Exercise Sheet 10 – Data Mining Wirtschaftsinformatik, HTW Berlin

Martin Spott

last revision 23 June 2020

This exercise is about Principal Components Analysis (PCA) which finds an orthonormal base for the feature space of data. The first base vector is oriented towards the direction of highest variance, the second one is the vector orthogonal to the first vector with highest variance and so on.

First load some libraries (install beforehand if necessary) and attach the data.

```r
# run this to install libraries straight from github
# install devtools first if necessary
library(devtools)
# install ggbiplot from github
#install_github("vqv/ggbiplot")
```

```r
# load libraries and data
library(ggbiplot)
data("iris")
```

Exercise 10.1

Please work through the exercise

10.4 Lab 1: Principal Components Analysis

in the book *An Introduction to Statistical Learning with Applications in R* by G. James, D. Witten, T. Hastie, R. Tibshirani (see http://www-bcf.usc.edu/~gareth/ISL/).

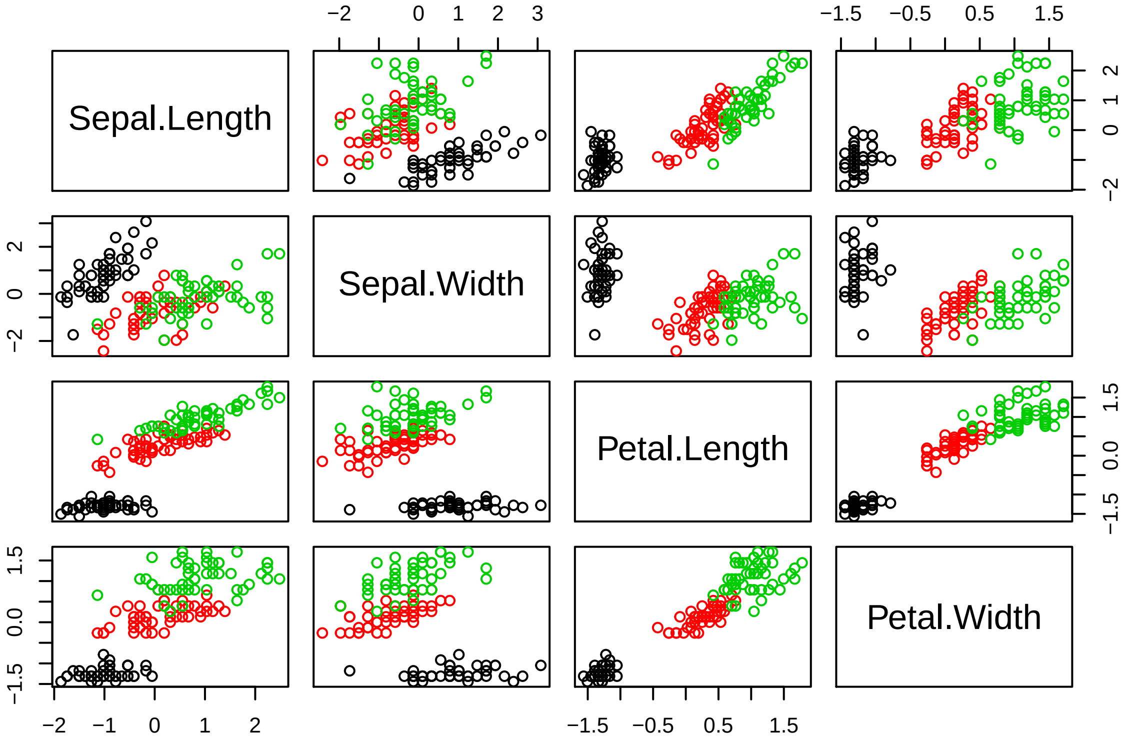
Exercise 10.2

Use PCA on the Iris data set as discussed in the lecture.

