Concepts in linear regression

Linear regression models

A **simple linear regression model** takes on the following form:

$$Y = \beta_0 + \beta_1 x + \varepsilon$$

- The variable Y is the **response variable**.
- The variable x is the **explanatory variable**.
- The constants β_0 and β_1 are the **regression coefficients**.
- The variable ε is the **error term** (assumed that $E(\varepsilon) = 0$).

The variable ε is a random variable, which means Y is also a random variable.

However x is not a random variable — you can think of x as the 'known' information.

The goal when creating a linear regression model is to *estimate* β_0 and β_1.

Example

Suppose we were investigating the effect of the amount of fertiliser on crop yield. We may choose a linear model. In that case, with

$$Y = \beta_0 + \beta_1 x + \varepsilon,$$

we take Y as a measure of yield (e.g. kg of usable plant matter). Then:

- Interpret x as "the amount of fertiliser"
- Interpret β_0 as "base crop yield"
- Interpret β_1 as "the effect of fertiliser"
- Interpret ε as "the effect of random factors"

The task is to determine β_0 and β_1 on the basis of sample data.

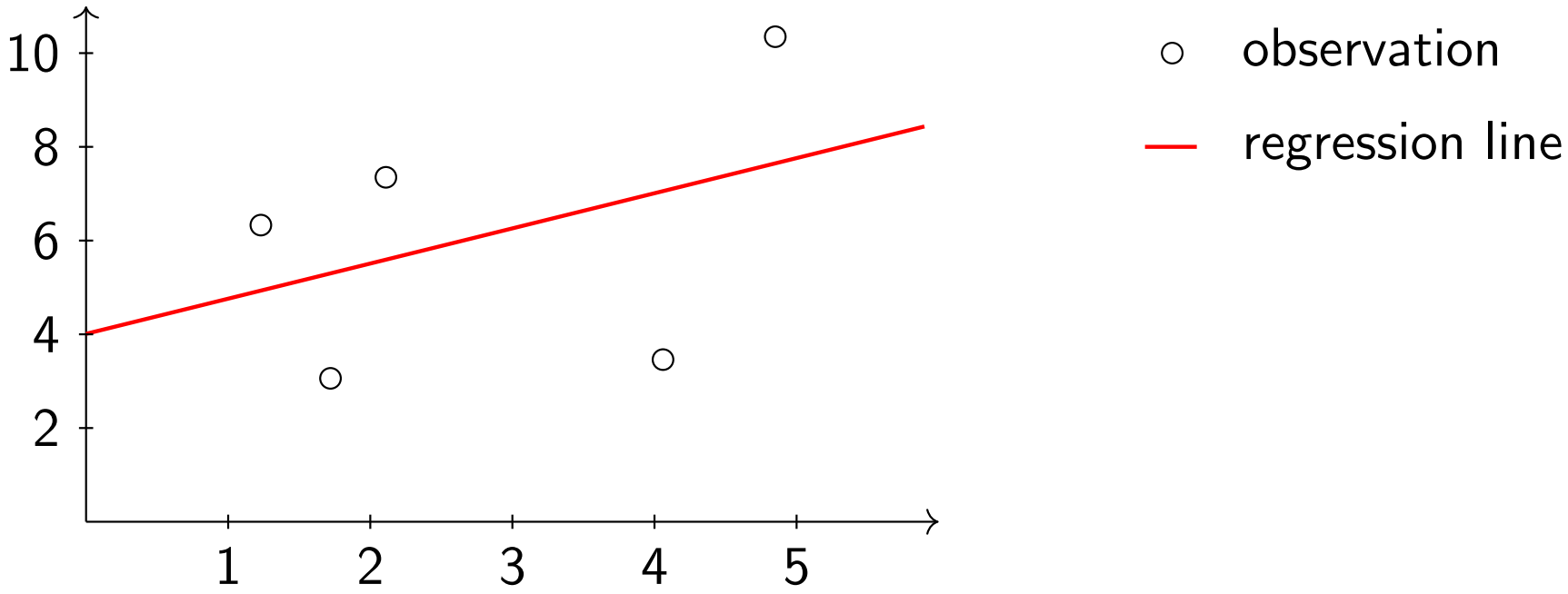
Visualising linear models

For this example, the observations are:

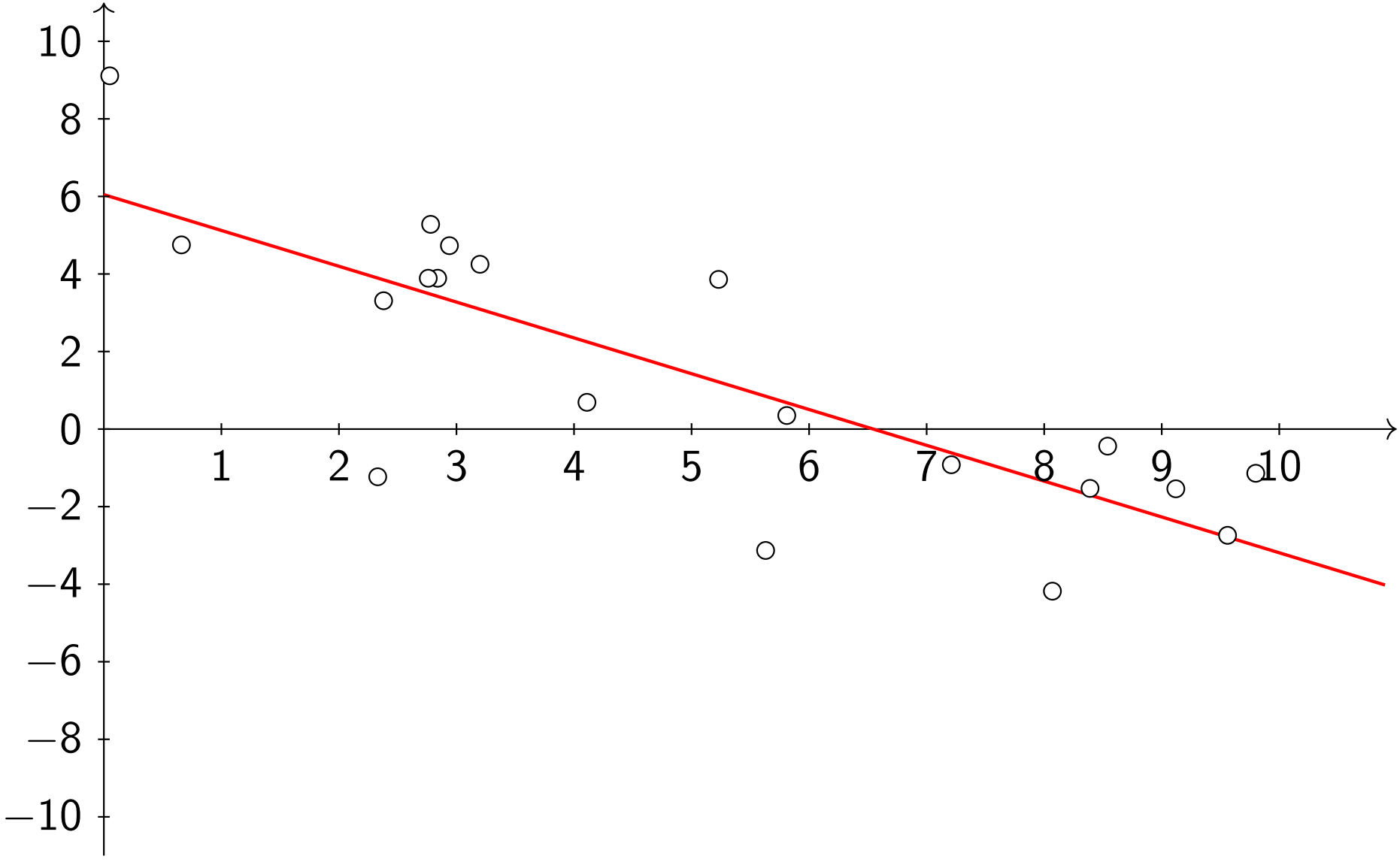
$$(1.23, 6.33), (1.72, 3.06), (2.11, 7.35), (4.06, 3.46), (4.85, 10.35)$$

According to R, the least squares regression model is (to two decimals)

