ETH Zurich
D-USYS
Institute of Agricultural Sciences

Solutions To Exam
Livestock Breeding and Genomics
FS 2022

Peter von Rohr

Date	*23. December 2022*
Begin	*09:15*
End	*11:15*

Name:

Legi-Nr:

Problem	Maximum Number of Points	Number of Points Reached
1	27	
2	25	
3	47	
4	28	
5	22	
Total	149	

Problem 1 Quantitative Genetics

The following dataset contains observations of a phenotypic trait and genotypes for two loci. The numbers in the columns SNP_1 and SNP_2 count the number of alleles with a positive effect on the observation. Genotypic values of heterozygous animals are assumed to be right between the genotypic values of homozygous animals ($d = 0$).

Der folgende Datensatz enthält Beobachtungen für ein phänotypisches Merkmal und Genotypen von zwei Loci. Die Zahlen in den Kolonnen SNP_1 *und* SNP_2 *zählen die Anzahl Allele mit einem positiven Effekt auf den Merkmalswert. Genotypische Werte der heterozygoten Tiere sind genau in der Mitte zwischen den genotypischen Werten der homozygoten Tiere* ($d = 0$).

Animal	SNP_1	SNP_2	Observation
1	0	2	141
2	0	1	120
3	2	1	189
4	1	2	172
5	1	1	158
6	1	1	152
7	1	0	141
8	0	1	116
9	1	2	176
10	0	0	107
11	1	0	132
12	1	2	173
13	1	0	131
14	0	2	144
15	1	2	176
16	1	2	175
17	0	0	103
18	1	1	154
19	2	0	176
20	0	1	129
21	2	1	187
22	1	2	181
23	1	0	138
24	1	0	132
25	0	1	119
26	0	0	94
27	1	0	139
28	0	2	144
29	0	1	124
30	0	1	124

The dataset as shown in the table above is available under:

Der oben gezeigte Datensatz ist verfügbar unter:

https://charlotte-ngs.github.io/lbgfs2022/data/lbgfs2022_exam_problem1.csv

a) Compute the genotypic values for all genotypes of the two loci `SNP_1` and `SNP_2` using the above dataset.

   Berechnen Sie die genotypischen Werte aller Genotypen für die zwei Loci `SNP_1` *und* `SNP_2` basierend auf dem oben gezeigtem Datensatz.

6

Solution:

The genotypic values of the homozygous genotypes can be computed based on a linear regression. This regression model can be formulated as follows.

```
lm_joint <- lm(Observation ~ SNP_1 + SNP_2, data = tbl_data_p01)
(smry_joint <- summary(lm_joint))
```

```
##
## Call:
## lm(formula = Observation ~ SNP_1 + SNP_2, data = tbl_data_p01)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9039 -2.7900 -0.1508  2.2395  7.1678
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  101.904      1.424   71.54   <2e-16 ***
## SNP_1         33.903      1.128   30.07   <2e-16 ***
## SNP_2         19.928      0.908   21.95   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.94 on 27 degrees of freedom
## Multiple R-squared:  0.9794, Adjusted R-squared:  0.9779
## F-statistic: 641.3 on 2 and 27 DF,  p-value: < 2.2e-16
```

The genotypic values can then be summarized in the following table

Genotype	SNP_1	SNP_2
0	-33.90299	-19.92832
1	0.00000	0.00000
2	33.90299	19.92832

■

b) Compute the breeding values and the dominance deviations for all animals in the above dataset. Allele frequencies can be determined based on the given dataset.

   Berechnen Sie die Zuchtwerte und die Dominanzabweichungen aller Tiere im oben gegebenen Datensatz. Die Allelefrequenzen sollen aufgrund des gegebenen Datensatzes bestimmt werden.

15

Solution:

The breeding values are computed as

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

where $\alpha = a + (q-p)d$ with $d = 0$, we get $\alpha = a$. The value for a corresponds to the absolute value of the genotypic values of the homozygous genotypes. The genotypic values are computed as shown in Problem 1a)

```
lm_joint <- lm(Observation ~ SNP_1 + SNP_2, data = tbl_data_p01)
smry_joint <- summary(lm_joint)
vec_genotypic_value <- c(smry_joint$coefficients["SNP_1", "Estimate"],
                         smry_joint$coefficients["SNP_2", "Estimate"])
```

The variables p and q represent the allele frequencies. The allele frequencies are determined using the following chunk

```
library(dplyr)
tbl_freq_allele <- tbl_data_p01 %>%
  summarise(freq_pos_allele_SNP1 = sum(SNP_1) / (2*nrow(tbl_data_p01)))
tbl_freq_allele <- dplyr::bind_cols(tbl_freq_allele,
                                    tbl_data_p01 %>%
  summarise(freq_pos_allele_SNP2 = sum(SNP_2) / (2*nrow(tbl_data_p01))))
tbl_freq_allele
```

```
## # A tibble: 1 x 2
##   freq_pos_allele_SNP1 freq_pos_allele_SNP2
##                  <dbl>                <dbl>
## 1                 0.35                0.483
```

```
vec_allele_freq <- c(tbl_freq_allele$freq_pos_allele_SNP1,
                     tbl_freq_allele$freq_pos_allele_SNP2)
```

The the following table gives an overview over the breeding values for every single-locus genotype.

Genotype	BVSNP1	BVSNP2
0	-23.73210	-19.2640472
1	10.17090	0.6642775
2	44.07389	20.5926022

The breeding values of all animals are

```
# snp1
tbl_breeding_value_bv1 <- tbl_breeding_value_results %>% select(Genotype, BVSNP1)
tbl_bv_snp1 <- tbl_data_p01 %>%
  inner_join(tbl_breeding_value_bv1, by = c("SNP_1" = "Genotype")) %>%
  select(Animal, BVSNP1)
# snp2
tbl_breeding_value_bv2 <- tbl_breeding_value_results %>% select(Genotype, BVSNP2)
tbl_bv_snp2 <- tbl_data_p01 %>%
  inner_join(tbl_breeding_value_bv2, by = c("SNP_2" = "Genotype")) %>%
  select(Animal, BVSNP2)
# total
tbl_bv_total <- bind_cols(tbl_bv_snp1, tbl_bv_snp2 %>% select(BVSNP2))
tbl_bv_total <- tbl_bv_total %>%
  mutate(BVTotal = BVSNP1 + BVSNP2)

# show the table
knitr::kable(tbl_bv_total, booktabs = TRUE, longtable = TRUE)
```

Animal	BVSNP1	BVSNP2	BVTotal
1	-23.73210	20.5926022	-3.139493
2	-23.73210	0.6642775	-23.067818
3	44.07389	0.6642775	44.738169
4	10.17090	20.5926022	30.763500
5	10.17090	0.6642775	10.835176
6	10.17090	0.6642775	10.835176
7	10.17090	-19.2640472	-9.093149
8	-23.73210	0.6642775	-23.067818
9	10.17090	20.5926022	30.763500
10	-23.73210	-19.2640472	-42.996143

