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

751-7602-00 V
Solutions for Exam in Livestock Breeding and Genomics
Fall Semester 2023

Date: 2023-12-22

Name:

Legi-Nr:

Problem	Maximum Number of Points	Number of Points Reached
1	48	
2	18	
3	12	
4	35	
5	50	
Total	163	

Problem 1: Numerator Relationship Matrix, Inbreeding and Pedigree

Given is the pedigree shown in the Table below.

Geben ist das Pedigree in der nachfolgenden Tabelle

animal	sire	dam
4	2	1
5	2	1
6	2	3
7	2	3
8	2	3
9	4	5
10	6	5
11	9	10

```
## https://charlotte-ngs.github.io/lbgfs2023/data/exam_pedigree_p1.csv
```

a) Setup the numerator relationship matrix A for the above given pedigree

Stellen Sie die additive-genetische Verwandtschaftsmatrix A auf für das oben gegebene Pedigree

12

Solution

- Read pedigree from given path

```
s_p1_pedigree_path <- file.path(s_data_url_root, "exam_pedigree_p1.csv")
tbl_pedigree_p1 <- readr::read_delim(s_p1_pedigree_path, delim = ",")
```

- Augment pedigree with founder animals, start by determine vector of founder animals

```
vec_fnd_sire <- setdiff(tbl_pedigree_p1$sire, tbl_pedigree_p1$animal)
vec_fnd_dam <- setdiff(tbl_pedigree_p1$dam, tbl_pedigree_p1$animal)
vec_founder <- c(vec_fnd_sire, vec_fnd_dam)
vec_founder <- vec_founder[order(vec_founder)]
vec_founder
```

```
## [1] 1 2 3
```

- Add records for founder animals at the top

```
n_nr_fnd <- length(vec_founder)
tbl_pedigree_p1_aug <- dplyr::bind_rows(tibble::tibble(animal = vec_founder,
                                                       sire = rep(NA, n_nr_fnd),
                                                       dam = rep(NA, n_nr_fnd)),
                                        tbl_pedigree_p1)
tbl_pedigree_p1_aug
```

```
## # A tibble: 11 x 3
##    animal  sire   dam
##     <dbl> <dbl> <dbl>
##  1      1    NA    NA
##  2      2    NA    NA
##  3      3    NA    NA
##  4      4     2     1
##  5      5     2     1
##  6      6     2     3
##  7      7     2     3
##  8      8     2     3
##  9      9     4     5
## 10     10     6     5
## 11     11     9    10
```

- Use `pedigreemm::getA()` to compute numerator relationship matrix

```
ped <- pedigreemm::pedigree(sire = tbl_pedigree_p1_aug$sire,
                            dam = tbl_pedigree_p1_aug$dam,
                            label = tbl_pedigree_p1_aug$animal)
(mat_A <- as.matrix(pedigreemm::getA(ped = ped)))
```

```
##        1   2     3      4      5      6      7      8     9     10     11
## 1  1.000 0.0 0.000 0.5000 0.5000 0.0000 0.0000 0.0000 0.500 0.2500 0.3750
## 2  0.000 1.0 0.000 0.5000 0.5000 0.5000 0.5000 0.5000 0.500 0.5000 0.5000
## 3  0.000 0.0 1.000 0.0000 0.0000 0.5000 0.5000 0.5000 0.000 0.2500 0.1250
## 4  0.500 0.5 0.000 1.0000 0.5000 0.2500 0.2500 0.2500 0.750 0.3750 0.5625
## 5  0.500 0.5 0.000 0.5000 1.0000 0.2500 0.2500 0.2500 0.750 0.6250 0.6875
## 6  0.000 0.5 0.500 0.2500 0.2500 1.0000 0.5000 0.5000 0.250 0.6250 0.4375
## 7  0.000 0.5 0.500 0.2500 0.2500 0.5000 1.0000 0.5000 0.250 0.3750 0.3125
## 8  0.000 0.5 0.500 0.2500 0.2500 0.5000 0.5000 1.0000 0.250 0.3750 0.3125
## 9  0.500 0.5 0.000 0.7500 0.7500 0.2500 0.2500 0.2500 1.250 0.5000 0.8750
## 10 0.250 0.5 0.250 0.3750 0.6250 0.6250 0.3750 0.3750 0.500 1.1250 0.8125
## 11 0.375 0.5 0.125 0.5625 0.6875 0.4375 0.3125 0.3125 0.875 0.8125 1.2500
```

b) Specify which of the animals given in the above pedigree is inbred and compute the inbreeding coefficient F for each animal. Please fill out the table below.

Geben Sie an, welches der Tiere im oben gegebenen Pedigree ingezüchtet ist und berechnen Sie für jedes Tier den Inzuchtkoeffizienten F. Füllen Sie dazu die unten angegebenen Tabelle aus.

22

Animal	Inbred (Y/N)	Inbreeding Coefficient
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

Solution

Animal	Inbred (Y/N)	Inbreeding Coefficient
1	N	0.000
2	N	0.000
3	N	0.000
4	N	0.000
5	N	0.000
6	N	0.000
7	N	0.000
8	N	0.000
9	Y	0.250
10	Y	0.125
11	Y	0.250

c) Construct the pedigree from the numerator relationship matrix A shown below.

