

Assignment 1: Task 1 - PCA

Group 11

16.10.2018

0. Load Libraries

```
library(tidyverse)
library(nFactors)
```

1. Problem Statement

The presented data from a conducted study shows the recovery status for 100 participants after taking two different drugs with differing doses. Recovery is measured as the percentage drop in body pathogens before and after taking the respective drug. In order to judge on the effectiveness of the drugs the principal component analysis is applied, to shrink the information to only a subset of the variables' dimensions. The associated identification of useful factor brandings and the reflection of the data on the identified subdimensions should help to evaluate the initial question.

The second section starts with the data preparation and a brief description of the data. Afterwards, in section three, the assumptions for using principal component analysis are described including the way we deal with them. In section four the principal component analysis is applied and the most relevant outputs are described. The last section covers the answering of first, how the factors in the subspace can be labeled appropriately and second, which drugs were effective.

2. Descriptive Statistics

Before dealing with the principal component analysis (PCA), it is important to prepare the data. After importing the data and performing the glimpse-function, one can see that the data is stored in a data.frame object with the dimension 100 x 9.

```
# Load data
drugdata <- read.table("http://feb.kuleuven.be/martina.vandebroek/public/STATdata/drugsrecovery.txt",
                      header=T)

# Show class of drugdata object
class(drugdata)

## [1] "data.frame"

# Show structure of dataframe
glimpse(drugdata)

## Observations: 100
## Variables: 9
## $ ID      <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1...
## $ L500    <int> 15, 10, 10, 10, 10, 20, 15, 5, 15, 10, 5, 20, 15, 20, 20...
## $ L1000   <int> 20, 15, 15, 15, 10, 20, 15, 5, 15, 10, 10, 20, 15, 30, 1...
## $ L2000   <int> 25, 5, 30, 5, 5, 20, 15, 5, 15, 5, 10, 25, 5, 20, 20, 20...
## $ L4000   <int> 30, 15, 30, 5, 25, 5, 35, 10, 55, 35, 20, 40, 30, 75, 30...
## $ R500    <int> 15, 15, 15, 5, 15, 15, 20, 5, 15, 5, 20, 10, 5, 20, 20, ...
```

```
## $ R1000 <int> 20, 20, 15, 10, 5, 20, 20, 10, 15, 10, 15, 10, 5, 20, 10...
## $ R2000 <int> 20, 20, 25, 5, 5, 15, 20, 15, 5, 5, 5, 20, 5, 15, 15, 20...
## $ R4000 <int> 30, 30, 40, 25, 65, 35, 25, 20, 25, 30, 20, 30, 25, 65, ...
```

First of all the first column has to be eliminated, as it only stores the ID values. It would be a major mistake taking this column into account when performing PCA, as the results would be heavily biased.

```
# Eliminate first column
drugdata <- drugdata[-1]
```

One can also see that all the remaining values are of class integer. This is important to know, as correlation matrices can only be calculated using either integers or numerical values. After selecting the relevant variables, it is also useful to check for missing values. This is done by using the apply-function, which outputs the amount of missing values for each column.

```
# Check for missing values
drugdata %>% apply(2, function(x){
  sum(is.na(x))
})
```

```
## L500 L1000 L2000 L4000 R500 R1000 R2000 R4000
##      0      0      0      0      0      0      0      0
```

Hence, there arent any missing values presented in the data. Even though the PCA has no distributional assumptions regarding the data it might still be useful to get an idea of the distribution of the variables.

```
# Extract 6-point statistic
summary(drugdata)
```

```
##      L500      L1000      L2000      L4000
## Min.   : 5.0    Min.   : 5.0    Min.   : 5.00   Min.   : 5.00
## 1st Qu.: 5.0    1st Qu.:10.0   1st Qu.:10.00  1st Qu.:23.75
## Median :10.0    Median :15.0   Median :15.00  Median :35.00
## Mean   :12.2    Mean   :14.5   Mean   :17.00  Mean   :36.35
## 3rd Qu.:20.0    3rd Qu.:20.0   3rd Qu.:21.25  3rd Qu.:50.00
## Max.   :30.0    Max.   :35.0   Max.   :60.00  Max.   :85.00
##      R500      R1000      R2000      R4000
## Min.   : 5.0    Min.   : 5.0    Min.   : 5.0    Min.   : 5.00
## 1st Qu.: 5.0    1st Qu.:10.0   1st Qu.:10.0    1st Qu.:20.00
## Median :10.0    Median :15.0   Median :15.0    Median :30.00
## Mean   :12.4    Mean   :14.3   Mean   :16.6    Mean   :36.35
## 3rd Qu.:15.0    3rd Qu.:20.0   3rd Qu.:20.0    3rd Qu.:45.00
## Max.   :40.0    Max.   :35.0   Max.   :50.0    Max.   :90.00
```

As one would expect, the range of the variables increase as the dose increases. Taking into consideration that the minimum of each variable is equal to five, this means that due to the higher dose there might be at least some participants whose percentage drop in body pathogens was quite high compared to the participants who got a lower dose. This can also be derived when looking at the mean values or the outlier resistant median values. This might give rise to the assumption that higher dose is more effective.