Animal	BVSNP1	BVSNP2	BVTotal
11	10.17090	-19.2640472	-9.093149
12	10.17090	20.5926022	30.763500
13	10.17090	-19.2640472	-9.093149
14	-23.73210	20.5926022	-3.139493
15	10.17090	20.5926022	30.763500
16	10.17090	20.5926022	30.763500
17	-23.73210	-19.2640472	-42.996143
18	10.17090	0.6642775	10.835176
19	44.07389	-19.2640472	24.809845
20	-23.73210	0.6642775	-23.067818
21	44.07389	0.6642775	44.738169
22	10.17090	20.5926022	30.763500
23	10.17090	-19.2640472	-9.093149
24	10.17090	-19.2640472	-9.093149
25	-23.73210	0.6642775	-23.067818
26	-23.73210	-19.2640472	-42.996143
27	10.17090	-19.2640472	-9.093149
28	-23.73210	20.5926022	-3.139493
29	-23.73210	0.6642775	-23.067818
30	-23.73210	0.6642775	-23.067818

Dominance deviations are all 0, because $d = 0$ for both loci.

■

c) Compute the genetic additive variance σ_A^2 and the dominance variance σ_D^2 for the trait in the given dataset over both loci.

Berechnen Sie die genetisch-additive Varianz σ_A^2 *und die Dominanzvarianz* σ_D^2 für das Merkmal im Datensatz über beide Loci.

6

Solution:

The genetic additive variance σ_A^2 for a single locus is computed as

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

with $d = 0$, we get $\alpha = a$ and $\sigma_A^2 = 2pqa^2$. This is computed for both loci and then added together.

Based on the results, we got so far, we get

```
n_sigma_a2_snp1 <- 2 * vec_allele_freq[1] * (1-vec_allele_freq[1]) *
  vec_genotypic_value[1]^2
n_sigma_a2_snp2 <- 2 * (vec_allele_freq[2]) * (1-vec_allele_freq[2]) *
  vec_genotypic_value[2]^2
n_sigma_a2_tot <- n_sigma_a2_snp1 + n_sigma_a2_snp2
```

$$\sigma_{A,SNP1}^2 = 2 * 0.35 * (1 - 0.35) * 33.9029937^2 = 522.9829057$$

$$\sigma_{A,SNP2}^2 = 2 * 0.4833333 * (1 - 0.4833333) * 19.9283247^2 = 198.3484303$$

$$\sigma_A^2 = \sigma_{A,SNP1}^2 + \sigma_{A,SNP2}^2 = 522.9829057 + 198.3484303 = 721.3313361$$

The dominance variance σ_D^2 is 0 for both loci, because $d = 0$.

■

Problem 2 Relationship and Inbreeding

Starting on 01.01.2023, the Swiss Federal Office of Agriculture will start to pay subsidies to owners of livestock animals of endangered breeds. The amount of subsidies will be based on an index which among other components is based on the average inbreeding coefficient of animals and on the average age of the owners of the animals. Below we are given two example pedigrees of two different breeds.

Ab 01.01.2023 wird das Bundesamt für Landwirtschaft Subventionen an Halter von Nutztieren seltener Rassen auszahlen. Die Höhe der Subventionen wird auf einem Index basieren. Dieser Index wird neben anderen Komponenten auch den durchschnittlichen Inzuchtkoeffizient der Tiere und das mittlere Alter der Halter beinhalten. Unten sind zwei Beispielpedigrees von zwei verschiedenen Rassen gegeben.

Breed 1:

```
## https://charlotte-ngs.github.io/lbgfs2022/data/lbgfs2022_exam_problem2_pedigree1.csv
```

Animal	Sire	Dam	AgeOwner
CH114000313	CH114000329	CH114000277	78
CH114000368	CH114000329	CH114000277	71
CH114000285	CH114000313	CH114000255	72
CH114000256	CH114000368	CH114000248	77
CH114000397	CH114000285	CH114000256	77

Breed 2:

```
## https://charlotte-ngs.github.io/lbgfs2022/data/lbgfs2022_exam_problem2_pedigree2.csv
```

Animal	Sire	Dam	AgeOwner
CH115000392	CH115000277	CH115000301	51
CH115000319	CH115000277	CH115000256	53
CH115000346	CH115000277	CH115000256	52
CH115000347	CH115000392	CH115000319	54

a) Setup the numerator relationship matrices for both breeds given above.

Stellen Sie die additiv genetischen Verwandtschaftsmatrizen für beide oben gegebenen Rassen auf.

6

Solution:

Breed 1

Reading the data

```
tbl_ped1 <- readr::read_delim(file = s_ped1_path,
                              delim = ",",
                              col_types = readr::cols(
                                Animal = readr::col_character(),
                                Sire = readr::col_character(),
                                Dam = readr::col_character(),
                                AgeOwner = readr::col_double()
                              ))
```

Adding founder animals to the pedigree

```
vec_sire_founder <- unique(tbl_ped1$Sire)
vec_sire_founder <- vec_sire_founder[!is.element(vec_sire_founder, tbl_ped1$Animal)]
vec_dam_founder <- unique(tbl_ped1$Dam)
vec_dam_founder <- vec_dam_founder[!is.element(vec_dam_founder, tbl_ped1$Animal)]
vec_founder <- c(vec_sire_founder, vec_dam_founder)
n_nr_founder <- length(vec_founder)
tbl_ped1_founder <- tibble::tibble(Animal = vec_founder,
                                   Sire = rep(NA, n_nr_founder),
                                   Dam = rep(NA, n_nr_founder),
                                   AgeOwner = rep(NA, n_nr_founder))
tbl_ped1_aug <- dplyr::bind_rows(tbl_ped1_founder, tbl_ped1)
```

Remapping the IDs to integer values

```
tbl_ped_ID_map <- tibble::tibble(DomainCharId = tbl_ped1_aug$Animal,
                                 ImageIntId = c(1:length(tbl_ped1_aug$Animal)))
library(dplyr)
# Animals
tbl_ped1_int_id <- tbl_ped1_aug %>%
  left_join(tbl_ped_ID_map, by = c("Animal" = "DomainCharId")) %>%
  rename(AnimalIntId = ImageIntId)
# Sire
tbl_ped1_int_id <- tbl_ped1_int_id %>%
  left_join(tbl_ped_ID_map, by = c("Sire" = "DomainCharId")) %>%
  rename(SireIntId = ImageIntId)
# Dam
tbl_ped1_int_id <- tbl_ped1_int_id %>%
  left_join(tbl_ped_ID_map, by = c("Dam" = "DomainCharId")) %>%
  rename(DamIntId = ImageIntId)
```