Erstellen Sie das Pedigree basierend auf der unten gegebenen additive-genetischen Verwandtschaftsmatrix A

14

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

```
## https://charlotte-ngs.github.io/lbgfs2023/data/exam_num_rel_mat_p1.csv
```

Solution

- Read the matrix from the csv-file first as tibble, then convert it to a matrix

```
s_p1_num_rel_mat_path <- file.path(s_data_url_root, "exam_num_rel_mat_p1.csv")
tbl_mat_A <- readr::read_delim(s_p1_num_rel_mat_path, delim = ",")
mat_mat_A <- as.matrix(tbl_mat_A)
```

- Compute cholesky decomposition of A

```
mat_R <- chol(mat_mat_A)
mat_R
```

```
##      1 2 3 4         5         6         7
## [1,] 1 0 0 0 0.0000000 0.5000000 0.0000000
## [2,] 0 1 0 0 0.5000000 0.0000000 0.2500000
## [3,] 0 0 1 0 0.5000000 0.5000000 0.2500000
## [4,] 0 0 0 1 0.0000000 0.0000000 0.5000000
## [5,] 0 0 0 0 0.7071068 0.0000000 0.3535534
## [6,] 0 0 0 0 0.0000000 0.7071068 0.0000000
## [7,] 0 0 0 0 0.0000000 0.0000000 0.7071068
```

- Compute matrix L which is $R \cdot S^{-1}$, but with a diagonal of all ones

```
mat_S_inv <- diag(1/diag(mat_R))
mat_L <- crossprod(mat_R, mat_S_inv)
mat_L
```

```
##   [,1] [,2] [,3] [,4] [,5] [,6] [,7]
## 1  1.0 0.00 0.00  0.0  0.0    0    0
## 2  0.0 1.00 0.00  0.0  0.0    0    0
## 3  0.0 0.00 1.00  0.0  0.0    0    0
## 4  0.0 0.00 0.00  1.0  0.0    0    0
## 5  0.0 0.50 0.50  0.0  1.0    0    0
## 6  0.5 0.00 0.50  0.0  0.0    1    0
## 7  0.0 0.25 0.25  0.5  0.5    0    1
```

- From the inverse L^{-1}, we get the matrix $P = I - L^{-1}$ which specifies the offspring parent relations

```
mat_L_inv <- solve(mat_L)
mat_P <- diag(nrow = nrow(mat_L)) - mat_L_inv
mat_P
```

```
##        1            2            3   4   5 6 7
## [1,] 0.0 0.000000e+00 0.000000e+00 0.0 0.0 0 0
## [2,] 0.0 0.000000e+00 0.000000e+00 0.0 0.0 0 0
## [3,] 0.0 0.000000e+00 0.000000e+00 0.0 0.0 0 0
## [4,] 0.0 0.000000e+00 0.000000e+00 0.0 0.0 0 0
## [5,] 0.0 5.000000e-01 5.000000e-01 0.0 0.0 0 0
## [6,] 0.5 0.000000e+00 5.000000e-01 0.0 0.0 0 0
## [7,] 0.0 5.551115e-17 5.551115e-17 0.5 0.5 0 0
```

- Extract elements relevant elements

```
mat_rel_elem <- which(mat_P > sqrt(.Machine$double.eps), arr.ind = TRUE)
```

```
anim_with_par <- unique(mat_rel_elem[, "row"])
anim_with_par <- anim_with_par[order(anim_with_par)]
anim_with_par
```

```
## [1] 5 6 7
```

The parents can be extracted from

```
for (i in anim_with_par){
  cat("Animal: ", i, " has parents: ", paste(mat_rel_elem[mat_rel_elem[,"row"] == i,"col"],
                                             sep = " and "), "\n")
}
```

```
## Animal:  5  has parents:  2 3
## Animal:  6  has parents:  1 3
## Animal:  7  has parents:  4 5
```

All other animals are founders.

Problem 2: Quantitative Genetics

The following dataset shows the influence of a single locus on the pigmentation of a number of animals.

Der unten gezeigte Datensatz zeigt den Einfluss eines einzelnen Locus auf die Pigemntierung von Tieren.

Animal	LocusC	Pigmentation
1	1	58
2	1	54
3	0	24
4	1	54
5	1	61
6	1	52
7	1	60
8	0	24
9	1	57
10	0	21
11	0	24
12	0	28
13	1	50
14	1	56
15	0	21
16	1	56
17	1	57
18	0	25
19	2	83
20	2	85
21	1	55
22	1	62
23	2	85
24	2	83
25	1	58
26	1	59
27	0	27
28	0	23
29	1	58
30	2	79
31	2	78
32	0	24

```
## https://charlotte-ngs.github.io/lbgfs2023/data/exam_qg_single_locus_p2.csv
```

a) Compute the allele frequencies (p and q) and the genotypic values (a and d) from the above shown dataset assuming that the number in column ‘LocusC‘ counts the number of alleles with a positive effect on pigmentation.

Berechnen Sie die Allelfrequenzen (p und q) und die genotypischen Werten (a und d) aufgrund des oben gezeigten Datensatzes. Die Zahlen in der Kolonne mit der Überschrift ‘LocusC‘ entspricht der Anzahl Allele mit positiver Wirkung auf die Pigmentierung

10

Solution

- Read the data

```
s_data_p2 <- file.path(s_data_url_root, "exam_qg_single_locus_p2.csv")
tbl_p2 <- readr::read_delim(s_data_p2, delim = ",")
```

The minor allele frequency p is computed as

```
n_maf_est <- mean(tbl_p2$LocusC)/2
```

Hence $p = 0.4375$ and $q = 1 - p = 0.5625$.