3. Assumptions

When dealing with PCA there is basically one assumption which has to be met, namely the variables have to be either mean centered or standardized. As all variables are measured in the same scale (from 0% to 100%) the variance observed is not artificial, but important information which has to be considered. For example, if higher drug doses lead to a higher variance then this information has to be included in the model. Therefore

only mean centering is applied when performing PCA. This is done in the upcoming section when using the `prcomp`-function with the `center` parameter set to `TRUE`, which is the default setting in this function.

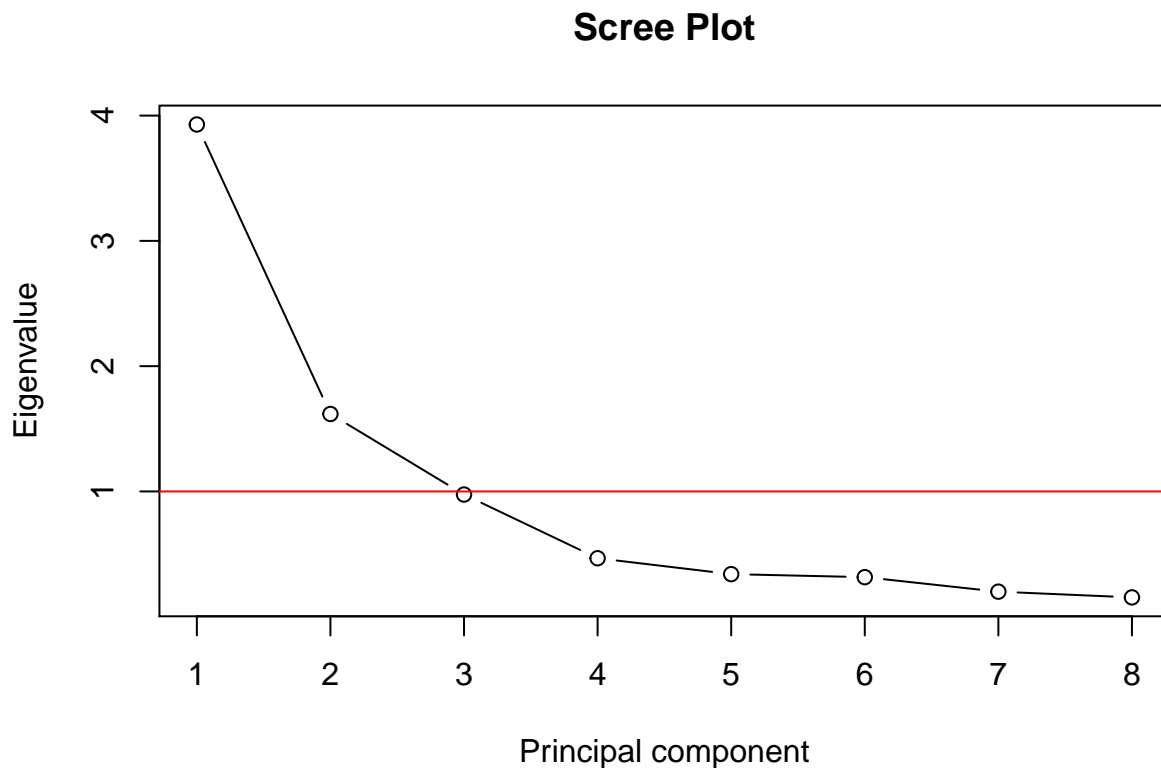
4. Method

Before applying the method the first task is to identify the appropriate number of principal components that effectively summarize the data. Hence, we will decide based on two commonly used procedures. First by visually analyzing the scree plot and second by performing Horn's Parallel procedure.

The scree plot is just the visualization of the eigenvalues in decreasing order. Hence, in the following chunk the eigenvalues are extracted from the correlation matrix of the data and the eigenvalues are plotted. The red line highlights the threshold value of one.

```
# Calculate eigenvalues of correlation matrix
evalcor <- eigen(cor(drugdata))$values

# Scree plot
plot(evalcor, xlab = "Principal component", ylab = "Eigenvalue",
     type = "b", main = "Scree Plot")
abline(h=1,col="red")
```

