

Lab 8 Part I

ANOVA, Signal:Noise & Pairwise Comparisons

Questions? montwe@ualberta.ca or isaacren@ualberta.ca

In this lab, we will run an ANalysis Of VAriance (ANOVA) just as we have conceptually derived it in the lecture: as a ratio of signal to noise. The signal being the variance due to the observed differences between the sample means and the noise being the variance within treatments. This can be done for two treatments, but the main advantage of ANOVA is that we can use it for as many treatments as we like.

7.1. Analysis of Variance, F-test

Nevertheless, let's start with two samples equivalent to the T-Tests in the previous lab, and also conduct an analysis of variance (ANOVA). Let's use just variety A and B at Farm 1 and see if the results from ANOVA are the same as from the T-Test.

Rather than importing and subsetting the data table, it's easier to type the numbers directly into vectors that represent the two samples for varieties A and B:

```
s1=c(720,740,690,760)
s2=c(515,480,545,492)
```

The variance within (vw) treatments is easily understood and calculated. We calculate it separately for each treatment, and then take the mean of that. Note that in doing so, we assume that those variances are about the same. Hence, the assumption of "homogeneity of variances" for ANOVA. That is our measure of "noise":

```
v1=var(s1)
v1
v2=var(s2)
v2
vw=mean(c(v1,v2))
vw
```

Our measure of "signal", the variance between (vb) treatments, is a little trickier to understand. The first steps are clear: calculate the treatment means and then the variance among the means. Big differences among treatments mean that this variance will be large, i.e. my measure of "signal" is large as well. So far so good.

```
x1=mean(s1)
x1
x2=mean(s2)
x2
vb=var(c(x1,x2))
vb
```

However, think about the calculation if there is no signal. In that case my signal to noise ratio should be about 1:1. However, the means of the samples will not be as variable as the sample observations themselves. If we were to repeatedly sample from the same population, the means of the samples would be closer to the population mean. (Larger sample sizes get closer than with small sample sizes).

If we want a measure of variance for the “signal” equivalent to that of the “noise” (1:1 ratio), we can back-calculate the variance with the equation for the standard error of the mean “ stdev/\sqrt{n} ”. If I take the square of that, then I get the variance “ var/n ”. Then we can multiply by the sample size (n):

```
vb=vb*4  
vb
```

We are now close to being done. Our F-value is simply the signal to noise ratio:

```
f=vb/vw  
f
```

We can look up the corresponding p-value in the Appendix of a statistics textbook, or we can use the built in R functions that serve the purpose as well.

We need two degrees of freedom for the p-value calculation: for variance between (v_b) in the numerator and variance within (v_w) in the denominator of the signal to noise ratio. For v_b it is the number of treatments minus one ($2-1=1$), and for v_w it is the total number of samples minus the number of treatments ($8-2=6$).

You can either evaluate whether your F-value is larger than a critical F-value for a pre-determined α -level, or simply calculate p directly (preferred, because it contains more information):

```
qf(0.95,1,6)      # critical F-value for  $\alpha=0.05$   
pf(f,1,6)        # your P-value (right side)  
1-pf(f,1,6)     # your P-value (right side, therefore, 1-p)
```

Can you confirm that this is the same p -value as for the T-test? (There may be minor differences due to rounding errors and the Welch correction for unequal variances.)

```
t.test(s1,s2)
```

Now, expand the above code to include all three varieties A, B, and C in the ANOVA. The sample three, corresponding to variety C at Farm 1 is:

```
s3=c(505,540,510,502)  
v3=var(s3)  
x3=mean(s3)  
vw=mean(c(v1,v2,v3))  
vb=var(c(x1,x2,x3))*4  
f=vb/vw  
qf(0.95,2,9)  
1-pf(f,2,9)
```

You can check your results developed from first principles with the ANOVA function, but note that now the data has to be arranged in the standard data table format, so first import the data table (reduced to Farm1) from one of the previous labs.

```
dat1 = read.csv("lentils-Farm1.csv")
out1 = lm(YIELD~VARIETY,data=dat1)
anova(out1)          # OR: anova(lm(YIELD~VARIETY,data=dat1))
```

As a side note, you can also test if two *variances* are significantly different with an F-test. Try this out with the variances of variety A and B (v1 and v2).

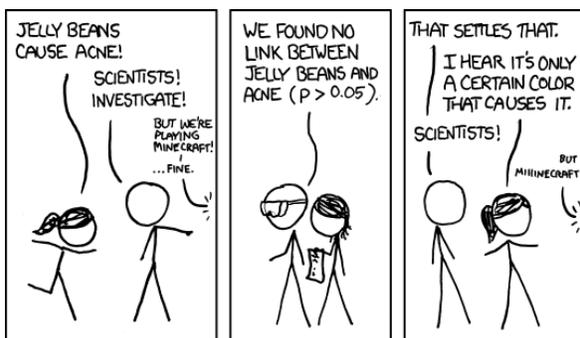
```
f=v1/v2                # Actual F-value
df1=4-1
df2=4-1
qf(0.95,df1,df2)      # critical F-value
pf(f,df1,df2)         # P-value
```

