Lab 4 Part 1

Isabel Sassoon

This lab will cover the following topics:

- Linear regression
- Interpreting the outputs and diagnostics
- transforming data and applying lm

Correlation

To demonstrate how to compute correlation in R we will make use of the cars data set that is available in base R.

```r
head(cars)
```

```
##   speed dist
## 1     4    2
## 2     4   10
## 3     7    4
## 4     7   22
## 5     8   16
## 6     9   10
```

As you have seen in the lecture this has two variables: speed of the car, and the distance it took to come to a halt.

How to compute correlation

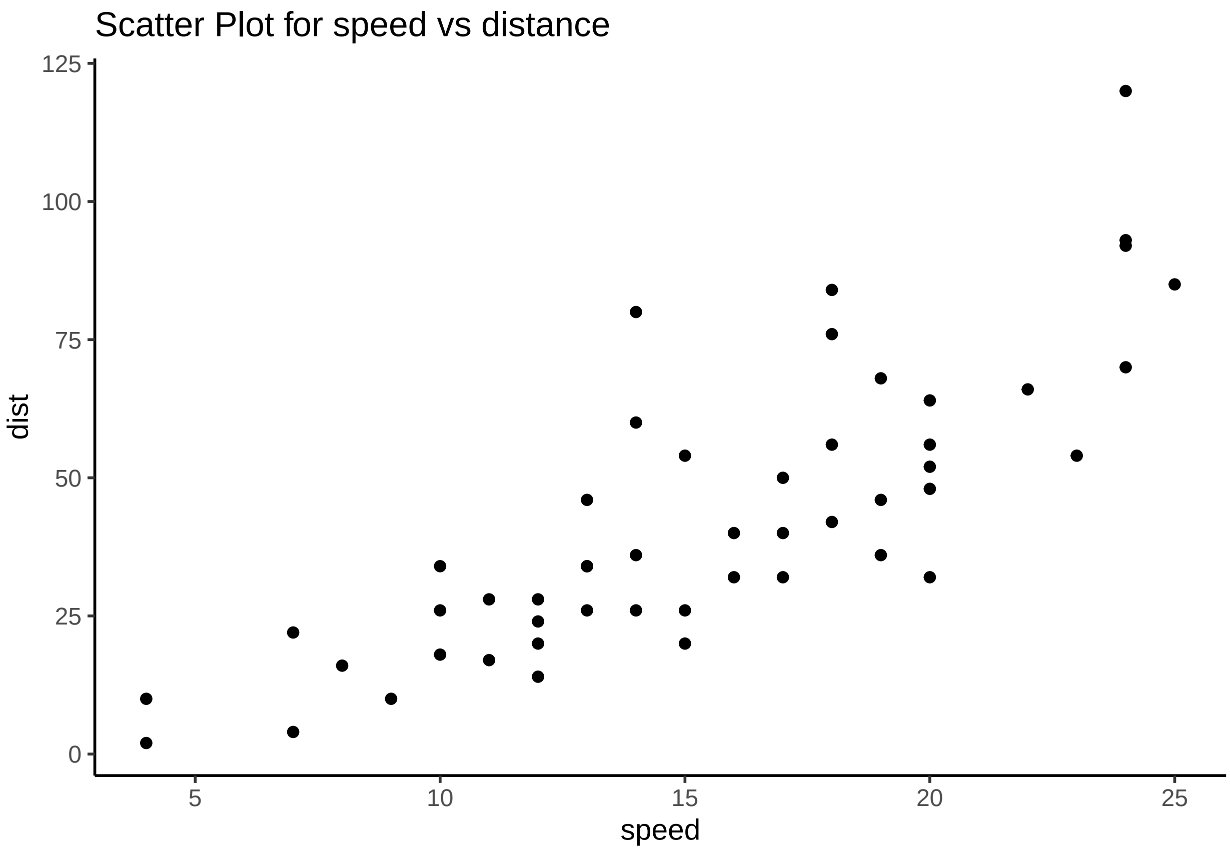
```r
cor(cars$speed, cars$dist)
```

```
## [1] 0.8068949
```

In order to visualize this data

```r
library(ggplot2)

ggplot(data=cars, aes(x=speed, y=dist)) + geom_point() + theme_classic() +
  ggtitle("Scatter Plot for speed vs distance")
```



It seems like this data is correlated. Lets test the correlation ratio. This will test an $H_0 : r = 0$ vs $H_1 : r \neq 0$

```
cor.test(cars$speed, cars$dist)
```

```
##
##  Pearson's product-moment correlation
##
## data:  cars$speed and cars$dist
## t = 9.464, df = 48, p-value = 1.49e-12
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6816422 0.8862036
## sample estimates:
##       cor
## 0.8068949
```

This results in a very small p-value, we reject the Null Hypothesis. These two variables are correlated.

Linear Regression

We may now want to model the relationship between the stopping distance and the speed. As the speed is what you can control it makes sense that it is the explanatory variable.

```
cars.lm<-lm(cars$dist~cars$speed, data=cars)
cars.lm
```

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

```
## lm(formula = cars$dist ~ cars$speed, data = cars)
##
## Coefficients:
## (Intercept)   cars$speed
##     -17.579        3.932
```

This output gives us the minimum information: the intercept and the coefficient. From this we can see that:

distance $= -17.6 + 3.9\times$ speed.

To get more diagnostics we can use

```r
summary(cars.lm)
```

```
##
## Call:
## lm(formula = cars$dist ~ cars$speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -29.069  -9.525  -2.272   9.215  43.201
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.5791     6.7584  -2.601   0.0123 *
## cars$speed    3.9324     0.4155   9.464 1.49e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.38 on 48 degrees of freedom
## Multiple R-squared:  0.6511, Adjusted R-squared:  0.6438
## F-statistic: 89.57 on 1 and 48 DF,  p-value: 1.49e-12
```

From the output above we learn that: - There is a significant difference in the variances - We also learn that 0.65 of the variance is explained by the model - Both coefficients are significant.

The relationship between the speed and the breaking distance is such that for every increase in speed of 1 - the breaking distance increases by 3.9324.

