Peter von Rohr
Institute of Agricultural Sciences
D-USYS
ETH Zurich

751-7602-00 V
Solutions for Exam in
Applied Statistical Methods
in Animal Sciences
Summer Semester 2022

Date: 2022-05-30

Name:

Legi-Nr:

Problem	Maximum Number of Points	Number of Points Reached
1	15	
2	6	
3	9	
4	33	
5	30	
Total	93	

Questions in German are in italics

Problem 1: Fixed Linear Effects Model

The following dataset on the slaughter weight and the sex of 12 beef animals is given.

Gegeben ist der folgende Datensatz zum Schlachtgewicht und zum Geschlecht von 12 Fleischrindern.

Animal	Sex	Slaughter Weight
1	female	276.4
2	female	274.6
3	female	285.3
4	female	275.7
5	castrate	304.2
6	castrate	293.4
7	castrate	284.8
8	castrate	296.5
9	male	286.6
10	male	300.9
11	male	290.5
12	male	302.9

The data is available from the address below and can be read by the function `readr::read_csv()`

Die Daten sind unter der nachfolgenden Adresse verfügbar und können mit der Funktion `readr::read_csv()` gelesen werden.

```
## https://charlotte-ngs.github.io/asmss2022/data/asm_exam_p01.csv
```

a) Do an F-test with the data above to answer the question whether the fixed effect of the 'Sex' of the animal has any influence at all on the slaugher weight.

Verwenden Sie einen F-Test zur Beantwortung der Frage ob der fixe Effekt des Geschlechts des Tieres überhaupt einen Einfluss auf das Schlachtgewicht hat.

2

Solution

The F-Test can be done either by an analysis of variance using `aov()` or by directly fitting the fixed linear effects model with `lm()`. The `lm()` is shown later, hence, we use `aov()` here.

```
aov_sw_sex <- aov(`Slaughter Weight` ~ Sex, tbl_sw_p01)
summary(aov_sw_sex)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## Sex          2  768.9   384.5   7.626 0.0116 *
## Residuals    9  453.7    50.4
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

From the output, we can see that it has an influence with a error probability of around 0.012.

b) Fit the linear fixed effects model showing the effects of the different levels of ‘Sex‘ on ‘Slaughter Weight‘. What is the order of the different levels of the factor ‘Sex‘ when ordering them according to the size of the effect obtained from the fitted model?

Passen Sie ein lineares fixes Modell an die Daten an, welches den Einfluss des Geschlechts auf das Schlachgewicht zeigt. Wie lautet die Reihenfolge der Effektstufen des Faktors Geschlecht, wenn diese nach der Effektgrösse aus dem geschätzten Modell sortiert werden?

4

Solution

The model is fitted as

```
lm_sw_sex <- lm(`Slaughter Weight` ~ Sex, tbl_sw_p01)
smry_sw_sex <- summary(lm_sw_sex)
smry_sw_sex
```

```
##
## Call:
## lm(formula = `Slaughter Weight` ~ Sex, data = tbl_sw_p01)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.925 -3.731 -1.462  6.081  9.475
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  294.725      3.550  83.019  2.7e-14 ***
## Sexfemale    -16.725      5.021  -3.331  0.00878 **
## Sexmale        0.500      5.021   0.100  0.92285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.1 on 9 degrees of freedom
## Multiple R-squared:  0.6289, Adjusted R-squared:  0.5464
## F-statistic: 7.626 on 2 and 9 DF,  p-value: 0.01155
```

The order of the effects from smallest to largest is given by

```
vec_eff_sex_est <- c(smry_sw_sex$coefficients["(Intercept)","Estimate"],
                     smry_sw_sex$coefficients["(Intercept)","Estimate"] +
                       smry_sw_sex$coefficients["Sexfemale","Estimate"],
                     smry_sw_sex$coefficients["(Intercept)","Estimate"] +
                       smry_sw_sex$coefficients["Sexmale","Estimate"])
names(vec_eff_sex_est) <- c("Sexcastrate","Sexfemale","Sexmale")
names(vec_eff_sex_est)[order(vec_eff_sex_est)]
```

```
## [1] "Sexfemale"   "Sexcastrate" "Sexmale"
```

c) Show how the different effect estimates (Intercept and factor levels of 'Sex') are computed from a solution to the least square normal equations using the data on slaughter weight and sex when treatment contrasts are used.

Zeigen Sie wie die Schätzwerte der verschiedenen Effekte (Achsenabschnitt und die Faktoren des Geschlechteffekts) aus einer Lösung der Least Squares-Normalgleichungen berechnet werden für die Daten zum Schlachtgewicht und Geschlecht unter der Verwendung von Treatment-Kontrasten.

9

Solution

Least squares normal equations are given by

$$X^T X b^0 = X^T y$$

A solution b^0 is obtained as

$$b^0 = (X^T X)^- X^T y$$

```
# model matrix
mat_X <- model.matrix(lm(`Slaughter Weight` ~ 0 + Sex, data = tbl_sw_p01))
attr(mat_X, "contrasts") <- NULL
attr(mat_X, "assign") <- NULL
colnames(mat_X) <- NULL
# add intercept
mat_X <- cbind(matrix(1, nrow = nrow(mat_X)), mat_X)
mat_X
```

```
##    [,1] [,2] [,3] [,4]
## 1     1    0    1    0
## 2     1    0    1    0
## 3     1    0    1    0
## 4     1    0    1    0
## 5     1    1    0    0
## 6     1    1    0    0
## 7     1    1    0    0
## 8     1    1    0    0
## 9     1    0    0    1
## 10    1    0    0    1
## 11    1    0    0    1
## 12    1    0    0    1
```

```
mat_xtx <- crossprod(mat_X)
mat_xtx_ginv <- MASS::ginv(mat_xtx)
vec_y <- tbl_sw_p01$`Slaughter Weight`
mat_xty <- crossprod(mat_X, vec_y)
mat_b0 <- mat_xtx_ginv %*% mat_xty
mat_b0
```

