Heterogeneous variances and weighting

Facundo Muñoz

2017-04-12 breedR version: 0.12

Contents

By default, the Linear Mixed Models fitted with breedR assume *homoscedasticity*. Meaning that given all the fixed and random effects, the unexplained variation follow a Normal distribution with residual variance σ^2.

Mathematically, that $\varepsilon \sim \mathcal{N}(0, \mathbf{I}\sigma^2)$ in the model equation

$$y = X\beta + Zu + \varepsilon$$

Sometimes this is obviously wrong, and we need models where some observations are observed with more or less residual variability than others

Here are a few common situations where heterogeneous variances are needed:

- The observations are actually derived or calculated from real measurements, such as an average. Thus, the variance depends on the number of averaged measurements (e.g. Daughter Yield Deviation measures).
- The observations are spread in time, and you want to model the residual variance as a function of time (e.g. longitudinal models).

Using weights

If the relative variation in the residual variances is know or can be estimated, it can be specified as a vector of *weights* w, such that

$$\varepsilon \sim \mathcal{N}(0, (w^{-1/2})'\mathbf{I}w^{-1/2}\sigma^2).$$

In other words, the residual variance for the observation i is σ^2/w_i.

Here is a simulation example of how to specify weights.

```
set.seed(123)

n <- 1e3   # n obs
sigma2 <- 4  # true residual variance (for a weight of 1)
w = runif(n, min = .5, max = 2)  # vector of weights

dat <-
  transform(
    data.frame(
      e = rnorm(n, sd = sqrt(sigma2))
    ),
    y = 10 + e/sqrt(w)  # simulated phenotype
  )
```

```
res <- remlf90(
  y ~ 1,
  data = dat,
  weights = w  # specification of weights
)
```

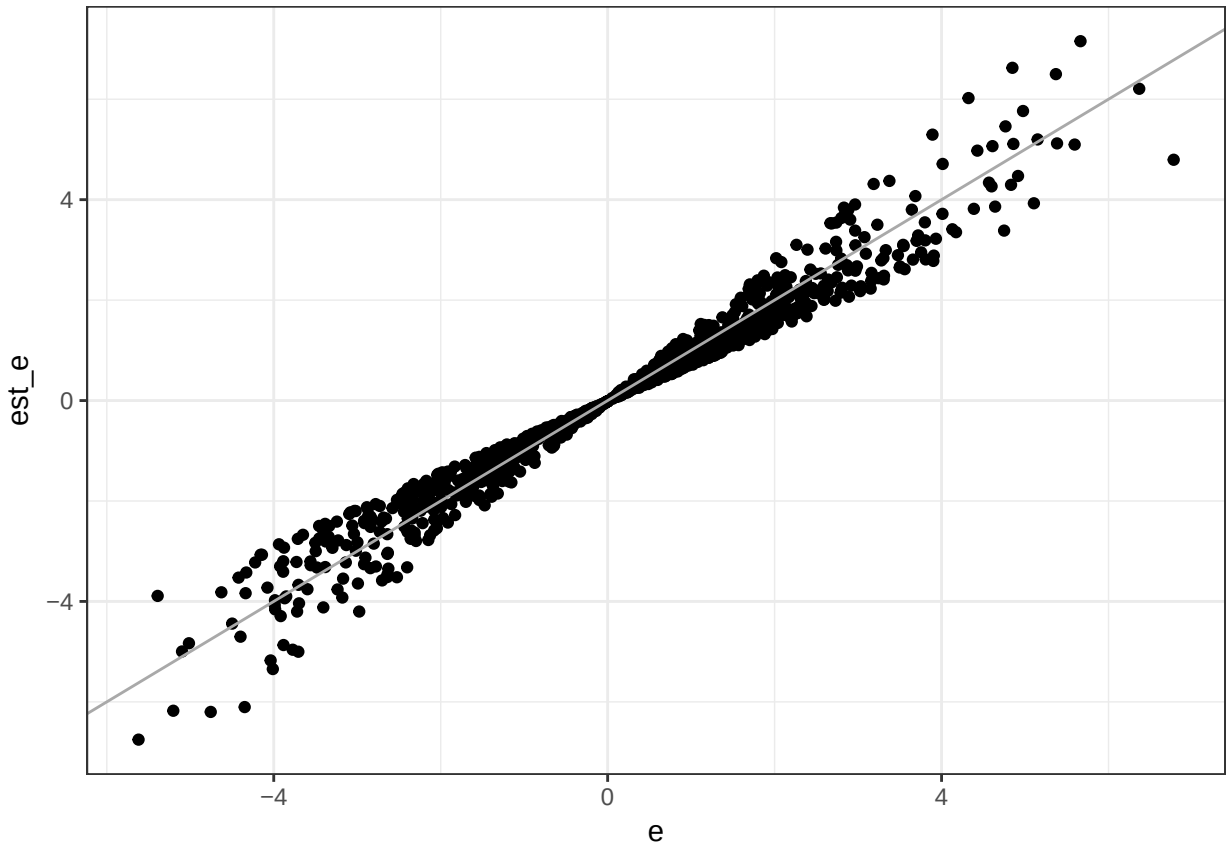
```
## Using default initial variances given by default_initial_variance()
## See ?breedR.getOption.
```

Note that the estimated residual variance is close to the true value. On the other hand, the residual prediction-error are expected to have non-constant variance.

```
summary(res)
```

```
## Formula: y ~ 0 + Intercept
##    Data: dat
##   AIC  BIC logLik
##  4080 4085  -2039
##
##
## Variance components:
##          Estimated variances   S.E.
## Residual               4.012 0.1618
##
## Fixed effects:
##            value   s.e.
## Intercept 10.016 0.0567
```

```
ggplot(transform(dat, est_e = residuals(res)), aes(e, est_e)) +
  geom_point() +
  geom_abline(intercept = 0, slope = 1, color = "darkgray")
```



Estimating residual variance heterogeneity

This is currently not available in breedR.

Different group-wise residual variances (e.g. multi-site) can be easily induced by using group-specific random effects.

For the general case, here are some notes to allow for some manual hacking if needed.

In general, we need to estimate a residual variance parameter as a function of some other variable x. We then write the residual variance as a linear combination of a few base functions:

$$\sigma^2(x) = \sum_{k=0}^{K} \psi_k(x)\, r_k = \mathbf{\Psi r},$$

where the parameters r_k are to be estimated.

This covers the case both for group-wise residual variances (such as multi-site, using a categorical variable x) or a continuously varying residual variance.

For the first case, the variable x is categorical, taking a finite number of values K, and we define ψ_k as the corresponding indicator functions.

For the continuous case, the variable x is continuous (e.g. age, temperature) and the base functions can be Legendre polynomials, splines, etc. up to some arbitrary order K.

We need to manually build the matrix $\mathbf{\Psi}$, and exploit the PROGSF90 options `hetres_pos` and `hetres_pol` available in AI-REML (see documentation).