

Züchtungslehre - Übung 7

Peter von Rohr

2017-11-17

Aufgabe 1:

Gegeben ist das folgende Pedigree. Berechnen Sie die Inzuchtkoeffizienten aller Tiere im Pedigree und die Elemente der Matrix R , wobei R aufgrund der Cholesky-Zerlegung der additiv genetischen Verwandtschaftsmatrix A definiert ist. Es gilt also

$$A = R * R^T \quad (1)$$

wobei R eine linke untere Dreiecksmatrix ist.

```
##   sire  dam
## 1 <NA> <NA>
## 2 <NA> <NA>
## 3   1   2
## 4   1   3
## 5   4   3
## 6   5   2
## 7   5   3
## 8   6   7
```

Hinweise: Sie können Ihre Resultate überprüfen mit folgenden R-Funktionen.

- Die Matrix R entspricht der Transponierten des Ergebnisses der Funktion `chol(matA)` wobei `matA` die genetisch additive Verwandtschaftsmatrix ist. Also können Sie die Matrix R mit folgenden Anweisungen in R berechnen.

```
library(pedigreeemm)
nNrAniInPedEx2 <- 8
pedEx2 <- pedigree(sire = c(NA, NA, 1, 1, 4, 5, 5, 6),
                  dam  = c(NA, NA, 2, 3, 3, 2, 3, 7),
                  label = as.character(1:nNrAniInPedEx2))
matA <- as.matrix(getA(ped = pedEx2))
matR <- t(chol(matA))
```

- Die Inzuchtkoeffizienten können mit der Funktion `inbreeding()` berechnet werden.

Aufgabe 2: Erweiterung der if-Bedingungen

In der letzten Übung hatten wir gesehen, wie der Programmablauf mit `if`-Bedingungen gesteuert wird. Im einfachsten Fall wird eine Bedingung überprüft und falls diese zutrifft, werden gewissen Anweisungen ausgeführt. Dieses Konzept kann auch erweitert werden. Wir können verschiedene Bedingungen gleichzeitig überprüfen und bei deren Zutreffen jeweils verschiedene Anweisungen ausführen.

Als Beispiel können wir bei den natürlichen Zahlen zwischen 1 und 10 überprüfen, ob diese durch 2, 3 oder 5 teilbar sind.

```

nLowerLimit <- 1
nUpperLimit <- 10
for (nIdx in nLowerLimit:nUpperLimit){
  if (identical(nIdx %% 2, 0)) {
    cat(nIdx, " ist durch 2 teilbar\n")
  } else if(identical(nIdx %% 3, 0)) {
    cat(nIdx, " ist durch 3 teilbar\n")
  } else if(identical(nIdx %% 5, 0)) {
    cat(nIdx, " ist durch 5 teilbar\n")
  } else {
    cat(nIdx, " ist weder durch 2 noch durch 3 noch durch 5 teilbar\n")
  }
}

```

```

## 1  ist weder durch 2 noch durch 3 noch durch 5 teilbar
## 2  ist durch 2 teilbar
## 3  ist durch 3 teilbar
## 4  ist durch 2 teilbar
## 5  ist durch 5 teilbar
## 6  ist durch 2 teilbar
## 7  ist weder durch 2 noch durch 3 noch durch 5 teilbar
## 8  ist durch 2 teilbar
## 9  ist durch 3 teilbar
## 10 ist durch 2 teilbar

```

Aufgrund des Resultats sehen wir klar, dass jede Bedingung nur einmal getestet wird. Dies erkennen wir aufgrund der Tatsache, dass für die Zahlen 6 und 10 nur die Teilbarkeit durch 2 aufgelistet wird. Falls keine der zu testenden Bedingungen in den runden Klammern nach den `if`-Anweisungen zutreffen wird der Teil nach der `else`-Anweisung ausgeführt.

Ihre Aufgabe

Finden Sie mit einer Kombination aus einem Loop und von verschiedenen `if`-Bedingungen die Zahlen, welche durch 2, 3 und 7 teilbar sind und alle Primzahlen zwischen 11 und 50. Für das Finden der Primzahlen reicht es, wenn Sie die Teilbarkeit bis zur Zahl 7 überprüfen. Ist eine Zahl nur durch 1 und sich selber teilbar, dann ist es eine Primzahl.