```
##          [,1]
## [1,] 216.9875
## [2,]  77.7375
## [3,]  61.0125
## [4,]  78.2375
```

Using treatment contrasts, we have

```
mat_cont_treat <- contrasts(as.factor(tbl_sw_p01$Sex))
mat_cont_treat <- cbind(matrix(1, nrow = nrow(mat_cont_treat)), mat_cont_treat)
mat_cont_treat
```

```
##            female male
## castrate 1      0    0
## female   1      1    0
## male     1      0    1
```

Estimable function

```
mat_estf <- solve(mat_cont_treat)
row.names(mat_estf)[1] <- "(Intercept)"
mat_estf
```

```
##             castrate female male
## (Intercept)        1      0    0
## female            -1      1    0
## male              -1      0    1
```

The intercept is obtained as the mean of all castrate animals

```
mean(tbl_sw_p01$`Slaughter Weight`[tbl_sw_p01$Sex == "castrate"])
```

```
## [1] 294.725
```

The effect for female are obtained from the second row

```
mat_b0[3] - mat_b0[2]
```

```
## [1] -16.725
```

The effect for male

```
mat_b0[4] - mat_b0[2]
```

```
## [1] 0.5
```

Comparing this to the output from `lm()`

```
summary(lm(`Slaughter Weight` ~ Sex, data = tbl_sw_p01))
```

```
##
## Call:
## lm(formula = `Slaughter Weight` ~ Sex, data = tbl_sw_p01)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.925 -3.731 -1.462  6.081  9.475
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  294.725      3.550  83.019  2.7e-14 ***
## Sexfemale    -16.725      5.021  -3.331  0.00878 **
## Sexmale        0.500      5.021   0.100  0.92285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.1 on 9 degrees of freedom
```

```
## Multiple R-squared:  0.6289, Adjusted R-squared:  0.5464
## F-statistic: 7.626 on 2 and 9 DF,  p-value: 0.01155
```

Problem 2: Linear Regression

The following dataset contains the logarithm of methane emission (`lCH4`) and the logarithm of dry matter intake (`lDMI`) of 15 cows.

Der folgende Datensatz enthält die logarithmierten Werte der Methanemmissionen (`lCH4`) und der täglichen Futteraufnahme (`lDMI`) für 15 Kühe.

Animal	lDMI	lCH4
1	9.51	12.97
2	13.79	20.15
3	10.30	11.70
4	9.35	15.23
5	12.87	17.74
6	1.60	0.85
7	10.77	16.35
8	13.09	19.73
9	6.33	8.79
10	5.38	11.05
11	2.74	3.16
12	9.66	12.79
13	11.15	11.77
14	8.08	16.95
15	4.80	4.70

The data is available from the address below and can be read by the function `readr::read_csv()`

Die Daten sind unter der nachfolgenden Adresse verfügbar und können mit der Funktion `readr::read_csv()` gelesen werden.

```
## https://charlotte-ngs.github.io/asmss2022/data/asm_exam_p02.csv
```

a) Fit the linear regression model of 'lCH4' on 'lDMI'.

Passen Sie ein lineares Regressionsmodell von 'lCH4' auf 'lDMI' an.

2

Solution

The linear regression model of `lCH4` on `lDMI` is given by

