Assignment 1

Group 11

The Date

1. Problem Statement: PCA

The file drugsrecovery.txt provides data on recovery status of patients after administration of different doses of two different drugs, L and R. The recovery status is measured as a percentage drop in body pathogens pre- and post-drug administration. A larger percentage drop implies better recovery. The administering of the drugs, at each of the dose levels, is assumed to not interfere with recovery levels for previous and/or subsequent dose(s). 100 participants took part in the study. Variables L500 to R4000, respectively refer to drug L at a dose level of 500 micrograms to drug R at 4000 micrograms. The ID is a patient's hospital identification number. Perform a principal components analysis to: I. Determine the appropriate number of components that can be used to effectively summarize the information in the data. Explain how you settled on the reported number of components. II. If possible, provide an interpretation for the chosen sample principal components III. Comment on the (bi-)plot for the first two components

2. Descriptive Statistics

check missing values and impute (not needed here), take away ID column, check if all columns are int

```
drugs <-read.delim("data/drugsrecovery.txt", header = TRUE, sep="",dec = ".")
sub_drugs <- subset(drugs, select = -c(ID))
sub_drugs <- data.frame(sub_drugs)
str(sub_drugs)
```

```
## 'data.frame':    100 obs. of  8 variables:
##  $ L500 : int  15 10 10 10 10 20 15 5 15 10 ...
##  $ L1000: int  20 15 15 15 10 20 15 5 15 10 ...
##  $ L2000: int  25 5 30 5 5 20 15 5 15 5 ...
##  $ L4000: int  30 15 30 5 25 5 35 10 55 35 ...
##  $ R500 : int  15 15 15 5 15 15 20 5 15 5 ...
##  $ R1000: int  20 20 15 10 5 20 20 10 15 10 ...
##  $ R2000: int  20 20 25 5 5 15 20 15 5 5 ...
##  $ R4000: int  30 30 40 25 65 35 25 20 25 30 ...
```

2. Assumptions

? scaling? princomp vs prcomp?

The function princomp() uses the spectral decomposition approach.

The functions prcomp() and PCA()[FactoMineR] use the singular value decomposition (SVD).

According to R help, SVD has slightly better numerical accuracy. Therefore, prcomp() is the preferred function.

3. Method

PCA explained?

```r
pr.out <- prcomp(sub_drugs, scale = TRUE)
summary(pr.out)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.9822 1.2721 0.9876 0.68321 0.58317 0.56204
## Proportion of Variance 0.4911 0.2023 0.1219 0.05835 0.04251 0.03949
## Cumulative Proportion  0.4911 0.6934 0.8153 0.87368 0.91619 0.95568
##                            PC7     PC8
## Standard deviation     0.44734 0.39303
## Proportion of Variance 0.02501 0.01931
## Cumulative Proportion  0.98069 1.00000
```

- extract values based on different approaches: extract P C 0 s to explain a given percentage of the variance • scree plot: plot the eigenvalues in decreasing order and find the elbow that distinguishes the mountain from the debris • retain only P C 0 s with eigenvalue larger than one (only for standar- dized data) • Horn’s Parallel procedure: compute eigenvalues associated with many simulated uncorrelated normal variables - retain the ith PC if the corresponding eigenvalue is larger than the 95th percentile of the distribution of the ith largest eigenvalue of the random data (same idea as the previous rule but taking random variation into account)

```r
# Eigenvalues
eval <- pr.out$sdev^2
# First two eigenvalues are bigger than one
```

