STATISTICAL MODELLING Part IIC

Practical 5: One- and two-way ANOVA IAC/Lent 2011

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

In this practical we will deal with the `PotashData`. It consists of the strengths of different bundles of cotton grown under five different treatments, i.e. levels of potash, a fertiliser. For each treatment level, there are three replications. The treatment levels (in lbs of potash per acre) were 36, 54, 72, 108, 144 respectively. Download the data `potash.txt` from the course webpage and store it in your `U:\Rwork` directory.

Rather than use `read.table`, it will be easiest to have the data in the form of a vector:

```
> PotashData <- scan("potash.txt", comment.char = "#")
```

Create a vector containing the amount of potash used for each observation as follows:

```
> x <- c(rep(36, 3), rep(54, 3), rep(72, 3), rep(108, 3),
+     rep(144, 3))
> x
```

```
 [1]  36  36  36  54  54  54  72  72  72 108 108 108 144 144 144
```

The function `tapply` is similar to `apply`, and can be used to give the mean strength for each treatment level:

```
> tapply(PotashData, x, mean)
```

```
      36       54       72      108      144
7.850000 8.053333 7.743333 7.513333 7.450000
```

```
> tapply(PotashData, x, mean)[1]
```

```
  36
7.85
```

```
> tapply(PotashData, x, mean)[[1]]
```

```
[1] 7.85
```

In other words, the function `mean` is applied to each group of values from `PotashData` given by a unique value in vector `x`. Figure 1 shows two different ways of plotting this

```
> par(mfrow = c(1, 2))
> plot(x, PotashData, xlab = "potash levels", ylab = "strengths",
+     main = "cotton strength v. treatment level")
> plot(as.factor(x), PotashData, xlab = "potash levels",
+     ylab = "strengths", main = "cotton strength v. treatment level")
```

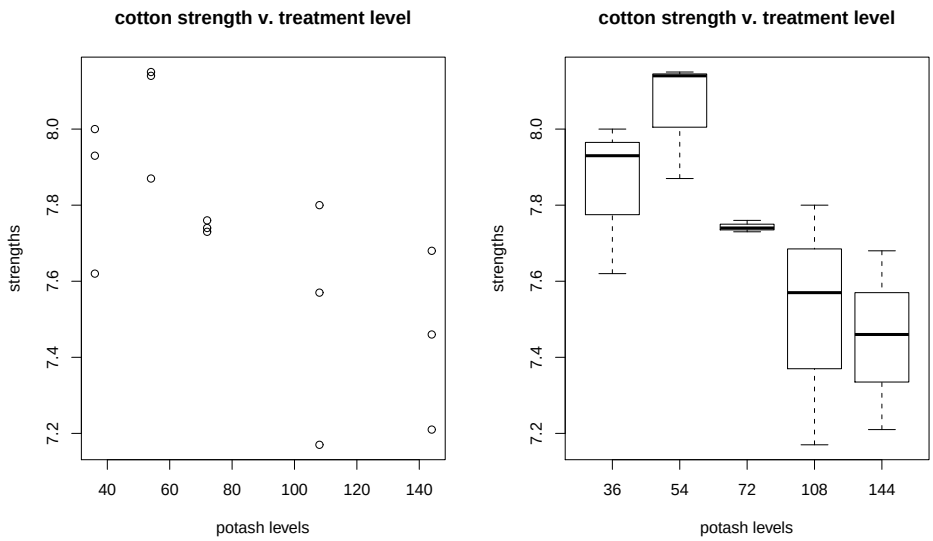

Figure 1: Plotting the potash data.

data. The first (*left*) is a plot of the data in its raw form. The *right–hand* plot is perhaps more useful when there are many replications in the data. This box plot is obtained by default when the x–argument in the plot is coerced to be `factor`. What values does the box plot display for each factor value?