```
lm_lch4_ldmi <- lm(lCH4 ~ lDMI, data = tbl_ch4_p02)
summary(lm_lch4_ldmi)
```

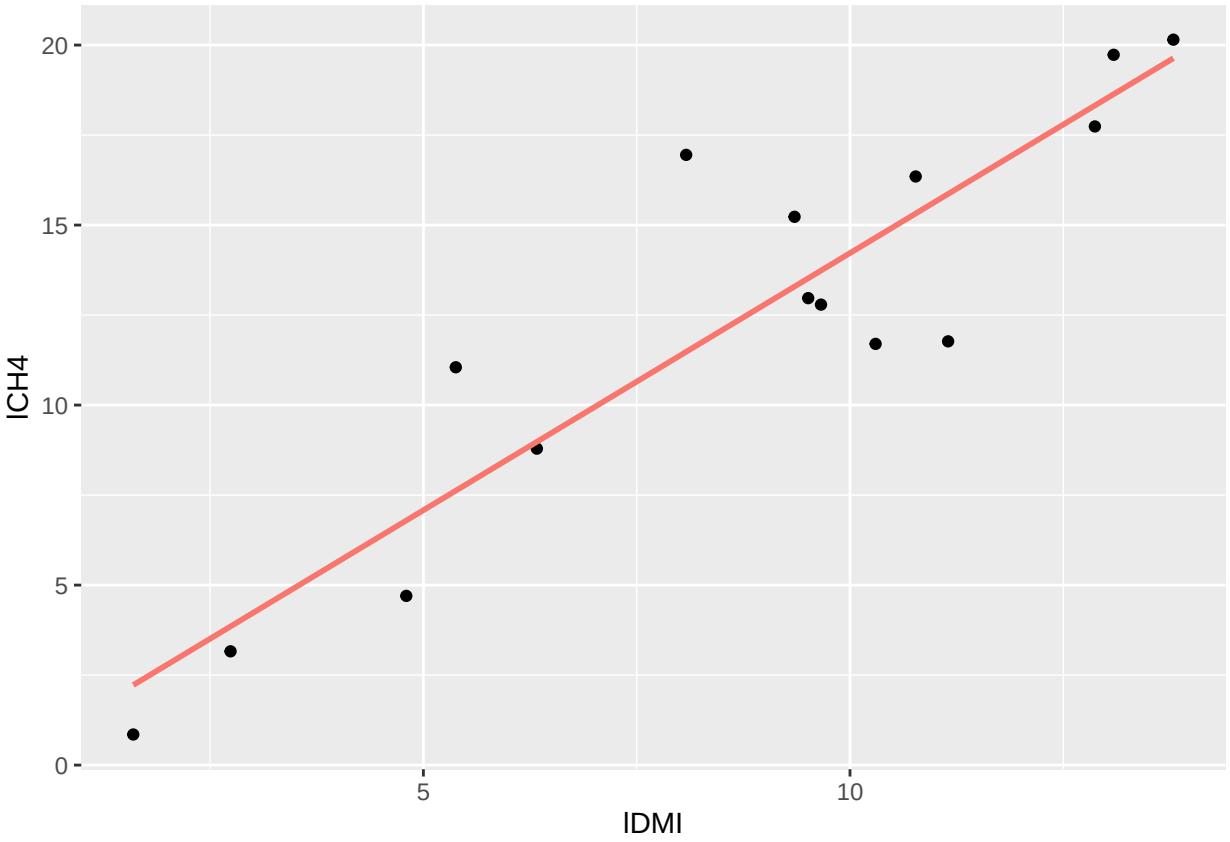
```
##
## Call:
## lm(formula = lCH4 ~ lDMI, data = tbl_ch4_p02)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4.0944 -1.1597 -0.5518  1.0615  5.4708
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.06212    1.67729  -0.037    0.971
## lDMI         1.42839    0.17933   7.965 2.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.508 on 13 degrees of freedom
## Multiple R-squared:  0.8299, Adjusted R-squared:  0.8168
## F-statistic: 63.44 on 1 and 13 DF,  p-value: 2.346e-06
```

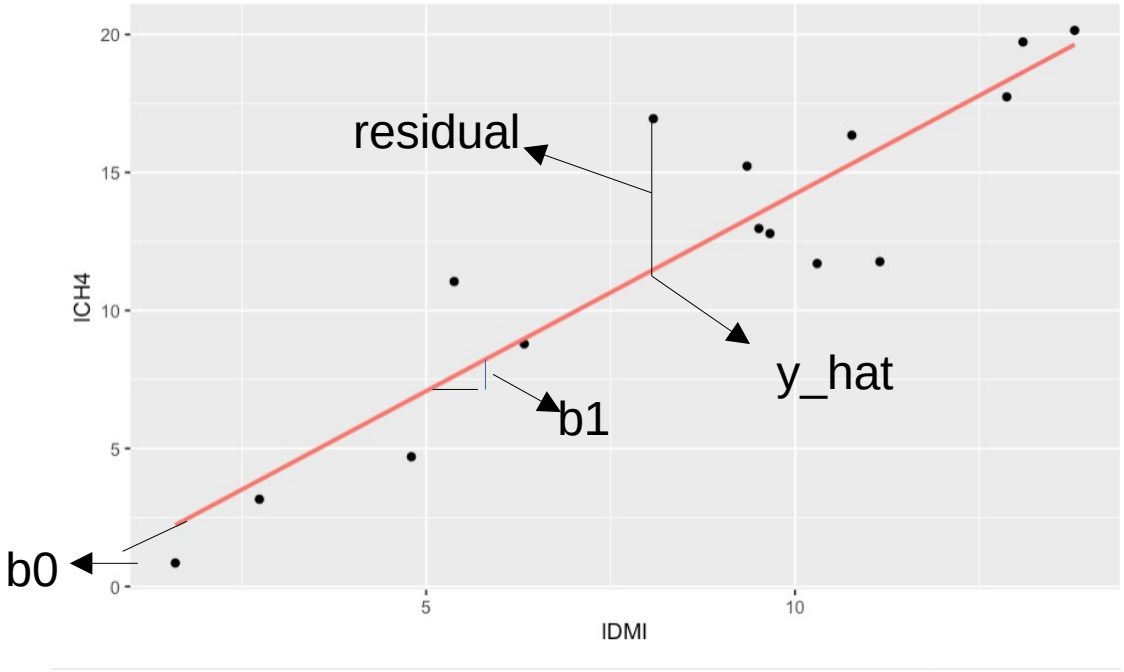
b) Show in the plot below, the estimates of the model coefficients obtained from the linear regression in Problem 2a. For a selected example observation, show the fitted value and the residual belonging to that selected observation.

Zeigen Sie im nachfolgenden Plot die geschätzten Modellkoeffizienten der linearen Regression aus der Aufgabe 2a). Für einen bestimmten Beobachtungswert zeigen Sie den Modellschätzwert und das Residuum, welches zur ausgewählten Beobachtung gehört.

4



Solution



Problem 3: Model Selection

The following dataset contains fat yield (`fat`) of dairy cows as a response variable. Lactation number (`lact`), days in milk (`dim`) and height of the cow (`hei`) are avilable as predictor variables.

Der folgende Datensatz enhält Fettleistung (`fat`) von Milchkühen als eine Zielvariable. Laktationsnummer (`lact`), Laktationslänge in Tagen (`dim`) und Grösse der Kuh (`hei`) sind verfügbar als beschreibende Variablen.

id	lact	dim	fat	hei
5649	2	301	936	153
5361	1	447	955	150
4312	2	368	615	147
6324	1	294	984	149
6190	2	384	1040	150
5923	2	399	899	148
6187	1	280	1198	149
5653	3	464	903	148
5670	4	305	881	145
5498	1	321	832	147
5907	1	328	1090	150
6299	3	291	745	148
5224	2	295	805	153
5339	3	337	822	145
6187	2	342	1121	147
5795	3	297	1146	148
6255	1	610	1204	146
6507	2	305	794	149
6470	1	323	761	148
5737	4	273	935	146

The data is available from the address below and can be read by the function `readr::read_csv()`

Die Daten sind unter der nachfolgenden Adresse verfügbar und können mit der Funktion `readr::read_csv()` gelesen werden.

