Module 2: Dissecting Mixed Model Equation: BLUEs and BLUPs

Fundamentals of Genomic Prediction and Data-Drive Crop Breeding

(August 4-8, 2025)

Waseem Hussain
Senior Scientist-I
International Rice Research Institute
Rice Breeding Innovations Platfrom
waseem.hussain@cgiar.org
whussain2.github.io

Mahender Anumalla
Scientist-I
International Rice Research Institute
South-Asian Hub, Hyderabad
m.anumalla@cgiar.org

August 3, 2025

Contents

Load the required libraries

```
>   library(agridat)
>   library(lme4)
```

Introduction

Here in this section we will dissect the Henderson's (1950) mixed model equation (MME) to extract the BLUEs and BLUPs. And we will compare the results by fitting mixed model with **lme4** R package. in the end we will be knowing how mixed models works and how blues and blups are being extracted.

- Here, we will use the **australia.soybean** data set which can be loaded from the **agridat** package. Here, I am downloading the **australia.soybean** data and saved it in folder.

Data upload and information

- Here we will upload the data, and looks its structure.
- The data has 464 observations with total 10 variables.
- For demo purpose we will use environment (env), genotype (gen) and yield variables to run mixed model equation.
- Data has 58 genotypes evaluated in 8 environments
- We will use yield data as **y** response variable

```
> # Load the Australia.soybean data from agridat package
>     australia.soybean<-read.csv(file="./Data/australia.soybean.csv")
```

```
>
> # Get the structure of data
>     str(australia.soybean)
'data.frame':   464 obs. of  10 variables:
 $ env    : chr  "L70" "L70" "L70" "L70" ...
 $ loc    : chr  "Lawes" "Lawes" "Lawes" "Lawes" ...
 $ year   : int  1970 1970 1970 1970 1970 1970 1970 1970 1970 1970 ...
 $ gen    : chr  "G01" "G02" "G03" "G04" ...
 $ yield  : num  2.39 2.28 2.57 2.88 2.39 ...
 $ height : num  1.45 1.45 1.46 1.26 1.33 ...
 $ lodging: num  4.25 4.25 3.75 3.5 3.5 4 3 3.25 3 3.75 ...
 $ size   : num  8.45 9.95 10.85 10.05 11 ...
 $ protein: num  36.7 37.5 37.8 38.5 37.5 ...
 $ oil    : num  20.9 20.7 21.3 22 22.1 ...
> # Convert env, loc and year into factors
>     australia.soybean$env<-as.factor(australia.soybean$env)
>     australia.soybean$loc<-as.factor(australia.soybean$loc)
>     australia.soybean$year<-as.factor(australia.soybean$year)
>     australia.soybean$gen<-as.factor(australia.soybean$gen)
> # Data can also be upload directly from agridat package
>     data(australia.soybean, package = "agridat")
>     head(australia.soybean)
```

env	loc	year	gen	yield	height	lodging	size	protein	oil
L70	Lawes	1970	G01	2.387	1.445	4.25	8.45	36.70	20.895
L70	Lawes	1970	G02	2.282	1.450	4.25	9.95	37.55	20.740
L70	Lawes	1970	G03	2.567	1.460	3.75	10.85	37.80	21.295
L70	Lawes	1970	G04	2.877	1.260	3.50	10.05	38.45	21.990
L70	Lawes	1970	G05	2.392	1.335	3.50	11.00	37.50	22.130
L70	Lawes	1970	G06	2.408	1.360	4.00	11.75	38.25	21.160

```
>     str(australia.soybean)
'data.frame':   464 obs. of  10 variables:
 $ env    : Factor w/ 8 levels "B70","B71","L70",..: 3 3 3 3 3 3 3 3 3 3 ...
 $ loc    : Factor w/ 4 levels "Brookstead","Lawes",..: 2 2 2 2 2 2 2 2 2 2 ...
 $ year   : int  1970 1970 1970 1970 1970 1970 1970 1970 1970 1970 ...
 $ gen    : Factor w/ 58 levels "G01","G02","G03",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ yield  : num  2.39 2.28 2.57 2.88 2.39 ...
 $ height : num  1.45 1.45 1.46 1.26 1.33 ...
 $ lodging: num  4.25 4.25 3.75 3.5 3.5 4 3 3.25 3 3.75 ...
 $ size   : num  8.45 9.95 10.85 10.05 11 ...
 $ protein: num  36.7 37.5 37.8 38.5 37.5 ...
 $ oil    : num  20.9 20.7 21.3 22 22.1 ...
> # Subset the data (environment, genotypes and yield) now for mixed model equation
>     demo.data<-australia.soybean[, c(1,4,5)]
> # Look for number of environments.
>     table(demo.data$env)

B70 B71 L70 L71 N70 N71 R70 R71
 58  58  58  58  58  58  58  58
> # Look for number of environments.
>     table(demo.data$gen)
```

```
G01 G02 G03 G04 G05 G06 G07 G08 G09 G10 G11 G12 G13 G14 G15 G16 G17 G18 G19 G20
  8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8
G21 G22 G23 G24 G25 G26 G27 G28 G29 G30 G31 G32 G33 G34 G35 G36 G37 G38 G39 G40
  8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8
G41 G42 G43 G44 G45 G46 G47 G48 G49 G50 G51 G52 G53 G54 G55 G56 G57 G58
  8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8   8
```