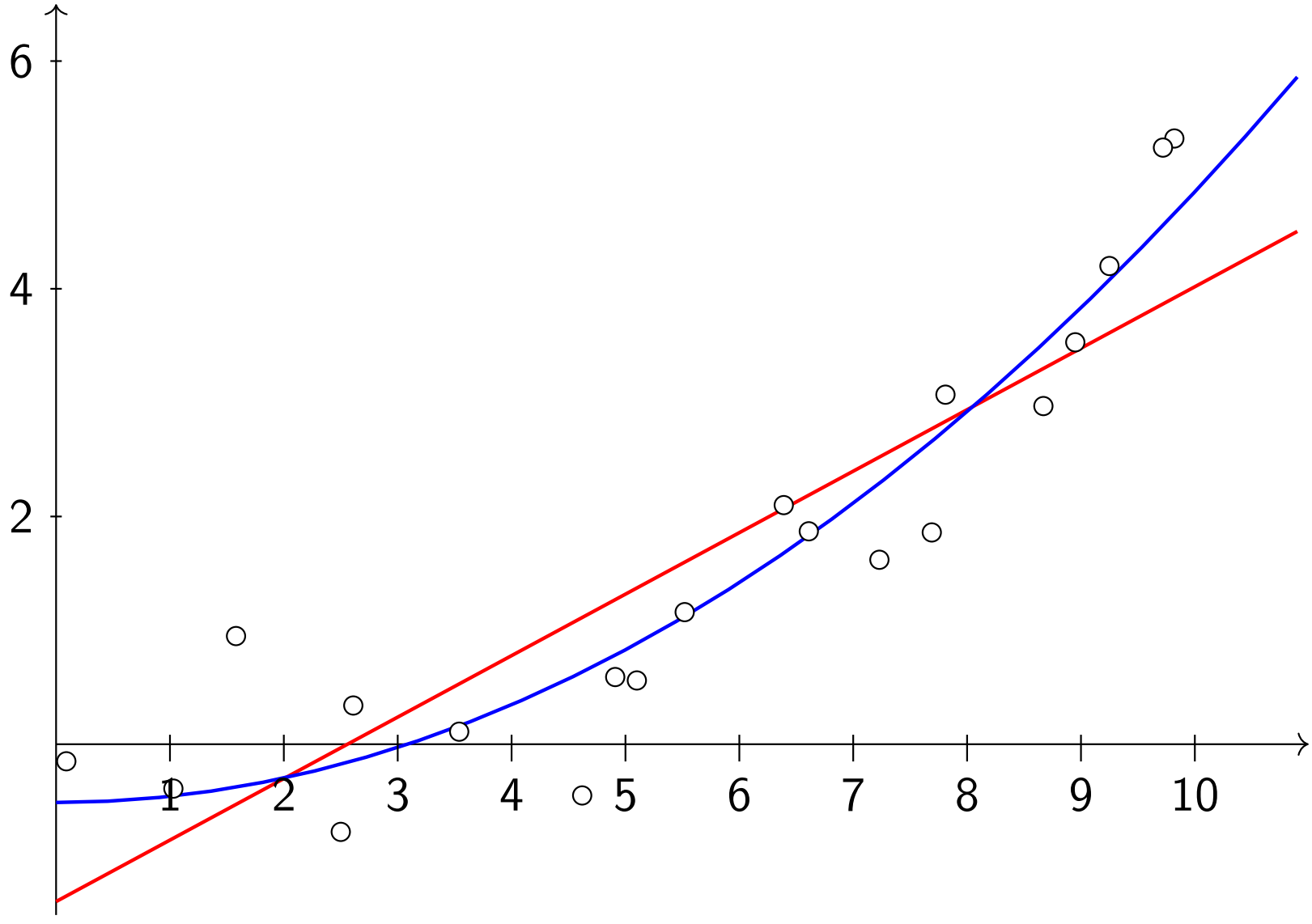
$$\widehat{y} = 4.01 + 0.75x$$



Visualising linear models



Visualising linear models



Visualising linear models: a problem

We can create a linear model with more than one explanatory variable:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \cdots + \beta_n x_n + \varepsilon$$

For example, you might predict house prices on the basis of:

- number of bedrooms,
- number of bathrooms,
- total area of the property,
- total area of the living room.

It's simply not possible to have a graph with four explanatory variables and one response variable.

We are going to see how to use *fits and residuals* to judge the linearity of a data set. Residuals are also used to generate the linear model. We will explore this in the next video.

Understanding fits and residuals

Setup

The situation:

- You have a set of observations $(x_1, y_1), \ldots, (x_n, y_n)$.
- You have a linear model $\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$.

We can then take our data and put it into the model, giving us a set of predictions (one for each data point).

Each "predicted value" is called a **fitted value**, or a **fit** for short:

$$\widehat{y}_i = \widehat{\beta}_0 + \widehat{\beta}_1 x_i$$

Then, the **residuals** are the differences between the fitted values and the observed values of y:

$$\begin{aligned} i\text{-th residual} &= \text{observed value} - \text{fitted value} \\ &= y_i - \widehat{y}_i \end{aligned}$$

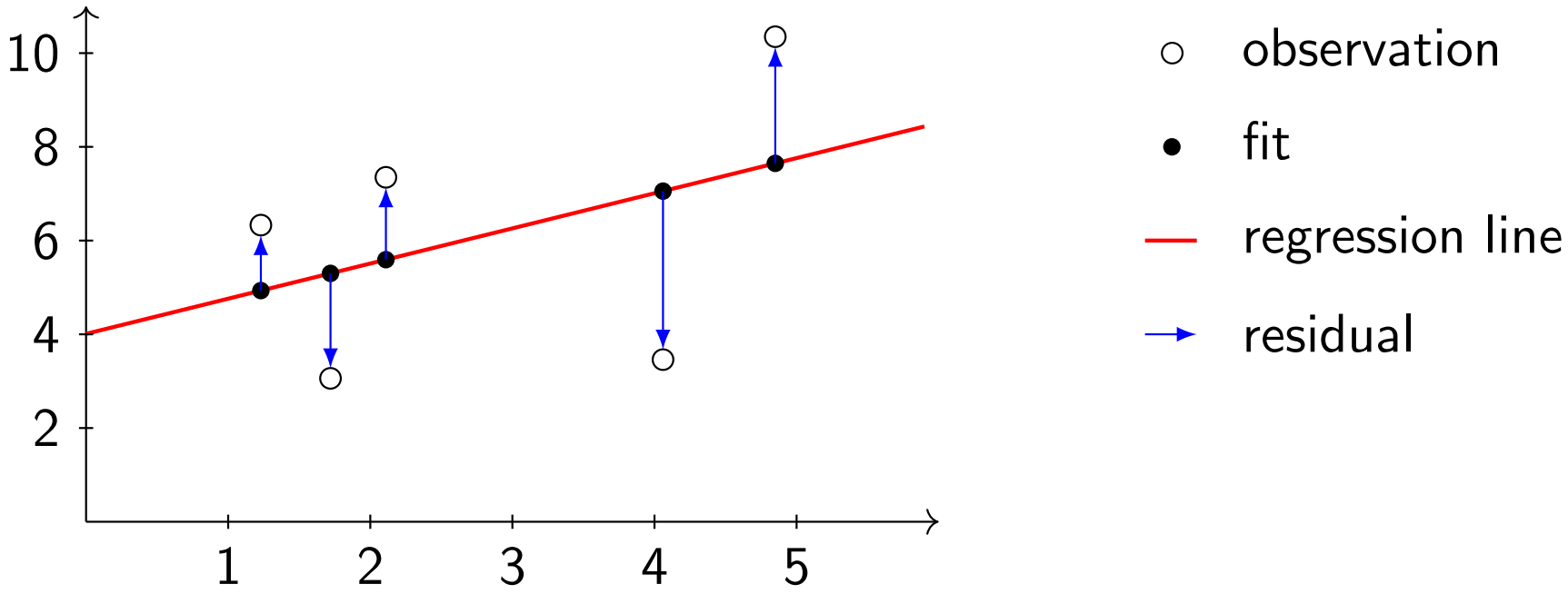
Fits and residuals

For this example, the observations are:

$$(1.23, 6.33), (1.72, 3.06), (2.11, 7.35), (4.06, 3.46), (4.85, 10.35)$$

According to R, the least squares regression model is (to two decimals)

$$\widehat{y} = 4.01 + 0.75x$$



Fits and residuals

For this example, the observations are:

$$(1.23, 6.33), (1.72, 3.06), (2.11, 7.35), (4.06, 3.46), (4.85, 10.35)$$

According to R, the least squares regression model is (to two decimals)

$$\widehat{y} = 4.01 + 0.75x$$

Numerically:

Estimating the coefficients using residuals

We will not be focusing on the calculations for estimating the coefficients, but a brief overview will be given here.

