Züchtungslehre - Lösung 2

Peter von Rohr

2017-10-06

Aufgabe 1: Matrixdefinitionen in R

In R werden Matrizen mit der Funktion `matrix` erstellt. In der Vorlesung hatten wir gesehen, dass die Funktion `matrix()` verschiedene Optionen aktzeptiert. Wir wollen uns hier anschauen, wie sich die Parameter auswirken.

Ihre Aufgabe wird es sein die Matrizen mit den verschiedenen Optionen zu erstellen und so besser zu verstehen, was die Optionen für eine Bedeutung haben.

Parameter `data`

- `data`: Angabe der Matrix-Elemente

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

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

- `data`: Ohne Angabe der Matrix-Elemente

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

```
##      [,1] [,2] [,3]
## [1,]   NA   NA   NA
## [2,]   NA   NA   NA
## [3,]   NA   NA   NA
```

- `data`: Spezifikation nicht aller Elemente

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

```
##      [,1] [,2] [,3]
## [1,]    1    1    1
## [2,]    2    2    2
## [3,]    3    3    3
```

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

```
## Warning in matrix(data = c(1, 2, 3, 4), nrow = 3, ncol = 3): data length
## [4] is not a sub-multiple or multiple of the number of rows [3]
```

```
##      [,1] [,2] [,3]
## [1,]    1    4    3
## [2,]    2    1    4
## [3,]    3    2    1
```

Parameter `nrow` und `ncol`

- Weglassen einer der beiden Parameter

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

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

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

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

Parameter `byrow`

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

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

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

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

Aufgabe 2: Matrixmultiplikation in R

Matrixmultiplikationen können in R mit dem Operator `%*%` oder mit den Funktionen `crossprod()` oder `tcrossprod()` ausgeführt werden. Der Vorteil von `crossprod()` und `tcrossprod()` gegenüber von `%*%` ist, dass wir mit `crossprod()` und `tcrossprod()` direkt Matrizen und Vektoren multiplizieren können. Das funktioniert mit `%*%` nicht. Bei der Matrix-Vektor-Multiplikation mit `%*%` muss der Vektor zuerst in eine Matrix verwandelt werden.

In einem ersten Teil der Aufgabe geht es um einen Vergleich zwischen `crossprod()`, `tcrossprod()` und `%*%` für die Matrix-Matrix-Multiplikation.

a) Gegeben sind die folgenden Matrizen

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

Finden Sie heraus welcher Multiplikationen mit `%*%` entspricht die folgende Anweisung?

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

```
##      [,1] [,2] [,3]
## [1,]   20   38   56
## [2,]   47   92  137
## [3,]   74  146  218
```

Lösung

Die Anweisung `crossprod(matA,matB)` entspricht der Matrixmultiplikation

```r
t(matA) %*% matB
```

```
##      [,1] [,2] [,3]
## [1,]   20   38   56
## [2,]   47   92  137
## [3,]   74  146  218
```

Alternativ dazu gibt es die Funktion `tcrossprod()`. Finden Sie, welche Matrixmultiplikation mit `%*%`

```r
tcrossprod(matA, matB)
```

```
##      [,1] [,2] [,3]
## [1,]   78   90  102
## [2,]   93  108  123
## [3,]  108  126  144
```

ausführt.

Lösung

```r
matA %*% t(matB)
```

```
##      [,1] [,2] [,3]
## [1,]   78   90  102
## [2,]   93  108  123
## [3,]  108  126  144
```

b) Gegeben ist zusätzlich der Vektor `vecB` als

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

Multiplizieren Sie die Matrix `matA` mit dem Vektor `vecB` einmal mit `%*%` und einmal mit `crossprod()`.

Hinweise: Ein Vektor kann mit der Funktion `as.matrix()` in eine Matrix verwandelt werden.

Lösung

```r
matA %*% as.matrix(vecB)
```

```
##      [,1]
## [1,]   68
## [2,]   82
## [3,]   96
```

```r
crossprod(t(matA), vecB)
```

```
##      [,1]
## [1,]   68
## [2,]   82
## [3,]   96
```

Aufgabe 3: Gleichungssysteme

Gegeben ist das folgende Gleichungssystem.

$$\begin{aligned} 2x_2 + 2x_3 &= 1 \\ 2x_1 + 4x_2 + 5x_3 &= 9 \\ x_1 - x_2 + 2x_3 &= 3 \end{aligned} \tag{1}$$

a) Bestimmen Sie die Lösungsmenge des Gleichungssystems (1) mit dem Gaussverfahren

Lösung

- Vertauschen der ersten und der zweiten Gleichung

$$\begin{aligned} 2x_1 + 4x_2 + 5x_3 &= 9 \\ 2x_2 + 2x_3 &= 1 \\ x_1 - x_2 + 2x_3 &= 3 \end{aligned}$$

- 1/2-fache der ersten Gleichung von dritter abziehen

$$\begin{aligned} 2x_1 + 4x_2 + 5x_3 &= 9 \\ 2x_2 + 2x_3 &= 1 \\ -3x_2 - \frac{1}{2}x_3 &= -\frac{3}{2} \end{aligned}$$

- Addition des 3/2-fache der zweiten zur dritten Gleichung

$$\begin{aligned} 2x_1 + 4x_2 + 5x_3 &= 9 \\ 2x_2 + 2x_3 &= 1 \\ \frac{5}{2}x_3 &= 0 \end{aligned}$$

Somit ist $x_3 = 0$.

