```r
set.seed(42)
```

Arbeitsblatt 7

Aufgabe 1

Wir haben 100 Personen mit jeweils 1000 Features. Die Hauptkomponenten sind Linearkombinationen der Features, um neue Featurevektoren zu bestimmen. Wenn man versucht möglichst hohe Varianz zu erzielen, dann bieten sich orthogonal zueinander stehende Vektoren an, da dies die maximal mögliche Abweichung ist. Das einzige Vektorsystem einer Matrix, das orthogonal aufeinander steht sind die Eigenvektoren der Matrix, welche dann als Hauptkomponenten (anhand der Kovarianzmatrix) gewählt werden und anschließend nach Eigenwert absteigend sortiert werden. Je größer der Eigenwert, desto größer die erklärte Varianz.

Sprich in diesem Beispiel erklärt die erste Hauptkomponente als Linearkombination aus 1000 Featurevektoren alleine schon 10% der Variation.

Aufgabe 2

```r
# create matrix
data = c(
  1, 0.2, 0.3, 0.4, .4, 0.05,
  0.2, 1, 0.45, .3, .15, .2,
  .3, .45, 1, .1, .2, .4,
  .4, .3, .1, 1, .15, .4,
  .4, .15, .2, .15, 1, .05,
  .05, .2, .4, .4, .05, 1
)
K = matrix(
  nrow=6,
  ncol=6,
  data = data
)

# calculate eigenvectors, eigenvalues
eig = eigen(K)

# norm to exercise
vmat = round(eig$vectors, 2)
v1 = vmat[,1]
v2 = vmat[,2]
v3 = vmat[,3]
v4 = vmat[,4]
v5 = vmat[,5]
v6 = vmat[,6]
lmat = eig$values
l1 = lmat[1]
l2 = lmat[2]
l3 = lmat[3]
l4 = lmat[4]
l5 = lmat[5]
l6 = lmat[6]

# data
```

```
X = c(2,1,8,6,0,7)

# PCA
# calculate <X, v>
pca = X %*% vmat
pcavec = pca[1,]
```

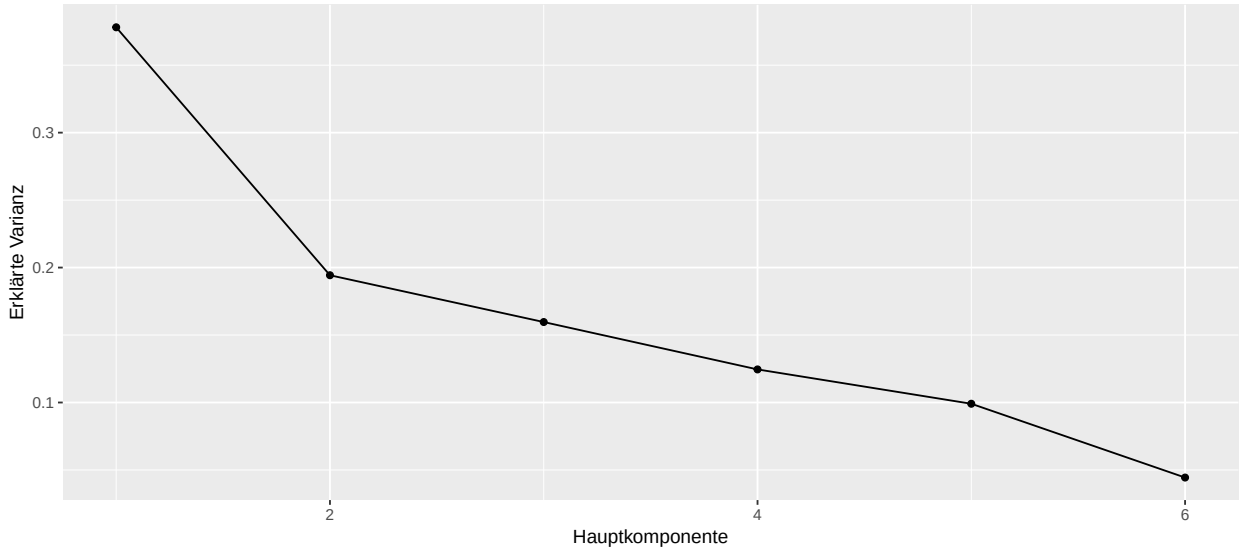
Wir erhalten als Hauptkomponenten für X: -10.21, -5.11, -1.41, 2.97, 2.63, -2.85

```
lsum = sum(lmat)
p1 = round(l1/lsum, 2)
p2 = round(l2/lsum, 2)
```

Die erste Hauptkomponente erklärt 0.38 und die zweite 0.19 der Varianz.

