Nested Designs

NRES 710

Last compiled: 2024-11-21

Read **Hurlbert 1984** paper for next class when we discuss Pseudoreplication. It is really long! Cited many many thousands of times. Read from bottom of page 198 (‘Pseudoreplication in manipulative experiments’) until the end.

Review Quiz 7

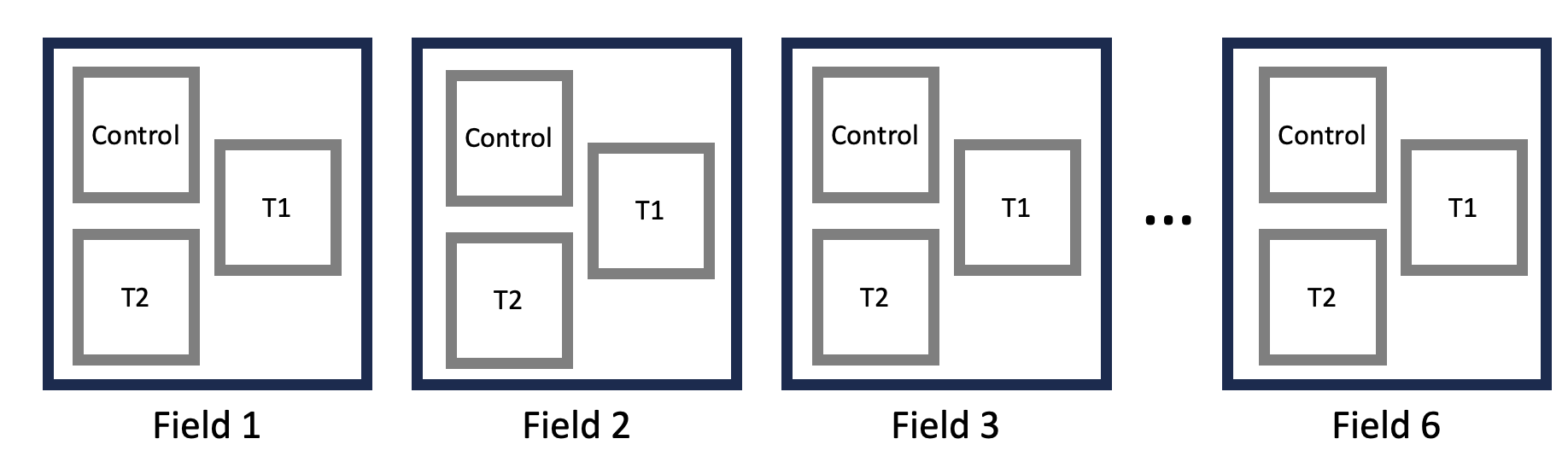
## Experimental Designs

So we have talked about some different experimental designs, and I want to continue on this but by introducing some new ones.

### Randomized Block Designs

One powerful approach to experimental design that we have discussed is using the **Randomized Block Design**. Let’s consider an example where we have six fields; within each field we have three plots, and plots are randomly assigned to one of three treatments: control, treatment 1, and treatment 2. Here, we have 6 fields with 3 treatments; technically 18 plots in total.

*Brian draw this on the board: field 1, 2, 3, … 6.; with three plots within each field*



