

STA 610L: MODULE 3.4

NESTED VS NON-NESTED RANDOM EFFECTS

DR. OLANREWaju MICHAEL AKANDE

NESTED VS NON-NESTED GROUPING

For the most part so far, we have only considered hierarchical models with one grouping factor.

Sometimes however, we may have to incorporate multiple grouping factors.

Broadly speaking, units at a certain level in a hierarchical specification are **nested** within a grouping variable if each unit belongs to a unique level of that variable.

Conversely, units at a certain level in a hierarchical specification are **non-nested or crossed** within a grouping variable if each unit belongs to multiple levels of that variable.

It is possible to have hierarchical structures that are a combination of both.

Let's look at some hypothetical examples to get a better idea.

NESTED VS NON-NESTED GROUPING

Example I: suppose we are studying students within classrooms within schools within counties.

Here, each student belongs to a unique classroom, each classroom belongs to a unique school, and each school belongs to a unique county.

We then have nesting at every level of this model.

Note: If you could somehow move the classes across schools, then each class would belong to multiple schools, so that you no longer have nesting at that level.

However, schools will remain nested within counties.

NESTED VS NON-NESTED GROUPING

Example II: suppose we have data on earnings for individuals, which are collected by different job categories but also by states.

If we assume the job categories do not overlap, then each individual are nested within job categories, so that each one belongs to a unique job-state combination

However, job categories are still shared (and thus non-nested) across states.

In this example, we have nesting at the first level but not at the second.

In practice, job categories actually do overlap, so that this becomes a clear example of non-nested grouping factors at multiple levels.

In any case, it is relatively straightforward to extend the models we have covered so far to these scenarios with more grouping variables, as long as we are careful about how to implement them (in R or any other programming language for that matter).

EXAMPLE

Consider a study in the semiconductor industry of variability in manufacture of silicon wafers.

For each wafer, the thickness of the oxide layer is measured at three different sites. The wafers themselves are sampled from eight different production lots. In this experiment, sites are nested in wafers, and wafers are nested in lots.

```
data(Oxide,package="nlme")
head(Oxide,10)
```

```
## Grouped Data: Thickness ~ 1 | Lot/Wafer
##   Source Lot Wafer Site Thickness
## 1      1  1    1    1    2006
## 2      1  1    1    2    1999
## 3      1  1    1    3    2007
## 4      1  1    2    1    1980
## 5      1  1    2    2    1988
## 6      1  1    2    3    1982
## 7      1  1    3    1    2000
## 8      1  1    3    2    1998
## 9      1  1    3    3    2007
## 10     1  2    1    1    1991
```

The site index repeats across wafers; wafer index repeats across lot. Lots are sort of nested within `Source` but we ignore that for this illustration.

MODEL

Let's consider a random effect for lot and a random effect for wafer in the model

$$Y_{ijk} = \gamma_0 + \alpha_i + \beta_{ij} + \varepsilon_{ijk}$$

where $\alpha_i \stackrel{iid}{\sim} N(0, \sigma_\alpha^2) \perp \beta_{ij} \stackrel{iid}{\sim} N(0, \sigma_\beta^2) \perp \varepsilon_{ijk} \stackrel{iid}{\sim} N(0, \sigma_\varepsilon^2)$.

Here i indexes the lot, j indexes the wafer within lot, and k indexes the site within wafer.

EXERCISE!