The estimated model takes on this form:

$$\widehat{y} = \widehat{\beta}_0 + \widehat{\beta}_1 x$$

The linear regression model performs a calculation which minimises the *sum of squared residuals*. That is, it finds β_0 and β_1 such that

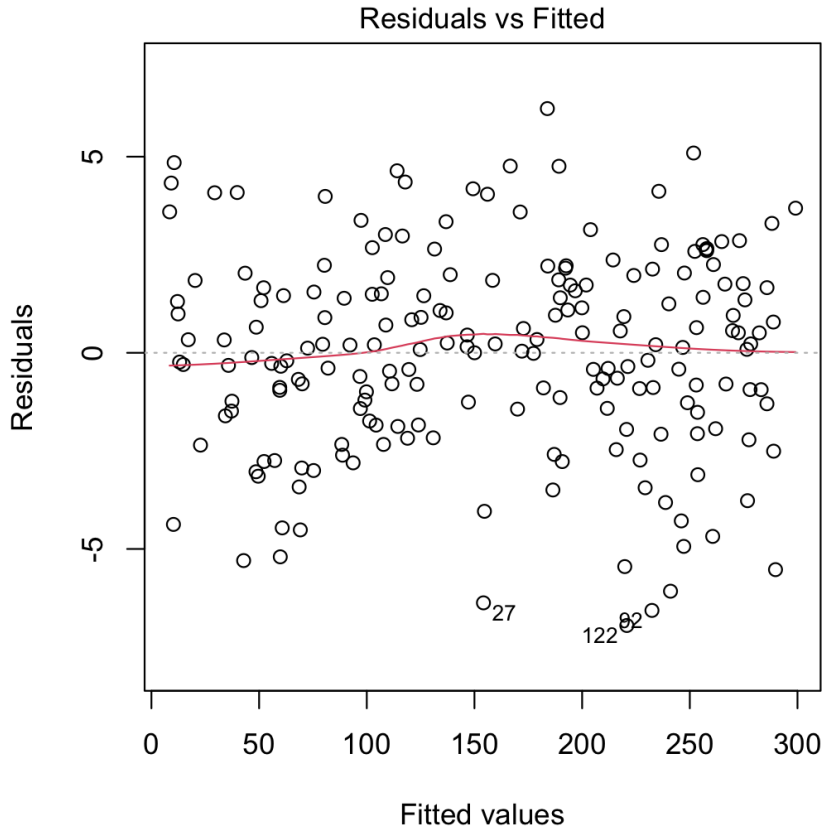
$$\sum_{i=1}^{n} (y_i - \widehat{y}_i)^2$$

is as small as possible.

The calculations involved are not difficult (search "how to calculate least squares regression" on your favourite search engine), but are best done by a computer. The `lm` function in R will do this for us.

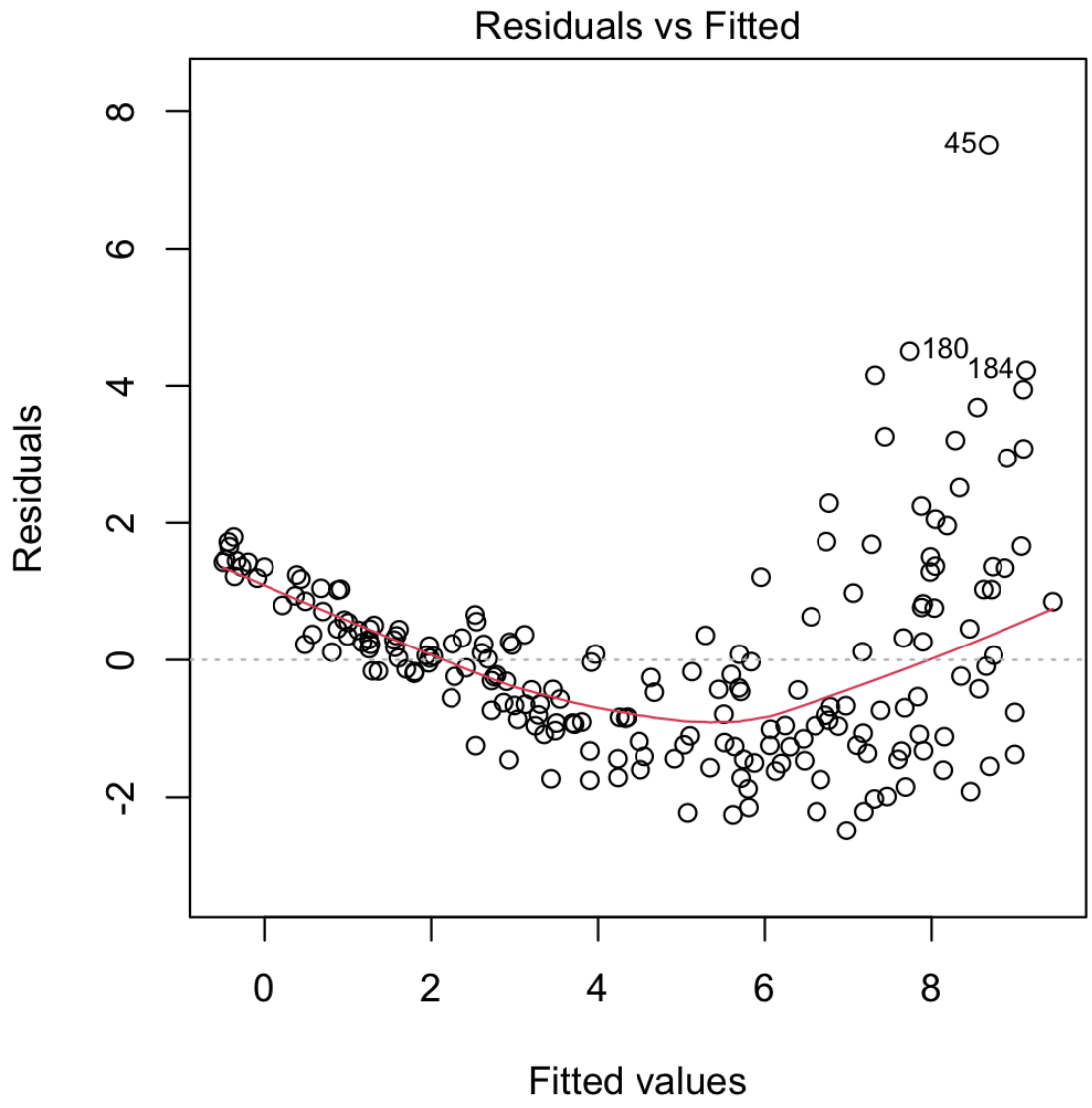
Fits versus residuals

After creating a linear model, we can create a *fits versus residuals* plot. Each fit is plotted on the horizontal axis, with its corresponding residual on the vertical axis.



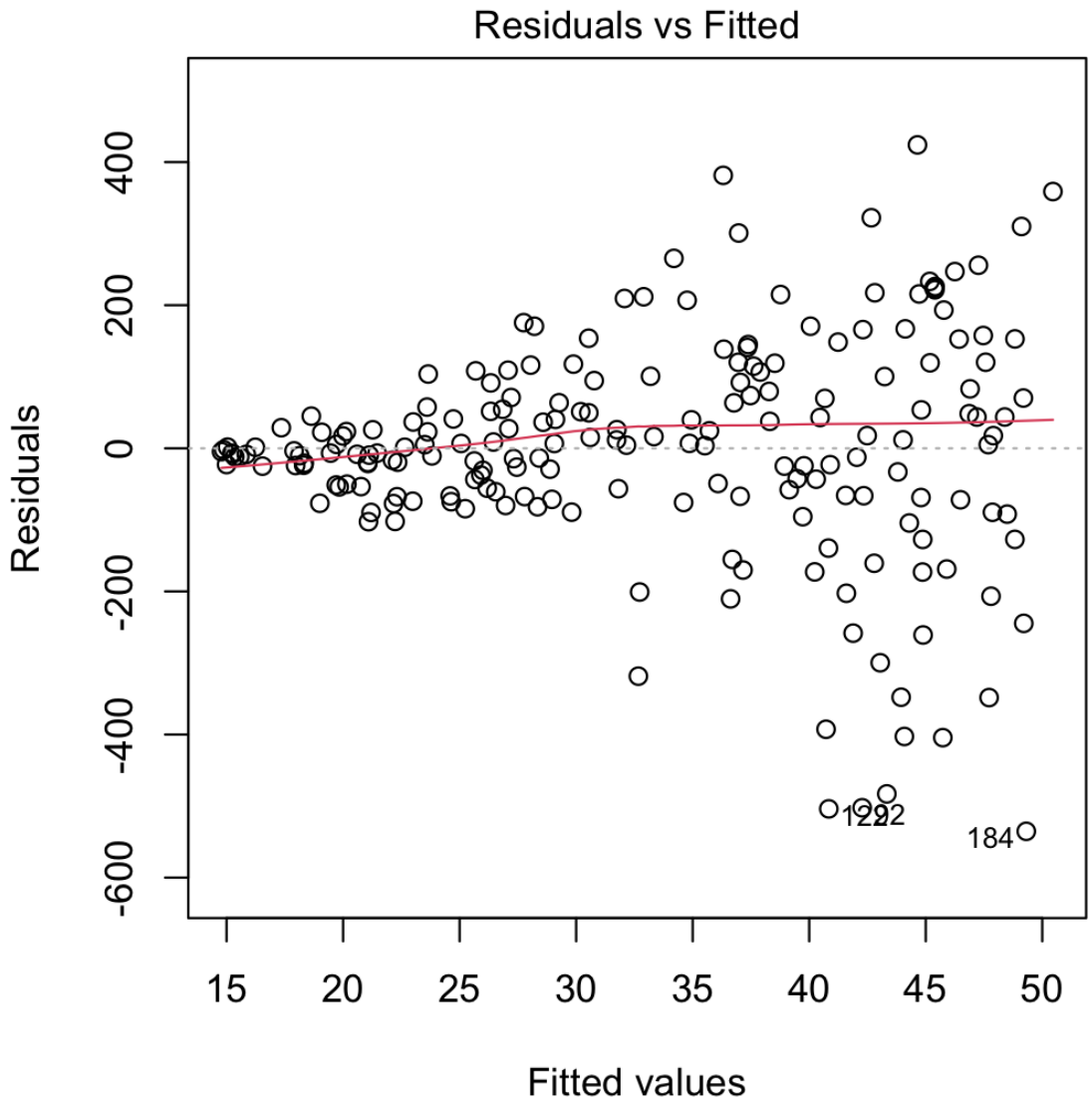
If the data is linear, then the fits versus residuals plot should demonstrate a pattern of randomness deviating from a horizontal line.

