GET BLUE or BLUP from lmer in R

The following text is copied and pasted from https://stats.stackexchange.com/questions/122218/why-do-the-estimated-values-from-a-best-linear-unbiased-predictor-blup-differ.

Q:

Why do the estimated values from a Best Linear Unbiased Predictor (BLUP) differ from a Best Linear Unbiased Estimator (BLUE)?

I understand that the difference between them is related to whether the grouping variable in the model is estimated as a fixed or random effect, but it's not clear to me why they are not the same (if they are not the same).

I am specifically interested in how this works when using small area estimation, if that's relevant, but I suspect the question is relevant to any application of fixed and random effects.

A:

The values that you get from BLUPs aren't estimated in the same way as the BLUE estimates of fixed effects; by convention BLUPs are referred to as predictions. When you fit a mixed effects model, what are estimated initially are the mean and variance (and possibly the covariance) of the random effects. The random effect for a given study unit (say a student) is subsequently calculated from the estimated mean and variance, and the data. In a simple linear model, the mean is estimated (as is the residual variance), but the observed scores are considered to be composed of both that and the error, which is a random variable. In a mixed effects model, the effect for a given unit is likewise a random variable (although in some sense it has already been realized).

You can also treat such units as fixed effects, if you like. In that case, the parameters for that unit are estimated as usual. In such a case however, the mean (for example) of the population from which the units were drawn is not estimated.

Moreover, the assumption behind random effects is that they were sampled at random from some population, and it is the population that you care about. The assumption underlying fixed effects is that you selected those units purposefully because those are the only units you care about.

If you turn around and fit a mixed effects model and predict those same effects, they tend to be 'shrunk' towards the population mean relative to their fixed effects estimates. You can think of this as analogous to a Bayesian analysis where the estimated mean and variance specify a normal prior and the BLUP is like the mean of the posterior that comes from optimally combining the data with the prior.

The amount of shrinkage varies based on several factors. An important determinate of how far the random effects predictions will be from the fixed effects estimates is the ratio of the variance of the random effects to the error variance. Here is a quick R demo for the simplest case with 5 'level 2' units with only means (intercepts) fit. (You can think of this as test scores for students within classes.)

```
library(lme4) # we'll need to use this package
 set.seed(1673) # this makes the example exactly reproducible
 nj = 5; ni = 5; g = as.factor(rep(c(1:nj), each=ni))


##### model 1
 pop.mean = 16; sigma.g = 1; sigma.e = 5
 r.eff1 = rnorm(nj, mean=0, sd=sigma.g)
 error = rnorm(nj*ni, mean=0, sd=sigma.e)
 y = pop.mean + rep(r.eff1, each=ni) + error


re.mod1 = lmer(y~(1|g))
 fe.mod1 = lm(y~0+g)
 df1 = data.frame(fe1=coef(fe.mod1), re1=coef(re.mod1)$g)
```

```
##### model 2
 pop.mean = 16; sigma.g = 5; sigma.e = 5
 r.eff2 = rnorm(nj, mean=0, sd=sigma.g)
 error = rnorm(nj*ni, mean=0, sd=sigma.e)
 y = pop.mean + rep(r.eff2, each=ni) + error



re.mod2 = lmer(y~(1|g))
 fe.mod2 = lm(y~0+g)
 df2 = data.frame(fe2=coef(fe.mod2), re2=coef(re.mod2)$g)



##### model 3
 pop.mean = 16; sigma.g = 5; sigma.e = 1
 r.eff3 = rnorm(nj, mean=0, sd=sigma.g)
 error = rnorm(nj*ni, mean=0, sd=sigma.e)
 y = pop.mean + rep(r.eff3, each=ni) + error



re.mod3 = lmer(y~(1|g))
 fe.mod3 = lm(y~0+g)
 df3 = data.frame(fe3=coef(fe.mod3), re3=coef(re.mod3)$g)
```

So the ratios of the variance of the random effects to the error variance is 1/5 for model 1, 5/5 for model 2, and 5/1 for model 3. Note that I used cell means coding for the fixed effects models. We can now examine how the estimated fixed effects and the predicted random effects compare for these three scenarios.

```
df1
 # fe1 re1
 # g1 17.88528 15.9897
 # g2 18.38737 15.9897
 # g3 14.85108 15.9897
 # g4 14.92801 15.9897
 # g5 13.89675 15.9897



df2
```

```
 # fe2 re2
 # g1 10.979130 11.32997
 # g2 13.002723 13.14321
 # g3 26.118189 24.89537
 # g4 12.109896 12.34319
 # g5 9.561495 10.05969
```

```
df3
 # fe3 re3
 # g1 13.08629 13.19965
 # g2 16.36932 16.31164
 # g3 17.60149 17.47962
 # g4 15.51098 15.49802
 # g5 13.74309 13.82224
```

Another way to end up with random effects predictions that are closer to the fixed effects estimates is when you have more data. We can compare model 1 from above, with its low ratio of random effects variance to error variance, to a version (model 1b) with the same ratio, but much more data (notice that ni = 500 instead of ni = 5).

```
##### model 1b
 nj = 5; ni = 500; g = as.factor(rep(c(1:nj), each=ni))
 pop.mean = 16; sigma.g = 1; sigma.e = 5
 r.eff1b = rnorm(nj, mean=0, sd=sigma.g)
 error = rnorm(nj*ni, mean=0, sd=sigma.e)
 y = pop.mean + rep(r.eff1b, each=ni) + error
```

```
re.mod1b = lmer(y~(1|g))
 fe.mod1b = lm(y~0+g)
 df1b = data.frame(fe1b=coef(fe.mod1b), re1b=coef(re.mod1b)$g)
 Here are the effects:
```

```
df1
 # fe1 re1
 # g1 17.88528 15.9897
```

```
 # g2 18.38737 15.9897
 # g3 14.85108 15.9897
 # g4 14.92801 15.9897
 # g5 13.89675 15.9897
```

df1b

```
 # fe1b re1b
 # g1 15.29064 15.29543
 # g2 14.05557 14.08403
 # g3 13.97053 14.00061
 # g4 16.94697 16.92004
 # g5 17.44085 17.40445
```

On a somewhat related note, Doug Bates (the author of the R package lme4) doesn't like the term "BLUP" and uses "conditional mode" instead (see pp. 22-23 of his draft lme4 book pdf). In particular, he points out in section 1.6 that "BLUP" can only meaningfully be used for linear mixed-effects models.