

# Linear regression with **R**

Paul Hewson

December 3, 2009

## Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Regression in R</b>                             | <b>1</b> |
| <b>2</b> | <b>The linear model</b>                            | <b>1</b> |
| <b>3</b> | <b>Context</b>                                     | <b>2</b> |
| <b>4</b> | <b>Getting started in R and loading the data</b>   | <b>2</b> |
| <b>5</b> | <b>How to present the independent variables</b>    | <b>4</b> |
| <b>6</b> | <b>Fitting the model</b>                           | <b>5</b> |
| 6.1      | Tasks . . . . .                                    | 5        |
| <b>7</b> | <b>Extracting particular pieces of information</b> | <b>6</b> |
| <b>8</b> | <b>Diagnostics</b>                                 | <b>6</b> |

## 1 Regression in R

R is an integrated package for data manipulation and statistical analysis, and includes a flexible object-oriented language (called **S**). It is widely used, and through add-in packages offers a huge range of tools for data analysis.

To find out more about **R** you might like to read the ‘Introduction to R’, which is now included on the website (<http://users.aims.ac.za/ben/R-intro.pdf>), although you will learn many useful functions as we progress through these examples.

## 2 The linear model

We are fitting a model to explain the relationship between two variables  $x$  and  $y$ . The model takes the form:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

This model relies on the assumption that the error terms are independent and identically distributed according to a Normal distribution with mean zero and variance  $\sigma^2$ :

$$\epsilon_i \sim \text{Normal}(0, \sigma^2).$$

Given an actual dataset  $(\mathbf{x}, \mathbf{y})$ , we can form estimates of  $\boldsymbol{\beta} = (\beta_0, \beta_1)$  using the method of least squares.

We will formalise our ideas about this model next week.

- we shall consider that the dependent/response variable  $y$  can be thought of in terms of a conditional distribution:

$$y_i \sim \text{Normal}(\beta_0 + \beta_1 x_i, \sigma^2)$$

- We shall also consider the distribution of the parameters  $\boldsymbol{\beta}$  under various modelling assumptions.

For now, we shall motivate our interest in the linear model by examining a few applications.

### 3 Context

Consider example 25 from Bolstad (2007). The scenario is that we have data from a food manufacturing company concerning moisture levels of food. There is an in-process measurement ( $x$ ) and a final level ( $y$ ). It is cheaper to measure moisture in-process, and so the company wishes to know whether the in-process measurement can give an adequate description of the final values.

### 4 Getting started in R and loading the data

You can run R interactively from the terminal within `gedit` (`textedit`), just as you do with Python. To begin, open `gedit` and create a new file to save your commands.

We start by loading the data:

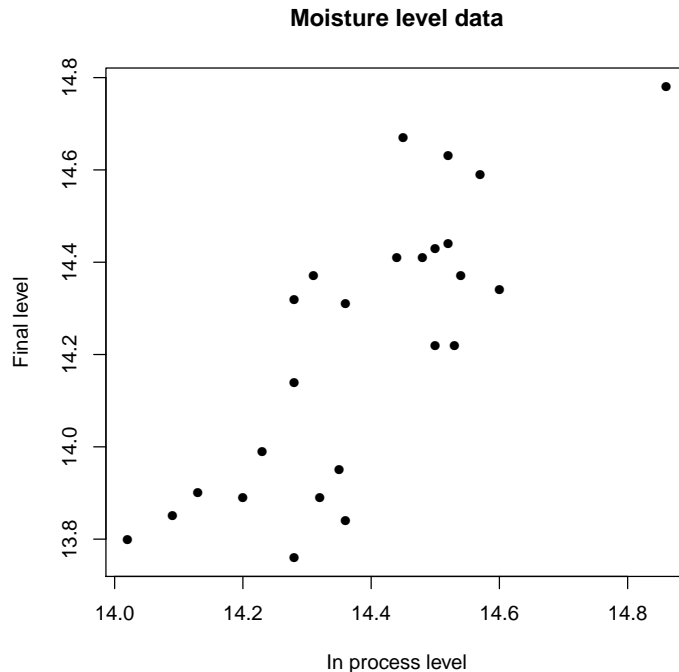
```
> x <- c(14.36, 14.48, 14.53, 14.52, 14.35, 14.31, 14.44, 14.23, 14.32, 14.57, 14.28,
+       14.36, 14.5, 14.52, 14.28, 14.13, 14.54, 14.6, 14.86, 14.28, 14.09, 14.2, 14.5,
+       14.02, 14.45)
> y <- c(13.84, 14.41, 14.22, 14.63, 13.95, 14.37, 14.41, 13.99, 13.89, 14.59, 14.32,
+       14.31, 14.43, 14.44, 14.14, 13.9, 14.37, 14.34, 14.78, 13.76, 13.85, 13.89, 14.22,
+       13.8, 14.67)
```

Note that in many cases we will want to read in data from a text file or database (which R allows) but because the dataset here is small we use the

function `c()` to concatenate a series of values and assign them to our variables `x` and `y`.