The genotypic value a is computed from fitting a linear regression to all homozygous animals. Hence, we first have to filter the dataset for the homozygous animals.

```
library(dplyr)
tbl_p2_homo <- tbl_p2 %>%
  filter(LocusC != 1)
```

Then a linear regression is fitted

```
lm_pig_geno_val_a <- lm(Pigmentation ~ LocusC, data = tbl_p2_homo)
(smry_pig_geno_val_a <- summary(lm_pig_geno_val_a))
```

```
##
## Call:
## lm(formula = Pigmentation ~ LocusC, data = tbl_p2_homo)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -4.167 -1.600 -0.100  1.383  3.900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  24.1000     0.8006   30.10 3.98e-14 ***
## LocusC       29.0333     0.6537   44.41  < 2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 2.532 on 14 degrees of freedom
## Multiple R-squared:  0.993,  Adjusted R-squared:  0.9924
## F-statistic:  1973 on 1 and 14 DF,  p-value: < 2.2e-16
```

The genotypic value a is obtained from the slope

```
n_geno_val_a_est <- smry_pig_geno_val_a$coefficients["LocusC","Estimate"]
```

Then $a = 29.0333333$

To get to the value d, we have to subtract from the mean of the homozygous animals the intercept and a.

```
n_intercept_est <- smry_pig_geno_val_a$coefficients["(Intercept)","Estimate"]
n_mean_het <- mean(tbl_p2$Pigmentation[tbl_p2$LocusC == 1])
n_geno_val_d_est <- n_mean_het - n_intercept_est - n_geno_val_a_est
```

Hence $d = 3.5541667$

b) Compute the breeding values and the dominance deviations for the genotypes of ‘LocusC‘ based on the above given dataset.

Berechnen Sie die Zuchtwerte und die Dominanzabweichungen für die Genotypen von ‘LocusC‘ basierend auf den oben gegebenen Daten.

6

Solution

- Breeding values are computed as

Genotype	Breeding Value
0	$-2p\alpha$
1	$(q-p)\alpha$
2	$2q\alpha$

The components for the breeding values are taken from the solution of Problem a)

```
n_alpha <- n_geno_val_a_est + (1-2*n_maf_est) * n_geno_val_d_est
vec_bv <- c(-2*n_maf_est * n_alpha,
            (1-2*n_maf_est) * n_alpha,
            2*(1-n_maf_est) * n_alpha)
n_nr_genotype <- length(vec_bv)
vec_bv_animal <- rep(NA, nrow(tbl_p2))
for (idx in 1:n_nr_genotype){
  vec_bv_animal[tbl_p2$LocusC == (idx-1)] <- vec_bv[idx]
}
vec_bv_animal
```

```
##  [1]   3.684701   3.684701 -25.792904   3.684701   3.684701   3.684701
##  [7]   3.684701 -25.792904   3.684701 -25.792904 -25.792904 -25.792904
## [13]   3.684701   3.684701 -25.792904   3.684701   3.684701 -25.792904
## [19]  33.162305  33.162305   3.684701   3.684701  33.162305  33.162305
## [25]   3.684701   3.684701 -25.792904 -25.792904   3.684701  33.162305
## [31]  33.162305 -25.792904
```

- Dominance deviations are given by

Genotype	Dominance Deviation
0	$-2p^2d$
1	$2pqd$
2	$-2q^2d$

The computation for each animal leads to

```
vec_dom_dev <- c(-2*n_maf_est^2*n_geno_val_d_est,
                 2*n_maf_est*(1-n_maf_est)*n_geno_val_d_est,
                 -2*(1-n_maf_est)^2*n_geno_val_d_est)
n_nr_genotype <- length(vec_dom_dev)
vec_dom_dev_animal <- rep(NA, nrow(tbl_p2))
for (idx in 1:n_nr_genotype){
  vec_dom_dev_animal[tbl_p2$LocusC == (idx-1)] <- vec_dom_dev[idx]
}
vec_dom_dev_animal
```

```
##  [1]  1.749316  1.749316 -1.360579  1.749316  1.749316  1.749316  1.749316
##  [8] -1.360579  1.749316 -1.360579 -1.360579 -1.360579  1.749316  1.749316
## [15] -1.360579  1.749316  1.749316 -1.360579 -2.249121 -2.249121  1.749316
## [22]  1.749316 -2.249121 -2.249121  1.749316  1.749316 -1.360579 -1.360579
## [29]  1.749316 -2.249121 -2.249121 -1.360579
```

Summary table of results

```
tbl_result_p2 <- tibble::tibble(Animal = tbl_p2$Animal,
                                `Breeding Value` = vec_bv_animal,
                                `Dominance Deviation` = vec_dom_dev_animal)
knitr::kable(tbl_result_p2, booktabs = TRUE, longtable = TRUE)
```

Animal	Breeding Value	Dominance Deviation
1	3.6847	1.749316
2	3.6847	1.749316
3	-25.7929	-1.360579
4	3.6847	1.749316
5	3.6847	1.749316
6	3.6847	1.749316
7	3.6847	1.749316
8	-25.7929	-1.360579
9	3.6847	1.749316
10	-25.7929	-1.360579
11	-25.7929	-1.360579
12	-25.7929	-1.360579
13	3.6847	1.749316
14	3.6847	1.749316
15	-25.7929	-1.360579
16	3.6847	1.749316
17	3.6847	1.749316
18	-25.7929	-1.360579
19	33.1623	-2.249121
20	33.1623	-2.249121
21	3.6847	1.749316
22	3.6847	1.749316
23	33.1623	-2.249121
24	33.1623	-2.249121
25	3.6847	1.749316
26	3.6847	1.749316
27	-25.7929	-1.360579

Animal	Breeding Value	Dominance Deviation
28	-25.7929	-1.360579
29	3.6847	1.749316
30	33.1623	-2.249121
31	33.1623	-2.249121
32	-25.7929	-1.360579

c) Compute the additive-genetic variance, the dominance variance and the total genetic variance for the data given above.

