Blatt 7

Christian Peters

H14)

a)

Lese die Daten ein, zentriere sie und berechne die empirische Kovarianzmatrix:

```
data <- read.csv('AufgabeH14.txt')
data <- scale(data, center = TRUE, scale = FALSE)
(S <- cov(data))
```

```
##               Punkte       Tore  Gegentore    Karten
## Punkte     56.852941  39.970588 -41.970588 -4.617647
## Tore       39.970588  37.702614 -21.591503 -7.624183
## Gegentore -41.970588 -21.591503  46.879085  6.552288
## Karten     -4.617647  -7.624183   6.552288 25.192810
```

Die Hauptkomponenten ergeben sich aus den normierten Eigenvektoren der empirischen Kovarianzmatrix. Die zugehörigen Koeffizientenvektoren befinden sich in den Spalten der Matrix, die von der Funktion *eigen* als Resultat zurückgegeben wird:

```
eigen_results <- eigen(S)
principal_components <- eigen_results$vectors
rownames(principal_components) <- colnames(data)
colnames(principal_components) <- c('HK1', 'HK2', 'HK3', 'HK4')
principal_components
```

```
##                  HK1        HK2        HK3        HK4
## Punkte    -0.6776585 -0.1617785 -0.1512833  0.7012275
## Tore      -0.4832710  0.1647439 -0.6453017 -0.5682380
## Gegentore  0.5433172  0.1391911 -0.7243746  0.4008912
## Karten     0.1097018 -0.9629709 -0.1896855 -0.1570730
```

b)

Kriterium der totalen Variabilität:

```
variances <- eigen_results$values
(p_total <- min(which(cumsum(variances) / sum(variances) >= 0.75)))
```

```
## [1] 2
```

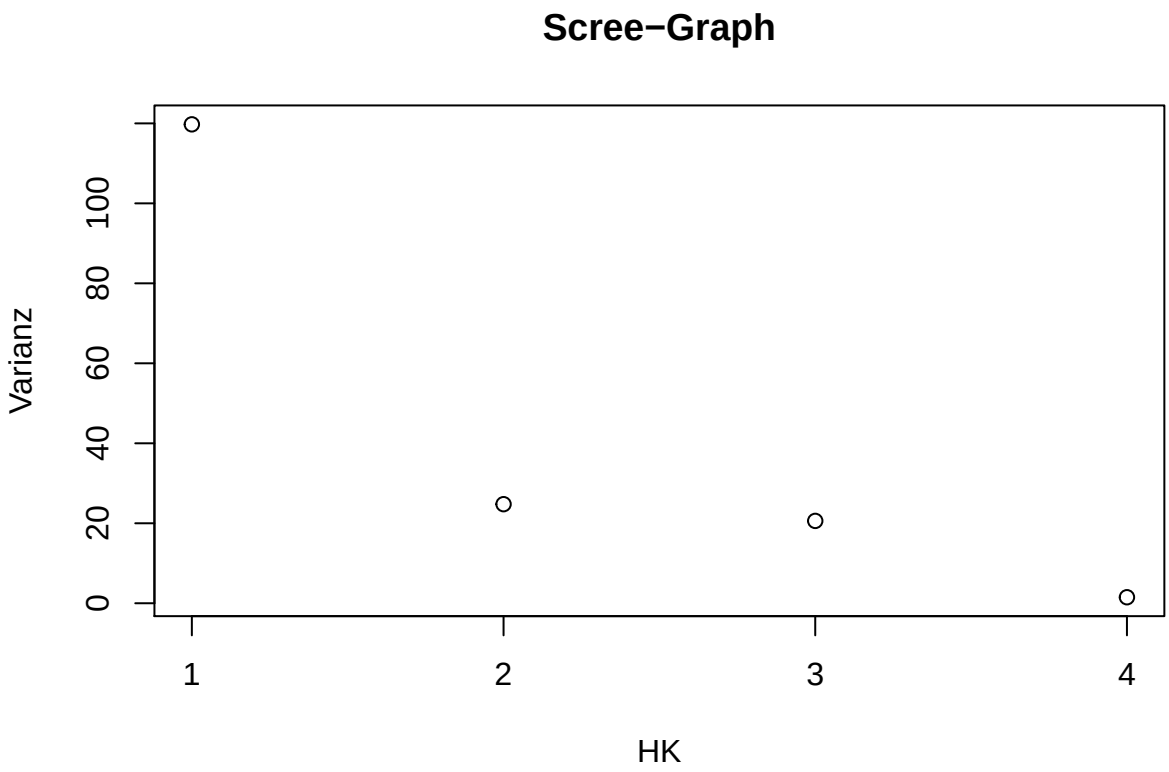
Kriterium der mittleren Variabilität:

```
(p_mean <- max(which(variances > mean(variances))))
```

```
## [1] 1
```

Scree-Graph:

```
plot(variances, main = "Scree-Graph", xlab = 'HK', ylab = 'Varianz', xaxt = "n")
axis(1, at = 1:4)
```



Anhand des Scree-Graphen würde man sich hier nur für die erste Hauptkomponente entscheiden.

c)

Durch die erste Hauptkomponente lassen sich Mannschaften, die viele Punkte und viele Tore erzielen von Mannschaften trennen, die viele Gegentore kassieren. Das Attribut "Karten" läd nur schwach auf HK1 und sollte daher bei der Interpretation nicht überbewertet werden.

