a categorical variable, not numerical. We can tell `R` to do this by specifying that `gear` is a factor. This will tell `R` to create dummy variables when making the model.

```
new_gear <- factor(gear)
```

We check the correlation between the other variables.

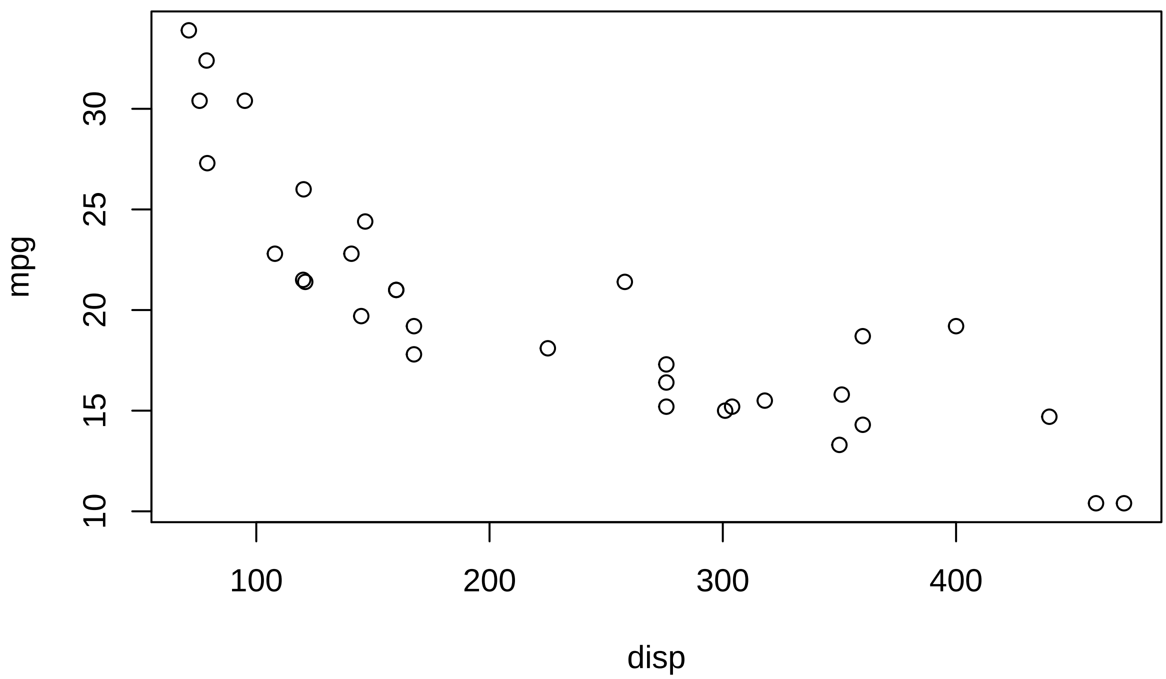
```
cor( data.frame(mpg, disp, hp, wt) )
```

```
            mpg       disp         hp         wt
mpg   1.0000000 -0.8475514 -0.7761684 -0.8676594
disp -0.8475514  1.0000000  0.7909486  0.8879799
hp   -0.7761684  0.7909486  1.0000000  0.6587479
wt   -0.8676594  0.8879799  0.6587479  1.0000000
```

