Introduction to Linear Regression in R

Ziyang Lyu, Aniko Toth, & Steefan Contractor

2024-10-26

Simple linear regression

For this analysis, we will use the `cars` dataset that comes with R by default. The data give the speed of cars (in miles per hour) and the distance it took them to stop (in feet) in an experiment recorded in the 1920s.

You can access the dataset by typing `cars` in your console. It's a good idea to explore the data first.

```r
head(cars)  # display the first 6 observations
```

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

```r
tail(cars)  # display the last 6 observations
```

```
##    speed dist
## 45    23   54
## 46    24   70
## 47    24   92
## 48    24   93
## 49    24  120
## 50    25   85
```

The goal is to establish a mathematical equation for `dist` as a function of `speed`, so you can use it to predict `dist` when only the `speed` of the car is known. In this case, `dist` is known as the "response variable" typically represented as Y, and `speed` is the "predictor variable", typically represented as X.

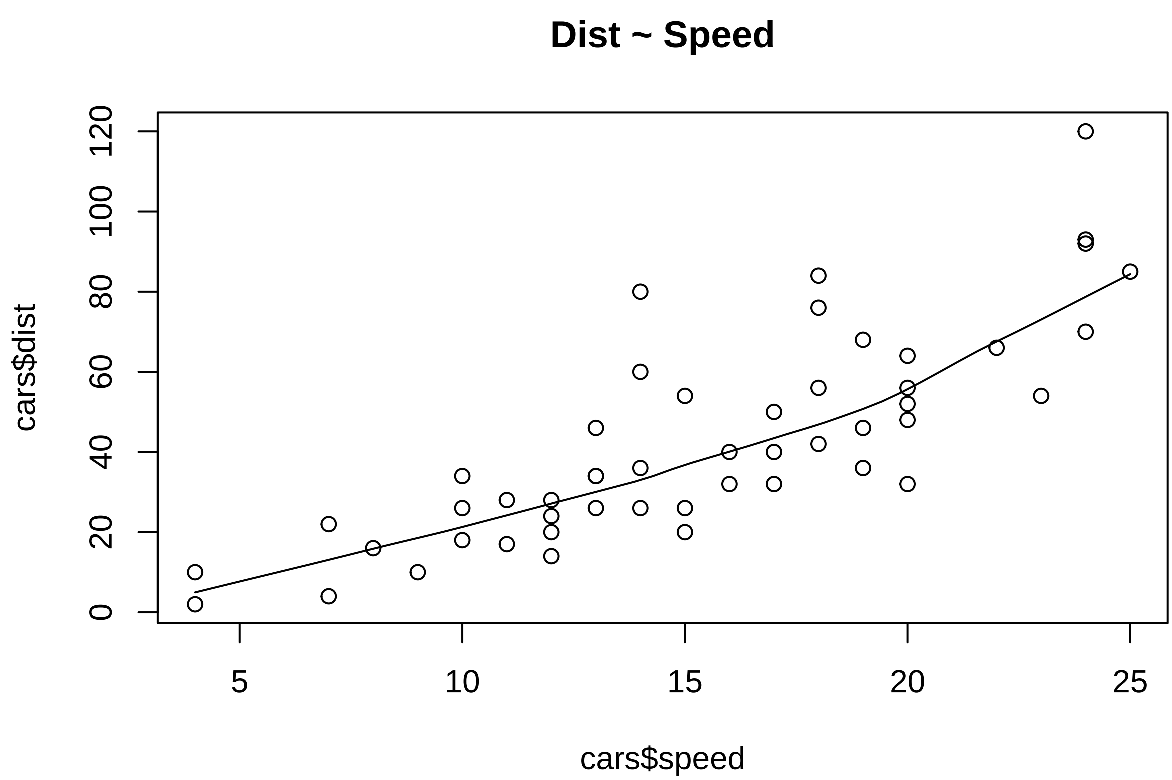
Discussion: What happens if we swap the roles of our variables?

Visualise the data

Scatter plots can help visualise linear relationships between the response and predictor variables.

Ideally, if you have many predictor variables, a scatter plot is drawn for each one of them against the response, along with the line of best fit.

```r
scatter.smooth(x=cars$speed, y=cars$dist, main="Dist ~ Speed")   # scatterplot
```



The scatter plot along with the smoothed line above suggests a linear and positive relationship between the `dist` and `speed`.

One of the underlying assumptions of linear regression is that the relationship between the response and predictor variable is linear.

