# Using the lsmeans Package

Russell V. Lenth
The University of Iowa
russell-lenth@uiowa.edu

August 14, 2012

## Introduction

Least-squares means (or LS means), popularized by SAS, are predictions from a linear model at combinations of specified factors. SAS's documentation describes them as "predicted population margins—that is, they estimate the marginal means over a balanced population" (SAS Institute 2012). Unspecified factors and covariates are handled by summarizing the predictions over those factors and variables. This vignette gives some examples of LS means and the `lsmeans` package. Some of the finer points of LS means are explained in the context of these examples.

Like most statistical calculations, it is possible to use least-squares means inappropriately; however, they are in fact simply predictions from the model. When used with due care, they can provide useful summaries of a linear model that includes factors.

## Split-Plot Example

The `nlme` package includes a famous dataset `Oats` that was used in 1935 by Yates as an example of a split-plot experiment. Here is a summary of the dataset.

```
R> library(nlme)
R> summary(Oats)

 Block          Variety        nitro           yield      
 VI :12   Golden Rain:24   Min.   :0.00   Min.   : 53.0  
 V  :12   Marvellous :24   1st Qu.:0.15   1st Qu.: 86.0  
 III:12   Victory    :24   Median :0.30   Median :102.5  
 IV :12                    Mean   :0.30   Mean   :104.0  
 II :12                    3rd Qu.:0.45   3rd Qu.:121.2  
 I  :12                    Max.   :0.60   Max.   :174.0  
```

The experiment was conducted in six blocks, and each block was divided into three plots, which were randomly assigned to varieties of oats. With just `Variety` as a factor, it is a randomized complete-block experiment. However, each plot was subdivided into 4 subplots and the subplots were treated with different amounts of nitrogen. Thus, `Block` is a blocking factor, `Variety` is the whole-plot factor, and `nitro` is the split-plot factor. The response variable is `yield`, the yield of each subplot in bushels per acre.

We now do a basic analysis of these data using the `lme` function in the `nlme` package. Our initial model will treat `nitro` as a 4-level factor rather than a numeric predictor, and to accommodate a current weakness in the `lsmeans` package, we must create a new variable (we'll name it `nitroF`) that represents `nitro` as a factor. We'll fit an additive model in the two primary factors, and specify that `Block` and `Variety %in% Block` serve as sources of random variation.

```
R> Oats$nitroF = factor(Oats$nitro)   # This also creates a copy of 'Oats' in our workspace
R> Oats.lme = lme(yield ~ nitroF + Variety, random = ~1 | Block/Variety, data=Oats)
R> anova(Oats.lme, type="marginal")

            numDF denDF  F-value p-value
(Intercept)     1    51 94.51352  <.0001
nitroF          3    51 41.05278  <.0001
Variety         2    10  1.48534  0.2724
```

## Least-squares means are predictions

Now, as a follow-up to this analysis, we might want specific information on how the factor levels compare. One way to approach this is to compute predicted values from the fixed-effects portion of the model for each combination of `Variety` and `nitroF`.

```
R> grid = with(Oats, expand.grid(Variety=levels(Variety), nitroF=levels(nitroF)))
R> predict(Oats.lme, new = grid, level = 0)

 [1]  79.91667  85.20833  73.04167  99.41667 104.70833  92.54167 114.75000
 [8] 120.04167 107.87500 123.91667 129.20833 117.04167
attr(,"label")
[1] "Predicted values"
```

These predictions are also easily obtained from the `lsmeans` function, simply by specifying the factor combinations in a formula):[1]

```
R> library(lsmeans)
R> lsmeans(Oats.lme, ~ Variety:nitroF)

$`Variety:nitroF lsmeans`
                 estimate       SE   t.ratio
Golden Rain,0    79.91667 8.220351  9.721807
Marvellous,0     85.20833 8.220351 10.365534
Victory,0        73.04167 8.220351  8.885468
Golden Rain,0.2  99.41667 8.220351 12.093968
Marvellous,0.2  104.70833 8.220351 12.737695
Victory,0.2      92.54167 8.220351 11.257629
Golden Rain,0.4 114.75000 8.220351 13.959257
Marvellous,0.4  120.04167 8.220351 14.602985
Victory,0.4     107.87500 8.220351 13.122918
Golden Rain,0.6 123.91667 8.220351 15.074376
Marvellous,0.6  129.20833 8.220351 15.718103
Victory,0.6     117.04167 8.220351 14.238037
```