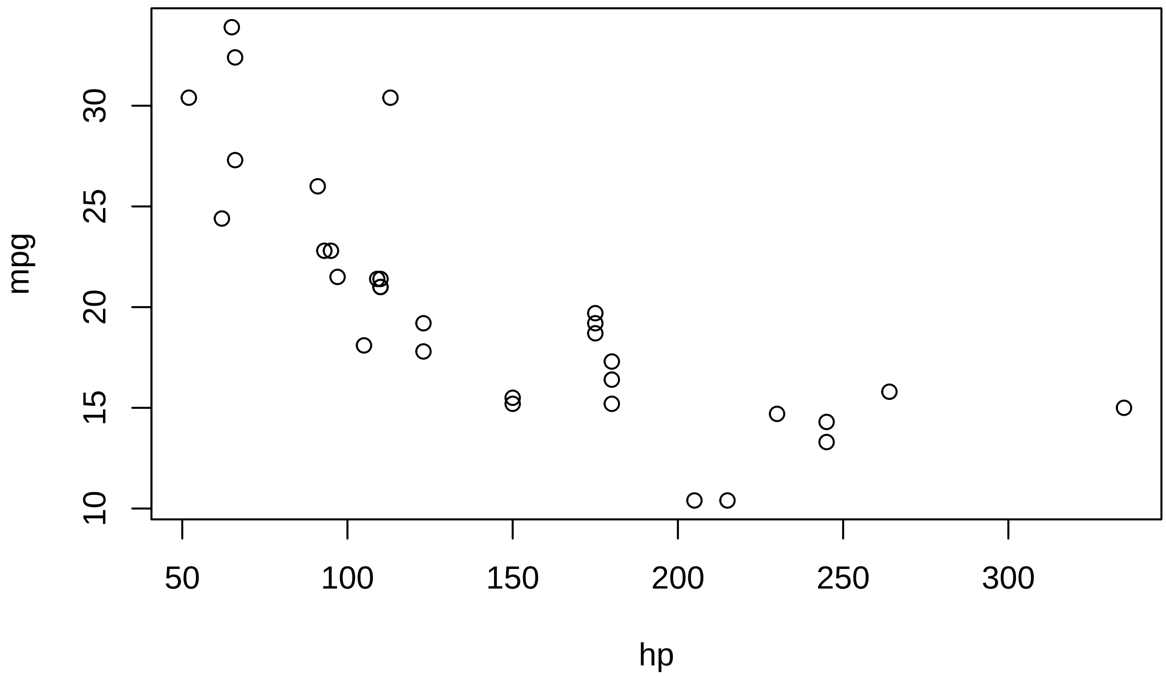
These all seem to have fairly stron linear relationships to `mpg`.

We will the scatterplots now.

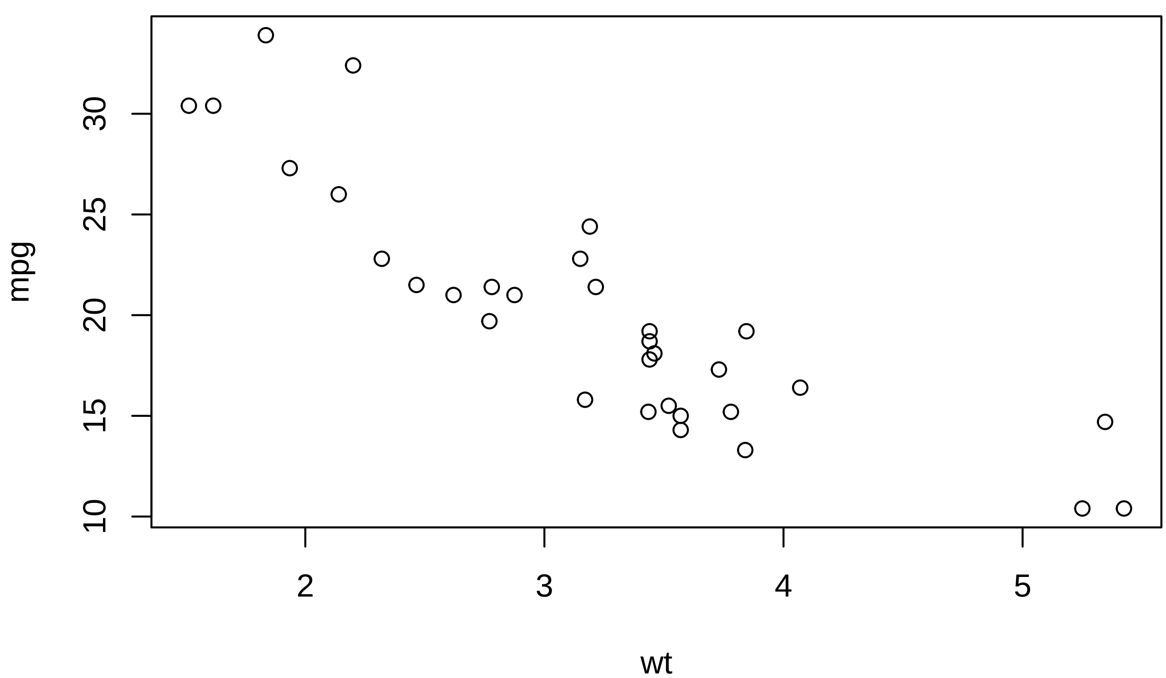
```
plot(mpg ~ disp)
```




```
plot(mpg ~ hp)
```

