ECOLOGICAL AND ENVIRONMENTAL ANALYSIS

Doing Basic Statistics and Data Visualisation in R

Dr. Kyle Dexter, kyle.dexter@ed.ac.uk

Exercise 1: Basic ANOVA

Workers at a tree farm wish to examine the effectiveness of three fertiliser mixtures on the growth of Norway maple, *Acer platanoides*. Seedlings were grown in soil treated with one of the three fertilisers for one year, and then the heights of the seedlings were measured (in cm). The workers wish to know if seedling height is different between the three treatments. The data are shown below.

Fertiliser A	Fertiliser B	Fertiliser C
115	115	155
120	145	75
135	75	110
155	60	145
160	95	85
170	95	105
175	170	140
200	105	75
205	130	140
220	120	110

Firstly, enter these data as a two-column data frame, with one column representing the fertiliser category and the other the numeric values for seedling height. Then combine these two columns into a data frame using the data.frame() function (as seen in R_Basics.pdf file).

The data can be entered as follows:

```
> A <- c(115,120,135,155,160,170,175,200,205,220)
> B <- c(115,145,75,60,95,95,170,105,130,120)
> C <- c(155,75,110,145,85,105,140,75,140,110)
> height <- c(A,B,C)
> fertiliser <- c(rep("A",length(A)),rep("B",length(B)),rep("C",length(C)))
> alldata <- data.frame(fertiliser,height)
```

Now, let's conduct an analysis of variance on these data to assess if different fertilisers lead to significantly different seedling heights. To do this, we must first build a statistical model, in this case a linear model. We can then execute an analysis of variance on that model. The two pertinent functions here are lm() and anova(). Also, check out using the summary function, summary(), on your statistical model object.

To build the statistical model:

```
> (height_lm <- lm(height~fertiliser,data=alldata))
```

To execute the analysis of variance (ANOVA):

```
> anova(height_lm)
```

And to use the summary function:

```
> summary(height_lm)
```

Notice the different output you get from just the statistical model, from the anova() function, and from the summary() function. The statistical model simply gives you the formula used for the model and the estimated coefficients from the model. The anova() function gives you an

ANOVA table, with all its constituent components, including the results of an F test. The summary function gives you all of the above and more. It first gives you the formula, followed by some summaries of the distribution of residuals. Then you get a table for the coefficients, which includes the estimated value of the coefficients, the standard errors around those estimates, a t-value for how far each coefficient departs from zero, and a test for significance. Don't worry too much about those significance tests right now. Lastly, at the bottom, one gets information about the residual standard error (which essentially measures how much of the variation in your data is not explained by your statistical model), degrees of freedom (here, total number of data points – number of categories), R^2 values, and the main results of an ANOVA, which tests whether fertiliser type significantly affects seedling height.

Next you want to determine if the data satisfy the assumptions necessary to conduct an ANOVA. You will need to use the function resid() to obtain the residuals, while the function shapiro.test() can be used to test the normality of the residuals, and the function bartlett.test() can be used to check equality of variances.

This can actually be executed in three lines of code as follows:

```
> height_resids <- resid(height_lm)
> shapiro.test(height_resids)
> bartlett.test(height~fertiliser,data=alldata)
```

Now use the versatile plot() function on your linear model object. This is a quick and handy way to see how well your model fits your data. Hit Return in the R Console window while keeping the plot window open to advance through the plots.

```
> plot(height_lm)
```

The first plot you see is the residual versus fitted plot. This lets you readily assess if you have constant variances. If your dependent variable responds to your independent variable in a non-linear manner, that will also be evident here (particularly relevant for continuous predictor variables). Notice that R gives the row numbers or names of the biggest outliers.

The next plot you see is the normal Q-Q plot. This allows you to assess if your residuals are normally distributed. If the points are close to the dashed line, then the residuals approximate a normal distribution. In our plot, the tails of the residual distribution deviate slightly from the normal line. This is a common feature of small datasets, and our Shapiro test above showed that our residuals are close enough to a normal distribution.