```
## https://charlotte-ngs.github.io/asmss2022/data/asm_exam_p03.csv
```

a) Use model selection based on the C_p-value on the above dataset to find the best model. Which predictor variables are included in the best model based on the C_p-value? Which are the parameter estimates of the best model?

Verwenden Sie Modellselektion basierend auf dem C_p-Wert für den oben gegebenen Datensatz. Welche beschreibenden Variablen sind im besten Modell nach C_p-Wert enthalten? Wie lauten die geschätzten Parameter des besten Modells?

6

Solution

Start with the full model

```r
lm_milk_full <- lm(fat ~ lact + dim + hei, data = tbl_milk_p03)
summary(lm_milk_full)
```

```
##
## Call:
## lm(formula = fat ~ lact + dim + hei, data = tbl_milk_p03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -326.33 -114.20  -17.78  100.74  263.82
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  652.4398  3070.5691   0.212    0.834
## lact         -27.3075    44.8676  -0.609    0.551
## dim            0.4207     0.5161   0.815    0.427
## hei            1.2835    19.9750   0.064    0.950
##
## Residual standard error: 167 on 16 degrees of freedom
## Multiple R-squared:  0.09409,    Adjusted R-squared:  -0.07577
## F-statistic: 0.5539 on 3 and 16 DF,  p-value: 0.6529
```

Run model selection

```r
olsrr::ols_step_best_subset(lm_milk_full)
```

```
##   Best Subsets Regression
## ---------------------------
## Model Index    Predictors
## ---------------------------
##      1         dim
##      2         lact dim
##      3         lact dim hei
## ---------------------------
##
##                                                    Subsets Regression Summary
## ------------------------------------------------------------------------------------------------------------
##                        Adj.        Pred
## Model    R-Square    R-Square    R-Square     C(p)       AIC         SBIC        SBC          MSEP
## ------------------------------------------------------------------------------------------------------------
##   1        0.0645      0.0125     -0.2701    0.5230    263.6594    207.6551    266.6466    512356.18
##   2        0.0939     -0.0128     -0.3178    2.0041    265.0213    209.6372    269.0043    527284.53
##   3        0.0941     -0.0758     -0.4252    4.0000    267.0162    212.1336    271.9948    562291.74
## ------------------------------------------------------------------------------------------------------------
## AIC: Akaike Information Criteria
##  SBIC: Sawa's Bayesian Information Criteria
##  SBC: Schwarz Bayesian Criteria
##  MSEP: Estimated error of prediction, assuming multivariate normality
##  FPE: Final Prediction Error
##  HSP: Hocking's Sp
##  APC: Amemiya Prediction Criteria
```

Based on the C_p value, the model with only `dim` is the best model.

The estimated parameters of the best model are

```r
lm_milk_best <- lm(fat ~ dim, data = tbl_milk_p03)
summary(lm_milk_best)
```

```
##
## Call:
## lm(formula = fat ~ dim, data = tbl_milk_p03)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -328.25 -102.60  -29.27  101.33  298.96
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 758.3638   161.0832   4.708 0.000175 ***
## dim           0.5024     0.4511   1.114 0.280008
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 160 on 18 degrees of freedom
## Multiple R-squared:  0.06448,    Adjusted R-squared:  0.0125
## F-statistic: 1.241 on 1 and 18 DF,  p-value: 0.28
```

b) Verify the result of the model selection using an analysis of variance (‘aov()‘) on the full model. Are you getting the same result as shown in Problem 3a?

Verifizieren Sie das Resultat der Modellselektion mit einer Varianzanalyse (aov()) auf dem vollen Modell. Erhalten Sie das gleiche Resultat, wie in Aufgabe 3a?

3

Solution

The analysis of variance on the full model gives

```
aov_milk <- aov(fat ~ lact + dim + hei, data = tbl_milk_p03)
summary(aov_milk)
```

```
##             Df Sum Sq Mean Sq F value Pr(>F)
## lact         1  26549   26549   0.952  0.344
## dim          1  19685   19685   0.706  0.413
## hei          1    115     115   0.004  0.950
## Residuals   16 446263   27891
```

Based on the computed values of the F-test statistic, it would be possible that `dim` and `hei` have about the same importance. Hence the result of the model selection could not be confirmed.

Problem 4: Pedigree-Based BLUP

The dataset shown below shows observations of a trait called `P` for 6 animals. The phenotypic variance is assumed to be 80. The heritability is 0.2.

Der unten gezeigte Datensatz zeigt Beobachtungen eines Merkmals namens `P` für 6 Tiere. Die phänotypische Varianz beträgt 80. *Die Erblichkeit ist* 0.2.

ID	SIRE	DAM	SEX	P
4	1	3	m	22.4
5	2	3	m	49.2
6	1	NA	m	18.0
7	2	NA	f	11.7
8	5	7	f	27.3
9	5	7	f	28.3

The data is available from the address below and can be read by the function `readr::read_csv()`

Die Daten sind unter der nachfolgenden Adresse verfügbar und können mit der Funktion `readr::read_csv()` gelesen werden.

```
## https://charlotte-ngs.github.io/asmss2022/data/asm_exam_p04.csv
```

a) Use the above shown dataset to predict breeding values using a sire model. In that model include `SEX` as a fixed effect. Specify all model components with expected values and variance-covariance matrices for all random effects in the model. The ratio between residual variance and sire variance can be assumed as 19.

Schätzen Sie Zuchtwerte mit dem oben gezeigten Datensatz mit einem Vatermodell. In diesem Modell soll `SEX` *als fixer Effekt modelliert werden. Geben Sie alle Modellkomponenten an und spezifizieren Sie Erwartungswerte und Varianz-Kovarianzmatrizen für alle zufälligen Effekte im Modell. Das Verhältnis zwischen Restvarianz und Vatervarianz kann angenommen werden als* 19.

15

Solution

The sire model is given by

$$y = Xb + Zs + e$$

with y the vector of observations, b the vector of fixed effects, s the vector of random sire breeding values, e the vector of random residuals and design matrices X and Z. Inserting information from the data leads to