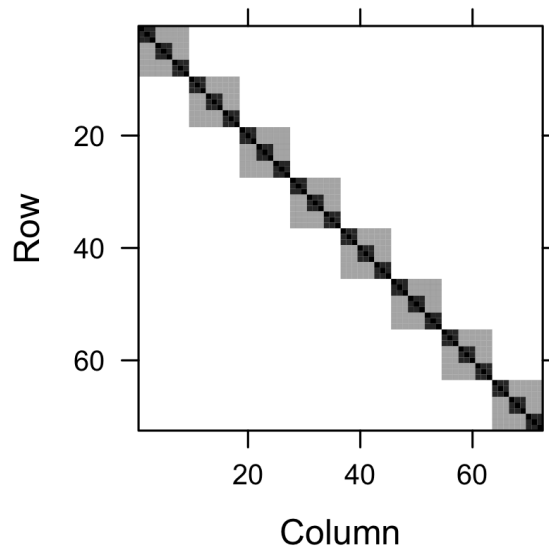
For this model, try to derive the following quantities by yourself.

- $\text{Var}(Y_{ijk})$
- $\text{Cov}(Y_{ijk}, Y_{ijk'})$ (different sites on same wafer in same lot)
- $\text{Cov}(Y_{ijk}, Y_{ij'k})$ (same lot, different wafer, site k)
- $\text{Cov}(Y_{ijk}, Y_{ij'k'})$ (same lot, different wafer, site k and site k')
- $\text{Cov}(Y_{ijk}, Y_{i'jk})$ (different lots)

Using the data ordering, put these values (and others) together to show the form of the full matrix $\text{Cov}(Y)$.

COVARIANCE STRUCTURE

Using the ordering in the data across the 72 oxide layer thickness measurements, we expect the covariance matrix to have the following block structure in lots and wafers.



Dimensions: 72 x 72

The darker, smaller squares (higher correlations) are for measures taken on the same wafers, and the larger squares are for measures taken in the same lot. Measures from different lots are independent.

MODEL

We need to be careful about specifying the model because the indices are nested.

We want a random effect for wafer and a random effect for lot, but we need to be careful about how we specify it because wafer 1 in lot 1 is not the same wafer as wafer 1 in lot 2.

To fit a model using the nested indices provided, we use the following code.

```
#specifying that wafer index is nested in lot  
ox1 <- lmer(Thickness ~ 1 + (1|Lot/Wafer), data = Oxide)  
summary(ox1)
```

If the lots had just been numbered differently per lot, the model above would be equivalent to

```
ox2 <- lmer(Thickness ~ 1 + (1|Lot) + (1|Wafer), data = Oxide)  
summary(ox2)
```

MODEL

Here is the first model:

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Thickness ~ 1 + (1 | Lot/Wafer)
##   Data: Oxide
##
## REML criterion at convergence: 454
##
## Scaled residuals:
##   Min      1Q  Median      3Q      Max
## -1.8746 -0.4991  0.1047  0.5510  1.7922
##
## Random effects:
##   Groups      Name                Variance Std.Dev.
## Wafer:Lot (Intercept)  35.87      5.989
## Lot          (Intercept) 129.91     11.398
## Residual                    12.57      3.545
## Number of obs: 72, groups: Wafer:Lot, 24; Lot, 8
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 2000.153      4.232   472.6
```

Wow, the lot explains a lot of the variability in the response! There is considerable variability across wafers as well.

MODEL

What if we used the second code?

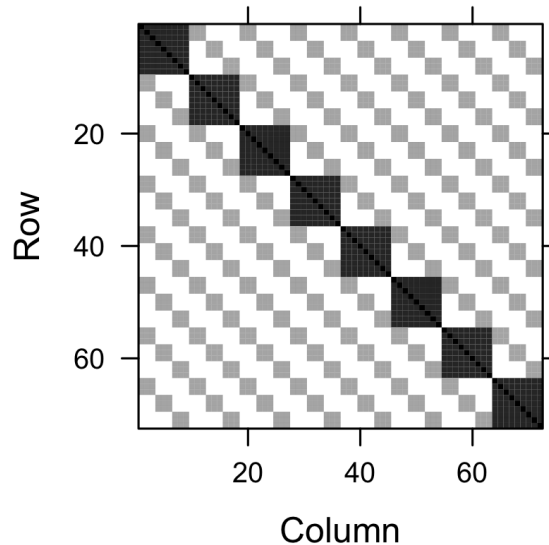
```
ox2 <- lmer(Thickness ~ 1 + (1|Lot) + (1|Wafer), data = Oxide)
summary(ox2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: Thickness ~ 1 + (1 | Lot) + (1 | Wafer)
## Data: Oxide
##
## REML criterion at convergence: 490.6
##
## Scaled residuals:
##   Min       1Q   Median       3Q      Max
## -2.6115 -0.4268  0.1087  0.3975  2.2815
##
## Random effects:
## Groups   Name                Variance Std.Dev.
## Lot      (Intercept) 138.998  11.790
## Wafer    (Intercept)   1.493   1.222
## Residual                    38.349   6.193
## Number of obs: 72, groups: Lot, 8; Wafer, 3
##
## Fixed effects:
##              Estimate Std. Error t value
## (Intercept) 2000.15      4.29    466.2
```

Well, the wafer effect went away, and the residual variance got larger. What happened?

MODEL

The model assumed wafer 1 was repeated in all 8 lots, wafer 2 was repeated in all 8 lots, etc. so that there were only 3 wafers (instead of 24). This watered down the wafer effect (wrong model!) and estimated a correlation that looks more like this.



Dimensions: 72 x 72

Yikes, observations from different lots should be independent, but we induced them because of the way the wafer index is coded in the data.



MINOR MODIFICATION

If you don't like using the nesting coding, we can fix the issue with the index for wafer and use our regular coding.

Below we make the index on wafer unique by appending it to the lot -- so the first digit of the wafer2 index designates lot number, and the 2nd digit designates the wafer within the lot.