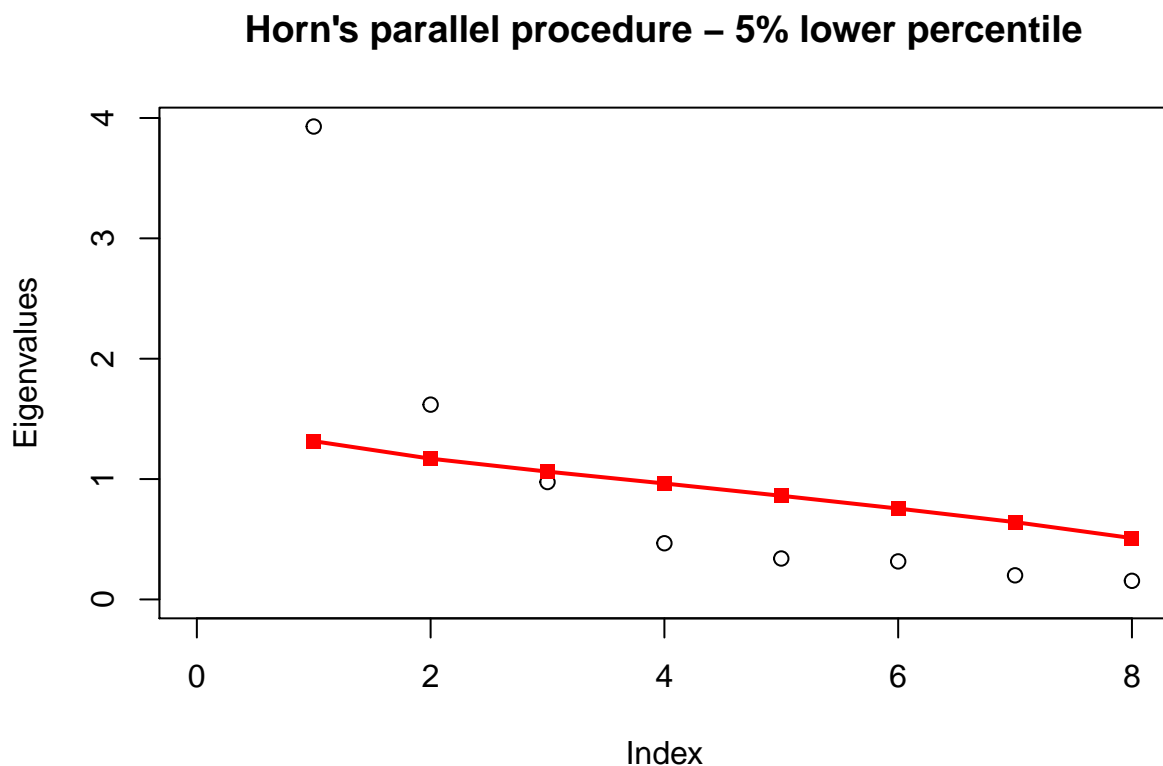


The first characteristic to find would be the so called elbow, which separates the mountain from the debris. As the slope has almost an exponential form, it is difficult to see whether such a distinction can be made. The second criterion to check is to identify which eigenvalues lay above the threshold value of one and which lay below. The first and the second principal component are clearly above the threshold value. The third principal component is a borderliner, as it's value of 0.9753248 lies slightly below the threshold value.

Using Horn's Parallel procedure might give a clearer understanding, especially regarding the borderline value. Here, eigenvalues associated with many simulated uncorrelated normal variables are computed. A principal component is then retained, if its eigenvalue is bigger than the 95th percentile of the simulated eigenvalue distribution.

```
# Perform Horn's Parallel Procedure
ap <- parallel(subject = nrow(drugdata), var = 8, rep = 1000, cent = 0.05)

# 5% lower percentile on eigenvalues plot
plot(evalcor, xlim=c(0,ncol(drugdata)), ylim = c(0,max(evalcor)),
     ylab = "Eigenvalues")
lines(ap$eigen$qevpea, type = "o", pch = 15, lwd = 2, col = "red")
title(main = "Horn's parallel procedure - 5% lower percentile")
```



When plotting the 5% lower percentile together with the eigenvalues of the correlation matrix, one can clearly see that the first two eigenvalues are clearly above the line and the third one below. Hence, the first two principal component are taken into account in the further analysis.

