Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering
IDP Institut für Datenanalyse und Prozessdesign

HS 2016

Statistisches Data Mining (StDM)

Woche 2

Aufgabe 1 PCA

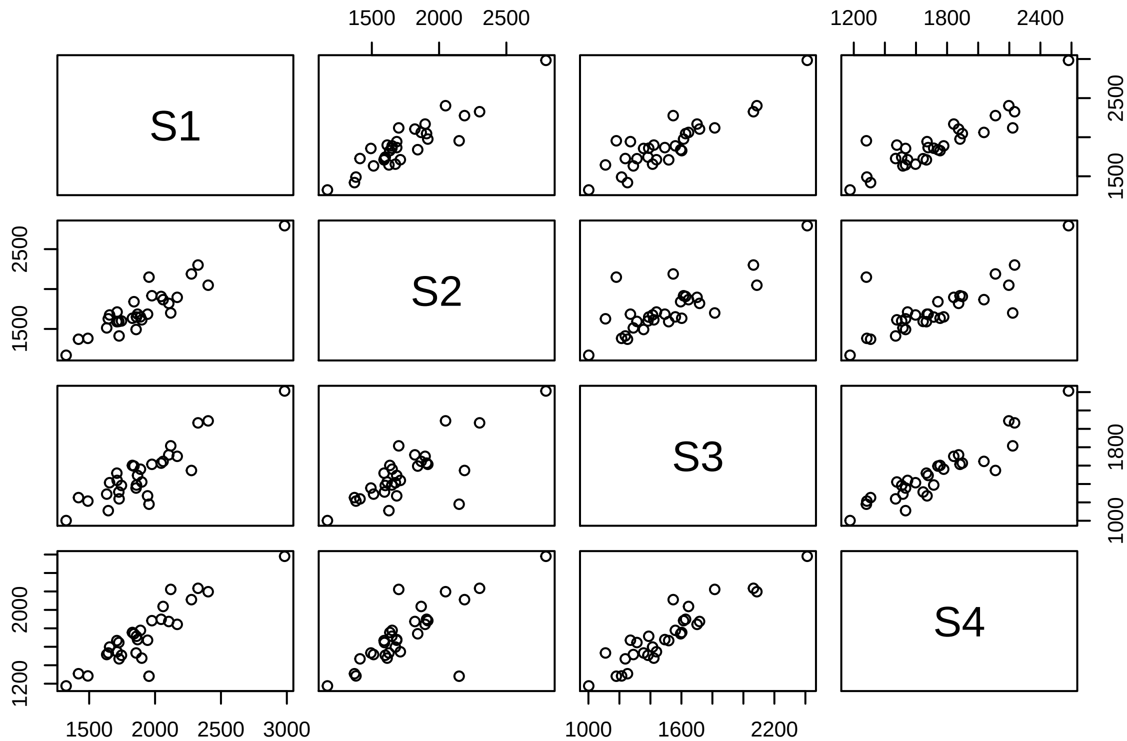
Der Datensatz `stiffness.da` enthält für $n = 30$ Holzbretter je vier verschiedenartige Messungen S1, S2, S3 und S4, die alle irgendwie die Festigkeit dieser Bretter messen. Die erste Messart besteht darin, dass eine Schockwelle durch das Brett gesendet wird. Die zweite wird während das Brett vibriert bestimmt, und die letzten beiden Messungen werden aus einem statischen Test ermittelt.

a) Verschaffen Sie sich einen Überblick über die und stellen Sie die Daten in einer Streudiagramm-Matrix dar. Was stellen Sie fest? Fallen Ihnen noch andere grafische Darstellungsmöglichkeiten für diese vierdimensionalen Daten ein?