OPTIONAL - it is also possible to have an ANOVA plot

```r
anova(cars.lm)
```

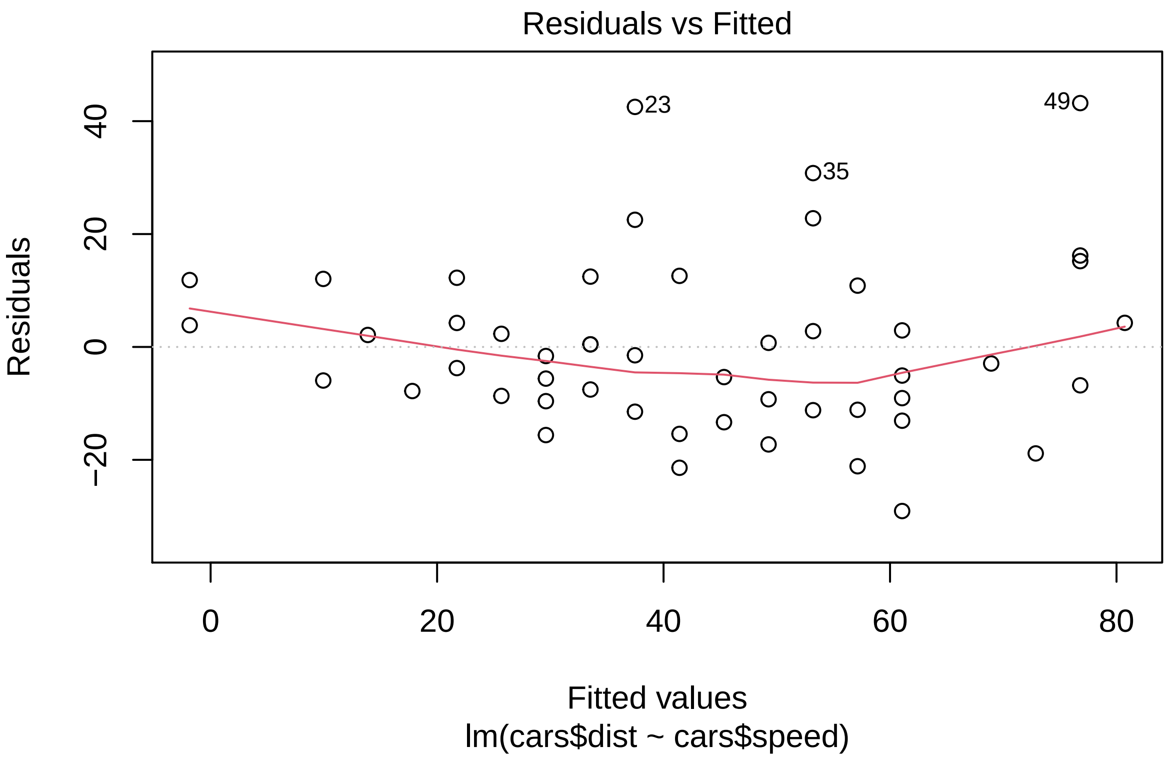
```
## Analysis of Variance Table
##
## Response: cars$dist
##            Df Sum Sq Mean Sq F value   Pr(>F)
## cars$speed  1  21186 21185.5  89.567 1.49e-12 ***
## Residuals  48  11354   236.