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 effects of media type and incubation temperature on the growth of heterotrophic bacteria isolated from river water.

Two different media (full strength PYBGF agar and 1/20 PYBGF agar) were tested, and six replicates were incubated aerobically at three different temperatures (10, 20 and 30^{o}C). After incubation for a week, the number of colonies formed on each plate was counted.

We can envisage the experiment as follows:

Media	10^{o}C	20^{o}C	30^{o}C
Full strength	6 replicates	6 replicates	6 replicates
1/20th strength	6 replicates	6 replicates	6 replicates

So in this case there are two **factors** (sets of treatments): media type, which has two **levels** (full strength and 1/20 strength) and temperature, which has three levels (10, 20 and 30). 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 colonies.csv file from Moodle and import into an dataframe called `colonies`. Start a new section in your script file for the commands in this session.

If you check the structure of the `colonies` dataframe, you should find that the `temperature` column is shown as integer data (int when you do `str(colonies)`, 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 media).

Factoring it in

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

```
# create a new column called 'f_temp' which is a factor
colonies$f_temp <- as.factor(colonies$temperature)
```

Run `str(colonies)` 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 media type in this case, is the same across all levels of the other factor, temperature 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 c1 (colonies 1)
c1 <- lm(colony_count~f_temp*media, data=colonies)
```

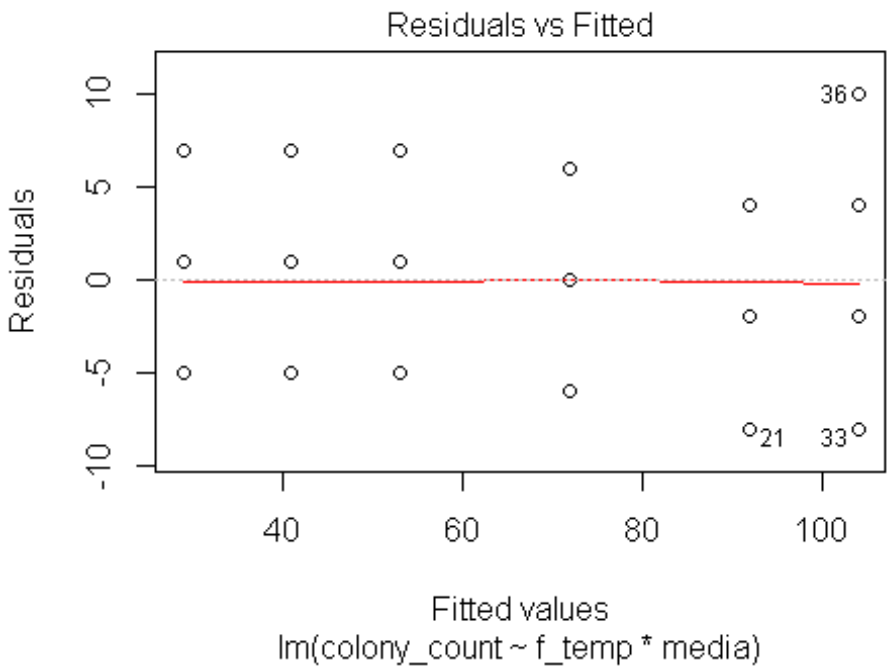
The * in the code above is essentially short for 'test each factor individually and also the interaction'.

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(c1, which = 1)
```