Mixed model

Model Description

$$Yield = Intercept + Gen + Env + error$$

- The same model above can be written in the matrix form:

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

where,

y : is the vector of observed values of the trait
X : is the incidence matrix for fixed component
β : is the fixed effect, which is environment here in demo data
Z : is the incidence matrix for random component
u : is the random effect, which is genotype here in demo data
ϵ : is the error component component

here,

$$u \sim N(0, I{\sigma_u}^2) \text{and} \epsilon \sim N(0, I{\sigma_\epsilon}^2)$$

To get variance components in the above model,we assume ${\sigma_u}^2$=0.18 and ${\sigma_\epsilon}^2$=0.25.

- Also here, Environment (Env) is 8 in data and genotypes (Gen).
- Genotype here are as random, thus we will extract BLUPs for it.
- Environment are as fixed, thus we will extract the BLUEs for it.
- And finally we design **X** incidence matrix for fixed part (which is environment), and **Z** design matrix for random part which are genotypes

Mixed model equation and build components

- Here we will build the individual components of **Henderson(1950)** mixed model equation. Recall, the mixed model equation (MME) as:

$$\begin{bmatrix} \acute{X}X & \acute{X}Z \\ \acute{Z}X & \acute{Z}Z + \lambda G^{-1} \end{bmatrix} \begin{bmatrix} \beta \\ u \end{bmatrix} = \begin{bmatrix} \acute{X}y \\ \acute{Z}y \end{bmatrix}$$

* Here we will fit the mixed model given below:

Create X Design Matrix

```
> # Building X matrix for fixed component, i.e., for environments
>     X <- model.matrix(~+env, demo.data)
>     head(X)
  (Intercept) envB71 envL70 envL71 envN70 envN71 envR70 envR71
1           1      0      1      0      0      0      0      0
2           1      0      1      0      0      0      0      0
3           1      0      1      0      0      0      0      0
4           1      0      1      0      0      0      0      0
5           1      0      1      0      0      0      0      0
6           1      0      1      0      0      0      0      0
> # Building Z matrix for random component, i.e., for genotypes
>     Z <- model.matrix(~-1 + gen, demo.data) # -1 here is for removing intercept
> # Now let us have variance components
>     sigmau <- 0.199
>     sigmae <- 0.25
```

- We will use ***model.matrix*** function to create incidence matrices

Creat Respone Variable

```
> # Yield variable as response variable
>     y <-demo.data$yield
```

Build $X^T X$

Creating the X'X matrix. X is incidence matrix for fixed effects.

```
> # Cross product and transpose
>     XtX <- t(X) %*% X
>     XtX
            (Intercept) envB71 envL70 envL71 envN70 envN71 envR70 envR71
(Intercept)         464     58     58     58     58     58     58     58
envB71               58     58      0      0      0      0      0      0
envL70               58      0     58      0      0      0      0      0
envL71               58      0      0     58      0      0      0      0
envN70               58      0      0      0     58      0      0      0
envN71               58      0      0      0      0     58      0      0
envR70               58      0      0      0      0      0     58      0
envR71               58      0      0      0      0      0      0     58
```

