(10) Dimensionsreduktion

Ziel dieser Sitzung ist es, die Hauptkomponentenanalyse eines simulierten Beispieldatensatzes auf der Implementationsebene nachzuvollziehen und die resultierenden Matrizen mithilfe von Colorplots darzustellen.

Datensatz generieren

Untenstehender **R**-Code erzeugt dazu zunächst einen Datensatz von n unabhängigen Beobachtungen ($n = 20$) eines m-dimensionalen multivariat normalverteilten Zufallsvektors ($m = 5$).

```
# Datensatzsimulation
library(MASS)                                          # multivariate Normalverteilung
set.seed(1)                                            # reproduzierbare Ergebnisse
m     = 5                                              # Datendimension
n     = 20                                             # Anzahl Realisierungen
mu    = rep(0,m)                                       # Erwartungswertparameter
Sigma = matrix(runif(m^2), nrow = m)                   # zufällige Matrix
Sigma = 0.5*(Sigma+t(Sigma))                           # symmetrische Matrix
Sigma = Sigma + m*diag(m)                              # positiv-definite Matrix
X     = t(mvrnorm(n,mu,Sigma))                         # Datensatzgenerierung
```

Anschließend lassen wir uns diesen Datensatz im Datenmatrixformat – mit Beobachtungen in Zeilen und Variablen bzw. Features in Spalten – ausgeben.

```
print(t(X))                                            # Ausgabe der Datenmatrix
```

```
            [,1]         [,2]        [,3]       [,4]       [,5]
 [1,] -2.09385746 -0.070707633  1.58116613  1.0629970  1.3041640
 [2,]  1.26748437  2.027510977  0.10231314 -1.1958062 -0.4543732
 [3,]  1.29412379  3.219195870 -1.04592641 -1.2764435  0.4141476
 [4,]  0.36723722  1.220567789 -2.56363564 -0.1226581 -0.4681759
 [5,]  2.70591708 -0.004982683  3.03837681  0.1090709 -0.4901049
 [6,] -3.22327265  2.205721892  1.42878907  0.1729683 -4.6650851
 [7,]  0.85203288  0.540485646  4.56771372  2.8617633 -2.0846905
 [8,]  2.04629454  2.311265206 -1.58286877  1.1664413 -3.2961878
 [9,] -1.30642123  1.687464574 -1.74007204 -1.9419141  1.2853801
[10,] -2.32263088  0.392919525  1.19835462 -1.2518442  1.2991077
[11,] -1.20745738 -2.648582544  0.08490397 -0.8138305 -0.2387816
[12,] -2.84848640 -1.431453394  1.69914214  1.4453424  1.5677502
[13,]  0.31562707 -3.198738090  0.64285102 -0.4258388 -0.2442774
[14,] -5.19519795 -2.151483293  1.37434634 -2.5357738  2.2697941
[15,]  3.70371430  2.004009760  0.34281686  0.1885251 -2.4783149
[16,]  1.44283270  4.142453070 -0.01419740  0.1314694  2.4989070
[17,]  2.33939815 -0.543357707  1.38631602 -2.5359563 -0.8465429
[18,] -0.08888899  1.890898552 -0.36175302 -1.5061357  1.8963343
[19,] -0.45977955 -0.450246299  0.03511706  3.0580989  1.2037480
[20,]  0.03237427  5.270465900 -1.22698235  0.4629124 -2.7957205
```

Hauptkomponentenanalyse

Folgender **R**-Code führt dann mithilfe der Methoden der Eigenanalyse eine Hauptkomponentenanalyse des Datensatzes durch.

```
# Hauptkomponentenanalyse durch Eigenanalyse
I_n     = diag(n)                                                  # Einheitsmatrix I_n
J_n     = matrix(rep(1,n^2), nrow = n)                             # 1_{nn}
C       = (1/(n-1))*(X %*% (I_n-(1/n)*J_n) %*% t(X))               # Stichprobenkovarianzmatrix X
D       = diag(1/sqrt(diag(C)))                                    # Kov-Korr-Transformationsmatrix
R       = D %*% C %*% D                                            # Stichprobenkorrelationsmatrix X
EA      = eigen(C)                                                 # Eigenanalyse von C
lambda  = EA$values                                                # Eigenwerte von C
Q       = EA$vectors                                               # Eigenvektoren von C
X_tilde = t(Q) %*% X                                               # transformierter Datensatz
C_tilde = (1/(n-1))*(X_tilde %*% (I_n-(1/n)*J_n) %*% t(X_tilde)) # Stichprobenkovarianzmatrix \tilde{X}
D_tilde = diag(1/sqrt(diag(C_tilde)))                              # Kov-Korr-Transformationsmatrix
R_tilde = D_tilde %*% C_tilde %*% D_tilde                          # Stichprobenkorrelationsmatrix \tilde{X}
```