Get the numerator relationship matrix with `pedigreemm`

```
ped_ped1 <- pedigreemm::pedigree(sire = tbl_ped1_int_id$SireIntId,
                                 dam =  tbl_ped1_int_id$DamIntId,
                                 label = as.character(tbl_ped1_int_id$AnimalIntId))
mat_A_ped1 <- as.matrix(pedigreemm::getA(ped = ped_ped1))
```

The numerator relationship matrix A_1 for the pedigree of breed 1 is

```
##      1    2    3    4     5     6      7      8      9
## 1 1.00 0.00 0.00 0.00 0.500 0.500 0.2500 0.2500 0.2500
## 2 0.00 1.00 0.00 0.00 0.500 0.500 0.2500 0.2500 0.2500
## 3 0.00 0.00 1.00 0.00 0.000 0.000 0.5000 0.0000 0.2500
## 4 0.00 0.00 0.00 1.00 0.000 0.000 0.0000 0.5000 0.2500
## 5 0.50 0.50 0.00 0.00 1.000 0.500 0.5000 0.2500 0.3750
## 6 0.50 0.50 0.00 0.00 0.500 1.000 0.2500 0.5000 0.3750
## 7 0.25 0.25 0.50 0.00 0.500 0.250 1.0000 0.1250 0.5625
## 8 0.25 0.25 0.00 0.50 0.250 0.500 0.1250 1.0000 0.5625
## 9 0.25 0.25 0.25 0.25 0.375 0.375 0.5625 0.5625 1.0625
```

Breed 2

Reading the data

```
tbl_ped2 <- readr::read_delim(file = s_ped2_path,
                              delim = ",",
                              col_types = readr::cols(
                                Animal = readr::col_character(),
                                Sire = readr::col_character(),
                                Dam = readr::col_character(),
                                AgeOwner = readr::col_double()
                              ))
```

Adding founder animals to the pedigree

```
vec_sire_founder <- unique(tbl_ped2$Sire)
vec_sire_founder <- vec_sire_founder[!is.element(vec_sire_founder, tbl_ped2$Animal)]
vec_dam_founder <- unique(tbl_ped2$Dam)
vec_dam_founder <- vec_dam_founder[!is.element(vec_dam_founder, tbl_ped2$Animal)]
vec_founder <- c(vec_sire_founder, vec_dam_founder)
n_nr_founder <- length(vec_founder)
tbl_ped2_founder <- tibble::tibble(Animal = vec_founder,
                                   Sire = rep(NA, n_nr_founder),
                                   Dam = rep(NA, n_nr_founder),
                                   AgeOwner = rep(NA, n_nr_founder))
tbl_ped2_aug <- dplyr::bind_rows(tbl_ped2_founder, tbl_ped2)
```

Remapping the IDs to integer values

```
tbl_ped_ID_map <- tibble::tibble(DomainCharId = tbl_ped2_aug$Animal,
                                 ImageIntId = c(1:length(tbl_ped2_aug$Animal)))
library(dplyr)
# Animals
tbl_ped2_int_id <- tbl_ped2_aug %>%
  left_join(tbl_ped_ID_map, by = c("Animal" = "DomainCharId")) %>%
  rename(AnimalIntId = ImageIntId)
# Sire
tbl_ped2_int_id <- tbl_ped2_int_id %>%
  left_join(tbl_ped_ID_map, by = c("Sire" = "DomainCharId")) %>%
  rename(SireIntId = ImageIntId)
# Dam
tbl_ped2_int_id <- tbl_ped2_int_id %>%
  left_join(tbl_ped_ID_map, by = c("Dam" = "DomainCharId")) %>%
  rename(DamIntId = ImageIntId)
```

Get the numerator relationship matrix with `pedigreemm`

```
ped_ped2 <- pedigreemm::pedigree(sire = tbl_ped2_int_id$SireIntId,
                                 dam =  tbl_ped2_int_id$DamIntId,
                                 label = as.character(tbl_ped2_int_id$AnimalIntId))
mat_A_ped2 <- as.matrix(pedigreemm::getA(ped = ped_ped2))
```

The numerator relationship matrix A_2 for the pedigree of breed 2 is

```
##     1    2    3     4     5     6     7
## 1 1.0 0.00 0.00 0.500 0.500 0.500 0.500
## 2 0.0 1.00 0.00 0.500 0.000 0.000 0.250
## 3 0.0 0.00 1.00 0.000 0.500 0.500 0.250
## 4 0.5 0.50 0.00 1.000 0.250 0.250 0.625
## 5 0.5 0.00 0.50 0.250 1.000 0.500 0.625
## 6 0.5 0.00 0.50 0.250 0.500 1.000 0.375
## 7 0.5 0.25 0.25 0.625 0.625 0.375 1.125
```

■

b) Compute the inbreeding coefficients for all animals in both pedigrees of the two breeds given above.

Berechnen Sie die Inzuchtkoeffizienten für alle Tiere in beiden Pedigrees der zwei Rassen, welche oben gegeben sind.

16

Solution:

Breed 1

Inbreeding coefficients of animals are

```
tbl_inb1 <- tibble::tibble(Animal = tbl_ped1_aug$Animal,
                           `Inbreeding Coefficients` = diag(mat_A_ped1) - 1)
```

The results of the inbreeding coefficients of all animals are shown in the following table

Animal	Inbreeding Coefficients
CH114000329	0.0000
CH114000277	0.0000
CH114000255	0.0000
CH114000248	0.0000
CH114000313	0.0000
CH114000368	0.0000
CH114000285	0.0000
CH114000256	0.0000
CH114000397	0.0625

Breed 2

Inbreeding coefficients of animals are

```
tbl_inb2 <- tibble::tibble(Animal = tbl_ped2_aug$Animal,
                           `Inbreeding Coefficients` = diag(mat_A_ped2) - 1)
```

The results of the inbreeding coefficients of all animals are shown in the following table

Animal	Inbreeding Coefficients
CH115000277	0.000
CH115000301	0.000
CH115000256	0.000
CH115000392	0.000
CH115000319	0.000
CH115000346	0.000
CH115000347	0.125

■

c) For our example breeds, we use a simplified version of the index (I_E) indicating which breeds are endangered. This version of the index is based on the average inbreeding coefficients of all animals in a given breed and on the average age of the owners of the animals in that breed. Compute for both given breeds the index I_E using a weight $w_F = 1$ for the average inbreeding coefficient and a weight $w_O = 0.002$ for the average age of the owners of the animals. The index can be computed as shown below. Determine based on the value of the index for both breeds which breed is considered to be more endangered.