The third plot is also primarily intended to identify non-constant variance, or heteroscedasticity.

The last plot, to compare leverage versus residual values is not very relevant in this context because we do not have continuous predictor variables. Rather, with only categorical predictors, it represents just another version of the first plot, but with the residual values standardised to a mean of 0 and a standard deviation of 1. Check this one out again when we do linear regression below.

Now that we have convinced ourselves that our model fit the data well, i.e. that we are not violating any of the assumptions of an ANOVA, based on our test, what can we conclude about the effect of the three fertiliser treatments on seedling height?

Fertiliser significantly affects seedling height. Technically, we cannot conclude that fertiliser A leads to significantly taller seedlings than fertilisers B and C.

We may be interested in statistically assessing which fertiliser(s) are driving our significant ANOVA result. To do this, we can perform a Tukey's test using the TukeyHSD() function in

R. One wrinkle here is that the TukeyHSD() function is looking for a certain type of object upon which to act, an object of class ‘aov’. We can create this object using the aov() function. It is essentially conducting an ANOVA in one step, rather than the two steps we used above.

```
> (height_aov <- aov(height~fertiliser,data=alldata))
> TukeyHSD(height_aov)
```

This test shows us that A and B have significantly different heights and A and C have significantly different heights, while B and C do not have significantly different heights. In other words, A differs significantly from B and C.

And just for fun, if you wanted to do an ANOVA by hand, it actually isn’t too tricky in R, following the equations in the lecture on linear models.

```
> total_seedlings <- length(A)+length(B)+length(C)
> number_categories <- 3
> overall_mean <- (sum(A)+sum(B)+sum(C))/total_seedlings

> (SSwithin <- sum((mean(A)-A)^2)+sum((mean(B)-B)^2)+sum((mean(C)-C)^2))
> (DFwithin <- total_seedlings -number_categories)
> (MSwithin <- SSwithin/DFwithin)

> sumA <- length(A)*((mean(A)-overall_mean)^2)
> sumB <- length(B)*((mean(B)-overall_mean)^2)
> sumC <- length(C)*((mean(C)-overall_mean)^2)
> (SSamong <- sumA+sumB+sumC)
> (DFamong <- number_categories-1)
> (MSamong <- SSamong/DFamong)

> (Fstatistic <- MSamong/MSwithin)

> pf(Fstatistic, DFamong, DFwithin, lower.tail=FALSE)
```

Exercise 2: More on Analyses with Single Categorical Explanatory Variable and Continuous Response

In the western lowland Amazon, there are two main habitat types: bottomland, floodplain forest that receives annual floods and upland, ‘terra firme’ forests that are never flooded. The annual floods carry sediment that is eroded from the Andes and is high in clay content and plant nutrients. It is therefore presumed that soils in floodplain and terra firme forest would be different, and this probably has significant impacts on plant and animal communities. In order to provide a preliminary examination of this soil variation, soil samples were collected at 25 sites across the southeastern Amazon of Peru. These soils were analysed for various attributes used to quantify soil variation. Look on LEARN and find the file labelled ‘Peru_Soil_Data.csv’, which gives the data for several soil variables. If you wish to see the raw data, including the units used in measurements, it can be found at: *http://www.esapubs.org/archive/mono/M080/009/suppl-1.htm*

i) Place this file in a folder on your computer. Then open this file in excel to get a feel for what the data look like. Next, set the working directory of R to the designated folder, and read the data file into R.

Use the read.csv() function to read in the data file. Note that the 1st column consists of names for each site, and we will not analyse this variable. Thus, we can set the 1st column as the names for the rows.

```
> soils <- read.csv("Peru_Soil_Data.csv",row.names=1,stringsAsFactors=TRUE)
```

If you haven’t managed to set your working directory properly, you can use the file.choose argument to import the data from wherever you saved it on your computer.

```
> soils <- read.csv(file.choose(), row.names=1,stringsAsFactors=TRUE)
```

Or, if you can’t get that to work either, you can use the import data option from the GUI menus in RStudio, but do be sure to subsequently convert the categorical variables in the data from character to factor.

