Lab 8 Part II
Multi-way-Anova, assumptions & transformations

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

Now that we have mastered the one-way ANOVA statistical method, we'll move on to look at comparisons when there is more than just one treatment (independent variable). Each treatment may still have many levels (like varieties A, B, C, D, etc.). However, multiple factors mean that your experiment has a qualitatively different set of treatment levels (like fertilizers N, P, K, …) as well. Usually, you have a complete factorial design, where each level of your first treatment is tested together with each level of your second treatment.

Experiments with two factors can be very insightful, because they can reveal interactions among treatments. However, it is hard to interpret 3-factor interactions. It is best to avoid such experiments and rather do a limited set of two-way ANOVAs for the combinations that really matter. 4-factor ANOVAs are even more difficult to understand, possibly setting yourself up for scientific failure due to low interpretability and weak conclusions – despite massive amounts of work. Instead, the purpose of experimentation is meant to reduce factors, so you can more easily understand and interpret the role of one or two factors that you purposely vary in a controlled manner.

8.3 Multi-factor Anova

For a multi-factor ANOVA, there are two effect types: main effects (one for each treatment) and an interaction effect (for combinations of treatments): ***Main effect*****:** the effect of an independent variable (treatment) on a group. ***Interaction*****:** the effect of one independent variable on the other.

We will use the lentil data for this lab, which has two factors (FARM and VARIETY), so start by importing the data into R. Then, run a multi-factor ANOVA by including both the FARM and VARIETY variables in the model. We will also use ggplot2 to visualize the interactions.

```
library(ggplot2)

dat1=read.csv("lentils.csv")

out1= lm(YIELD~FARM*VARIETY,data=dat1)
anova(out1)
```

As before, we can follow this up with pairwise comparisons, but now with two factors, we have options to structure the adjustments with a "by=" option. Normally, you want to know the differences among your first treatment levels (say VARIETY) within levels of your second treatment (FARM) or vice versa. However, you normally don't care to know whether variety A at farm 1 is different from variety B at farm 2.

```
library(lsmeans)
```

```
library(multcompView)
out2=lsmeans(out1, ~VARIETY*FARM)
cld(out2, adjust="tukey", by="FARM", Letters=letters, sort=F)
cld(out2, adjust="tukey", by="VARIETY", Letters=letters, sort=F)
cld(out2, adjust="tukey", Letters=letters, sort=T)
```

Selected subsets of contrasts can massively cut back on the number of pairwise comparisons you are making. That increases your statistical power to detect a difference because your experiment-wise α-level stays relatively low. This experiment is not ideal to show that because most p-values are already <0.0001. However, compare the p-values for the Farm1-VarietyB-C pair under 6 pairwise comparisons versus under 15 comparisons (The option details=T will give us the actual p-values):

```
cld(out2, adjust="tukey", by="FARM", details=T)
cld(out2, adjust="tukey", details=T)
```

A useful graphical check if you have significant interactions is this plot, where you list your two independent variables, and then your dependent variable. Note that this works independently of the ANOVA and reads the raw data. Thus, in 3-way designs you can do multiple interaction plots for three pairs of independent variables to visualize their interactions:

```
ggplot(dat1, aes(x=FARM,y=YIELD,
group=VARIETY,linetype=VARIETY,col=VARIETY)) +
  geom_smooth(method = lm, se=F) +
  theme_bw()
```

8.4 Checking the assumptions of ANOVA

You can use a boxplot for a visual check of normality and homogeneity of variances for multi-level experiments by listing all your treatments with a multiplier. You can use more than two treatments.

```
ggplot(dat1, aes(x=FARM,y=YIELD, fill=VARIETY)) +
  geom_boxplot() +
  theme_bw()
```

What you are looking for are roughly symmetrical boxes and whiskers (i.e. normality), and roughly similar size boxes among treatments (i.e. homogeneity of variance, also known as homoscedasticity). Here, the concern would be that the assumption of homogeneity of variances is violated.

Technically, you can also use histograms for each treatment combination to inspect your data. The histograms should show a symmetrical distribution. However, this does not work very well for small sample sizes:

```
ggplot(dat1[dat1$VARIETY=="B"&dat1$FARM=="Farm1",], aes(x=YIELD)) +
  geom_histogram(binwidth=30)+
  theme_bw()
```

Instead of checking normality for each treatment combination, you can also calculate residuals (i.e. the variance left after treatment effects have been accounted for). Then, you can look at the histogram of all your data points together. The first command below will give you multiple graphs

that you sequentially visualize by repeatedly hitting enter. The one we care about (i.e. that we covered in class) are the residuals over predicted (fitted) values.

```
plot(lm(YIELD~FARM*VARIETY,data=dat1))  # several versions
out1=lm(YIELD~FARM*VARIETY,data=dat1)
plot(residuals(out1)~fitted(out1)) # just residuals
```

We can see that there is symmetry around the horizontal zero line, but a clear wedge-shape growing wider toward the right. That means that homogeneity of variances is likely violated, where treatments with larger mean values (toward the right) are also more variable.