```
# vector y
vec_y <- tbl_data_p04$P
# matrix X
mat_X <- model.matrix(lm(P ~ 0 + SEX, data = tbl_data_p04))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
# vector b
```

```r
vec_b <- c("b_{f}", "b_{m}")
# martix Z
mat_Z <- model.matrix(lm(P ~ 0 + as.factor(SIRE), data = tbl_data_p04))
attr(mat_Z, "assign") <- NULL
attr(mat_Z, "contrasts") <- NULL
colnames(mat_Z) <- NULL
# vector s
vec_s <- c("s_1", "s_2", "s_5")
# vector e
vec_e <- sapply(1:nrow(tbl_data_p04), function(x) paste("e_", x, sep = ""))
# sigma_e2
sigma_e2 <- sigma_p2 - sigma_s2
```

$$y = \begin{bmatrix} 22.4 \\ 49.2 \\ 18 \\ 11.7 \\ 27.3 \\ 28.3 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, b = \begin{bmatrix} b_f \\ b_m \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}, s = \begin{bmatrix} s_1 \\ s_2 \\ s_5 \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix}$$

The expected values are defined as

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

The variance-covariance matrices

$$var \begin{bmatrix} y \\ s \\ e \end{bmatrix} = \begin{bmatrix} ZSZ^T + R & ZS & R \\ SZ^T & S & 0 \\ R & 0 & R \end{bmatrix}$$

with

- $var(e) = R = I * \sigma_e^2$,
- $var(s) = S = A_s * \sigma_s^2$
- $\sigma_e^2 = 76$ the residual variance component
- $\sigma_s^2 = 4$ the sire variance component
- $\lambda_s = \sigma_e^2 / \sigma_s^2 = 19$
- A_s the sire relationship matrix

The sire relationshipmatrix A_s is

```r
vec_sire_id <- c(1,2,5)
ped_sire <- pedigreemm::pedigree(sire = c(NA, NA, 2),
                                 dam  = c(rep(NA, length(vec_sire_id))),
                                 label = as.character(vec_sire_id))
mat_A_sire <- as.matrix(pedigreemm::getA(ped = ped_sire))
mat_A_sire_inv <- as.matrix(pedigreemm::getAInv(ped = ped_sire))
```

$$A_s = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.5 \\ 0 & 0.5 & 1 \end{bmatrix}$$

Setting up mixed model equations

```
# coefficient matrix
mat_xtx <- crossprod(mat_X)
mat_xtz <- crossprod(mat_X, mat_Z)
mat_ztx <- t(mat_xtz)
mat_ztz_ainv_lambda <- crossprod(mat_Z) + lambda_s * mat_A_sire_inv
mat_coef_sire <- rbind(cbind(mat_xtx, mat_xtz), cbind(mat_ztx, mat_ztz_ainv_lambda))
# right hand side
mat_xty <- crossprod(mat_X, vec_y)
mat_zty <- crossprod(mat_Z, vec_y)
mat_rhs_sire <- rbind(mat_xty, mat_zty)
```

Solving MME

```
(mat_sol_sire <- solve(mat_coef_sire, mat_rhs_sire))
```

```
##         [,1]
##   21.6996387
##   30.2803502
## 1 -0.9600334
## 2  0.6790161
## 5  0.7610339
```

Solutions for fixed effects

```
n_nr_fix_effects <- length(vec_b)
mat_sol_sire[1:n_nr_fix_effects,]
```

```
##
## 21.69964 30.28035
```

Predicted breeding values for sires

```
(vec_sol_sire <- mat_sol_sire[(n_nr_fix_effects+1):nrow(mat_sol_sire),])
```

```
##          1          2          5
## -0.9600334  0.6790161  0.7610339
```

b) Predict breeding values for all animals using an animal model. `SEX` is modelled as fixed effect. Specify all model components with expected values and variance-covariance matrices for all random effects in the model.

Schätzen Sie Zuchtwerte für alle Tiere mit dem Tiermodell. `SEX` *soll als fixer Effekt modelliert werden. Geben Sie alle Modellkomponenten an und spezifizieren Sie Erwartungswerte und Varianz-Kovarianzmatrizen für alle zufälligen Effekte im Modell.*

15

Solution

The animal model is given by

$$y = Xb + Zu + e$$

with y the vector of observations, b the vector of fixed effects, u the vector of random breeding values, e the vector of random residuals and design matrices X and Z. Inserting information from the data leads to

```
# vector y
vec_y <- tbl_data_p04$P
n_nr_rec_p4 <- length(vec_y)
# matrix X
mat_X <- model.matrix(lm(P ~ 0 + SEX, data = tbl_data_p04))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
# vector b
vec_b <- c("b_{f}", "b_{m}")
# martix Z
mat_Z <- cbind(matrix(0, nrow = n_nr_rec_p4, ncol = 3), diag(1, nrow = n_nr_rec_p4))
# vector u
n_nr_ani_ped <- 9
vec_u <- sapply(1:n_nr_ani_ped, function(x) paste("u_", x, sep = ""))
# vector e
vec_e <- sapply(1:nrow(tbl_data_p04), function(x) paste("e_", x, sep = ""))
# sigma_e2
sigma_e2 <- sigma_p2 - sigma_u2
```

$$y = \begin{bmatrix} 22.4 \\ 49.2 \\ 18 \\ 11.7 \\ 27.3 \\ 28.3 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, b = \begin{bmatrix} b_f \\ b_m \end{bmatrix}, Z = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, u = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \end{bmatrix}$$

The expected values are defined as

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

The variance-covariance matrices

$$var \begin{bmatrix} y \\ u \\ e \end{bmatrix} = \begin{bmatrix} ZUZ^T + R & ZU & R \\ UZ^T & U & 0 \\ R & 0 & R \end{bmatrix}$$

with

- $var(e) = R = I * \sigma_e^2$,
- $var(u) = U = A * \sigma_u^2$
- $\sigma_e^2 = 64$ the residual variance component
- $\sigma_u^2 = 16$ the genetic additive variance component
- $\lambda = \sigma_e^2 / \sigma_u^2 = 4$
- A the numerator relationship matrix

The numerator relationshipmatrix A is