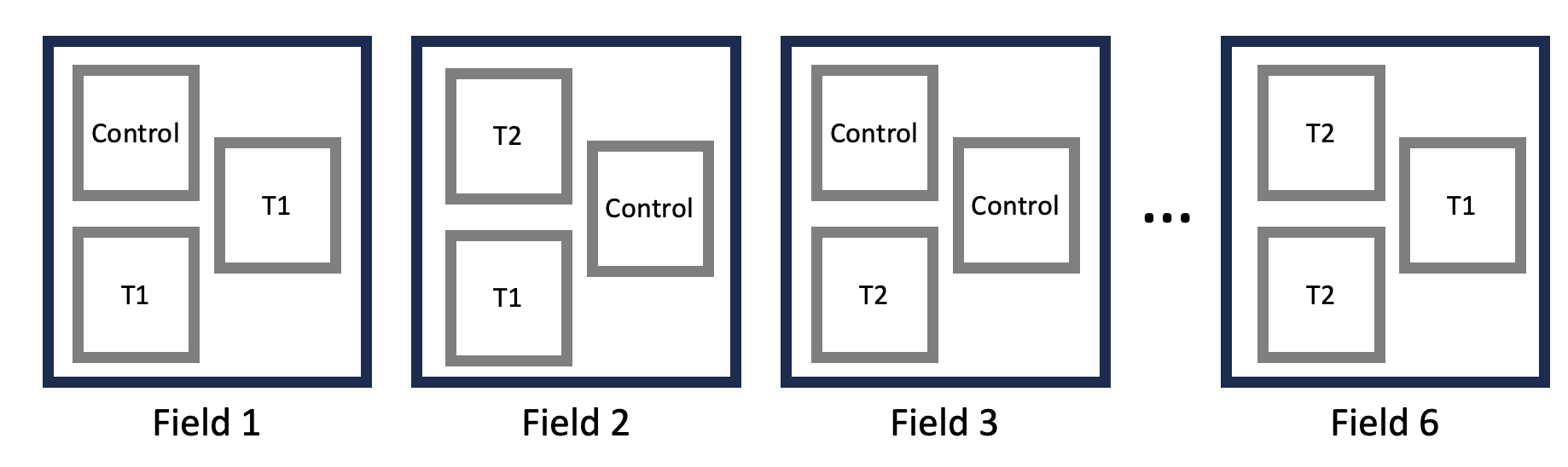
**We ensure that every field receives every treatment;** this is a ‘fully factorial design’. We randomly decide what the first plot will be treated with in field 1 (e.g., Treatment 1), then with two remaining plots, we randomly determined whether the other two plots will be Control or Treatment 2. This is a randomized block design, and it is very powerful. You don’t have to include ‘field’ in your analysis… but by doing so, it will increase your power. This is something that we want to try for. But, it can be hard to do.

Any questions about this design?

### Completely Randomized Design

Let’s talk about another design: a **completely randomized design**. Here, we ignore the blocks (our fields), and instead just randomly assign a treatment to any given plot. We have lumped our blocks together and randomly assign treatments to each plot.

*Brian draw this on the board: fields still have three plots, but (1) they don’t necessary have all three treatments, (2) instead each plot is randomly assigned treatment independent of the block, such that (3) some fields have mostly controls, others mostly different treatments*



We are not ensuring that every field gets every treatment. This results in an **unbalanced design.**

This is a ~fine design, and people use it all the time. In the analysis phase, you can still include ‘field’ as a random effect.

**Q:** What might be a limitation of this design?

**Q:** What if most of the fields that have ‘control’ plots are also slightly wetter, while other fields with treatments are slightly dryer?

*Brian draw dry, dry, wet, …, dry above fields on the board.*

This would introduce **collinearity**! Collinearity makes it difficult to distinguish effects from one another. In this case, it would be difficult to distinguish the effects of individual **fields** from effects of **treatments within the fields**. The **unbalanced design** here introduces collinearity between fixed treatment effects and random field effects.

For example, if **wet fields** have more biomass productivity, then it might make it look like control treatments have more biomass. In this case, the field effect becomes a ***confounding effect***: it has it’s own effect, which is collinear with our treatment.

**Q:** If we have 30 fields, is there likely to be a confounding effect due to collinearity? **No.** There could still be collinearity, but if we increase our sample size to be very large, the confounding effects of collinearity will ~disappear – it will ‘wash out’ due to the large sample. Field’s will still have their own effects, but it becomes less of a confounding effect because the collinearity decreases across the larger, randomized sample. But with only six fields, there’s more likely to be confounding effects due to collinearity, because of the small sample size.

**Q:** How did we deal with collinearity before?

* We made sure that all of the X-variables were in the model; we fit different effects to statistically control for one another.
* If we try to do that in the case illustrated on the board, we would we hope to estimate both a field effect and a treatment effect while statistically controlling for eachother.
* However, we might encounter a problem here. Remember that I said that the mixed-effects model were fit using REML… REML involved estimating all the fixed effects first, and then the random effects second. If we have collinearity, it may be unable to give us unbiased betas for fixed effect of treatment, because it has not yet statistically controlled for the effects of field. **It’s cannot accounting for the effect of field when estimating those fixed-effect betas**.

**Disclaimer:** I have not yet built a simulation to demonstrate this, like I did for our collinearity lecture a few weeks ago. I might try to do this tomorrow. I’ll report back.