Für die zwei Beispielrassen wird eine vereinfachte Form des Gefährungsindexes I_E *berechnet. Diese Version des Indexes basiert auf dem durchschnittlichen Inzuchtkoeffizienten aller Tiere im Pedigree und auf dem durchschnittlichen Alter der Tierhalter einer bestimmten Rasse. Berechnen Sie für beide gegebenen Rassen den index* I_E *mit den Gewichten* $w_F = 1$ *für den durchschnittlichen Inzuchtkoeffizient und* $w_O = 0.002$ *für das durchschnittliche Alter der Tierhalter. Der Index kann wie nachfolgend gezeigt berechent werden. Entscheiden Sie basierend auf dem Indexwert für beide Rassen, welche Rasse als gefährdeter eingestuft würde.*

3

For a given breed the endangerment index I_E is computed as

Für eine bestimmte Rasse kann der Gefährdungsindex I_E *wie folgt berechnet werden*

$$I_E = w_F * \bar{F} + w_O * \bar{O}$$

where $\bar{F}$ is the average inbreeding coefficient of all animals in the pedigree and $\bar{O}$ is the average age of all owners of animals of a given breed.

Solution:

Breed 1

```
vec_av_inb1 <- mean(diag(mat_A_ped1) - 1)
vec_av_age1 <- mean(tbl_ped1$AgeOwner)
n_index_e1 <- n_weight_AvInb * vec_av_inb1 + n_weight_AvAge * vec_av_age1
```

$$I_E = 1 * 0.0069444 + 0.002 * 75 = 0.1569444$$

Breed 2

```
vec_av_inb2 <- mean(diag(mat_A_ped2) - 1)
vec_av_age2 <- mean(tbl_ped2$AgeOwner)
n_index_e2 <- n_weight_AvInb * vec_av_inb2 + n_weight_AvAge * vec_av_age2
```

$$I_E = 1 * 0.0178571 + 0.002 * 52.5 = 0.1228571$$

Based on the much higher average age of the owners of the animals in breed 1, breed 1 is considered to be more endangered.

■

Problem 3 Genomics

The following dataset is used to predict genomic breeding values. The column `y` contains the observations and columns `S1` to `S15` contain genotypes of 15 SNP marker loci.

Der folgende Datensatz wird für die Schätzung genomischer Zuchtwerte verwendet. Die Kolonne `y` *enthält die beobachteten Merkmalswerte und die Kolonnen* `S1` *bis* `S15` *enthalten die Genotypen von* 15 SNP-Marker.

ID	sex	y	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
1	M	59.0	1	1	0	0	0	0	1	1	1	0	1	0	2	1	1
2	M	33.5	0	0	0	1	0	1	1	1	1	0	1	1	0	0	0
3	F	75.1	0	0	0	1	0	1	1	1	1	0	0	1	2	1	0
4	F	83.9	0	0	0	1	1	1	1	1	1	0	1	2	1	0	0
5	F	52.4	0	0	1	1	0	1	1	0	1	0	0	1	0	0	0
6	M	62.5	0	0	1	0	1	0	0	2	2	0	0	0	1	1	1
7	F	59.7	1	1	0	0	1	1	1	1	0	1	0	0	0	1	0
8	M	81.1	1	0	2	1	1	1	1	0	1	0	1	1	1	1	2
9	M	54.4	0	0	0	1	0	1	0	1	1	1	1	2	1	0	0
10	M	29.0	0	0	0	1	0	1	1	0	1	0	0	1	0	0	2

The above dataset is available under:

Der oben gezeigte Datensatz ist verfügbar unter:

```
## https://charlotte-ngs.github.io/lbgfs2022/data/lbgfs2022_exam_problem3.csv
```

a) Use a marker effect model (MEM) to predict genomic breeding values for the animals given in the dataset shown above. Specify the marker effect model as a mathematical formula and list the meaning of all model components. Indicate the expected values and the variance-covariance matrices for all random effects in the model. You can assume $\lambda = \sigma_e^2/\sigma_q^2 = 1$.

Verwenden Sie ein Marker-Effekt-Modell (MEM) zur Schätzung der genomischen Zuchtwerte für die Tiere im oben gezeigten Datensatz. Spezifizieren Sie das Marker-Effekt-Modell als mathematische Formel und notieren Sie die Bedeutung aller Modellkomponenten. Geben Sie die Erwartungswerte und die Varianz-Kovarianz-Matrizen aller zufälligen Effekte an. Wir nehmen an, dass $\lambda = \sigma_e^2/\sigma_q^2 = 1$.

20

Solution:

Read the data

```
s_pr03_data_path <- file.path(s_data_root, "lbgfs2022_exam_problem3.csv")
tbl_pr3 <- readr::read_delim(s_pr03_data_path,
                             delim = ",",
                             col_types = readr::cols(
                               sex = readr::col_factor(),
                               y = readr::col_double(),
                               .default = readr::col_integer()
                             ))
n_nr_animal <- nrow(tbl_pr3)
n_nr_loci <- length(grep("S", colnames(tbl_pr3)))
```

Set up a marker effect model which in this case has to be a linear mixed effect model

$$y = X\beta + Wq + e$$

with $\beta^T = \left[\begin{array}{cc} \beta_M & \beta_F \end{array}\right]$ the vector of fixed sex effects, q the vector of random marker effects, e the vector of random residuals and y the vector of observations. X is the design matrix linking fixed effects to observations. The matrix W contains the genotypes in a $-1, 0, 1$ encoding.

The expected values and the variance-covariance matrices for all random effects are

$$E\left[\begin{array}{c} y \\ q \\ e \end{array}\right] = \left[\begin{array}{c} X\beta \\ 0 \\ 0 \end{array}\right], var\left[\begin{array}{c} y \\ q \\ e \end{array}\right] = \left[\begin{array}{ccc} WW^T\sigma_q^2 + R & W\sigma_q^2 & R \\ W^T\sigma_q^2 & I\sigma_q^2 & 0 \\ R & 0 & R \end{array}\right]$$

with $R = I\sigma_e^2$. The components σ_e^2 and σ_q^2 are the residual variance components and the marker variance components.