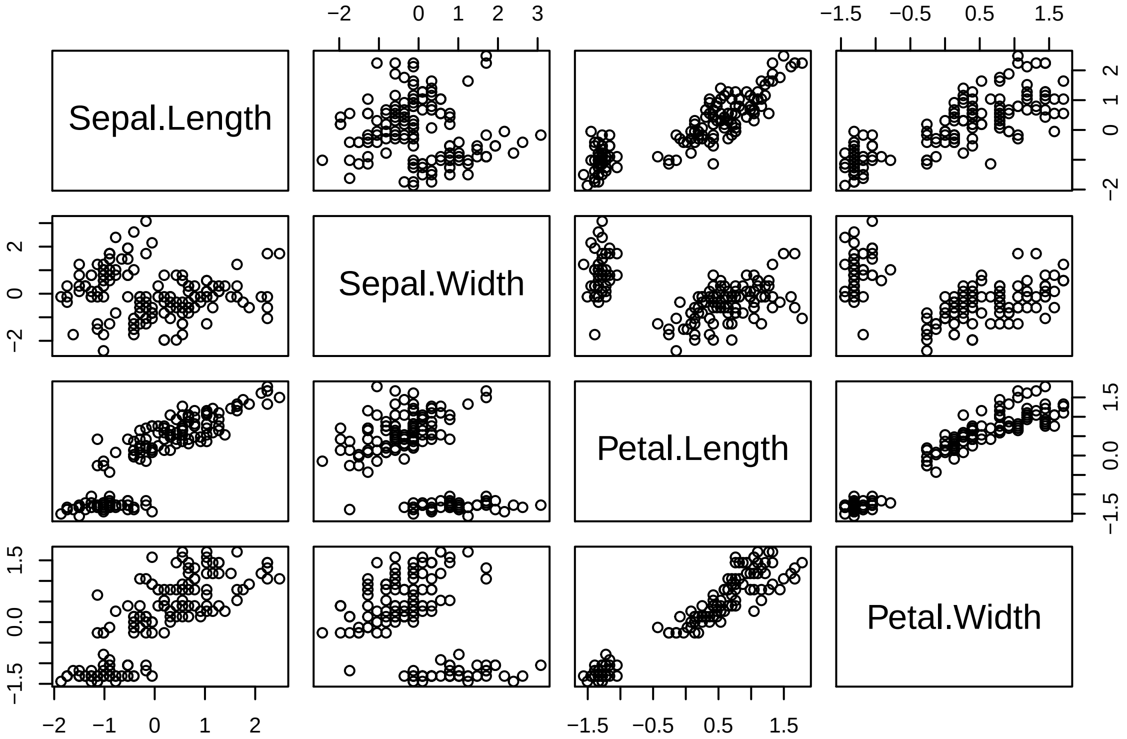
a) Visualise the data.

```r
# Scale the data for better comparison with PCA visualisation
iris_s <- as.data.frame(scale(iris[,1:4]))
# look at the Iris data set with and without class information

plot(iris_s, col=iris$Species)
```


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
−2
0
1
2
3
−1.5
−0.5
0.5
1.5
−2
−1
0
1
2
−1.5
0.0
1.5

```
plot(iris_s)
```


Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
−2
0
1
2
3
−1.5
−0.5
0.5
1.5
−2
−1
0
1
2
−1.5
0.0
1.5

b) Compute the PCA with `scale = TRUE` and discuss the results following Exercise 10.1.

```r
iris_pca <- prcomp(iris[,1:4], scale = TRUE)

# look at the result of the PCA
print(iris_pca)
```

```
## Standard deviations (1, .., p=4):
## [1] 1.7083611 0.9560494 0.3830886 0.1439265
##
## Rotation (n x k) = (4 x 4):
##                     PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

```r
# elements of the PCA object in R
names(iris_pca)
```

```
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

```r
# the data has been normalised (scaled)
iris_pca$center     # the mean values of the features
```

```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##     5.843333     3.057333     3.758000     1.199333
```

```r
iris_pca$scale      # 1 / standard deviation of features
```