Exercise 1

1. Read the lung capacity data from the LungCap.txt file.

```r
lc <- read.delim("LungCap.txt") # read data
```

2. Explore the data. Pick a suitable predictor and response variable.

```r
head(lc)
```

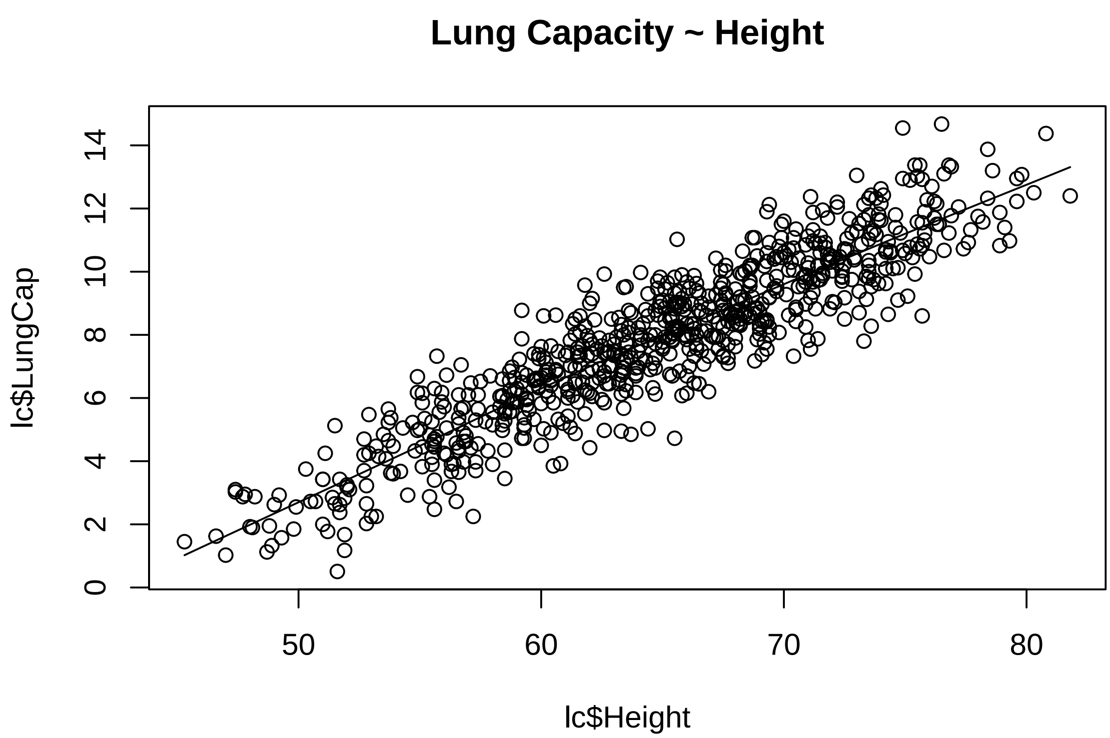
```
##   LungCap Age Height Smoke Gender Caesarean
## 1   6.475   6   62.1    no   male        no
## 2  10.125  18   74.7   yes female        no
## 3   9.550  16   69.7    no female       yes
## 4  11.125  14   71.0    no   male        no
## 5   4.800   5   56.9    no   male        no
## 6   6.225  11   58.7    no female        no
```

```r
tail(lc)
```

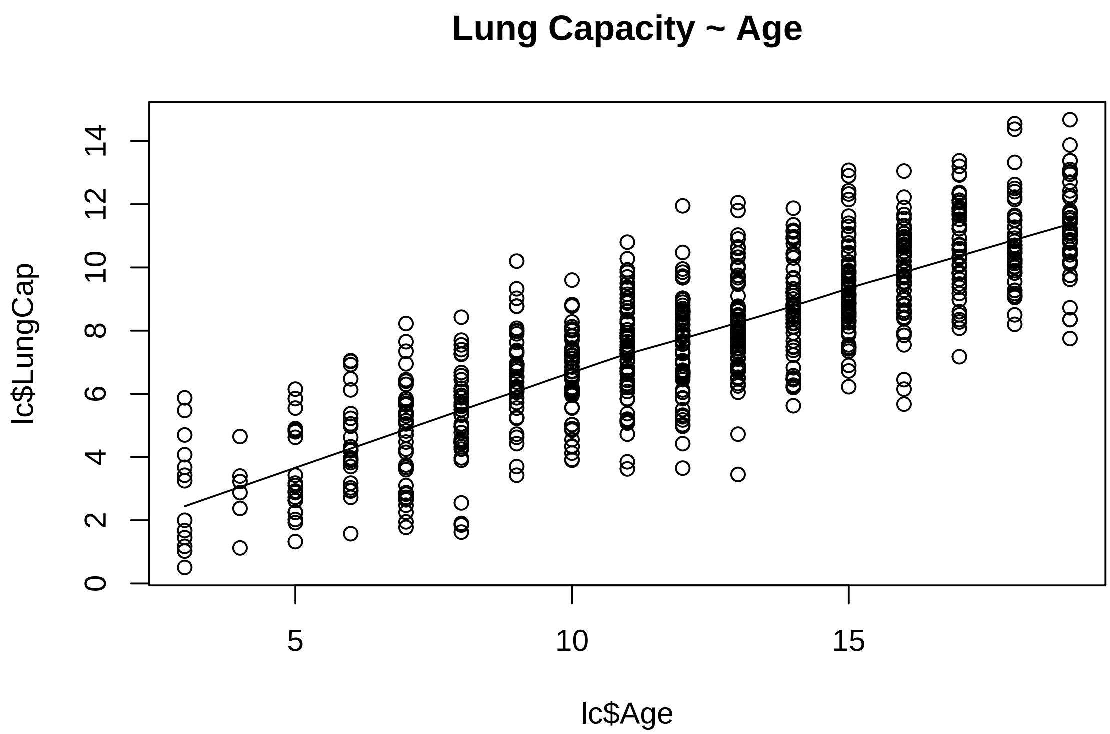
```
##     LungCap Age Height Smoke Gender Caesarean
## 720   7.325   9   66.3    no   male        no
## 721   5.725   9   56.0    no female        no
## 722   9.050  18   72.0   yes   male       yes
## 723   3.850  11   60.5   yes female        no
## 724   9.825  15   64.9    no female        no
## 725   7.100  10   67.7    no   male        no
```

3. Use a scatter plot to visualise the relationship between the chosen variables.

```r
scatter.smooth(x=lc$Height, y=lc$LungCap, main="Lung Capacity ~ Height")
```