You can get a quick feel for the data by using the summary function or other functions.

```
> dim(soils)
> names(soils)
> soils$Soil_pH
> soils[,c(4,7)]
> head(soils)
> summary(soils)
```

ii) Explore each variable individually using histograms with the hist() function. Can you identify any outliers in the data? Which data are not normally distributed? Do not transform or alter these data yet, but keep the results in mind for when we examine relationships between variables. Check out help(hist) to learn how you can alter the appearance of histograms. Here some examples for Soil_pH.

```
> hist(soils$Soil_pH)
> hist(soils$Soil_pH,breaks=10)
> hist(soils$Soil_pH,breaks=10,col="grey")
> hist(soils$Soil_pH,breaks=10,col="grey",xlab="Soil pH",main="")
```

iii) To add a line to an existing histogram, use the function abline(). Let’s add the mean and median to soil pH. I want this to be a vertical line at a specific value on the x-axis, so I will use the commands as follows. Check out help(abline). We will explore below how to alter what that line looks like. Note, you must first have a plot open to add a line it.

```
> abline(v=median(soils$Soil_pH))
> abline(v=mean(soils$Soil_pH))
```

iv) Now let’s examine how our continuous data on soil properties varies across our discrete categorical variables. This is most readily done using box plots. Try examining how different variables compare across habitat types.

```
> plot(Soil_pH~Habitat,data=soils)
> plot(Potassium~Habitat,data=soils)
```

v) Use analyses of variance (ANOVAs) to determine if these Soil_pH and Potassium vary significantly with habitat type (floodplain versus terra firme). Assess if the assumptions underlying an analysis of variance are met.
As above, we have to be careful about following the correct notation when conducting linear models or ANOVAs. I have chosen to use the lm() function here as it will be useful later on.

```
> lm_pH <- lm(Soil_pH~Habitat,data=soils)
> anova(lm_pH)
> lm_K <- lm(Potassium~Habitat,data=soils)
> anova(lm_K)
```

In order to check the assumptions, we again extract the residuals, determine if they follow a normal distribution, and determine if the variances for floodplain and terra firme are the same.

```
> lm_pH_resids <- resid(lm_pH)
> shapiro.test(lm_pH_resids)
> bartlett.test(Soil_pH ~Habitat,data=soils)
> lm_K_resids <- resid(lm_K)
> shapiro.test(lm_K_resids)
> bartlett.test(Potassium ~Habitat,data=soils)
```

You can also do this visually using the plot.lm() function

```
> plot(lm_pH)
> plot(lm_K)
```

Potassium clearly does not have homogeneous variances across floodplain and terra firme.

vi) You may have found out that the above assumptions are not met. Visually examine the data using histograms and/or box plots to have a better idea of how the data are distributed and how the assumptions of the ANOVA may have been violated.
In using the histogram function, one can see that the data for potassium are right-skewed. I use the ‘breaks’ argument in histogram in order to break the data into more bins and provide a cleaner view of the data.

```
> hist(soils$Soil_pH,breaks=10)
> hist(soils$Potassium,breaks=10)
```

In order to simultaneously assess the distribution of the data in floodplain and terra firme, one can readily create box plots. One could either use the boxplot() function, or R, being quite smart, will figure out to use the boxplot function if you simply use plot().

```
> plot(Soil_pH~Habitat,data=soils)
> plot(Potassium~Habitat,data=soils)
```

vii) How could we solve the problem that the assumptions of the ANOVA may have been violated? (**HINT:** Log-transforming data often helps when the assumptions are violated). Re-conduct the analyses once you have transformed the data and assess if the data now follow the assumptions of an ANOVA.
I will first create a new variable called log_K, which I can then use in analysis.

```
> soils$log_K <- log(soils$Potassium)
```