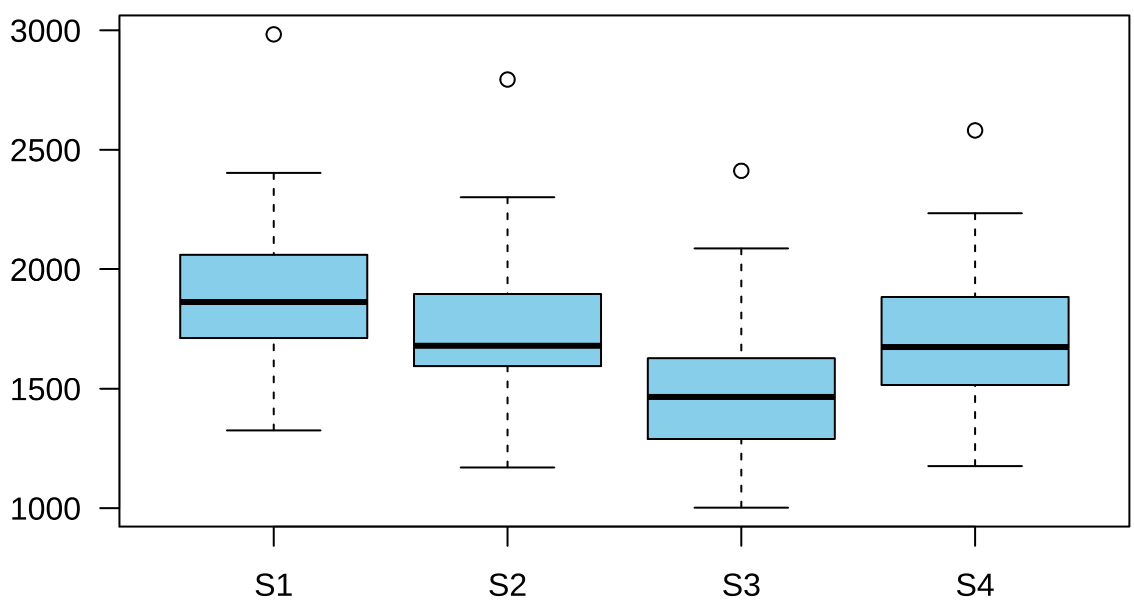
```
data<-read.table(file.path(baseDir,"stiffness.dat"), header=T, quote="\"")
summary(data)
```

```
##        S1             S2             S3             S4
##  Min.   :1325   Min.   :1170   Min.   :1002   Min.   :1176
##  1st Qu.:1715   1st Qu.:1596   1st Qu.:1296   1st Qu.:1520
##  Median :1863   Median :1680   Median :1466   Median :1674
##  Mean   :1906   Mean   :1750   Mean   :1509   Mean   :1725
##  3rd Qu.:2057   3rd Qu.:1889   3rd Qu.:1624   3rd Qu.:1881
##  Max.   :2983   Max.   :2794   Max.   :2412   Max.   :2581
```

```
pairs(data)
```




```
# Die Verschiedenen Variablen korrelieren stark untereinander, speziell S1 und S2 bzw. S3 und
boxplot(data,col="skyblue",las=1)
```




```
apply(data,2,sd)
```

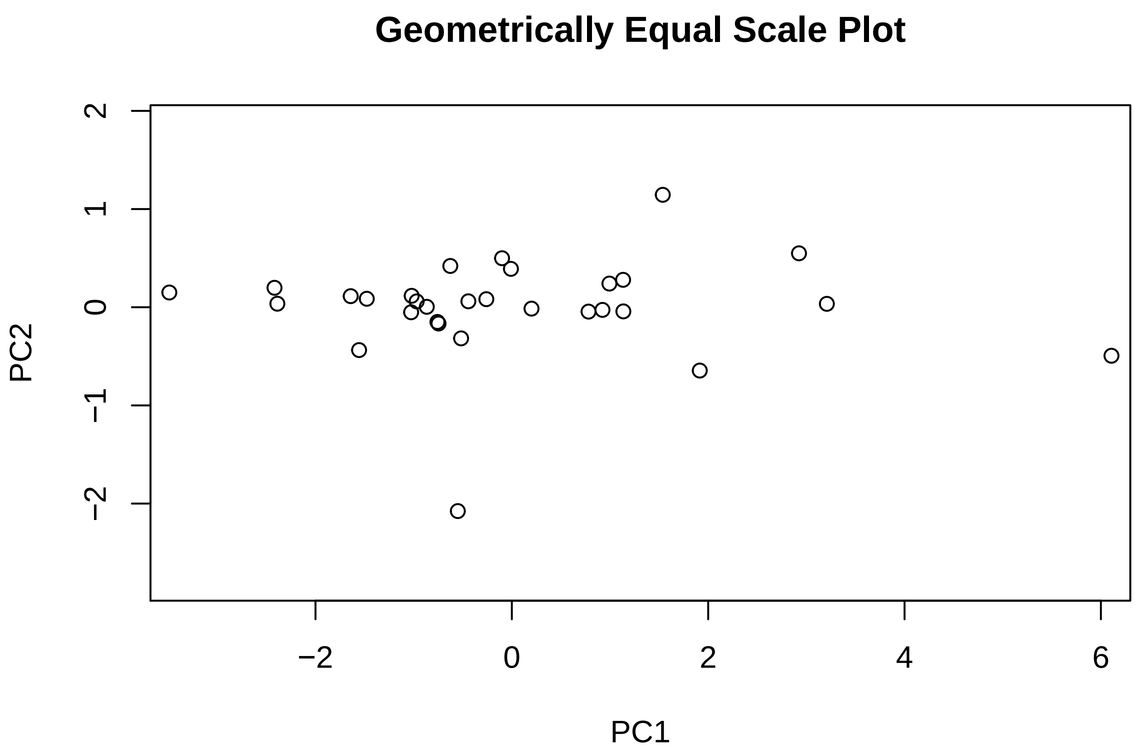
```
##        S1       S2       S3       S4
## 324.9866 318.6065 303.1783 322.9525
```

```
#     S1       S2       S3       S4
# 324.9866 318.6065 303.1783 322.9525
# Die Standardabweichung ist bei allen Variablen im aehlichen Bereich. Dies kann anhand der Zahlen
# oder aber auch anhand der hoehe (Interquartile range) der Boxen im boxplot festgestellt werden.
```

b) Soll die PCA auf skalierten Variablen durchgeführt werden.
Es ist sinnvoll, die PCA mit skalierten Daten durchzufuehren, da die Variablen nicht die gleichen physikalischen Groesse besitzen. Allerdings werden sich die Resultate kaum von einer PCA mit den unskalierten Daten unterscheiden, da alle Variablen etwa die gleiche Streuung aufweisen.