Often, though, people are interested in marginal results. The LS means are simply the averages of the above results over the levels of the other factor:

```
R> lsmeans(Oats.lme, list( ~ Variety, ~ nitroF))

$`Variety lsmeans`
            estimate       SE  t.ratio
Golden Rain 104.5000 7.797492 13.40174
Marvellous  109.7917 7.797492 14.08038
Victory      97.6250 7.797492 12.52005
```

[1] Interestingly, an `LSMEANS` statement in SAS will refuse to output predictions for factor combinations unless the interaction is in the model. However, they are unambiguously defined.

```
$`nitroF lsmeans`
     estimate       SE  t.ratio
0    79.38889 7.132357 11.13081
0.2  98.88889 7.132357 13.86483
0.4 114.22222 7.132357 16.01465
0.6 123.38889 7.132357 17.29987
```

## Comparisons and contrasts

Often, we want comparisons or other contrasts among the LS means. The `lsmeans` function allows specifying a family of such contrasts in the left-hand side of the formulas. In this example, we might want to compare the `Variety` means with one another, while orthogonal-polynomial contrasts are more in order for `nitroF` since it is quantitative. Thus:

```
R> lsmeans(Oats.lme, list(pairwise ~ Variety, poly ~ nitroF))

$`Variety lsmeans`
            estimate       SE  t.ratio
Golden Rain 104.5000 7.797492 13.40174
Marvellous  109.7917 7.797492 14.08038
Victory      97.6250 7.797492 12.52005

$`Variety pairwise differences`
                          estimate      SE    t.ratio
Golden Rain - Marvellous -5.291667 7.07891 -0.7475256
Golden Rain - Victory     6.875000 7.07891  0.9711947
Marvellous - Victory     12.166667 7.07891  1.7187204

$`nitroF lsmeans`
     estimate       SE  t.ratio
0    79.38889 7.132357 11.13081
0.2  98.88889 7.132357 13.86483
0.4 114.22222 7.132357 16.01465
0.6 123.38889 7.132357 17.29987

$`nitroF polynomial contrasts`
           estimate        SE    t.ratio
linear    147.33333 13.439537 10.9626791
quadratic -10.33333  6.010344 -1.7192583
cubic      -2.00000 13.439537 -0.1488146
```

## Covariate model

The above results would convince us that the cubic term of `nitroF` is not needed; some might also toss out the quadratic term but that decision is less clear. Suppose that we decide to fit a new model treating `nitro` as a quantitative variable, and account for both linear and quadratic terms.

```
R> OatsPoly.lme = lme(yield ~ poly(nitro, 2) + Variety, random = ~ 1 | Block/Variety, data = Oats)
R> lsmeans(OatsPoly.lme, pairwise ~ Variety)

$`Variety lsmeans`
            estimate       SE  t.ratio
Golden Rain 107.7292 8.016378 13.43863
Marvellous  113.0208 8.016378 14.09874
```

```
Victory     100.8542 8.016378 12.58101

$`Variety pairwise differences`
                          estimate       SE    t.ratio
Golden Rain - Marvellous -5.291667 7.078916 -0.7475250
Golden Rain - Victory     6.875000 7.078916  0.9711939
Marvellous - Victory     12.166667 7.078916  1.7187188
```

The LS means obtained are somewhat different than what we had before, but the pairwise comparisons are very nearly identical. The above model is an example of a model that includes covariates—in this case the linear and quadratic terms for `nitro`. In such cases, LS means are comparable to what is often termed adjusted means: predicted values at each factor level, obtained by substituting the average value of each covariate. In this case, the LS means are predictions when `nitro` is set at its average value, 0.30. Noting that the quadratic effect is negative, the fitted curves are concave in `nitro`; thus it makes sense that the predicted values at the average `nitro` (i.e., the LS means for `OatsPoly.lme`) are greater than the averages of the predictions at the four levels of `nitroF`, which is what we had when we obtained the LS means from `Oats.lme`.

As a side note, it is definitely desirable to use basis functions like `poly()` or `ns()` rather than manual coding. Had our model specified `nitro` and `I(nitro^2)`, these covariates would have been averaged separately in the LS means, so we would have obtained the predictions when `nitro` $= 0.30$ and `nitro^2` $= 0.14 \neq 0.30^2$.

`lsmeans` allows an `at` argument if prediction at a different covariate value is desired:

```
R> lsmeans(OatsPoly.lme, ~ Variety, at = c(nitro=0.60))

$`Variety lsmeans`
            estimate       SE  t.ratio
Golden Rain 124.0167 8.185583 15.15062
Marvellous  129.3083 8.185583 15.79708
Victory     117.1417 8.185583 14.31073
```

Note that these results are quite close to those obtained earlier for `Variety:nitroF` where `nitroF = 0.6`.