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

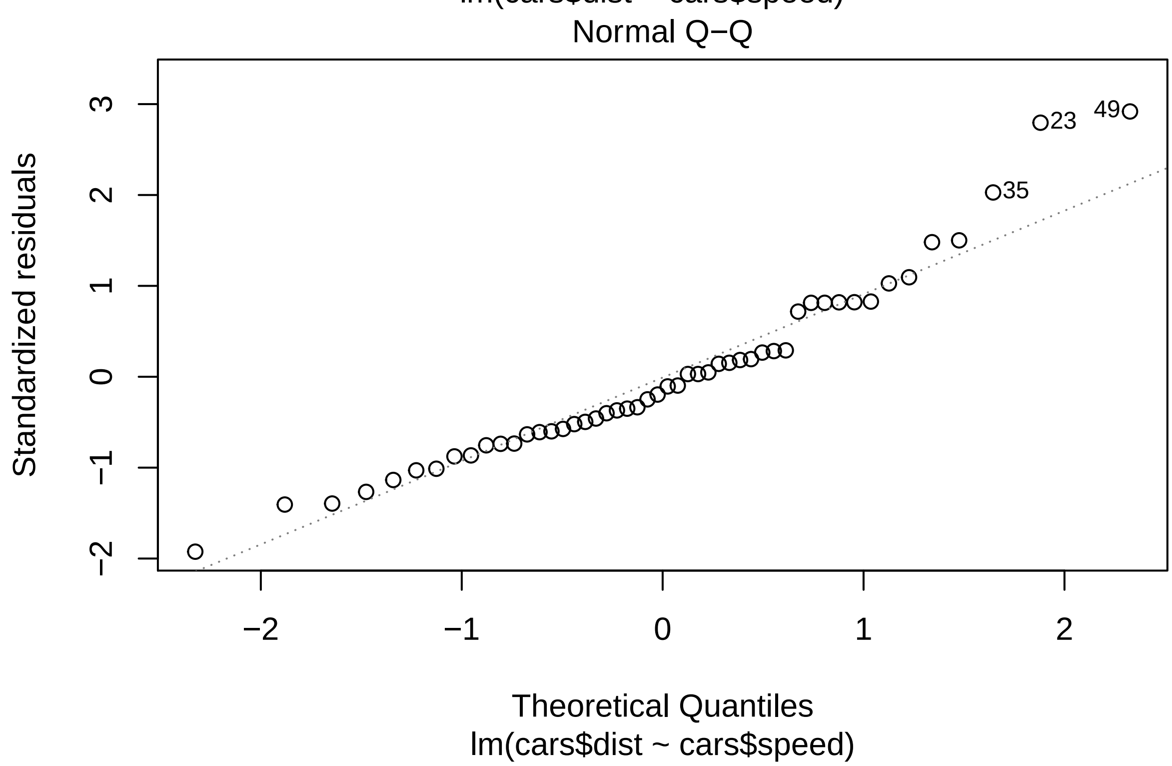
This gives us much more detail on how the model is doing.

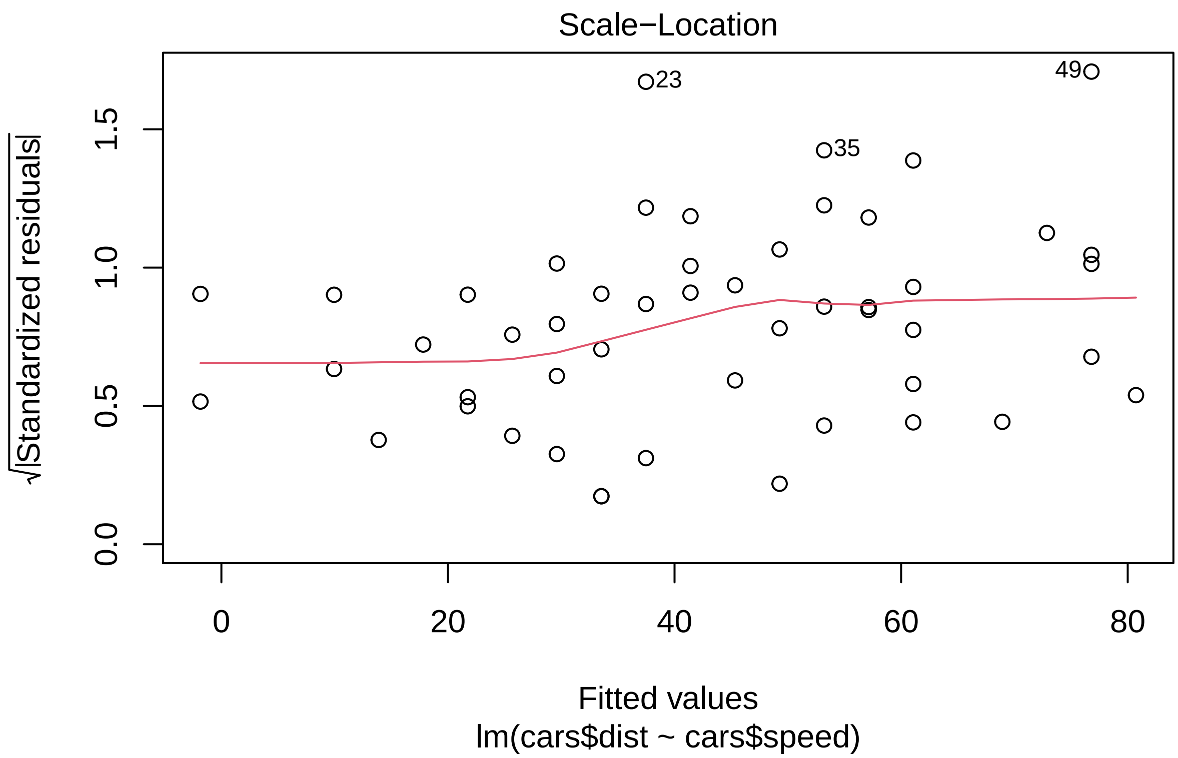
Diagnsotic plots

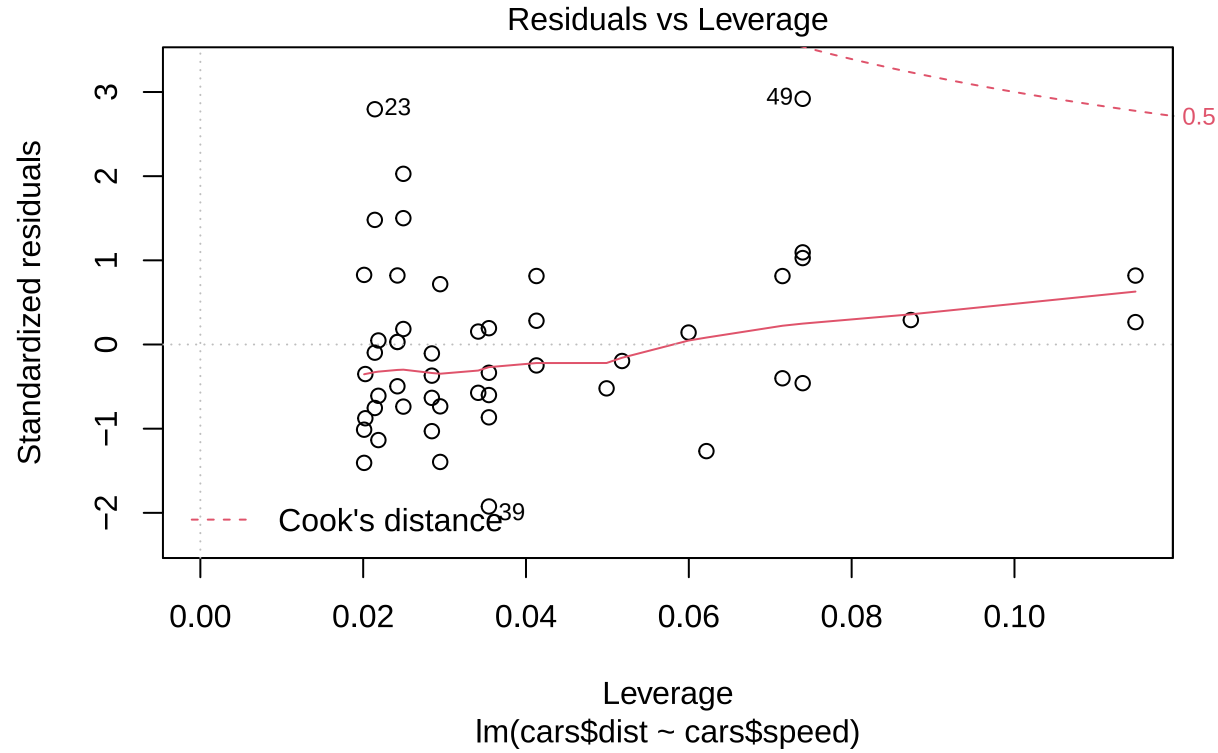
It is important to inspect the plots too:

```r
plot(cars.lm)
```









The residuals are acceptable but we can consider modeling this with the squared speed too.

```
cars.lm2<-lm(cars$dist~cars$speed+ I(cars$speed^2))
```

How does our new model look?

```r
cars.lm2
```

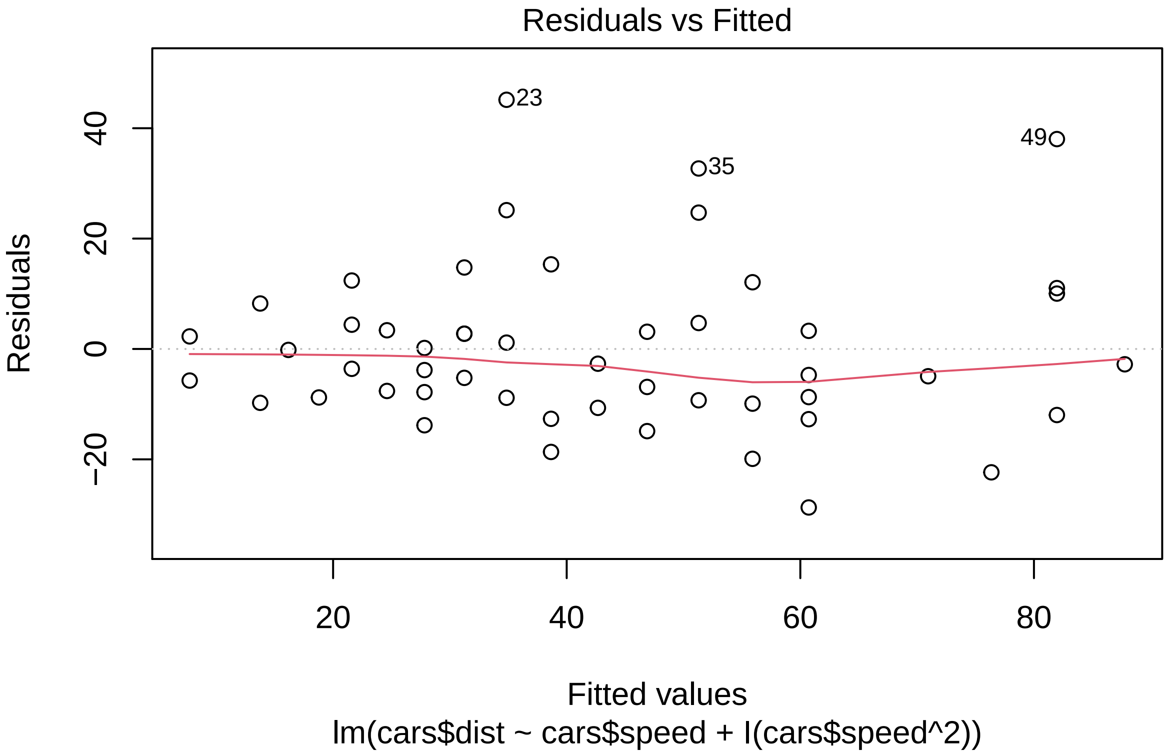
```
##
## Call:
## lm(formula = cars$dist ~ cars$speed + I(cars$speed^2))
##
## Coefficients:
##     (Intercept)       cars$speed  I(cars$speed^2)
##         2.47014          0.91329          0.09996
```

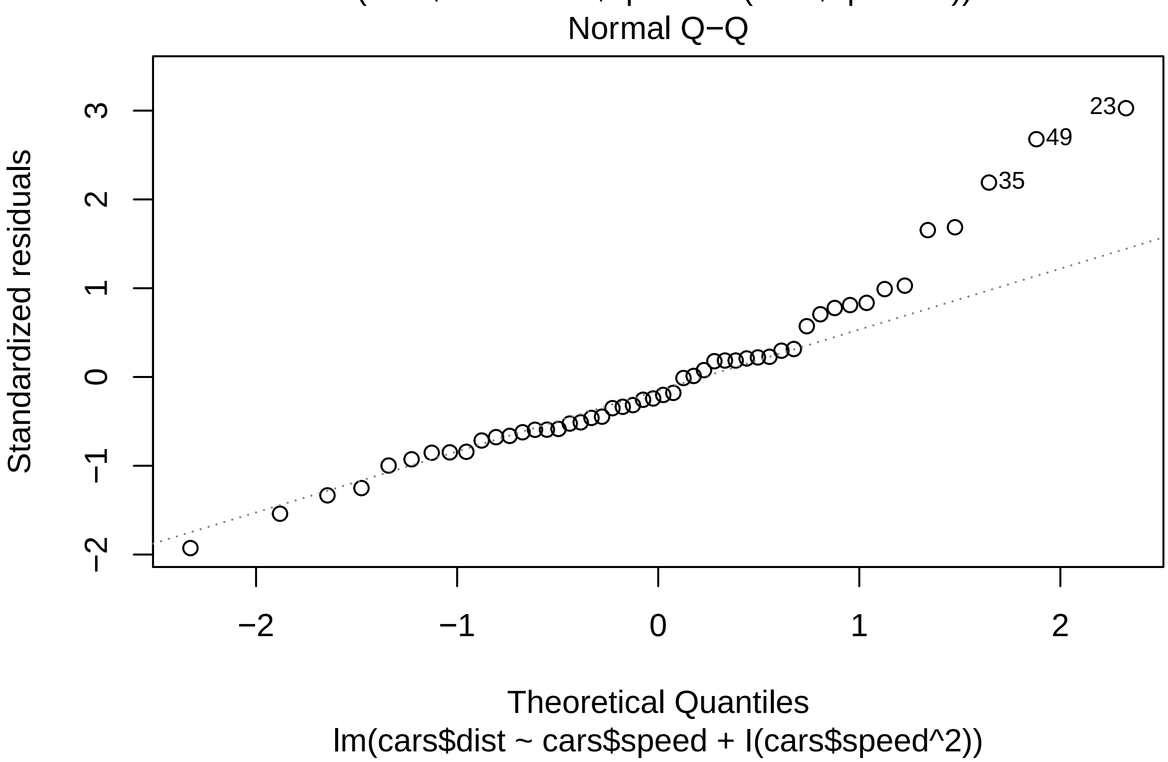
```r
summary(cars.lm2)
```

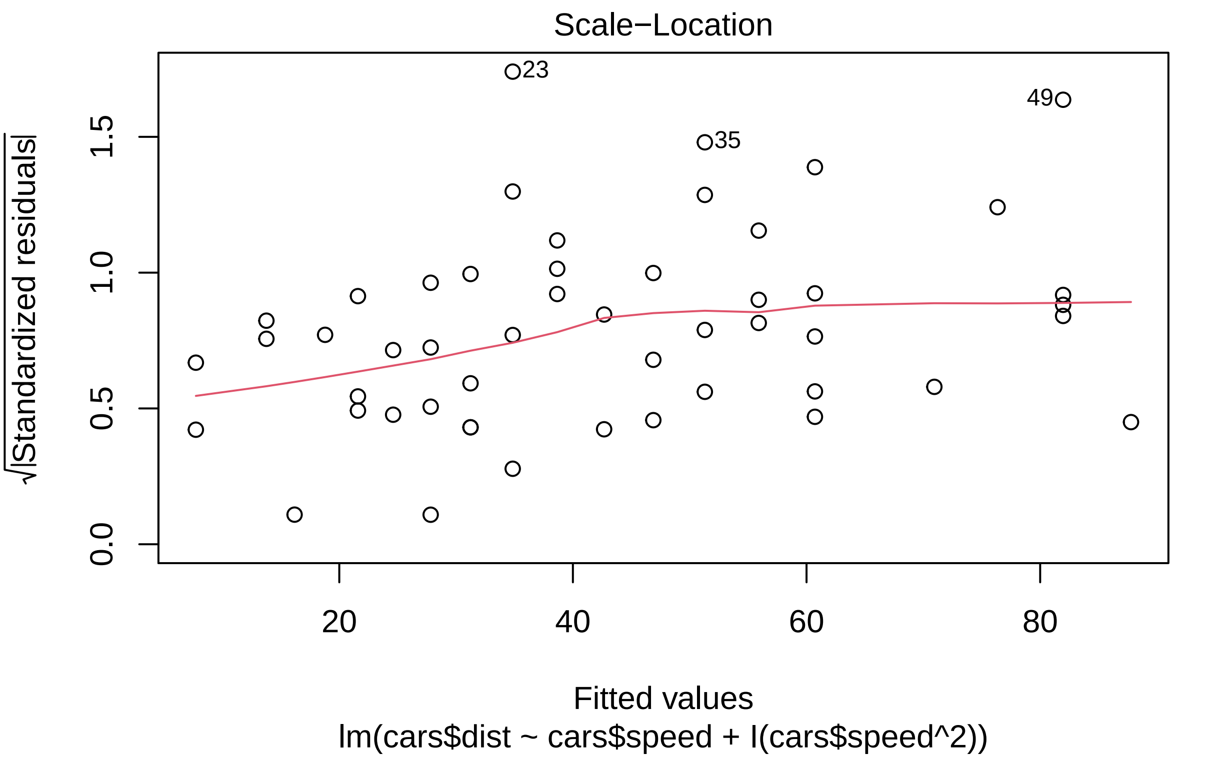
```
