

Livestock Breeding and Genomics - Exercise 2

Peter von Rohr

2020-10-02

Problem 1: Breeding Values For a Monogenic Trait

We assume that the absorption of cholesterol is determined by a certain enzyme. The level of enzyme production is determined by a single bi-allelic locus E . The genotype frequencies and the genotypic values for the two dairy cattle populations **Original Braunvieh** and **Brown Swiss** are given in the following table.

Variable	Original Braunvieh	Brown Swiss
$f(E_1E_1)$	0.0625	0.01
$f(E_1E_2)$	0.3750	0.18
$f(E_2E_2)$	0.5625	0.90
a	15.0000	29.00
d	3.0000	0.00

Hints

- Assume that allele E_1 is the allele with the positive effect on the enzyme level
- Assume that the Hardy-Weinberg Equilibrium holds in both populations

Your Task

Compute the breeding values for all three genotypes in both populations.

Problem 2: Matrices in R

In R, matrices are constructed using the function `matrix()`. This function accepts different options. We want to see, how these options work.

Your Task: Construct matrices using the different options to better understand the meaning of the different options.

Parameter data

- data: Specify the different matrix elements

```
(matA <- matrix(data = c(1:9), nrow = 3, ncol = 3))
```

- data: without specifying the matrix elements

```
(matB <- matrix(nrow = 3, ncol = 3))
```

- data: specifying not all matrix elements

```
(matC <- matrix(data = c(1,2,3), nrow = 3, ncol = 3))
```

```
(matC2 <- matrix(data = c(1,2,3,4), nrow = 3, ncol = 3))
```

Parameters nrow and ncol

- Leaving out one of both parameters

```
(matD <- matrix(data = c(1:9), nrow = 3))
```

```
(matE <- matrix(data = c(1:9), ncol = 3))
```

Parameter byrow

```
(matF <- matrix(data = c(1:9), nrow = 3, ncol = 3, byrow = TRUE))
```

```
(matG <- matrix(data = c(1:9), nrow = 3, ncol = 3, byrow = FALSE))
```

Problem 3: Matrix multiplication in R

In R, matrices can be multiplied using the operator `%*%` or with the functions `crossprod()` or `tcrossprod()`. With `crossprod()` and `tcrossprod()` vectors and matrices can be multiplied directly. The conversion of vectors to matrices is done automatically inside of these functions. The result will always be a matrix. When doing matrix-vector multiplications with `%*%` the vector has to be converted first into a matrix using the function `as.matrix()`.

In a first part of this problem, compare the results of the functions `crossprod()`, `tcrossprod()` and `%*%`.

- a) Given are the following matrices

```
matA <- matrix(data = c(1:9), ncol = 3)
matB <- matrix(data = c(2:10), ncol = 3)
```

Find out which matrix multiplication with `%*%` corresponds to the following statement?

```
crossprod(matA,matB)
```

b) Given is the vector `vecB`

```
vecB <- c(-3,16,1)
```

Multiply the matrix `matA` with the vector `vecB` once using `%*%` and then with the function `crossprod()`.

Hint: a vector can be converted to a matrix using the function `as.matrix()`.

Problem 4: Quantitative Genetics

In a population the following numbers of genotypes were counted for a given genetic locus called *A*.

```
dfGenotypeFreq <- data.frame(Genotypes = c("$A_1A_1$", "$A_1A_2$", "$A_2A_2$"),
                             Numbers    = c(24, 53, 23),
                             stringsAsFactors = FALSE)
knitr::kable(dfGenotypeFreq)
```

Genotypes	Numbers
A_1A_1	24
A_1A_2	53
A_2A_2	23

- Compute the genotype frequencies
- Compute the allele frequencies
- Compute the population mean μ under the following assumptions
 - the difference between the genotypic values of the homozygous genotypes is 20 and
 - the genotypic value of the heterozygous genotype is 2.