Design matrix for fixed effects

```
# info from dataset to model component
vec_y <- tbl_pr3$y
# design matrix for fixed effects
mat_X <- model.matrix(lm(y ~ 0 + sex, data = tbl_pr3))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
mat_X
```

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

Genotype Matrix

```
tbl_SNP <- dplyr::select(tbl_pr3, dplyr::starts_with("S", ignore.case = FALSE))
mat_W <- as.matrix(tbl_SNP) - 1
colnames(mat_W) <- NULL
mat_W
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0   -1   -1   -1   -1    0    0    0    -1     0    -1     1
##  [2,]   -1   -1   -1    0   -1    0    0    0    0    -1     0     0    -1
##  [3,]   -1   -1   -1    0   -1    0    0    0    0    -1    -1     0     1
##  [4,]   -1   -1   -1    0    0    0    0    0    0    -1     0     1     0
##  [5,]   -1   -1    0    0   -1    0    0   -1    0    -1    -1     0    -1
##  [6,]   -1   -1    0   -1    0   -1   -1    1    1    -1    -1    -1     0
##  [7,]    0    0   -1   -1    0    0    0    0   -1     0    -1    -1    -1
##  [8,]    0   -1    1    0    0    0    0   -1    0    -1     0     0     0
##  [9,]   -1   -1   -1    0   -1    0   -1    0    0     0     0     1     0
## [10,]   -1   -1   -1    0   -1    0    0   -1    0    -1    -1     0    -1
##       [,14] [,15]
##  [1,]     0     0
##  [2,]    -1    -1
##  [3,]     0    -1
##  [4,]    -1    -1
##  [5,]    -1    -1
##  [6,]     0     0
##  [7,]     0    -1
##  [8,]     0     1
##  [9,]    -1    -1
## [10,]    -1     1
```

Mixed model equations

```
lambda_mem <- 1
mat_xtx <- crossprod(mat_X)
mat_xtw <- crossprod(mat_X, mat_W)
mat_wtx <- t(mat_xtw)
mat_wtwlI <- crossprod(mat_W) + lambda_mem * diag(nrow = n_nr_loci)
mat_coef <- rbind(cbind(mat_xtx,mat_xtw),cbind(mat_wtx,mat_wtwlI))
mat_rhs <- rbind(crossprod(mat_X, vec_y),
                 crossprod(mat_W, vec_y))
mat_sol <- solve(mat_coef,mat_rhs)
mat_sol
```

```
##             [,1]
##  [1,] 69.9596943
##  [2,] 91.6355481
##  [3,]  3.4193888
##  [4,] -0.9991921
##  [5,]  6.4841815
##  [6,]  0.2877344
##  [7,]  8.0271800
##  [8,] -0.3455088
##  [9,] -1.9298590
## [10,]  2.6912638
## [11,]  1.3447009
## [12,]  0.5851581
## [13,]  6.1528755
## [14,]  5.0365200
## [15,]  9.4815198
## [16,]  4.2981433
## [17,]  2.5147730
```

The genomic breeding values are computed as

```
vec_marker_sol <- mat_sol[3:(nrow(mat_sol)),1]
tbl_mem_dgw <- tibble::tibble(Animal = c(1:n_nr_animal),
                              `Genomic BV`= crossprod(t(mat_W), vec_marker_sol))
knitr::kable(tbl_mem_dgw, booktabs = TRUE, longtable = TRUE)
```

Animal	Genomic BV
1	-10.593745
2	-33.811152
3	-16.702845
4	-11.265933
5	-36.171110

Animal	Genomic BV
6	-8.171152
7	-31.302305
8	6.721725
9	-16.778095
10	-37.625746

Rank animals according to the DGW

```
order(tbl_mem_dgw$`Genomic BV`, decreasing = TRUE)
```

```
##  [1]  8  6  1  4  3  9  7  2  5 10
```

■

b) Predict genomic breeding values using a breeding value based model (BVM) for all animals in the above given dataset. Specify the model as a mathematical formula and list the meaning of all model components. Indicate the expected values and the variance-covariance matrices for all random effects in the model. You can assume $\lambda = \sigma_e^2/\sigma_g^2 = 3$.

Schätzen Sie genomische Zuchtwerte mit einem Zuchtwert-basierten Modell (BVM) für alle Tiere im oben gezeigten Datensatz. Spezifizieren Sie das Modell als mathematische Formel und notieren Sie die Bedeutung aller Modellkomponenten. Geben Sie die Erwartungswerte und die Varianz-Kovarianz-Matrizen aller zufälligen Effekte an. Wir nehmen an, dass $\lambda = \sigma_e^2/\sigma_g^2 = 3$.

25

Solution:

We assume that the dataset is read as shown in the solution of 3a). Futhermore the definition of the number of animals and the number of loci are executed.

The breeding value based model is

$$y = X\beta + Zg + e$$

with $\beta^T = \left[\begin{array}{cc} \beta_M & \beta_F \end{array}\right]$ the vector of fixed sex effects, g the vector of random genomic breeding values, e the vector of random residuals and y the vector of observations. X is the design matrix linking fixed effects to observations. The matrix Z is the design matrix linking genomic breeding values to observations.

The expected values and the variance-covariance matrices for all random effects are

$$E\left[\begin{array}{c} y \\ g \\ e \end{array}\right] = \left[\begin{array}{c} X\beta \\ 0 \\ 0 \end{array}\right], var\left[\begin{array}{c} y \\ g \\ e \end{array}\right] = \left[\begin{array}{ccc} ZHZ^T + R & ZH & R \\ HZ^T & H & 0 \\ R & 0 & R \end{array}\right]$$

with $R = I\sigma_e^2$ and $H = G * \sigma_g^2$. The matrix G is the genomic relationship matrix and σ_e^2 and σ_g^2 are the residual and the genetic variance components.

Design matrix X for fixed effects

```
# info from dataset to model component
vec_y <- tbl_pr3$y
# design matrix for fixed effects
mat_X <- model.matrix(lm(y ~ 0 + sex, data = tbl_pr3))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
```

```
colnames(mat_X) <- NULL
mat_X
```

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

Design matrix Z for the genomic breeding values

```
mat_Z <- diag(nrow = n_nr_animal)
mat_Z
```

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

The genomic relationship matrix is computed using the following function

```
#' 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)
}
```

For our dataset the genomic relationship matrix G is

```
tbl_SNP <- dplyr::select(tbl_pr3, dplyr::starts_with("S", ignore.case = FALSE))
mat_W <- as.matrix(tbl_SNP) - 1
colnames(mat_W) <- NULL
mat_G <- computeMatGrm(pmatData = mat_W)
mat_G
```

```
##              [,1]        [,2]        [,3]         [,4]         [,5]        [,6]
##  [1,]  0.97807593 -0.27437446  0.13977567 -0.291630716 -0.464193270  0.27782571
##  [2,] -0.27437446  0.39136324 -0.06729940  0.191544435  0.191544435 -0.27437446
##  [3,]  0.13977567 -0.06729940  0.52941329  0.088006903 -0.084555651 -0.03278689
##  [4,] -0.29163072  0.19154443  0.08800690  0.529413287  0.001725626 -0.29163072
##  [5,] -0.46419327  0.19154443 -0.08455565  0.001725626  0.529413287 -0.29163072
##  [6,]  0.27782571 -0.27437446 -0.03278689 -0.291630716 -0.291630716  1.15063848
##  [7,]  0.15703192 -0.05004314 -0.15358067 -0.239861950 -0.067299396 -0.01553063
##  [8,]  0.03623814 -0.34339948 -0.27437446 -0.188093184 -0.015530630  0.03623814
##  [9,] -0.29163072  0.19154443  0.08800690  0.346850733  0.001725626 -0.29163072
## [10,] -0.25711821  0.05349439 -0.22260569 -0.136324418  0.208800690 -0.25711821
##              [,7]        [,8]         [,9]       [,10]
##  [1,]  0.15703192  0.03623814 -0.291630716 -0.25711821
##  [2,] -0.05004314 -0.34339948  0.191544435  0.05349439
##  [3,] -0.15358067 -0.27437446  0.088006903 -0.22260569
##  [4,] -0.23986195 -0.18809318  0.346850733 -0.13632442
##  [5,] -0.06729940 -0.01553063  0.001725626  0.20880069
##  [6,] -0.01553063  0.03623814 -0.291630716 -0.25711821
##  [7,]  1.08161346 -0.25711821 -0.239861950 -0.20534944
##  [8,] -0.25711821  1.18515099 -0.360655738  0.19154443
##  [9,] -0.23986195 -0.36065574  0.701975841 -0.13632442
## [10,] -0.20534944  0.19154443 -0.136324418  0.77100086
```