I then re-do the analyses above, but with the new variable.

```
> lm_log_K <- lm(log_K ~Habitat,data=soils)
> anova(lm_log_K)
```

And finally, I check to see if the data now conform to the assumptions of an ANOVA.

```
> lm_log_K_resids <- resid(lm_K)
> shapiro.test(lm_log_K_resids)
> bartlett.test(log_K ~Habitat,data=soils)
> plot(lm_log_K)
```

Exercise 3: Relationships between Continuous Variables

The pH of a soil is strongly affected by the amount of base cations in the soil relative to acidic ions. Clays can bind and retain cations in the soil, and thus higher clay content can lead to higher pH. However, soil with a given amount of clay will only have a given amount of binding sites for cations, and not all of those binding sites are necessarily occupied. The base saturation gives a measure of how many binding sites are occupied by base cations. We might expect that soil pH shows a better relationship with base saturation than clay content. This exercise focuses on whether base saturation and clay show relationships with soil pH.

i) The main tool for examining relationships between two continuous variables is 2 dimensional or x-y scatter plots. Some questions to think about when looking at these plots and which may inform your statistical tests: Which relationships appear to be linear? Which relationships may require transformation of one or both variables to become linear? The workhorse function here is the plot(), which can be used in two different ways. I will generally use the second notation as it parallels the notation used in statistical analyses.

```
> plot(soils$Total_Base_Saturation, soils$Soil_pH)
> plot(Soil_pH~Total_Base_Saturation,data=soils)
```

ii) Check out help(plot) to see some options for how you can configure plots. There is actually limited information here. In fact, it is the help file for par(), which is linked to in the help file for plot() that explains many of the options one can use to shape the appearance of plots.

```
> plot(Soil_pH~Total_Base_Saturation,data=soils,pch=21,bg="blue")
> plot(Soil_pH~Total_Base_Saturation,data=soils,pch=21,bg="blue",cex=2)
> plot(Soil_pH~Total_Base_Saturation,data=soils,pch=21,bg="blue",cex=2,cex.lab=1.5)
```

iii) Now let's figure out how to add a best-fit trend line to this plot. The first step here involves a statistical analysis as we have to conduct a linear regression to figure out the parameters (i.e. slope and intercept) that describe the line. That will be covered below.

```
> bestfit <- lm(Soil_pH~Total_Base_Saturation,data=soils)
> abline(bestfit)
```

iv) In order to figure out how to alter the appearance of this line, we will again have to go to help(par) as help(abline) doesn't give too much help. Also, note to add the line with different formatting, we should first recreate the plot from scratch.

```
> plot(Soil_pH~Total_Base_Saturation,data=soils,pch=21,bg="blue",cex=2,cex.lab=1.5)
> abline(bestfit,lty=2,lwd=2,col="red")
```

v) Now, we will consider a linear model between two continuous variables. Assess the relationships between Soil_pH and Clay soil variables using linear regression. Think about which variable should be the explanatory variable and which the response variable. Here, we cannot use a Bartlett test to assess homoscedasticity, but using the plot() function with the linear model object will allow us to assess how well that assumption is met. Again, don't forget to visualise relationships. That will help you interpret the model assessment plots.

```
> plot(Soil_pH~Clay, data=soils)
```

Next, one can use the lm() function along with anova() and summary() to assess the relationships.

```
> pH_vs_Clay <- lm(Soil_pH~Clay,data=soils)
> anova(pH_vs_Clay)
> summary(pH_vs_Clay)
```

Finally, as above, one can extract the residuals and determine if they follow a normal distribution. In this case, heteroscedasticity is better evaluated by ‘plotting’ the linear model.

```
> pH_vs_Clay_resids <- resid(pH_vs_Clay)
> shapiro.test(pH_vs_Clay_resids)
> plot(pH_vs_Clay)
```

vi) Uh-oh. The very last plot indicates that we may have a problem. Blanquillo_floodplain has very high leverage and a high residual. If we plot out Soil_pH versus Clay again, we can figure out what exactly this means. It helps to label the points (using the text() function) and add a line to the plot that represents the linear relationship we have fit (using the abline() function). We see that Blanquillo_floodplain is near the edge of the range of the independent variable (Clay) and is far from the regression line. Once you have visualised this naughty data point, create a modified version of the soils data frame removing the Blanquillo_floodplain site and conduct the statistical analysis, with verification, again.