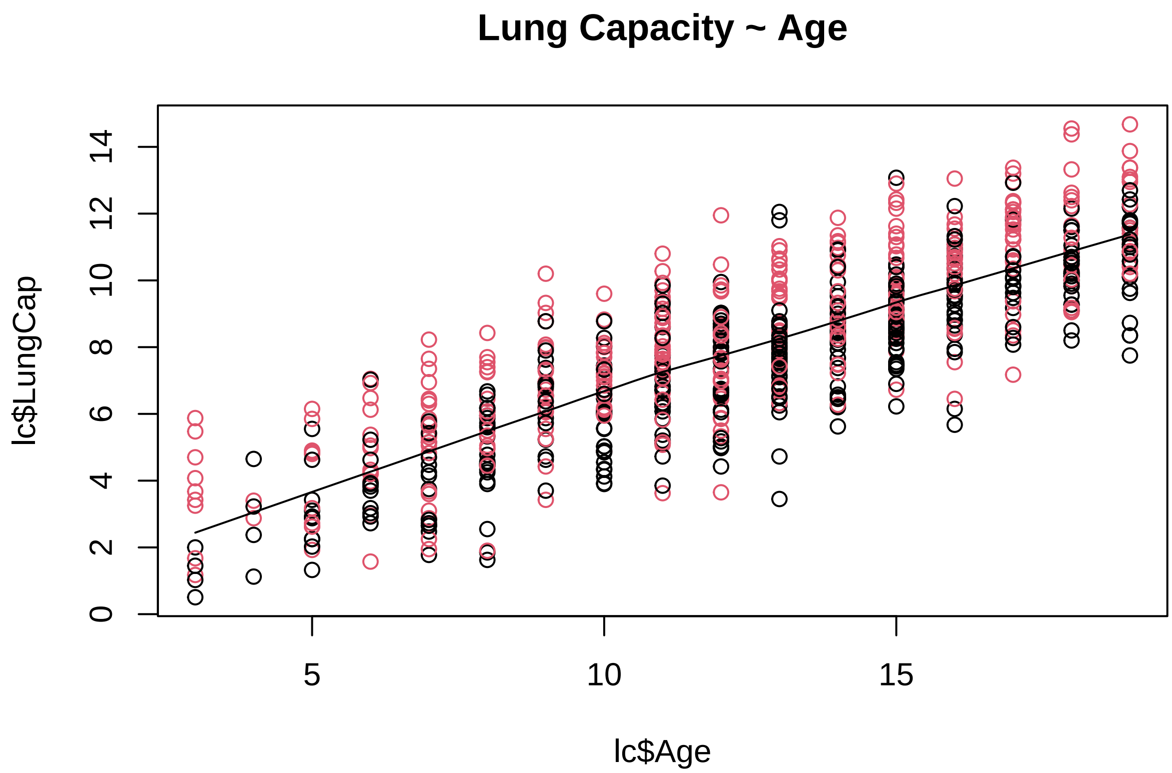



```r
scatter.smooth(x=lc$Age, y=lc$LungCap, main="Lung Capacity ~ Age")
```



BONUS: colour the points by gender. Do you think gender affects your response variable?

```
scatter.smooth(x=lc$Age, y=lc$LungCap, col = as.factor(lc$Gender), main="Lung Capacity ~ Age")
```



Correlation Analysis

Now that we have evidence that the relationship between our two variables is linear, it is helpful to estimate the strength of that relationship. Enter correlation analysis.

The correlation between two variables can take values between -1 to +1. High values indicate that one variable consistently increases with the other, while low values (close to -1) indicate that one variable consistently decrease while the other increases.

Values near 0 suggest a weak relationship between the variables, meaning much of variation of the response variable is unexplained by the predictor. In that case, you may need to look for better explanatory variables.

Correlation does not imply causation: if two variables have high correlation, it does not mean one variable 'causes' the value of the other variable to increase. You can use reasoning or expertise to make that judgement.

Let's compute the correlation of `speed` and `dist` in the cars dataset.

```
cor(cars$speed, cars$dist)  # calculate correlation between speed and distance
```

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

Build the Linear Regression Model

Now that you have visualised the linear relationship in the scatter plot and estimated its strength through correlation, let's build the linear regression model.

The function used for building linear models is`lm()`. It requires two arguments: the formula summarising the relationship between the variables and the data table.

```
model <- lm(dist ~ speed, data=cars)  # build linear regression model on full data
print(model)
```

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

By building the linear regression model, we have established the relationship between speed and stopping distance in the form of a mathematical formula. Notice the print function reports two Coefficients: Intercept: -17.579, speed: 3.932.

In other words, dist = -17.579 + 3.932*speed

Exercise 2

1. Calculate the correlation between your response and predictor variable from the Lung Capacity data.

```
cor(lc$LungCap, lc$Age)  # calculate correlation between speed and distance
```

```
## [1] 0.8196749
```

2. Build a linear model and find its coefficients

```
model2 <- lm(LungCap ~ Age, data=lc)
print(model2)
```

```
##
## Call:
## lm(formula = LungCap ~ Age, data = lc)
##
## Coefficients:
## (Intercept)          Age
##      1.1469       0.5448
```

3. Write down the equation that represents the relationship between your variables.

BONUS: Add gender to your model. Does being a male affect lung capacity?

```r
model3 <- lm(LungCap ~ Age + Gender, data=lc)
print(model3)
```

```
##
## Call:
## lm(formula = LungCap ~ Age + Gender, data = lc)
##
## Coefficients:
## (Intercept)          Age   Gendermale
##      0.5737       0.5488       1.0367
```

