STATISTICAL MODELLING Part IIC

Practical 3: Introduction to linear models IAC/Lent 2011

Comments and corrections to ioana@statslab.cam.ac.uk

Start R and change the current working directory to `U:\Rwork`. Next, download the `WeldData` from my web page

`http://www.statslab.cam.ac.uk/~ioana/teaching`

Save the file `weld.txt` in the current working directory (`U:\Rwork`). These data come from The Welding Institute, Abingdon. The first column is the current used in Amps, and the second is the minimum diameter of the weld in millimetres.

Read in the data with with R

```
> WeldData <- read.table("weld.txt", header = TRUE)
> t(WeldData)
```

```
        [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
curr    7.82  8.0 7.95 8.07 8.08 8.01 8.33 8.34 8.32  8.64  8.61  8.57
mindiam 3.40  3.5 3.30 3.90 3.90 4.10 4.60 4.30 4.50  4.90  4.90  5.10
        [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
curr     9.01  8.97  9.05  9.23  9.24  9.24  9.61   9.6  9.61
mindiam  5.50  5.50  5.60  5.90  5.80  6.10  6.30   6.4  6.20
```

The 'header' option allows each column to have headings located on the first line of the data file. When no header is present, R gives its own labels for rows and columns. These are not really part of the `WeldData` matrix, as you can see from

```
> dim(WeldData)
```

```
[1] 21  2
```

```
> x <- WeldData[, 1]
> y <- WeldData[, 2]
> rbind(x, y)
```

```
  [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
x 7.82   8.0 7.95 8.07 8.08 8.01 8.33 8.34 8.32  8.64  8.61  8.57  9.01
y 3.40   3.5 3.30 3.90 3.90 4.10 4.60 4.30 4.50  4.90  4.90  5.10  5.50
  [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21]
x  8.97  9.05  9.23  9.24  9.24  9.61   9.6  9.61
y  5.50  5.60  5.90  5.80  6.10  6.30   6.4  6.20
```

Exercise: Plot the data.

As an initial model, we assume that the $n = 21$ observed values $y_1, \ldots, y_n$ are realisations of independent random variables $Y_1, \ldots, Y_n$ from the model $Y_i = \alpha + \beta(x_i - \bar{x}) + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$ for $i = 1, \ldots, n$.

```
> w <- x - mean(x)
> LinMod1 <- lm(y ~ w)
```

Fitting this model rather than $Y_i = \alpha + \beta x_i + \epsilon_i$ ensures that α and β are orthogonal (what does this mean in terms of the joint distribution of the MLEs of these parameters?) and in this case makes the intercept term more stable and interpretable. The function `lm` is one of the most important that we will meet in this course. Notice how the intercept term is automatically included in the model; it could be excluded with `lm(y~w-1)`, though we certainly don't want to do that here. The output of the function is stored as an `"lm"-class` object by R. Various functions access different information stored in the object, perhaps the most important of which is:

```
> summary(LinMod1)


Call:
lm(formula = y ~ w)

Residuals:
     Min       1Q   Median       3Q      Max 
-0.42623 -0.07282  0.01637  0.08269  0.34586 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  4.93810    0.04391  112.46  < 2e-16 ***
w            1.65793    0.07531   22.01 5.53e-15 ***
---
```

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

Residual standard error: 0.2012 on 19 degrees of freedom
Multiple R-squared: 0.9623,       Adjusted R-squared: 0.9603
F-statistic: 484.6 on 1 and 19 DF,  p-value: 5.529e-15
```

After repeating the formula used in the model, and giving summary statistics on the residuals, we see the estimates $\hat{\alpha}$ and $\hat{\beta}$, together with their standard errors – how are these computed? To answer this question, you will first have to work out the formula used to compute the residual standard error appearing lower in the summary. Since $\epsilon_i \sim N(0, \sigma^2)$, the residual standard error is an estimator of σ. Is this done via the MLE or the unbiased estimator of σ^2?

Returning to the coefficients in the summary, the t-statistic for testing the null hypothesis that the parameter is zero is just the ratio of the parameter estimate to its standard error (why?), and the p-value for this t-statistic is given as the final column of numbers. The stars give a quick visual way of assessing which parameter estimates are significantly different from zero. What do the multiple R^2 and adjusted R^2 mean? The F-statistic is for testing the null hypothesis $H_0 : \beta = 0$. Compare the F–statistic with that from `anova`:

```
> anova(LinMod1)
```

```
Analysis of Variance Table

Response: y
          Df  Sum Sq Mean Sq F value    Pr(>F)
w          1 19.6203 19.6203  484.61 5.529e-15 ***
Residuals 19  0.7692  0.0405
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

What do the entries in the `anova` table give? Try also

```
> coef(LinMod1)
```

```
(Intercept)           w
   4.938095    1.657925
```

```
> coef(LinMod1)[1]
```

```
(Intercept)
   4.938095
```

```
> coef(LinMod1)[[1]]
```

```
[1] 4.938095
```

The function `names(LinMod1)` gives alternative ways of accessing this information, e.g.,

```
> LinMod1$coefficients
```

```
(Intercept)           w
   4.938095    1.657925
```

together with other features with which we will be less concerned.