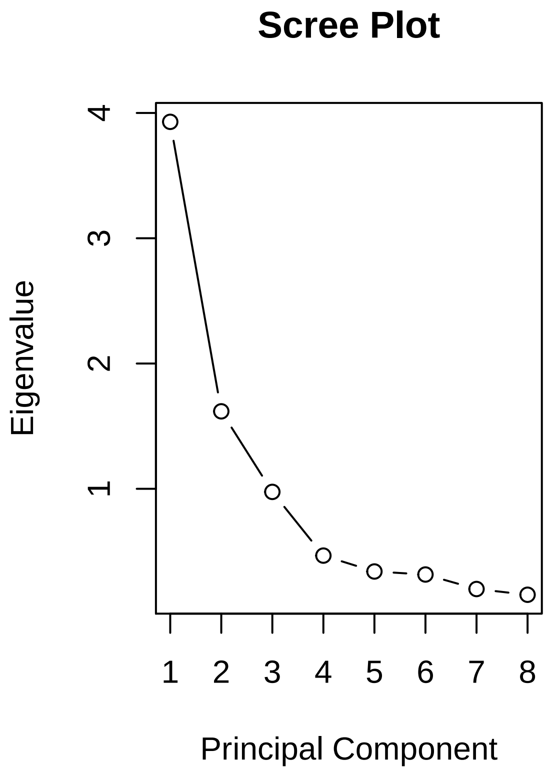
```r
# Eigenvectors
evec <- pr.out$rotation
```

```r
# Standard deviations and proportion of variance
summary(pr.out)
```

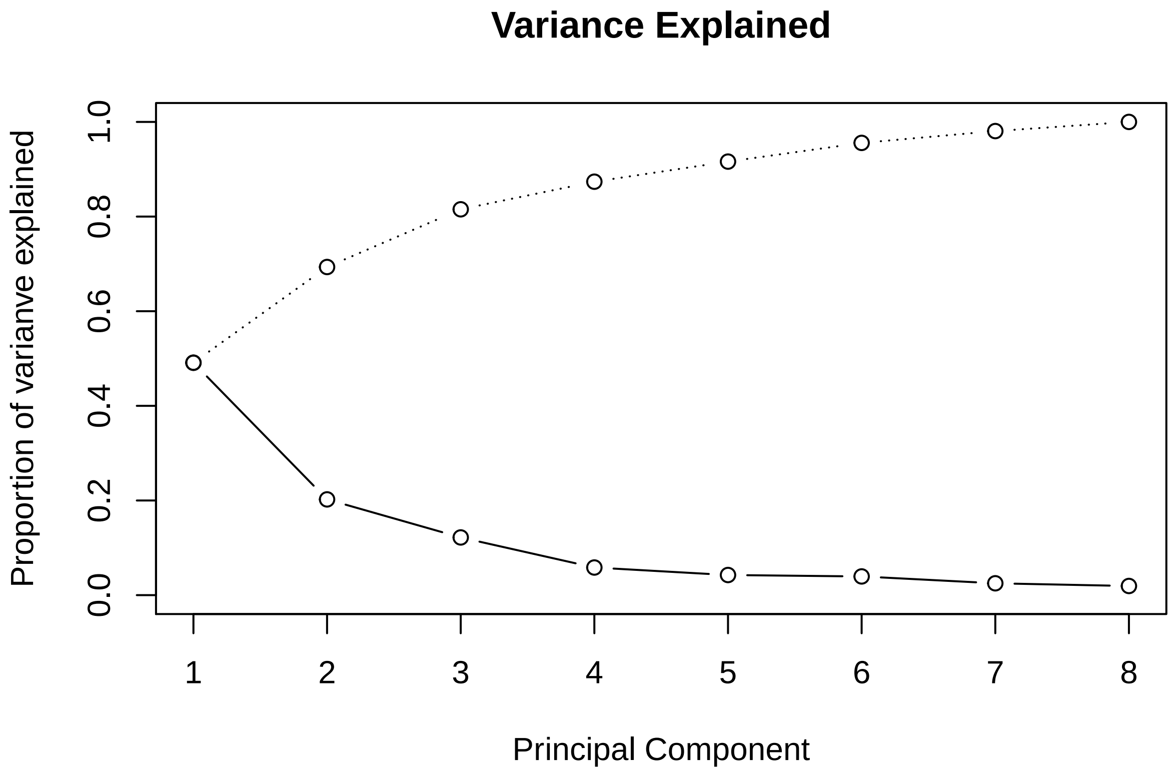
```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     1.9822 1.2721 0.9876 0.68321 0.58317 0.56204
## Proportion of Variance 0.4911 0.2023 0.1219 0.05835 0.04251 0.03949
## Cumulative Proportion  0.4911 0.6934 0.8153 0.87368 0.91619 0.95568
##                            PC7     PC8
## Standard deviation     0.44734 0.39303
## Proportion of Variance 0.02501 0.01931
## Cumulative Proportion  0.98069 1.00000
```

```r
# First two principal components explain 69.34% of the variance
```

```r
# Scree Plot
par(mfrow = c(1,2))
plot(eval, xlab = "Principal Component", ylab = "Eigenvalue",
     type = "b", main = "Scree Plot")
```



```
# Variance Explained Plot
prop_varex <- eval / sum(eval)
plot(prop_varex, xlab = "Principal Component", ylab = "Proportion of varianve explained",
     type = "b", main = "Variance Explained", ylim = c(0,1))
lines(cumsum(prop_varex), type = "b", lty = 3)
```



```
par(mfrow = c(1,1))
```