Then we need to calculate some summary statistics, and in this case to view the scatterplot:

```
> plot(y ~ x, main = "Moisture level data", xlab = "In process level", ylab = "Final level",  
+       pch = 16)  
> mean(x)  
[1] 14.3888  
  
> mean(y)  
[1] 14.2208  
  
> cor(x, y)  
[1] 0.7956151  
  
> sd(x)  
[1] 0.1847142  
  
> sd(y)  
[1] 0.30173  
  
> cov(x, y)  
[1] 0.04434267
```



Note that in most cases R commands have sensible defaults set, so that we could have typed `plot(x,y)` to see a scatter plot without the labels, and without changing the symbol used in the plot (e.g. `pch=16`).

## 5 How to present the independent variables

One aspect of linear modelling that takes some thought is how to present the independent (predictor) variables to the model. In the case of quantitative variables we have to decide:

- Whether it would be better to center the predictor variables (i.e., to subtract the mean). This will change the interpretation of the intercept.
- On occasion, we might need to decide whether to apply a suitable transformation (logarithms, square roots etc.)

The **R** command `scale(object, center=TRUE, scale=TRUE)` lets us center (remove the mean) and scale (divide by the standard deviation) a variable (type `?scale` for help on the function). In this case we set `scale=FALSE` as we only wish to subtract the mean, creating a new variable `x.c`:

```
> x.c <- scale(x, scale = FALSE)
```

## 6 Fitting the model

We can fit a linear regression in **R** using the `lm()` function. This creates an object (which we here call `model1`). This object has class `lm`. Because **R** is (rather loosely<sup>1</sup>) object oriented in nature, this object has a number of associated methods, such as `summary()`, `coef()` and others illustrated below.

```
> model1 <- lm(y ~ x.c)
> summary(model1)
```

Call:

```
lm(formula = y ~ x.c)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-0.34337 -0.14532  0.01753  0.12266  0.36966
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept) 14.22080    0.03734 380.805 < 2e-16 ***
x.c          1.29963    0.20634   6.298 1.99e-06 ***
```

---

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.1867 on 23 degrees of freedom

Multiple R-squared: 0.633, Adjusted R-squared: 0.617

F-statistic: 39.67 on 1 and 23 DF, p-value: 1.994e-06

### 6.1 Tasks

Can you interpret the output from the summary of the `lm` object? In particular, can you see:

- The slope coefficient (the change in moisture level in process for a unit change in moisture measured at the end)?
- The significance of the overall model fit (the F test)?
- The significance of the individual parameter ( $\beta_1$ )?
- The proportion of variation in the data explained by the model (the  $R^2$ )?
- Do you think this is a ‘good’ model?

---

<sup>1</sup>Here, the model object is of S3 type, there is also a newer format called S4

## 7 Extracting particular pieces of information

We can extract a number of other features from the `lm` object. First, we can extract the coefficients alone, and can extract (classical/frequentist) confidence intervals for these values:

```
> coef(model1)

(Intercept)          x.c
  14.220800      1.299635

> confint(model1, level = 0.95)

                2.5 %    97.5 %
(Intercept) 14.143548 14.298052
x.c          0.872786  1.726483
```

## 8 Diagnostics

The first “diagnostic” we need to do is to check that the residuals are “normally distributed”. Note below that we use two extractor functions, applied to the `lm` object:

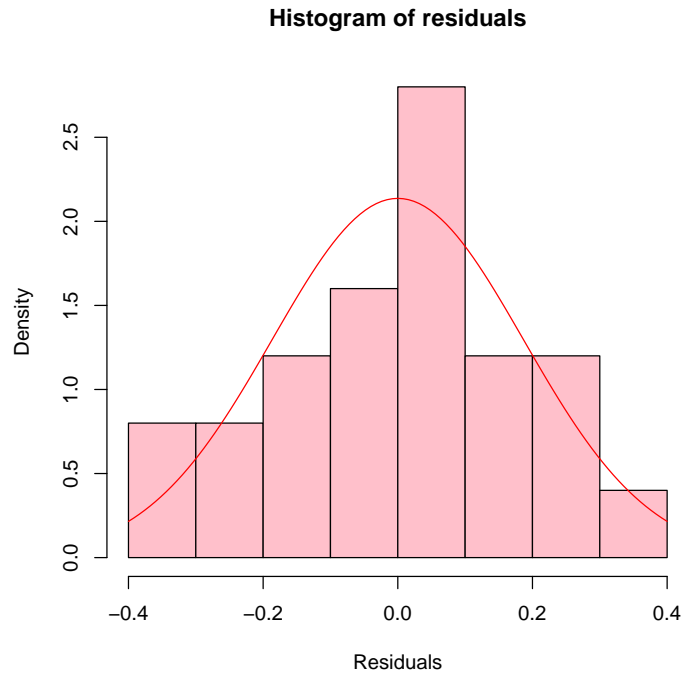
- `residuals(model1)` extracts the residuals (which could be stored in another vector object if we wishes)
- `fitted(model1)` extracts the fitted values, i.e.,  $\hat{y}_i = \beta_0 + \beta_1 x_i$