##
## Call:
## lm(formula = cars$dist ~ cars$speed + I(cars$speed^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -28.720  -9.184  -3.188   4.628  45.152
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)      2.47014   14.81716   0.167    0.868
## cars$speed       0.91329    2.03422   0.449    0.656
## I(cars$speed^2)  0.09996    0.06597   1.515    0.136
##
## Residual standard error: 15.18 on 47 degrees of freedom
## Multiple R-squared:  0.6673, Adjusted R-squared:  0.6532
## F-statistic: 47.14 on 2 and 47 DF,  p-value: 5.852e-12
```

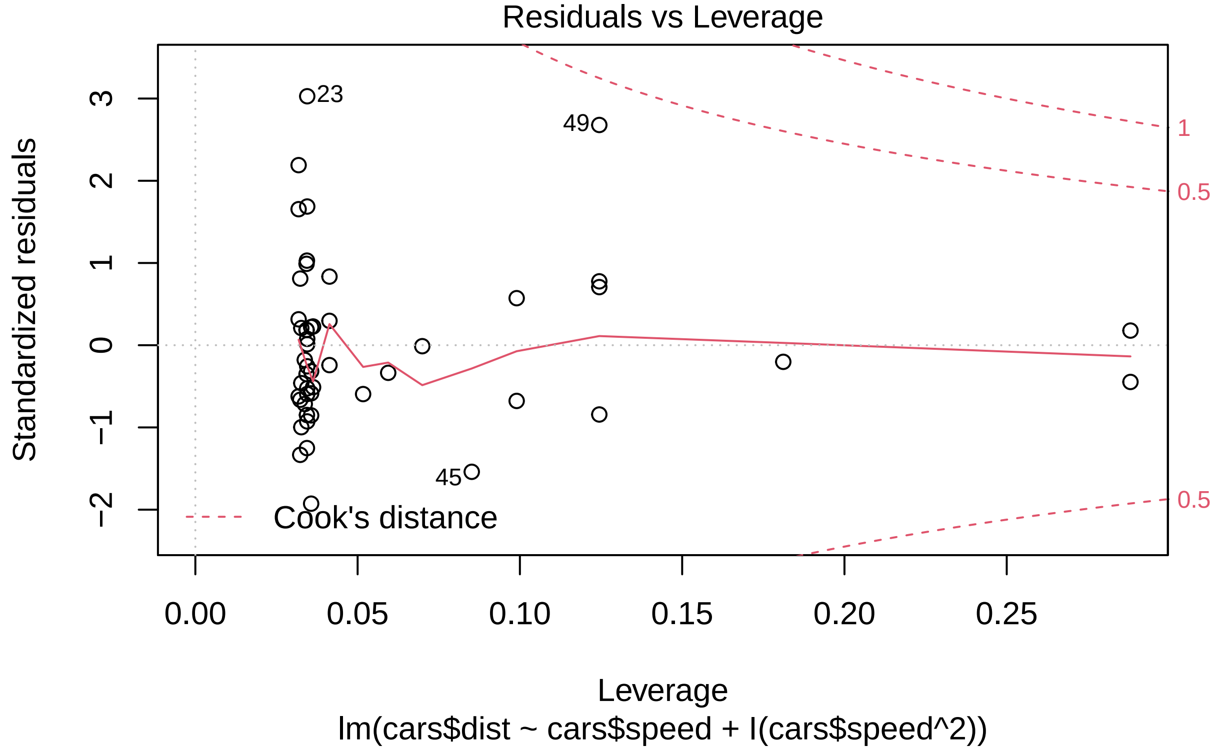
and we should also look at the plots

