Salinity: variance components

Philip Dixon

8/30/2020

salinity.r: estimating variance components

There are two libraries for random effects and mixed models, nlme with the lme() function and lme4 with the lmer() function. lme() fits a larger collection of models but lmer() is much faster and easier to use. We will use lme() when we discuss repeated measures later in the semester. We use lmer() now. And, emmeans() for post-processing the model fit.

```
library(lme4)
library(emmeans)
```

Read the csv file and create a factor version of the treatment variable. This file has 5 columns: trt, container number, plant number, weight, and short. short is a variation on the container number discussed below.

```
salinity <- read.csv('../data/salinity.csv', as.is=T)
salinity$trt.f <- as.factor(salinity$trt)
```

In this data set, each container has a unique number: 1,2,3,4,5, and 6, so it is easy to specify the random effect. Fixed effects are specified just as in lm(). Random effects go in (). The syntax is very general. Inside the (), you specify what parameter is altered by the random effect and what defines the levels of that random effect. For the salinity data, the random effects of container increase or decrease the mean, i.e. the intercept. That is (1 | container). 1 because we are changing the intercept and it varies by container. Almost all of the random effects will have the form (1 | something).

The variables used on the right hand side of the random specification can be factors or not. lmer() looks at each unique value.

```
sal.lme <- lmer(weight ~ trt.f + (1|container),
  data=salinity)
sal.lme
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: weight ~ trt.f + (1 | container)
##    Data: salinity
## REML criterion at convergence: 41.1584
## Random effects:
##  Groups    Name        Std.Dev.
##  container (Intercept) 1.9079
##  Residual              0.5571
## Number of obs: 18, groups:  container, 6
## Fixed Effects:
## (Intercept)       trt.fh       trt.fl
##       9.315       -5.287       -4.472
```

You can get the variance components either by printing the lmer() result or using summary on it. The variance components are given under the Random effects: header. Printing the object just gives you the standard deviation = sqrt(Variance) for each random effect, including the residual. Summary() gives you both the variance and its square root. Note: the std. dev is not the precision of the estimated variance component. It's just the square root calculated for your convenience.

All the usual "after the model fit" functions can be used here. As with the ANOVA models last week, the emmeans library simplifies getting most of the post-ANOVA results.

In particular, you will see that anova() applied to an lmer object calculates the F statistic but doesn't give you a p-value. That's because computing the appropriate denominator df can be hard (conceptually and computationally), and anova() is chicken.

The joint_tests() function in the emmeans library uses a widely-accepted approximation called the Kenward-Rogers approximation. This is a generalization of another widely used approximation, the Satterthwaite approximation. For most problems we see in this class, the two are identical. When they're not, the K-R is more appropriate.

```r
summary(sal.lme)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: weight ~ trt.f + (1 | container)
##    Data: salinity
## 
## REML criterion at convergence: 41.2
## 
## Scaled residuals: 
##      Min       1Q   Median       3Q      Max 
## -1.74880 -0.47082 -0.07196  0.45271  1.75159 
## 
## Random effects:
##  Groups    Name        Variance Std.Dev.
##  container (Intercept) 3.6402   1.9079  
##  Residual              0.3103   0.5571  
## Number of obs: 18, groups:  container, 6
## 
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)    9.315      1.368   6.809
## trt.fh        -5.287      1.935  -2.732
## trt.fl        -4.472      1.935  -2.311
## 
## Correlation of Fixed Effects:
##        (Intr) trt.fh
## trt.fh -0.707       
## trt.fl -0.707  0.500
```

```r
anova(sal.lme)
```

```
## Analysis of Variance Table
##       Df Sum Sq Mean Sq F value
## trt.f  2 2.6864  1.3432  4.3282
```

```r
sal.emm <- emmeans(sal.lme, 'trt.f')
sal.emm
```

```
##  trt.f emmean   SE df lower.CL upper.CL
##  0       9.31 1.37  3    4.961    13.67
##  h       4.03 1.37  3   -0.326     8.38
##  l       4.84 1.37  3    0.489     9.20
## 
## Degrees-of-freedom method: kenward-roger 
## Confidence level used: 0.95
```

```r
joint_tests(sal.emm)
```

```
##  model term df1 df2 F.ratio p.value
##  trt.f        2   3   4.328 0.1306
```

The next three items demonstrate useful things to appreciate

1) What if containers are numbered 1,2 within each treatment?