```
#library(tidyverse)  
Oxide <- mutate(Oxide, Wafer2 = as.numeric(paste0(Lot, Wafer)))
```

MINOR MODIFICATION

```
head(Oxide, 15)
```

```
## Grouped Data: Thickness ~ 1 | Lot/Wafer
##      Source Lot Wafer Site Thickness Wafer2
## 1         1   1     1    1      2006     11
## 2         1   1     1    2      1999     11
## 3         1   1     1    3      2007     11
## 4         1   1     2    1      1980     12
## 5         1   1     2    2      1988     12
## 6         1   1     2    3      1982     12
## 7         1   1     3    1      2000     13
## 8         1   1     3    2      1998     13
## 9         1   1     3    3      2007     13
## 10        1   2     1    1      1991     21
## 11        1   2     1    2      1990     21
## 12        1   2     1    3      1988     21
## 13        1   2     2    1      1987     22
## 14        1   2     2    2      1989     22
## 15        1   2     2    3      1988     22
```

MINOR MODIFICATION

```
#now we can also use the coding we're used to  
ox3 <- lmer(Thickness ~ 1 + (1|Lot) + (1|Wafer2), data = Oxide)  
summary(ox3)
```

```
## Linear mixed model fit by REML ['lmerMod']  
## Formula: Thickness ~ 1 + (1 | Lot) + (1 | Wafer2)  
## Data: Oxide  
##  
## REML criterion at convergence: 454  
##  
## Scaled residuals:  
##      Min       1Q   Median       3Q      Max  
## -1.8746 -0.4991  0.1047  0.5510  1.7922  
##  
## Random effects:  
## Groups   Name                Variance Std.Dev.  
## Wafer2   (Intercept)    35.87     5.989  
## Lot      (Intercept)  129.91    11.398  
## Residual                    12.57     3.545  
## Number of obs: 72, groups: Wafer2, 24; Lot, 8  
##  
## Fixed effects:  
##              Estimate Std. Error t value  
## (Intercept) 2000.153      4.232   472.6
```

Same result as before!

Easy to incorporate more "grouping factors" using the same approach. Just be careful with the way your data is indexed.



WHAT'S NEXT?

MOVE ON TO THE READINGS FOR THE NEXT MODULE!