Session 2: Analysis with more than one set of treatments or groups

Factors and levels

In the last session we looked at comparing the averages of measurements in more than two groups using the one-way Analysis of Variance (parametric test) and the Kruskal-Wallis test (non-parametric version). In this session we will look at a more complex situation, where you are trying to compare the mean values of data that are grouped by more than one set of treatments **(or factors)**.

The example that we will be looking at is a study of the number of moth species collected at light traps. Six traps were set up in replicate areas of coniferous and deciduous forestry, and in three different months (June, July and August). The number of different species of moths captured in each trap was recorded after being left out overnight.

We can envisage the study as follows:

Forest type	6 (June)	7 (July)	8 (August)
Deciduous	6 traps	6 traps	6 traps
Coniferous	6 traps	6 traps	6 traps

So in this case there are two **factors** (sets of treatments): forest type, which has two **levels** (deciduous and coniferous) and month, which has three levels (6, 7, 8, representing the months). So in total we will be comparing 6 mean values, each represented by a combination of the two factors in the table above. As we have two factors, this is known as a **two-way ANOVA**. We will only be covering parametric techniques at this point as there are no comparable non-parametric techniques.

Download the **moth_traps.csv** file from Moodle and import into a dataframe called `moths`. Start a new section in your script file for the commands in this session.

If you check the structure of the `moths` dataframe, you should find that the `month` column is shown as integer data (**int** when you do `str(moths)`, i.e. whole numbers). R has recognised that the data are whole numbers and made a choice to define it as a column of integers. This is fine, but you can't use integers as a factor in ANOVA. Therefore we need to change the data to a different type (factor, like forest).

Factoring it in

We will do this by creating a new column in the dataframe, which is a 'factor' version of the month.

```
# create a new column called 'f_month' which is a factor
moths$f_month <- as.factor(moths$month)
```

Run `str(moths)` again: now we have both factors in the right format, we can proceed.

Starting the analysis

Writing the code for a two-way ANOVA is fairly similar to that for a one-way ANOVA, except that we want to look at the effects of both factors, and also determine if they *interact*. This means whether any effect of one factor, say *forest type* in this case, is the same across all levels of the other factor, *month* in this case. If the effect of one factor varies depending on the level of the other factor, then this is known as an **interaction**.

```
# load the 'car' library before we get going
library(car)

# store the result in an object called m1 (moths 1)
m1 <- lm(species_richness~f_month*forest, data=moths)
```

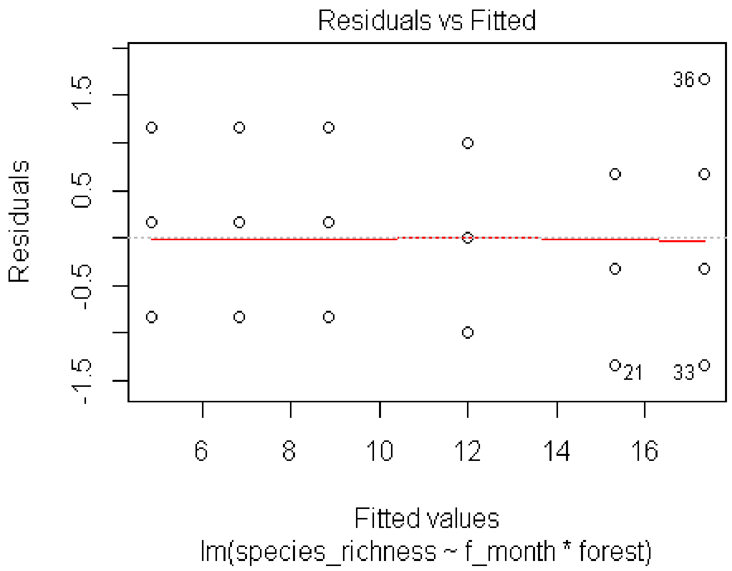
The * in the code above is essentially short for ‘test each factor individually and also the interaction’.

Don’t make assumptions

Before we go any further, we should set about checking the assumptions of the analysis (equal variances in each group, normally distributed residuals).