Der Scree Plot ist ein Scatterplot der geordneten Eigenwerte.

```
packageTest('ggplot2')
data = data.frame(lmat/sum(lmat))
gg = ggplot(
  data = data,
  mapping = aes(
    x = seq(6),
    y = data$lmat
  )
)
gg = gg + xlab('Hauptkomponente')
gg = gg + ylab('Erklärte Varianz')
gg + geom_point() + geom_line()
```



Aufgabe 3

a)

```
packageTest('datasets')
data <- USArrests
head(data)
```

```
#>            Murder Assault UrbanPop Rape
#> Alabama      13.2     236       58 21.2
#> Alaska       10.0     263       48 44.5
#> Arizona       8.1     294       80 31.0
#> Arkansas      8.8     190       50 19.5
#> California    9.0     276       91 40.6
#> Colorado      7.9     204       78 38.7
```

Laut Doku:

Dieser Datensatz enthält Statistiken, in Festnahmen pro 100.000 Einwohner, wegen Körperverletzung, Mord und Vergewaltigung in jedem der 50 US-Bundesstaaten für das Jahr 1973. Ebenfalls angegeben ist der Prozentsatz der Bevölkerung, der in städtischen Gebieten lebt.

Format:

Ein Dataframe mit 50 Beobachtungen und 4 Variablen.

[,1] Murder (numerisch): Mord Verhaftungen (pro 100.000)

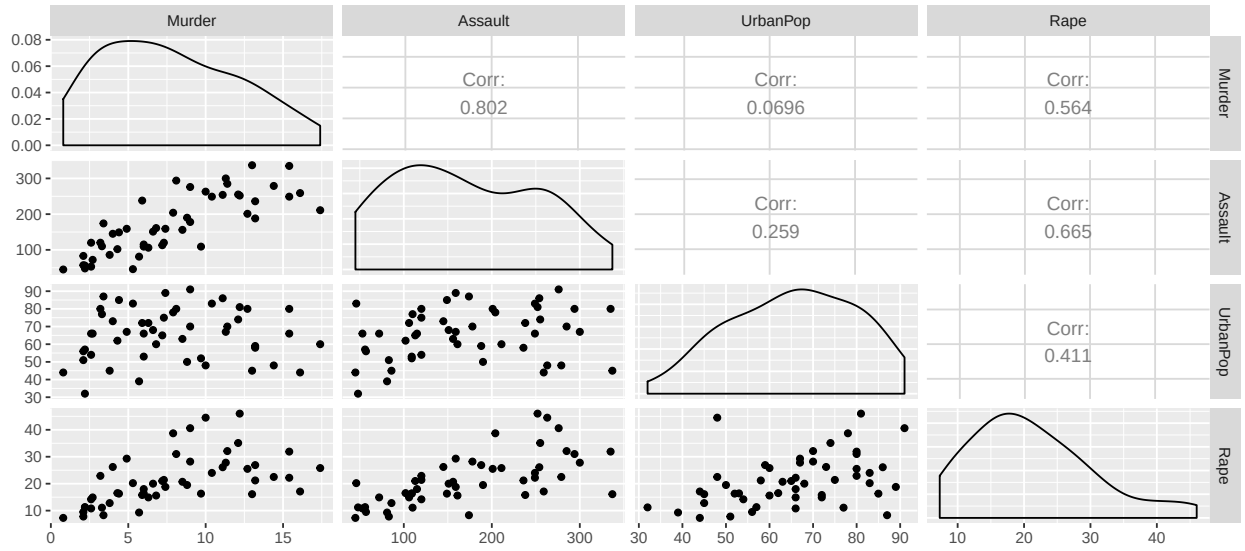
[,2] Assault (numerisch): Verhaftungen wegen Körperverletzung (pro 100.000)