Fits versus residuals



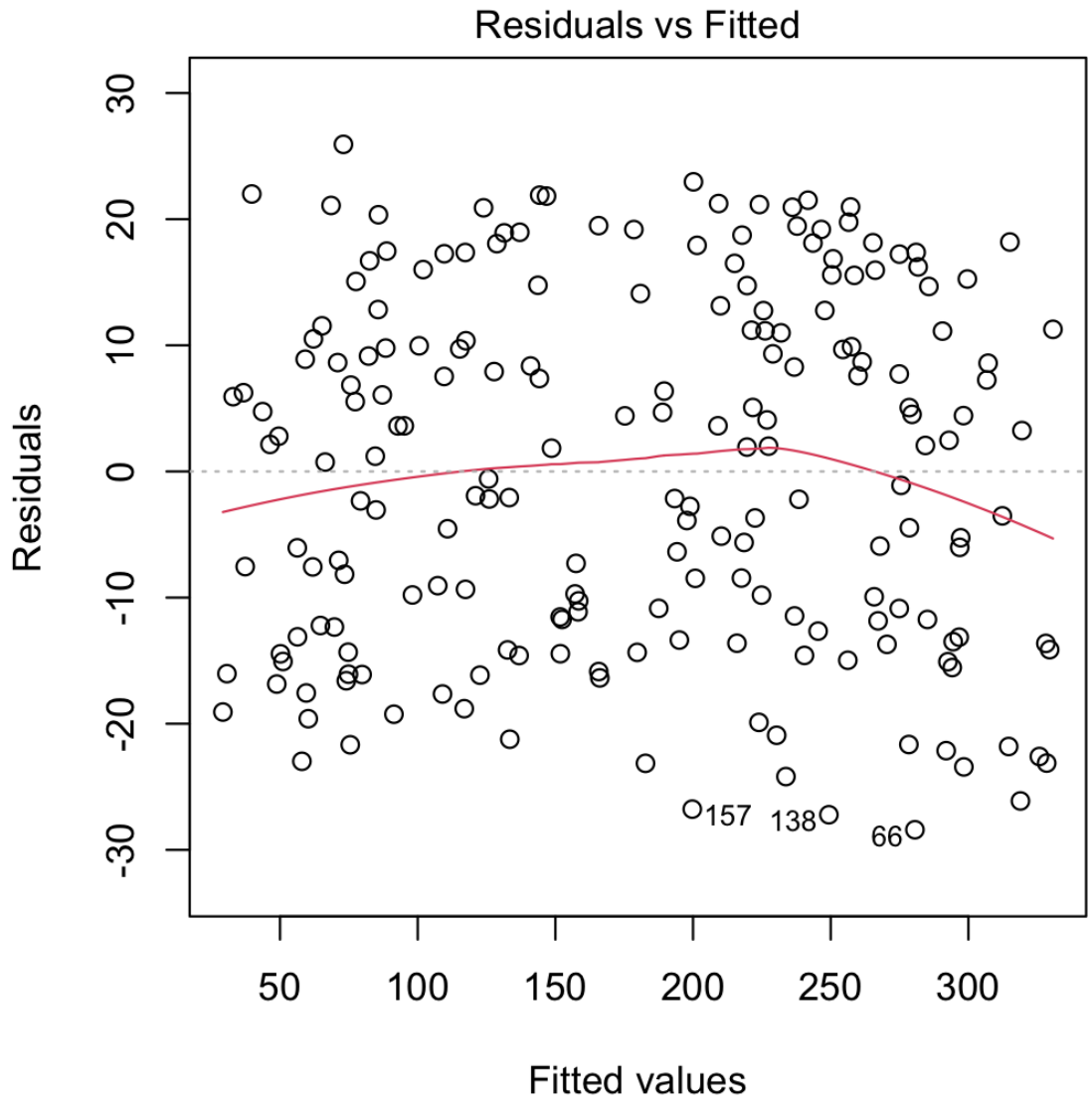
Clearly violates linearity.

Fits versus residuals



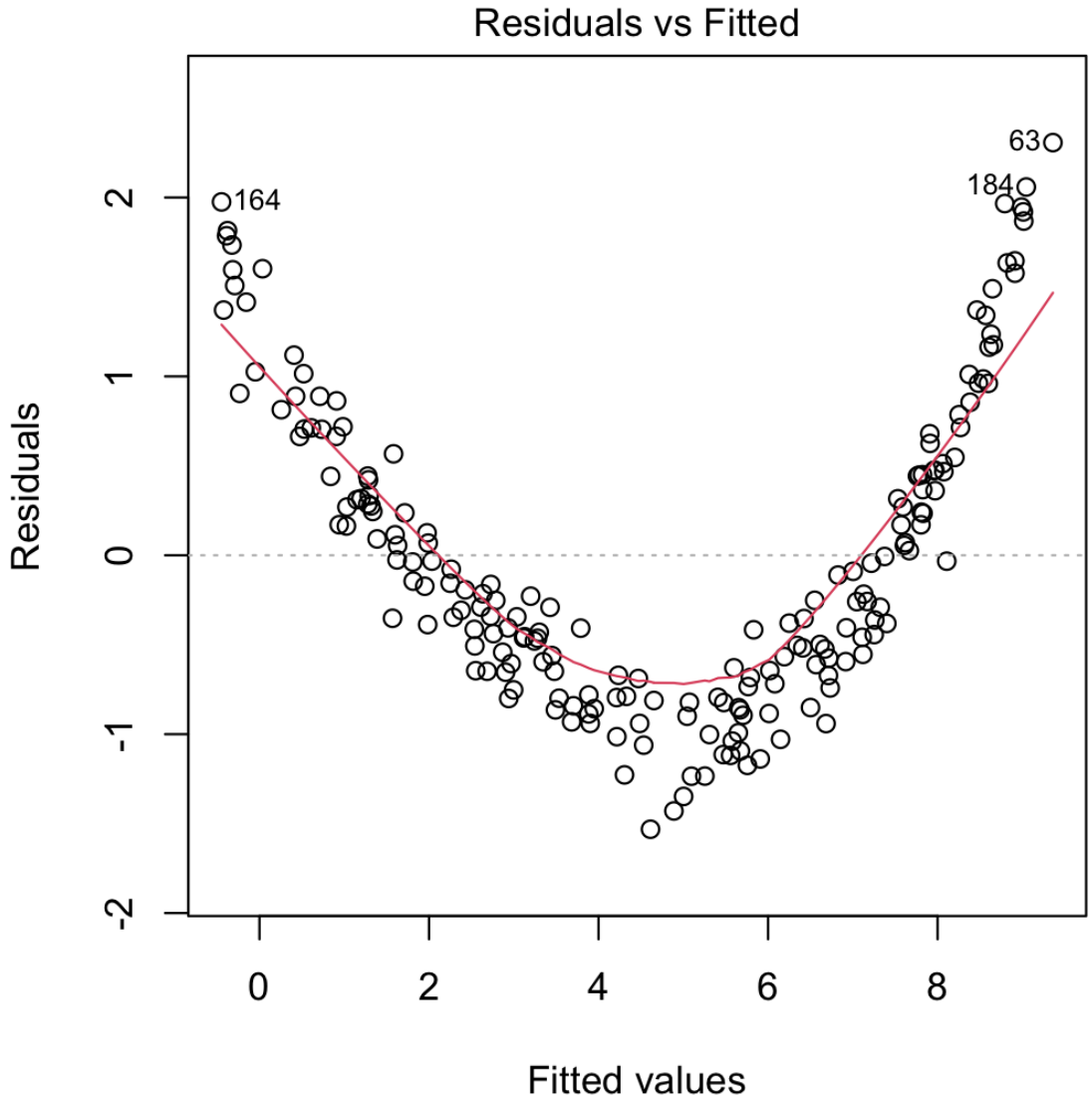
Appears linear, but has another violation which will be discussed later.

Fits versus residuals



Appears linear.

Fits versus residuals



Clearly violates linearity.