c) Führen Sie eine PCA auf den geeigneten Daten durch und stellen Sie die Daten in den ersten beiden Hauptkomponenten dar, verwenden Sie dazu die function `eqscplot`. Beschreiben Sie die Struktur der Daten in dieser Darstellung.

```
library(MASS)
data.pca<-prcomp(data,retx=T,scale=T)
eqscplot(data.pca$x[,1],data.pca$x[,2],main="Geometrically Equal Scale Plot",xlab="PC1",ylab="PC2"
```




```
#identify(data.pca$x[,1],data.pca$x[,2])
# Dies ist ein "Geometrically Equal Scale Plot" oder kurz "equal scale Plot". Der Name bezieht sic
# auf die identisch skalierten Achsen (x-Achse, y-Achse) im dargestellten Plot, diese Eigenschaft
# ist bei der Betrachtung im 2-dimensinalen Raum sehr hilfreich ( 1cm auf der x-Achse ist gleich
# lang wie 1cm in y-Richtung).
# In unsere konkreten Darstellung ist sichtbar, dass es ein Gruppe gibt, welche fast alle Punkt
# beinhaltet. Nur Beobachtung Nr. 9 oder auch Nr. 16 koennten Ausreisser sein. Im Weitern kann
# gesagt werden, dass die Streuung der ersten PC dominiert. Die Streuung der zweiten PC ist im
# Verhaetnis zur ersten PC sehr klein. Dass die Streuung der ersten PC groesser ist als die der
```

```r
# zweiten PC ist kein Zufall, dies ist aufgrund der Konstruktion der PC immer so (die erste PC
# wird in die Richtung der groessten Streuung der konstruiert)
```

d) Stellen Sie eine Tabelle mit den prozentualen Beiträgen zur totalen Varianz jedes Eigenwertes zusammen. Genügen die beiden ersten Hauptkomponenten, um die Variabilität der Daten sinnvoll zu approximieren?

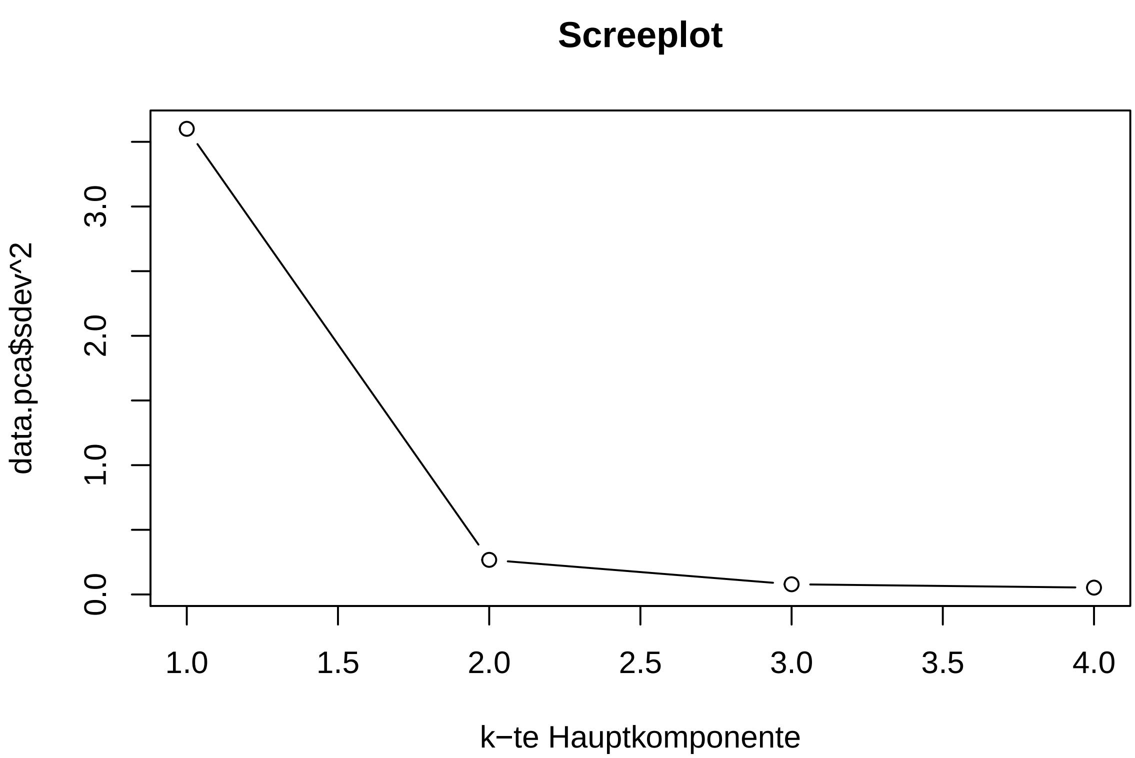
```r
summary(data.pca)
```

```
## Importance of components:
##                           PC1     PC2     PC3     PC4
## Standard deviation     1.8975 0.51727 0.28091 0.22995
## Proportion of Variance 0.9002 0.06689 0.01973 0.01322
## Cumulative Proportion  0.9002 0.96705 0.98678 1.00000
```