Check if the model meets assumptions

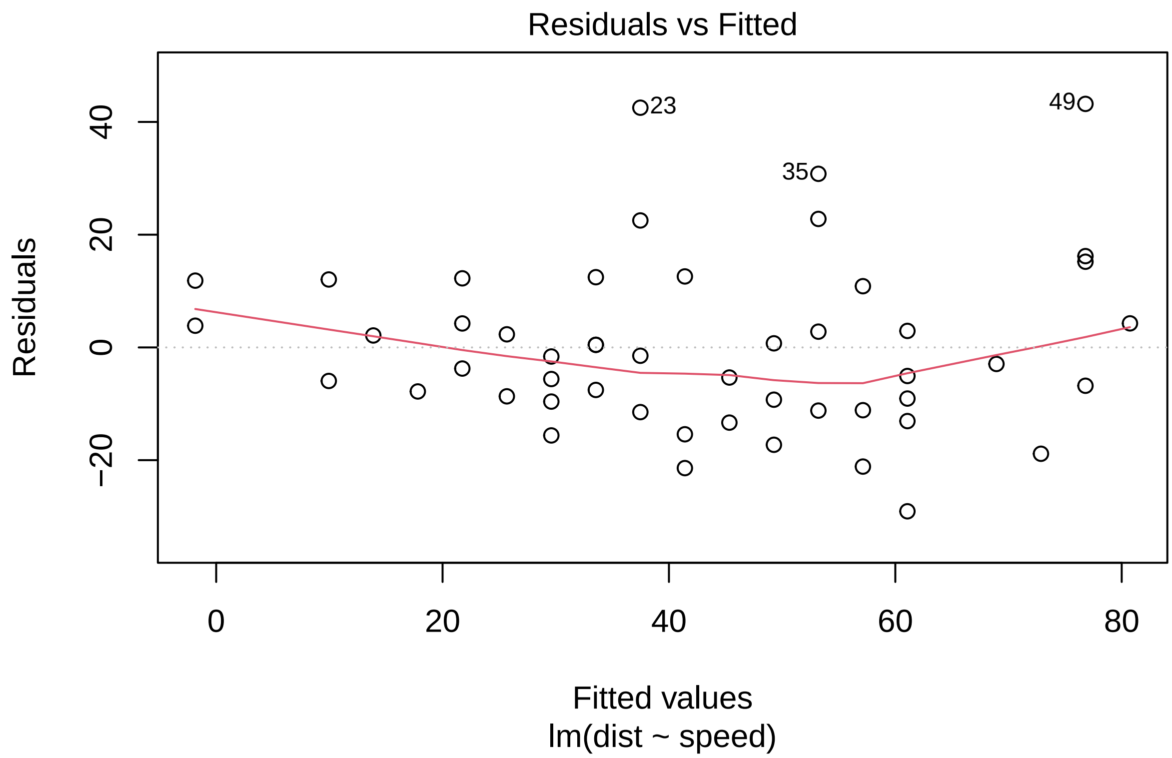
Now the linear model is built and you have a formula that you can use to predict the `dist` value if a corresponding `speed` is known. Is this enough to actually use this model? NO!

Simple linear regression belongs to a family of linear models that must all meet the following assumptions:

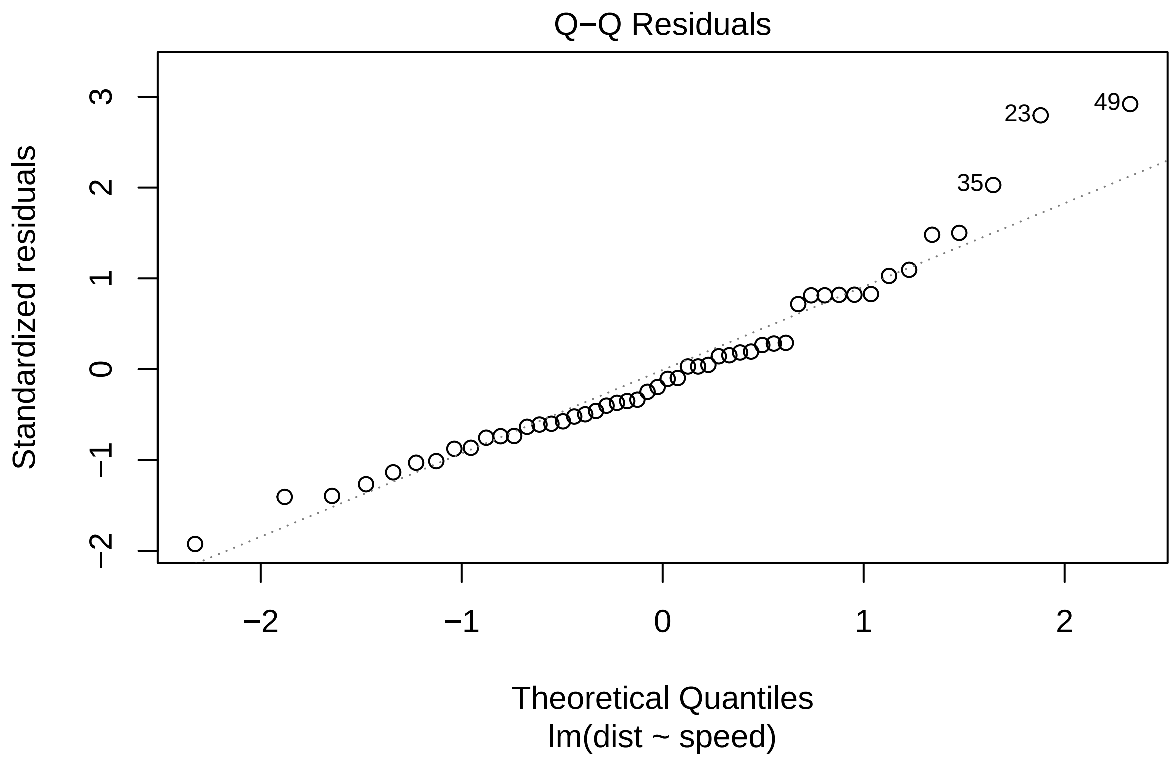
- **Linearity**: The relationship between variables is linear.
- **Independence**: Data points are independent of each other.
- **Homoscedasticity**: Constant variance of errors.
- **Normality**: The residuals (errors) should be normally distributed.

