ANOVA in R

1-Way ANOVA

We're going to use a data set called InsectSprays. 6 different insect sprays (1 Independent Variable with 6 levels) were tested to see if there was a difference in the number of insects found in the field after each spraying (Dependent Variable).

```
> attach(InsectSprays)
> data(InsectSprays)
> str(InsectSprays)
'data.frame':   72 obs. of  2 variables:
 $ count: num  10 7 20 14 14 12 10 23 17 20 ...
 $ spray: Factor w/ 6 levels "A","B","C","D",..: 1 1 1 1 1 1 1 1 1 1 ...
```

1. **Descriptive statistics**
    a. Mean, variance, number of elements in each cell
    b. Visualise the data – boxplot; look at distribution, look for outliers

We'll use the tapply() function which is a helpful shortcut in processing data, basically allowing you to specify a response variable, a factor (or factors) and a function that should be applied to each subset of the response variable defined by each level of the factor. I.e. Instead of doing:

```
> mean(count[spray=="A"])   # and the same for B, C, D etc.
```

We use tapply(response,factor,function-name) as follows

- Let's look at the means:

```
> tapply(count, spray, mean)
       A         B         C         D         E         F
14.500000 15.333333  2.083333  4.916667  3.500000 16.666667
```

- The variances:

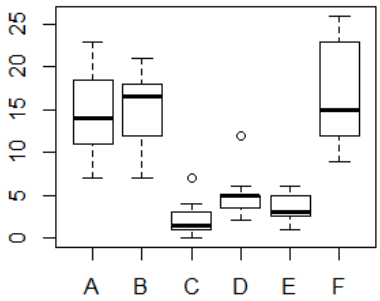
```
> tapply(count, spray, var)
       A         B         C         D         E         F
22.272727 18.242424  3.901515  6.265152  3.000000 38.606061
```

- And sample sizes

```
> tapply(count, spray, length)
 A  B  C  D  E  F
12 12 12 12 12 12
```

- And a boxplot:

```
> boxplot(count ~ spray)
```



    o How does the data look?

A couple of Asides

- Default order is alphabetical. R needs, for example, the control condition to be 1st for treatment contrasts to be easily interpreted.
- If they're not automatically in the correct order – i.e. if they were ordered variables, but came out alphabetically (e.g. "Very.short","Short","Long","Very.long" or "A", "B", "Control"), re-order the variables for ordered IV:

To change to, for example, F < B < C < D < E < A, use:

```
> Photoperiod<-ordered(spray,levels=c("F","B","C","D","E","A"))
```

Check it:

```
> tapply(count,Photoperiod,mean)
        F         B         C         D         E         A
16.666667 15.333333  2.083333  4.916667  3.500000 14.500000
```

- If you want to check that a variable is a factor (especially for variables with numbers as factor levels). We use the `is.factor` directive to find this out

```
is.factor(spray)
[1] TRUE
```

2. **Run 1-way ANOVA**
    a. **Oneway.test ( )**

- Use, for example:

```
> oneway.test(count~spray)
        One-way analysis of means (not assuming equal variances)
data:  count and spray
F = 36.0654, num df = 5.000, denom df = 30.043, p-value = 7.999e-12
```

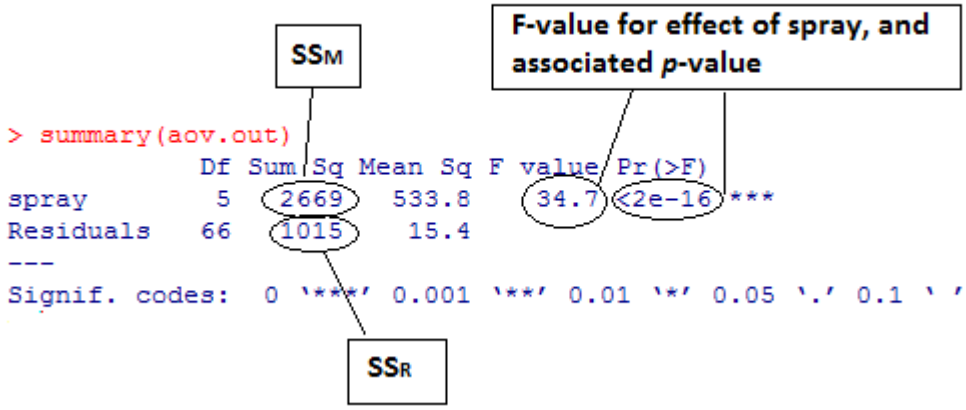
- Default is equal variances (i.e. homogeneity of variance) not assumed – i.e. Welch's correction applied (and this explains why the denom df (which is normally k*(n-1)) is not a whole number in the output)
    - To change this, set "`var.equal=`" option to TRUE