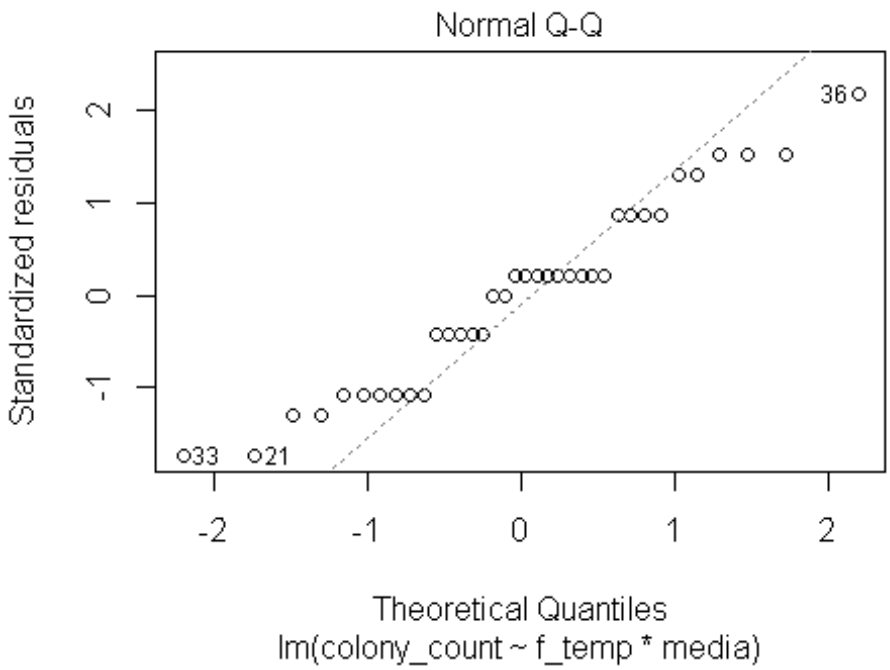

```
leveneTest(colony_count~f_temp*media,data=colonies)
```

```
## 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(c1, which = 2)
```




```
shapiro.test(c1$residuals)

##
##  Shapiro-Wilk normality test
##
## data:  c1$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(c1, type=3)

## Anova Table (Type III tests)
##
## Response: colony_count
##              Sum Sq Df  F value    Pr(>F)
## (Intercept)    5046  1 198.6614 9.061e-15 ***
## f_temp         1728  2  34.0157 1.933e-08 ***
## media          5547  1 218.3858 2.605e-15 ***
## f_temp:media    128  2   2.5197   0.09738 .
## Residuals       762 30
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

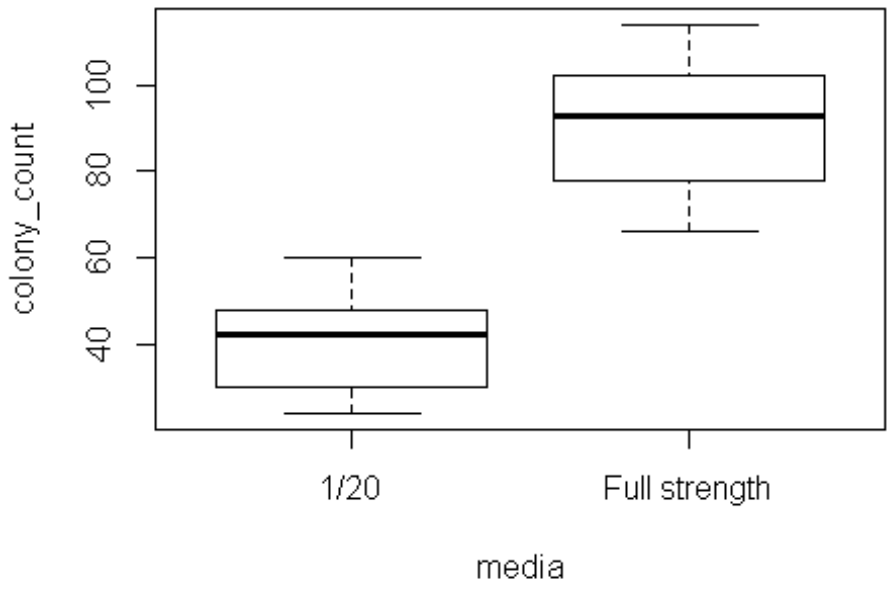
The thing to look at in the ANOVA table first is the interaction line (labelled **`f_temp:media`**). In this case the interaction is not significant (p value is > 0.05). If the interaction is not significant, we would then proceed to the next stage, which is to look at the two factors individually. These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

Looking at the main effects

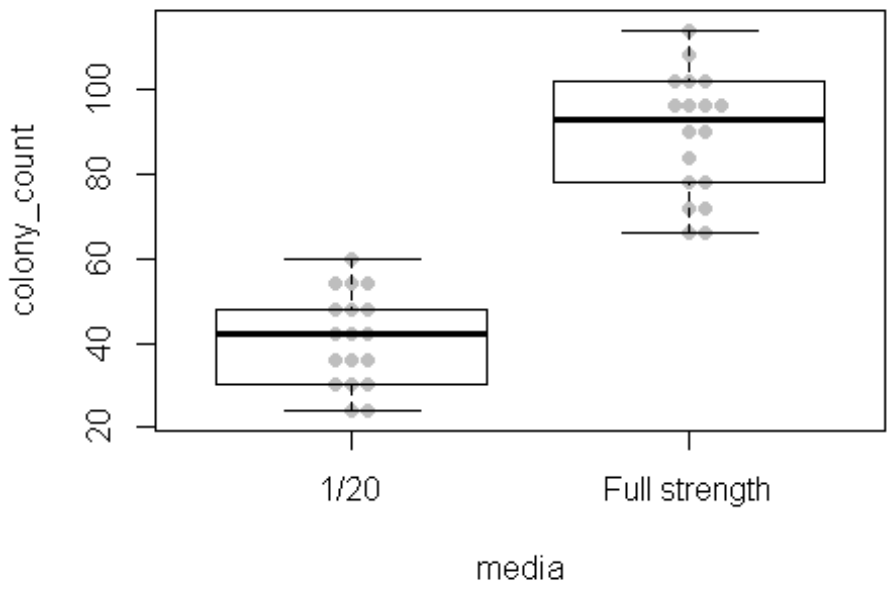
Both media type and temperature have a significant effect (both p values are very much < 0.05). So the mean number of colonies varies with media type and with temperature.

For media 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 count.