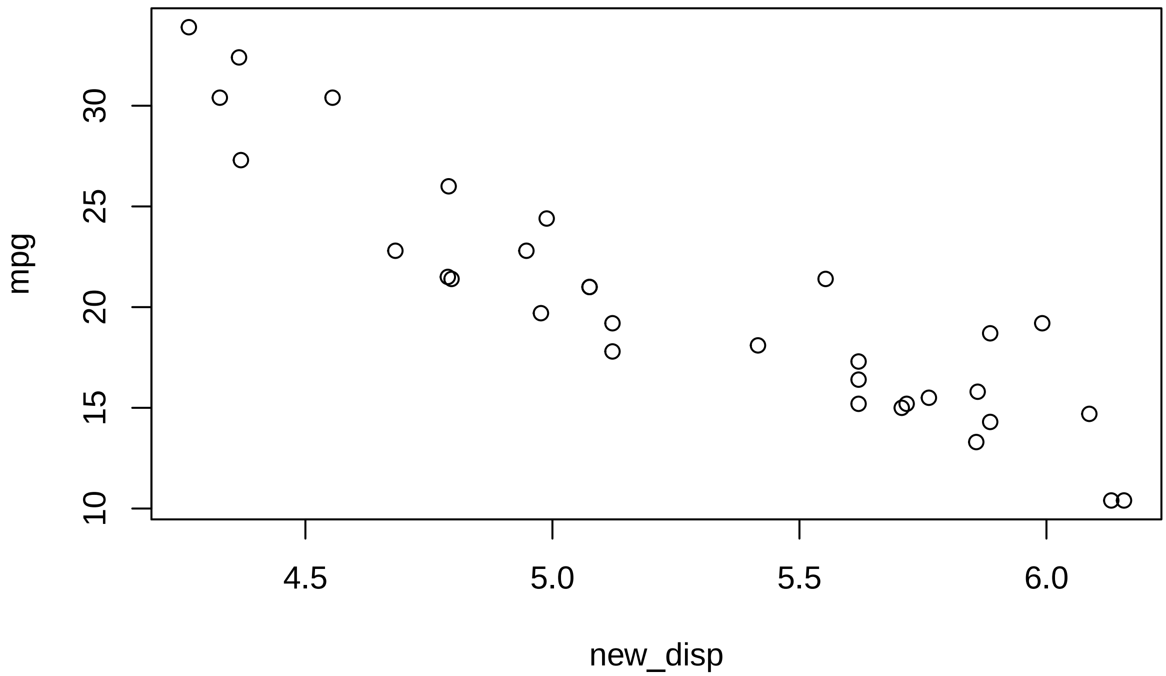



```
plot(mpg ~ wt)
```

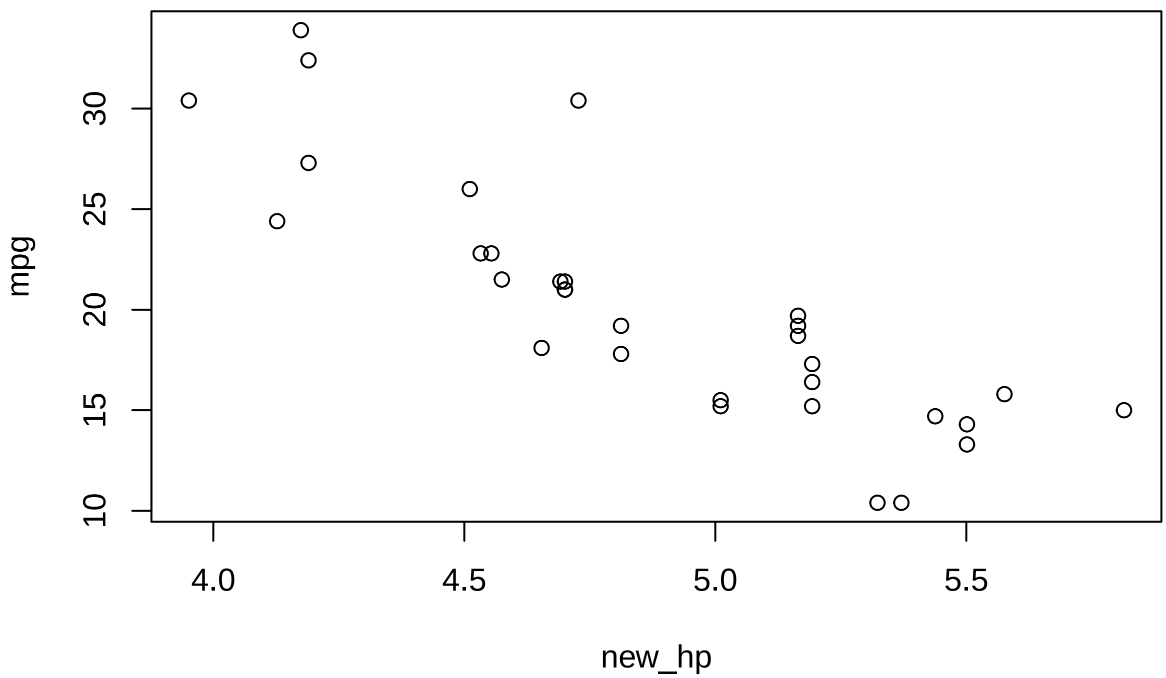


Displacement and horsepower both show some non-linearity. We will transform both and then attempt to build the model.

```
new_disp <- log(disp)
new_hp <- log(hp)

plot(mpg ~ new_disp)
```




```r
plot(mpg ~ new_hp)
```



These scatterplots show a much more linear relationship. We check the correlation coefficients to verify:

```r
cor( data.frame(mpg, new_disp, new_hp, wt) )
```

```
                mpg   new_disp     new_hp         wt
mpg       1.0000000 -0.9071119 -0.8487707 -0.8676594
new_disp -0.9071119  1.0000000  0.8617723  0.8845389
new_hp   -0.8487707  0.8617723  1.0000000  0.7158277
wt       -0.8676594  0.8845389  0.7158277  1.0000000
```

The correlation coefficients for `mpg` with both `new_disp` and `new_hp` have increased.

Building the Model

Finally we are ready to generate our first model. We will use all variables initially then use backward elimnation to remove unnecessary variables.

```
mpgModel <- lm( mpg ~ new_disp + new_hp + wt + new_gear)
summary(mpgModel)
```

```

Call:
lm(formula = mpg ~ new_disp + new_hp + wt + new_gear)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.2294 -1.1997 -0.4561  0.6932  4.8744 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)   70.080      8.964   7.818 2.71e-08 ***