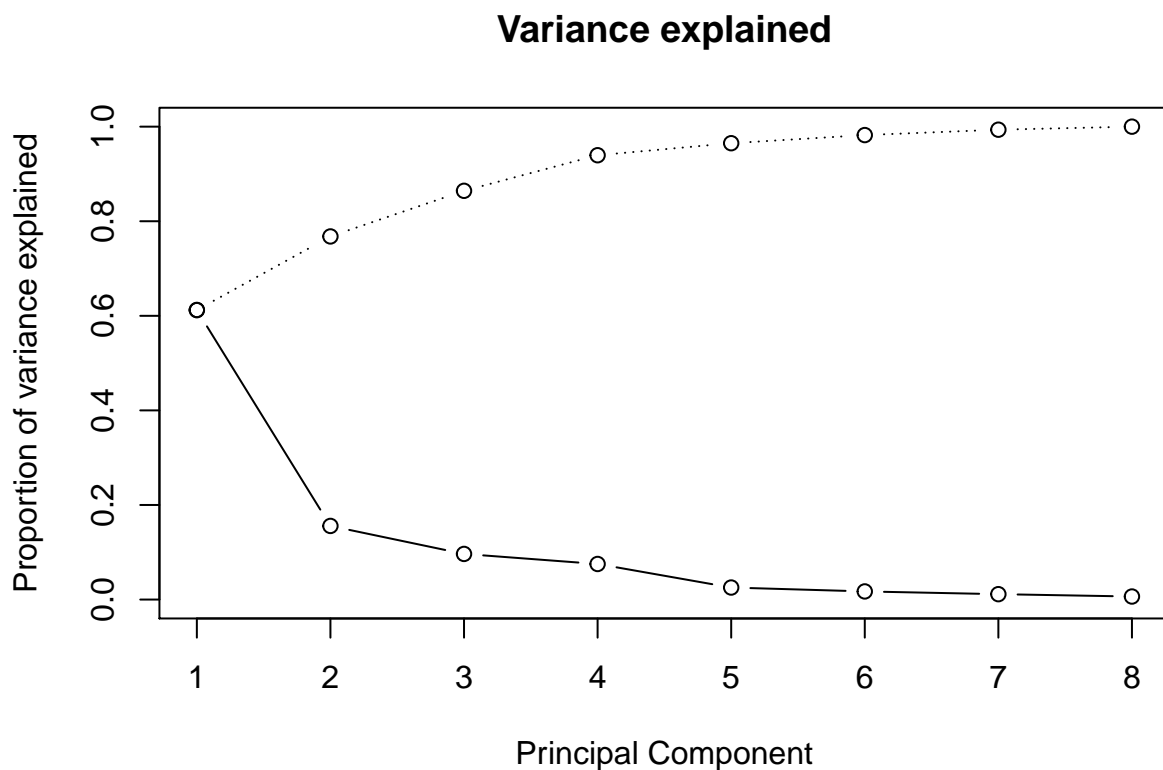
The PCA is performed using the `prcomp` function of the `stats` package. The default value of the parameter `center` is `TRUE` and `FALSE` for the `scale` parameter, which is exactly the configuration required. Therefore, the data is included and the results are saved in the object `pr.out`. Note that the analysis is done using the covariance matrix, if the `scale` parameter is set to `FALSE`.

```
# Perform PCA
pr.out <- prcomp(drugdata)

# Investigate importance of components
summary(pr.out)
```

```
## Importance of components:
##               PC1      PC2      PC3      PC4      PC5      PC6
## Standard deviation 26.5856 13.4059 10.55302 9.31934 5.41904 4.45328
## Proportion of Variance 0.6122 0.1557 0.09646 0.07523 0.02544 0.01718
## Cumulative Proportion 0.6122 0.7679 0.86434 0.93957 0.96500 0.98218
##               PC7      PC8
## Standard deviation 3.6274 2.72292
## Proportion of Variance 0.0114 0.00642
## Cumulative Proportion 0.9936 1.00000

# Variance explained plot
sum <- summary(pr.out)
plot(sum$importance[2,], xlab = "Principal Component",
     ylab = "Proportion of variance explained", type = "b",
     main = "Variance explained", ylim = c(0,1))
lines(cumsum(sum$importance[2,]), type = "b", lty = 3)
```



It is interesting to see how much variance is explained using the first two principal components. By using the summary function one can see that 61.22% of the variance is explained by the first principal component and 76.22% is explained when including the second principal component.

5. Interpretation

In order to be able to interpret the data in the two-dimensional eigenvector space the correlation coefficients between the principal components and original variables, which are also called structural loadings, have to

be calculated and interpreted. As the model is performed using the covariance matrix, the eigenvalues and eigenvectors of that matrix are needed. The structural loadings are then calculated as follows:

$$\text{corr}(PC_j, x_k) = \frac{e_{jk}\sqrt{\lambda_j}}{s_k}. \quad (1)$$

```
# Extract eigenvectors and eigenvalues of covariance matrix
evec <- pr.out$rotation
eval <- pr.out$sdev^2

# Structural loadings calculation
varnames <- names(drugdata)
strloadings <- matrix(nrow = 2, ncol = ncol(drugdata), byrow = T)
colnames(strloadings) <- varnames
rownames(strloadings) <- c("PC1", "PC2")

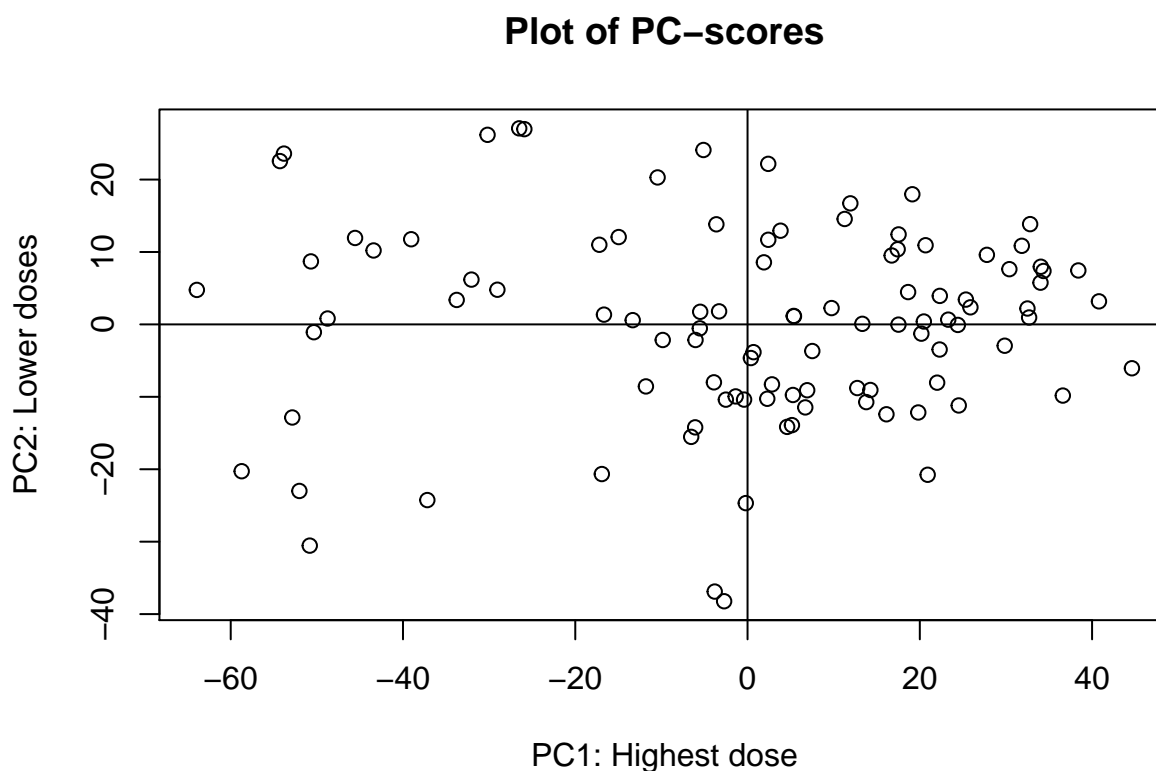
for( n in 1:length(rownames(strloadings)) ){
  for( m in 1:nrow(evec) ){
    strloadings[n,m] <- evec[m,n] * sqrt(eval)[n] / sd(drugdata[,m])
  }
}
strloadings
```

```
##          L500      L1000      L2000      L4000      R500      R1000
## PC1 -0.3463650 -0.3832302 -0.5401066 -0.91916518 -0.2470347 -0.370277
## PC2 -0.6141806 -0.7051469 -0.6835020  0.07945561 -0.5230198 -0.653630
##          R2000      R4000
## PC1 -0.4884279 -0.9022052
## PC2 -0.5403942  0.2359788
```