**But the point of all this is:** while there are cases where you could analyze this design, doing so has limitations and creates problems. In this case, accounting for the field as a random effect does not necessarily solve the problems we have identified.

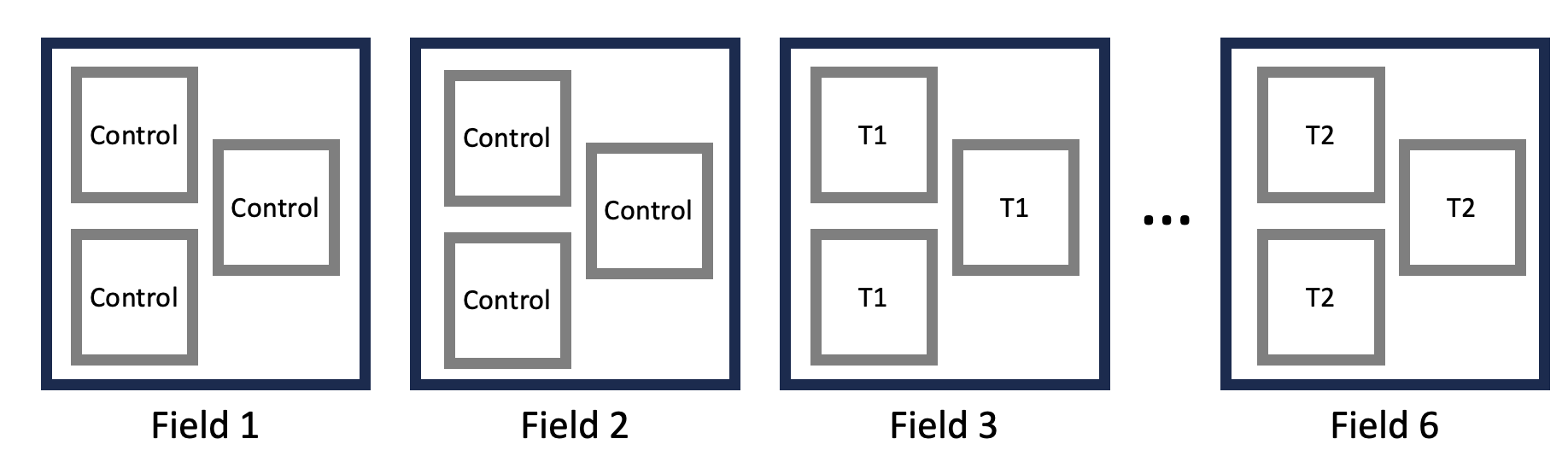
In this case, having lots of fields would reduce collinearity. But the best way to fix this problem is to fix it before we start our experiment… and design the experiment with a randomized-block design.

**Questions?**

## Pseudoreplication

Let’s say we still have six fields with three plots per field. But let’s say we didn’t consult with a statistician or your advisor before you planned your experiment… and you designed it like this:

*Brian use six-fields on board but now they have all the controls in one field, treatment 1s in one field, etc.*



Basically, we randomly determined what each field was going to get, and then each plot got the same treatment within each field.

**Q:** Can anyone see a problem with this design?

In the end, we will have 18 rows in our Excel file, one for each plot. But, is your sample size really 18? **No!**

This is what we might call ‘**pseudoreplication**’! Pseudoreplication is…

**Pseudoreplication – analysis assumes you have more samples than you really do.**

You might be asking yourself: why aren’t these three samples different? They have different measures for biomass for each sample…?

The problem is that, let’s say the average effect on biomass in the **three plots in Field 1** is really **high**. How do we know whether the biomass is high because of the **control effect** or because of the **field effect**? We don’t. And that’s why we can’t treat them as separate samples.

Easy **rule of thumb** for identifying **pseudoreplication** within samples: your sample is the **largest grouping you can create within a given group/treatment**. Here, the smallest grouping we can make is the plot, and the largest grouping within a given treatment is the field. Because we can group all of these ‘samples’ together in each field, then our sample is not the ‘plot’, but rather it is this largest group – the field.

**Q:** We have 18 ‘samples’, but I actually just said that we only truly have 6 samples. **How can we fix this?** What’s an easy way to fix this?

* Take an average! Average of the three plots in each field, and make that be our single ‘sample’ for each field. In our Excel file, we would replaced the three rows for each field with a single field.
* If we do this, we will have six rows.
* **Q:** If we average the three samples within each plot, can we include field as a random effect with those data? **No, because then we would only have 1 row per field.**
* **This would be a perfectly valid way of analyzing these data that eliminates pseudoreplication.** But there is a better way to do it!