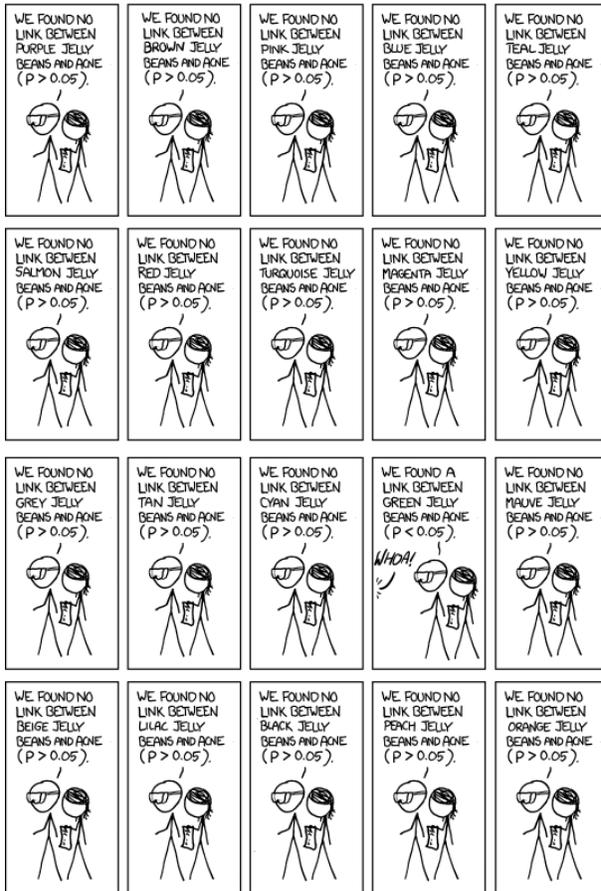
7.2. Pair-wise comparisons, α -level adjustments

Once you have done your analysis of variance with more than two treatment levels and get a significant result, you usually want to follow this up by checking which treatment levels are significantly different from each other. In our case A versus B, A versus C, and B versus C. We can easily do this with three T-Tests. You may ask why we bothered with an ANOVA in the first place.

With more treatment levels, say 10 different varieties, your number of pairwise comparisons quickly goes through the roof, though. The formula for this is $(t*(t-1))/2$, so $(3*2)/2=3$ comparisons for three treatment levels, but $(10*9)/2=45$ comparisons for 10 treatment levels, and 190 comparisons for 20 treatment levels.

That has ramifications for our significance values. If we accept an α -level of 0.05 for one T-test, it means that 1 in 20 times (5% chance), we get a false positive (i.e. we believe there is an effect when there really is none). For multiple pairwise comparisons, I get a $1-(1-\alpha)^t$ probability of at least one false positive. For $t=3$ that is 14% chance, for $t=10$ that grows to a 40% chance of at least one false positive. So, if you keep stringing up those T-tests, this is the type of science that you get as a result:

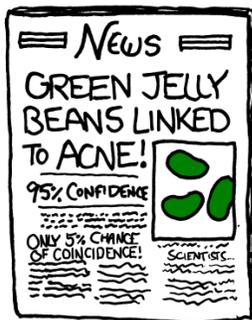




Further reading for your interest on what that means for science:

Jonathan Lehrer. 2010. The truth wears off. *The New Yorker Magazine* December 13, 2010. Pages 53-57 – gives you a bigger picture perspective on how this affects science.

Rice, W.R. 1989. Analyzing tables of statistical tests. *Evolution* 43: 223-225 – provides you with some simple solutions on how to handle this situation for any set of statistical tests.



Analysis of variance already protects you against this type of experiment-wise α -level inflation (at least from the perspective of a single study or researcher). If you get a significant F-value, you can be sure that one or more of your treatments differ significantly from another at a non-inflated α -level of your choosing. I.e. the p -value truly represents the probability of getting a false positive result. It is for the pairwise comparisons that we have to make p -value adjustments.

One of the most common p -value adjustments is Tukey's Honest Significant Difference test (Tukey's HSD), but there are many other options. There is actually no perfect solution for this

problem, so many people put something forward that is approximately right. Bonferroni's test is one of the first and simplest, but overcompensates. The sequential Bonferroni test, developed by Holms is a better solution (and explained in the paper by Rice (above). There are others.

I recommend that you use Tukey's HSD, implemented by the `lsmeans` package. This has other desirable features as well that will become important for pairwise tests in factorial and unbalanced designs, so we might as well use it here. Now we can run our analysis of variance.

Next, we estimate the means and their standard errors (and 95% confidence intervals) from the Analysis of Variance output. Note that those means may slightly differ from the treatment means calculated straight from the raw data (more on this later) and that standard error of those estimates are all the same (if your sample size is the same). This is what you should report in your graphs and tables, so you may want to export them as a `.csv` file:

```
library(lsmeans)
out2=lsmeans(out1, ~VARIETY)
summary(out2)
out3=data.frame(summary(out2))
out3
write.csv(out3, "Means and SE.csv")
```

Lastly, we use the compact letter design (`cld`) function of the `lsmeans` package to run our Tukey's pairwise test. (The `cld` function requires the `multcompView` package to be installed, so make sure to install it if it isn't already loaded.) There are many clever options to arrange that table. Check the help file. You can also write that result out. Assign it to an output file, convert it to a data frame as above but without the `summary()` function, then write it as a CSV.

```
install.packages("multcompView")
library(multcompView)
?cld
cld(out2, alpha=0.05, adjust="tukey", Letters="abcd", sort=F)
cld(out2, alpha=0.05, adjust="tukey", Letters="||||", sort=T)
```