4. Interpretation

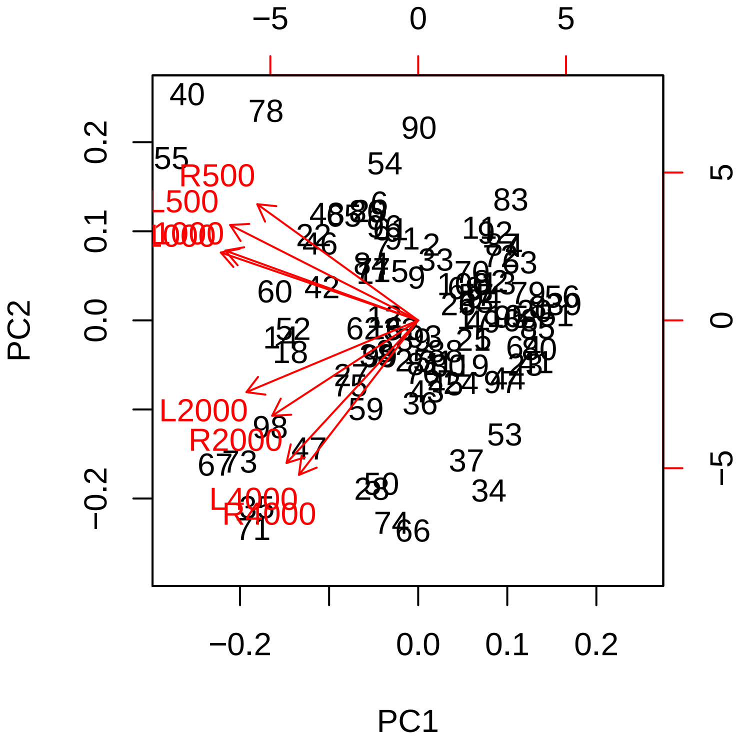
```
# Correlation coefficients between PC's and initial variables
varnames <- names(drugs)[2:ncol(drugs)]
corrcoef <- matrix(nrow = 2, ncol = ncol(drugs)-1, byrow = T)
colnames(corrcoef) <- varnames
rownames(corrcoef) <- c("PC1", "PC2")

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

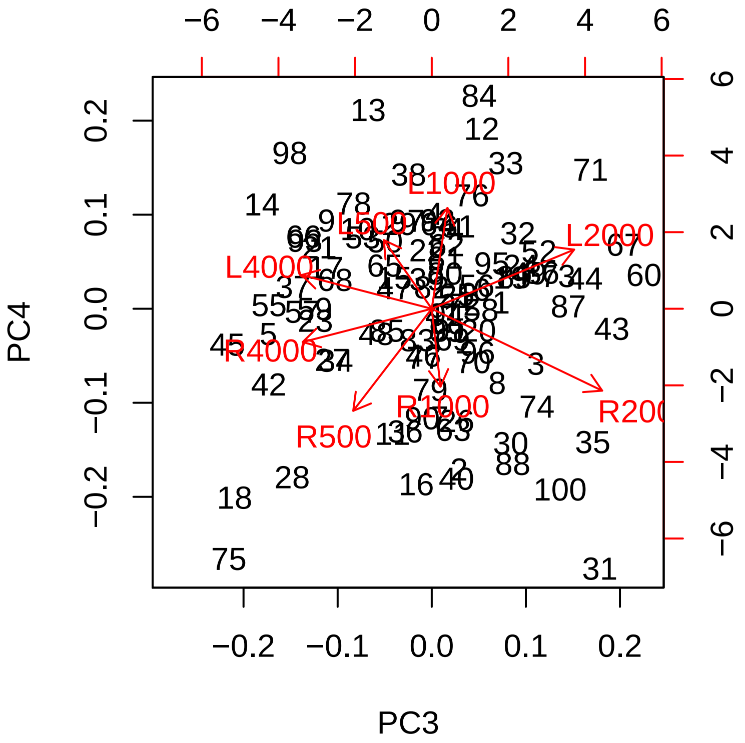
```
##           L500      L1000      L2000      L4000       R500      R1000
## PC1 -0.7950389 -0.8344760 -0.7262179 -0.5567047 -0.6803825 -0.8155068
## PC2  0.4032200  0.2868202 -0.3035224 -0.6031874  0.4910683  0.2948454
##          R2000      R4000
## PC1 -0.6175423 -0.5039101
## PC2 -0.4033411 -0.6532555
```