The better way to analyze these data would be to **include a field effect**. This model would look like:

When we do this, R (and most stats packages) will **automatically recognize** the pseudoreplication structure in your data, and it will automatically use the **field error** as the residual error when it **calculates the effect of treatment**. In this case, it’s not the between-plot error that is your residual error for calculating the effect of treatment. Rather, it is your **between-field error** that is your residual error for **testing the effect of treatment**, because the field is your sample.

In general, it is the between-sample error that is your residual error for calculating p-values for any fixed-effect. In this case, since we cannot distinguish field effects from treatment effects within each field, our test for treatment effects requires the ‘between-field error’ as the between-sample error for that test – not the between-plot error.

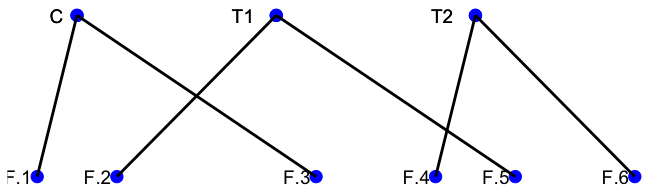
## Nested Designs

There is a term for this statistical model, which is the **‘Nested Design’**!

**Nested Design – e.g., fields are nested within treatments**

In this particular case, the fields are ‘nested’ within the treatments. An easy way to visualize this is by listing out the treatments (C, T1, T2), the fields, and connecting them with lines:

*Brian draw on fresh board*

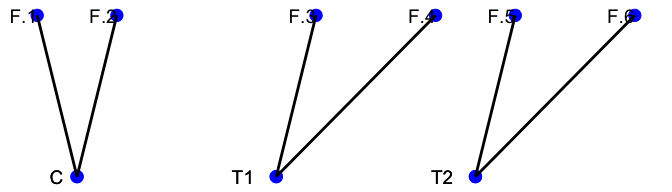


For a nested design, the hierarchy should always *spread out* from the top down to the bottom.

* The thing that there is more of is nested inside of the thing that there is less of. Fields are nested within treatments.
* It’s almost always a random effect that is nested within the fixed effect… although this can get complicated, which we will see next class.

Let’s say alternatively, you did this the other way around. You thought the fixed treatment effect was nested within the field effect, it would look like this:

*Brian draw this on the board*



This doesn’t work because it spreads out upward. This doesn’t work, so we have set this up incorrectly.

### R handles this for us

One cool thing is that you don’t necessarily need to recognize that there is nesting structure in your data – you just have to include that field effect! Just like with collinearity and swamping, you don’t have to recognize these features, you just have to include all of the X-effects. Same thing with pseudoreplication – you don’t have to recognize it, you just have to include all of the appropriate random variables.

But, it is important to understand what **pseudoreplication is** and **how it exists**. When we have a nested design, we automatically get pseudoreplication. **Nesting happens when some random samples only get certain fixed effects**. Some fields only get controls, some only get treatment 2, etc. That’s the nesting.

Notice that nesting and collinearity are not quite the same.

* For the **completely randomized unbalanced design**, this was not nesting, because each field did get more than one treatment in most examples.
* Nesting is essentially **‘perfect collinearity’** – each field only got a certain type of treatment.

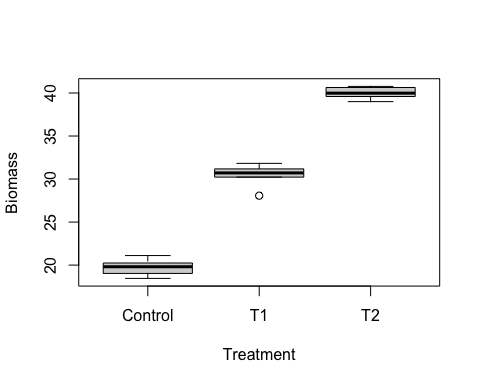
The most simple design – one plot per field – is also like ~nesting. But, there’s no pseudoreplication, we don’t have more than one sample per field, so we can’t include a field effect, and therefore we don’t have to worry about trying to distinguish between the field effect and the treatment effect, because we only have 1 treatment per field.