Now that we understand the assumptions that must be satisfied, the following plots can help us check them.

```r
plot(model, which = 1) # check homoscedasticity and linearity
```




```
plot(model, which = 2) # check normality
```



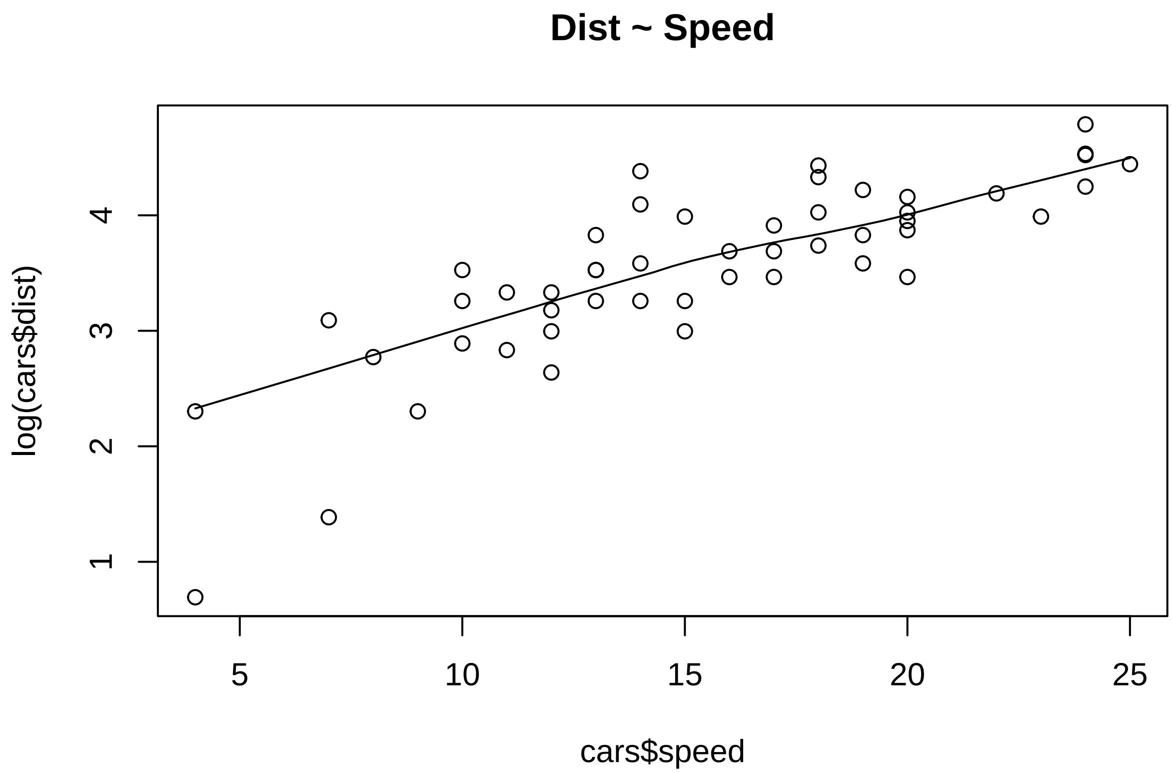
What happens if the assumptions are not satisfied?

- Try transforming the response or predictor variables.
- If that doesn't work, try a different model/predictor variable.

These are some useful transformations

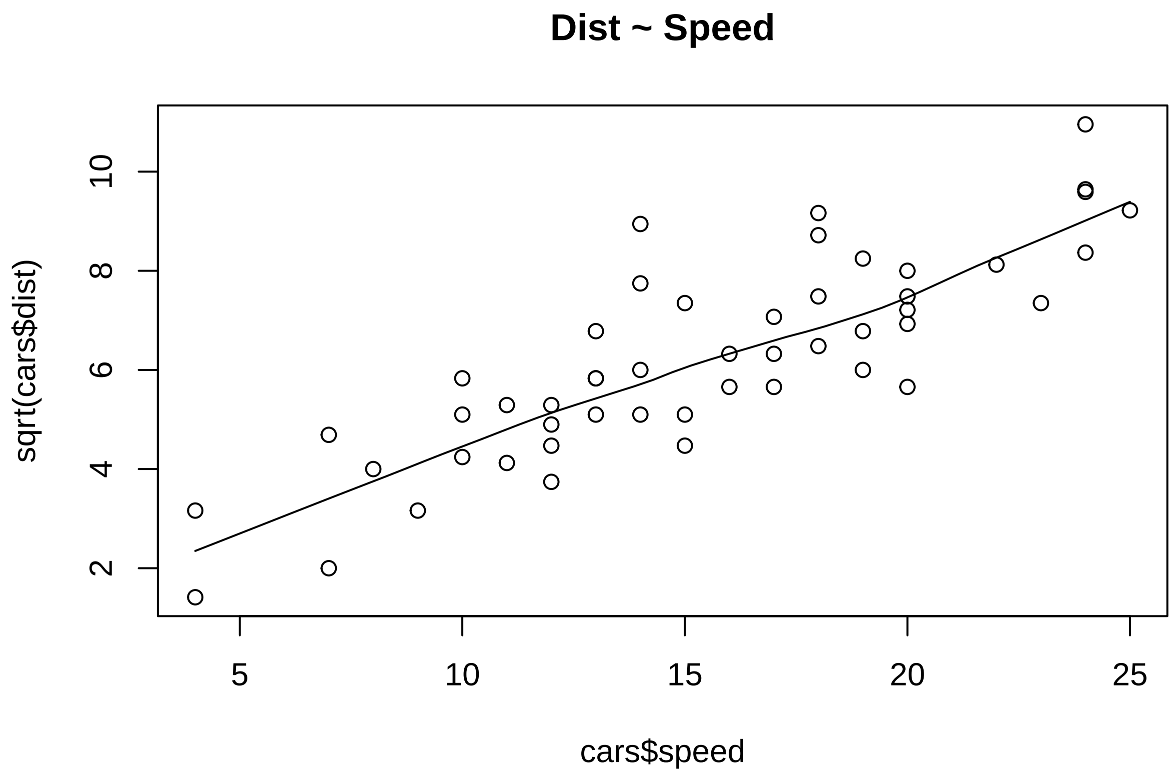
Log transformation: `log()` to stabilise variance and heteroscedasticity in the residuals

```
scatter.smooth(x=cars$speed, y=log(cars$dist), main="Dist ~ Speed")  # scatterplot
```



