

Züchtungslehre - Lösung 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.

Lösung

- Inzuchtkoeffizienten: berechnen sich aus den Diagonalelementen a_{ii} der Verwandtschaftsmatrix. Diese werden aus den Elementen der i -ten Zeile der Matrix R aus (1) berechnet. Genauer gilt

$$a_{ii} = \sum_{j=1}^i r_{ij}^2$$

- Die Diagonalelemente r_{ii} der Matrix R berechnen sich als

$$r_{ii} = \sqrt{1 - 0.25(a_{ss} + a_{dd})}$$

wobei s und d die Eltern von i sind und a_{dd} (a_{ss}) dem Diagonalelement der Verwandtschaftsmatrix A entspricht, welches zu Tier d (s) gehört. Ist ein Elternteil unbekannt, dann ist das entsprechende Diagonalelement aus A gleich 0.

- Die Offdiagonalelemente r_{ij} (für $i \neq j$) entsprechen

$$r_{ij} = \frac{1}{2}(r_{sj} + r_{dj})$$

wobei s und d die Eltern von i sind. Für unbekannte Eltern d (s) sind die Werte r_{dj} (r_{sj}) gleich 0.

Werden die Elemente der Matrix R nach den soeben beschriebenen Regeln berechnet, erhalten wir das folgende Resultat.

$$R = \begin{bmatrix} 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 1.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.5000 & 0.5000 & 0.7071 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.7500 & 0.2500 & 0.3536 & 0.7071 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.6250 & 0.3750 & 0.5303 & 0.3536 & 0.6614 & 0.0000 & 0.0000 & 0.0000 \\ 0.3125 & 0.6875 & 0.2652 & 0.1768 & 0.3307 & 0.6374 & 0.0000 & 0.0000 \\ 0.5625 & 0.4375 & 0.6187 & 0.1768 & 0.3307 & 0.0000 & 0.6374 & 0.0000 \\ 0.4375 & 0.5625 & 0.4419 & 0.1768 & 0.3307 & 0.3187 & 0.3187 & 0.5863 \end{bmatrix}$$

Die Inzuchtkoeffizienten der Tiere sind in der folgenden Tabelle aufgelistet.

Tier	Inzuchtkoeffizient
1	0.0000
2	0.0000
3	0.0000
4	0.2500
5	0.3750
6	0.1875
7	0.4375
8	0.3906

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.

Lösung

```

nLowerLimit <- 11; nUpperLimit <- 50
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 if(identical(nIdx %% 7, 0)) {
    cat(nIdx, " ist durch 7 teilbar\n")
  } else {
    cat(nIdx, " ist eine Primzahl\n")
  }
}

```

```
## 11 ist eine Primzahl
## 12 ist durch 2 teilbar
## 13 ist eine Primzahl
## 14 ist durch 2 teilbar
## 15 ist durch 3 teilbar
## 16 ist durch 2 teilbar
## 17 ist eine Primzahl
## 18 ist durch 2 teilbar
## 19 ist eine Primzahl
## 20 ist durch 2 teilbar
## 21 ist durch 3 teilbar
## 22 ist durch 2 teilbar
## 23 ist eine Primzahl
## 24 ist durch 2 teilbar
## 25 ist durch 5 teilbar
## 26 ist durch 2 teilbar
## 27 ist durch 3 teilbar
## 28 ist durch 2 teilbar
## 29 ist eine Primzahl
## 30 ist durch 2 teilbar
## 31 ist eine Primzahl
## 32 ist durch 2 teilbar
## 33 ist durch 3 teilbar
## 34 ist durch 2 teilbar
## 35 ist durch 5 teilbar
## 36 ist durch 2 teilbar
## 37 ist eine Primzahl
## 38 ist durch 2 teilbar
## 39 ist durch 3 teilbar
## 40 ist durch 2 teilbar
## 41 ist eine Primzahl
## 42 ist durch 2 teilbar
## 43 ist eine Primzahl
## 44 ist durch 2 teilbar
## 45 ist durch 3 teilbar
## 46 ist durch 2 teilbar
## 47 ist eine Primzahl
## 48 ist durch 2 teilbar
## 49 ist durch 7 teilbar
## 50 ist durch 2 teilbar
```