[,3] UrbanPop (numerisch): Prozent der städtischen Bevölkerung

[,4] Rape (numerisch): Verhaftungen wegen Vergewaltigung (pro 100.000)

b)

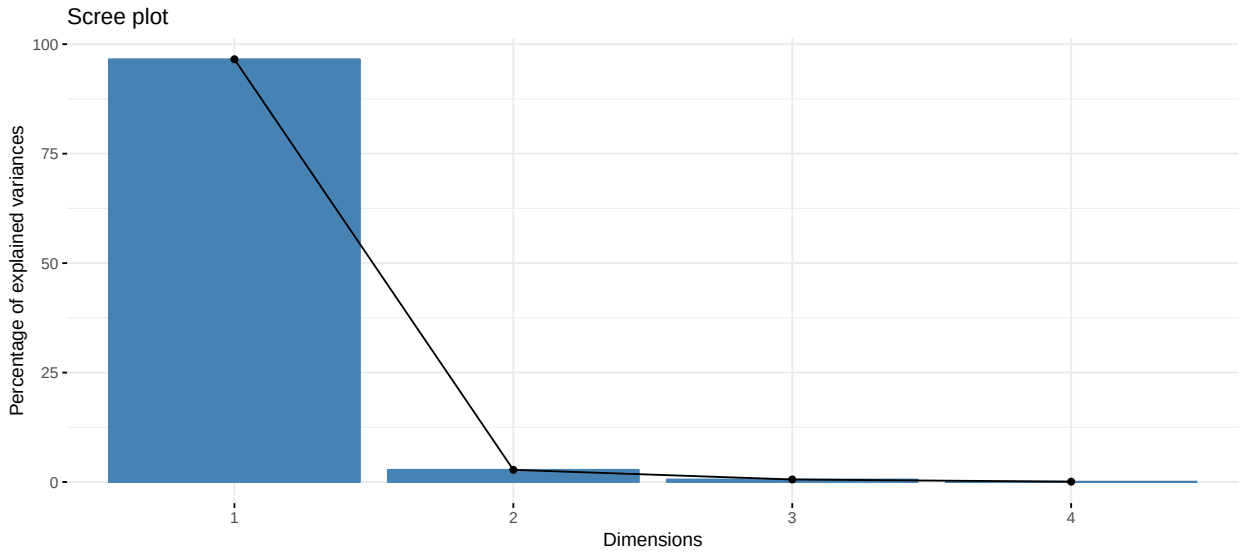
```
packageTest('ggplot2')
packageTest('GGally')
ggpairs(data)
```



c)

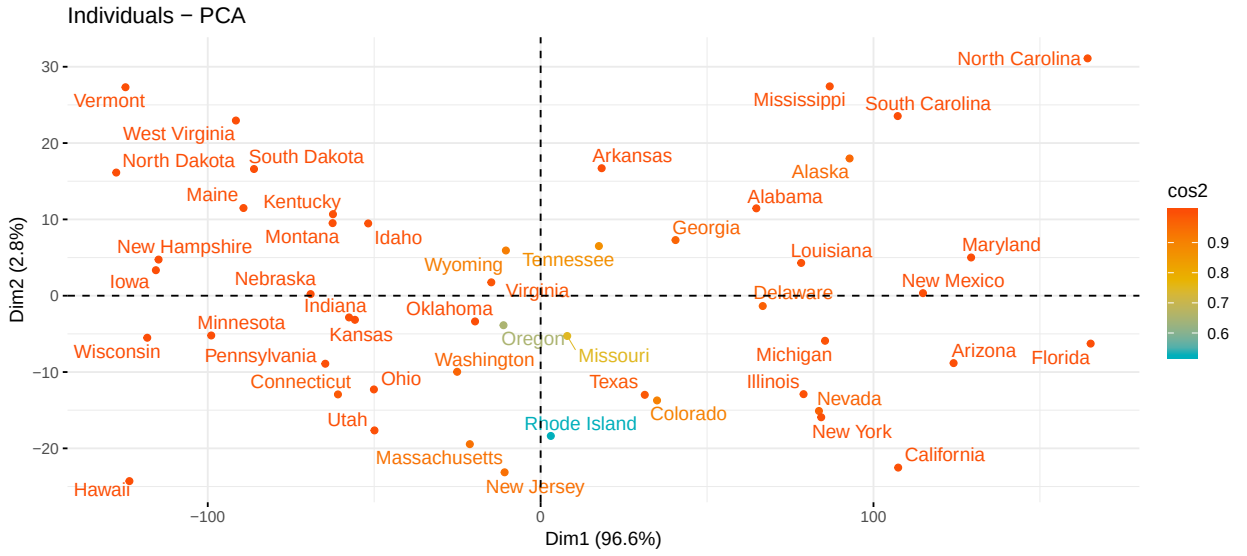
```
packageTest('factoextra')
```

```
pca <- princomp(data)
fviz_eig(pca)
```