   Berechnen Sie die additive-genetische Varianz, die Dominanz-Varianz und die totale genetische Varianz für die oben angegeben Daten.

2

Solution

- The additive genetic variance (σ_A^2) is given by

$$\sigma_A^2 = 2pq\alpha^2$$

```
n_sigma_a2 <- 2 * n_maf_est * (1-n_maf_est) * n_alpha^2
cat(" * Genetic additive variance: ", n_sigma_a2, "\n")
```

```
##  * Genetic additive variance:  427.6761
```

- Dominance variance

$$\sigma_D^2 = (2pqd)^2$$

```
n_sigma_d2 <- (2*n_maf_est*(1-n_maf_est)*n_geno_val_d_est)^2
cat(" * Dominance variance: ", n_sigma_d2, "\n")
```

```
##  * Dominance variance:  3.060108
```

Problem 3: Variance and Inbreeding

Starting in the year 1953 the cattle breed Simmental was divided into 4 subpopulations of equal size. Since the separations the subpopulations stayed isolated without any exchange of genetic material. Before the separation the number of dams in the Simmental breed was 4×10^4. The generation interval can be assumed to be 5 years. The males in the populations can be ignored in this problem.

Seit 1953 wurde die Rinderrasse Simmental in 4 gleich grosse Subpopulationen unterteilt. Seit der Aufteilung gab es keinen Austausch von genetischem Material zwischen den Subpopulationen. Vor der Aufteilung umfasste die Rasse Simmental 4×10^4 Kühe. Das Generationeninterval wird mit 5 Jahren angegeben. Die Stiere in der Population werden in dieser Aufgabe nicht berücksichtigt.

a) Compute the inbreeding coefficient F_t in the subpopulations that can be expected today.

Berechnen Sie den Inzuchtkoeffizienten F_t in den Subpopulationen, welchen wir heute erwarten können.

4

Solution

The inbreeding coefficient F_t can be computed as

$$F_t = 1 - (1 - \Delta F)^t$$

where t is the number of generations and $\Delta F = \frac{1}{2N}$

```
n_nr_gen <- (n_export_year-n_exile_year) / n_gen_int
Delta_F <- 1/(2*n_nr_dam_per_sub_pop)
Ft <- 1 - (1 - Delta_F)^n_nr_gen
```

Inserting this

$$F_t = 1 - (1 - \Delta F)^t = 1 - (1 - 5 \times 10^{-5})^{14} = 6.9977255 \times 10^{-4}$$

b) Breeders try to avoid inbreeding because they want to prevent inbreeding depression. For a quantitative trait influenced by a single locus inbreeding depression is to be computed. The allele frequency is $p = 0.325$ and the genotypic values are $a = 50$ and $d = 35$. The inbreeding coefficient can be taken from the result of problem a. (In case problem a) was not solved, an inbreeding coefficient $F_t = 0.001$ can be assumed.)

ZüchterInnen sind bestrebt Inzucht zu begrenzen, da Inzuchtdepression vermieden werden soll. Für ein quantitatives Merkmal, welches von einem Locus beeinflusst wird, soll die Inzuchtdepression berechnet werden. Die Allelfrequenz beträgt $p = 0.325$ und die genotypischen Werte sind $a = 50$ und $d = 35$. Der Inzuchtkoeffizient kann aus der Lösung von Aufgabe a übernommen werden. (Falls Aufgabe a) nicht gelöst wurde, dann kann ein Inzuchtkoeffizient von $F_t = 0.001$ angenommen werden.)

4

Solution

```
DeltaM <- n_two_pqd * Ft
```

Inbreeding depression is computed as

$$\Delta M = M_F - M_0 = 2pqdF = 0.0107459$$

With assumed value of inbreeding of 0.001

```
DeltaM_assumed <- n_two_pqd * Ft_assumed
```

$$\Delta M = M_F - M_0 = 2pqdF = 0.0153563$$

c) How many generations does it take until the inbreeding depression as computed under Problem b) is larger than 0.1?

Wie viele Generationen wird es dauern bis die Inzuchtdepression, so wie sie in Aufgabe b) berechnet wurde grösser ist als 0.1?

4

Solution

In breeding coefficient such that inbreeding depression reaches 0.1

```
F_target <- DeltaM_target/n_two_pqd
```

$$F = \frac{\Delta M}{2pqd} = \frac{0.1}{2 * 0.325 * 0.675 * 35} = 0.006512$$

The number of generations (t) it takes to reach a certain level of inbreeding

$$t = \frac{log(1 - F)}{log(1 - \Delta F)} = \frac{log(1 - (\Delta M / 2pqd))}{log(1 - \Delta F)}$$

```
n_nr_gen <- (log(1 - (DeltaM_target/n_two_pqd))) / (log(1 - Delta_F))
cat(" ==> it takes ", ceiling(n_nr_gen),
    " generations until inbreeding depression is larger than ",
    DeltaM_target, "\n")
```

```
##  ==> it takes  131  generations until inbreeding depression is larger than  0.1
```

Problem 4: BLUP

The following dataset is to be used to predict breeding values for the response variable `weight`. Use a linear mixed model with `sex` and `Breast Circumference` (BC) as fixed effects. The heritability (h^2) for weight is assumed to be $h^2 = 0.36$. The phenotypic variance can directly be computed based on the given observations in the dataset.

