# Lab 8 Part III
# ANCOVA, Blocking

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

This lab gives you the opportunity to work your way through examples of analysis of covariance (ANCOVA), blocking in experimental designs, and the use of least squares means for treatment mean and standard error estimation. Download or enter the following dataset. It is a new version of the “lentil” dataset, but with a different experimental design at a third farm.

Here we have the plots located along a slope, and the experimenter suspected a gradient in soil nitrogen concentrations that might influence the results. Therefore, blocks were placed perpendicular to the slope direction (BLOCK), with B1 near the top of the hill and B4 at the bottom, and at each plot location several soil samples were taken, bulked for each plot, and sent to the lab for a nitrogen essay (NITROGEN). As response variable, YIELD was recorded as before:

| ID | VARIETY | BLOCK | YIELD | NITROGEN |
|---|---|---|---|---|
| F3-A-B1 | A | B1 | 520 | 0.8 |
| F3-A-B2 | A | B2 | 542 | 2.2 |
| F3-A-B3 | A | B3 | 558 | 3.4 |
| F3-A-B4 | A | B4 | 580 | 4.3 |
| F3-B-B1 | B | B1 | 476 | 1.1 |
| F3-B-B2 | B | B2 | 492 | 2.5 |
| F3-B-B3 | B | B3 | 524 | 3.8 |
| F3-B-B4 | B | B4 | 536 | 4.9 |
| F3-C-B1 | C | B1 | 484 | 1.0 |
| F3-C-B2 | C | B2 | 506 | 2.0 |
| F3-C-B3 | C | B3 | 524 | 3.2 |
| F3-C-B4 | C | B4 | 530 | 5.7 |

Start with some quick data visualization so that you understand the nature of the dataset.

```
library(ggplot2)
dat1=read.csv("lentil_blocked.csv")
head(dat1)

p1 =ggplot(dat1,aes(x=VARIETY,y=YIELD))+
  geom_boxplot()+
  theme_bw()
p1

p2=ggplot(dat1,aes(x=BLOCK,y=YIELD,col=VARIETY))+
  geom_point()+
  theme_bw()
p2
```

Now we can see all varieties on all blocks. However, there is some over-plotting. Use geom_jitter to fix this:

```
p3=ggplot(dat1,aes(x=BLOCK,y=YIELD,col=VARIETY))+
  geom_jitter()+
  theme_bw()
p3
```

Now, let's look at the effect nitrogen might have:

```
p4=ggplot(dat1,aes(x=NITROGEN,y=YIELD,col=VARIETY))+
  geom_jitter()+
  theme_bw()+
  stat_smooth(method="lm",se=F)
p4
```

## 8.7 Analysis of Covariance (ANCOVA)

You can use ANalysis of COVAriance if you have a covariate (i.e., a variable that influences your dependent variable, but that you cannot control experimentally). For example, in this field trial you may have variation in nitrogen levels or soil moisture. You can measure them, but you can't do much about it.

For this example, let's pretend that we used a completely randomized design for this experiment, and only later realized that the lentil plots near the bottom of the hill all grew much better than those closer to the top of the slope. You suspect that there is a nutrient gradient causing this. Fortunately, it is never too late to measure covariates, you can do that any time before, during (preferred), or even after your study, if need be.

Analysis of covariance combines analysis of variance and regression. If you can reduce the "noise" that is caused by the covariate, your "signal" to "noise" ratio in a statistical test (such as the F-test or T-test) becomes larger, making your test more powerful. You can do this by introducing a continuous variable as a covariate into your ANOVA model:

Explore the effect of adding a covariate. If your covariate accounts for a significant portion of the variance, your F-values for the treatment effects should become larger and your p-value smaller for the effect that you are interested in (i.e. your test should become more powerful to detect a difference among varieties A, B, and C). Normally, your statistical power should notably increase if the effect of the covariate on your dependent variable is also significant:

```
anova(lm(YIELD~VARIETY,data=dat1))
anova(lm(YIELD~VARIETY+NITROGEN,data=dat1))
```

Note that the `lm()` procedure distinguishes between treatments and covariates by means of the variable type. A numeric variable will be treated as covariate, and a treatment must be a factor (i.e. character variable). If you entered a treatment category as a number (i.e. 1 and 2 for farm1 and farm2), you have to convert it to a factor with the `as.factor()` command.