The structural loadings show a clear distinction between the highest dose of both drugs (L4000 and R4000) and the lower doses (L/R500 to L/R2000). The two highest doses are very highly negatively correlated with the first principal component. Only the absolute value of the correlation between L2000 and PC1 is also higher than 0.5 but since L2000 has a higher absolute value of correlation with PC2 and the correlation with PC1 (0.5401) is only just over 0.5 one can safely assume that PC1 mainly represents L/R4000. One possible interpretation is that the drugs themselves are very similar and only the difference in the dosage leads to significantly different effects. The most logical labels for the two PCs that are used are:

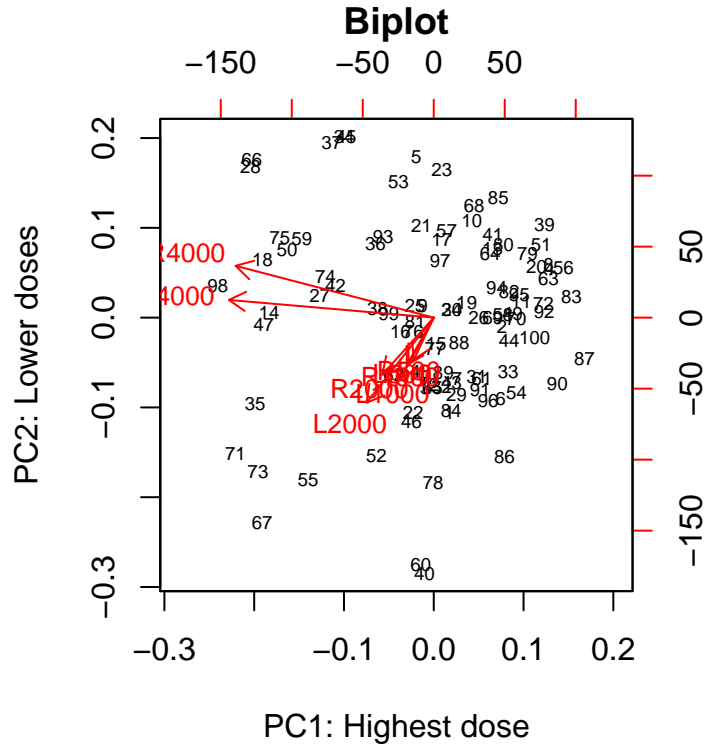
- PC1: Highest dose (4000)
- PC2: Low doses (500-2000).

```
# PC scores plot
plot(pr.out$x[,1], pr.out$x[,2], xlab = "PC1: Highest dose",
     ylab = "PC2: Lower doses", main = "Plot of PC-scores")
abline(v=0, h=0)
```



The plot of PC-scores enables us to discriminate between four kinds of patients: Patients, who responded well to both drugs (upper right), patients who only responded well to high doses (lower right), patients who only responded well to lower doses (upper left) and patients who did not respond well to any of the doses (lower left). A good response to a drug is supposed when the patient's mean pathogen drop is higher than the mean pathogen drop. One can also see that the patients who did not respond well to the highest dose (left from $x = 0$) have a higher variance of pathogen drop level, i.e. the points on the left of the plot are more spread out than on the right.

```
# Biplot
biplot(pr.out, choices = c(1,2), cex=c(0.6,0.8), xlab = "PC1: Highest dose",
       ylab = "PC2: Lower doses", main = "Biplot")
```



The biplot supports the assumption that the doses make a larger difference in percentage drop than the kind of drug (i.e. R or L). This is visible through the angles between the vectors that represent the variables. R4000 and L4000 are highly correlated with each other but almost uncorrelated with the rest of the doses for both drugs (orthogonal vectors mean a correlation of 0 because $\cos(90) = 0$).