```
boxplot(colony_count~media, data=colonies)
```



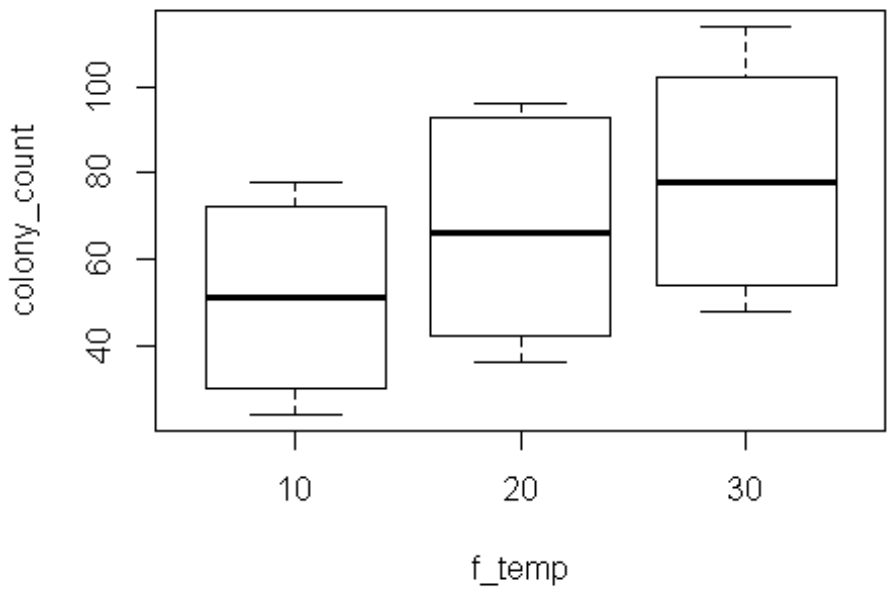

```
# we could do a beeswarm version alternatively
library(beeswarm)
beeswarm(colony_count~media, data=colonies, pch=19, col="Grey")
boxplot(colony_count~media, data=colonies, col="#00000000", add = TRUE)
```



Either way, it is obvious that the full strength media has a higher mean number of colonies.

For the temperature 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(colony_count~f_temp, data=colonies)
```



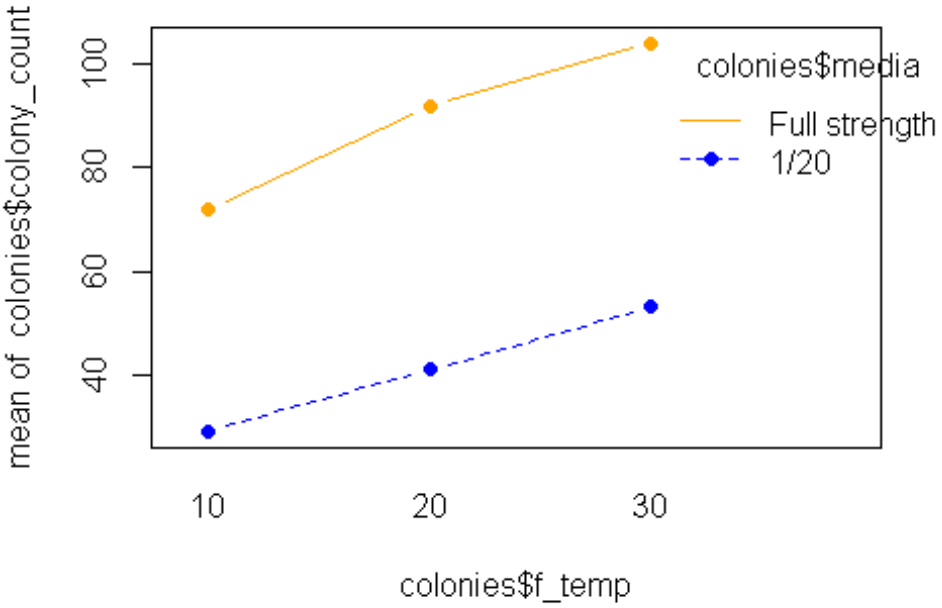

```
# now do the Tukey post-hoc test, for the 'f_temp' factor
TukeyHSD(aov(c1), which=c("f_temp"))

##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = c1)
##
## $f_temp
##       diff       lwr      upr   p adj
## 20-10   16 10.927688 21.07231 0.0e+00
## 30-10   28 22.927688 33.07231 0.0e+00
## 30-20   12  6.927688 17.07231 6.5e-06
```

Interpreting the output from the Tukey test shows that each temperature is significantly different from the others, and the mean number of colonies increases with temperature over the range tested.

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(colonies$f_temp, colonies$media, colonies$colony_count,
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 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.