```r
plot(cars.lm2)
```


Residuals vs Fitted
Residuals
Fitted values
lm(cars$dist ~ cars$speed + I(cars$speed^2))
40
20
0
−20
20
40
60
80
23
35
49


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(cars$dist ~ cars$speed + I(cars$speed^2))
3
2
1
0
−1
−2
−2
−1
0
1
2
23
49
35





The residuals for this model are fine, and dont point to any violations of the assumptions. The R^2 value is slightly higher for this latter model, but it is a more complex model.

estimating distance for a new value of speed

Either model can be used to get an estimate for a distance.

What is the distance when the speed is 21?

If we use the first simpler model:

distance = -17.6+ 3.9 * speed

```
dist.21<--17.6+ 3.9* 21
dist.21
```

```
## [1] 64.3
```

For the more complex model 2.47014 0.91329 0.09996

```
dist2.21<-2.47+0.913*21+0.0996*21*21
dist2.21
```

```
## [1] 65.5666
```

The models give similar yet differnt estimates for the distance.

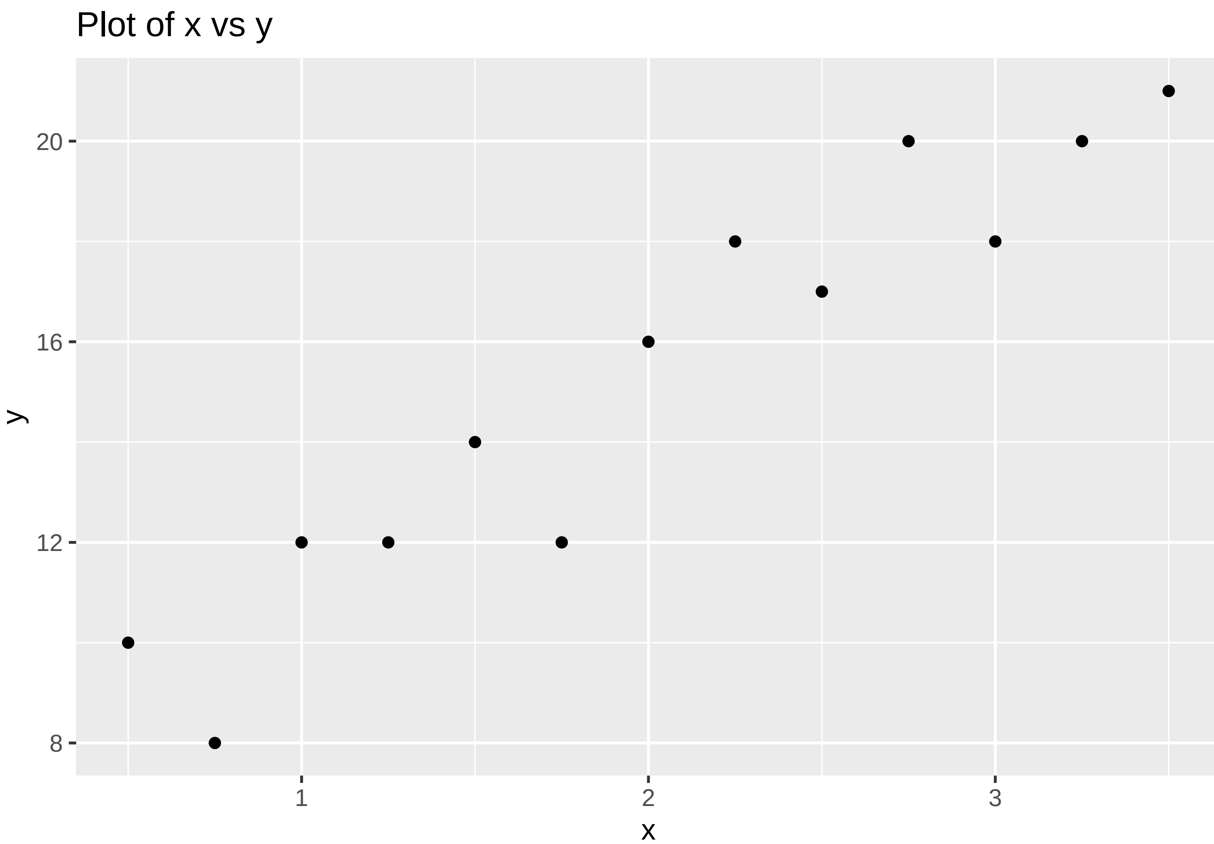
Linear Regression

In another example for Linear Regression A drug is developed to reduce pulse rate. The independent variable x is the dosage of the drug in mg; the dependent variable y is the reduction in pulse rate in beats per minute.

```
x<-c(0.50,0.75 ,1.00,1.25,1.50,1.75,2.00,2.25,2.50,2.75,3.00,3.25,3.50)
y<-c(10,8,12,12,14,12,16,18,17,20,18,20,21)
pulse<-as.data.frame(cbind(x,y))
```

Plot the data

```
ggplot(pulse, aes(x=x, y=y)) + geom_point() + ggtitle("Plot of x vs y")
```



Run a linear regression