Notice that `as.factor(x)` treats the potash levels as class labels rather than as numerical values of a quantitative explanatory variable. This distinction gives two different ways of modelling the data.

```
> LinMod1 <- lm(PotashData ~ x)
> summary(LinMod1)


Call:
lm(formula = PotashData ~ x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.42572 -0.04112 -0.01612  0.12448  0.28368

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  8.136925   0.132184  61.558  < 2e-16 ***
x           -0.005011   0.001446  -3.466  0.00418 **
---
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: 0.2171 on 13 degrees of freedom
Multiple R-squared: 0.4803,        Adjusted R-squared: 0.4403
F-statistic: 12.01 on 1 and 13 DF,  p-value: 0.004177
```

Exercise: Write down the standard linear regression model that is being fitted here. Plot the data and add the fitted line. Also examine the residual plots (by plotting the fitted models).

For the second model, we consider the potash level as a factor:

```
> Factor.x <- as.factor(x)
> LinMod2 <- lm(PotashData ~ Factor.x)
> summary(LinMod2)


Call:
lm(formula = PotashData ~ Factor.x)

Residuals:
     Min       1Q   Median       3Q      Max
-0.34333 -0.09833  0.01667  0.09167  0.28667

Coefficients:
```

```
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.8500     0.1220  64.338 2.00e-14 ***
Factor.x54    0.2033     0.1725   1.178   0.2659
Factor.x72   -0.1067     0.1725  -0.618   0.5503
Factor.x108  -0.3367     0.1725  -1.951   0.0796 .
Factor.x144  -0.4000     0.1725  -2.318   0.0429 *
---
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: 0.2113 on 10 degrees of freedom
Multiple R-squared: 0.6212,        Adjusted R-squared: 0.4697
F-statistic:   4.1 on 4 and 10 DF,  p-value: 0.03202
```

This fits the one-way ANOVA model $Y_{ij} = \mu + \alpha_i + \epsilon_{ij}$, where $\epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$. What is the default identifiability constraint which is being imposed? What is the *baseline level* for this model?

Exercise: In the residual plots for the second (ANOVA) model, unlike for the first model, the last graph informs us that the leverage is constant – why is this so? To see how R constructs the design matrix, try

```
X <- model.matrix(PotashData~Factor.x)
```

Exercise: The p-values of the coefficients in `LinMod2` suggest the possibility of fitting a reduced model, with factor levels `"Low"` and `"High"` for the amount of potash used. Where would be an appropriate cut-off point? Experiment with one or two reduced models. Test the reduced model against `LinMod2`.

First, define a new variable `Factor.x.2` with level `Low` if $x \leq 54$ and `High` if $x \geq 72$.

```
> Factor.x.2 <- Factor.x
> levels(Factor.x.2) <- c("low", "low", "high", "high",
+     "high")
> LinMod3 <- lm(PotashData ~ Factor.x.2)
> summary(LinMod3)


Call:
lm(formula = PotashData ~ Factor.x.2)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.39889 -0.09528  0.04833  0.17972  0.23111

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)      7.9517     0.0914  87.001   <2e-16 ***
Factor.x.2high  -0.3828     0.1180  -3.244   0.0064 **
---
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: 0.2239 on 13 degrees of freedom
Multiple R-squared: 0.4474,        Adjusted R-squared: 0.4049
F-statistic: 10.52 on 1 and 13 DF,  p-value: 0.006401
```

```
> rss0 <- sum(LinMod3$residuals^2)
> rss1 <- sum(LinMod2$residuals^2)
> F <- ((rss0 - rss1)/(5 - 2)/(rss1/(15 - 5)))
> (pval <- 1 - pf(F, 3, 10))
```

```
[1] 0.2666148
```

Exercise: What do you conclude? How can you perform this test with a single R command? Try a different cut-off point.