Linear model assumptions

Assumptions

Linearity of the data is an obvious assumption for a linear model. There are two other assumptions we require:

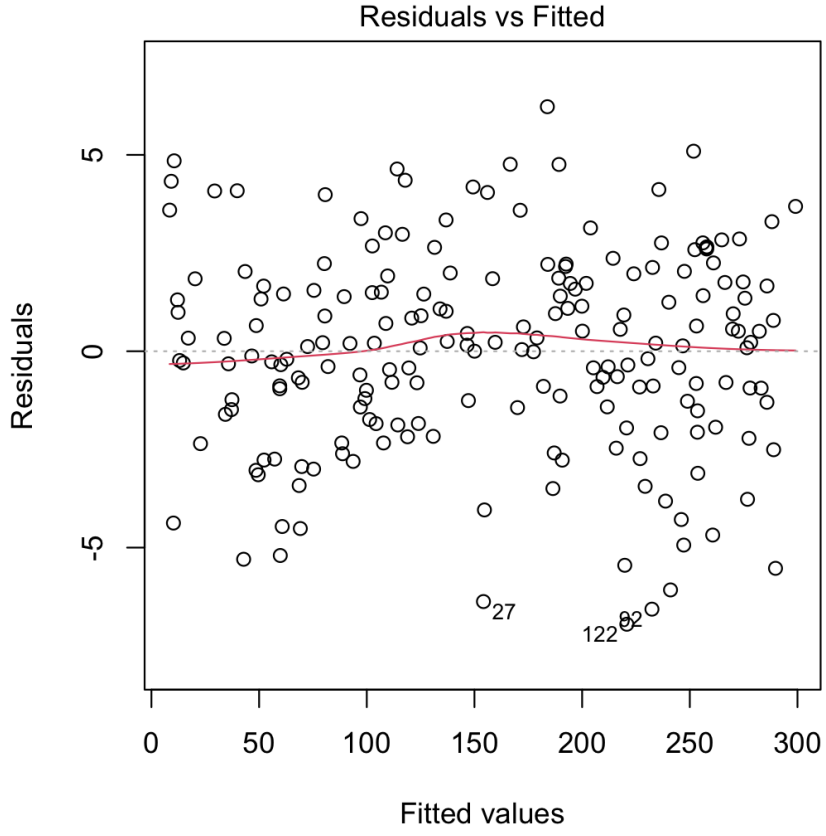
- Constant error variance, known as **homoscedasticity**.
- Normality of the error term.

Homoscedasticity is required to ensure that least squares regression results in the *best linear unbiased estimator*.

Normality of the error term is required to ensure hypothesis testing procedures are valid.

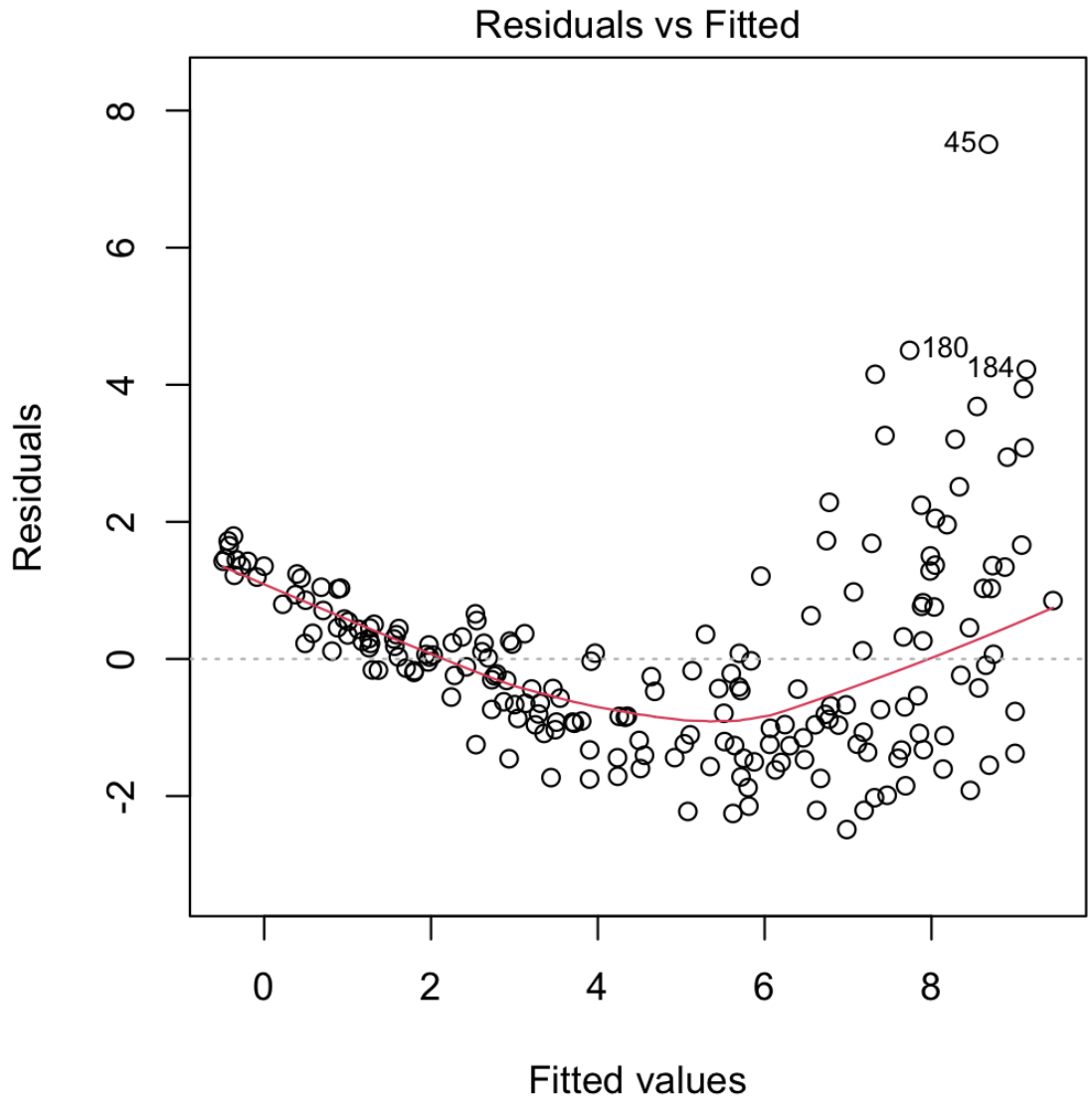
Verifying homoscedasticity

Fits versus residuals plots can be used to verify homoscedasticity.



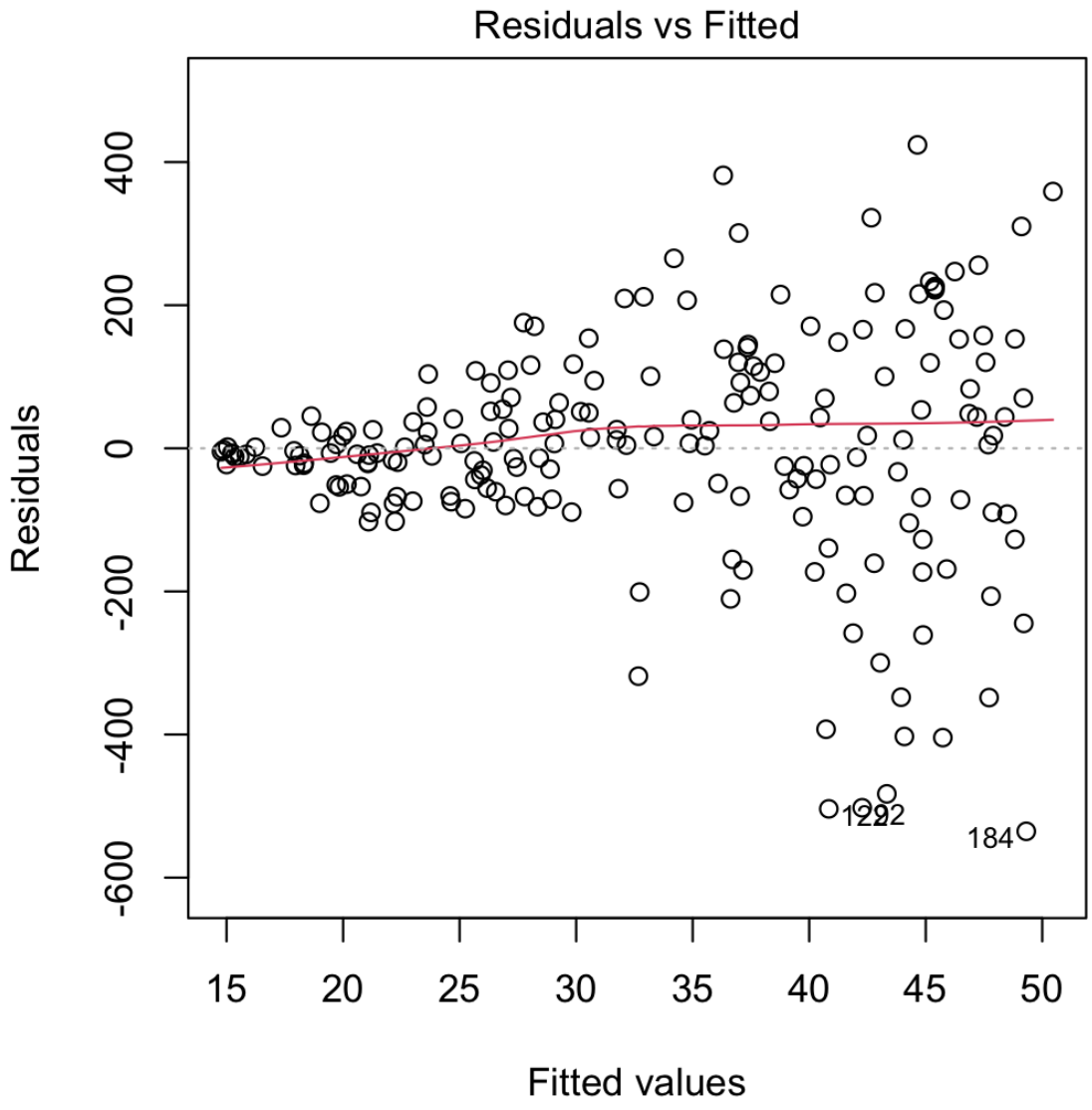
Data with homoscedasticity will appear to have constant variance in the fits versus residuals plot.