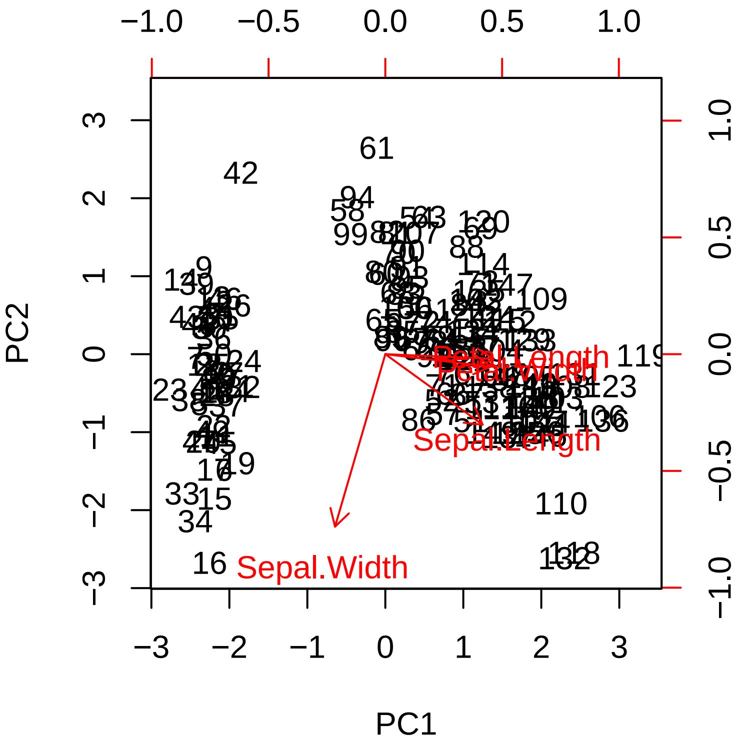
```
## Sepal.Length  Sepal.Width Petal.Length  Petal.Width
##    0.8280661    0.4358663    1.7652982    0.7622377
```

```r
iris_pca$rotation   # the loadings
```

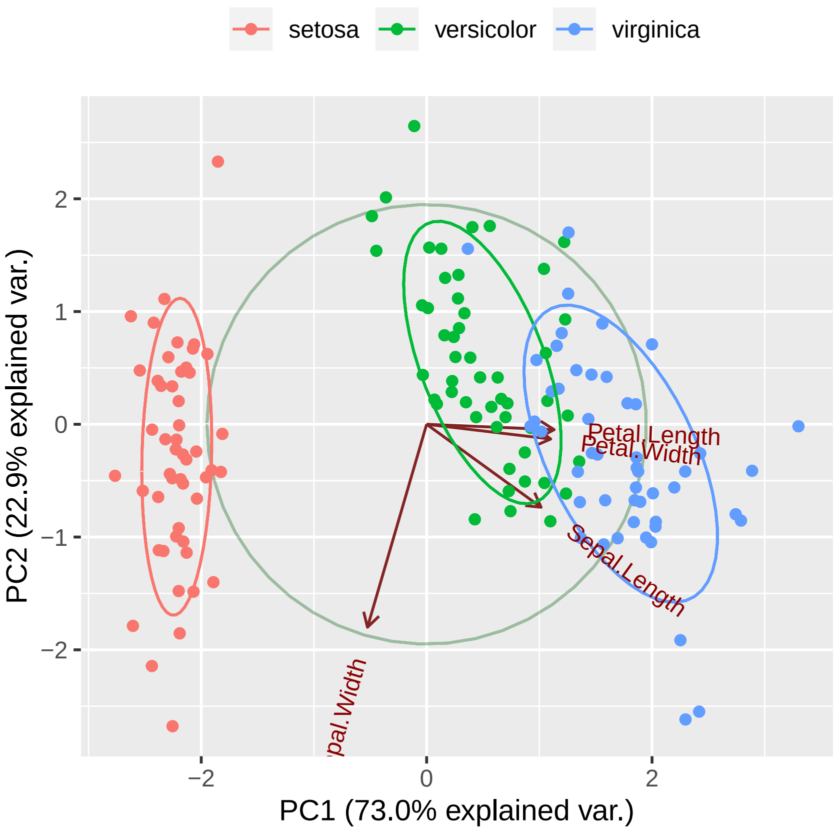
```
##                     PC1         PC2        PC3        PC4
## Sepal.Length  0.5210659 -0.37741762  0.7195664  0.2612863
## Sepal.Width  -0.2693474 -0.92329566 -0.2443818 -0.1235096
## Petal.Length  0.5804131 -0.02449161 -0.1421264 -0.8014492
## Petal.Width   0.5648565 -0.06694199 -0.6342727  0.5235971
```

c) Produce a biplot following the code in Exercise 10.1. Use the following code for a better visualisation of the biplot using `ggbiplot()` in the library of the same name.

```r
# standard biplot in R
biplot(iris_pca, scale = 0)
```



```
# nicer biplot with ggplot2
g <- ggbiplot(iris_pca, scale = 0,
              groups = iris$Species, ellipse = TRUE,
              circle = TRUE)
g <- g + scale_color_discrete(name = '')
g <- g + theme(legend.direction = 'horizontal',
               legend.position = 'top')
print(g)
```



Interpret the visualisation.