- `Oneway.test( )` corrects for non-homogeneity, but doesn't give much information – i.e. just *F*, *p*-value and *df*s for numerator and denominator – no MS etc.

    b. **Run an ANOVA using aov( )**

- Use this function and store output and use extraction functions to extract what you need.

```
> aov.out = aov(count ~ spray, data=InsectSprays)
> summary(aov.out)
```



∴ ***F*(5,66) = 34.7; *p* < .000**

3. **Post Hoc tests**

- **Tukey HSD(Honestly Significant Difference) is default in R**

```
> TukeyHSD(aov.out)
Tukey multiple comparisons of means
95% family-wise confidence level
Fit: aov(formula = count ~ spray, data = InsectSprays)

$spray
            diff         lwr          upr           p adj
B-A    0.8333333  -3.866075  5.532742 0.9951810
C-A -12.4166667 -17.116075 -7.717258 0.0000000
D-A  -9.5833333 -14.282742 -4.883925 0.0000014
E-A -11.0000000 -15.699409 -6.300591 0.0000000
F-A    2.1666667  -2.532742  6.866075 0.7542147
C-B -13.2500000 -17.949409 -8.550591 0.0000000
D-B -10.4166667 -15.116075 -5.717258 0.0000002
E-B -11.8333333 -16.532742 -7.133925 0.0000000
F-B    1.3333333  -3.366075  6.032742 0.9603075
D-C    2.8333333  -1.866075  7.532742 0.4920707
E-C    1.4166667  -3.282742  6.116075 0.9488669
F-C   14.5833333   9.883925 19.282742 0.0000000
E-D   -1.4166667  -6.116075  3.282742 0.9488669
F-D   11.7500000   7.050591 16.449409 0.0000000
F-E   13.1666667   8.467258 17.866075 0.0000000
>
```

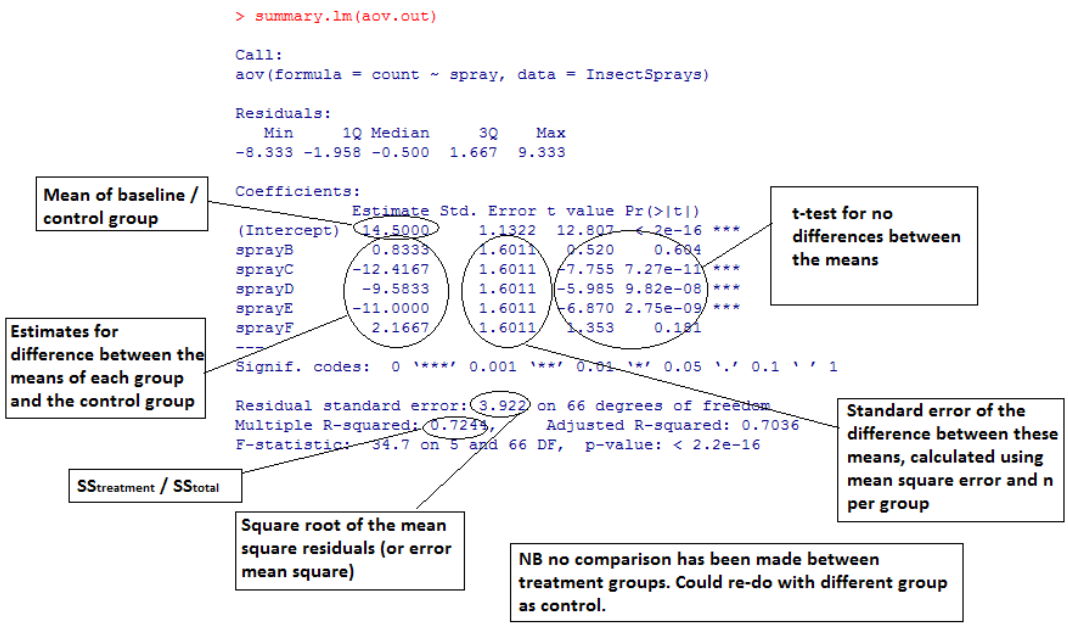
4. **Contrasts**

NB: ANOVA and linear regression are the same thing – more on that tomorrow. For the moment, the main point to note is that you can look at the results from aov() in terms of the linear regression that was carried out, i.e. you can see the parameters that were estimated.

```
> summary.lm(aov.out)
```