```
# First principal component is significantly negatively correlated
# with all variables
# Second component discriminates between L2000, L4000, R2000, R4000 on one hand
# (high dose) and L500, L1000, R500 and R1000 on the other hand (low dose)
```

```
biplot(pr.out)
```



```
biplot(pr.out, choices = c(3,4))
```



first plot shows two groupings of dosage amount independant of the drug

second plot barely shows any relevant information as there is barely any group seperation -> was to be expected since they explain small portion of the variance only

For the majority of the patients, both drugs had barely any effect on the percentage drop. Lower dose seems to be more effective compared to higher dose, as with a lower dose more patients had a percentage drop.

1. Problem Statement Task 2:

exploratory factor analysis: • explain the correlation structure among observed variables • try to find underlying dimensions that can explain the observed correlations • example: the correlation between scores on mathematics, statis- tics and physics exams can be explained because they all measure somehow quantitative intelligence

1. State the problem
2. Descriptive Statistics (to check data, to find outliers)
3. Test (or at least state) the assumptions of the method, if any
4. Conduct the method (describe in more detail the “best” approach you have found)
5. Interpret the solution
6. Compare the results briefly with alternative solutions, if any
7. Conclusion

Descriptive Statistics

```
library(psych)
```

```
## Warning: package 'psych' was built under R version 3.4.4
```

```
corr <-read.delim("data/screening.txt", header = TRUE, sep="",dec = ".", skipNul = FALSE)
corr <- subset(corr, select = -c(X_name_))