The mixed model equations

```
lambda_bvm <- 3
mat_xtx <- crossprod(mat_X)
mat_xtz <- crossprod(mat_X, mat_Z)
mat_ztx <- crossprod(mat_Z, mat_X)
stopifnot(min(mat_ztx - t(mat_xtz)) == 0)
stopifnot(max(mat_ztx - t(mat_xtz)) == 0)
mat_ztz_lGinv <- crossprod(mat_Z) + lambda_bvm * solve(mat_G)
# coefficient matrix
mat_coef <- rbind(cbind(mat_xtx,mat_xtz),cbind(mat_ztx,mat_ztz_lGinv))
# rhs
mat_rhs <- rbind(crossprod(mat_X, vec_y),
                 crossprod(mat_Z, vec_y))
# solve
mat_sol <- solve(mat_coef, mat_rhs)
mat_sol
```

```
##             [,1]
##  [1,] 52.2384937
##  [2,] 69.2922594
##  [3,]  4.4199386
##  [4,] -4.9378309
##  [5,]  1.6020347
##  [6,]  1.1369785
##  [7,] -5.1539382
##  [8,]  4.8672289
##  [9,] -3.6541127
## [10,]  7.8556420
## [11,] -0.5679495
## [12,] -5.5679915
```

The solutions for the estimates of the fixed effects are

```
mat_sol[1:2,]
```

```
## [1] 52.23849 69.29226
```

The solutions for the predicted breeding values are

```
tbl_pred_bv <- tibble::tibble(Animal = c(1:n_nr_animal),
                              `Predicted Genomic BV` = mat_sol[3:(nrow(mat_sol)),1])
knitr::kable(tbl_pred_bv)
```

Animal	Predicted Genomic BV
1	4.4199386
2	-4.9378309
3	1.6020347
4	1.1369785
5	-5.1539382
6	4.8672289
7	-3.6541127
8	7.8556420
9	-0.5679495
10	-5.5679915

The ranking of the animals according to the predicted genomic breeding values are

```
order(tbl_pred_bv$`Predicted Genomic BV`, decreasing = TRUE)
```

```
##  [1]  8  6  1  3  4  9  7  2  5 10
```

■

c) Compare the order of the predicted breeding values of all animals based on the marker effect model (MEM) and based on the breeding value model (BVM)

   Vergleichen Sie die Reihenfolge der Tiere aufgrund der geschätzten genomischen Zuchtwerte basierend auf dem Markereffektmodell (MEM) und basierend auf dem Zuchtwert-basierten Modell (BVM)

2

Solution:

```
order(tbl_mem_dgw$`Genomic BV`, decreasing = TRUE)
```

```
##  [1]  8  6  1  4  3  9  7  2  5 10
```

```
order(tbl_pred_bv$`Predicted Genomic BV`, decreasing = TRUE)
```

```
##  [1]  8  6  1  3  4  9  7  2  5 10
```

■

Problem 4 BLUP Animal Model

id	sire	dam	herd	y
4	2	1	1	26.52
6	2	3	1	39.71
8	4	5	1	25.11
9	6	7	2	24.02
11	8	9	2	6.15
13	11	10	2	8.81
15	11	12	2	18.56
17	13	14	3	47.81
18	15	16	3	51.72
20	18	19	3	47.33

The above dataset is available under:

Der oben gezeigte Datensatz ist verfügbar unter:

```
## https://charlotte-ngs.github.io/lbgfs2022/data/lbgfs2022_exam_problem4.csv
```

a) Use the above shown dataset to predict breeding values for all animals in the pedigree using a BLUP animal model. Use 'herd' as fixed effect and compute estimates for all the levels of the fixed effect. The column entitled 'y' contains the values for an observed trait of interest. Specify the model as mathematical formula and list the meaning of all model components. Write down the expected values and the variance-covariance matrices for all random effects in the model. Assume the phenotypic variance to be $\sigma_p^2 = 64$ and the heritability to be $h^2 = 0.16$.

Verwenden Sie den oben gezeigten Datensatz für die Schätzung von Zuchtwerten für alle Tiere im Pedigree mit einem BLUP-Tiermodell. Modellieren Sie `herd` *als fixen Effekt. Berechnen Sie Schätzwerte für alle Stufen des fixen Effekts. Die Kolonne mit dem Titel* `y` *enthält die Beobachtungswerte eines Merkmals von züchterischem Interesse. Spezifizieren Sie das Modell als mathematische Formel und benennen Sie die Bedeutung jeder Modellkomponente. Notieren Sie die Erwartungswerte und die Varianz-Kovarianzmatrizen aller zufälligen Effekte im Modell. Die phänotypische Varianz wird angenommen als* $\sigma_p^2 = 64$ *und die Erblichkeit sei* $h^2 = 0.16$.

22

Solution:

The BLUP animal model is

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

with $\beta^T = \left[\begin{array}{ccc} \beta_{h1} & \beta_{h2} & \beta_{h3} \end{array}\right]$ the vector of fixed herd effects, u the vector of random breeding values, e the vector of random residuals and y the vector of observations. X is the design matrix linking fixed effects to observations. The matrix Z is the design matrix linking the breeding values to observations.

The expected values and the variance-covariance matrices for all random effects are

$$E\left[\begin{array}{c} y \\ u \\ e \end{array}\right] = \left[\begin{array}{c} X\beta \\ 0 \\ 0 \end{array}\right], var\left[\begin{array}{c} y \\ u \\ e \end{array}\right] = \left[\begin{array}{ccc} ZGZ^T + R & ZG & R \\ GZ^T & G & 0 \\ R & 0 & R \end{array}\right]$$

with $R = I\sigma_e^2$ and $G = A * \sigma_g^2$. The matrix A is the numerator relationship matrix and σ_e^2 and σ_u^2 are the residual and the genetic variance components.