We can check these using the same visual and formal methods as for a one-way ANOVA. So for the equal variances:

```
plot(m1, which = 1)
```



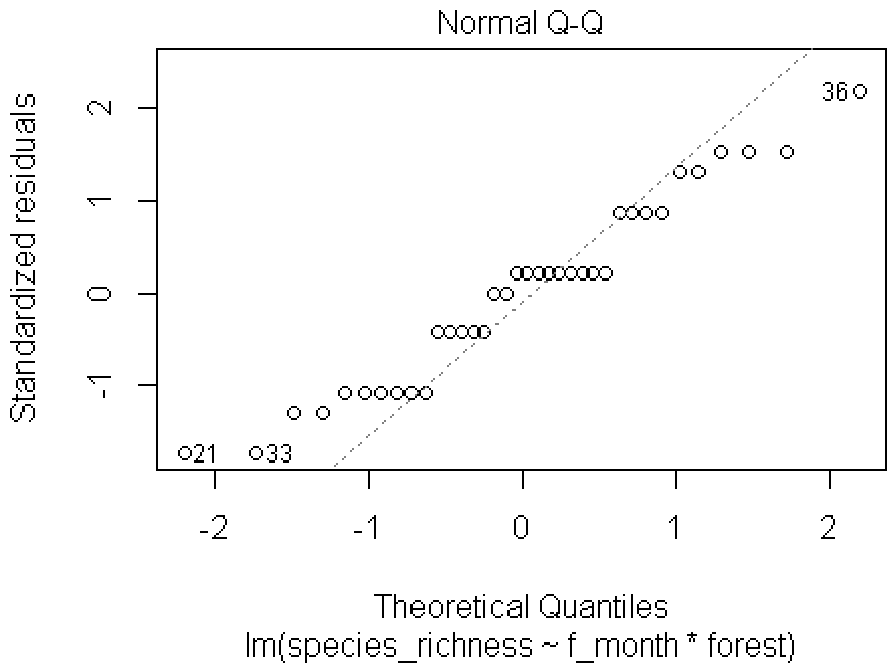
```
leveneTest(species_richness~f_month*forest,data=moths)
```

```
## Levene's Test for Homogeneity of Variance (center = median)
##       Df F value Pr(>F)
## group  5    0.15 0.9785
##       30
```

The plot of the residuals vs the fitted values looks OK - no consistent change in the spread of values above and below the ‘0’ from the left to the right and the Levene test is not significant.

So onto the normality of the residuals.

```
plot(m1, which = 2)
```



```
shapiro.test(m1$residuals)

## 
##  Shapiro-Wilk normality test
## 
## data:  m1$residuals
## W = 0.95552, p-value = 0.1559
```

Interaction or not?