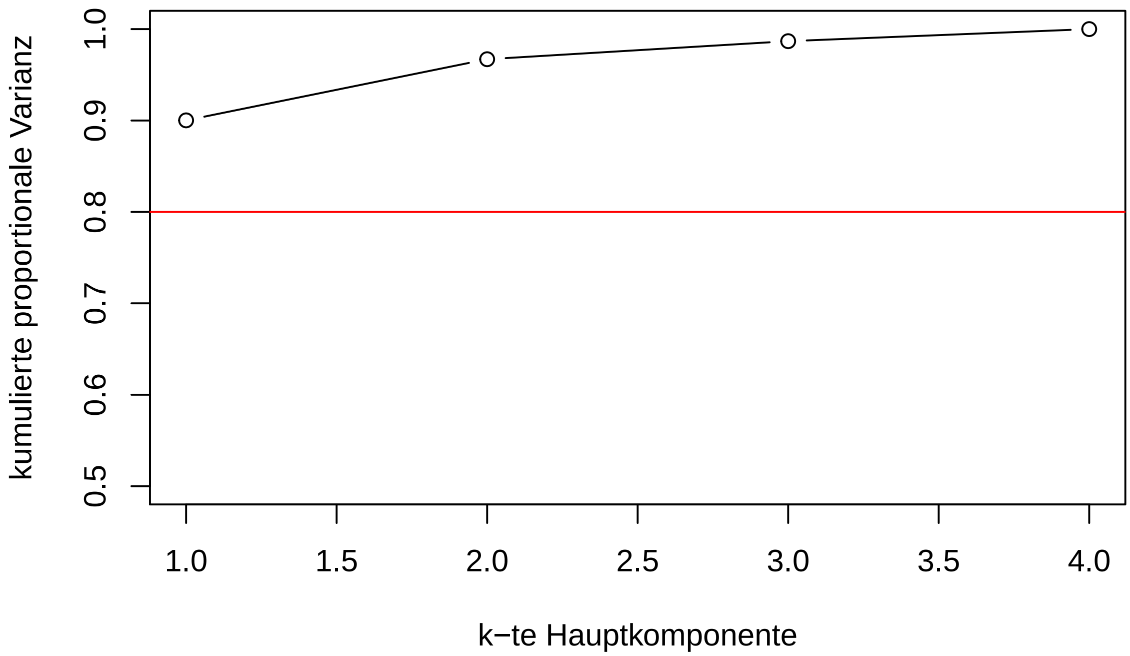
```r
# Laut der Faustregel, dass mind. 80% der totalen Varianz in der Abbildung der ersten
# x-PC's (hier der 2-PC) erklaert werden sollten, ist dies hier mehr als erfuellt. Denn bereits die
# erste PC erklaert schon ueber 90% (90.02%), die ersten zwei sogar 96.71%! Dass heisst, die
# Approximation mit den ersten beiden PC ist hervorragend.
```

e) Benutzen Sie für die Beurteilung der Approximation das Scree-Diagramm. Kommen Sie zum gleichen Schluss wie in (d)?

```r
plot(data.pca$sdev^2,type="b",main="Screeplot",xlab="k-te Hauptkomponente")
```



```r
# Andere Darstellung (proportional kumulierte Varianz der PC)
plot(cumsum(data.pca$sdev^2)/sum(data.pca$sdev^2), type="b",ylab="kumulierte proportionale Varianz",
     xlab="k-te Hauptkomponente",ylim=c(0.5,1))
abline(h=0.8,col="red")
```




```r
  # Im Screeplot ist der sogneannte Knick sehr gut sichtbar. Wie wir wissen, gehoert der Knick in
# diesem Plot zum Geroell, also waehlen wir nur die erste PC.
# Dies wiederspigelt das Resultat von d), als wir gesehen haben, dass die Varianz in der
# ersten PC bereits ueber 90% der totalen Varianz erklaert.
```

f) Wie könnte man die Hauptkomponenten interpretieren?

```r
round(data.pca$rotation,2)
```

```
##     PC1   PC2   PC3   PC4
## S1 0.51 -0.21  0.25 -0.80
## S2 0.48 -0.73 -0.15  0.46
## S3 0.50  0.47 -0.73 -0.02
## S4 0.50  0.45  0.62  0.40
```