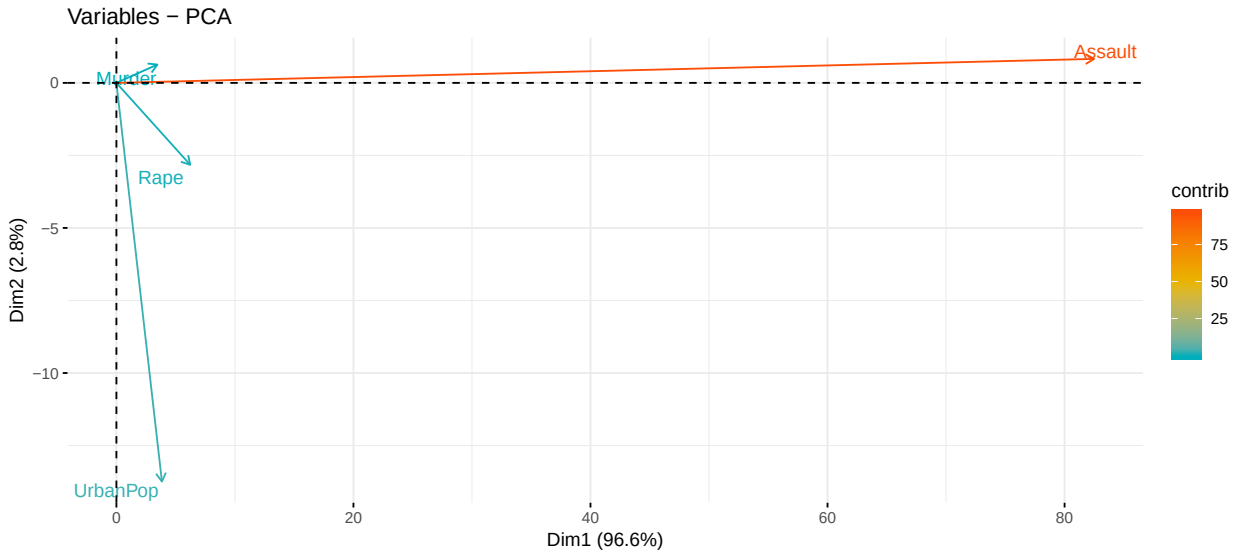


- Erste Hauptkomponente erklärt bereits 96.6% der Varianz.

```
fviz_pca_ind(pca,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```