```
vec_ani_id <- c(1:n_nr_ani_ped)
ped_ani <- pedigreemm::pedigree(sire = c(NA, NA, NA, tbl_data_p04$SIRE),
                                dam  = c(NA, NA, NA, tbl_data_p04$DAM),
                                label = as.character(vec_ani_id))
mat_A_ani <- as.matrix(pedigreemm::getA(ped = ped_ani))
mat_A_ani_inv <- as.matrix(pedigreemm::getAInv(ped = ped_ani))
```

$$A = \begin{bmatrix} 1 & 0 & 0 & 0.5 & 0 & 0.5 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0.5 & 0 & 0.5 & 0.5 & 0.5 \\ 0 & 0 & 1 & 0.5 & 0.5 & 0 & 0 & 0.25 & 0.25 \\ 0.5 & 0 & 0.5 & 1 & 0.25 & 0.25 & 0 & 0.125 & 0.125 \\ 0 & 0.5 & 0.5 & 0.25 & 1 & 0 & 0.25 & 0.625 & 0.625 \\ 0.5 & 0 & 0 & 0.25 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.25 & 0 & 1 & 0.625 & 0.625 \\ 0 & 0.5 & 0.25 & 0.125 & 0.625 & 0 & 0.625 & 1.125 & 0.625 \\ 0 & 0.5 & 0.25 & 0.125 & 0.625 & 0 & 0.625 & 0.625 & 1.125 \end{bmatrix}$$

Setting up mixed model equations

```
# coefficient matrix
mat_xtx <- crossprod(mat_X)
mat_xtz <- crossprod(mat_X, mat_Z)
mat_ztx <- t(mat_xtz)
mat_ztz_ainv_lambda <- crossprod(mat_Z) + lambda_u * mat_A_ani_inv
mat_coef_ani <- rbind(cbind(mat_xtx, mat_xtz), cbind(mat_ztx, mat_ztz_ainv_lambda))
# right hand side
mat_xty <- crossprod(mat_X, vec_y)
mat_zty <- crossprod(mat_Z, vec_y)
mat_rhs_ani <- rbind(mat_xty, mat_zty)
```

Solving MME

```
(mat_sol_ani <- solve(mat_coef_ani, mat_rhs_ani))
```

```
##         [,1]
##   20.5585929
##   29.6244238
## 1 -1.9129278
## 2  1.9129278
```

```
## 3  1.6838035
## 4 -0.9045468
## 5  4.2721538
## 6 -2.6408786
## 7  0.1151733
## 8  2.6989684
## 9  2.8100795
```

Solutions for fixed effects

```
n_nr_fix_effects <- length(vec_b)
mat_sol_ani[1:n_nr_fix_effects,]
```

```
##
## 20.55859 29.62442
```

Predicted breeding values for sires

```
(vec_sol_ani <- mat_sol_ani[(n_nr_fix_effects+1):nrow(mat_sol_ani),])
```

```
##          1          2          3          4          5          6          7
## -1.9129278  1.9129278  1.6838035 -0.9045468  4.2721538 -2.6408786  0.1151733
##          8          9
##  2.6989684  2.8100795
```

c) Compare the order of the sires according to the predicted breeding values from Problem 4a and 4b.

Vergleichen Sie die Reihenfolge der Stiere aufgrund der geschätzten Zuchtwerte aus den Aufgaben 4a und 4b.

3

Solution

The order of the breeding values for the sires with the sire model is

```
vec_sol_sire[order(vec_sol_sire, decreasing = TRUE)]
```

```
##          5          2          1
##  0.7610339  0.6790161 -0.9600334
```

Animal model

```
vec_sol_ani[order(vec_sol_ani, decreasing = TRUE)]
```

```
##          5          9          8          2          3          7          4
##  4.2721538  2.8100795  2.6989684  1.9129278  1.6838035  0.1151733 -0.9045468
##          1          6
## -1.9129278 -2.6408786
```

The order is the same.

Problem 5: Genomic Prediction of Breeding Values

The following dataset is used to predict genomic breeding values.

Der nachfolgende Datensatz wird zur Schätzung von genomischen Zuchtwerten verwendet.

ID	SIRE	DAM	SEX	P	SNP1	SNP2	SNP3
5	1	3	m	70.5	0	0	1
6	2	3	f	74.5	1	1	1
7	1	4	m	79.7	0	0	1
8	2	4	f	111.1	2	0	2
9	7	8	f	63.1	1	0	2
10	5	6	f	89.2	1	1	2
11	7	8	f	82.2	1	0	1
12	5	6	f	59.3	1	1	1

The data is available from the address below and can be read by the function `readr::read_csv()`

Die Daten sind unter der nachfolgenden Adresse verfügbar und können mit der Funktion ***`readr::read_csv()`*** *gelesen werden.*

```
## https://charlotte-ngs.github.io/asmss2022/data/asm_exam_p05.csv
```

a) Predict genomic breeding values based on the dataset shown above using a marker effect model. The ratio between residual variance and marker effect is 1.

Schätzen Sie genomische Zuchtwerte aufgrund des oben gezeigten Datensatzes mit einem Markereffektmodell. Das Verhältnis zwischen Restvarianz und Markervarianz beträgt 1.

15

Solution

In a marker effect model, the first thing is to estimate marker effects. Because, we have only three SNPs, this can be done with a linear regression.