d) Generate the scree plots for the percentage of variance explained by the principal components (normal and cumulative).

```
# summary gives us information
#   about the percentage of variance explained
summary(iris_pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4
## Standard deviation     1.7084 0.9560 0.38309 0.14393
## Proportion of Variance 0.7296 0.2285 0.03669 0.00518
## Cumulative Proportion  0.7296 0.9581 0.99482 1.00000
```

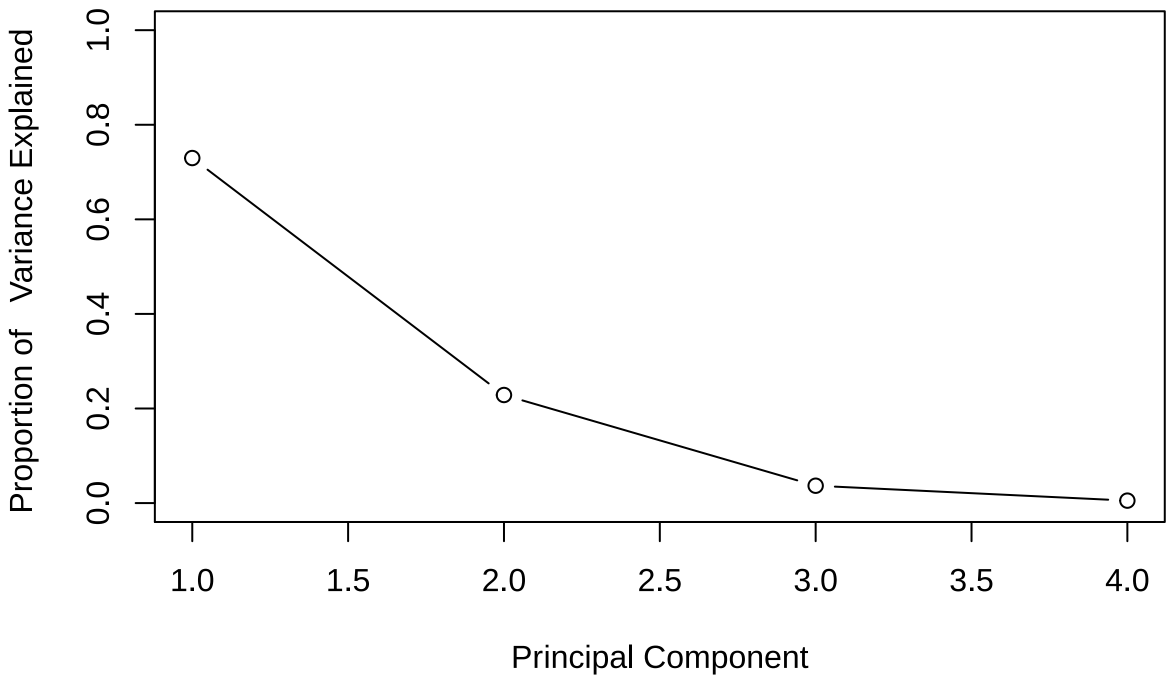
```
# compute variances and variance explained by hand
iris_pca.var <- iris_pca$sdev^2
iris_pca.var
```

```
## [1] 2.91849782 0.91403047 0.14675688 0.02071484
```