```
lm(pulse$y~pulse$x, data =pulse)
```

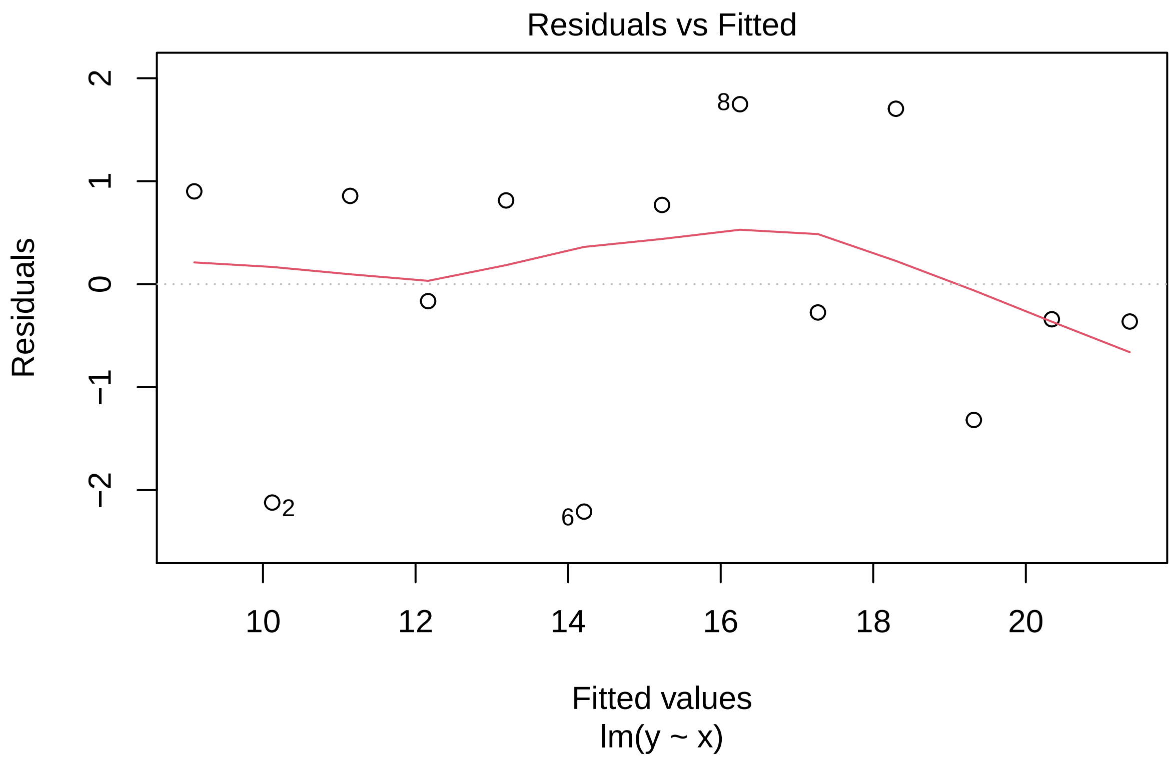
```
##
## Call:
## lm(formula = pulse$y ~ pulse$x, data = pulse)
##
## Coefficients:
## (Intercept)      pulse$x
##       7.055        4.088
```

From this we can see tjat the relationship between x and y is:

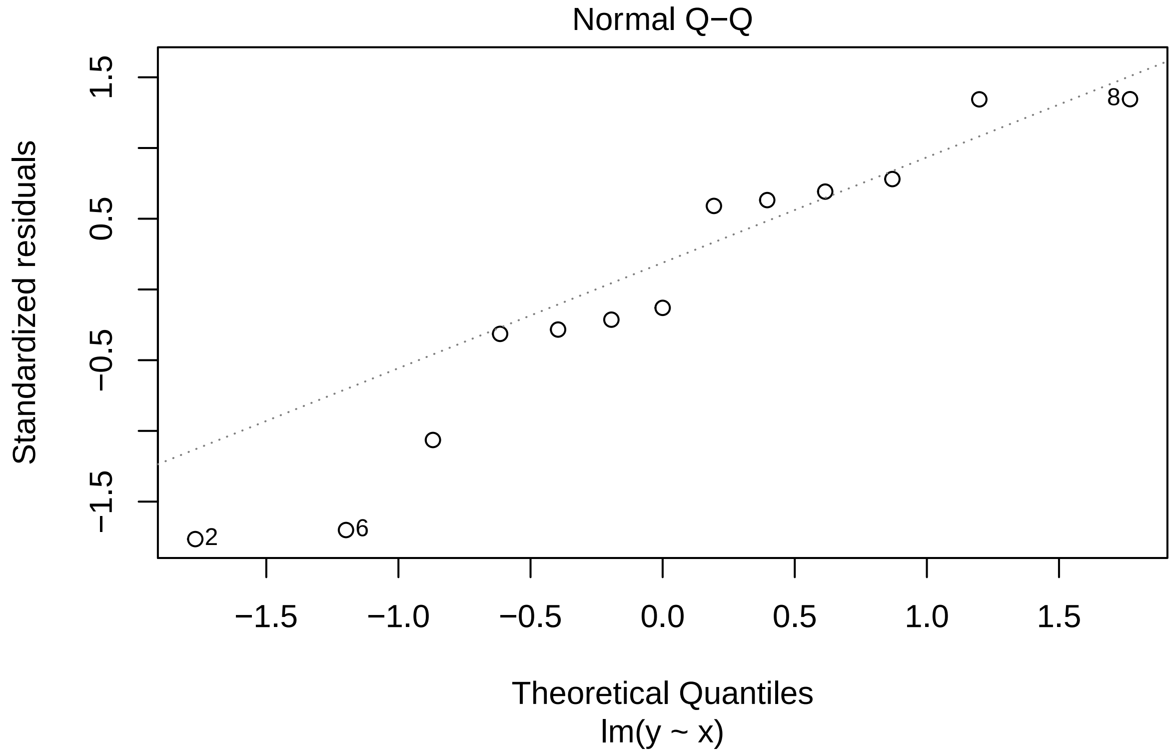
$y = 7.055 + 4.088 \times x$

Running the diagnostic plots

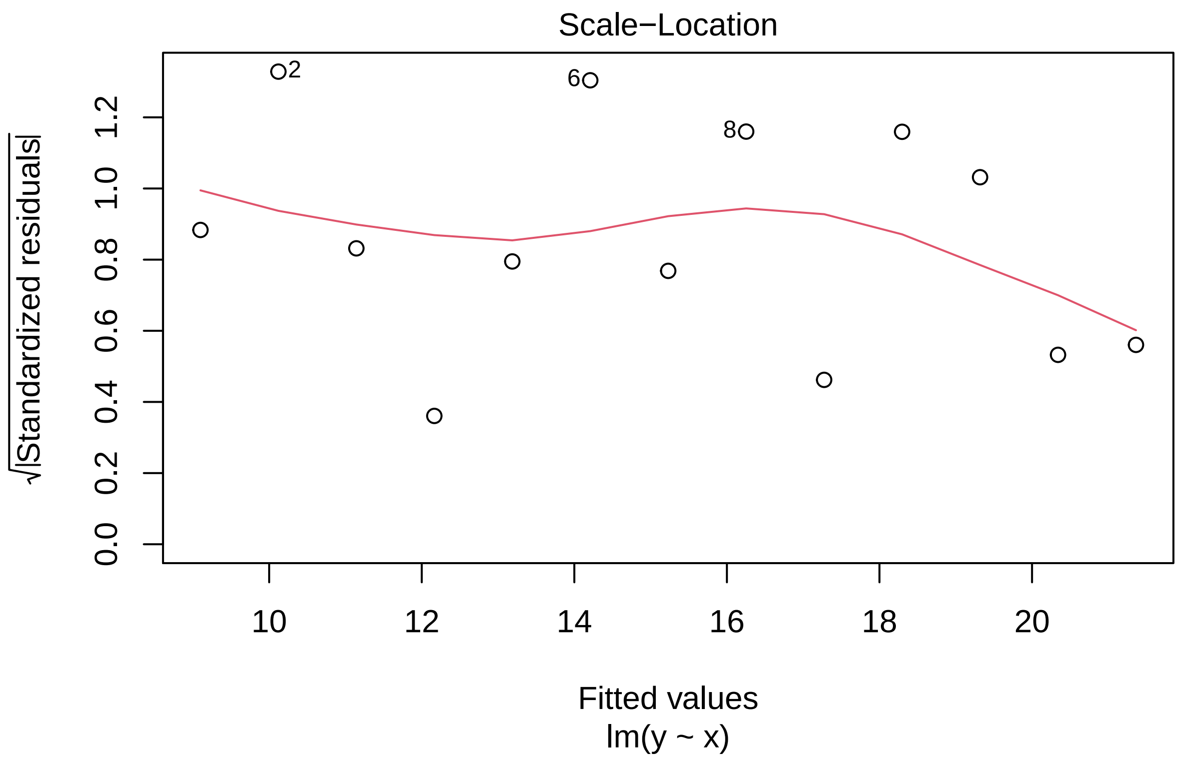
```
plot(lm(y~x))
```

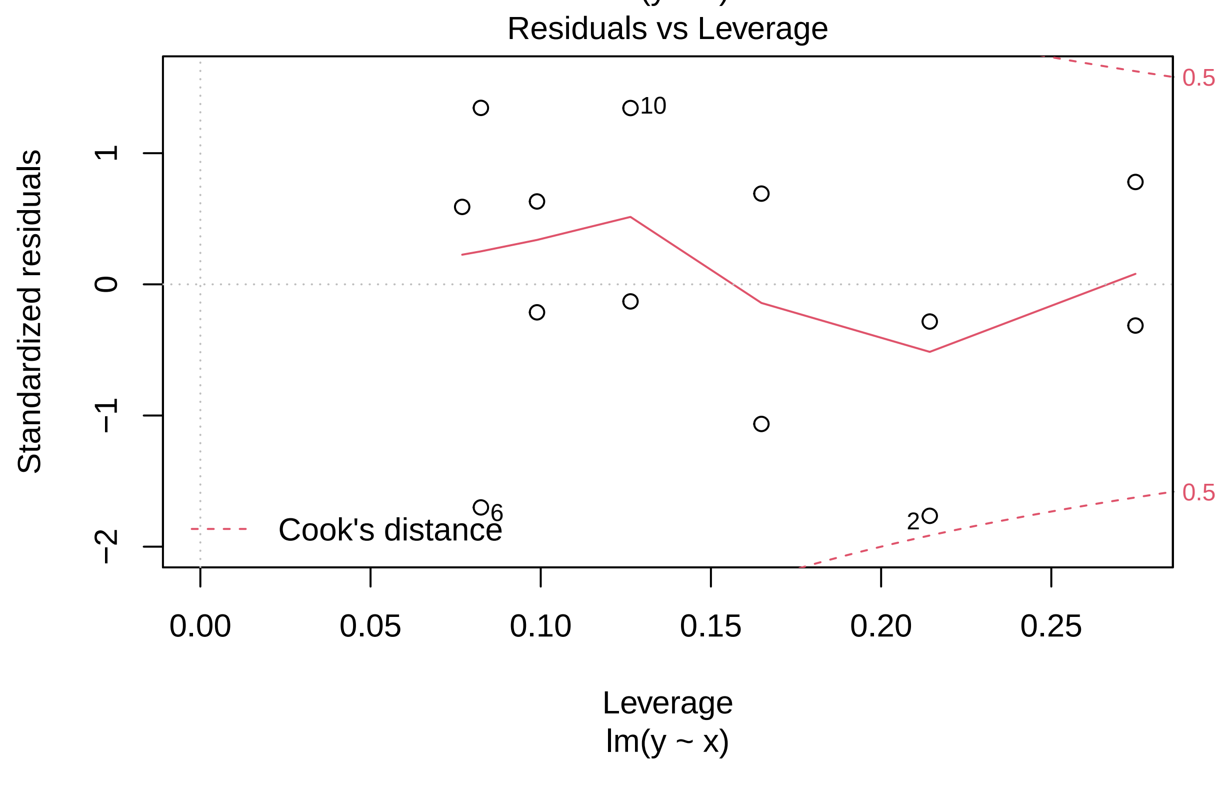


Residuals vs Fitted
Residuals
2
1
0
−1
−2
8
2
6
10
12
14
16
18
20
Fitted values
lm(y ~ x)



Normal Q−Q
Standardized residuals
1.5
0.5
−0.5
−1.5
8
2
6
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(y ~ x)





Summary

This below is another more detailed output

```r
summary(lm(y~x))
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2088 -0.3626 -0.1648  0.8571  1.7472
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.0549     0.8876   7.949 6.94e-06 ***
## x             4.0879     0.4020  10.169 6.25e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.356 on 11 degrees of freedom
## Multiple R-squared:  0.9039, Adjusted R-squared:  0.8951
## F-statistic: 103.4 on 1 and 11 DF,  p-value: 6.25e-07
```

This model has a very high r^2 and the difference between the variances is significant. So the model is very effective at explaining the variation in our Y values. Both coefficients are also significant.

Predict - for a new x value compute the y predicted

If we have a value for x, we can use the regression equation to find an **estimate** for y.

$y = 7.055 + 4.088 \times x$

Let find an estimate for y when x=1.15

```r
y.est<-7.055+4.088*1.15
y.est
```

```
## [1] 11.7562
```

Transforming y?

We may want to transform y and see if we get a better model.