```
lm_marker <- lm(P ~ SEX + SNP1 + SNP2 + SNP3, data = tbl_data_p05)
(smry_marker <- summary(lm_marker))
```

```
##
## Call:
## lm(formula = P ~ SEX + SNP1 + SNP2 + SNP3, data = tbl_data_p05)
##
## Residuals:
##          1          2          3          4          5          6          7
## -4.600e+00  1.686e+00  4.600e+00 -1.554e-15 -1.183e+01  1.183e+01  1.183e+01
##          8
## -1.351e+01
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   29.643     28.854   1.027    0.380
```

```
## SEXm          40.900     27.486   1.488    0.233
## SNP1          36.171     19.236   1.880    0.157
## SNP2           2.443     13.602   0.180    0.869
## SNP3           4.557     13.602   0.335    0.760
##
## Residual standard error: 14.69 on 3 degrees of freedom
## Multiple R-squared:  0.6552, Adjusted R-squared:  0.1954
## F-statistic: 1.425 on 4 and 3 DF,  p-value: 0.4015
```

From this we get the marker effect estimates as

```
(vec_marker_est <- smry_marker$coefficients[c("SNP1", "SNP2", "SNP3"), "Estimate"])
```

```
##      SNP1      SNP2      SNP3
## 36.171429  2.442857  4.557143
```

The predicted genomic breeding values are the dot product of the genotypes times the marker effects.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
tbl_geno <- tbl_data_p05 %>% select(SNP1, SNP2, SNP3)
mat_geno <- as.matrix(tbl_geno)
(vec_geno_bv_mem <- mat_geno %*% vec_marker_est)
```

```
##           [,1]
## [1,]  4.557143
## [2,] 43.171429
## [3,]  4.557143
## [4,] 81.457143
## [5,] 45.285714
## [6,] 47.728571
## [7,] 40.728571
## [8,] 43.171429
```

Alternatively, marker effects can also be estimated using a linear mixed effects model with the vector q as random effects, then we get

$$y = Xb + Wq + e$$

with y the vector of observations, b the vector of fixed effects, q the vector of random marker effects, e the vector of random residuals and design matrices X and W. Inserting information from the data leads to

```
# vector y
vec_y <- tbl_data_p05$P
n_nr_rec_p5 <- length(vec_y)
# matrix X
mat_X <- model.matrix(lm(P ~ 0 + SEX, data = tbl_data_p05))
attr(mat_X, "assign") <- NULL
```

```
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
# vector b
vec_b <- c("b_{f}", "b_{m}")
# martix W
tbl_geno <- tbl_data_p05 %>% select(SNP1, SNP2, SNP3)
mat_W <- as.matrix(tbl_geno)
# vector q
vec_q <- sapply(1:ncol(mat_W), function(x) paste("q_", x, sep = ""))
# vector e
vec_e <- sapply(1:n_nr_rec_p5, function(x) paste("e_", x, sep = ""))
```

$$y = \begin{bmatrix} 70.5 \\ 74.5 \\ 79.7 \\ 111.1 \\ 63.1 \\ 89.2 \\ 82.2 \\ 59.3 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, b = \begin{bmatrix} b_f \\ b_m \end{bmatrix}, W = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ 0 & 0 & 1 \\ 2 & 0 & 2 \\ 1 & 0 & 2 \\ 1 & 1 & 2 \\ 1 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, q = \begin{bmatrix} q_1 \\ q_2 \\ q_3 \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{bmatrix}$$

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

The variance-covariance matrices

$$var \begin{bmatrix} y \\ q \\ e \end{bmatrix} = \begin{bmatrix} ZQZ^T + R & ZQ & R \\ QZ^T & Q & 0 \\ R & 0 & R \end{bmatrix}$$

with

- $var(e) = R = I * \sigma_e^2$,
- $var(q) = Q = I * \sigma_q^2$
- σ_e^2 the residual variance component
- σ_q^2 the marker variance component

Setting up mixed model equations

```
# coefficient matrix
mat_xtx <- crossprod(mat_X)
mat_xtw <- crossprod(mat_X, mat_W)
mat_wtx <- t(mat_xtw)
mat_wtw <- crossprod(mat_W)
mat_wtw_qinv_lambda <- mat_wtw + lambda_q * diag(1,nrow = nrow(mat_wtw))
mat_coef_marker <- rbind(cbind(mat_xtx, mat_xtw), cbind(mat_wtx, mat_wtw_qinv_lambda))
# right hand side
mat_xty <- crossprod(mat_X, vec_y)
mat_wty <- crossprod(mat_W, vec_y)
mat_rhs_marker <- rbind(mat_xty, mat_wty)
```

Solving MME

```
(mat_sol_marker <- solve(mat_coef_marker, mat_rhs_marker))
```

```
##           [,1]
##      54.986667
##      69.063333
## SNP1 14.680000
## SNP2 -2.536667
## SNP3  6.036667
```

Solutions for fixed effects

```
n_nr_fix_effects <- length(vec_b)
mat_sol_marker[1:n_nr_fix_effects,]
```

```
##
## 54.98667 69.06333
```

The solution for the marker effects

```
(vec_sol_geno <- mat_sol_marker[(n_nr_fix_effects+1):nrow(mat_sol_marker),])
```

```
##      SNP1      SNP2      SNP3
## 14.680000 -2.536667  6.036667
```

Predicted genomic breeding values for genotyped animals

```
mat_W %*% vec_sol_geno
```

```
##           [,1]
## [1,]  6.036667
## [2,] 18.180000
## [3,]  6.036667
## [4,] 41.433333
## [5,] 26.753333
## [6,] 24.216667
## [7,] 20.716667
## [8,] 18.180000
```

b) Predict genomic breeding values based on the dataset shown above using a breeding value-based model. The ratio between residual variance and genomic variance is 1.

Schätzen Sie genomische Zuchtwerte aufgrund des oben gezeigten Datensatzes mit einem Zuchtwertmodell. Das Verhältnis der Restvarianz zur genomischen Varianz beträgt 1.