## Model with interaction

Returning momentarily to `Oats.lme`, suppose that we include the interaction in the model.

```
R> OatsInt.lme = update(Oats.lme, . ~ . + Variety:nitroF)
R> anova(OatsInt.lme, type="marginal")

               numDF denDF  F-value p-value
(Intercept)        1    45 77.16728  <.0001
nitroF             3    45 13.02274  <.0001
Variety            2    10  1.22454  0.3344
nitroF:Variety     6    45  0.30282  0.9322
```

Up to now, I believe that all LS mean examples presented so far are uncontroversial; however, the following one is:

```
R> lsmeans(OatsInt.lme, ~ nitroF)

$`nitroF lsmeans`
     estimate       SE  t.ratio
0    79.38889 7.174685 11.06514
0.2  98.88889 7.174685 13.78303
0.4 114.22222 7.174685 15.92017
0.6 123.38889 7.174685 17.19781
```

Some would argue that you shouldn't examine marginal effects when the interaction is in the model; others would say it's OK because the interaction is nonsignificant anyway. I will not wade into that argument, and just say that if you understand what it is you are doing (in this case, averaging three predictions together to obtain each LS mean), you can decide whether it is appropriate or not to do so. The idea certainly seems less and less commendable as the strength of the interaction increases.

Returning (I hope) to something people will agree on, when there is an interaction it is fairly common to want to do comparisons of one factor at each level of the other factor. This may be done by using a conditioning symbol, `|`, in the formula, like this:

```
R> lsmeans(OatsInt.lme, poly ~ nitroF | Variety)

$`nitroF:Variety lsmeans`
                 estimate       SE   t.ratio
0,Golden Rain    80.00000 9.106959  8.784491
0.2,Golden Rain  98.50000 9.106959 10.815905
0.4,Golden Rain 114.66667 9.106959 12.591104
0.6,Golden Rain 124.83333 9.106959 13.707466
0,Marvellous     86.66667 9.106959  9.516532
0.2,Marvellous  108.50000 9.106959 11.913966
0.4,Marvellous  117.16667 9.106959 12.865619
0.6,Marvellous  126.83333 9.106959 13.927079
0,Victory        71.50000 9.106959  7.851139
0.2,Victory      89.66667 9.106959  9.845950
0.4,Victory     110.83333 9.106959 12.170180
0.6,Victory     118.50000 9.106959 13.012027

$`nitroF:Variety polynomial contrasts`
                          estimate       SE    t.ratio
linear | Golden Rain    150.666667 24.29564  6.2013868
quadratic | Golden Rain  -8.333333 10.86534 -0.7669647
cubic | Golden Rain      -3.666667 24.29564 -0.1509187
linear | Marvellous     129.166667 24.29564  5.3164544
quadratic | Marvellous  -12.166667 10.86534 -1.1197685
cubic | Marvellous       14.166667 24.29564  0.5830950
linear | Victory        162.166667 24.29564  6.6747227
quadratic | Victory     -10.500000 10.86534 -0.9663756
cubic | Victory         -16.500000 24.29564 -0.6791342
```

## Even nonsignificant interactions can make a big difference

Lest you think that there's little difference between `Oats.lme` and `OatsInt.lme`, that's not really the case when you consider comparing the cell LS means. Figure 1 displays the first six comparisons of the `Variety:nitroF` LS means with each model. With the interaction in the model (left), and a balanced design, the standard error of such a comparison can be one of two values, depending on whether the comparison is on the same whole plot or between different whole plots. With the additive model (right), there are three different standard errors: one when `Variety` is the same, one when `nitroF` is the same, and one when both factors are at different levels. This seems alarming until you realize that in the first two respective cases, the estimates and standard errors are the same as for the marginal LS means of `nitroF` and `Variety`, respectively.

## Custom contrasts

The built-in families of contrasts available in the `lsmeans` package are `pairwise`, `poly`, `revpairwise`, `trt.vs.ctrl`, `trt.vs.ctrl1`, and `trt.vs.ctrlk`. The first two have been illustrated here; `revpairwise` is like `pairwise` except the subtractions are done in the reverse direction (higher levels minus lower levels). `trt.vs.ctrl` generates comparisons of each level versus a specified level that you need to provide; `trt.vs.ctrl1` and