new_disp      -3.137      2.569  -1.221   0.2329    
new_hp        -5.703      2.255  -2.529   0.0178 *  
wt            -1.720      1.032  -1.666   0.1077    
new_gear4     -0.715      1.461  -0.489   0.6287    
new_gear5      1.491      1.734   0.860   0.3976    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.313 on 26 degrees of freedom
Multiple R-squared:  0.8764,	Adjusted R-squared:  0.8527 
F-statistic: 36.88 on 5 and 26 DF,  p-value: 5.28e-11
```

The variable `new_gear` is the least significant, so we remove it first.

```
mpgModel <- lm( mpg ~ new_disp + new_hp + wt)
summary(mpgModel)
```

```

Call:
lm(formula = mpg ~ new_disp + new_hp + wt)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.0949 -1.4954 -0.3474  0.7356  4.6082 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  63.8126     5.6977  11.200 7.45e-12 ***
new_disp     -3.0519     2.1093  -1.447   0.1590    
new_hp       -4.1506     1.7443  -2.380   0.0244 *  
wt           -2.2784     0.9213  -2.473   0.0197 *  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 2.296 on 28 degrees of freedom
Multiple R-squared:  0.8689,	Adjusted R-squared:  0.8549 
F-statistic: 61.88 on 3 and 28 DF,  p-value: 1.791e-12
```

The transformed displacement does not seem significant. We remove it.

```
mpgModel <- lm( mpg ~ new_hp + wt)
summary(mpgModel)
```