Ergebnisse visualisieren

Folgender **R**-Code visualisiert die Datensatzmatrix X und die PCA-transformierte Datensatzmatrix $\tilde{X}$, wie in Abbildung 1 gezeigt wird.

```
# R-Pakete
library(latex2exp)
library(plot.matrix)

# Abbildungsparameter
par(
    family      = "sans",
    mfcol       = c(2,1),
    pty         = "m",
    bty         = "l",
    lwd         = 1,
    las         = 1,
    mgp         = c(2,1,0),
    xaxs        = "i",
    yaxs        = "i",
    font.main   = 1,
    cex         = 1,
    cex.main    = 2)

# Visualisierung X
plot(X,
    breaks      = c(-5,5),
    col         = topo.colors,
    fmt.key     = "%.0f",
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$X$"))

# Visualisierung \tilde{X}
plot(X_tilde,
```

```
    breaks      = c(-5,5),
    col         = topo.colors,
    fmt.key     = "%.0f",
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$\\tilde{X}$"))

# PDF-Speicherung
dev.copy2pdf(
    file        = "./Abbildungen/X_tildeX.pdf",
    width       = 10,
    height      = 7)
dev.off()
```

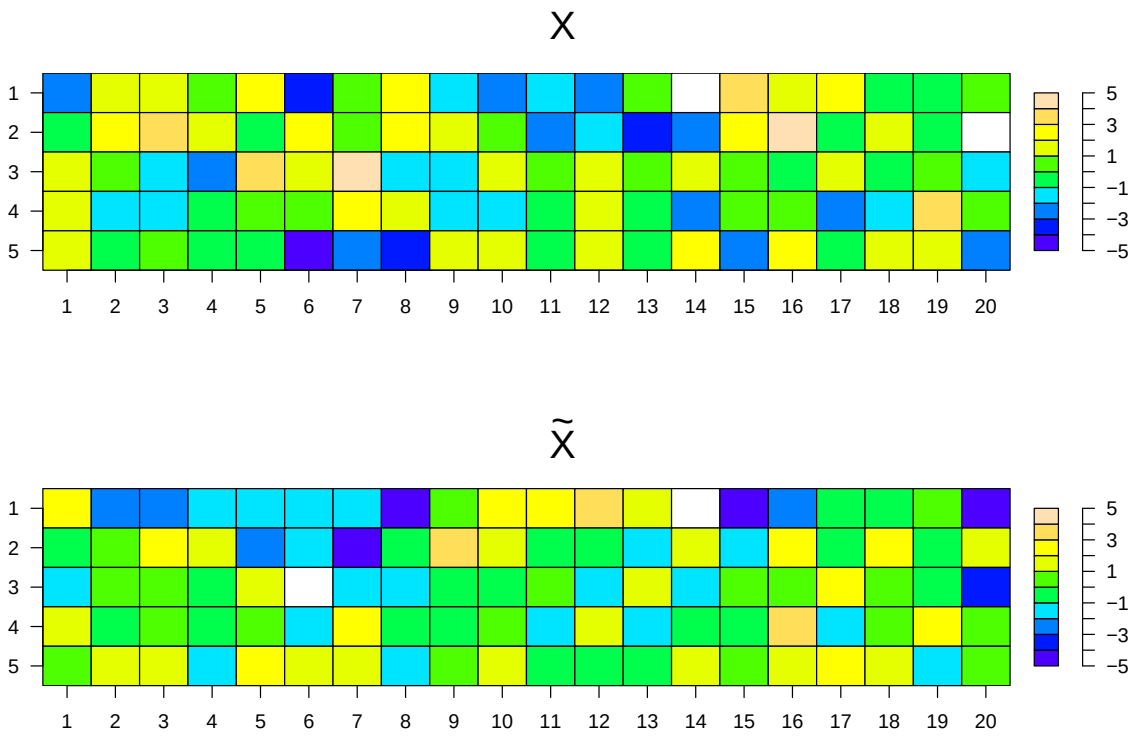

Abbildung 1. Datensatz X und PCA-transformierter Datensatz $\tilde{X}$.