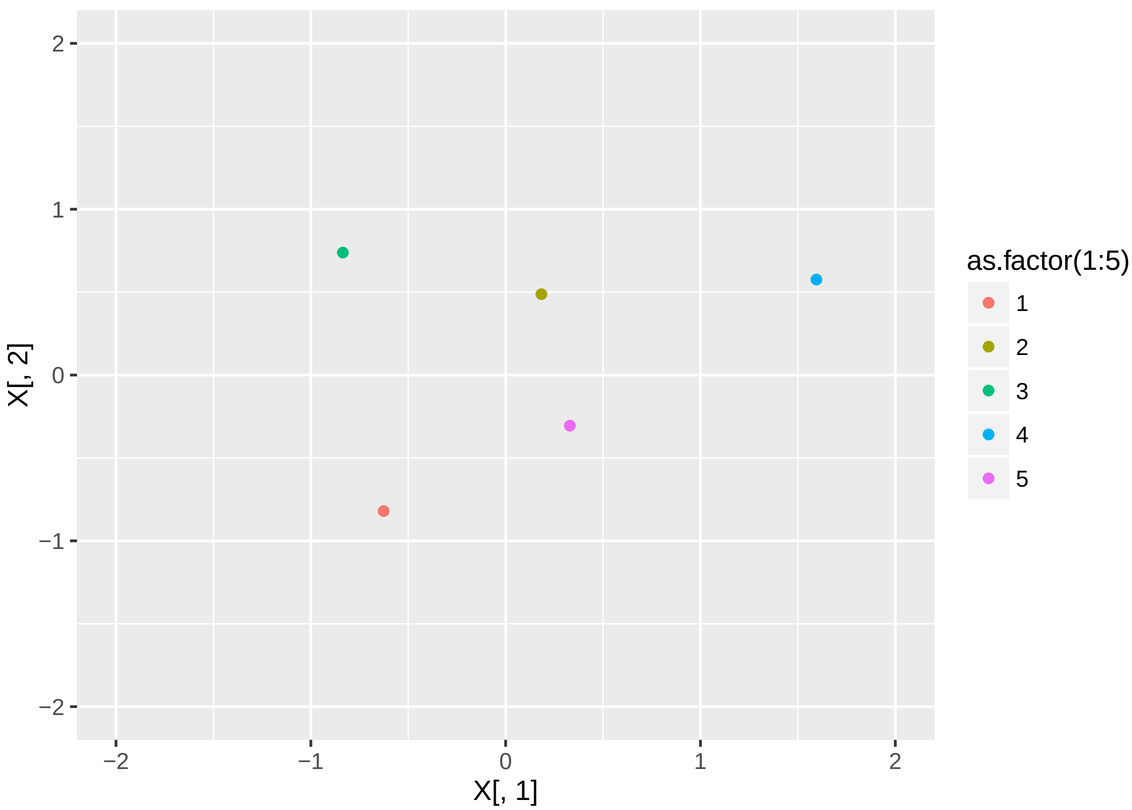
```r
#     PC1   PC2   PC3   PC4
# S1 0.51 -0.21  0.25 -0.80
# S2 0.48 -0.73 -0.15  0.46
# S3 0.50  0.47 -0.73 -0.02
# S4 0.50  0.45  0.62  0.40

# Hier sieht man, wie die urspruenglichen Variablen gewichten werden muessen, um sie linear zu
# den PCs zu transformiern. Diese Eintraege werden auch Loadings genannt.
# Bei der ersten PC kann festgestellt werden, dass so etwas wie ein Mittelwert aller Variablen
# gebildet wird. Bei der zweiten PC zum Beispiel dominiert die zweite Variable (S2). Es soll aber
# darauf hingewiesen werden, dass die Relevanz der PC abnimmt, sprich die wichtigste die erste ist.
```

Aufgabe 2 Metric MDS vs PCA

a) Create a data matrix with dimension 5x2 (5 examples, 2 features) by drawing random numbers from a Gaussian `X <- matrix(rnorm(10), nrow = 5)` and plot the data matrix.

```
set.seed(1)
X <- matrix(rnorm(10), nrow = 5)
library(ggplot2)
qplot(x=X[,1],y=X[,2], col=as.factor(1:5)) + xlim(-2,2)+ylim(-2,2)
```