Exercise: How are the fitted residuals related to the fitted values?

```
> resid(LinMod1)
```

```
          1           2           3           4           5           6
-0.11070064 -0.30912716 -0.42623091 -0.02518192 -0.04176118  0.27429358
          7           8           9          10          11          12
 0.24375754 -0.07282171  0.16033679  0.02980074  0.07953850  0.34585550
         13          14          15          16          17          18
 0.01636844  0.08268545  0.05005144  0.05162491 -0.06495434  0.23504566
         19          20          21
-0.17838665 -0.06180739 -0.27838665
```

```
> fitted(LinMod1)
```

```
       1        2        3        4        5        6        7        8
3.510701 3.809127 3.726231 3.925182 3.941761 3.825706 4.356242 4.372822
       9       10       11       12       13       14       15       16
4.339663 4.870199 4.820462 4.754144 5.483632 5.417315 5.549949 5.848375
      17       18       19       20       21
5.864954 5.864954 6.478387 6.461807 6.478387
```

Figure 1 plots y against w, and adds the fitted line using the code below.

```
> plot(w, y, main = "MLE fit of y~w")
> abline(LinMod1, lwd = 2)
> xp <- predict.lm(LinMod1, se.fit = TRUE, interval = "prediction")
> lines(w, xp$fit[, 1], col = 2, lty = 2, lwd = 2)
> lines(w, xp$fit[, 2], col = 3, lty = 3, lwd = 2)
> lines(w, xp$fit[, 3], col = 3, lty = 3, lwd = 2)
```

Look at other options for the `abline()` function. The `predict()` function with argument `se.fit=TRUE` gives the fitted values and standard errors at the input locations (explanatory variables) `w`. For for more information, try

```
?predict.lm
```

Exercise: Explain what the three `lines()` functions are illustrating in the plot. How would you use `predict.lm()` to obtain fitted values at new input locations? In particular, how would you predict the response at a new data point $w = 0.4$, and its prediction interval?

Figure 2 shows some very useful residual plots obtained by calling the `plot()` function with the linear model object as an argument. The statistics behind each of the four plots are discussed in the lecture notes. Notice the quadratic trend in the (top–left) plot of the residuals against the fitted values and the heavy tails in the fitted residuals. This might suggest fitting another linear model with a quadratic term:

```
> LinMod2 <- lm(y ~ w + I(w^2))
```

(In R formulae, the operators '+', '−', '∗' and '^' have a special interpretation – the function `I()` is necessary here to convince R to interpret the square in the normal arithmetic sense.) Write down the model that is being fitted and look at the summary statistics. Notice that the estimates of the intercept and linear terms have changed – why?