Folgender **R**-Code schließlich visualisiert die zentralen Matrizen der Hauptkomponentenanalyse des Datensatzes (Q und Λ, C und $\tilde{C}$, R und $\tilde{R}$), wie in Abbildung 2 gezeigt wird.

```
# Matrizen
C_tilde[C_tilde < 0.001]      = 0                  # \tilde{C}: kleine Werte auf 0 setzen
R_tilde[abs(C_tilde) < 0.001] = 0                  # \tilde{R}: kleine Werte auf 0 setzen

# Abbildungsparameter
par(
    family      = "sans",
    mfcol       = c(2,3),
    pty         = "m",
    bty         = "l",
    lwd         = 1,
    las         = 1,
    mgp         = c(2,1,0),
    xaxs        = "i",
    yaxs        = "i",
    font.main   = 1,
    cex         = 1,
    cex.main    = 2)

# Q und \Lambda
plot(Q,
    col         = topo.colors,
    digits      = 2,
    key         = NULL,
    cex         = 0.8,
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$Q$"))
plot(diag(lambda),
    col         = topo.colors,
    digits      = 2,
    key         = NULL,
    cex         = 0.8,
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$\\Lambda$"))

# C und \tilde{C}
plot(C,
    col         = topo.colors,
    digits      = 2,
    key         = NULL,
    cex         = 0.8,
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$C$"))
plot(C_tilde,
    col         = topo.colors,
    digits      = 2,
    key         = NULL,
    cex         = 0.8,
```

```
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$\\tilde{C}$"))

# R und \tilde{R}
plot(R,
    col         = topo.colors,
    digits      = 2,
    key         = NULL,
    cex         = 0.8,
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$R$"))
plot(R_tilde,
    col         = topo.colors,
    digits      = 2,
    key         = NULL,
    cex         = 0.8,
    polygon.key = NULL,
    axis.key    = NULL,
    xlab        = "",
    ylab        = "",
    main        = TeX("$\\tilde{R}$"))

# PDF-Speicherung
dev.copy2pdf(
    file        = "./Abbildungen/Q_Lambda_C.pdf",
    width       = 11,
    height      = 8)
dev.off()
```

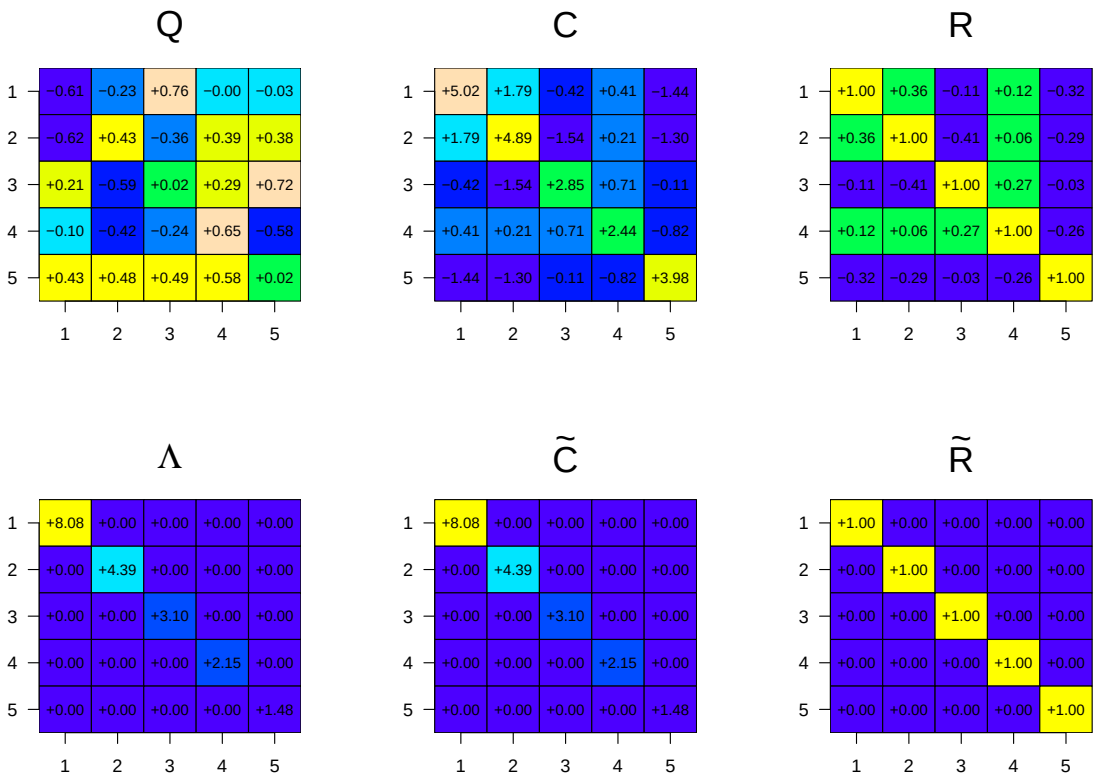

Abbildung 2. Zentrale Matrizen der Hauptkomponentenanalyse.