15

Solution

The genomic breeding values using a breeding value based model are obtained by the following linear mixed effects model

$$y = Xb + Zg + e$$

with y the vector of observations, b the vector of fixed effects, g the vector of random genomic breeding values, e the vector of random residuals and design matrices X and Z. Inserting information from the data leads to

```
# vector y
vec_y <- tbl_data_p05$P
n_nr_rec_p5 <- length(vec_y)
# matrix X
mat_X <- model.matrix(lm(P ~ 0 + SEX, data = tbl_data_p05))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
# vector b
vec_b <- c("b_{f}", "b_{m}")
# martix Z
mat_Z <- diag(1, nrow = n_nr_rec_p5)
# vector g
vec_g <- sapply(1:n_nr_rec_p5, function(x) paste("g_", x, sep = ""))
# vector e
vec_e <- sapply(1:nrow(tbl_data_p05), function(x) paste("e_", x, sep = ""))
```

$$y = \begin{bmatrix} 70.5 \\ 74.5 \\ 79.7 \\ 111.1 \\ 63.1 \\ 89.2 \\ 82.2 \\ 59.3 \end{bmatrix}, X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, b = \begin{bmatrix} b_f \\ b_m \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, g = \begin{bmatrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \\ g_6 \\ g_7 \\ g_8 \end{bmatrix}, e = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{bmatrix}$$

The expected values are defined as

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

The variance-covariance matrices

$$var \begin{bmatrix} y \\ g \\ e \end{bmatrix} = \begin{bmatrix} ZHZ^T + R & ZH & R \\ HZ^T & H & 0 \\ R & 0 & R \end{bmatrix}$$

with

- $var(e) = R = I * \sigma_e^2$,
- $var(g) = H = G * \sigma_g^2$
- σ_e^2 the residual variance component
- σ_g^2 the genomic variance component
- G the genomic relationship matrix

The numerator relationshipmatrix G is

```
tbl_geno <- tbl_data_p05 %>% select(SNP1, SNP2, SNP3)
mat_geno <- as.matrix(tbl_geno)
# function to compute genomic relationship matrix
computeMatGrm <- function(pmatData) {
  matData <- pmatData
  # check the coding, if matData is -1, 0, 1 coded, then add 1 to get to 0, 1, 2 coding
  if (min(matData) < 0) matData <- matData + 1
  # Allele frequencies, column vector of P and sum of frequency products
  freq <- apply(matData, 2, mean) / 2
  P <- 2 * (freq - 0.5)
  sumpq <- sum(freq*(1-freq))
  # Changing the coding from (0,1,2) to (-1,0,1) and subtract matrix P
  Z <- matData - 1 - matrix(P, nrow = nrow(matData),
                            ncol = ncol(matData),
                            byrow = TRUE)
  # Z%*%Zt is replaced by tcrossprod(Z)
  return(tcrossprod(Z)/(2*sumpq))
}
# genomic relationship matrix
mat_G <- computeMatGrm(pmatData = mat_geno)
# test full rank
if (Matrix::rankMatrix(mat_G) < nrow(mat_G)){
  mat_G_star <- mat_G + 0.01 * diag(1, nrow = nrow(mat_G))
} else {
  mat_G_star <- mat_G
}
```

$$G = \begin{bmatrix} 0.8635 & -0.1656 & 0.8535 & -0.879 & -0.1656 & -0.4713 & 0.1401 & -0.1656 \\ -0.1656 & 0.4559 & -0.1656 & -0.2675 & -0.3694 & 0.1401 & -0.0637 & 0.4459 \\ 0.8535 & -0.1656 & 0.8635 & -0.879 & -0.1656 & -0.4713 & 0.1401 & -0.1656 \\ -0.879 & -0.2675 & -0.879 & 1.475 & 0.5478 & 0.242 & 0.0382 & -0.2675 \\ -0.1656 & -0.3694 & -0.1656 & 0.5478 & 0.4559 & 0.1401 & -0.0637 & -0.3694 \\ -0.4713 & 0.1401 & -0.4713 & 0.242 & 0.1401 & 0.6597 & -0.3694 & 0.1401 \\ 0.1401 & -0.0637 & 0.1401 & 0.0382 & -0.0637 & -0.3694 & 0.252 & -0.0637 \\ -0.1656 & 0.4459 & -0.1656 & -0.2675 & -0.3694 & 0.1401 & -0.0637 & 0.4559 \end{bmatrix}$$

Setting up mixed model equations

```
# coefficient matrix
mat_xtx <- crossprod(mat_X)
mat_xtz <- crossprod(mat_X, mat_Z)
mat_ztx <- t(mat_xtz)
mat_ztz_ginv_lambda <- crossprod(mat_Z) + lambda_g * solve(mat_G_star)
mat_coef_geno <- rbind(cbind(mat_xtx, mat_xtz), cbind(mat_ztx, mat_ztz_ginv_lambda))
# right hand side
mat_xty <- crossprod(mat_X, vec_y)
```

```
mat_zty <- crossprod(mat_Z, vec_y)
mat_rhs_geno <- rbind(mat_xty, mat_zty)
```

Solving MME

```
(mat_sol_geno <- solve(mat_coef_geno, mat_rhs_geno))
```

```
##              [,1]
##  [1,]  75.7126408
##  [2,]  87.6620777
##  [3,] -12.6076222
##  [4,]  -2.1995681
##  [5,] -12.5165331
##  [6,]  19.4215957
##  [7,]   6.0585527
##  [8,]   3.6793120
##  [9,]   0.5143261
## [10,]  -2.3500631
```

Solutions for fixed effects

```
n_nr_fix_effects <- length(vec_b)
mat_sol_geno[1:n_nr_fix_effects,]
```

```
## [1] 75.71264 87.66208
```

Predicted genomic breeding values for genotyped animals

```
(vec_sol_geno <- mat_sol_geno[(n_nr_fix_effects+1):nrow(mat_sol_geno),])
```

```
## [1] -12.6076222  -2.1995681 -12.5165331  19.4215957   6.0585527   3.6793120
## [7]   0.5143261  -2.3500631
```