Fits versus residuals



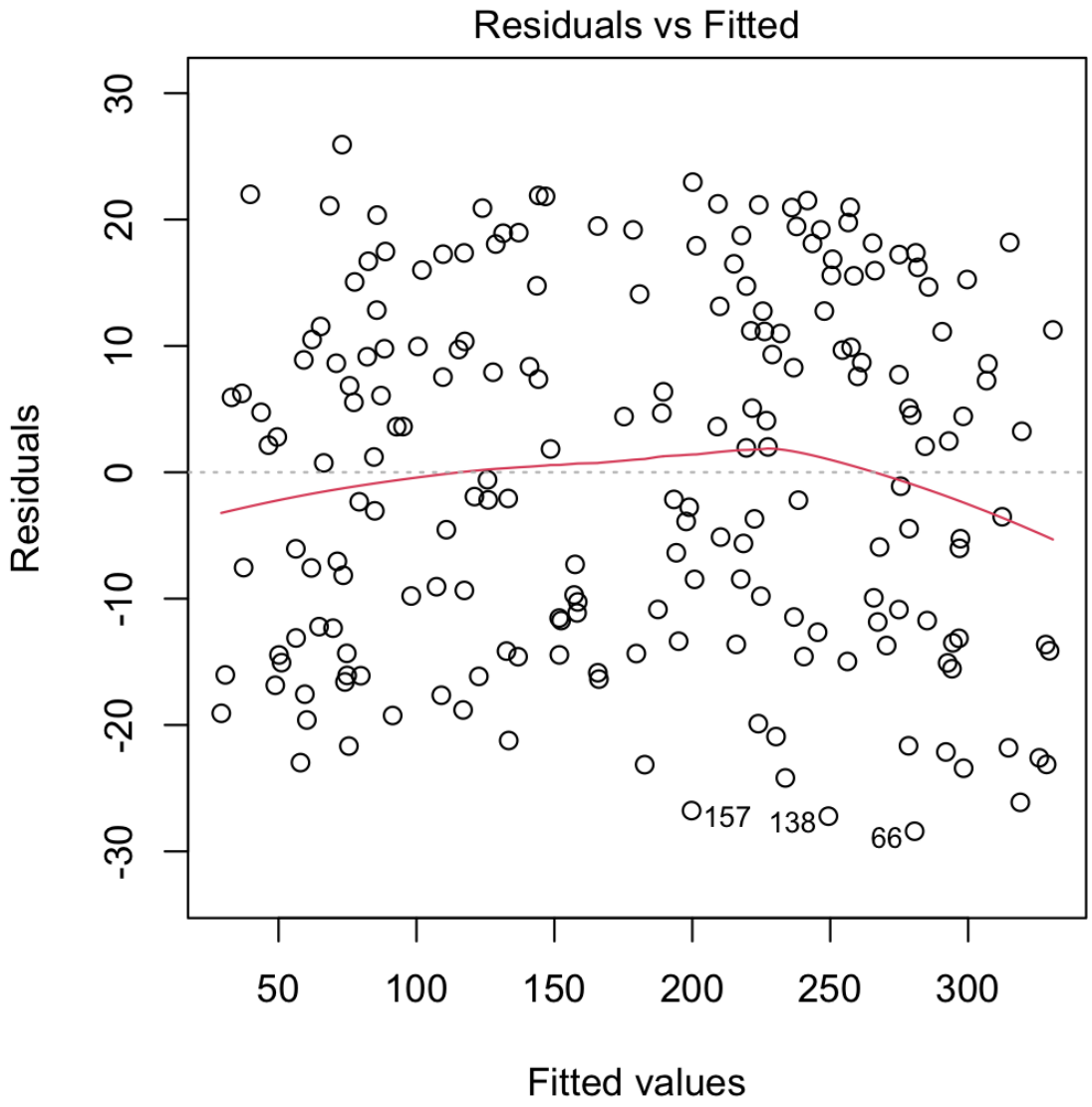
Clearly violates linearity. Also appears to violate homoscedasticity.

Fits versus residuals



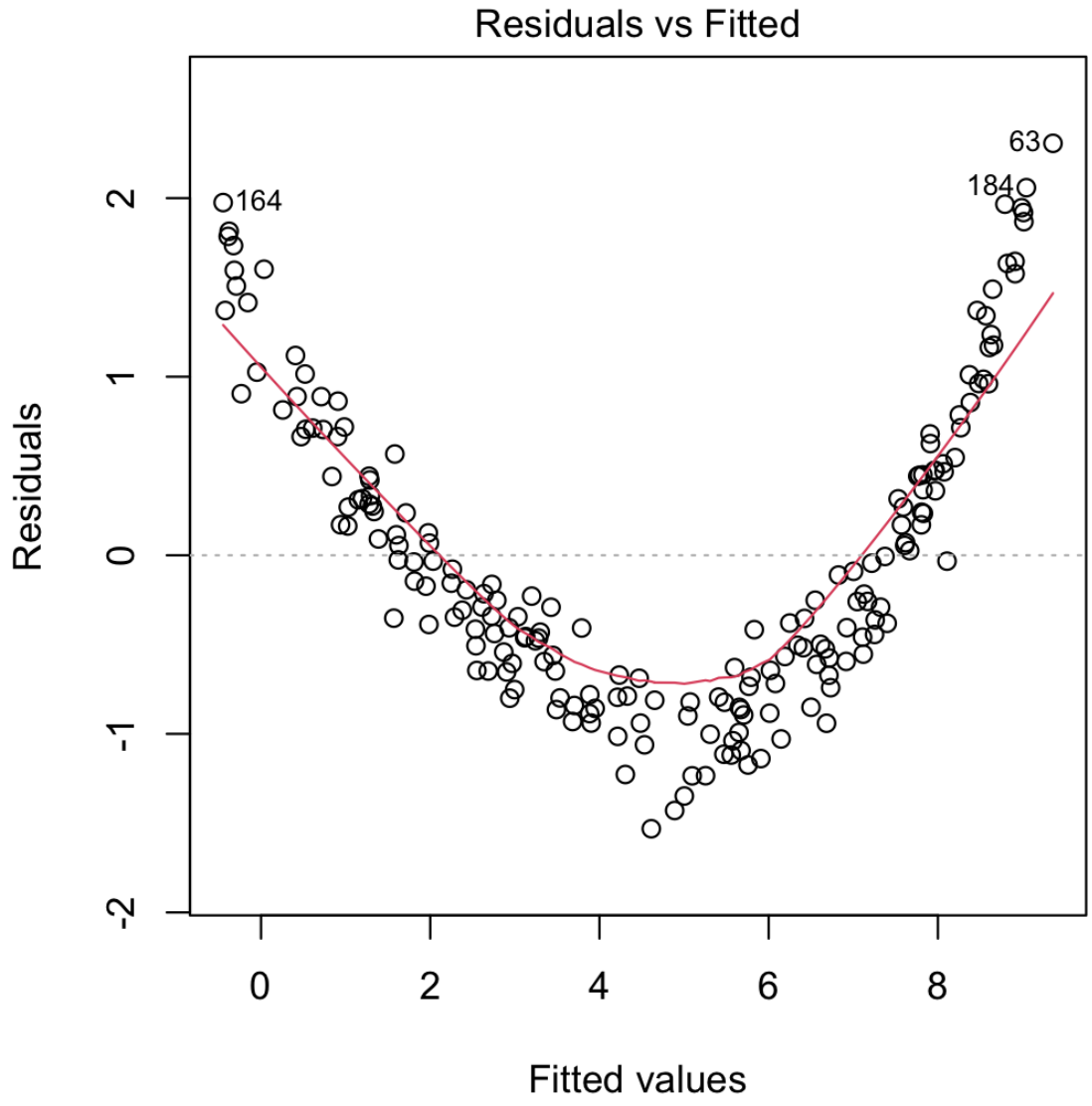
Appears linear, but appears to violate homoscedasticity.