## Example in R

For today’s dataset, we have six fields, each field has three plots, each plot received the same treatment within each field. You can examine how the data were simulated in the code for ‘Truth’. **Truth** is:

Let’s examine the data:

### Nested design analysis in R  
  
# Read the data  
datum <- read.csv("lecture\_18\_dataset1.csv")  
  
# Examine it  
head(datum, 10)  
  
# Make 'Treatment' a factor  
datum$Treatment <- as.factor(datum$Treatment)  
  
# Plot it  
plot(Biomass ~ Treatment, data = datum)



Let’s look at a simple ‘lm()’ first:

# Simple linear model  
results <- lm(Biomass ~ Treatment, data = datum)  
summary(results)

##   
## Call:  
## lm(formula = Biomass ~ Treatment, data = datum)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -2.384 -0.377 0.108 0.611 1.369   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 19.74 0.41 48.2 < 2e-16 \*\*\*  
## TreatmentT1 10.71 0.58 18.5 9.9e-12 \*\*\*  
## TreatmentT2 20.25 0.58 34.9 8.7e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 1 on 15 degrees of freedom  
## Multiple R-squared: 0.988, Adjusted R-squared: 0.986   
## F-statistic: 611 on 2 and 15 DF, p-value: 4.25e-15

We get good estimates of effects that include Truth within their uncertainty. But we have pseudoreplicated here! The summary output says that we have 15 degrees of freedom, which tells us that the analysis is using all 18 samples to calculate p-value using our residual error. Our p-values are incorrect, they are smaller than they should be, because we have assumed we have more samples than we really do.

If we include a random effect of field, we can estimate this all using just the 6 samples, our fields:

# Linear mixed-effects model  
library(nlme)  
results2 <- lme(Biomass ~ Treatment, data = datum, random = ~1|Field)  
summary(results2)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Field  
## (Intercept) Residual  
## StdDev: 4.771e-05 1.004  
##   
## Fixed effects: Biomass ~ Treatment   
## Correlation:   
## (Intr) TrtmT1  
## TreatmentT1 -0.707   
## TreatmentT2 -0.707 0.500  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.3750 -0.3750 0.1078 0.6083 1.3639   
##   
## Number of Observations: 18  
## Number of Groups: 6

When we examine the summary output, one thing that’s noteworth is that it gives us a ‘Degrees of Freedom’ column that we have not seen before. This is because when you run an LME(), it’s looking for these nested designs. R goes in and asks ‘do you have nesting in your data?’, and if it detects it, R attempts to use the appropriate random deviation structure as the error for each term.

Look at the bottom:

* Number of observations: 18
* Number of groups: 6

More importantly, in looking at the effect of treatment, the degrees of freedom has decreased from 15 in the LM() down to 3 (!) in the LME().

‘Degrees of freedom’ is an estimate of our sample size minus the number of comparisons we are exploring. So in this case, we have 6 samples (our fields) and three treatment groups (control, T1, T2), so: 6 - 3 = 3 DF.

The p-values are much larger. Previously, they were e-12 and e-16, now they are e-4 and e-4. We lost a lot of power. But that’s the way it should be! **We don’t have 18 samples to test for the effects of treatment; we only have 6.**

The cool thing is that we don’t have to tell R that we have pseudoreplication or nesting. We just provide the random effect of field, and **R fixes the estimation automatically.**

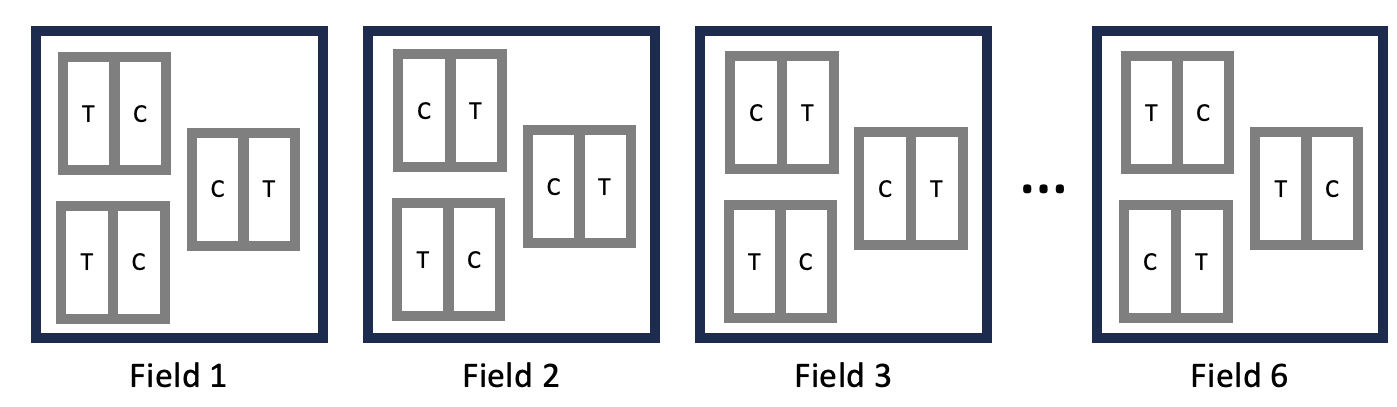
Make sure you check your degrees of freedom to make sure everything lines up. The degrees of freedom needs to be *less* than the number of **samples** you have. If it’s larger than the number of samples, something isn’t right.