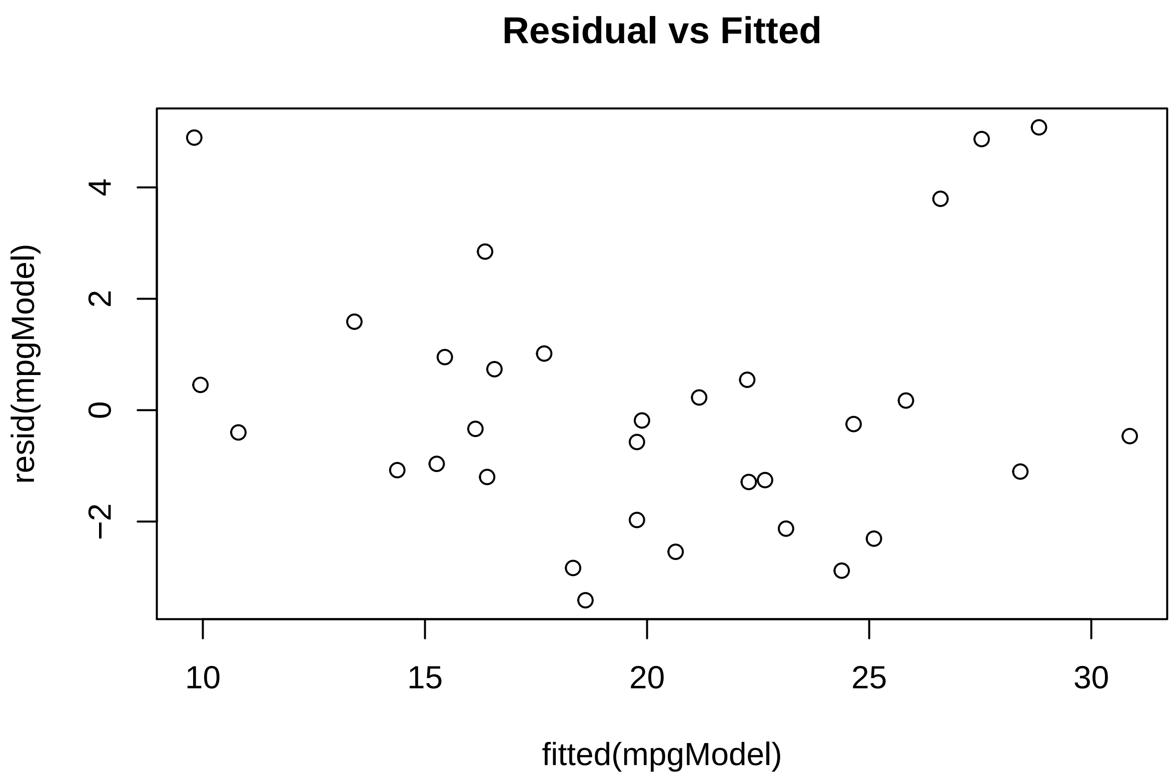
```
Call:
lm(formula = mpg ~ new_hp + wt)

Residuals:
    Min      1Q  Median      3Q     Max 
-3.4130 -1.2642 -0.3679  0.7902  5.0780 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  59.5709     4.9769  11.970 9.64e-13 ***
new_hp       -5.9218     1.2658  -4.678 6.20e-05 ***
wt           -3.2856     0.6148  -5.344 9.74e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 2.339 on 29 degrees of freedom
Multiple R-squared:  0.8591,	Adjusted R-squared:  0.8494 
F-statistic: 88.44 on 2 and 29 DF,  p-value: 4.542e-13
```

Both of the remaingin variables and the model overall are significant.