Mit HK2 allerdings lassen sich Mannschaften, die viele Karten kassieren von Mannschaften trennen, die nur wenige Karten kassieren. Hieran lassen sich "faire" Teams von weniger fairen Teams unterscheiden. Die anderen Attribute laden nur schwach auf HK2, daher sollte man sie auch hier nicht überinterpretieren.

d)

Kommunalitäten der Variablen mit der ersten Hauptkomponente:

```
H1 <- diag(variances[1] * principal_components[, 1] %*% t(principal_components[, 1]))
H1 <- rbind(H1, H1 / diag(S)) # percentages
colnames(H1) <- colnames(data)
rownames(H1) <- c('Kommunalitaeten', 'Anteil')
H1
```

```
##                    Punkte       Tore  Gegentore     Karten
## Kommunalitaeten 54.994304 27.9690231 35.3510898 1.44119583
## Anteil           0.967308  0.7418325  0.7540909 0.05720663
```

Kommunalitäten der Variablen mit den ersten beiden Hauptkomponenten:

```
H2 <- diag(principal_components[, 1:2] %*% diag(variances[1:2]) %*%
             t(principal_components[, 1:2]))
H2 <- rbind(H2, H2 / diag(S))
colnames(H2) <- colnames(data)
rownames(H2) <- c('Kommunalitaeten', 'Anteil')
H2
```

```
##                    Punkte       Tore  Gegentore     Karten
## Kommunalitaeten 55.6427035 28.6414108 35.8310710 24.4147211
## Anteil           0.9787128  0.7596664  0.7643296  0.9691146
```

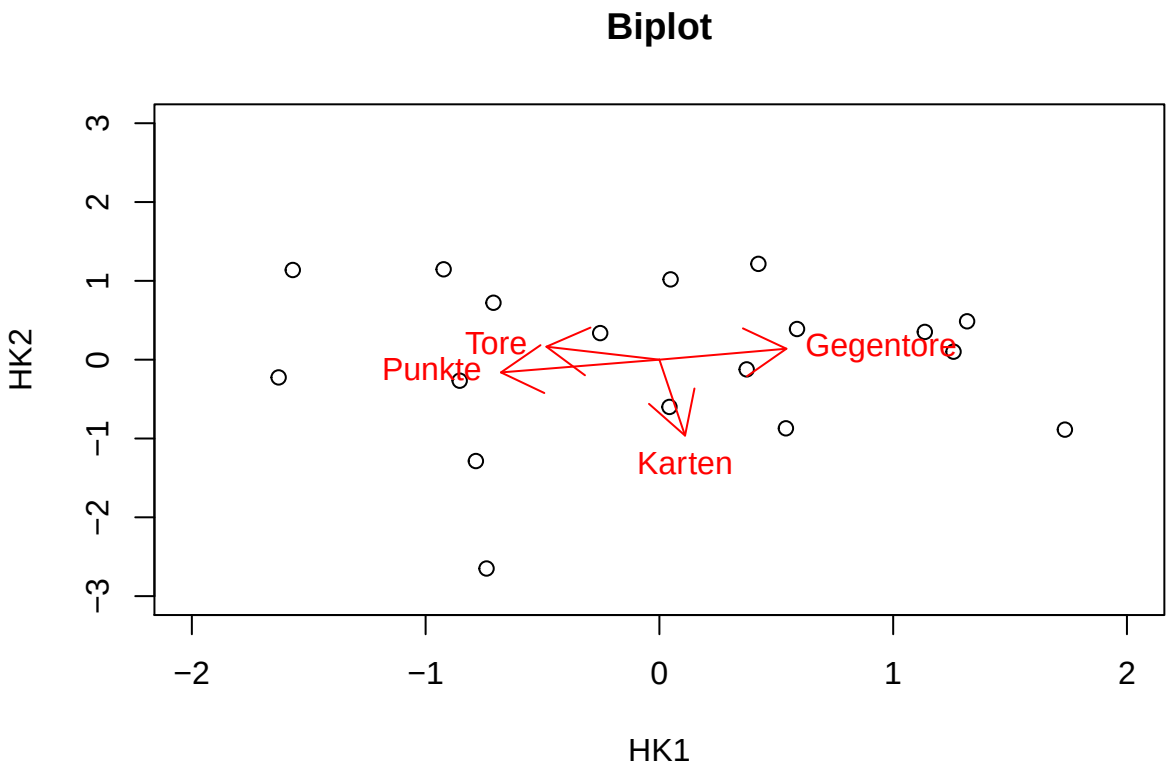
e)