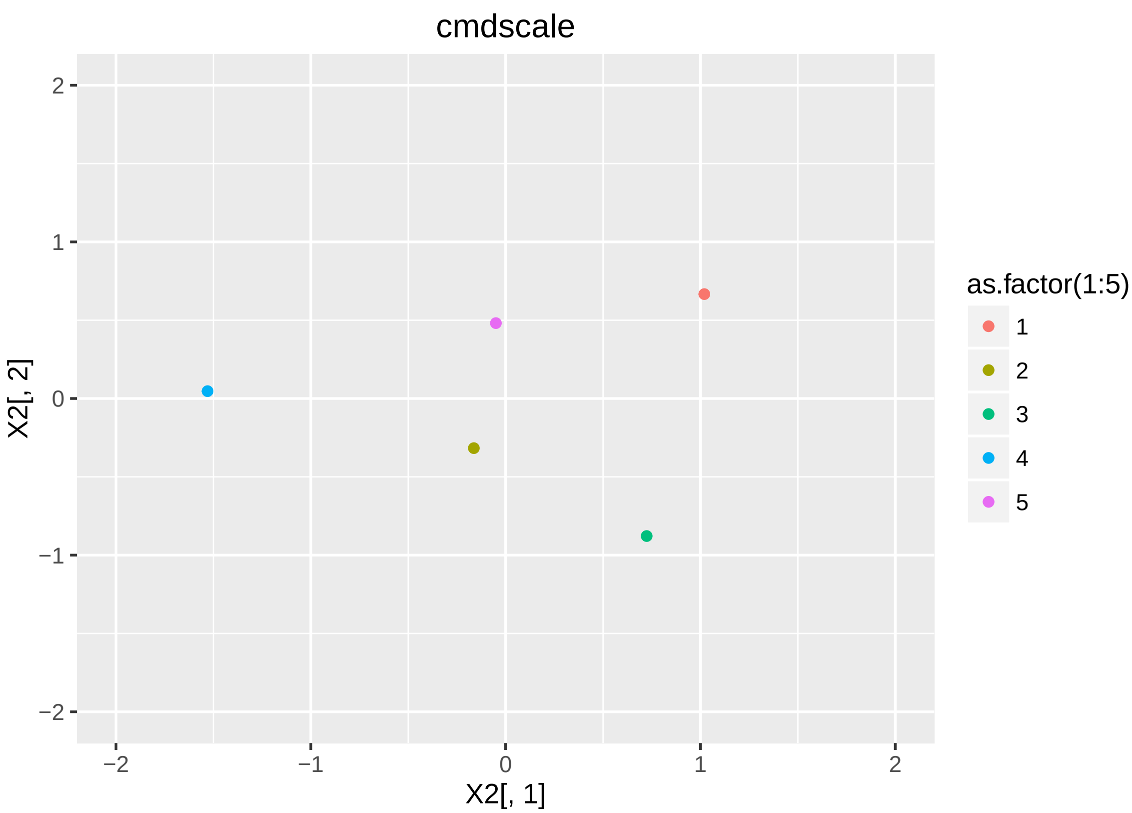


b) Calculate all pairwise Euklidean distances with (`dist`) and do a metric MDS (`cmdscale`). Print the eigenvalues.

```
d = dist(X)
res = cmdscale(d, eig=TRUE, k=2)
round(res$eig,2)
```

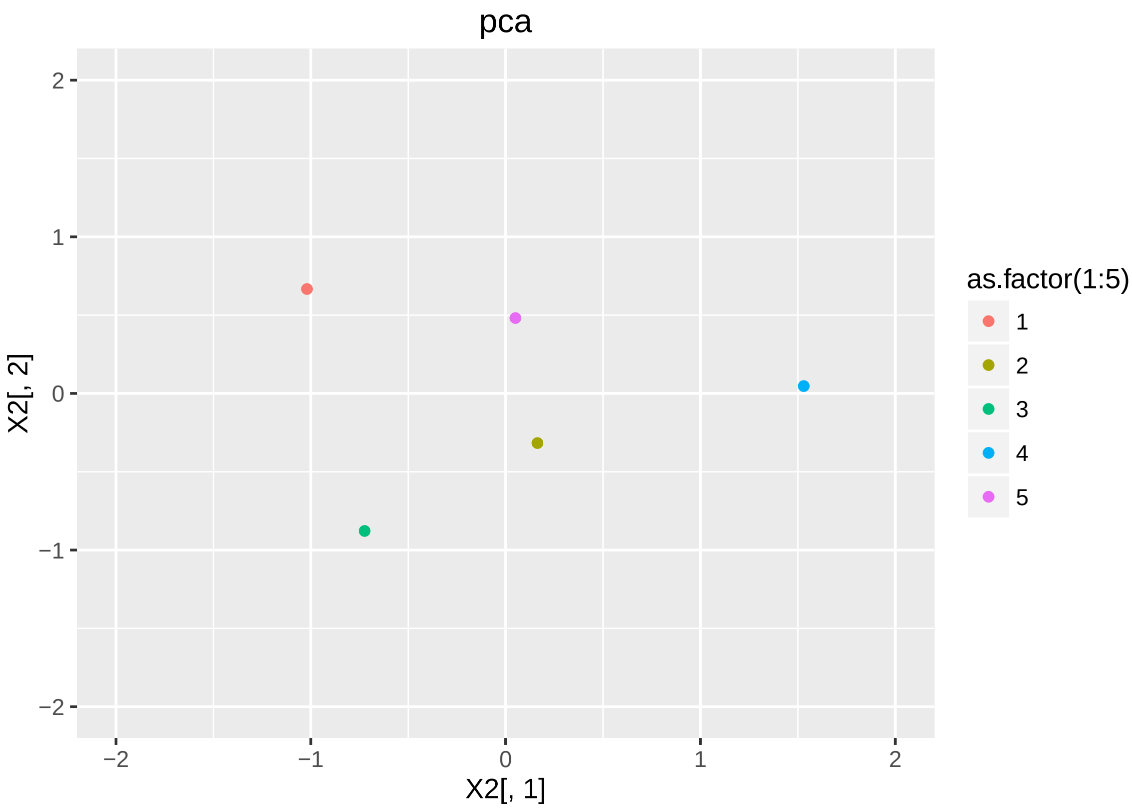
```
## [1] 3.93 1.55 0.00 0.00 0.00
```

```
X2 = res$points
qplot(x=X2[,1],y=X2[,2], col=as.factor(1:5),main='cmdscale')  + xlim(-2,2)+ylim(-2,2)
```



c) Do a principal component analysis and compare it to a) and b).

```
res = prcomp(X)
X2 = res$x
qplot(x=X2[,1],y=X2[,2], col=as.factor(1:5), main='pca')  + xlim(-2,2)+ylim(-2,2)
```