Build $X^T Z$

Creating the X‘Z matrix. Z is incidence matrix for random effects and X is incidence matrix for fixed effects.

```
> # Cross product and transpose
>     XtZ <- t(X) %*% Z
>     XtZ[, 1:10]
            genG01 genG02 genG03 genG04 genG05 genG06 genG07 genG08 genG09
(Intercept)      8      8      8      8      8      8      8      8      8
envB71           1      1      1      1      1      1      1      1      1
envL70           1      1      1      1      1      1      1      1      1
envL71           1      1      1      1      1      1      1      1      1
envN70           1      1      1      1      1      1      1      1      1
envN71           1      1      1      1      1      1      1      1      1
envR70           1      1      1      1      1      1      1      1      1
envR71           1      1      1      1      1      1      1      1      1
            genG10
(Intercept)      8
envB71           1
envL70           1
envL71           1
envN70           1
envN71           1
envR70           1
envR71           1
```

Build $Z^T X$

Creating the Z‘X matrix. Z is incidence matrix for random effects and X is incidence matrix for fixed effects.

```
> # Cross product and transpose
>     ZtX <- t(Z) %*% X
>     ZtX[1:8, ]
       (Intercept) envB71 envL70 envL71 envN70 envN71 envR70 envR71
genG01           8      1      1      1      1      1      1      1
genG02           8      1      1      1      1      1      1      1
genG03           8      1      1      1      1      1      1      1
genG04           8      1      1      1      1      1      1      1
genG05           8      1      1      1      1      1      1      1
genG06           8      1      1      1      1      1      1      1
genG07           8      1      1      1      1      1      1      1
genG08           8      1      1      1      1      1      1      1
```

Build $Z^T Z$

Creating the Z‘Z matrix. Z is incidence matrix for random effects.

```
> # Cross product and transpose
>     ZtZ <- t(Z) %*% Z
>     ZtZ[1:7,1:7]
       genG01 genG02 genG03 genG04 genG05 genG06 genG07
genG01      8      0      0      0      0      0      0
genG02      0      8      0      0      0      0      0
genG03      0      0      8      0      0      0      0
```

```
genG04    0    0    0    8    0    0    0
genG05    0    0    0    0    8    0    0
genG06    0    0    0    0    0    8    0
genG07    0    0    0    0    0    0    8
```

Variance components

- Here we will not fit G matrix (relationship matrix), and assume G=1, covariance structure is absent.
- We will also observe λ i.e, shrinkage factor

```
>     I <- diag(ncol(Z))  # assuming G = I, No markers
>     lambda <- sigmae/sigmau #
```

Build $X^T y$

- Creating the X'y matrix. X is incidence matrix for fixed effects and Y is response variable.

```
> # Cross product and transpose
>     Xty <- t(X) %*% y
>     Xty
              [,1]
(Intercept) 950.006
envB71      142.478
envL70      129.687
envL71      145.687
envN70      109.483
envN71      133.362
envR70       95.098
envR71      103.862
```

Build $Z^T y$

- Creating the Z'y matrix. Z is incidence matrix for fixed effects and y is response variable.

```
> # Cross product and transpose
>     Zty <- t(Z) %*% y
>     head(Zty)
         [,1]
genG01 15.304
genG02 16.155
genG03 16.874
genG04 19.550
genG05 17.981
genG06 16.930
```

Now build LHS and RHS of Equation

```
> # Left hand side
>     LHS1 <- cbind(XtX, XtZ)
>     LHS2 <- cbind(ZtX, ZtZ + I * lambda)
>     LHS <- rbind(LHS1, LHS2)
```

```
> # Right Hand side
>     RHS <- rbind(Xty, Zty)
```

Solve the equation

- Here we will solve the MME by equation and obtain the **BLUEs** and **BLUPs**.
- We will add intercept to the BLUPs to obtain final breeding values/phenotypic BLUPs

```
> # Sol function to get solution
>     sol <- solve(LHS, RHS)
>     dim(sol)
[1] 66  1
> # First eight are fixed effects (BLUEs) for environments
>   Blues.env<-data.frame(Blues.env=sol[1:8, ])
>   Blues.env
```