Fits versus residuals



Appears linear and appears homoscedastic.

Fits versus residuals



Clearly violates linearity, but appears homoscedastic.

Verifying normality

A **Q-Q plot** is a plot that compares given data with quantiles of a known distribution. Since we assume that the error terms are normally distributed, we use a **normal Q-Q plot**.

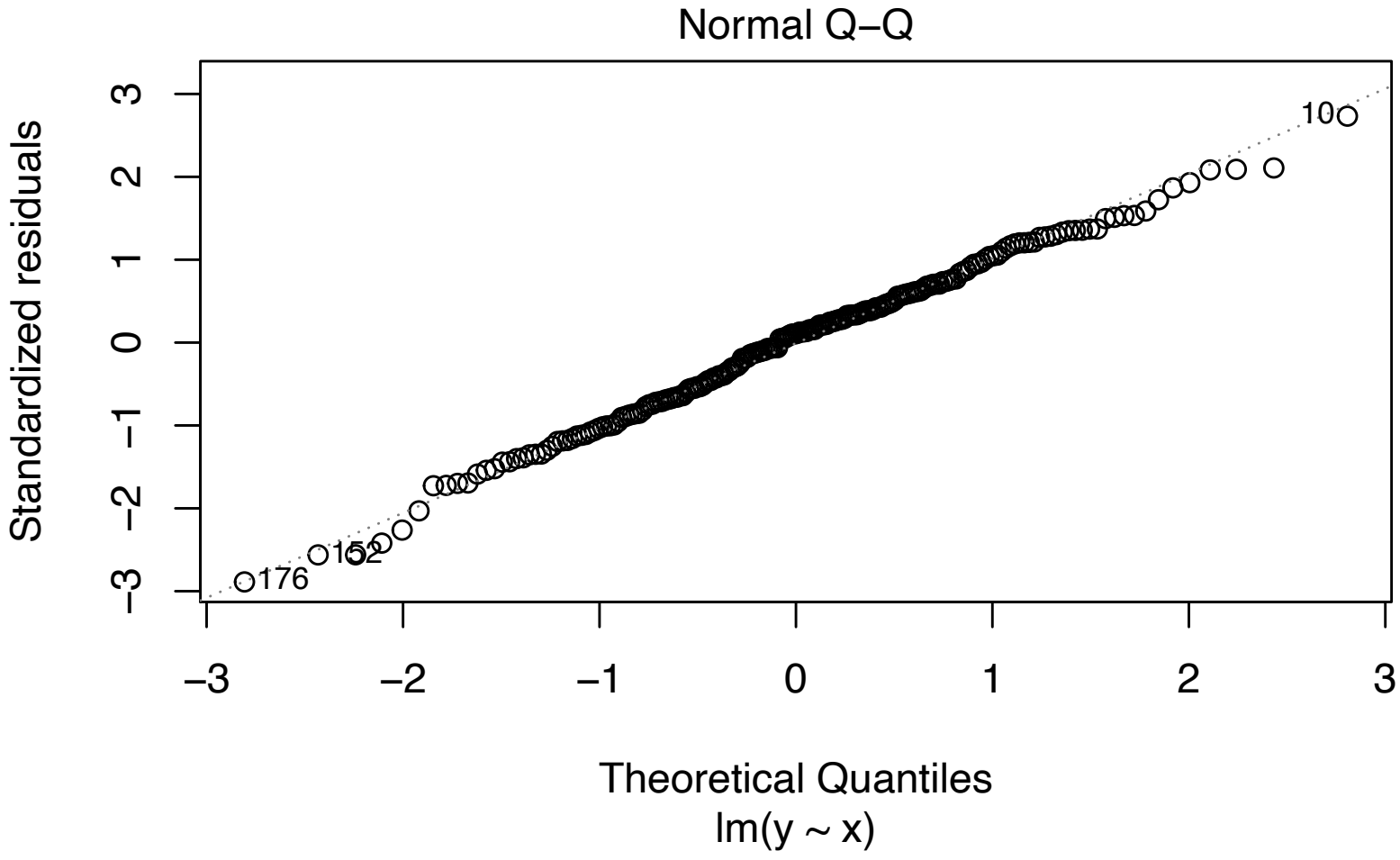
We will not describe the process in full detail, but a rough outline is:

- Standardise the residuals by dividing by an estimate of their standard deviation.
- Order the standardised residuals from smallest to largest.
  - Call these $r_1, r_2, \ldots, r_n$.
- Compute n quantiles from a $N(0, 1)$ distribution using equally spaced values over the interval $[0, 1]$.
  - Call these $z_1, z_2, \ldots, z_n$.
- Plot the pairs (r_i, z_i) for each i.

If the error term is normally distributed, then the Q-Q plot should look approximately linear.

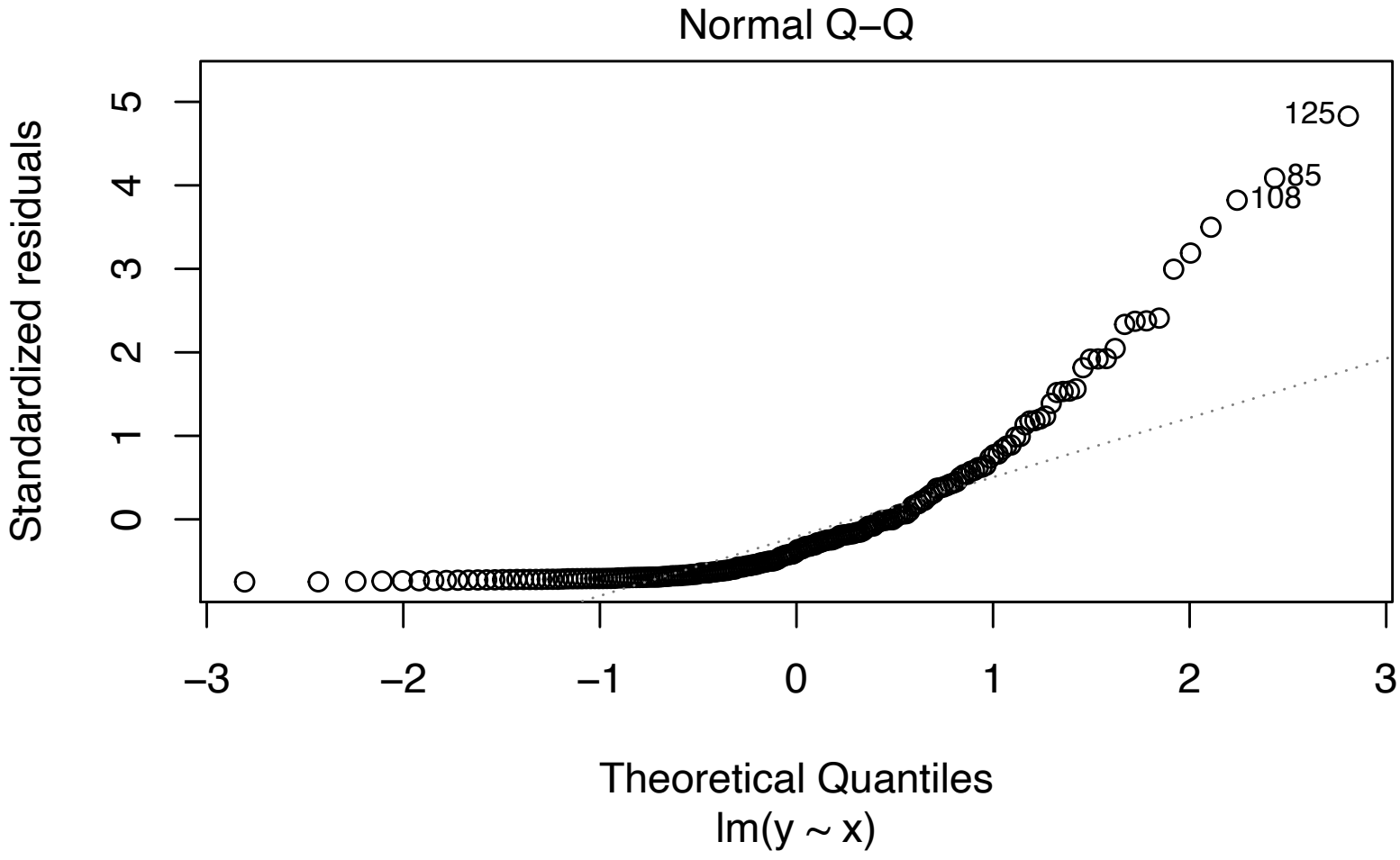
The plot is easily produced using R.