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)
```

Assumptions of Methods

assuming standardized data and factors + uncorrelated factors

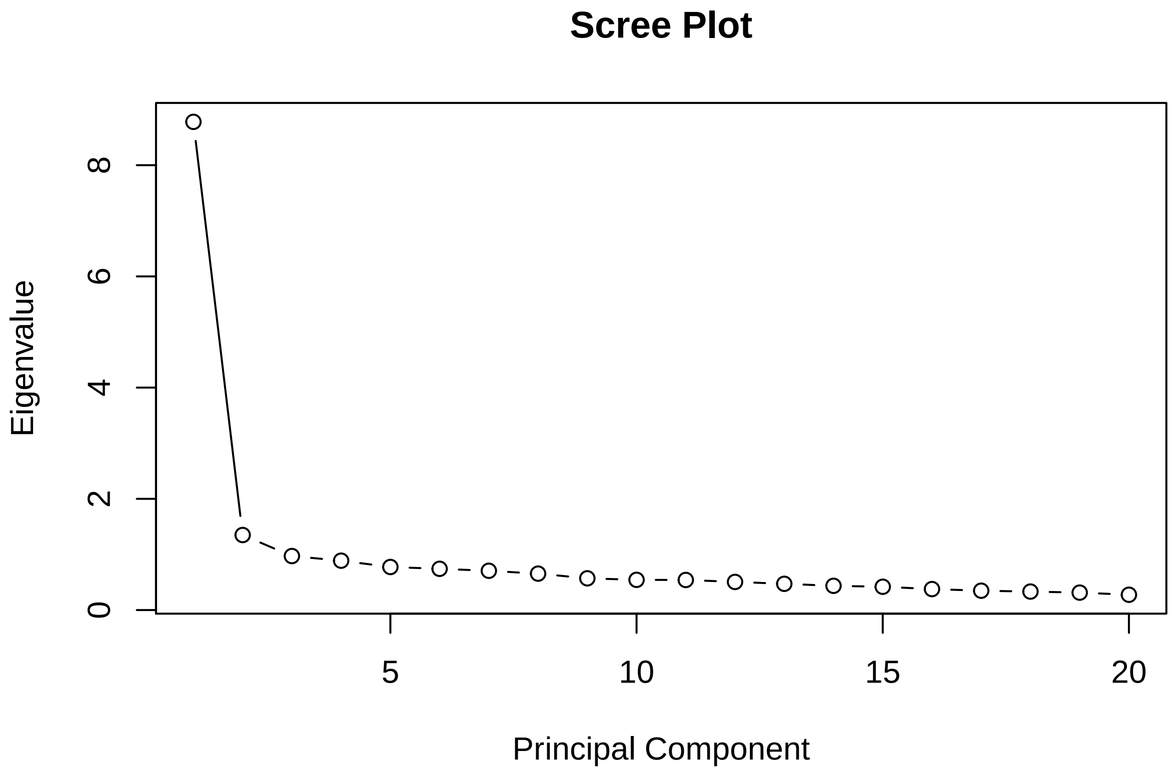
```
# Perform Kaiser's MSA to evaluate appropriateness of data
KMO(corr)
```

```
## 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
```

Method

```
# Define the amount of factors
eval <- eigen(corr)$values
plot(eval, xlab = "Principal Component", ylab = "Eigenvalue",
     type = "b", main = "Scree Plot")
```