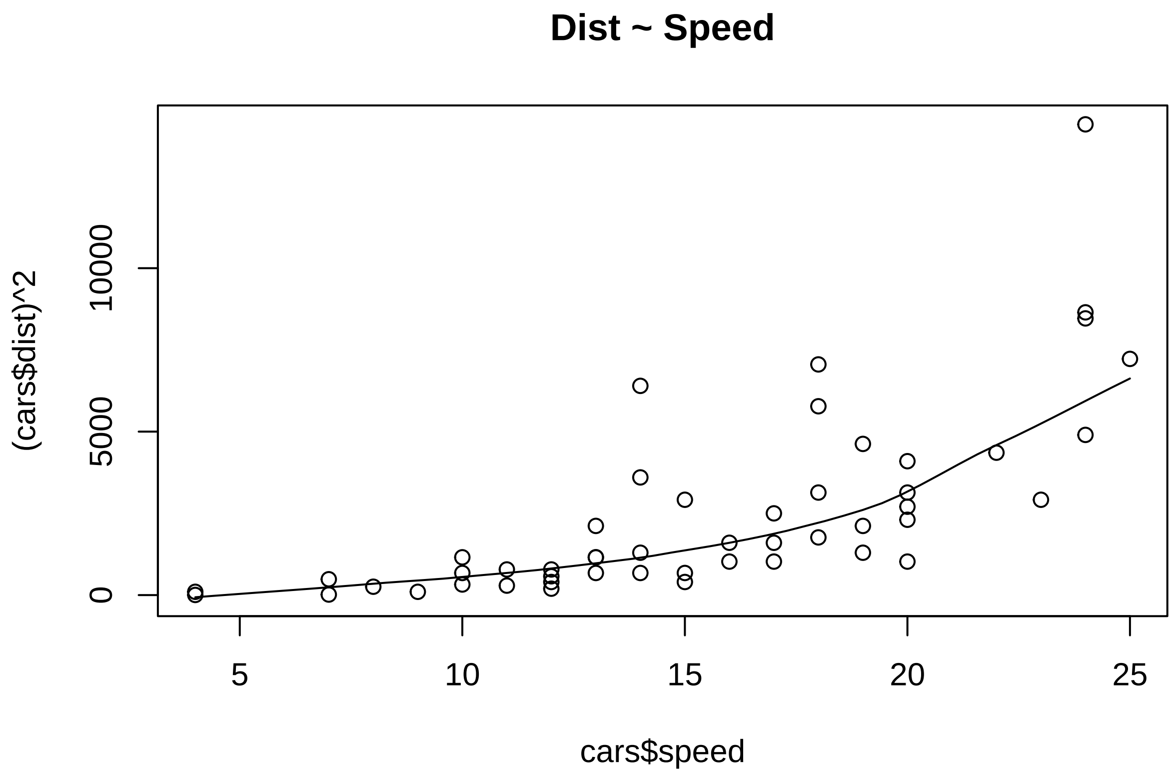
Square root: `sqrt()` to normalise skewed data or reduce the influence of high values

```
scatter.smooth(x=cars$speed, y=sqrt(cars$dist), main="Dist ~ Speed")  # scatterplot
```



Square: `variable^2` to linearise noninear/curved relationships.

```
scatter.smooth(x=cars$speed, y=(cars$dist)^2, main="Dist ~ Speed")  # scatterplot
```



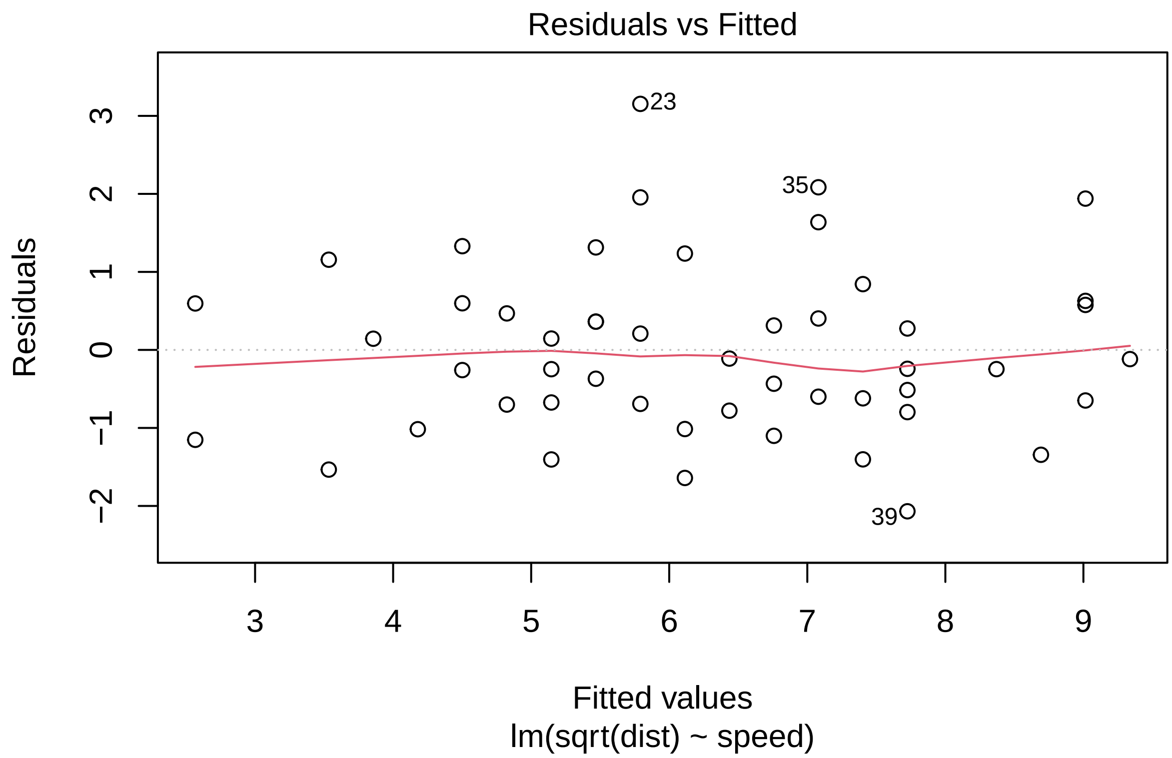
Which of these transformations appears to be the most appropriate?