## 8.8 Blocking

Blocking is basically the same idea as ANCOVA, but your “covariate” is a class variable instead of a continuous variable. Again, you eliminate “noise”, so that your “signal to noise” ratio in a statistical test becomes larger, making your test more powerful. Blocks are typically generated by subdividing your experimental site into spatial units with similar environmental conditions. However, blocks can also be imposed as a “time” factor or as an “observer” factor, if you can’t do all measurements/observations by yourself or at one time.

Explore the effect of adding a block structure. If your blocking accounts for a significant portion of the variance, your F-values for the treatment effects (here, VARIETY) should become larger and your p-values should become smaller (i.e. your test becomes more powerful to detect a difference):

```
anova(lm(YIELD~VARIETY,data=dat1))
anova(lm(YIELD~VARIETY+BLOCK,data=dat1))
```

## 8.9 Least squares means and standard errors

You normally present ANOVA results in the form of a bar graph/dot plot to show treatment means with errors of the estimate, and a table (or letters in the graph) to indicate significant differences.

Now, the standard error of the means is different when calculated manually for individual treatment combinations versus when calculated through a proper analysis of variance implemented by procedures such as lm.

First, we calculate means and standard errors for individual treatment combinations with the `tapply()` function. Then, we use the least square means package (lsmeans) to extract the means and the standard errors of the estimate from the lm and lmer outputs:

```
# Calculate SE manually with tapply():

se=function(x) sd(x)/sqrt(length(x))
tapply(dat1$YIELD, dat1$VARIETY, mean)
tapply(dat1$YIELD, dat1$VARIETY, se)

# Now calculate SE with the lsmeans function accounting for block:

library(lsmeans)
out1=lm(YIELD~VARIETY+BLOCK,data=dat1)
out2=lsmeans(out1, ~VARIETY)
lsmeans (out1, ~VARIETY)
```

As you can see, there is a massive difference in the standard errors for treatment means, which have shrunken from around 10 to 14 kg/ha for individual treatments to just 2.7 kg/ha. The improvement in precision is due to the block effect being accounted for, rather than being part of the "noise". Also note that the standard errors for all treatment levels are equal (if the sample sizes are equal). This is as it should be, because ANOVA calculates a single, pooled error variance (hence the assumption of homogeneity of variances among treatment levels).

It is therefore **good practice to use means and standard error estimates from the Anova output for displaying in graphs**. Don't calculate them individually from your raw data. For factorial or blocked designs, this will normally make a big difference.

## 8.10 Mixed models

Ideally, we should use a *mixed model*, with fixed effects and random effects for this analysis. Fixed effect means that the treatment is controlled (fixed) by the experimenter. Random effects are effects where you do not care about their significance (such as a block treatment meant to account for noise).

The `lm()` procedure cannot handle mixed models and we have to install the lme4 or the extended lmerTest mixed model package.

The equivalent to `lm()` in the lme4 or lmerTest packages is the procedure `lmer()`. The syntax is basically the same, except that random effects are indicated by brackets, the number one, and a vertical divider as shown in the code below. The `anova()` prefix returns an ANOVA table (with the random effects missing). Rather than calling the library lme4, we use the library lmerTest, which is a wrapper for lme4 that provides p-values as well.

One of the key theoretical advantages of mixed models is that means and variances of random effects are not estimated, and therefore you may increase your statistical power in some designs. Compare the results of your mixed model with the fixed effects model above. For normal experiments and balanced designs, there is no difference.

`lsmeans` and `cld` have a lot of handy functions for multiple pairwise comparisons (see previous labs) that work with both `lm` and to some degree with `lmer` outputs.

Please note that while you can get LSMeans estimates with a slightly different syntax for a mixed model (code below), **the standard error estimates are incorrect**. That is by design, because the means of random effects are not estimated, and thus cannot be accounted for. Get the standard errors of your least squares means estimates from a fixed effects model (i.e., use the SE values from the "out2" object produced from the lm model above).

```
library(lmerTest)
out3=lmer(YIELD~VARIETY+(1|BLOCK),data=dat1)
out4=lsmeans(out3, ~VARIETY)
out4 # Note the "wrong" standard errors, not accounting for BLOCK!
cld(out4, adjust="tukey", Letters=letters, sort=F)
```