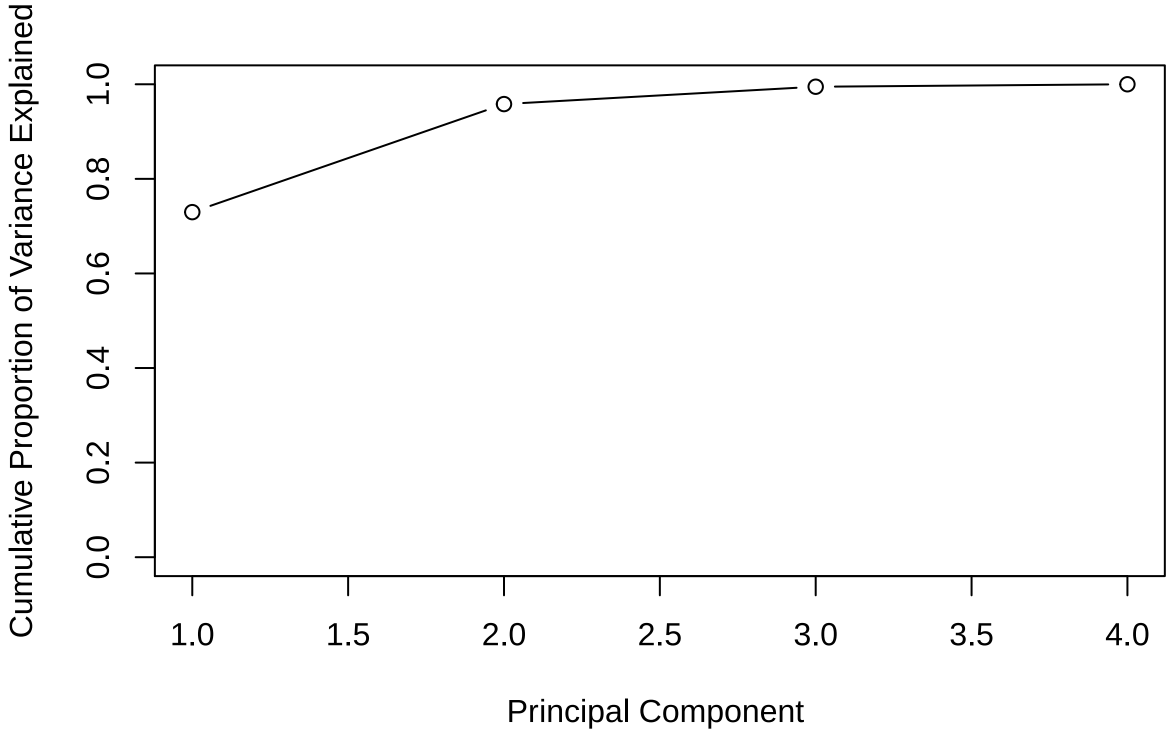
```
iris_pca.ve <- iris_pca.var / sum(iris_pca.var)
iris_pca.ve
```

```
## [1] 0.729624454 0.228507618 0.036689219 0.005178709
```

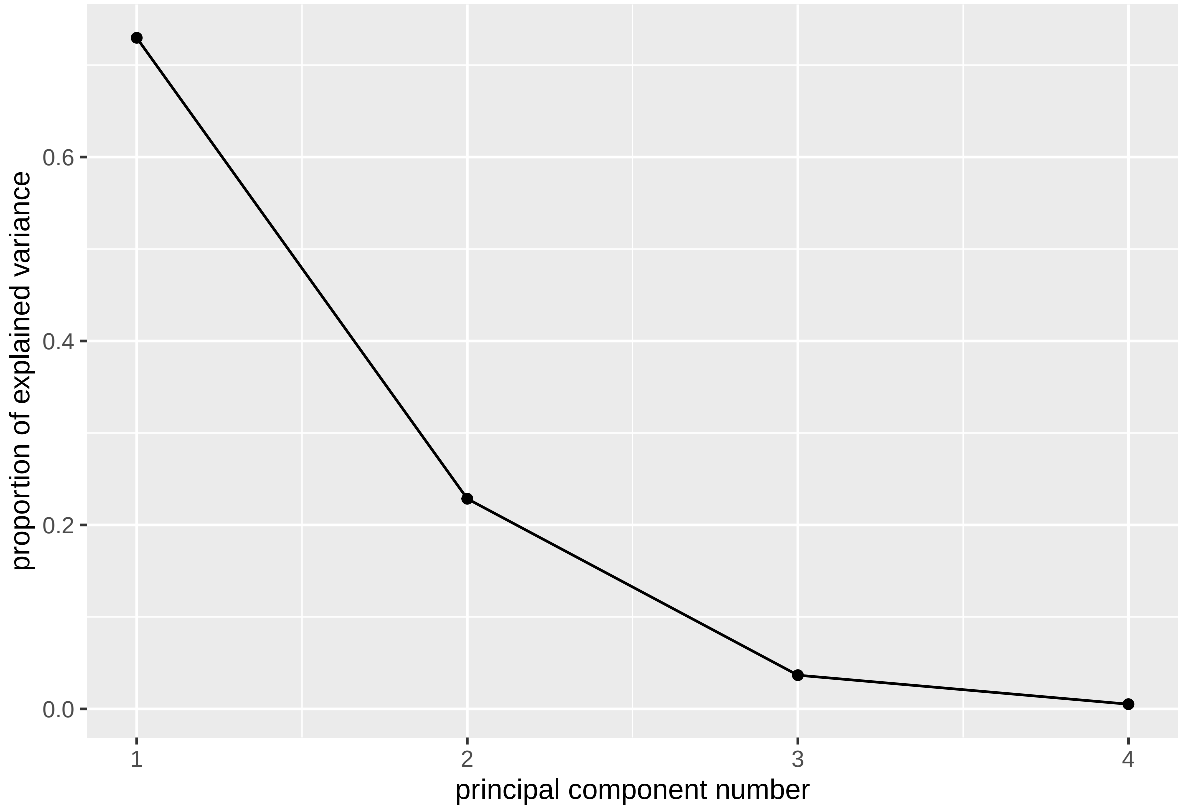
```
# scree plots of variance explained by number of principal components
plot(iris_pca.ve, xlab="Principal Component",
     ylab="Proportion of   Variance Explained ",
     ylim=c(0,1),type='b')
```