The vectors for R/L4000 are clearly longer than those for the rest of the drugs and doses. One can conclude that R/L4000 include a higher ratio of the explained variance than the others. This is supported by the fact that the first PC explains roughly 61% of the variance and it almost only contains R/L4000.

Assignment 1: Task 2 - EFA

Group 11

16.10.2018

0. Load Libraries

```
library(psych)
```

1. Problem Statement

The correlation matrix illustrates the information about anxiety symptoms which are regarded as the reflection to estimate the overall anxiety related to screening exercises for several commonly occurring medical conditions. A reasonable and accurate exploratory factor analysis can be regarded as a method to identify a small number of common but unobservable factors and solve the initial questions. This paper will focus on the relationship between each factor and determine a reasonable number of factors that captures the information available in the data. A reasonable and accurate exploratory factor analysis can help researchers identify a small number of common but unobservable factors and solve the initial questions. The second section includes the description of the data and the visualisation of the correlation matrix. Then, the assumptions which can be applied to this case are demonstrated. Furthermore, exploratory factor analysis is implemented with fundamental algorithms of the method. The last section focuses on the interpretation the solution that explained the optimal factors.

2. Descriptive Statistics

There are some preparations of the data which are done before implementing exploratory factor analysis (EFA).

```
# Import data
corr <- read.delim("data/screening.txt", header = TRUE, sep=" ", dec = ".", skipNul = FALSE)
```

The first should be eliminated because it only contains X_name values. It cannot be applied to the EFA as the results will be heavily biased.

```
# Eliminate first column
corr <- subset(corr, select = -c(X_name_))
```

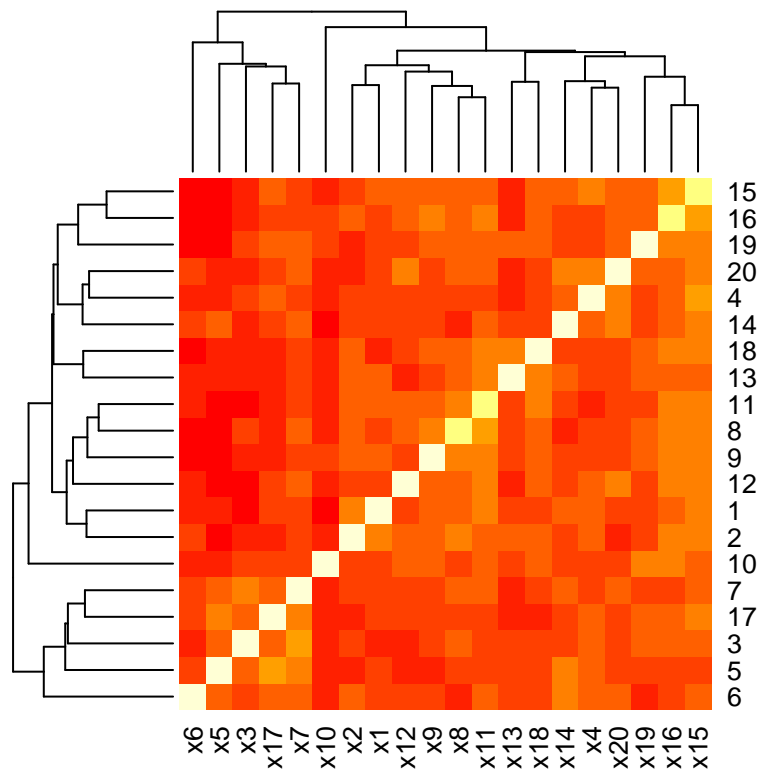
The following function processes the supplied lower correlation matrix into the desired format.

```
m <- matrix(NA, 20, 20)
m[lower.tri(m, diag=TRUE)] <- 1:10

makeSymm <- function(m) {
  m[upper.tri(m)] <- t(m)[upper.tri(m)]
  return(m)
}
```

```
corr <- makeSymm(corr)

corr <- makeSymm(corr)
corr<- as.matrix(corr)
heatmap(corr)
```



According to the heatmap, lighter colors represent a higher correlation between corresponding two variables. Variable x4 is highly correlated to variable x15, and the correlation between x5 and x10 is relatively weak. Most of the variables are equally high correlated.