First up, we plot the data again. We can also add the names of the sites using the text() command, after the plot is already up. After that I add a line, colouring it red, for the model.

```
> plot(Soil_pH~Clay, data=soils)
> text(Soil_pH~Clay, data=soils,labels=rownames(soils))
> abline(pH_vs_Clay,col="red")
```

To remove the offending point, it helps to figure out which row it represents, then we can subtract it from the data frame and do the analysis again as follows:

```
> rownames(soils)
> pH_vs_Clay_new <- lm(Soil_pH~Clay,data=soils[-1,])
> anova(pH_vs_Clay_new)
> plot(pH_vs_Clay_new)
```

Exercise 4: Non-parametric Alternatives to Simple Linear Models

i) Think back to when we examined the distribution of potassium over habitat types. In order to satisfy the assumptions of the statistical test, we had to log-transform potassium. Let’s look at another variable, sodium concentration, and assess how it varies across habitat types.

```
> salt <- lm(Sodium~ Habitat,data=soils)
> summary(salt)
> plot(salt)
```

ii) According to our linear model, sodium does not differ significantly between the two habitat types. However, when we use the plot function on that linear model, we see that there are some ‘issues’ here. We might try log-transforming the response variable to see if that helps.

```
> lm_log_salt <- lm(log(Sodium)~ Habitat,data=soils)
> summary(lm_log_salt)
> plot(lm_log_salt)
```

iii) We now have a significant result! But, we still have some issues of non-normality, heteroscedasticity and an outlier. These are difficult data! In such times, we might resort to a non-parametric statistical test. Let’s try using a Kruskal-Wallis test.

```
> kruskal.test(Sodium~ Habitat,data=soils)
```

We obtain the value for the test-statistic (the Kruskal-Wallis chi-squared) and the probability of that test statistic under our null hypothesis that habitat does not affect sodium concentrations in soils. In this case, the p-value is very low (much less than 0.05), and it is fairly safe to say that habitat significantly affects sodium concentrations. In order to infer how they differ we can go back to the raw data and observe that on average values in floodplain are higher. Also, note that I conducted the analysis on the original values for sodium rather than the log-transformed data. In general, this is what one should do when conducting non-parametric statistical tests. However, as this is a rank-based test, you will actually obtain identical results whether you log-transform the response variable or not. The relative rankings of samples from floodplain and uplands will not change.

v) Now let's look at spearman's rank correlation test. Check out the relationship between total base saturation and sand content in soil.

```
> lm_sand_TBS <- lm(Total_Base_Saturation~Sand, data=soils)
> summary(lm_sand_TBS)
> plot(lm_sand_TBS)
```

You will see that there are non-normality and heteroscedasticity issues here. Try log-transforming either or both variables. You will see that the issues do not really go away. In order to conduct a rank-based non-parametric test, you can do the following.

```
> cor.test(soils$Sand,soils$Total_Base_Saturation,data=soils,
      method="spearman")
```

Note, that identical results are obtained if you log-transform one or both of these variables. That is because in this case, the variables are each ranked independently, and we are then assessing the correlation between the ranks. What do you infer based on whether the Spearman's rho is positive or negative?

Also, you will see that R delivers a warning message. It is not a formal error message, which usually appears when there is a problem with the data or codes that prevent the function from actually running. However, such warnings should be headed, as they can indicate that we cannot be fully confident of the function output, in this case the results of our statistical test. In this case, we would check the help function and see that the exact argument can be set to FALSE, which may alleviate this warning.

```
> cor.test(soils$Sand,soils$Total_Base_Saturation,data=soils,
      method="spearman",exact=FALSE)
```