Model Assumptions

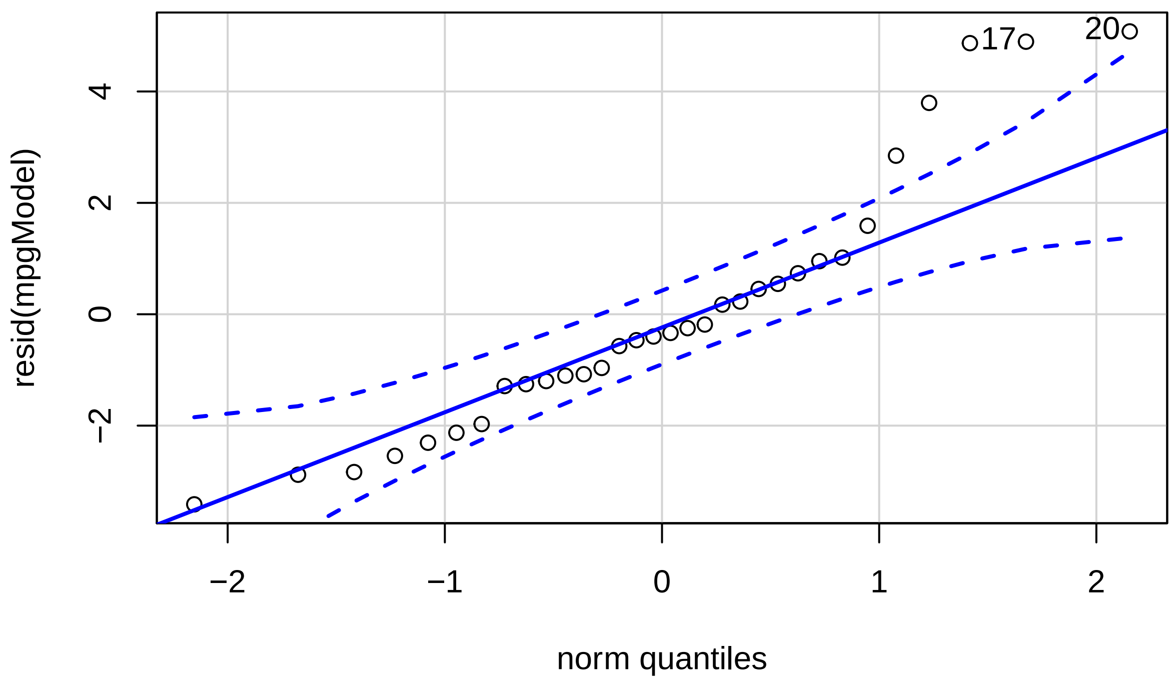
We will check model assumptions to see if any more transformations are necessary.

```
plot(resid(mpgModel) ~ fitted(mpgModel), main="Residual vs Fitted")
```



This is not terrible, but not perfect either. There seems to be some nonlinearity, but the variance seems roughly equal.

```
library(car) # this library lets me make the nice plot with dashed lines
qqPlot(resid(mpgModel))
```




```
[1] 20 17
```

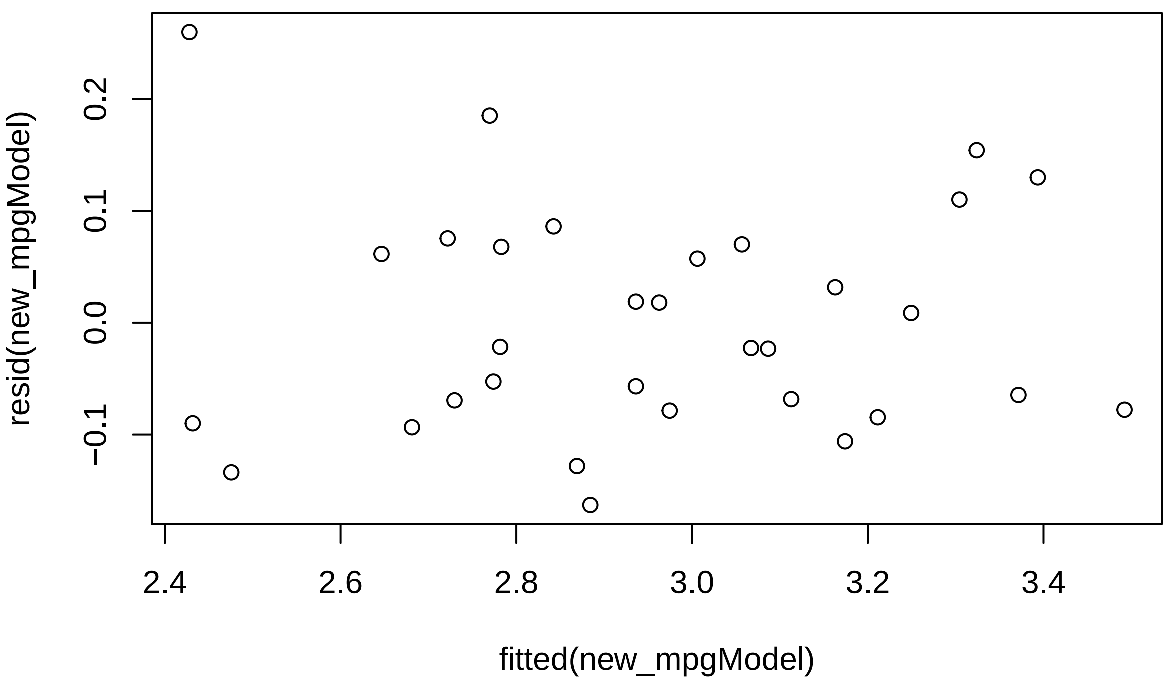
There are a several points on the large end that leave the dashed lines. This combined with the residual plot above indicates a data transformation would be helpful. We will perform a `log` transformation on the response variable.

```
newMpg <- log(mpg)
new_mpgModel <- lm(newMpg ~ new_hp + wt)
summary(new_mpgModel)
```