Design matrix X for fixed effects

```
mat_X <- model.matrix(lm(y ~ 0 + as.factor(herd), data = tbl_data_p04_read))
attr(mat_X, "assign") <- NULL
attr(mat_X, "contrasts") <- NULL
colnames(mat_X) <- NULL
mat_X
```

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

The pedigree

```
vec_founder_sire <-
  unique(tbl_data_p04_read$sire[!is.element(tbl_data_p04_read$sire,
                                            tbl_data_p04_read$id)])
```

```
vec_founder_dam <-
  unique(tbl_data_p04_read$dam[!is.element(tbl_data_p04_read$dam,
                                           tbl_data_p04_read$id)])
vec_founder_cmb <- c(vec_founder_sire, vec_founder_dam)
n_nr_founder <- length(vec_founder_cmb)
tbl_ped_aug <- dplyr::bind_rows(tibble::tibble(id = vec_founder_cmb,
                                               sire = rep(NA, n_nr_founder),
                                               dam = rep(NA, n_nr_founder)),
                                dplyr::select(tbl_data_p04_read, id, sire, dam))
n_nr_ani_ped <- nrow(tbl_ped_aug)
```

The inverse numerator relationship matrix

```
ped_p04 <- pedigreemm::pedigree(sire = as.integer(tbl_ped_aug$sire),
                                dam  = as.integer(tbl_ped_aug$dam),
                                label = tbl_ped_aug$id)
mat_A_inv_p04 <- as.matrix(pedigreemm::getAInv(ped = ped_p04))
```

Design matrix Z for the random breeding values

```
mat_Z <- cbind(matrix(0, nrow = n_nr_obs, ncol = n_nr_founder), diag(nrow = n_nr_obs))
mat_Z
```

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
##  [1,]    0    0    0    0    0    0    0    0    0     0     1     0     0
##  [2,]    0    0    0    0    0    0    0    0    0     0     0     1     0
##  [3,]    0    0    0    0    0    0    0    0    0     0     0     0     1
##  [4,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [5,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [6,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [7,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [8,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##  [9,]    0    0    0    0    0    0    0    0    0     0     0     0     0
## [10,]    0    0    0    0    0    0    0    0    0     0     0     0     0
##       [,14] [,15] [,16] [,17] [,18] [,19] [,20]
##  [1,]     0     0     0     0     0     0     0
##  [2,]     0     0     0     0     0     0     0
##  [3,]     0     0     0     0     0     0     0
##  [4,]     1     0     0     0     0     0     0
##  [5,]     0     1     0     0     0     0     0
##  [6,]     0     0     1     0     0     0     0
##  [7,]     0     0     0     1     0     0     0
##  [8,]     0     0     0     0     1     0     0
##  [9,]     0     0     0     0     0     1     0
## [10,]     0     0     0     0     0     0     1
```

The mixed model equations

```
# genetic parameter
n_sigma_u2 <- n_h2 * n_sigma_p2
n_sigma_e2 <- n_sigma_p2 - n_sigma_u2
n_lambda <- n_sigma_e2 / n_sigma_u2
# mme components
vec_y_read <- tbl_data_p04_read$y
mat_xtx <- crossprod(mat_X)
mat_xtz <- crossprod(mat_X, mat_Z)
mat_ztx <- crossprod(mat_Z, mat_X)
stopifnot(min(mat_ztx - t(mat_xtz)) == 0)
stopifnot(max(mat_ztx - t(mat_xtz)) == 0)
mat_ztzlAinv <- crossprod(mat_Z) + n_lambda * mat_A_inv_p04
mat_coef <- rbind(cbind(mat_xtx,mat_xtz), cbind(mat_ztx,mat_ztzlAinv))
# rhs
mat_rhs <- rbind(crossprod(mat_X, vec_y_read),
                 crossprod(mat_Z, vec_y_read))
# sol
mat_sol <- solve(mat_coef, mat_rhs)
```

The estimates for the fixed herd effects are

```
n_nr_herd_levels <- ncol(mat_X)
mat_sol[1:n_nr_herd_levels,]
```

```
##
## 30.79364 14.47310 49.08568
```

The predicted breeding values are

```
mat_sol[(n_nr_herd_levels+1):(nrow(mat_sol)),]
```

```
##           2           1           3           5           7          10
##  0.22132323 -0.69454426  0.91586749 -0.75242546  0.41613474 -0.46384100
##          12          14          16          19           4           6
##  0.43740398 -0.06012234  0.14004106 -0.15983744 -0.93115477  1.48446286
##           8           9          11          13          15          17
## -1.59421558  1.36643354 -0.83980476 -1.10841636  0.22936915 -0.64439168
##          18          20
##  0.32474616 -0.07738308
```

■

b) Compute the reliabilities for the predicted breeding values of all animals under Problem 4a).

   Berechnen Sie die Bestimmtheitsmasse der unter Aufgabe 4a) geschätzten Zuchterte.

6

Solution:

```
n_nr_herd_levels <- ncol(mat_X)
mat_coef_full <- mat_coef / n_sigma_e2
vec_pev <- diag(solve(mat_coef_full))[(n_nr_herd_levels+1):nrow(mat_coef)]
mat_A_p04 <- as.matrix(pedigreemm::getA(ped = ped_p04))
vec_A_diag <- diag(mat_A_p04)
vec_rel <- 1 - (vec_pev / (n_sigma_u2 * vec_A_diag))
vec_rel
```