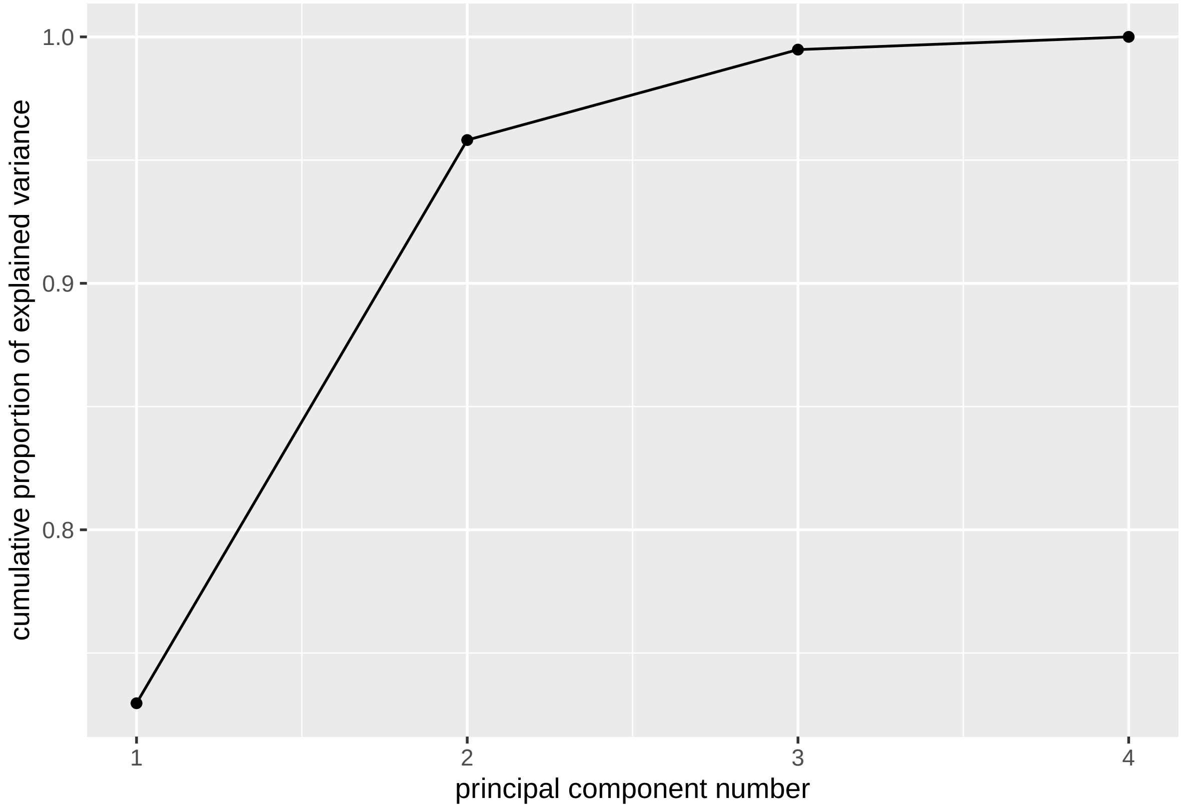
```r
plot(cumsum(iris_pca.ve), xlab="Principal Component ",
     ylab=" Cumulative Proportion of Variance Explained ",
     ylim=c(0,1), type='b')
```



```r
# same with ggplot
# proportion of variance explained
print(ggscreeplot(iris_pca, type = "pev"))
```



```r
# cumulative proportion
print(ggscreeplot(iris_pca, type = "cev"))
```



Exercise 10.3

Train a classification model like a decision tree on the PCA-version of the Iris data set. Compare the results (performance) with a similar model trained on the original data set.