Figure 1: Selected cell-mean comparisons for the interaction model (left) versus the additive model (right)

```
R> lsmeans(OatsInt.lme, pairwise~Variety:nitroF
R>   ) [[2]] [1:6 ,1:2]
                                    estimate       SE
Golden Rain,0 - Marvellous,0     -6.666667 9.715030
Golden Rain,0 - Victory,0         8.500000 9.715030
Golden Rain,0 - Golden Rain,0.2 -18.500000 7.682956
Golden Rain,0 - Marvellous,0.2  -28.500000 9.715030
Golden Rain,0 - Victory,0.2      -9.666667 9.715030
Golden Rain,0 - Golden Rain,0.4 -34.666667 7.682956
```

```
R> lsmeans(Oats.lme, pairwise~Variety:nitroF
R>   ) [[2]] [1:6 ,1:2]
                                    estimate       SE
Golden Rain,0 - Marvellous,0     -5.291667 7.078910
Golden Rain,0 - Victory,0         6.875000 7.078910
Golden Rain,0 - Golden Rain,0.2 -19.500000 4.249955
Golden Rain,0 - Marvellous,0.2  -24.791667 8.256699
Golden Rain,0 - Victory,0.2     -12.625000 8.256699
Golden Rain,0 - Golden Rain,0.4 -34.833333 4.249955
```

---

`trt.vs.ctrlk` are convenience versions of `trt.vs.ctrl` predefine the control group as the fist and the last levels, respectively.

If you want to define some other contrast set, you may provide it as a named entry in a list in the `contr` argument, and refer to that name in the formula, like this:

```
R> lsmeans(Oats.lme, my.own ~ Variety,
R>   contr = list(my.own = list(G.vs.M = c(1,-1,0), GM.vs.V = c(.5,.5,-1),
R>     total = c(1,1,1)))) [[2]]

          estimate        SE    t.ratio
G.vs.M   -5.291667  7.078910 -0.7475256
GM.vs.V   9.520833  6.130516  1.5530232
total   311.916667 19.921723 15.6571127
```

The third one isn't even a contrast, which is OK—any linear combination is allowed.

There is another way to provide custom contrasts. The built-in families are actually implemented via functions `pairwise.lsmc`, `poly.lsmc`, … . You may write your own `.lsmc` function and use the first part of its name in a formula. In the following example, we define a function for Helmert contrasts:

```
R> helmert.lsmc = function(levs, ...) {
R>   M = as.data.frame(contr.helmert(levs))
R>   names(M) = paste(levs[-1],"vs earlier")
R>   attr(M, "desc") = "Helmert contrasts"
R>   M
R> }
R> lsmeans(Oats.lme, helmert ~ nitroF)

$`nitroF lsmeans`
     estimate       SE  t.ratio
0    79.38889 7.132357 11.13081
0.2  98.88889 7.132357 13.86483
0.4 114.22222 7.132357 16.01465
0.6 123.38889 7.132357 17.29987

$`nitroF Helmert contrasts`
                estimate        SE  t.ratio
0.2 vs earlier 19.50000  4.249955 4.588284
0.4 vs earlier 50.16667  7.361138 6.815070
0.6 vs earlier 77.66667 10.410221 7.460617
```

The `desc` attribute is optional, and used in the labeling of the output list (if not provided, the function name is used).

### Custom treatment of extraneous variables

You may override the defaults for handling covariates and combining factor levels via the `cov.reduce` and `fac.reduce` arguments. For example, suppose (for some very odd reason) we want our adjusted means to be at the upper quartile of each covariate; then do this:

```
R> lsmeans(OatsPoly.lme, ~ Variety,
R>   cov.reduce = function(x, name) {
R>     q75 = quantile(x,.75)
R>     cat(paste("Predictions are made at", name, "=", q75, "\n"))
R>     q75
R>   })

Predictions are made at nitro = 0.45
$`Variety lsmeans`
            estimate       SE  t.ratio
Golden Rain 117.3260 7.927476 14.79992
Marvellous  122.6177 7.927476 15.46743
Victory     110.4510 7.927476 13.93269
```

By default, LS means are averaged with equal weight given to levels of extraneous factors. (This is comparable, more or less, to the "unweighted means" analysis used in the olden days for unbalanced data.) We can change this by specifying a function in `fac.reduce` that collapses the rows of a matrix of coefficients. For example, we could just use the last row:

```
R> lsmeans(Oats.lme, ~ Variety, fac.reduce = function(X, lev) X[nrow(X), ])

$`Variety lsmeans`
            estimate       SE  t.ratio
Golden Rain 123.9167 8.220351 15.07438
Marvellous  129.2083 8.220351 15.71810
Victory     117.0417 8.220351 14.23804
```

These are of course just the LS means at `nitroF = .6`, seen earlier in this vignette.

## Reference

**SAS Institute Inc. (2012)** Online documentation; Shared concepts; LSMEANS statement, http://support.sas.com/documentation/cdl/en/statug/63962/HTML/default/viewer.htm#statug_introcom_a0000003362.htm, accessed August 14, 2012.