Der folgende Datensatz soll für die Schätzung von Zuchtwerten für das Merkmal 'weight' verwendet werden. Verwenden Sie ein gemischtes lineares Modell mit 'sex' und 'Brustumfang' (BC) als fixe Effekte. Die Erblichkeit (h^2) für das Merkmal 'weight' wird mit $h^2 = 0.36$ angenommen. Die phänotypische Varianz kann aus den beobachteten Daten berechnet werden.

id	sire	dam	sex	BC	weight
5	1	2	1	152	404
6	3	4	2	150	367
7	1	4	2	149	358
8	5	6	1	152	410
9	3	2	2	151	360
10	8	7	1	154	416
11	8	9	2	150	371
12	10	11	2	154	396
13	10	9	2	151	365
14	8	11	2	151	360
15	5	7	2	153	378
16	8	6	2	151	369

```
## https://charlotte-ngs.github.io/lbgfs2023/data/exam_blup_p4.csv
```

a) Predict breeding values for the response variable ‘weight‘ using a BLUP animal model. Specify the model with a formula using matrix-vector notation. Explain all the symbols used and insert the information from the dataset to the known model components. For all random effects in the model, specify the expected values and the variance-covariance matrices.

Schätzen Sie Zuchtwerte für die Zielgrösse ‘weight‘ mit einem BLUP-Tiermodell. Geben Sie das Modell mit einer Formel in Matrix-Vektor-Schreibweise an. Erläutern Sie die verwendeten Symbole im Modell und setzen Sie die Informationen aus dem Datensatz in die bekannten Modellkomponenten ein. Für all zufälligen Effekte im Modell, geben Sie bitte die Erwartungswerte und die Varianz-Kovarianz-Matrizen an.

25

Solution

The data is read using

```
s_data_url_p4 <- file.path(s_data_url_root, "exam_blup_p4.csv")
s_delim_char <- ","
tbl_p4 <- readr::read_delim(s_data_url_p4, delim = s_delim_char)
```

The model is

$$y = Xb + Zu + e$$

- with the vectors
  - y: of length n with observations
  - b: fixed effects of `sex` - differnces, slope and intercept for `BC`
  - u: of length q with random breeding values
  - e: of length n random residual effects
- where n is the number of observations in the dataset and q the number of animals in the pedigree.
- with design Matrices X and Z linking fixed effects and breeding values to observations, respectively
- Expected values of random components are given as

$$E \left[\begin{array}{c} y \\ u \\ e \end{array}\right] = \left[\begin{array}{c} Xb \\ 0 \\ 0 \end{array}\right]$$

- Variance-Covariance matrices

$$var \left[\begin{array}{c} y \\ u \\ e \end{array}\right] = \left[\begin{array}{ccc} V & ZG & R \\ GZ^T & G & 0 \\ R & 0 & R \end{array}\right]$$

- with $var(u) = G = A\sigma_u^2$, where A is the numerator relationship matrix of the animals in the pedigree and σ_u^2 is the additive-genetic variance,

- with $var(e) = R = I\sigma_e^2$ where I is the identity matrix and σ_e^2 is the residual variance
- with $var(y) = V = ZGZ^T + R$
- Known components in the model are

```
# design matrix X
matX <- model.matrix(lm(weight ~ BC + as.factor(sex), data = tbl_p4))
attr(matX,"assign") <- NULL
attr(matX,"contrasts") <- NULL
colnames(matX) <- NULL
# design matrix Z
vec_fnds <- c(setdiff(tbl_p4$sire, tbl_p4$id), setdiff(tbl_p4$dam, tbl_p4$id))
vec_fnds <- vec_fnds[order(vec_fnds)]
n_nr_fnds <- length(vec_fnds)
n_nr_obs <- nrow(tbl_p4)
n_nr_ani <- n_nr_fnds + n_nr_obs
matZ <- cbind(matrix(0, nrow = n_nr_obs, ncol = n_nr_fnds),
              diag(nrow = n_nr_obs))
```

$$y = \begin{bmatrix} 404 \\ 367 \\ 358 \\ 410 \\ 360 \\ 416 \\ 371 \\ 396 \\ 365 \\ 360 \\ 378 \\ 369 \end{bmatrix}, X = \begin{bmatrix} 1 & 152 & 0 \\ 1 & 150 & 1 \\ 1 & 149 & 1 \\ 1 & 152 & 0 \\ 1 & 151 & 1 \\ 1 & 154 & 0 \\ 1 & 150 & 1 \\ 1 & 154 & 1 \\ 1 & 151 & 1 \\ 1 & 151 & 1 \\ 1 & 153 & 1 \\ 1 & 151 & 1 \end{bmatrix}$$

Compute the inverse numerator relationship matrix

```
ped_p4 <- pedigreemm::pedigree(sire = c(rep(NA, n_nr_fnds), tbl_p4$sire),
                                dam  = c(rep(NA, n_nr_fnds), tbl_p4$dam),
                                label = 1:n_nr_ani)
matA_inv <- as.matrix(pedigreemm::getAInv(ped = ped_p4))
```