Implicitly this can be understood as a set of (non-orthogonal) contrasts of the first group against each of the other groups. R uses these so-called 'Treatment' contrasts as the default, but you can request alternative contrasts (see later)

Interpreting a Treatment Contrasts Output



5. Test assumptions

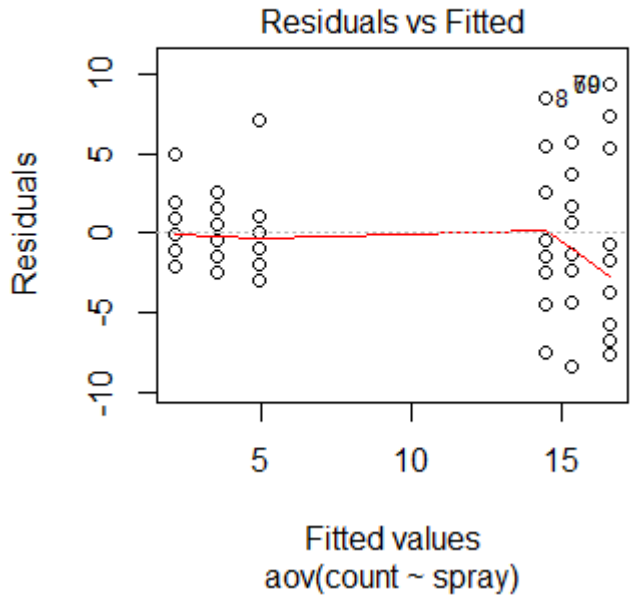
a. Homogeneity of variance

```
bartlett.test(count ~ spray, data=InsectSprays)
        Bartlett test of homogeneity of variances
data:  count by spray
Bartlett's K-squared = 25.9598, df = 5, p-value = 9.085e-05
```

∴ *Significant result, therefore variances cannot be assumed to be equal*

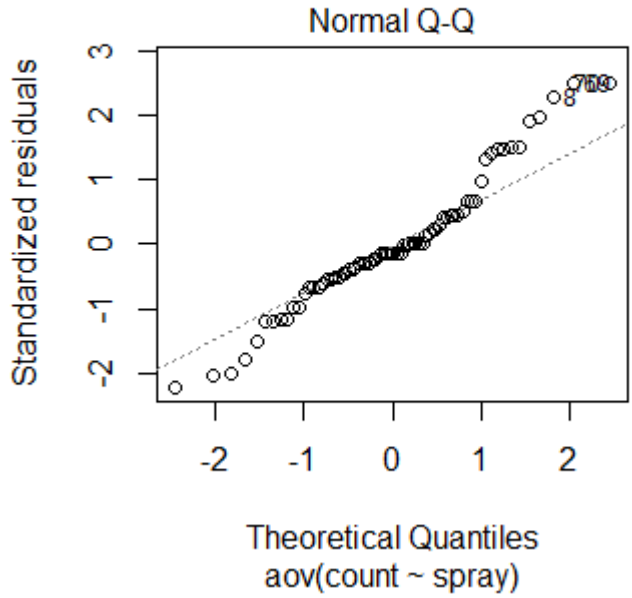
b. Model checking plots

```
> plot(aov.out)       # the aov command prepares the data for these plots
```

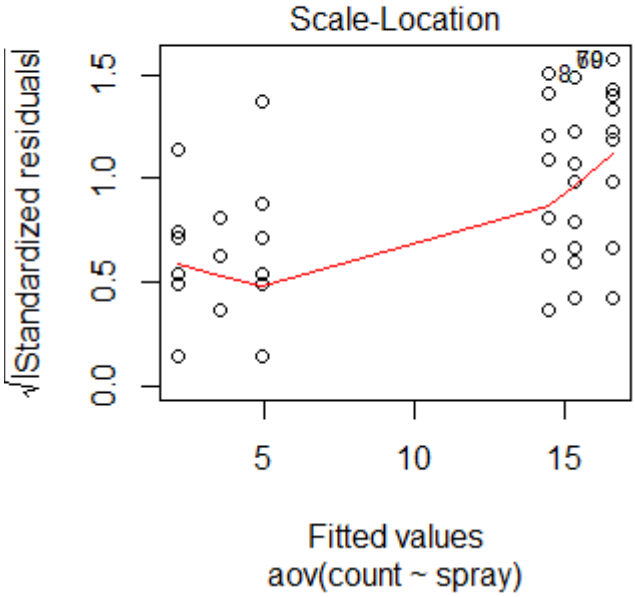


This shows if there is a pattern in the residuals, and ideally should show similar scatter for each condition. Here there is a worrying effect of larger residuals for larger fitted values. This

