

Züchtungslehre - Lösung 4 - Ergänzung zu Aufgabe 2

Peter von Rohr

November 6, 2015

Aufgabe 2 (8)

Es soll eine alternative Lösung für Aufgabe 2 in Übung 4 vorgestellt werden. In diesem alternativen Lösungsansatz wird auf die Verwendung von "if-else"-Statements verzichtet.

In der Aufgabe war folgender Code schon gegeben.

```
> # Definition des Pedigrees als data.frame
> Tier <- c("1" , "2" , "3" , "6" , "4", "5", "7", "8" , "9", "10")
> Mutter <- c("NA", "NA", "NA", "NA", "1", "1", "3", "NA", "5", "7")
> Vater <- c("NA", "NA", "NA", "NA", "2", "2", "4", "6" , "6", "8")
> dfPedigree <- data.frame(Tier, Mutter, Vater, stringsAsFactors = FALSE)
> # Anzeige des Pedigress
> print(dfPedigree)
```

	Tier	Mutter	Vater
1	1	NA	NA
2	2	NA	NA
3	3	NA	NA
4	6	NA	NA
5	4	1	2
6	5	1	2
7	7	3	4
8	8	NA	6
9	9	5	6
10	10	7	8

Die Verwandtschaftsmatrix wird wie folgt definiert und initialisiert.

```
> # Definition und Initialisierung der Verwandtschaftsmatrix
> matAddGenVer <- matrix(NA, nrow = (nrow(dfPedigree)+1),
+                          ncol = (nrow(dfPedigree)+1),
+                          dimnames = list(c("NA", dfPedigree[, "Tier"]),
```

```

+                                     c("NA", dfPedigree[, "Tier"])))
> matAddGenVer["NA",] <- 0
> matAddGenVer[, "NA"] <- 0

```

Die Diagonalelemente wurden schon berechnet. Neu gefragt war ein Loop innerhalb des gegebenen Loops, welcher für jede Zeile die Offdiagonal-elemente berechnet

```

> nAnzahlTiere <- nrow(dfPedigree)
> for (idx in 1:(nAnzahlTiere-1)) {
+   sTier <- dfPedigree[idx, "Tier"]
+   sMutter <- dfPedigree[idx, "Mutter"]
+   sVater <- dfPedigree[idx, "Vater"]
+   nInzuchtGrad <- 1/2 * matAddGenVer[sMutter, sVater]
+   # berechne Diagonalelement
+   matAddGenVer[sTier, sTier] <- 1 + nInzuchtGrad
+   # loop ueber Nicht-Diagonalelemente
+   # Bitte Ihren Code hier eintragen ...
+   # Berechnung der Off-Diagonalelemente bis zur zweitletzten Zeile
+   for (jdx in (idx+1):nAnzahlTiere){
+     sTierJ <- dfPedigree[jdx, "Tier"]
+     sMutterJ <- dfPedigree[jdx, "Mutter"]
+     sVaterJ <- dfPedigree[jdx, "Vater"]
+     matAddGenVer[sTier, sTierJ] <- 1/2*(matAddGenVer[sTier, sMutterJ]
+                                       + matAddGenVer[sTier, sVaterJ])
+     # Uebertrag direkt bei Berechnung
+     matAddGenVer[sTierJ, sTier] <- matAddGenVer[sTier, sTierJ]
+   }
+ }
> # Berechnung des letzten Diagonalelements
> sTier <- dfPedigree[nAnzahlTiere, "Tier"]
> sMutter <- dfPedigree[nAnzahlTiere, "Mutter"]
> sVater <- dfPedigree[nAnzahlTiere, "Vater"]
> nInzuchtGrad <- 1/2 * matAddGenVer[sMutter, sVater]
> # berechne Diagonalelement
> matAddGenVer[sTier, sTier] <- 1 + nInzuchtGrad
> # Elimination der Zeile und der Spalte fuer NA
> matAddGenVer <- matAddGenVer[2:nrow(matAddGenVer), 2:ncol(matAddGenVer)]
> print(matAddGenVer)

```

```

      1    2    3    6    4    5    7    8    9   10
1  1.000 0.000 0.00 0.00 0.50 0.500 0.250 0.00 0.2500 0.1250
2  0.000 1.000 0.00 0.00 0.50 0.500 0.250 0.00 0.2500 0.1250
3  0.000 0.000 1.00 0.00 0.00 0.000 0.500 0.00 0.0000 0.2500
6  0.000 0.000 0.00 1.00 0.00 0.000 0.000 0.50 0.5000 0.2500
4  0.500 0.500 0.00 0.00 1.00 0.500 0.500 0.00 0.2500 0.2500
5  0.500 0.500 0.00 0.00 0.50 1.000 0.250 0.00 0.5000 0.1250

```

```

7 0.250 0.250 0.50 0.00 0.50 0.250 1.000 0.00 0.1250 0.5000
8 0.000 0.000 0.00 0.50 0.00 0.000 0.000 1.00 0.2500 0.5000
9 0.250 0.250 0.00 0.50 0.25 0.500 0.125 0.25 1.0000 0.1875
10 0.125 0.125 0.25 0.25 0.25 0.125 0.500 0.50 0.1875 1.0000

```

Kontrolle mit pedigreeemm

Als Kontrolle stellen wir die additiv genetische Verwandtschaftsmatrix mit Funktion `getA()` aus dem R-Package “pedigreeemm” auf.

```

> library(pedigreeemm)
> pedA1 <- pedigree(sire = as.numeric(dfPedigree$Vater),
+                  dam = as.numeric(dfPedigree$Mutter),
+                  label = as.numeric(dfPedigree$Tier))
> spmVerwPedMM <- getA(pedA1)
> matVerwPedMM <- as.matrix(spmVerwPedMM)
> print(matVerwPedMM)

```

	1	2	3	6	4	5	7	8	9	10
1	1.000	0.000	0.00	0.00	0.50	0.500	0.250	0.00	0.2500	0.1250
2	0.000	1.000	0.00	0.00	0.50	0.500	0.250	0.00	0.2500	0.1250
3	0.000	0.000	1.00	0.00	0.00	0.000	0.500	0.00	0.0000	0.2500
6	0.000	0.000	0.00	1.00	0.00	0.000	0.000	0.50	0.5000	0.2500
4	0.500	0.500	0.00	0.00	1.00	0.500	0.500	0.00	0.2500	0.2500
5	0.500	0.500	0.00	0.00	0.50	1.000	0.250	0.00	0.5000	0.1250
7	0.250	0.250	0.50	0.00	0.50	0.250	1.000	0.00	0.1250	0.5000
8	0.000	0.000	0.00	0.50	0.00	0.000	0.000	1.00	0.2500	0.5000
9	0.250	0.250	0.00	0.50	0.25	0.500	0.125	0.25	1.0000	0.1875
10	0.125	0.125	0.25	0.25	0.25	0.125	0.500	0.50	0.1875	1.0000

Als Kontrolle auf einen Blick berechnen wir das Maximum der Differenz zwischen den beiden Matrizen.

```

> max(matAddGenVer-matVerwPedMM)
[1] 1.110223e-16

```