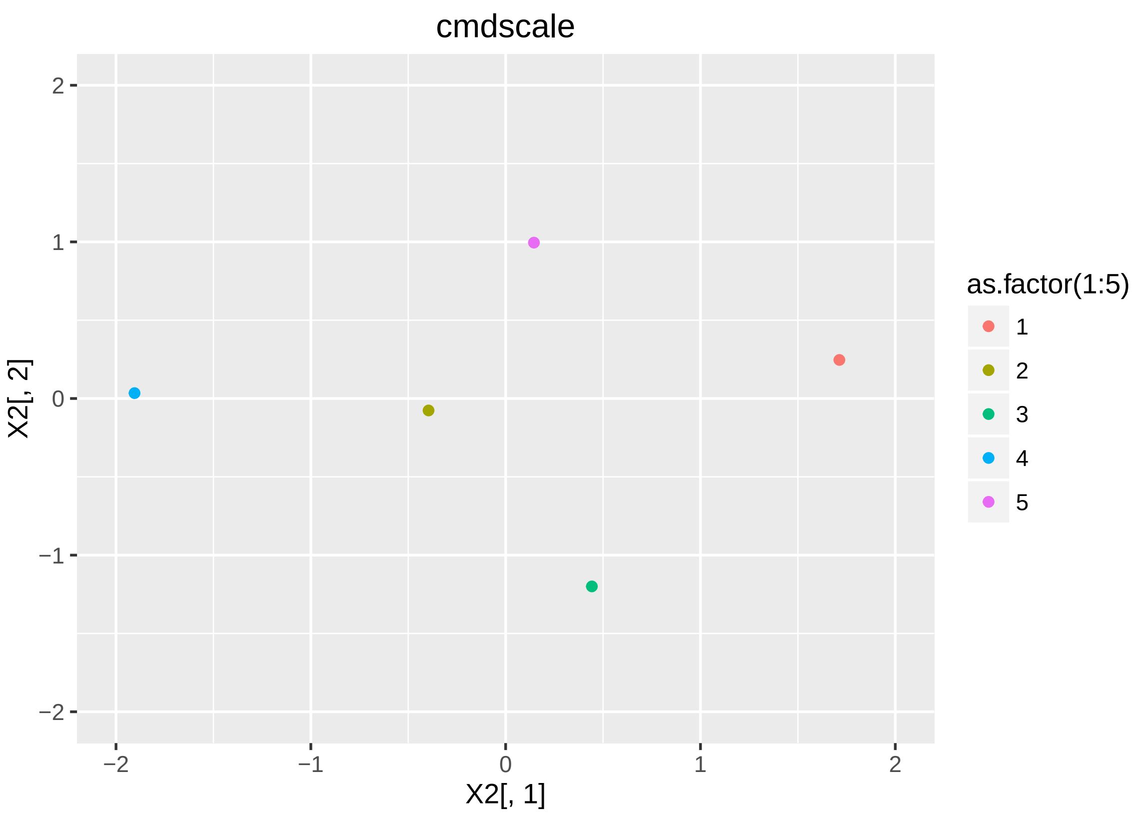
All aproaches reproduce up to a flip and rotation the original data in a)

d) Repeat b) but now use the manhattan distance.

```
d = dist(X, 'manhattan')
res = cmdscale(d, eig=TRUE, k=2)
round(res$eig,2)
```

```
## [1]  6.94  2.50  0.00 -0.02 -0.65
```

```
X2 = res$points
qplot(x=X2[,1],y=X2[,2], col=as.factor(1:5),main='cmdscale')  + xlim(-2,2)+ylim(-2,2)
```



e) Metric MDS can also used (for slightly) non-Euklidean Distances. The file `airdist.Rmd` contains the distance between some US airports (in miles). These distances can't be Euclidean since we live on a sphere.