Q-Q plots



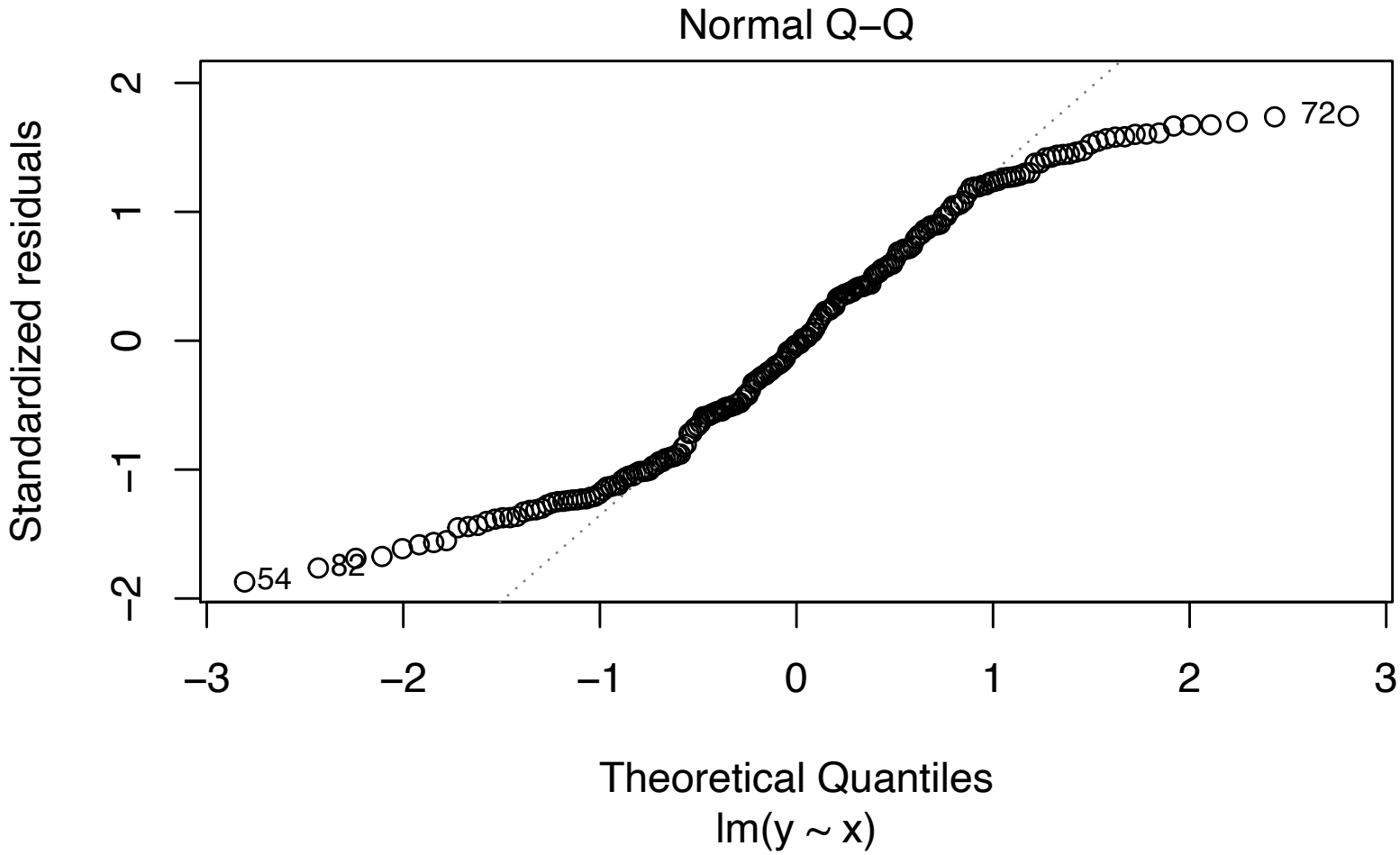
Appears linear, suggesting normality.

Q-Q plots



Clearly non-linear, suggesting non-normality.

Q-Q plots



Linearity deviates at the extremes, indicating a distribution similar to a normal distribution but with heavier tails.

Linear regression in R

Simple linear regression in R

There are two ways to create the model in R.

- If the observations are stored in vectors, then you could do it in the following manner:

```
# assumes you have already defined two vectors:
#   yield, which contains the crop yield
#   fert, which contains the fertiliser used
model <- lm(yield ~ fert)
summary(model)
```

- If the observations are stored in a data frame, then the data frame must be passed as an argument, and the column names are used in the formula.

```
# assumes you have a data frame named Observations,
# with two columns, Output and Input
model <- lm(Output ~ Input, data=Observations)
summary(model)
```

Interpreting a model

The R output could look something like this:

```
Residuals:
     Min       1Q   Median       3Q      Max
-2.31067 -0.93972  0.07314  0.92095  2.11057

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.4167     1.5884   0.892  0.39847
x             3.1514     0.7559   4.169  0.00313 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.407 on 8 degrees of freedom
Multiple R-squared:  0.6848,    Adjusted R-squared:  0.6454
F-statistic: 17.38 on 1 and 8 DF,  p-value: 0.003125
```

- The coefficients are in the ‘estimate’ column.
- The *R-squared* tells us how well the model fits the data.
- Other parts of the output will be discussed momentarily.

Interpretation

From the previous slide, estimates are $\beta_0 = 1.4167$ and $\beta_1 = 3.1514$. The R^2 value is 0.6848.

- $\beta_0 = 1.42$ means that, when the explanatory variable is zero, the estimated response is 1.42.
- $\beta_1 = 3.15$ means that, on average, Y will increase by 3.15 when x increases by 1.

The value R^2 is called the **coefficient of determination**. It is restricted to the interval $[0, 1]$, and can be interpreted in the following manner:

- $R^2 = 1$ is a 'perfect fit', meaning all observed values y_i fall on the estimated line.
- $R^2 = 0$ means no fit whatsoever; there is no correlation between response and explanatory variables. As bad a fit as one can get.
- Closer to 1 means a better fit; closer to 0 means a worse fit.

Low values of R^2 do not always mean that the model is useless. It may just mean that there is a lot of randomness or uncertainty to account for.

Example

A company wants to determine how effective it is to increase the number of staff working on site. To do so, they have observed 57 different days with varying numbers of staff working on those days. The effectiveness was measured by revenue in dollars, representing the amount of monetary worth attained by the day's production.

The data is contained in the file `staff.csv`.

This is a simulated data set, for explanatory purposes.

Hypothesis testing and confidence intervals

Predicting

Suppose, for example, that your linear model had the formula

$$\widehat{y} = 1.4167 + 3.1514x$$

Then, for example, when $x = 4$ we have $y = 14.0223$.

Is this useful for prediction? *Only partially*.

Recall that the original model has some randomness involved:

$$Y = \beta_0 + \beta_1 x + \varepsilon.$$

Using the estimates alone does not account for the error term ε.

Therefore, the model $\widehat{y}$ is best thought of as an estimate of $E(Y)$.

Predicting

Assume the error term ε is normally distributed. Given a particular value of x and a linear model $\widehat{y} = \beta_0 + \beta_1 x$, we can form two different intervals:

- a confidence interval for $E(Y)$, and
- a prediction interval for Y

They are also based on a t-distribution, and very much depend on the normality assumptions described earlier. These are computed in R using the `predict` function.

```
new.data <- data.frame(x = 3) # predict using x = 3
predict(model, new.data, interval="confidence")
predict(model, new.data, interval="prediction")
```

The main difference is that the prediction interval will be wider than the confidence interval, because the prediction interval must account for individual variation.

Hypothesis testing the coefficients

If these model assumptions are satisfied, then we can perform hypothesis testing on the coefficient of the explanatory variable. The following hypotheses are tested:

$$H_0 : \beta_1 = 0 \text{ versus } H_1 : \beta_1 \neq 0.$$

The null hypothesis $\beta_i = 0$ means "there is no linear association between the response and explanatory variables". Using the `lm` command in R produces the p-values for us automatically. They are based on a t-distribution with $n - 2$ degrees of freedom.

Similarly, confidence intervals can be calculated for the coefficients. The intervals for β_0 and β_1 are

$$\widehat{\beta}_j \pm t_{n-2,0.975} \times \text{SE},$$

where SE is an estimate of the standard error for that coefficient. We will not calculate SE ourselves; it is done for us in R.

Multiple regression in R

Multiple linear regression

The techniques considered so far also apply when we have more than one explanatory variable. The multiple linear regression model takes on the form

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \varepsilon,$$

based on explanatory variables $x_1, x_2, \ldots, x_p$.

In that case, there is one important difference, when computing confidence intervals for the coefficients. Specifically, the degrees of freedom is different: $n - p - 1$ instead of $n - 2$, like so:

$$\widehat{\beta}_i \pm t_{n-p-1,0.975} \times \text{SE}.$$

We will demonstrate this using the file `cube.csv`, adopted from here: `https://www.youtube.com/watch?v=U2jr880LBHg`