Let's refit the model with our transformed data. Are the assumptions met?

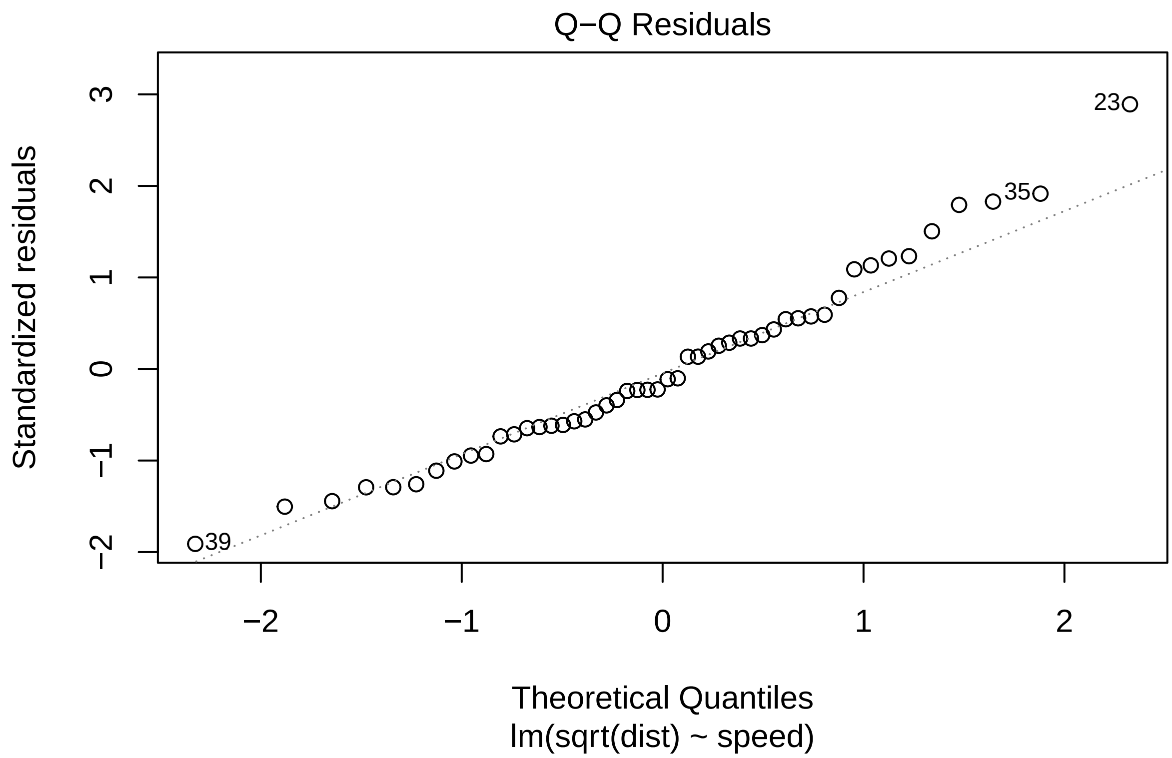
```
model <- lm(sqrt(dist) ~ speed, data=cars)
print(model)
```

```
##
## Call:
## lm(formula = sqrt(dist) ~ speed, data = cars)
##
## Coefficients:
## (Intercept)        speed
##      1.2771       0.3224
```

