Marginal means with lmer

Sasha D. Hafner

18 November, 2024 Nov:11

Summary

Just confirming that emmeans does what I think it does with mixed-effects models without balance.

0. Packages

```
library(lme4)
library(emmeans)
library(data.table)
```

```
source('dfcombos.R')
```

1. Data

Five levels of a random effect.

```
rv <- data.table(rv = factor(letters[1:5]), re = -2:2)
rv
```

```
##        rv    re
##    <fctr> <int>
```

```
## 1:      a    -2
## 2:      b    -1
## 3:      c     0
## 4:      d     1
## 5:      e     2
```

(Hmm, should probably try with a mean different from 0.)

Three treatments make up a fixed effect.

```
ff <- data.table(trt = factor(letters[1:3]), fe = c(-1, 0, 1))
```

Combine.

```
d0 <- dfcombos(rv, ff)
```

Eliminate balance.

```
d0 <- d0[-c(9:10, 12:14), ]
d0
```

```
##    rv re trt fe
## 1   a -2   a -1
## 2   b -1   a -1
## 3   c  0   a -1
## 4   d  1   a -1
## 5   e  2   a -1
## 6   a -2   b  0
## 7   b -1   b  0
## 8   c  0   b  0
## 11  a -2   c  1
## 15  e  2   c  1
```

```
table(d0$trt)
```

```
##
## a b c
## 5 3 2
```

```r
table(d0$rv)
```

```
##
## a b c d e
## 3 2 2 1 2
```

Add response variable

```r
setDT(d0)
d0[, y := re + fe + rnorm(nrow(d0), sd = 0.2)]
d0
```

```
##         rv    re    trt    fe           y
##     <fctr> <int> <fctr> <num>       <num>
##  1:      a    -2      a    -1 -2.85217702
##  2:      b    -1      a    -1 -1.92267825
##  3:      c     0      a    -1 -0.74072057
##  4:      d     1      a    -1 -0.16071167
##  5:      e     2      a    -1  0.67947487
##  6:      a    -2      b     0 -1.81334981
##  7:      b    -1      b     0 -0.63878215
##  8:      c     0      b     0 -0.01130073
##  9:      a    -2      c     1 -0.62281774
## 10:      e     2      c     1  3.31567669
```

2. Model

```r
m1 <- lmer(y ~ trt + (1|rv), data = d0)
summary(m1)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ trt + (1 | rv)
##    Data: d0
##
```

```
## REML criterion at convergence: 18.6
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7665 -0.5892  0.1489  0.4727  0.7765
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rv       (Intercept) 2.04697  1.4307
##  Residual             0.04147  0.2036
## Number of obs: 10, groups:  rv, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept)  -0.9994     0.6463  -1.546
## trtb          1.0436     0.1622   6.433
## trtc          2.4095     0.1938  12.431
##
## Correlation of Fixed Effects:
##      (Intr) trtb
## trtb -0.079
## trtc -0.066  0.240
```

```
emmeans(m1, 'trt')
```

```
##  trt  emmean    SE   df lower.CL upper.CL
##  a   -0.9994 0.646 4.06   -2.784    0.785
##  b    0.0443 0.654 4.24   -1.731    1.820
##  c    1.4101 0.663 4.45   -0.358    3.178
##
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

Compare to mean by treatment.

```
d0[, mean(y), by = trt]
```

```
##       trt         V1
```

```
##    <fctr>      <num>
## 1:      a -0.9993625
## 2:      b -0.8211442
## 3:      c  1.3464295
```

3. Higher n

Instead of running this many times, I'll just increase the sample size.

```
d1 <- d0[rep(1:nrow(d0), each = 300), ]
table(d1$trt)
```

```
##
##    a    b    c
## 1500  900  600
```

And recreate y values.

```
set.seed(1)
d1[, y := re + fe + rnorm(nrow(d1), sd = 0.2)]
d1
```

```
##           rv    re    trt    fe         y
##       <fctr> <int> <fctr> <num>     <num>
##    1:      a    -2      a    -1 -3.125291
##    2:      a    -2      a    -1 -2.963271
##    3:      a    -2      a    -1 -3.167126
##    4:      a    -2      a    -1 -2.680944
##    5:      a    -2      a    -1 -2.934098
##   ---
## 2996:      e     2      c     1  2.962648
## 2997:      e     2      c     1  2.954128
## 2998:      e     2      c     1  3.326037
## 2999:      e     2      c     1  2.567066
## 3000:      e     2      c     1  2.784445
```

```r
m2 <- lmer(y ~ trt + (1|rv), data = d1)
summary(m2)
```

```
## Linear mixed model fit by REML ['lmerMod']
## Formula: y ~ trt + (1 | rv)
##    Data: d1
##
## REML criterion at convergence: -870.8
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -3.4077 -0.6384 -0.0169  0.6887  3.6976
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  rv       (Intercept) 2.5117   1.5848
##  Residual             0.0429   0.2071
## Number of obs: 3000, groups:  rv, 5
##
## Fixed effects:
##             Estimate Std. Error t value
## (Intercept) -1.001752   0.708775  -1.413
## trtb         0.999657   0.009539 104.798
## trtc         2.001700   0.011401 175.571
##
## Correlation of Fixed Effects:
##      (Intr) trtb
## trtb -0.004
## trtc -0.004  0.239
```

```r
emmeans(m2, 'trt')
```

```
##  trt  emmean    SE df lower.CL upper.CL
##  a   -1.0018 0.709  4   -2.970    0.966
##  b   -0.0021 0.709  4   -1.970    1.966
##  c    0.9999 0.709  4   -0.968    2.968
##
```

```
## Degrees-of-freedom method: kenward-roger
## Confidence level used: 0.95
```

Compare to mean by treatment.

```
d1[, mean(y), by = trt]
```

```
##       trt        V1
##    <fctr>     <num>
## 1:      a -1.001752
## 2:      b -1.003417
## 3:      c  1.005306
```

Notice that `lmer` coefficients (`Estimate`) and marginal means look identical.

Conclusion

The emmeans package seems to be a simple way to get marginal means from mixed-effects models.