Next, we will work with the `SexistData`, available on the course webpage. The data are the results of a survey in which the percentages of people having equal confidence in both sexes for various occupations were recorded. The occupations, in order, were bus/train driver, surgeon, barrister, and MP. Save the file as `sexist.txt`, and read in the file with

```
> SexistData <- read.table("sexist.txt", header = TRUE)
```

Create vectors consisting of the four jobs and the 12 countries with

```
> job <- names(SexistData)[1:4]
> country <- SexistData[, 5]
```

We want to create two factor vectors: one for the jobs to which each figure refers, and one for the countries. We can do this with the 'generate level' function `gl()`. As with the `PotashData` it is helpful to have the `SexistData` in vector form.

```
> Job <- gl(4, 1, 48, labels = job)
> Country <- gl(12, 4, 48, labels = country)
> Sexist <- as.vector(t(SexistData[, 1:4]))
```

We now plot the data to get a first impression: Make sure that you understand how

```
> par(mfrow = c(1, 2))
> plot(Job, Sexist)
> plot(Country, Sexist)
```

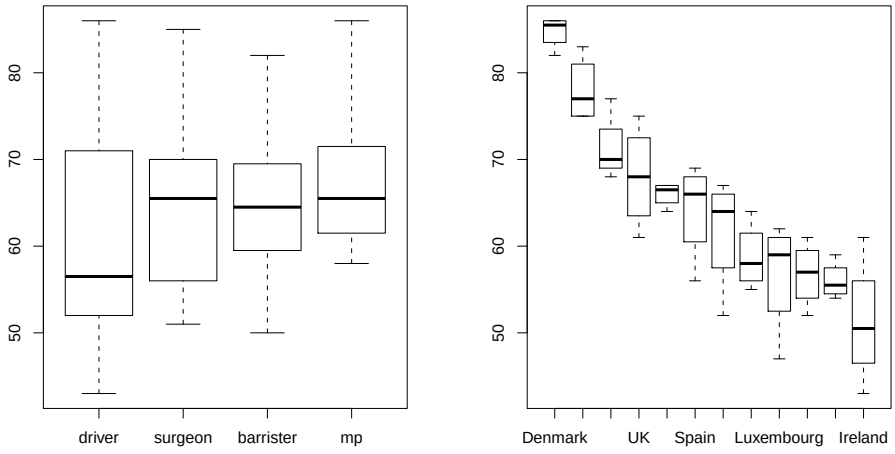

Figure 2: Plots of sexist data as a function of occupation (*left*) and country (*right*).

the `gl()` function works (above) by examining the code and resulting plots in Figure 2. Couple this with the mean (marginal) percentages across occupations and countries:

```
> tapply(Sexist, Job, mean)
```

```
   driver   surgeon barrister        mp
 60.58333  65.16667  64.66667  67.41667
```

```
> tapply(Sexist, Country, mean)

    Denmark Netherlands      France          UK     Belgium
      84.75       78.00       71.25       68.00       66.00
      Spain    Portugal   W.Germany  Luxembourg      Greece
      64.25       61.75       58.75       56.75       56.75
      Italy     Ireland
      56.00       51.25
```

Exercise: Write down the models being fitted below, and study the output (including the residual plots) carefully. Which variables does R use as the baseline? What do the coefficient estimates represent?

```
> LinMod1.sexist <- lm(Sexist ~ Job + Country)
> LinMod2.sexist <- lm(Sexist ~ Job)
> LinMod3.sexist <- lm(Sexist ~ Country)
```

Compare the following

```
> anova(LinMod1.sexist)
> anova(LinMod2.sexist, LinMod1.sexist)
> anova(LinMod3.sexist, LinMod1.sexist)
```

Exercise: The `TouristData` may be studied in much the same way as `SexistData`. The data is located on the course web page in a file called `tourist.txt`. It gives a table of prices, in pounds, for a three-course meal, bottle of beer, suntan lotion, taxi (5km), film (24 exposures) and car hire (per week) in 14 different popular tourist destinations. Plot a graph of the mean price for each item against the variance of the price for that item. Explain why we should take logarithms of the prices. Fit an additive two-way ANOVA model. Which is the most expensive resort? What happens if we remove the final column (car hire)?