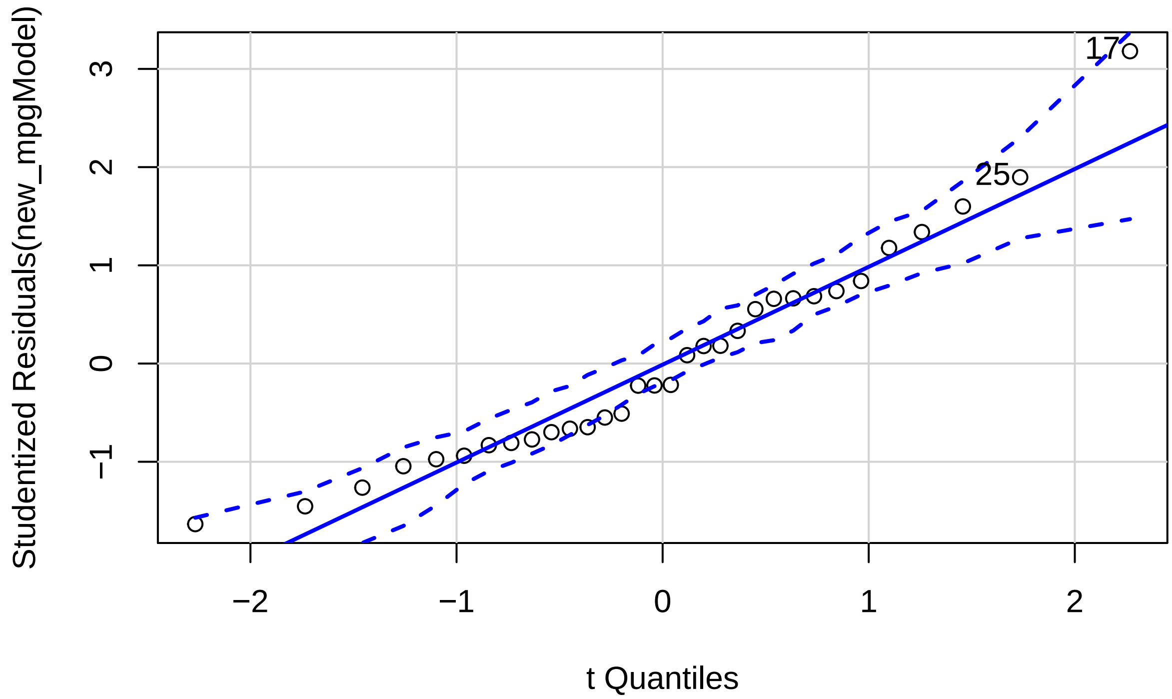
```
Call:
lm(formula = newMpg ~ new_hp + wt)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.16296 -0.07799 -0.02210  0.06837  0.25985 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.83167    0.22198  21.766  < 2e-16 ***
new_hp      -0.26566    0.05646  -4.706 5.75e-05 ***
wt          -0.17942    0.02742  -6.543 3.63e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1043 on 29 degrees of freedom
Multiple R-squared:  0.8852,    Adjusted R-squared:  0.8773 
F-statistic: 111.8 on 2 and 29 DF,  p-value: 2.338e-14
```

So far the model looks better with a slightly larger adjusted R-squared and even smaller p-values for several of the variables. Let's check the model assumptions with some residual plots.

```
plot(resid(new_mpgModel) ~ fitted(new_mpgModel))
```




```
qqPlot(new_mpgModel)
```




```
[1] 17 25
```

These both seem improved. There is no more non-linearity in the residual plots. The normal-probability plot has most of the points very close to the straight line. This seems to be the best model we can construct from this dataset.

Conclusion

Overall the model meets all assumptions and the R-squared value is rather high at 88%. We should be confident in using this model to predict mpg for cars based on their weight and horsepower.