8.5 Data transformations

Such problems can often be fixed with transformations of your data. Try different transformations below and then revisit the residual plots or check the result of transformations with boxplots. Do you see improvements?

```
sqrt_YIELD=sqrt(dat1$YIELD)
log10_YIELD=log10(dat1$YIELD)
inv_YIELD=1/dat1$YIELD

out1=lm(log10_YIELD ~ dat1$FARM * dat1$VARIETY)
plot(residuals(out1)~fitted(out1)) # just residuals

ggplot(dat1, aes(x=FARM,y=sqrt_YIELD, fill=VARIETY)) +
  geom_boxplot() +
  theme_bw()

ggplot(dat1, aes(x=FARM,y=log10_YIELD, fill=VARIETY)) +
  geom_boxplot() +
  theme_bw()

ggplot(dat1, aes(x=FARM,y=inv_YIELD, fill=VARIETY)) +
  geom_boxplot() +
  theme_bw()
```

You can fine-tune the transformations: they become more powerful if you subtract a constant, so that the smallest value in the dataset approaches 1. You can also change the base of the logarithm to smaller values to make the log-transformation less powerful:

```
sqrt2_YIELD=sqrt(dat1$YIELD-160)
ln_YIELD=log(dat1$YIELD)
log2_YIELD=log2(dat1$YIELD)
```

Once you have arrived at a satisfactory residual plot, use that transformation to calculate your p-values, but in graphs and tables it is usually preferable to report your results in original units. If the data is severely skewed, use boxplots and quantiles rather than bar charts and standard deviations.

8.6 Why not statistically test for violation of assumptions?

Now all this may seem a little subjective as we are doing a visual inspection of the plots to determine if the data is normal enough to proceed with parametric analyses. Wouldn't it be great if there was an easy criterion to test assumptions of normality? There are no good options, however. You may have heard that there are statistical tests that you can use to test for significant deviations from normality and homogeneity of variances, but the significance values for these tests are primarily driven by sample size and to a lesser degree by the true distribution of your data. Therefore, they only really work when sample size is high – which makes them redundant because the central limit theorem should kick in and create normality. When sample size is low, they do not have the power required to detect non-normality. Accordingly, these tests can never really give you a meaningful answer to determine if doing parametric analyses is appropriate or not.

Let's do some simulations to illustrate this. First, we create two large random populations of 5,000 individuals each that have a normal distribution (`pop1`) and a skewed distribution (`pop2`). We can look at them and apply a Shapiro test of normality. The null hypothesis for a Shapiro-Wilk test is that a sample comes from a normally distributed population (i.e., if the p-value is less than your pre-determined alpha level, then you can reject your null hypothesis and provide evidence that the data is not from a normal distribution). You should get the answer you expect. There should be a significant departure from normality for `pop2`:

```
pop1=rnorm(5000,5)
pop2=log(pop1)

pop=as.data.frame(cbind(pop1=pop1,pop2=pop2))

ggplot(pop,aes(x=pop1))+
  geom_histogram()
shapiro.test(pop1)# p-value is not low, fail to reject Ho

ggplot(pop,aes(x=pop2))+
  geom_histogram()
shapiro.test(pop2)# very low p-value, reject Ho
```

So far no issues. In statistics we work with samples and not with the entire population, however. Often the number of truly independent sampling units or experimental units is quite small. In our lentil field trial, it was only n=4 per treatment combination, but let's try taking samples of 250, 25, and 5 units from the skewed `pop2` and test for normality:

```
samp250=sample(pop2,250)
shapiro.test(samp250)

samp25=sample(pop2,25)
shapiro.test(samp25)

samp5=sample(pop2,5)
```

```
shapiro.test(samp5)
```

For `samp250`, you should be able to detect non-normality. For `samp25` you usually won't detect a departure from normality (run it a few times, refreshing the sample), and for `samp5`, you have no chance to ever detect departures from normality.

For large sample sizes, your distribution absolutely does not matter. The central limit theorem takes care of this. Your distribution of the means (and that's the true assumption to make the p-value calculation work) will be normal with a sample size of 250 and even with a sample size that is as small as 25 or 15.

In conclusion: These tests fail when they are most needed (lower sample sizes); instead of relying on these tests, we should use our good judgement.

You can develop that judgment by comparing what effect transformations have on your p-values. If you have your own data that shows normality in your residual, try a moderate transformation (or an intensive one to really skew your data). Then, recalculate your p-values. Did it make a difference? Generally, you may likely find that ANOVA is quite robust against violating assumptions of normality. However, you may come across cases where your p-values are off by a degree that you personally cannot tolerate. That's where you switch to non-parametric tests.

You can explore this via simulation. Let's take two normally distributed samples and create a treatment effect (+0.5 standard deviations). You might need to run this set of commands a couple of times before you get a nice visual treatment effect, nice normal distributions, and an appealing p-value. Replace the +0.5 with +0.7 or +1.0 to get stronger differences and smaller p-values.

```
VarA=sample(pop1,25)
VarB=sample(pop1,25)+0.5
boxplot(VarA, VarB)
t.test(VarA, VarB)
```

Now, once you have a nice p-value that you like (e.g. around 0.001, perhaps), apply a log transformation or an even stronger inverse transformation that should clearly throw your T-test (or ANOVA) off track:

```
boxplot(log(VarA), log(VarB))
t.test(log(VarA), log(VarB))

boxplot(1/(VarA), 1/(VarB))
t.test(1/(VarA), 1/(VarB))
```

How do your p-values differ between the normal and skewed distributions?

You should see that there is not too much of a difference between the p-values. This means that we can rely on the central limit theorem and don't have to get hung up on the assumption of normality if your sample size is large.