Setting up mixed model equations

```
lambda <- (1-n_h2_weight)/n_h2_weight
mat_xtx <- crossprod(matX)
mat_xtz <- crossprod(matX, matZ)
mat_ztx <- t(mat_xtz)
mat_ztz_lainv <- crossprod(matZ) + lambda * matA_inv
mat_coef <- rbind(cbind(mat_xtx, mat_xtz), cbind(mat_ztx,mat_ztz_lainv))
mat_rhs <- rbind(crossprod(matX, tbl_p4$weight),
                 crossprod(matZ, tbl_p4$weight))
mat_sol <- solve(mat_coef, mat_rhs)
mat_sol
```

```
##              [,1]
##    -504.41628785
##       5.98638818
##     -31.39119022
## 1    -0.01281456
## 2    -1.23431247
## 3     0.01281456
## 4     1.23431247
## 5    -0.71761106
## 6     1.77664300
## 7     0.69198194
## 8     1.25013143
## 9    -1.75101387
## 10    0.95055472
## 11    1.67036549
## 12    3.00755718
## 13   -0.97130007
## 14   -0.54229735
## 15   -0.47315349
## 16    1.37058905
```

From this we get:

```
cat(" * Intercept: ", mat_sol[1,1], "\n")
```

```
##  * Intercept:  -504.4163
```

```
cat(" * Slope: ", mat_sol[2,1], "\n")
```

```
##  * Slope:  5.986388
```

```
cat(" * Sex-diff: ", mat_sol[3,1], "\n" )
```

```
##  * Sex-diff:  -31.39119
```

```
cat(" * Breeding values:\n")
```

```
##  * Breeding values:
```

```
mat_sol[4:(nrow(mat_sol)),]
```

```
##           1           2           3           4           5           6
## -0.01281456 -1.23431247  0.01281456  1.23431247 -0.71761106  1.77664300
##           7           8           9          10          11          12
##  0.69198194  1.25013143 -1.75101387  0.95055472  1.67036549  3.00755718
##          13          14          15          16
## -0.97130007 -0.54229735 -0.47315349  1.37058905
```

b) Compute the reliabilities ($B\%$) for the predicted breeding values from problem a).

Berechnen Sie die Bestimmtheitsmasse ($B\%$) für die unter Aufgabe a) geschätzten Zuchtwerte.

10

Solution

Reliabilities are computed from the inverse of the general form of the mixed-models coefficients matrix. The general form is obtained by dividing the coefficient matrix by σ_e^2. This has to be computed first based on the phenotypic variance σ_p^2 which is obtained from the observations.

```
# determine variance components
sigma_p <- var(tbl_p4$weight)
sigma_u <- n_h2_weight * sigma_p
sigma_e <- sigma_p - sigma_u
# inverse of general form of coefficient matrix
mat_coef_gen_inv <- solve(mat_coef / sigma_e)
# extract lower right corner of coefficient matrix
mat_C22 <- mat_coef_gen_inv[(ncol(matX)+1):nrow(mat_coef_gen_inv),
                            (ncol(matX)+1):ncol(mat_coef_gen_inv)]
# compute inbreeding coefficient
vec_inb <- pedigreemm::inbreeding(ped = ped_p4)
# get reliability B
vec_B <- 1 - diag(mat_C22) / ((1+vec_inb) * sigma_u)
cat(" * Reliabilities in % are: \n")
```

```
##  * Reliabilities in % are:
```

```
100*vec_B
```

```
##         1         2         3         4         5         6         7         8 
##  9.251623  9.384698  9.251623  9.384698 15.245739 23.020094 24.721334 15.532517 
##         9        10        11        12        13        14        15        16 
## 23.271058 14.080520 23.059421 13.562660 17.092133 24.970633 20.370260 24.933033
```

Problem 5: Genomic Breeding Values

The following dataset is given to predict genomic breeding values. The column with the header `BC` contains the covariable `Breast Circumference` in cm. The response variable (y) is in the column with the header `weight`.

Der folgende Datensatz wird verwendet um genomische Zuchtwerte zu schätzen. Die Kolonne mit der Überschrift enthält die Kovariable ‘Brustumfang‘. Die Zielgrösse (y) befindet sich in der Kolonne mit der Überschrift ‘weight‘.

id	sire	dam	sex	BC	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15	weight
5	1	2	1	152	0	0	2	0	2	2	0	1	1	1	2	2	2	1	2	443
6	3	4	2	150	1	1	1	2	2	1	0	2	0	1	0	1	1	1	1	377
7	1	4	2	149	0	0	2	1	2	2	1	0	0	2	2	1	2	1	2	423
8	5	6	1	152	0	0	1	1	2	2	0	1	1	2	1	1	1	1	1	402
9	3	2	2	151	0	2	1	2	2	2	1	1	0	1	1	1	2	1	1	431
10	8	7	1	154	0	0	2	0	2	2	0	0	1	2	1	0	2	1	1	398
11	8	9	2	150	0	1	1	1	2	2	1	1	0	2	1	0	1	1	0	350
12	10	11	2	154	0	0	2	0	2	2	0	0	1	2	1	0	1	1	1	361
13	10	9	2	151	0	1	1	1	2	2	1	1	0	1	2	1	2	2	0	401
14	8	11	2	151	0	1	1	2	2	2	0	2	1	2	1	1	2	2	1	453
15	5	7	2	153	0	0	2	0	2	2	1	1	1	1	2	1	2	1	2	445
16	8	6	2	151	0	0	0	1	2	2	0	1	0	2	0	2	0	0	2	336

```
## https://charlotte-ngs.github.io/lbgfs2023/data/exam_geno_p5.csv
```

a) Use a marker-effects model to predict genomic breeding values for the response variable 'weight'. Use 'BC' and 'sex' as fixed effects. The columns 'S1'-'S15' contain marker genotypes to be used in the prediction of genomic breeding values. The ratio $\lambda_q = \sigma_e^2/\sigma_q^2$ is assumed to be $\lambda_q = 3$.