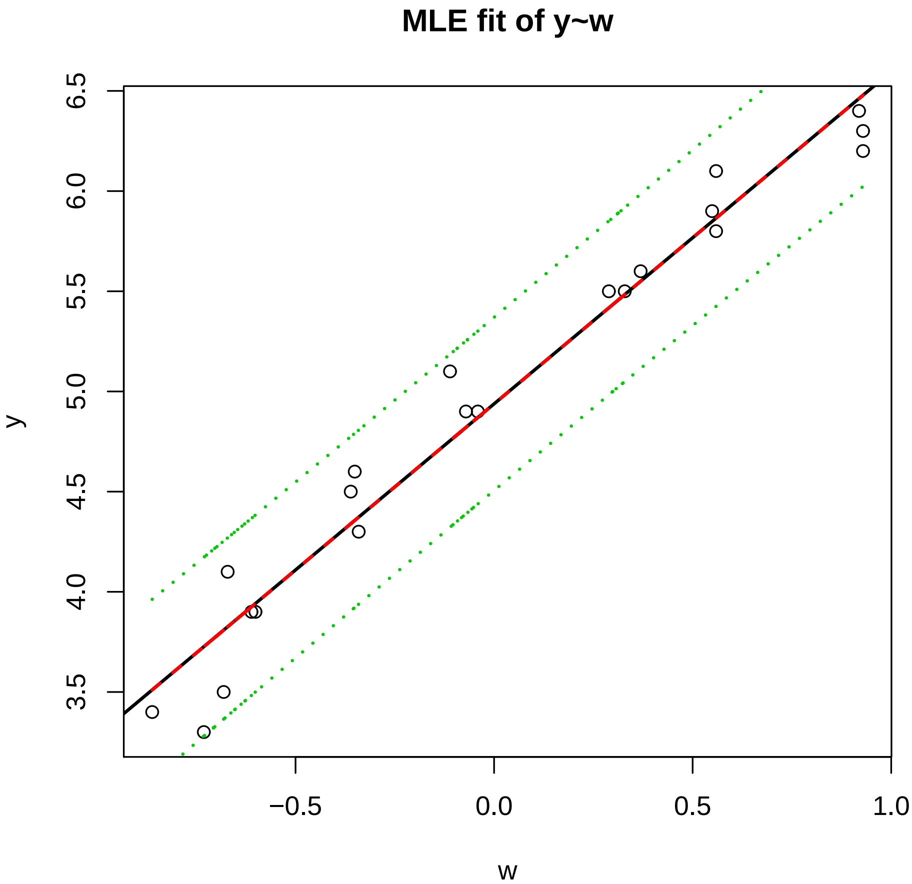

Figure 1: First linear model fit to weld data.

Exercise: Make a plot analogous to the one in Figure 1 using the MLEs obtained in the second model (`LinMod2`). Why does `abline(LinMod2)` not do the right thing in this case? Examine the residual plots for the second linear model. Is there an improvement?

Since one model is nested inside the other, we can compare them with

```
> anova(LinMod1, LinMod2)
```

```
Analysis of Variance Table

Model 1: y ~ w
Model 2: y ~ w + I(w^2)
```

```
> par(mfrow = c(2, 2))
> plot(LinMod1)
```

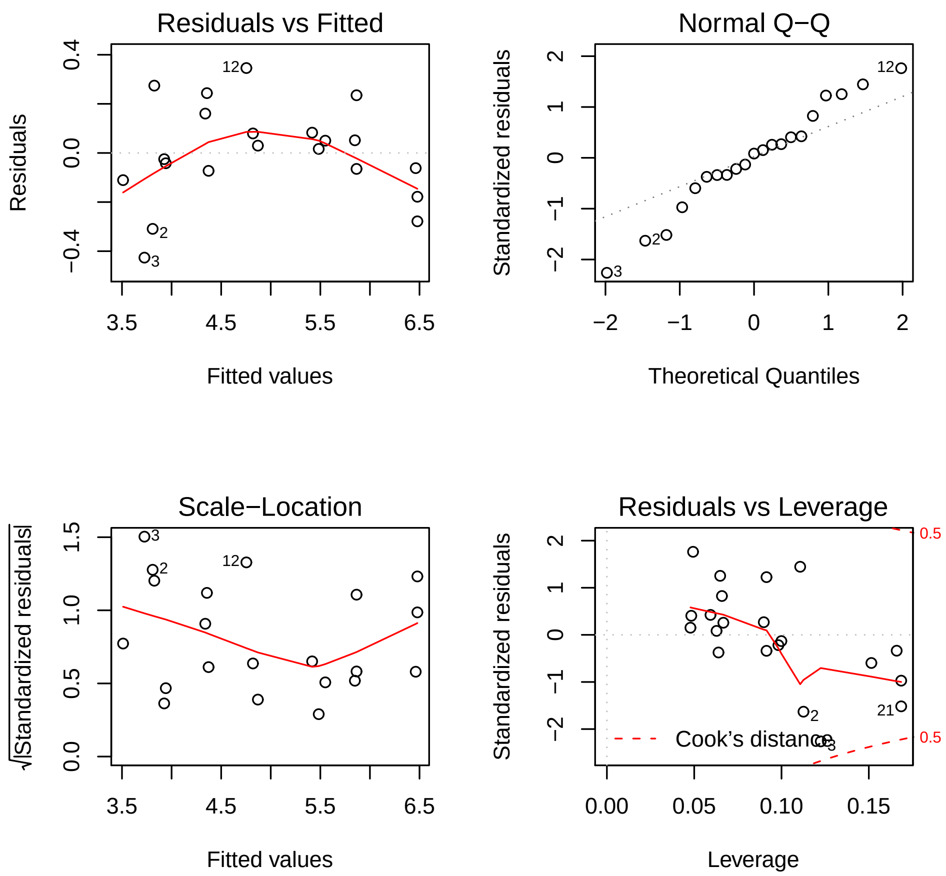

Figure 2: Four residual plots for the first linear model fit to the weld data.

```
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1     19 0.76924
2     18 0.48637  1   0.28287 10.469 0.004589 **
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1
```

Exercise: What are the entries in this table? Explain which model you prefer, and why.