	Blues.env
(Intercept)	1.5577414
envB71	0.8987759
envL70	0.6782414
envL71	0.9541034
envN70	0.3298966
envN71	0.7416034
envR70	0.0818793
envR71	0.2329828

```
>   str(Blues.env)
'data.frame':   8 obs. of  1 variable:
 $ Blues.env: num  1.558 0.899 0.678 0.954 0.33 ...
> # Nine to rest are eight are random effects (BLUPs)
> # BLUP
>   Blups.gy<-sol[9:66, ]
>   head(Blups.gy)
     genG01      genG02      genG03      genG04      genG05      genG06
-0.11618205 -0.02424449  0.05343249  0.34253347  0.17302696  0.05948244
> # Final genotypic values/breeding values for grain yield
>     bv.gy<-data.frame(Yield.blups=Blups.gy+sol[1,1])
```

Fit Mixed Model with lme4 Package

- In this we will use in built functions of package **lme4**to fit the mixed effect model with same variables used above.
- We will the obtain BLUPs and BLUEs and compare it with lme4 results.

```
>     mixemodel.fit<-lmer(yield ~ env + (1 | gen), data = demo.data)
>     summary(mixemodel.fit)
Linear mixed model fit by REML ['lmerMod']
Formula: yield ~ env + (1 | gen)
   Data: demo.data
```

```
REML criterion at convergence: 809.7

Scaled residuals:
    Min      1Q  Median      3Q     Max
-2.3272 -0.6881 -0.0006  0.5947  3.6310

Random effects:
 Groups   Name        Variance Std.Dev.
 gen      (Intercept) 0.1991   0.4462
 Residual             0.2509   0.5009
Number of obs: 464, groups:  gen, 58

Fixed effects:
            Estimate Std. Error t value
(Intercept)  1.55774    0.08808  17.686
envB71       0.89878    0.09301   9.663
envL70       0.67824    0.09301   7.292
envL71       0.95410    0.09301  10.258
envN70       0.32990    0.09301   3.547
envN71       0.74160    0.09301   7.973
envR70       0.08188    0.09301   0.880
envR71       0.23298    0.09301   2.505

Correlation of Fixed Effects:
       (Intr) envB71 envL70 envL71 envN70 envN71 envR70
envB71 -0.528
envL70 -0.528  0.500
envL71 -0.528  0.500  0.500
envN70 -0.528  0.500  0.500  0.500
envN71 -0.528  0.500  0.500  0.500  0.500
envR70 -0.528  0.500  0.500  0.500  0.500  0.500
envR71 -0.528  0.500  0.500  0.500  0.500  0.500  0.500
> # Now extract the fixed effects (BLUEs) for environmenet
>     Blues.env.lme4<-data.frame(Blues.env=mixemodel.fit@beta)
>     Blues.env.lme4
```

Blues.env
1.5577414
0.8987759
0.6782414
0.9541034
0.3298966
0.7416034
0.0818793
0.2329828

```
> # Now extract the random effects (BLUPs)
>     Blups.gy.lme4<-ranef(mixemodel.fit)$gen
> # Now extract the intercept and add it to random effects
>     intercept <- fixef(mixemodel.fit)[1]
> # Now add intercept to blups to get genotypic values
>     Bv.gy.lme4<-data.frame(Yield.blup=intercept+ranef(mixemodel.fit)$gen)
```

Now check the results of MME we solve with lme4

- We will round the values first and check whether two are equal.

```
> # Blues from both
>     table(round(Blues.env$Blues.env,2)==round(Blues.env.lme4$Blues.env,2))

TRUE
   8
> # Now for Blups
>     table(round(bv.gy$Yield.blups, 2)==round(Bv.gy.lme4$X.Intercept., 2))

TRUE
  58
```

*Note: We confirm our results with **lme4** package.

Additional Literature on Mixed Models

Here, I am giving link to some of useful resources on Mixed model analysis in Crops

- Application of mixed models in Plant Breeding
- Towards understanding and use of mixed-model analysisof agricultural experiments
- Mixed models with R
- Introduction to linear mixed models
- Fitting Linear Mixed-Effects Models Using lme4
- Long-Term Experiments with cropping systems: Case studies on data analysis
- A brief introduction to mixed effects modelling and multi-model inference in ecology
- Perils and pitfalls of mixed-effects regression models in biology
- Genetic Data Analysis for Plant and Animal Breeding(Book)
- Linear Mixed-Effects Model (Book)

Credit and Courtsey to Dr. Gota Morota from VT University, USA

For any suggestions or comments, please feel to reach at waseem.hussain@irri.org; m.anumalla@irri.org