The short variable is like this. We still want a random effect with 6 levels, one for each combination of treatment and container number. (1|short) is not sufficient because that will fit 2 levels, one for all containers numbered 1 and one for all containers numbered 2. We have to specify all combinations. That's done by trt:short or trt.f:short.

The output, either from printing sal.lme or summary(sal.lme), includes a very useful thing to check that you have fit the intended model. The line "number of obs . . . " tells you how many observations were used and the number of groups for each random effect in the model.

You see that trt:short has 6 groups, which is what we want.

```
lmer(weight ~ trt.f + (1|trt:short),
  data=salinity)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: weight ~ trt.f + (1 | trt:short)
##    Data: salinity
## REML criterion at convergence: 41.1584
## Random effects:
##  Groups    Name        Std.Dev.
##  trt:short (Intercept) 1.9079
##  Residual              0.5571
## Number of obs: 18, groups:  trt:short, 6
## Fixed Effects:
## (Intercept)       trt.fh       trt.fl
##       9.315       -5.287       -4.472
```

But just specifying short gives you only two groups.

```
lmer(weight ~ trt.f + (1|short),
  data=salinity)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: weight ~ trt.f + (1 | short)
##    Data: salinity
## REML criterion at convergence: 60.1719
## Random effects:
##  Groups   Name        Std.Dev.
##  short    (Intercept) 0.8582
##  Residual             1.4327
## Number of obs: 18, groups:  short, 2
## Fixed Effects:
## (Intercept)       trt.fh       trt.fl
##       9.315       -5.287       -4.472
```

Advice: Check that the number of groups matches what you expect.

2) fitting a random effect for plant

If you think about the data, you realize there are 2 random effects: containers and plants within containers. You could try to fit that model and R will complain. Plant(container) uniquely identifies each observation. The residual is plant(container).

```
# not run because the error halts the knitring
#sal.lme3 <- lmer(weight ~ trt.f + (1|container) + (1|container:plant),
#  data=salinity)
```

3) “301/587” solution: average plants within each container

We can use dplyr tools to do the averaging. Salinity2 has 6 observations, one for each container, with the average weight for that container.

```
library(dplyr)

salinity2 <- salinity %>% group_by(trt.f, container) %>%
  summarize(meanwt = mean(weight))
salinity2
```

```
## # A tibble: 6 x 3
## # Groups:   trt.f [3]
##   trt.f container meanwt
##   <fct>     <int>  <dbl>
## 1 0             1  11.2
## 2 0             2   7.47
## 3 h             5   5.20
## 4 h             6   2.86
## 5 l             3   3.92
## 6 l             4   5.77
```

If we fit a one-way ANOVA model to that data, we get the same results for treatment means we did from the random effects model. We just don’t get the two variance components.

```
sal2.lm <- lm(meanwt ~ trt.f, data=salinity2)
sal2.emm <- emmeans(sal2.lm, 'trt.f')
sal2.emm
```

```
##  trt.f emmean   SE df lower.CL upper.CL
##  0       9.31 1.37  3    4.961    13.67
##  h       4.03 1.37  3   -0.326     8.38
##  l       4.84 1.37  3    0.489     9.20
##
## Confidence level used: 0.95
```

```
anova(sal2.lm)
```

```
## Analysis of Variance Table
##
## Response: meanwt
##           Df Sum Sq Mean Sq F value Pr(>F)
## trt.f      2 32.406 16.2030  4.3281 0.1306
## Residuals  3 11.231  3.7436
```