   Verwenden Sie ein Marker-Effekt Modell zur Schätzung von genomischen Zuchtwerten für die Zielgrösse 'weight'. Verwenden Sie 'BC' und 'sex' als fixe Effekte. Die Kolonnen 'S1'-'S15' enthalten Markergenotypen, welche für die genomische Zuchtwertschätzung verwendet werden sollen. Das Verhältnis $\lambda_q = \sigma_e^2/\sigma_q^2$ soll den Wert $\lambda_q = 3$ haben.

25

Solution

The marker effect model is given by

$$y = Xb + Wq + e$$

- with the vectors
  - y: of length n with observations
  - b: fixed effects of `sex` - differnces, slope and intercept for `BC`
  - q: of length l with random marker effects
  - e: of length n random residual effects
- where n is the number of observations in the dataset and l the number of marker loci.
- with design Matrices X and W linking fixed effects and marker effects to observations, respectively
- Expected values of random components are given as

$$E \left[\begin{array}{c} y \\ q \\ e \end{array}\right] = \left[\begin{array}{c} Xb \\ 0 \\ 0 \end{array}\right]$$

- Variance-Covariance matrices

$$var \left[\begin{array}{c} y \\ q \\ e \end{array}\right] = \left[\begin{array}{ccc} V & ZQ & R \\ QZ^T & Q & 0 \\ R & 0 & R \end{array}\right]$$

- with $var(q) = Q = I\sigma_q^2$, where I is the identity matrix and σ_q^2 is the marker-effects variance,
- with $var(e) = R = I\sigma_e^2$ where I is the identity matrix and σ_e^2 is the residual variance
- with $var(y) = V = ZQZ^T + R$
- Known components in the model are

```r
# design matrix X
matX <- model.matrix(lm(weight ~ BC + as.factor(sex), data = tbl_p5))
attr(matX,"assign") <- NULL
attr(matX,"contrasts") <- NULL
colnames(matX) <- NULL
matX
```

```
##    [,1] [,2] [,3]
## 1     1  152    0
## 2     1  150    1
## 3     1  149    1
## 4     1  152    0
## 5     1  151    1
## 6     1  154    0
## 7     1  150    1
## 8     1  154    1
## 9     1  151    1
## 10    1  151    1
## 11    1  153    1
## 12    1  151    1
```

Genotype matrix

```
# matrix W is the genotype matrix in -1/0/1 coding
mat_geno <- as.matrix(dplyr::select(tbl_p5, dplyr::starts_with("S", ignore.case = FALSE)))
matW <- mat_geno - 1
matW
```

```
##       S1 S2 S3 S4 S5 S6 S7 S8 S9 S10 S11 S12 S13 S14 S15
##  [1,] -1 -1  1 -1  1  1 -1  0  0   0   1   1   1   0   1
##  [2,]  0  0  0  1  1  0 -1  1 -1   0  -1   0   0   0   0
##  [3,] -1 -1  1  0  1  1  0 -1 -1   1   1   0   1   0   1
##  [4,] -1 -1  0  0  1  1 -1  0  0   1   0   0   0   0   0
##  [5,] -1  1  0  1  1  1  0  0 -1   0   0   0   1   0   0
##  [6,] -1 -1  1 -1  1  1 -1 -1  0   1   0  -1   1   0   0
##  [7,] -1  0  0  0  1  1  0  0 -1   1   0  -1   0   0  -1
##  [8,] -1 -1  1 -1  1  1 -1 -1  0   1   0  -1   0   0   0
##  [9,] -1  0  0  0  1  1  0  0 -1   0   1   0   1   1  -1
## [10,] -1  0  0  1  1  1 -1  1  0   1   0   0   1   1   0
## [11,] -1 -1  1 -1  1  1  0  0  0   0   1   0   1   0   1
## [12,] -1 -1 -1  0  1  1 -1  0 -1   1  -1   1  -1  -1   1
```

Mixed model equations

```
mat_xtx <- crossprod(matX)
mat_xtw <- crossprod(matX,matW)
mat_wtx <- t(mat_xtw)
mat_wtw_linv <- crossprod(matW) + lambda_q * diag(nrow = ncol(matW))
mat_coef <- rbind(cbind(mat_xtx, mat_xtw),cbind(mat_wtx, mat_wtw_linv))
mat_rhs <- rbind(crossprod(matX, tbl_p5$weight),
                 crossprod(matW, tbl_p5$weight))
mat_sol <- solve(mat_coef, mat_rhs)
```

Solutions for marker effects are

```
(mat_mrk_sol <- mat_sol[(ncol(matX)+1):nrow(mat_sol),])
```

```
##            S1            S2            S3            S4            S5
## -2.333411e+00  4.694423e+00  6.702739e+00  9.027892e+00  2.135329e-13
##            S6            S7            S8            S9           S10 
##  2.333411e+00  2.993754e+00  7.367576e+00  8.790509e+00 -3.300825e+00 
##           S11           S12           S13           S14           S15 
##  1.035856e+01  5.393424e+00  1.812901e+01  8.037491e+00  1.146274e+01
```

The genomic breeding values are computed by multiplying the genotype matrix with the marker solutions

```
crossprod(t(matW), mat_mrk_sol)
```

```
##             [,1]
##  [1,]  39.997224
##  [2,]  -5.747350
##  [3,]  27.166536
##  [4,]  -6.322179
##  [5,]  27.727637
##  [6,]  -3.279323
##  [7,] -24.280678
##  [8,] -21.408333
##  [9,]  20.938626
## [10,]  40.934212
## [11,]  37.597554
## [12,] -41.484318
```

b) Use a breeding-value based genomic BLUP model to predict genomic breeding values for the response variable ‘weight‘. Use ‘BC‘ and ‘sex‘ as fixed effects. The columns ‘S1‘-‘S15‘ contain marker genotypes to be used in the prediction of genomic breeding values. The ratio $\lambda_g = \sigma_e^2/\sigma_g^2$ is assumed to be $\lambda_g = 5$.