```
fviz_pca_var(pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```

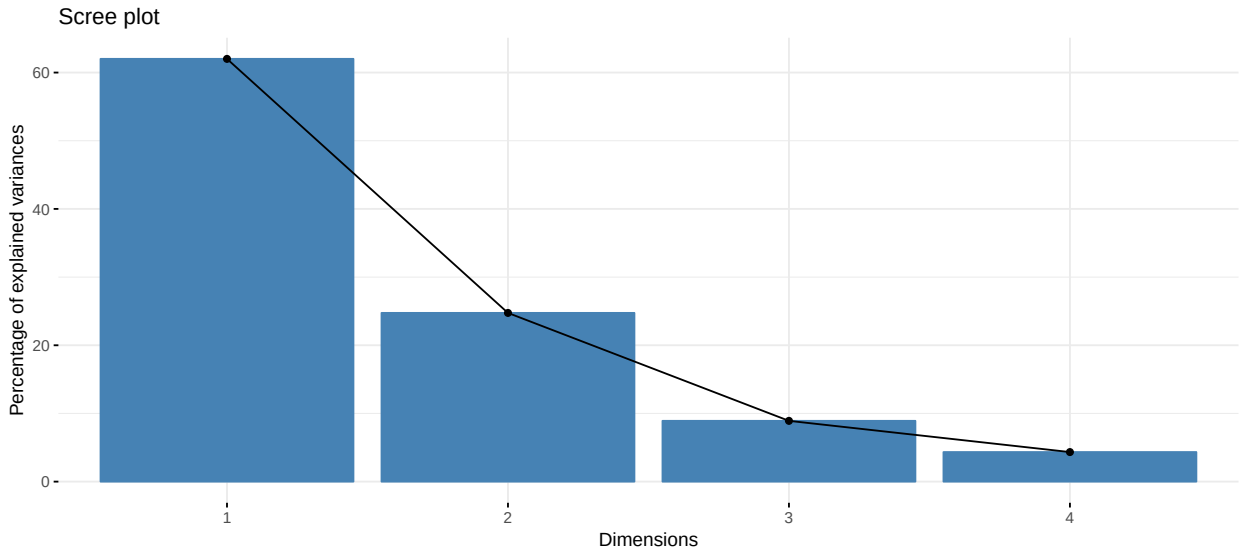


- Loadings erklären die Korrelation zwischen Variablen und Hauptkomponenten.
- Laut Grafik: Assault stark positiv mit Hauptkomponente 1 korreliert, UrbanPop stark negativ mit Hauptkomponente 2 korreliert.
- Loadings:

```
pca$loadings
#>
#> Loadings:
#>          Comp.1 Comp.2 Comp.3 Comp.4
#> Murder                          0.995
#> Assault   0.995
#> UrbanPop         -0.977 -0.201
#> Rape             -0.201  0.974
#>
#>                Comp.1 Comp.2 Comp.3 Comp.4
#> SS loadings      1.00   1.00   1.00   1.00
#> Proportion Var   0.25   0.25   0.25   0.25
#> Cumulative Var   0.25   0.50   0.75   1.00
```

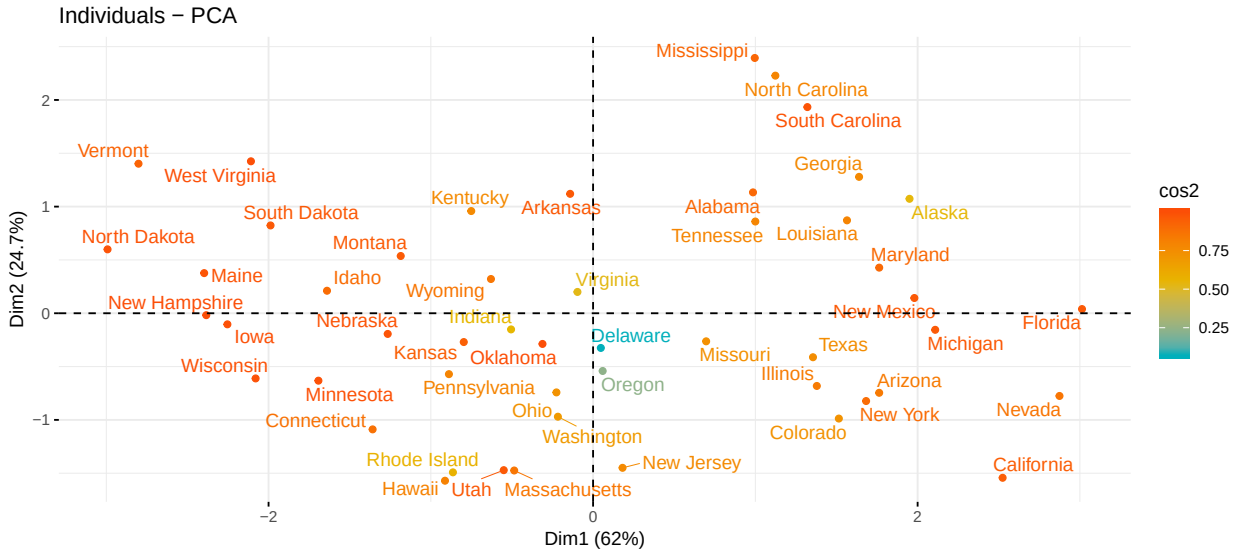
d)

```
pca_std <- princomp(data, cor = TRUE)
fviz_eig(pca_std)
```