3. Assumptions

When dealing with exploratory factor analysis, there is one fundamental assumption which should be satisfied, which is that the factors are not correlated, implying an orthogonal model. Additionally, the data should be standardized, in others words, the variables have to be either mean centered and standardized. Since we are given a correlation matrix as input, this assumption can be regarded as true. Moreover, there should be a linear relationship between each variable used in this case. This means that the variables need to be at least reasonably related to each other, otherwise there will be no difference between the number of factors and the number of initial variables. For example, if there is no linear relationship between 20 variables in this case, exploratory factor analysis will be meaningless. Therefore, we assume that the data is standardized and there exists linear relationship between each variable.

```
# Perform Kaiser's MSA to evaluate appropriateness of data
KMO(corr)
```

```
## Kaiser-Meyer-Olkin factor adequacy
```

```
## Call: KMO(r = corr)
## Overall MSA = 0.95
## MSA for each item =
##   x1  x2  x3  x4  x5  x6  x7  x8  x9  x10 x11 x12 x13 x14 x15
## 0.95 0.94 0.94 0.95 0.89 0.95 0.95 0.96 0.97 0.97 0.96 0.96 0.95 0.96 0.96
##   x16 x17 x18 x19 x20
## 0.96 0.93 0.96 0.94 0.95
```

```
# Kaiser MSA = 0.95 > 0.8 --> appropriate data
```

4. Method

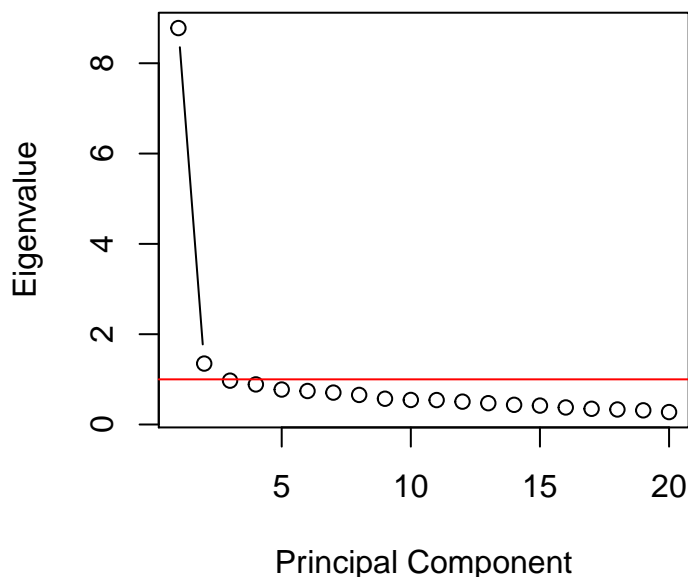
There are different methods of obtaining a factor model, such as principal component factoring, iterative principal components factoring or the maximum likelihood method. First, the number of relevant factors to be extracted has to be determined. This can be inferred from the eigenvalues λ_i and eigenvectors ϵ_i of the observed correlation matrix R^{obs} . There are a number of rules of thumb which can be applied to the computed values:

- Retain only those factors with an eigenvalue larger than 1 (Guttman-Kaiser rule)
- Make a scree-plot and extract the amount of factors before the knee point of the slope
- Horn's parallel procedure

```
# Define the amount of factors
eval <- eigen(corr)$values

plot(eval, xlab = "Principal Component", ylab = "Eigenvalue",
     type = "b", main = "Scree Plot")
abline(h=1,col="red")
```

Scree Plot



Considering the Scree Plot we should only retain the first factor. However, we also took the Kaiser-Guttman rule into account and decided to keep two factors. Factor analysis can be executed with different factoring methods to extract the latent variables. We used principal factoring and maximum likelihood. They both employ iterative approaches of estimating the correlation matrix from the observed Matrix. Principal factoring assumes that the initial communalities are 1, meaning that there is no error at the starting point. In each iteration, these values then replace the diagonal in the correlation matrix which is used to recompute the set of factors. Maximum likelihood assumes a normal distribution of the dataset and iteratively adjusts distribution parameters to better fit the model to the observed data.

As there is an infinite number of different factoring solutions, rotations are applied to find the best possible interpretation of the model. For orthogonal models, which is one of our assumptions for this task, the most common procedures are varimax and quartimax. The latter focuses on identifying factor structure such that all variables have fairly high loadings on a few factors and have near zero loadings on the other factors. Varimax on the other hand tries to maximize the variance of loadings for each factor, such that every factor has high loadings on a few variables and low loadings for the other variables.

```
# Perform Factor Analysis
fa.out.ml <- fa(r = corr, nfactors = 2, fm="ml", rotate = "varimax", residuals = TRUE, SMC=FALSE,
               max.iter = 10)

fa.out.pa <- fa(r = corr, nfactors = 2, fm="pa", rotate = "varimax", residuals = TRUE, SMC=FALSE)
fa.out.pa

## Factor Analysis using method = pa
## Call: fa(r = corr, nfactors = 2, rotate = "varimax", residuals = TRUE,
##        SMC = FALSE, fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      PA1  PA2   h2   u2 com
## x1  0.54 0.31 0.39 0.61 1.6
## x2  0.60 0.24 0.41 0.59 1.3
## x3  0.29 0.51 0.35 0.65 1.6
## x4  0.47 0.47 0.44 0.56 2.0
## x5  0.12 0.70 0.51 0.49 1.1
## x6  0.28 0.45 0.28 0.72 1.7
## x7  0.38 0.64 0.55 0.45 1.6
## x8  0.67 0.26 0.51 0.49 1.3
## x9  0.66 0.22 0.48 0.52 1.2
## x10 0.45 0.22 0.25 0.75 1.5
## x11 0.70 0.28 0.58 0.42 1.3
## x12 0.58 0.31 0.43 0.57 1.5
## x13 0.52 0.30 0.36 0.64 1.6
## x14 0.40 0.53 0.45 0.55 1.9
## x15 0.68 0.42 0.64 0.36 1.7
## x16 0.73 0.28 0.61 0.39 1.3
## x17 0.31 0.65 0.52 0.48 1.4
## x18 0.62 0.28 0.47 0.53 1.4
## x19 0.57 0.33 0.43 0.57 1.6
## x20 0.51 0.42 0.44 0.56 1.9
##
##
##      PA1  PA2
## SS loadings      5.60 3.48
## Proportion Var    0.28 0.17
## Cumulative Var    0.28 0.45
## Proportion Explained 0.62 0.38
## Cumulative Proportion 0.62 1.00
```

```
##
## Mean item complexity = 1.5
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are 190 and the objective function was 9.5
## The degrees of freedom for the model are 151 and the objective function was 0.89
##
## The root mean square of the residuals (RMSR) is 0.04
## The df corrected root mean square of the residuals is 0.04
##
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##
## Correlation of (regression) scores with factors    PA1  PA2
## Multiple R square of scores with factors          0.91 0.86
## Minimum correlation of possible factor scores      0.83 0.73
##                                                     0.66 0.47
```