- Rückwärtseinsetzen in der zweiten Gleichung führt zu $x_2 = 1/2$. Aufgrund der ersten Gleichung folgt $x_1 = 7/2$.

b) Verwandeln Sie das Gleichungssystem (1) in Matrix-Vektor-Schreibweise

Lösung

$$A \cdot x = b$$

wobei die sogenannte Koeffizientenmatrix A, der Vektor x und die rechte Handseite b wie folgt definiert sind

$$A = \begin{bmatrix} 0 & 2 & 2 \\ 2 & 4 & 5 \\ 1 & -1 & 2 \end{bmatrix}$$

,

$$x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

und

$$b = \begin{bmatrix} 1 \\ 9 \\ 3 \end{bmatrix}$$

c) Wie lautet die Lösung des Gleichungssystem (1) in Matrix-Vektor-Schreibweise

$$x = A^{-1} \cdot b$$

d) Berechnen Sie die Lösung aus c) mit R

Hinweis * Für die Multiplikation der Matrix A^{-1} mit dem Vektor b muss der Vektor b in eine Matrix verwandelt werden. Dies kann mit der Funktion `as.matrix()` gemacht werden.

Lösung

```
matA <- matrix(data = c(0,2,2,2,4,5,1,-1,2),nrow = 3,ncol = 3, byrow = TRUE)
matAInv <- solve(matA)
vecB <- c(1,9,3)
sol <- matAInv %*% as.matrix(vecB)
print(sol)
```

```
##              [,1]
## [1,] 3.500000e+00
## [2,] 5.000000e-01
## [3,] 2.220446e-16
```

Wir erhalten die Lösungen des Gleichungssystems auch mit einem einzigen Aufruf der Funktion `solve()`.

```
solve(matA, vecB)
```

```
## [1] 3.5 0.5 0.0
```

Aufgabe 4: Quantitative Genetik

In einer Population wurden für einen Genort folgende Häufigkeiten bei Genotypen gezählt

Genotypen	Anzahl
A_1A_1	24
A_1A_2	53
A_2A_2	23

a) Bestimmen Sie die Genotypfrequenzen

Lösung

```
nTotNrInd <- sum(dfGenotypeFreq$Anzahl)
vGenoTypeFreq <- dfGenotypeFreq$Anzahl / nTotNrInd
cat(paste("Genotyp-Frequenz", dfGenotypeFreq$Genotypen[1]), ": ", vGenoTypeFreq[1])
```

```
## Genotyp-Frequenz $A_1A_1$ :  0.24
```

```
cat(paste("Genotyp-Frequenz", dfGenotypeFreq$Genotypen[2]), ": ", vGenoTypeFreq[2])
```

```
## Genotyp-Frequenz $A_1A_2$ :  0.53
```

```
cat(paste("Genotyp-Frequenz", dfGenotypeFreq$Genotypen[3]), ": ", vGenoTypeFreq[3])
```

```
## Genotyp-Frequenz $A_2A_2$ :  0.23
```

b) Bestimmen Sie die Allelfrequenzen

Lösung

```
vGenFreqP <- vGenoTypeFreq[1] + 0.5*vGenoTypeFreq[2]
vGenFreqQ <-  vGenoTypeFreq[3] + 0.5*vGenoTypeFreq[2]
cat("Allelfrequenz fuer A1: ", vGenFreqP)
```

```
## Allelfrequenz fuer A1:  0.505
```

```
cat("Allelfrequenz fuer A2: ", vGenFreqQ)
```

```
## Allelfrequenz fuer A2:  0.495
```

c) Berechnen Sie das Populationsmittel μ unter der Annahme, dass die genotypischen Werte zwischen den homozygoten Genotypen 20 Einheiten auseinanderliegen und dass der heterozygote Genotyp einen genotypischen Wert von 2 hat.

Lösung

```
nDeltaHom <- 20
### # additiver Wert A
nAddValue <- nDeltaHom / 2
nDom <- 2
### # Populationsmittel
nMu <- (vGenFreqP-vGenFreqQ) * nAddValue + 2 * vGenFreqP * vGenFreqQ * nDom
cat("Populationsmittel: ", nMu, "\n")
```

```
## Populationsmittel:  1.0999
```