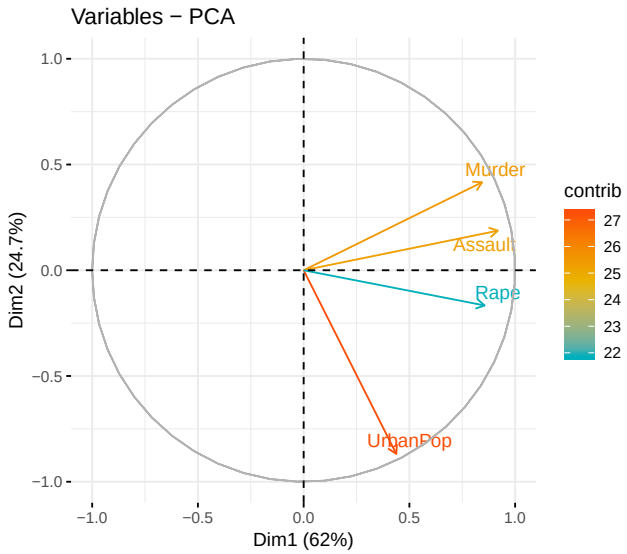


- Erste Hauptkomponente erklärt nun "nur" noch 62% der Varianz.

```
fviz_pca_ind(pca_std,
             col.ind = "cos2", # Color by the quality of representation
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```




```
fviz_pca_var(pca_std,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
             )
```



- Loadings:

```
pca_std$loadings
#>
#> Loadings:
#>          Comp.1 Comp.2 Comp.3 Comp.4
#> Murder    0.536  0.418  0.341  0.649
#> Assault   0.583  0.188  0.268 -0.743
#> UrbanPop  0.278 -0.873  0.378  0.134
#> Rape      0.543 -0.167 -0.818
#>
#>                Comp.1 Comp.2 Comp.3 Comp.4
#> SS loadings      1.00   1.00   1.00   1.00
#> Proportion Var   0.25   0.25   0.25   0.25
#> Cumulative Var   0.25   0.50   0.75   1.00
```

Standardisierung ist für PCA wichtig, da es sich um eine Varianzmaximierungsaufgabe handelt. Es projiziert die Originaldaten auf Richtungen, die die Varianz maximieren. Im nicht-standardisierten Fall in unserem Beispiel scheint es, als würde die erste Komponente die ganze Varianz in den Daten erklären (siehe Screeplot).

Wenn Sie sich den gleichen Plot nach Standardisierung der Daten ansehen, wird klar, dass auch die anderen Komponenten zur Varianzerklärung beitragen. Der Grund dafür ist, dass PCA versucht, die Varianz jeder Komponente zu maximieren.

Schauen wir uns die Kovarianzmatrix des nicht-standardisierten Datensatzes an:

```
cov(data)
#>              Murder   Assault   UrbanPop      Rape
#> Murder    18.970465  291.0624   4.386204  22.99141
#> Assault  291.062367 6945.1657 312.275102 519.26906
#> UrbanPop   4.386204  312.2751 209.518776  55.76808
#> Rape      22.991412  519.2691  55.768082  87.72916
```

Nun wird, klar, dass die PCA auf den nicht-standardisierten Daten natürlich “entscheidet”, stark in Richtung der Variable “Assault” zu projizieren, da deren Varianz weitaus größer ist, als die der anderen Variablen.

Eine Standardisierung in unserem Beispiel ist deshalb empfehlenswert.