```
# Choose 2 factors

# CHECK USING HORNS

# Perform Factor Analysis
fa.out <- fa(r = corr, nfactors = 2, rotate = "varimax")
fa.out
```

```
## Factor Analysis using method =  minres
## Call: fa(r = corr, nfactors = 2, rotate = "varimax")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      MR1  MR2   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 0.50 0.50 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
##
##                        MR1  MR2
## SS loadings           5.59 3.49
## 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
##                                                    MR1  MR2
## Correlation of (regression) scores with factors   0.91 0.86
## Multiple R square of scores with factors          0.83 0.73
## Minimum correlation of possible factor scores     0.66 0.47
```

5. Interpretation of Solution

RMSR = 0.04

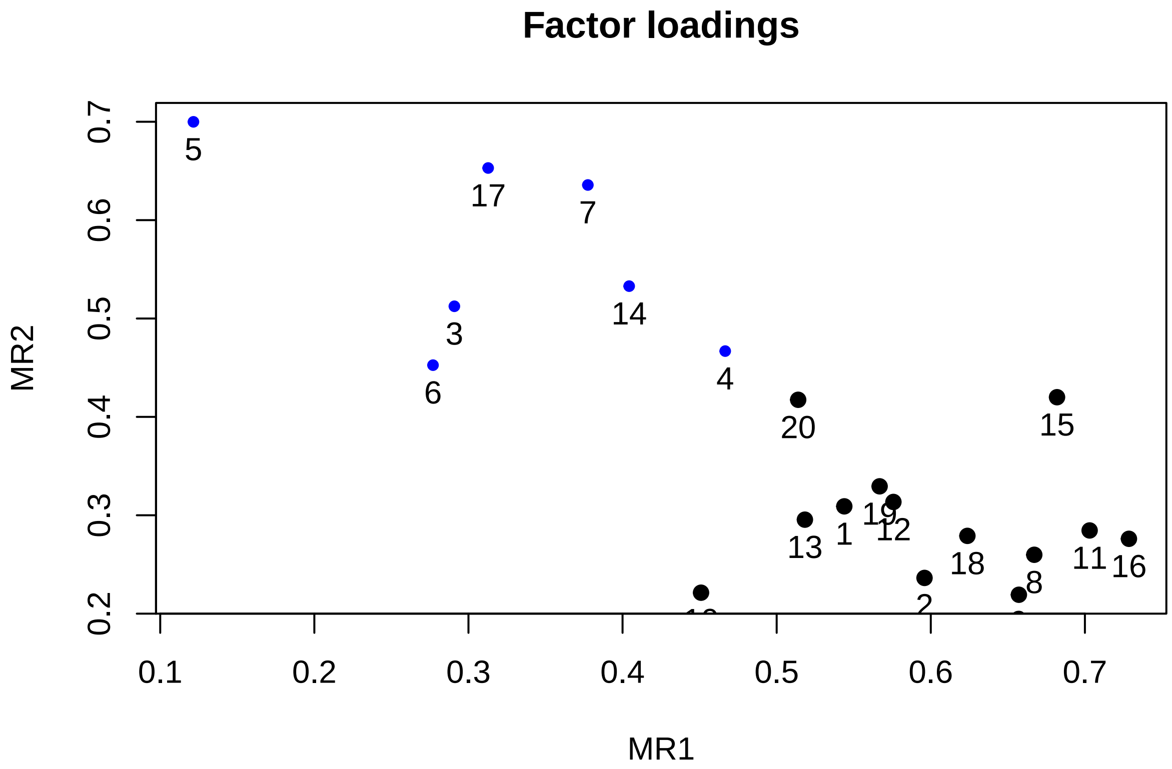
First factor represents: 1,2,4,8,9,10,11,12,13,15,16,18,19,20

Second factor represents: 3,5,6,7,14,17

First factor might be called "High intense anxiety"

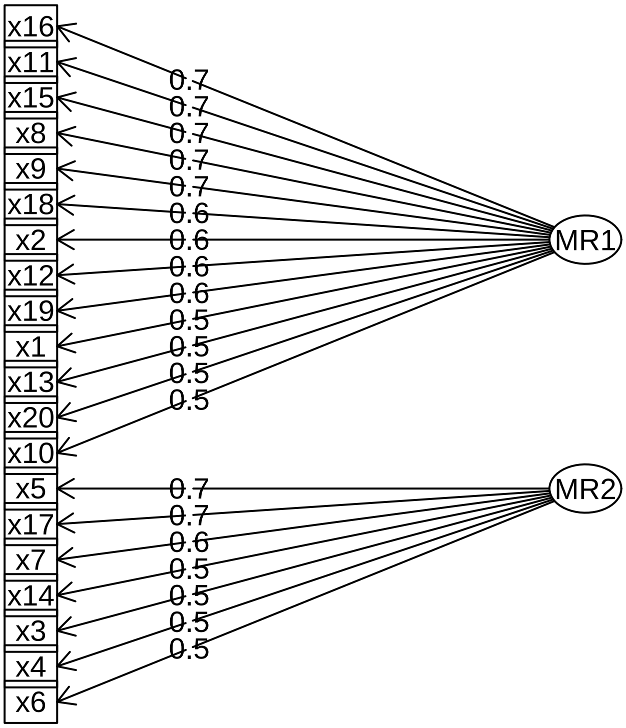
Second factor might be called "Low intense anxiety"

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



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

Structural diagram


x16
x11
x15
x8
x9
x18
x2
x12
x19
x1
x13
x20
x10
x5
x17
x7
x14
x3
x4
x6
0.7
0.7
0.7
0.7
0.7
0.6
0.6
0.6
0.6
0.5
0.5
0.5
0.5
MR1
0.7
0.7
0.6
0.5
0.5
0.5
0.5
MR2