```
##           2           1           3           5           7          10 
## 0.009724671 0.021964884 0.030040689 0.032716767 0.015855630 0.036699753 
##          12          14          16          19           4           6 
## 0.035434729 0.026335939 0.020322458 0.028235027 0.053088807 0.083373076 
##           8           9          11          13          15          17 
## 0.079780594 0.075725855 0.040087576 0.075891408 0.071096186 0.106740332 
##          18          20 
## 0.083547715 0.078504525
```

■

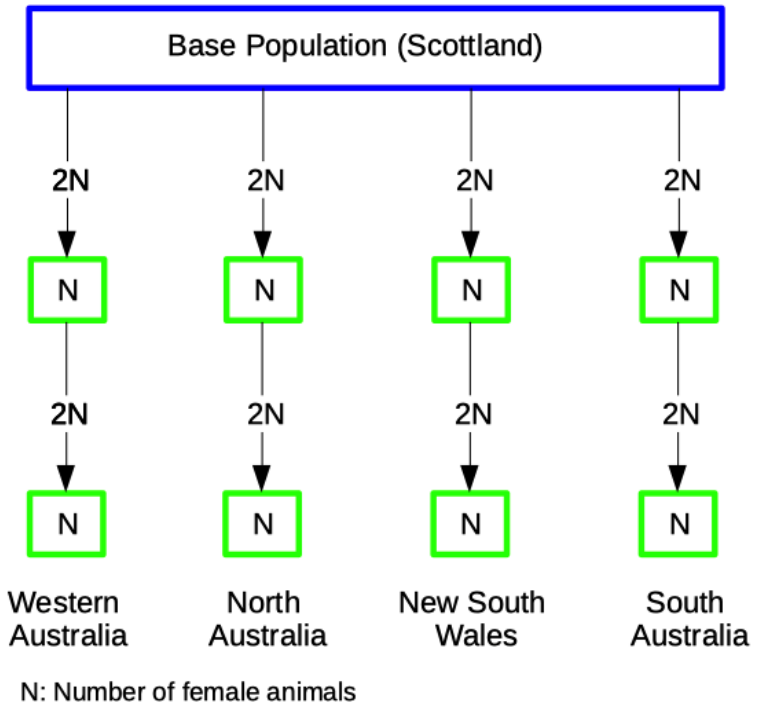
Problem 5 Variance and Inbreeding

In the year 1892 a group of Scottish farmers landed with their sheep in Australia. The farmers took a total of 2200 sheeps from Scotland to Australia. Once the farmers arrived in Australia, they separated in 4 subgroups of equal sizes. Each of the subgroups went to a different state of Australia (Western Australia, North Australia, New South Wales and South Australia). In the year 2022 Australian sheep farmers want to export some of their breeding animals. For this problem you can work with the following assumptions

- There was no exchange of animals between the sub-populations after the split into subgroups.
- The ratio between the number of female animals and the number of male animals is 10.
- The size of the sub-populations N which is relevant for the computation of the inbreeding coefficient is set to the number of female animals in each sub-population.
- The generation interval can assumed to be 1.3 years.

Im Jahr 1892 wanderte eine Gruppe von Schottischen Farmern mit ihren Schafen nach Australien aus. Die Farmer brachten 2200 Schafe nach Australien. Als die Farmer in Australien ankamen teilten sie sich in 4 gleich grosse Gruppen auf. Jede Gruppe ging in einen anderen Staat in Australien (Western Australia, North Australia, New South Wales and South Australia). Im Jahr 2022 möchten die Farmer einige ihrer Zuchttiere exportieren. Für diese Aufgabe können Sie die folgenden Annahmen treffen.

- *Nach der Aufteilung der Tiere in die Gruppen kam es zu keinem Tieraustausch zwischen den Gruppen.*
- *Das Verhältnis zwischen der Anzahl weiblicher Tiere und der Anzahl männlicher Tiere beträgt 10.*
- *Die Popultationsgrösse N, welche für die Berechnung des Inzuchtkoeffizienten wichtig ist, kann mit der Anzahl weiblichen Tiere in jeder Subpopulation gleichgesetzt werden.*
- *Das Generationeninterval betrage 1.3 Jahre.*



a) Compute the inbreeding coefficient F_t for the breeding animals that the farmers want to sell.

   Berechnen Sie den Inzuchtkoeffizienten F_t für die Zuchttiere, welche die Farmer verkaufen möchten.

4

Solution:

The inbreeding coefficient F_t is computed as

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

where ΔF corresponds to $1/(2N)$ and t is equal to the number of generations. Inserting the number leads to

$$F_t = 1 - (1 - 0.001)^{100} = 0.0952$$

■

b) The sheep farmers are concerned that inbreeding in their population does not increase too much. In which year is the inbreeding coefficient F_t going to be bigger than 0.1?

Die Farmer möchten den Inzuchtgrad nicht zu stark ansteigen lassen. In welchem Jahr wird der Inzuchtgrad F_t grösser sein als 0.1?

2

Solution:

We are given the limit of F_t and we want to know the value for t when this limit is reached. Hence we have to solve the equation for F_t after t. Hence

$$\begin{aligned} F_t &= 1 - (1 - \Delta F)^t \\ (1 - \Delta F)^t &= 1 - F_t \\ t &= \frac{log(1 - F_t)}{log(1 - \Delta F)} \\ &= \frac{log(1 - 0.1)}{log(1 - 0.001)} = 105.3078266 \end{aligned}$$

This means after 106 generations the limit is reached. With a generation interval of 1.3 years, this will be the limit of the inbreeding coefficient will be reached in the year 2030.

■

c) One reason to control the inbreeding coefficient is that breeders want to avoid inbreeding depression. We assume that locus W is mainly responsible for wool fibre diameter (FD). The favorable allele W_1 occurs with a frequency of $p = 0.045$. The difference between the homozygous genotypes W_1W_1 and W_2W_2 in fiber diameter is 100 micrometer (μm). The genotypic value of the heterozygous genotype W_1W_2 is 15. Compute the inbreeding depression at locus W, if the inbreeding coefficient has reached the limiting value of Problem 1b of 0.1.

Züchter wollen die Inzucht begrenzen, da sie Inzuchtdepressionen vermeiden wollen. Wir nehmen an, dass das Merkmal Wollfaserdurchmesser hauptsächlich von einem Genort W beeinflusst wird. Das vorteilhafte Allel W_1 *kommt mit einer Häufigkeit von* $p = 0.045$ *vor. Die Differenz zwischen den homozygoten Genotypen* W_1W_1 *und* W_2W_2 *im Merkmal Wollfaserdurchmesser beträgt* 100 *Mikrometer (μm). Der genotypische Wert der Heterozygoten* W_1W_2 *beträgt 15. Berechnen Sie die Inzuchtdepression am Genort W unter der Annahme, dass der Inzuchtkoeffizient den Grenzwert aus Aufgabe 1b von 0.1 erreicht hat.*

6

Solution:

The inbreeding depression is computed as

$$M_0 - M_F = 2d\bar{p}\bar{q}F = 2 * 15 * 0.045 * (1 - 0.045) * 0.1 = 0.1289$$

■

d) Inbreeding has an influence on the genetic additive variance, as it is split into a between line and a within line component. Please, fill out the following table with the different genetic variance components for the locus W from Problem 5c. We assume a value of 0.1 for the inbreeding coefficient F.

Inzucht hat einen Einfluss auf die additive genetische Varianz, da diese Varianz durch die Inzucht in eine Komponente innerhalb Linie und eine Komponente zwischen Linien aufgeteilt wird. Bitte füllen Sie die folgende Tabelle mit den unterschiedlichen Varianzkomponenten am Genort W aus Aufgabe 5c aus. Als Inzuchtkoeffizienten F nehmen wir einen Wert von 0.1 *an.*

10

Solution:

Source	Variance
Between lines	
Within lines	
Total additive	
Dominance	
Total genetic	

Solution:

Source	Variance	Values
Between lines	$2FV_U$	42.98
Within lines	$(1-F)V_U$	193.39
Total additive	$(1+F)V_U$	236.36
Dominance	V_D	1.66
Total genetic	V_G	238.02

■