In order to check the results, we use the Anova command as for the one-way ANOVA, but adding on a ‘type’ argument this time.

```
Anova(m1, type=3)
```

```
## Anova Table (Type III tests)
## 
## Response: species_richness
##                  Sum Sq Df  F value    Pr(>F)    
## (Intercept)     140.167  1 198.6614 9.061e-15 ***
## f_month          48.000  2  34.0157 1.933e-08 ***
## forest          154.083  1 218.3858 2.605e-15 ***
## f_month:forest    3.556  2   2.5197   0.09738 .  
## Residuals        21.167 30                      
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

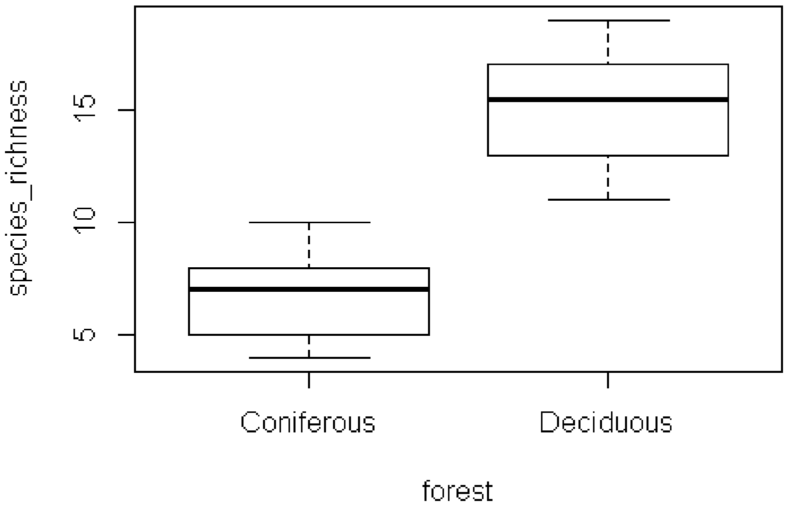
The thing to look at in the ANOVA table is the interaction line (labelled **`f_month:media`**). In this case the interaction is not significant (p value is > 0.05). If the interaction is not significant, we would then proceed to the next stage, which is to look at the two factors individually. These are known as the **main effects**.

Looking at the main effects

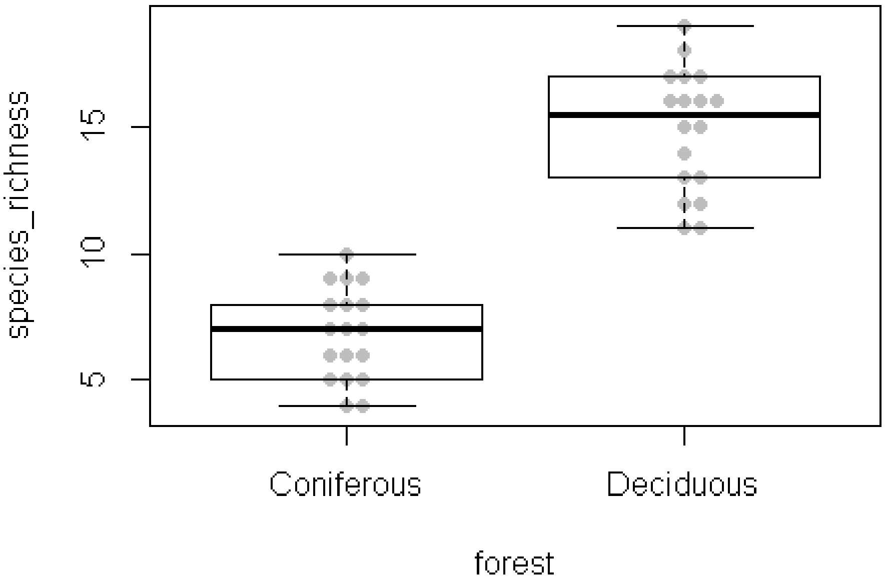
Both forest type and month have a significant effect (both p values are very much < 0.05). So the mean number of moth species trapped varies with forest type and month.

For forest type, we only have two levels, so there is no real need to do any post-hoc testing. If we plot the data, we should be able to see which level has the higher mean species richness.

```
boxplot(species_richness~forest, data=moths)
```



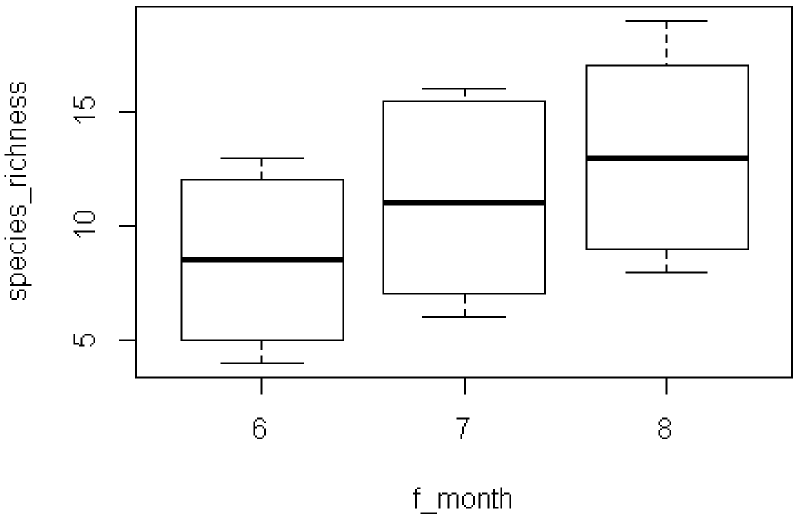
```
# we could do a beeswarm version alternatively
library(beeswarm)
beeswarm(species_richness~forest, data=moths, pch=19, col="Grey")
boxplot(species_richness~forest, data=moths, col="#00000000", add = TRUE)
```



Either way, it is obvious that the deciduous forest has a higher mean moth species richness.

For the month factor, we can do the same plotting, but we need to do a post-hoc test to find out where the differences actually are. We can use the TukeyHSD command as before, but this time we need to specify which factor we want to test.