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



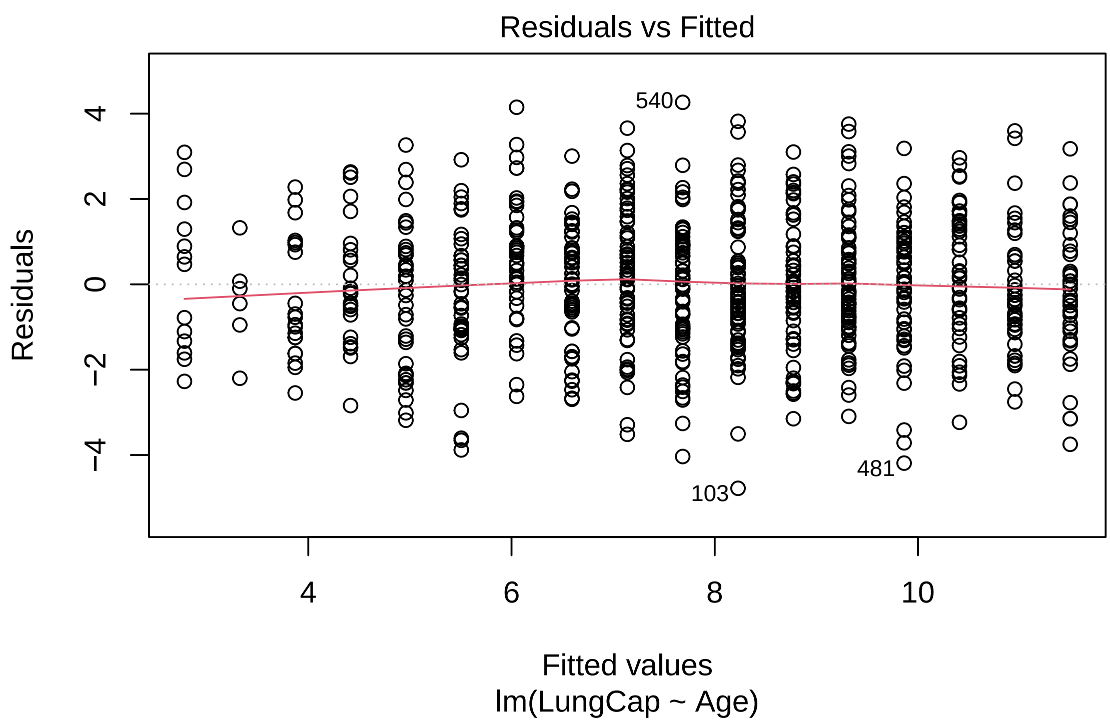

```r
plot(model, which = 2)
```



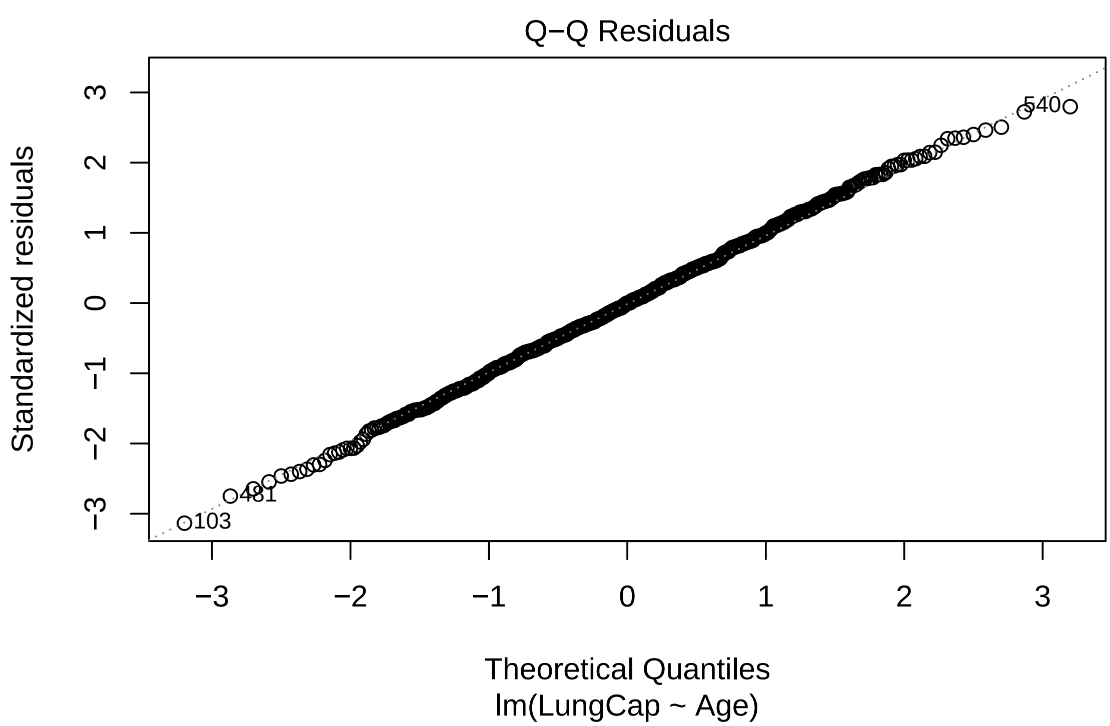
Exercise 3

1. Check the homoscedasticity and normality assumptions in your lung capacity model. Are the assumptions met?

```
plot(model2, which = 1)
```




```r
plot(model2, which = 2)
```



Checking the goodness of fit

Let's begin by printing the summary statistics for `model`.

```
summary(model)  # model summary
```

```
##
## Call:
## lm(formula = sqrt(dist) ~ speed, data = cars)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.0684 -0.6983 -0.1799  0.5909  3.1534
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.27705    0.48444   2.636   0.0113 *
## speed        0.32241    0.02978  10.825 1.77e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.102 on 48 degrees of freedom
## Multiple R-squared:  0.7094, Adjusted R-squared:  0.7034
## F-statistic: 117.2 on 1 and 48 DF,  p-value: 1.773e-14
```

Exercise 4

1. Is your transformed speed vs stopping distance model a good fit?
2. Is speed a significant indicator of stopping distance?
3. Is your lung capacity model a good fit?
4. Is your chosen predictor a significant indicator of lung capacity?

Prediction

To make predictions with a fitted model use the predict() function.

```
newspeed <- data.frame(speed=c(10,20,5))
newdist <- predict(model, newspeed)
# in this case, we transformed the variables, so we need to reverse the transformation
dist <- newdist^2
```

Exercise

1. Predict the lung capacity of a 16 year old individual.

   ```
   newage <- data.frame(Age=c(16))
   newLC <- predict(model2, newage)
   ```

BONUS: Predict the lung capacity of a 10 year old female.

```r
newdat <- data.frame(Age = c(10), Gender = c("female"))
predict(model3, newdat)
```

```
##        1
## 6.061402
```

Challenge

Repeat the analysis with the speed_vs_drag.txt dataset!