5. Interpretation of Solution

As it is the objective of EFA to explore the variable structure, the labeling of meaningful latent factors is ambiguous. Maximum likelihood and principal factoring achieved very similar results, both generating the same root mean square of the residuals (RMSR) of 0.04 and the same variable groupings implied by the factor loadings. The RMSR measure indicates the sum of remaining error components, which is desired to be small. The factor model explained a cumulative variance of 0.45. However, the RMSR is the most important measure for the quality of the model.

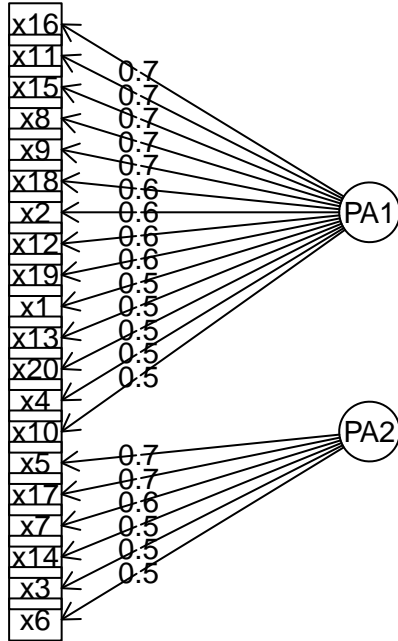
In regard of rotation approaches, varimax provided the better possible factor explanation as 14 variables were loading higher on the first factor and 6 on the second. Quartimax resulted in all but one variable loading on the first factor, which we did not consider a useful split.

We thought of two possible interpretations of the structure. Variables loading strong on the first factor PA1 included statements such as “Lack of confidence during tests” (x1), “Heart beating fast during tests” (x18), “Screening bothers me” (x12) which we labeled as “self-confidence during test situations” or “high intense anxiety”. The latent factor of the other group of variables (PA2) consisted of statements such as “Thinking about test results” (x3), “The harder I try to contain myself, the less assured I get” (x6), “Defeat myself during tests” (x14) and can be called “self-manipulative thoughts during test situations” or “low intense anxiety”.

A full structural diagram displaying the variable groupings can be seen below.

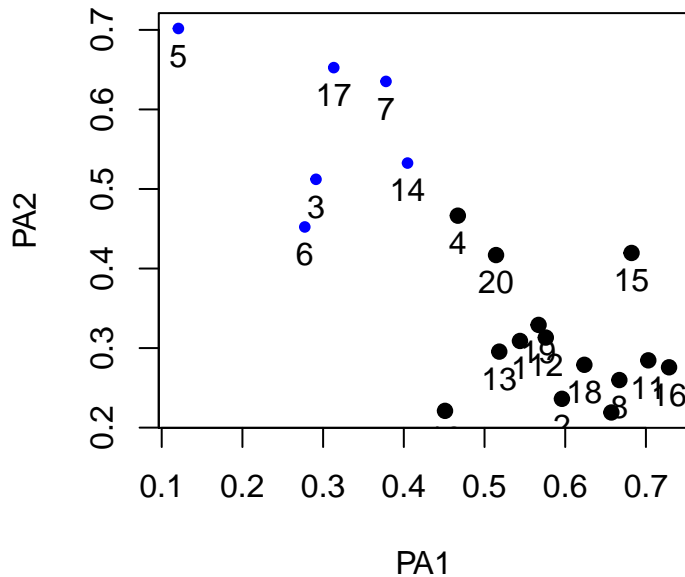
```
# Structural diagram
fa.diagram(fa.out.pa, main = "Structural diagram")
```

Structural diagram



```
# Plot of factor loadings
plot(fa.out.pa, title = "Factor loadings")
```

Factor loadings



From the factor loadings plot we can graphically verify that the variables load higher on one of the factors and very few (x4 and x20) share similarly high loadings with both. We can conclude that this factor model captures useful information about the structure of the measured variables. Specifically one group of variables shares the component PA1 of “self-confidence during test situations” or “high intense anxiety” and the other one can be regarded as “self-manipulative thoughts during test situations” or “low intense anxiety”.