Verwenden Sie ein Zuchtwert-basiertes genomisches BLUP Modell für die Schätzung von genomischen Zuchtwerten für die Zielgrösse ‘weight‘. Verwenden Sie ‘BC‘ und ‘sex‘ als fixe Effekte. Die Kolonnen ‘S1‘-‘S15‘ enthalten Markergenotypen, welche für die genomische Zuchtwertschätzung verwendet werden sollen. Das Verhältnis $\lambda_g = \sigma_e^2/\sigma_g^2$ soll den Wert $\lambda_g = 5$ haben.

25

Solution

The breeding value based model is

$$y = Xb + Zg + e$$

- with the vectors
  - y: of length n with observations
  - b: fixed effects of `sex` - differnces, slope and intercept for `BC`
  - g: of length q with random genomic breeding values
  - e: of length n random residual effects
- where n is the number of observations in the dataset and q the number of animals with genotypes.
- with design Matrices X and Z linking fixed effects and genomic breeding values to observations, respectively
- Expected values of random components are given as

$$E\left[\begin{array}{c} y \\ g \\ e \end{array}\right] = \left[\begin{array}{c} Xb \\ 0 \\ 0 \end{array}\right]$$

- Variance-Covariance matrices

$$var\left[\begin{array}{c} y \\ g \\ e \end{array}\right] = \left[\begin{array}{ccc} V & ZH & R \\ HZ^T & H & 0 \\ R & 0 & R \end{array}\right]$$

- with $var(g) = H = G\sigma_g^2$, where G is the genomic relationship matrix and σ_g^2 is the genomic variance,
- with $var(e) = R = I\sigma_e^2$ where I is the identity matrix and σ_e^2 is the residual variance
- with $var(y) = V = ZHZ^T + R$
- Known components in the model are

```r
# design matrix X
matX <- model.matrix(lm(weight ~ BC + as.factor(sex), data = tbl_p5))
attr(matX,"assign") <- NULL
attr(matX,"contrasts") <- NULL
colnames(matX) <- NULL
matX
```

```
##    [,1] [,2] [,3]
## 1     1  152    0
## 2     1  150    1
## 3     1  149    1
## 4     1  152    0
## 5     1  151    1
## 6     1  154    0
## 7     1  150    1
## 8     1  154    1
## 9     1  151    1
## 10    1  151    1
## 11    1  153    1
## 12    1  151    1
```

```
# design matrix Z
n_nr_obs <- nrow(tbl_p5)
matZ <- diag(nrow = n_nr_obs)
matZ
```

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

- Genomic relationship matrix

```
#' Compute genomic relationship matrix based on data matrix
computeMatGrm <- function(pmatData, pn_max_iter = 10, pn_min_eig_val = 0.0001) {
  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)
  matG_result <- tcrossprod(Z)/(2*sumpq)
  # check for positive definiteness
  n_min_eig_matG_result <- min(eigen(matG_result, only.values = TRUE)$values)
  n_iter_idx <- 0
  while (n_min_eig_matG_result < pn_min_eig_val & n_iter_idx < pn_max_iter){
    matG_result <- matG_result + 0.01 * diag(nrow = nrow(matG_result))
    n_min_eig_matG_result <- min(eigen(matG_result, only.values = TRUE)$values)
    n_iter_idx <- n_iter_idx + 1
  }
  # check for convergence
  if (n_iter_idx > pn_max_iter){
    stop(" *** ERROR: No convergence of bending genomic relationship matrix")
  }
  return(matG_result)
}
mat_geno <- as.matrix(dplyr::select(tbl_p5, dplyr::starts_with("S", ignore.case = FALSE)))
mat_grm_inv <- solve(computeMatGrm(pmatData = mat_geno))
```

- set up mixed model equations

```
mat_xtx <- crossprod(matX)
mat_xtz <- crossprod(matX, matZ)
mat_ztx <- t(mat_xtz)
mat_ztz_lgrm_inv <- crossprod(matZ) + lambda_g * mat_grm_inv
mat_coef <- rbind(cbind(mat_xtx,mat_xtz),
                  cbind(mat_ztx,mat_ztz_lgrm_inv))
mat_rhs <- rbind(crossprod(matX,tbl_p5$weight),
                 crossprod(matZ,tbl_p5$weight))
mat_sol <- solve(mat_coef,mat_rhs)
```

- Show the solutions for breeding values

```
cat(" * Genomic breeding values: \n")
```

```
##  * Genomic breeding values:
```

```
mat_sol[(ncol(matX)+1):nrow(mat_sol),]
```

```
##  [1]  12.744640  -6.781439   7.330412  -6.148771   6.150814  -2.442660
##  [7] -10.108413  -8.748461   7.984163   8.338193  12.224698 -20.543176
```