## Another nesting example

Nesting doesn’t always generate pseudoreplication, but it is important for us to understand nesting for other reasons as well.

In this case, we have 6 fields, 3 plots in each field, and each plot is sub-divided and sub-plots are randomly assigned to ‘control’ or ‘treatment’.

*Brian draw this on the board*



**Q:** What kind of design is this? **Randomized block design**. We have plots that are ‘blocked’, where we randomly assign treatments within; all treatments are assigned within each block. (Also called a **‘split-plot design’**.)

**Q:** When you run this analysis, what will your block be? Is it a field, or a plot?

In truth, we really are most interested in the **plots**. And we could run this model:

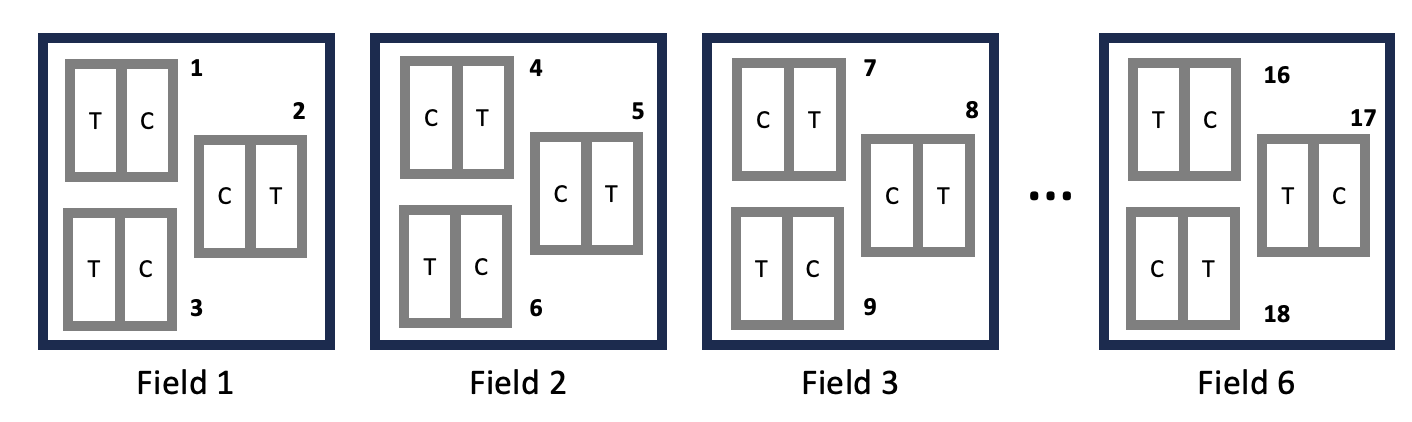
We want to include the error term due to plot because: plots can be different from each other. Some plots might be wet, others might be dry. Some might get more sun, other less. Etc. There might be variation among plots, and we want to account for that. Our plots are our blocks; within each block, we have a random assignment of treatments.

But you might say: but what if this whole field is wet? Or what if this whole field has rich soils?

In truth, **plots are nested within fields**. These three plots are only in field 1. These three plots are only in field 2. Etc.