```
# first do a boxplot to see what the data look like
boxplot(species_richness~f_month, data=moths)
```



```
# now do the Tukey post-hoc test, for the 'f_month' factor
TukeyHSD(aov(m1), which=c("f_month"))
```

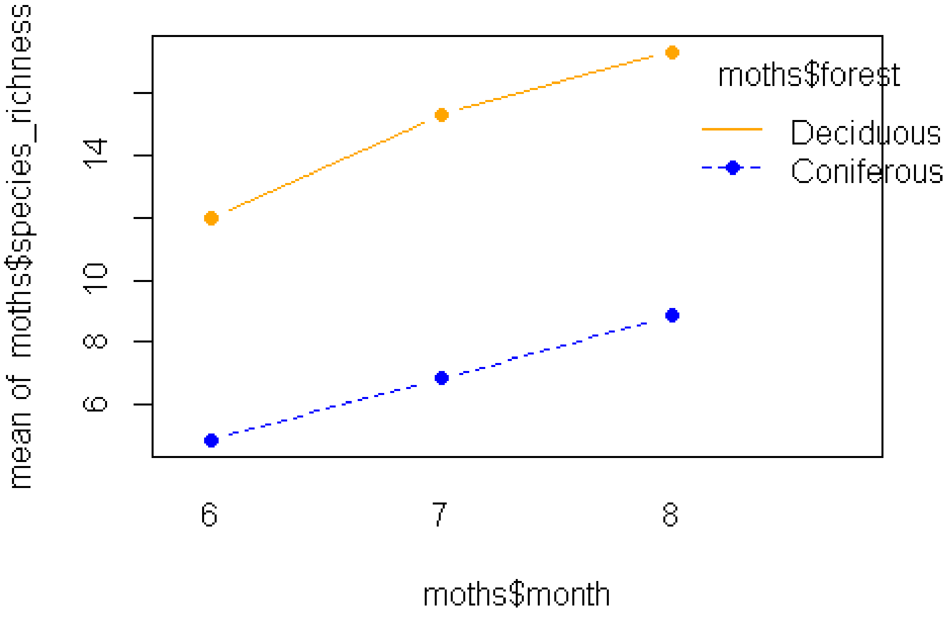
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = m1)
##
## $f_month
##         diff      lwr      upr   p adj
## 7-6 2.666667 1.821281 3.512052 0.0e+00
## 8-6 4.666667 3.821281 5.512052 0.0e+00
## 8-7 2.000000 1.154615 2.845385 6.5e-06
```

Interpreting the output from the Tukey test shows that each month is significantly different from the others, and the mean number of moth species recorded increases between June and August.

A useful way of summarising the pattern of variation between the mean values of all treatments is through what is known as an `interaction.plot`. This is a compact way of showing the means for each treatment combination and how they relate to each other.

We do this as follows:

```
interaction.plot(moths$month, moths$forest, moths$species_richness, type="b",
pch=19, col=c("Blue", "Orange"))
```



The first argument is which factor is going to be on the x-axis. The second argument is the other factor and the third is what the response or DV is. Type is what type of plot to do (here `"b"` means both points and lines. The symbol (`pch`) and colours (`col`) can also be specified - try different options if you want.

Make sure that you have written sufficient comments in your script to be able to go back to it in future sessions - you will need to be able to interpret your commands for other examples and for the tests in due course.