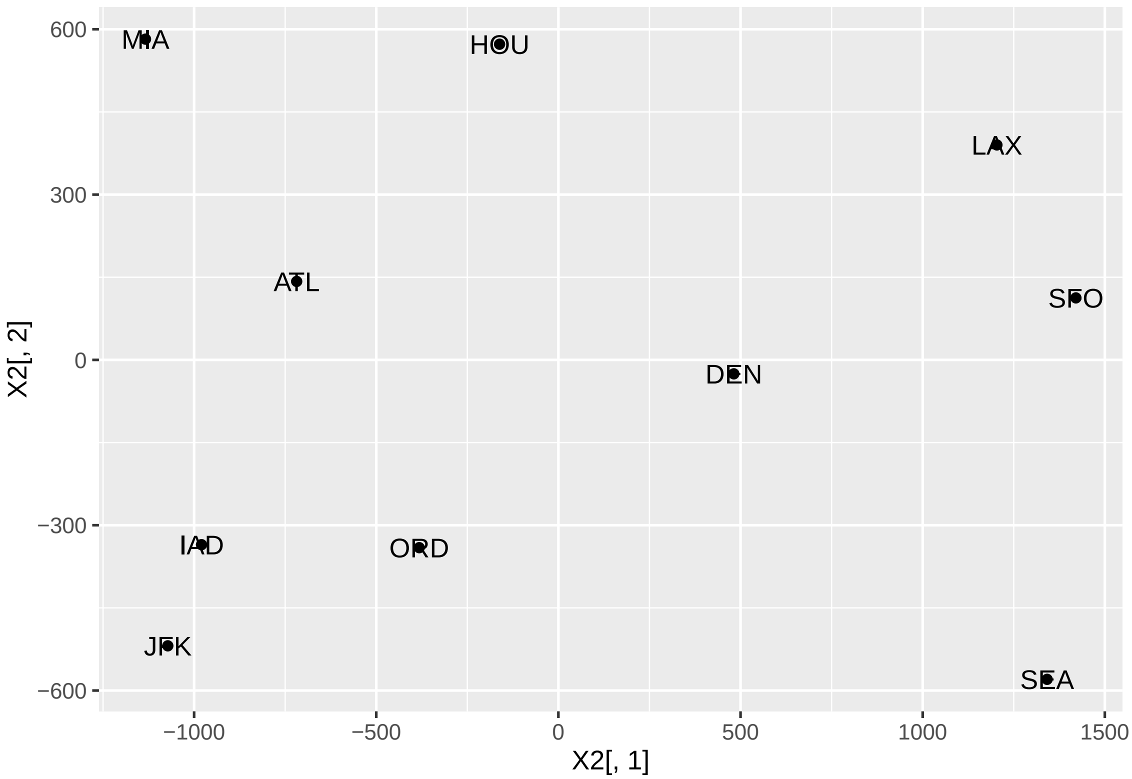
```
load(file.path(baseDir, 'airdist.rda'))
print(airdist)
```

```
##      ATL  ORD  DEN  HOU  LAX  MIA  JFK  SFO  SEA
## ORD  587
## DEN 1212  920
## HOU  701  940  879
## LAX 1936 1745  831 1374
## MIA  604 1188 1726  968 2339
## JFK  748  713 1631 1420 2451 1092
## SFO 2139 1858  949 1645  347 2594 2571
## SEA 2181 1737 1021 1891  959 2734 2408  678
## IAD  543  597 1494 1220 2300  923  205 2442 2329
```

```
X2 = cmdscale(airdist)
str(X2)
```

```
##  num [1:10, 1:2] -718 -382 482 -161 1204 ...
##  - attr(*, "dimnames")=List of 2
##   ..$ : chr [1:10] "ATL" "ORD" "DEN" "HOU" ...
##   ..$ : NULL
```

```r
df = data.frame(X2)
qplot(x=X2[,1],y=X2[,2]) + geom_text(label=rownames(df))
```

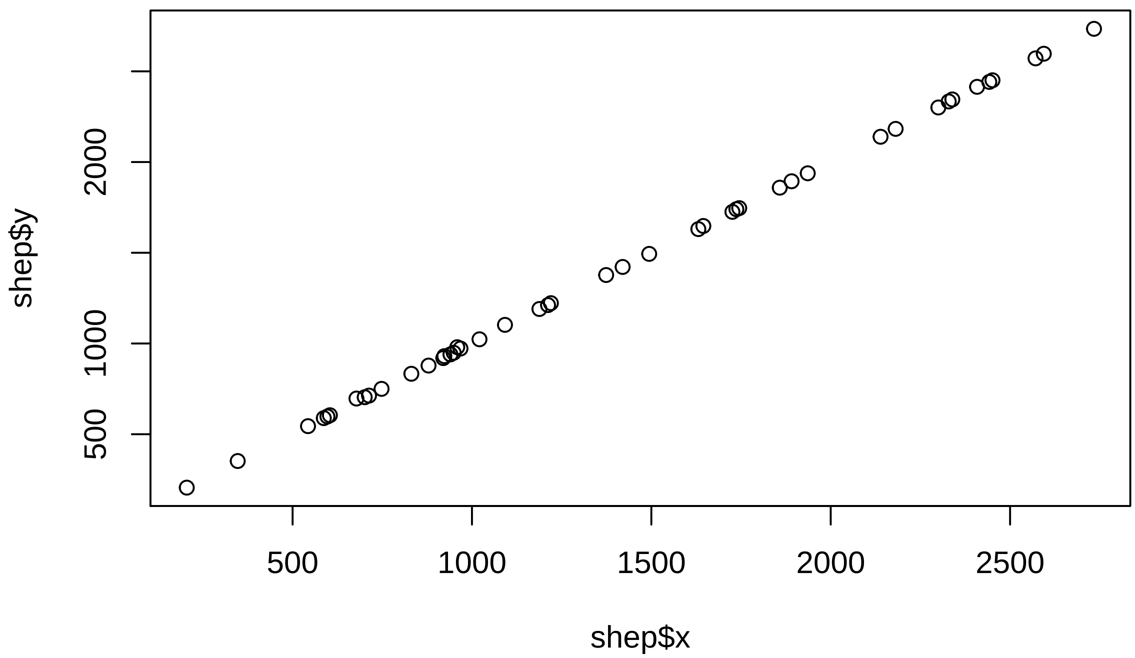



```r
# Aficionados
# Some of the eigenvalues are negative, indicating that the original distances are
# non-Eulikean
res = cmdscale(airdist, eig = TRUE)
res$eig
```

```
## [1] 9.581705e+06 1.686606e+06 7.736930e+03 1.466986e+03 6.523333e+02
## [6] 1.851352e+02 -6.621121e-10 -6.669100e+02 -5.630189e+03 -3.524780e+04
```

f) Using the `Shepard` function in the `MASS` package, plot the distances before and after the MDS against each other. What is the largest difference?

```r
library(MASS)
shep <- Shepard(airdist, X2)
plot(shep)
```




```
max(abs(shep$x-shep$y))
```

```
## [1] 20.4741
```