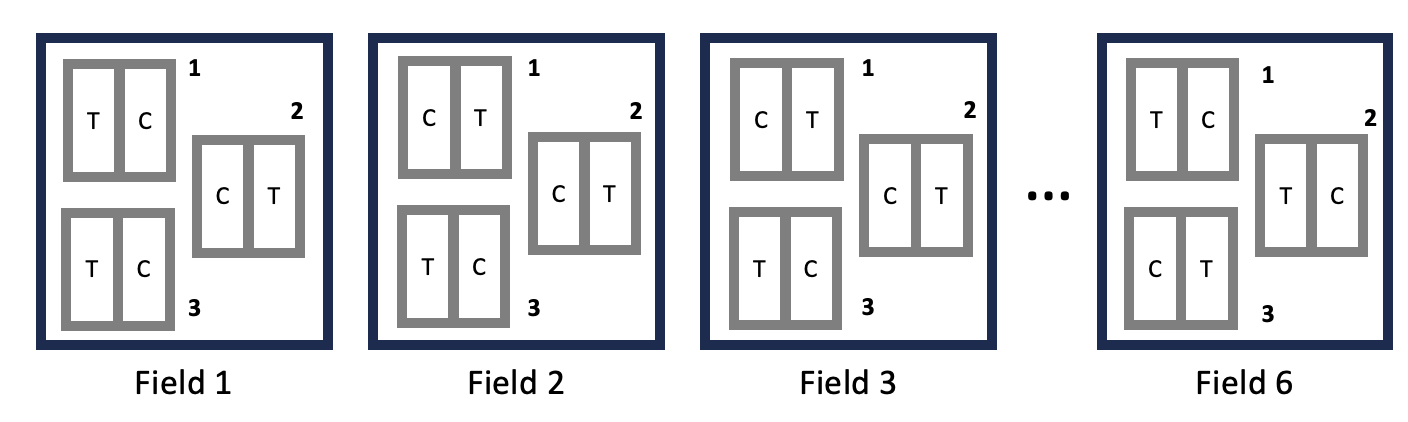
Do we also need to account for error due to field? ***It depends!*** It depends how you identified different plots numerically.

If we identified each plot in our dataset with 1, 2, 3, … 18, like this:



Then all we have to do is include an . This model works pretty well! But it cannot partition the error due to field from the error due to plots.

But sometimes people identify the plots differently:



Within each field, the three plots are numbered 1, 2, and 3. This can be problematic!

The problem here is that when we tell R to estimate an error due to plot, what’s it going to assume? It will group all of the 1s together, and all of the 2’s together, etc. It will thing all of the 1 samples came from the same plot – but they aren’t. In truth, this is plot 1 in field 1, plot 1 in field 2, plot 1 in field 3, etc. In this case, we **have to communicate this to R**, we have to tell R to identify the additional nested structure of our data: fields > plots > treatments.

Plots are nested within fields. We can communicate this with:

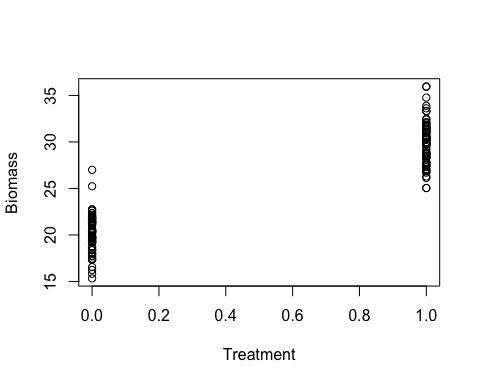
This is telling R that we have random error due to fields, and then within fields, there is also random error due to plots. *Plots are nested within fields*. R annotates this as with ‘field | plot’, so we can annotate it similarly.

Let’s look at some data I have created:

### Another example analysis in R  
# Read the data  
datum <- read.csv("lecture\_18\_dataset2.csv")  
  
# Examine it  
head(datum, 15)  
  
# Structure  
str(datum)

## 'data.frame': 144 obs. of 7 variables:  
## $ Field : int 1 1 1 1 1 1 2 2 2 2 ...  
## $ Plot : int 1 1 2 2 3 3 1 1 2 2 ...  
## $ PlotOrder : int 1 1 2 2 3 3 4 4 5 5 ...  
## $ Treatment : int 0 1 0 1 0 1 0 1 0 1 ...  
## $ FieldError: num -0.56 -0.56 -0.56 -0.56 -0.56 ...  
## $ Error : num 2.187 1.533 -0.236 -1.026 -0.71 ...  
## $ Biomass : num 20.4 29.7 15.8 25 20.4 ...

