


Applied Statistical Methods - Solution 10

AUTHOR
Peter von Rohr

PUBLISHED
May 13, 2024

WEBR STATUS
 Ready!

Problem 1: Sire Model

Use the dataset available from the address shown below to predict sire-breeding values using a sire model.

https://charlotte-ngs.github.io/asmasss2024/data/asm_ped_sim_data.csv

Hints

- The variance component σ_s^2 of the sire effect can be assumed to be 2.25.
- The variance component σ_e^2 of the random residuals is 36.
- Sex is modelled as a fixed effect.
- The inverse sire relationship matrix can be computed using the function `getAInv()` from the `pedigreemm` package.

Solution

Specify the model

$$y = Xb + Zs + e$$

with vectors

- y of length n containing known phenotypic observations
- b of length p containing unknown fixed effects
- s of length q containing unknown random sire breeding values
- e of length n containing unknown random residuals

Known design matrices

- X of dimension $n \times p$ linking fixed effects to observations and
- Z of dimension $n \times q$ linking random breeding values to observations

The expected values and co-variance matrices of the random effects are

$$E \begin{bmatrix} y \\ s \\ e \end{bmatrix} = \begin{bmatrix} Xb \\ 0 \\ 0 \end{bmatrix}$$

$$\text{var} \begin{bmatrix} y \\ s \\ e \end{bmatrix} = \begin{bmatrix} V & ZG_s & R \\ G_s Z^T & G_s & 0 \\ R & 0 & R \end{bmatrix}$$

with $R = I * \sigma_e^2$, $G_s = A_s \sigma_s^2$ and $V = ZG_s Z^T + R$.

Read the data

 Run Code

```
1 # read data to data.frame
2 s_ex10_p01 <- "https://charlotte-ngs.github.io/asmasss2024/data/asm_ped_sim_data.csv"
3 df_lme <- read.table(s_ex10_p01, header = T, sep = ",")
```

4 df_lme

	ID	SIRE	DAM	SEX	P
1	5	1	4	f	16.7
2	6	2	4	f	13.9
3	7	1	3	m	26.0
4	8	2	3	m	4.3
5	9	1	6	m	18.8
6	10	8	5	m	5.2
7	11	1	6	m	6.6
8	12	8	5	f	27.5

- Inverse Sire Relationship Matrix

▶ Run Code



```

1 # construct vector of all sires
2 vec_sire <- unique(df_lme$SIRE)
3 vec_sire <- vec_sire[!is.na(vec_sire)]
4 n_nr_sire <- length(vec_sire)
5 # get sire of sires
6 vec_sire_sire <- sapply(vec_sire,
7                           function(x) ifelse(is.element(x, df_lme$ID),
8                                                 df_lme$SIRE[df_lme$ID == x], NA))
9 # specify sire pedigree
10 library(pedigreemm)
11 ped_sire <- pedigree(sire = vec_sire_sire,
12                      dam = rep(NA, n_nr_sire),
13                      label = as.character(vec_sire))
14 # inverse sire relationship matrix
15 mat_inv_sire <- as.matrix(getAInv(ped_sire))
16 dimnames(mat_inv_sire) <- NULL
17 mat_inv_sire

```

	[,1]	[,2]	[,3]
[1,]	1	0.0000000	0.0000000
[2,]	0	1.3333333	-0.6666667
[3,]	0	-0.6666667	1.3333333

Setup mixed model equations

$$\begin{bmatrix} X^T X & X^T Z \\ Z^T X & Z^T Z + \lambda_s * A_s^{-1} \end{bmatrix} \begin{bmatrix} \hat{b} \\ \hat{s} \end{bmatrix} = \begin{bmatrix} X^T y \\ Z^T y \end{bmatrix}$$

Get the known components from the data into the mixed-model equations

- Design matrix X

▶ Run Code



```

1 # matrix X
2 mat_X <- model.matrix(P ~ SEX, data = df_lme)
3 attr(mat_X, "assign") <- NULL

```

```

4 attr(mat_X, "contrasts") <- NULL
5 dimnames(mat_X) <- NULL
6 mat_X

```

```

[,1] [,2]

```

```

[1,] 1 0
[2,] 1 0
[3,] 1 1
[4,] 1 1
[5,] 1 1
[6,] 1 1
[7,] 1 1
[8,] 1 0

```

- Design matrix Z

▶ Run Code



```

1 # matrix Z
2 mat_Z <- model.matrix(P ~ 0 + as.factor(SIRE), data = df_lme)
3 attr(mat_Z, "assign") <- NULL
4 attr(mat_Z, "contrasts") <- NULL
5 dimnames(mat_Z) <- NULL
6 mat_Z

```

```

[,1] [,2] [,3]

```

```

[1,] 1 0 0
[2,] 0 1 0
[3,] 1 0 0
[4,] 0 1 0
[5,] 1 0 0
[6,] 0 0 1
[7,] 1 0 0
[8,] 0 0 1

```

- Variance ration $\lambda_s = \sigma_e^2 / \sigma_s^2$

▶ Run Code



```

1 # variance components
2 sigma_u2 <- 9
3 sigma_s2 <- sigma_u2 / 4
4 sigma_e2 <- 36
5 # lambda
6 n_lambda_s <- sigma_e2 / sigma_s2
7 n_lambda_s

```

```

[1] 16

```

- Mixed model equations

▶ Run Code



```

1 # coefficient matrix
2 mat_xtx <- crossprod(mat_X)

```

```

3  mat_xtz <- crossprod(mat_X, mat_Z)
4  mat_ztx <- t(mat_xtz)
5  mat_ztz_lam_a_inv <- crossprod(mat_Z) + n_lambda_s * mat_inv_sire
6  mat_coef <- rbind(cbind(mat_xtx, mat_xtz),
7                    cbind(mat_ztx, mat_ztz_lam_a_inv))
8  # right-hand side
9  mat_rhs <- rbind(crossprod(mat_X, df_lme$P),
10                  crossprod(mat_Z, df_lme$P))
11 # solutions
12 mat_sol <- solve(mat_coef, mat_rhs)
13 mat_sol

```

```

      [,1]

```

```

[1,] 19.4721453
[2,] -7.4820069
[3,]  0.6328720
[4,] -0.6878893
[5,] -0.2614187

```

Results

The first two numbers of the solutions correspond to estimates \hat{b} which contains the intercept and the difference between group means of sex **f** and **m**. The remaining numbers in the solutions are the predicted breeding values of the three sires **1**, **2** and **8**. At this point the numeric values of the predicted breeding values are not interesting. What we are interested is the ranking of the sires according to the breeding values. This is obtained by

▶ Run Code



```

1  vec_sire_pbv <- mat_sol[3:nrow(mat_sol),1]
2  names(vec_sire_pbv) <- c("Sire 1", "Sire 2", "Sire 8")
3  vec_sire_pbv[order(vec_sire_pbv, decreasing = T)]

```

```

Sire 1      Sire 8      Sire 2
0.6328720 -0.2614187 -0.6878893

```