We extract the residuals in order to plot a histogram (which should have a classical Gaussian bell-shaped curve), centred on zero. The variance around the least squares fit line is given by:

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum_{i=1}^n \hat{y}_i$$

and the square root of this value is used to superimpose a  $Normal(0, \sigma^2)$  model curve over the histogram.

```
> hist(residuals(model1), freq = FALSE, main = "Histogram of residuals", xlab = "Residuals",
+      col = "pink")
> sigma2hat <- sum((fitted(model1) - y)^2)/model1$df
> curve(dnorm(x, 0, sqrt(sigma2hat)), add = TRUE, col = "red")
```

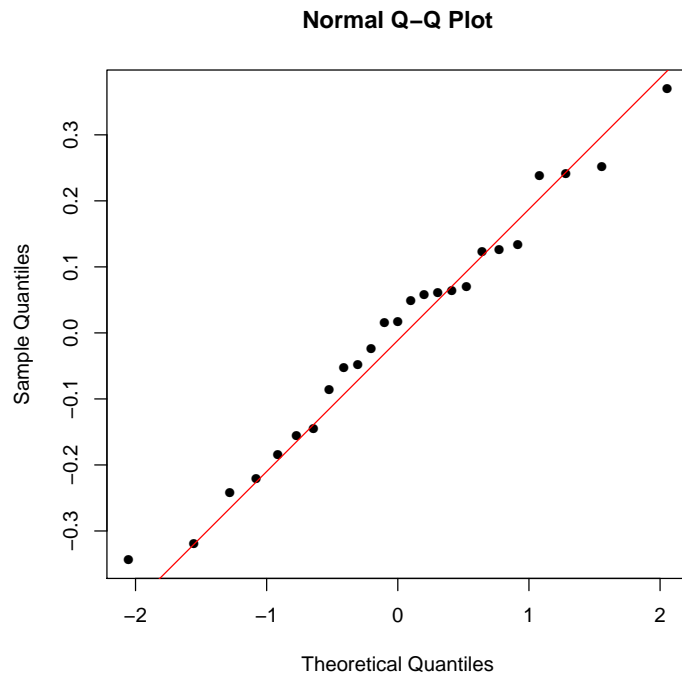


Sometimes it's easier to make this comparison by using a qq plot. This plots the observed quantiles<sup>2</sup> of the residuals against the expected quantiles assuming a Normal distribution (for example we expect to see 1 observation in 40 more than 2 standard deviations from the mean). The hope is that if our residuals are Normally distributed, the qq plot will be a straight line.

```
> qqnorm(residuals(modell), pch = 16)
> qqline(residuals(modell), col = "red")
```

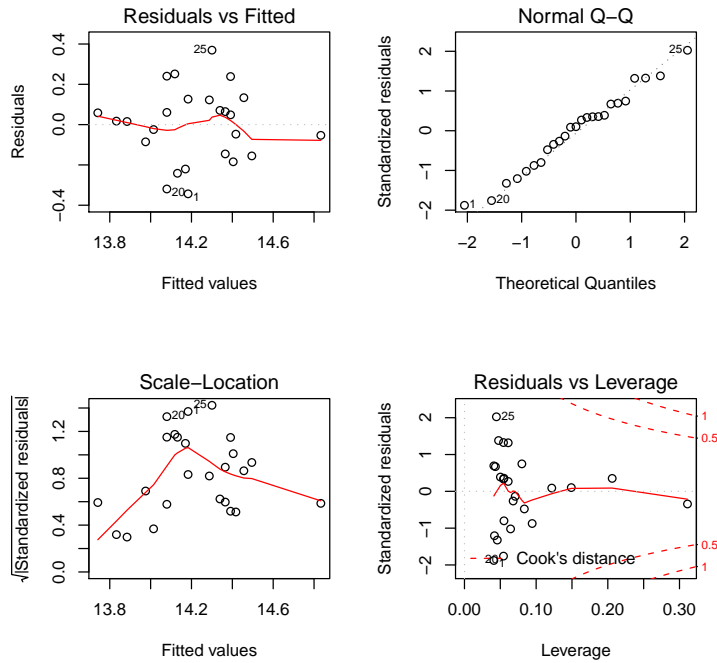
---

<sup>2</sup>Quantiles are points taken at regular intervals from the cumulative distribution function (CDF) of a random variable. Dividing ordered data into q essentially equal-sized data subsets, the quantiles are the data values marking the boundaries between consecutive subsets.



It turns out that a number of useful diagnostics are wrapped up in a single `plot()` method for the `lm` object

```
> par(mfrow = c(2, 2))  
> plot(model1)
```



For example, plotting the residuals against the fitted values should yield a nice random spread. Particular patterns seen here can help inform the way we should include additional variables in the model. We are most concerned that the variation around 0 should be constant along the whole range.