# Plot it  
plot(Biomass ~ Treatment, data = datum)



Now, I am going to have to analyze this in a specific way – because of the way I have numbered my plots. I have numbered the plots 1-3 within each field. The plots numbered 1 in the first field are not the same as the plots numbered 1 in the second, third, fourth, etc. fields. I need to tell R that, and specify that plots are nested within fields to correctly partition the variance.

# Linear mixed-effects model: incorrect random-effect structure for this design  
results <- lme(Biomass ~ Treatment, data = datum, random = ~1|Plot)  
summary(results)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Plot  
## (Intercept) Residual  
## StdDev: 0.05162 2.153  
##   
## Fixed effects: Biomass ~ Treatment   
## Correlation:   
## (Intr)  
## Treatment -0.702  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -2.33697 -0.64228 0.08825 0.61703 3.16156   
##   
## Number of Observations: 144  
## Number of Groups: 3

# This says: only 3 groups (plots). But we had more! This is not specified correctly; this is assuming a different structure than we really have.

Look at the analysis results. It says there are only ‘3 groups’ – or 3 plots! We have many more plots than this.

For treatment, it says we have 140 degrees of freedom. This is fine! We have 144 observations, and it’s using every single one of these observations when calculating the effect of treatment. This is fine because we have a paired, treatment and control blocking design with no pseudoreplication. So this is working well.

But it’s not giving us a good estimate of plot. It is saying there are only three plots with many replicates within each plot. This causes us to lose power, because some of the variation due to plot is not accurately being recorded and partitioned. The true residual variation was 1, but it’s saying it’s 2.15.

What if we run it using the ‘PlotOrder’ variable, that numbers plots among all the fields as 1, 2, 3, 4, 5, … up to 72? This has no information about ’Field.

# Linear mixed-effects model: okay, but non-ideal random-effect structure for this design  
results2 <- lme(Biomass ~ Treatment, data = datum, random = ~1|PlotOrder)  
summary(results2)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | PlotOrder  
## (Intercept) Residual  
## StdDev: 1.891 1.031  
##   
## Fixed effects: Biomass ~ Treatment   
## Correlation:   
## (Intr)  
## Treatment -0.339  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.99822 -0.44024 -0.03219 0.45261 2.53862   
##   
## Number of Observations: 144  
## Number of Groups: 72

# This says: only 72 groups (1 for each plot). This is correct! But it fails to measure partition random error due to field.

To get this model correct, we need to run it like this:

# Linear mixed-effects model: most correct random-effect structure with plots nested within fields  
results3 <- lme(Biomass ~ Treatment, data = datum, random = ~1|Field/Plot)  
summary(results3)

## Linear mixed-effects model fit by REML  
## Data: datum   
##   
## Random effects:  
## Formula: ~1 | Field  
## (Intercept)  
## StdDev: 0.4442  
##   
## Formula: ~1 | Plot %in% Field  
## (Intercept) Residual  
## StdDev: 1.839 1.031  
##   
## Fixed effects: Biomass ~ Treatment   
## Correlation:   
## (Intr)  
## Treatment -0.325  
##   
## Standardized Within-Group Residuals:  
## Min Q1 Med Q3 Max   
## -1.96582 -0.43215 -0.02867 0.44220 2.53281   
##   
## Number of Observations: 144  
## Number of Groups:   
## Field Plot %in% Field   
## 24 72

“Random = tidle 1 | with plots nested within field”. We need to tell R that plots are nested within fields.

Now things are different! At the bottom, we now see that we have 144 observations, 24 fields, and *72 plots %in% fields*! So it is now correctly recognizing that we have 72 plots now.

We only have 71 degrees of freedom, as we have lost degrees of freedom due to the large number of plots. But that’s okay.

Now our residual variation is good: it’s now ~1, the ‘plot in field’ error is ~2, and the field error is 0.44. The field error is a little low (truth was 1), but this is likely due to random chance.

* Note: from time to time, random variation in the data causes our estimates to be different from Truth. What do we call this? Type I error! I simulated the data again with a different seed, and then it correctly estimated the variances to field, plot in field, and residual.

Most importantly, look what happened to our power. The larger the t-value, the more power we have. Before, our t-value for the treatment effect was 27. Now it is 57! We have ~doubled our power.

**Questions?**