Transformiere die Daten zunächst in den Raum der ersten beiden Hauptkomponenten und skaliere sie anhand der Standardabweichung entlang der jeweiligen Hauptkomponente:

```
data_transformed <- t(diag(1/sqrt(variances[1:2])) %*% t(principal_components[, 1:2]) %*%
                        t(data))
```

Erzeuge nun anhand dieser Daten den Biplot:

```
plot(data_transformed, xlim = c(-2, 2), ylim = c(-3, 3),  xlab = 'HK1', ylab = 'HK2',
     main = 'Biplot')
arrows(0, 0, principal_components[, 1], principal_components[, 2], col = 'red')
text(principal_components[, 1], principal_components[, 2], labels = colnames(data),
     col = 'red', pos = c(2, 2, 4, 1))
```



In diesem Biplot ist keine klare Trennung der Daten in Gruppen zu erkennen. Dies deutet darauf hin, dass

die Daten im Raum der ersten beiden Hauptkomponenten recht gleichmäßig verteilt sind. Da HK1 die Mannschaften mit vielen Toren und Punkten von denen mit vielen Gegentoren trennt, bedeutet das in etwa, dass es ungefähr gleich viele "gute" wie "schlechte" Teams gibt. Für HK2 sind hauptsächlich die Karten ausschlaggebend. Hier erkennt man auch keine Clusterbildung.

H16)

Lese zunächst die Daten ein:

```
(R <- matrix(c(1, 0.83, 0.78, 0.83, 1, 0.67, 0.78, 0.67, 1), nrow = 3))
```

```
##      [,1] [,2] [,3]
## [1,] 1.00 0.83 0.78
## [2,] 0.83 1.00 0.67
## [3,] 0.78 0.67 1.00
```

Im Folgenden wird die Hauptkomponentenmethode verwendet um zu einer Approximation $\hat{R} = LL' + V$ zu gelangen.

Erste Iteration:

```
eigen_res <- eigen(R)
L <- sqrt(eigen_res$values[1]) * eigen_res$vectors[, 1]
V <- diag(diag(R - L %*% t(L)))
L
```

```
## [1] -0.9514150 -0.9098748 -0.8881720
```

```
V
```

```
##            [,1]      [,2]      [,3]
## [1,] 0.09480941 0.0000000 0.0000000
## [2,] 0.00000000 0.1721278 0.0000000
## [3,] 0.00000000 0.0000000 0.2111505
```

Zweite Iteration:

```
eigen_res <- eigen(R - V)
L <- sqrt(eigen_res$values[1]) * eigen_res$vectors[, 1]
V_old <- V
V <- diag(diag(R - L %*% t(L)))
L
```

```
## [1] -0.9457019 -0.8758364 -0.8395906
```

```
V
```

```
##           [,1]      [,2]      [,3]
## [1,] 0.1056479 0.0000000 0.0000000
## [2,] 0.0000000 0.2329106 0.0000000
## [3,] 0.0000000 0.0000000 0.2950876
```

Iteriere nun solange, bis sich die Matrix V stabilisiert:

```
iterations <- 2
while(norm(V - V_old, type = 'F') > 1e-6){
  eigen_res <- eigen(R - V)
  L <- sqrt(eigen_res$values[1]) * eigen_res$vectors[, 1]
  V_old <- V
  V <- diag(diag(R - L %*% t(L)))
```

```
  iterations <- iterations + 1
}
L
```

```
## [1] 0.9829885 0.8443635 0.7934980
```

```
V
```

```
##             [,1]      [,2]      [,3]
## [1,] 0.03373358 0.0000000 0.0000000
## [2,] 0.00000000 0.2870502 0.0000000
## [3,] 0.00000000 0.0000000 0.3703609
```

```
iterations
```

```
## [1] 39
```

Probe zum Vergleich mit der ursprünglichen Korrelationsmatrix:

```
L %*% t(L) + V
```

```
##           [,1]      [,2]      [,3]
## [1,] 1.0000000 0.8299996 0.7799995
## [2,] 0.8299996 1.0000000 0.6700008
## [3,] 0.7799995 0.6700008 1.0000000
```

Man erkennt also, dass sich das gewählte Modell mit einem Faktor recht gut zur Beschreibung der Daten eignet. Das hier gefundene L ist nur bis auf orthogonale Transformationen (Rotationen) eindeutig.