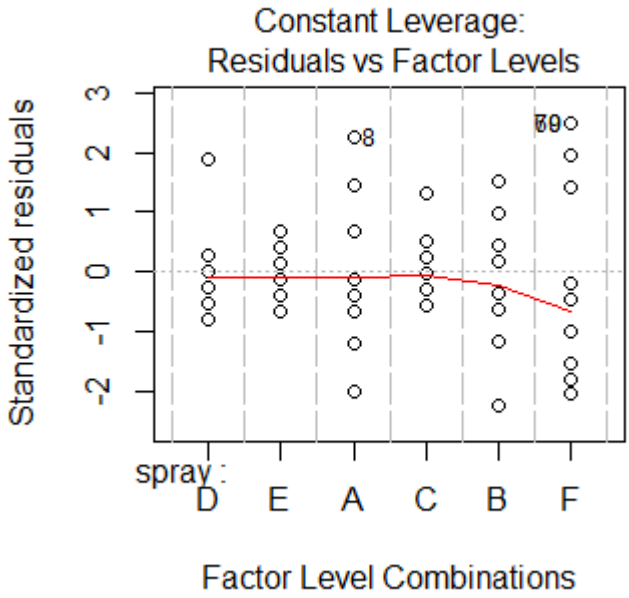
is called ‘heteroscedascity’ meaning that not only is variance in the response not equal across groups, but that the variance has some specific relationship with the size of the response. In fact you could see this in the original boxplots. It contradicts assumptions made when doing an ANOVA.



This looks for normality of the residuals; if they are not normal, the assumptions of ANOVA are potentially violated.



This is like the first plot but now to specifically test if the residuals increase with the fitted values, which they do.



This gives an idea of which levels of the factor are best fitted.

6. Non-parametric alternative to ANOVA:

```
> kruskal.test(count ~ spray, data=InsectSprays)
        Kruskal-Wallis rank sum test
data:  count by spray
Kruskal-Wallis chi-squared = 54.6913, df = 5, p-value = 1.511e-10
```

As for the Wilcoxon test (or Mann-Whitney test) with two samples, this test converts the response values to ranks, and tests whether the ranks are distributed equally across the conditions, as would be expected under the null hypothesis.

7. ANOVA as Linear Regression Analysis

This time, rather than 'attaching' the data frame, we will use the 'with' construct (see session one) to name the data frame and then do operations on variables within it.

```
> summary(PlantGrowth)
     weight       group
 Min.   :3.590   ctrl:10
 1st Qu.:4.550   trt1:10
 Median :5.155   trt2:10
 Mean   :5.073
 3rd Qu.:5.530
 Max.   :6.310
> with(PlantGrowth, tapply(weight, group, mean))
 ctrl  trt1  trt2
5.032 4.661 5.526
> with(PlantGrowth, tapply(weight, group, var))
     ctrl      trt1      trt2
0.3399956 0.6299211 0.1958711
> with(PlantGrowth, bartlett.test(weight ~ group))
        Bartlett test of homogeneity of variances
```

```
data:  weight by group
Bartlett's K-squared = 2.8786, df = 2, p-value = 0.2371
```

Now instead of running an ANOVA with aov(), we will run a linear regression with lm()

```
> lm.out = with(PlantGrowth, lm(weight ~ group))
> summary(lm.out)    # the default summary display will be the linear
                        regression

Call:
lm(formula = weight ~ group)

Residuals:
    Min      1Q  Median      3Q     Max
-1.0710 -0.4180 -0.0060  0.2627  1.3690

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   5.0320     0.1971  25.527   <2e-16 ***
grouptrt1    -0.3710     0.2788  -1.331   0.1944
grouptrt2     0.4940     0.2788   1.772   0.0877 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.6234 on 27 degrees of freedom
Multiple R-squared: 0.2641,   Adjusted R-squared: 0.2096
F-statistic: 4.846 on 2 and 27 DF,  p-value: 0.01591

> summary.aov(lm.out)      # we can ask for the corresponding ANOVA table

            Df Sum Sq Mean Sq F value Pr(>F)
group        2  3.766  1.8832   4.846 0.0159
Residuals   27 10.492  0.3886
```

There is a difference, but where does this difference lie?
Post Hoc test:

```
> TukeyHSD(results)
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = weight ~ group)

$group
             diff        lwr       upr     p adj
trt1-ctrl -0.371 -1.0622161 0.3202161 0.3908711
trt2-ctrl  0.494 -0.1972161 1.1852161 0.1979960
trt2-trt1  0.865  0.1737839 1.5562161 0.0120064
```