

FA08/07

Hazel Zhao

30/10/2020

```
knitr::opts_chunk$set(echo= TRUE )
```

Group variables by their domains. The analysis is conducted within each domain.

```
library("dplyr")
library("tidyverse")
library(haven)
library(dineq)
library(haven)
library("corrplot")
library(openxlsx)
library(psych)
library(polycor)
```

```
# Group variables by their domains. The analysis is conducted within each domain.
```

```
classified_list=list()
classified_list[[1]]=c("hehelf4")
#Self-rated general health
```

```
classified_list[[2]]=c("hemobwa4", "hemobsi4", "hemobch4", "hemobcs4", "hemobcl4",
  "hemobst4", "hemobre4", "hemobpu4", "hemobli4", "hemobpi4")
#Mobility: 10 items concerning difficulty walking, getting up, lifting, etc.
```

```
classified_list[[3]]=c("headldr4", "headlwa4", "headlba4", "headlea4", "headlbe4", "headlwc4")
#Disabilities: Activities of Daily Living (ADLs)
```

```
classified_list[[4]]=c("headlma4", "headlpr4", "headlsh4", "headlph4", "headlme4", "headlmo4", "headlhg4")
#Disabilities: Instrumental Activities of Daily Living (IADLs)
```

```
classified_list[[5]]=c("hedimbp4", "hediman4", "hedimmi4", "hedimhf4", "hedimar4", "hedimdi4", "hedimst4")
```

```

#Physician diagnosed conditions:heart problem

classified_list[[6]]=c("hediblu4", "hedibas4", "hedibar4", "hedibos4","hedibca4",
                      "hedibpd4", "hedibps4", "hedibad4", "hedibde4")
#Physician diagnosed conditions:others

classified_list[[7]]=c("heeye4", "hehear4")
#Eyesight and hearing - self-reported

classified_list[[8]]=c("hefla4", "hefrac4","heji4","mmpain4")
#bone

classified_list[[9]]=c("psceda4", "pscedb4", "pscedc4", "pscedd4", "pscede4","pscedf4","pscedg4", "pscedh4")
#mental

classified_list[[10]]=c("cfdatd4", "cfdatm4", "cfdaty4", "cfday4", "cfmem4","cflisenqs4","cflisdqs4","cflisds4")
#cognitive function


# Basic profiling

#separate deficits variables and demographic variables
demog=c(64:66)
candi=c(1:63,83,84,87)
load("tdata1.Rda")
wave4data=tdata1
wave4data=wave4data[wave4data$r4agey>=60,candi] #remove people under 60 years old
total_missing=sum(complete.cases(wave4data))/nrow(wave4data)
frailty_missing=sum(complete.cases(wave4data[,1:63]))/nrow(wave4data)
#casp19_missing=sum(complete.cases(wave4data[,64:82]))/nrow(wave4data)
demo_missing=sum(complete.cases(wave4data[,demog]))/nrow(wave4data)
print(paste("completed cases for all:",total_missing)) # for all variables

## [1] "completed cases for all: 0.782343772141136"

print(paste("frailty items completed cases:",frailty_missing))# for 62 deficits

## [1] "frailty items completed cases: 0.797931132209154"

#print(paste("quality of life completed cases:",casp19_missing))
print(paste("demographic items completed cases:",demo_missing)) # age, gender, income, wealth

## [1] "demographic items completed cases: 0.972934674790988"

mydata=wave4data[complete.cases(wave4data),]
dim(mydata)

## [1] 5521 66

```

```
save(mydata,file ="mydata.Rda")
```

When exploring missing data, it is important to address the reason of missingness. Except for deficits of pain whilst walking, missingness of the others can be assumed missing completely at random (MCAR), because the percentage of missing value is not large, ignorance of the missingness should not affect the analysis result with bias. However, the missingness of “pain whilst walking” is special, it is related to other variables, such as the participants being unable to walk, therefore this variable is not applicable, this is been measure in other deficits, in this case, the missingness is still considered ignorable and missing at random (MAR). As a result, participants having missing data is deleted and not including in the sample, which means 78.2% eligible participants are included in the study.

Put the annotation on variables

```
load("mydata.Rda")
item_an=read.xlsx("deficits ano.xlsx")
for (i in 1:62){
  attr(mydata[,i+1],"label")=item_an[i,2]
}
```

Calculate frailty score for each person – The frailty index

The frailty index (FI) consisted of 62 deficits, including 10 mobility difficulty deficits, 13 disability deficits, 16 physician diagnosed diseases, eyesight and hearing, self-rated general health, 4 movement deficits, 8 CES-D depression items and 8 cognitive function items.

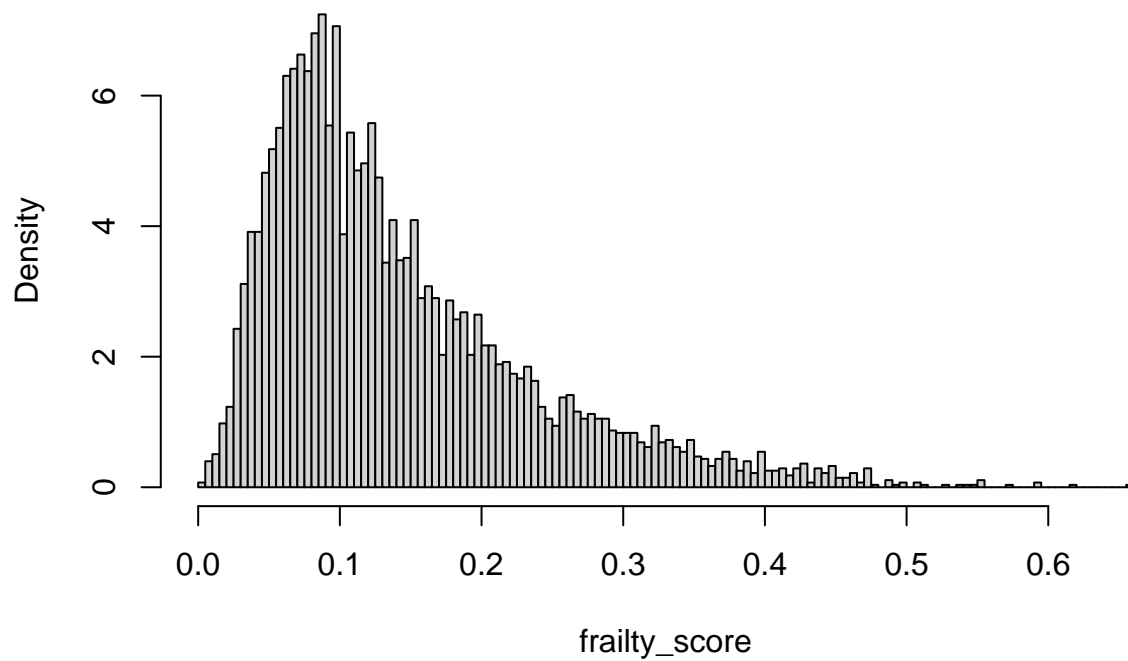
It is a gamma distribution, which is proved using a goodness of fit test with p-value < 0.01.

```
frailty_score=apply(mydata[2:63],1,mean)
print(summary(frailty_score))
```

```
##      Min.   1st Qu.   Median     Mean  3rd Qu.    Max.
## 0.004032 0.073871 0.116129 0.142025 0.187258 0.658871
```

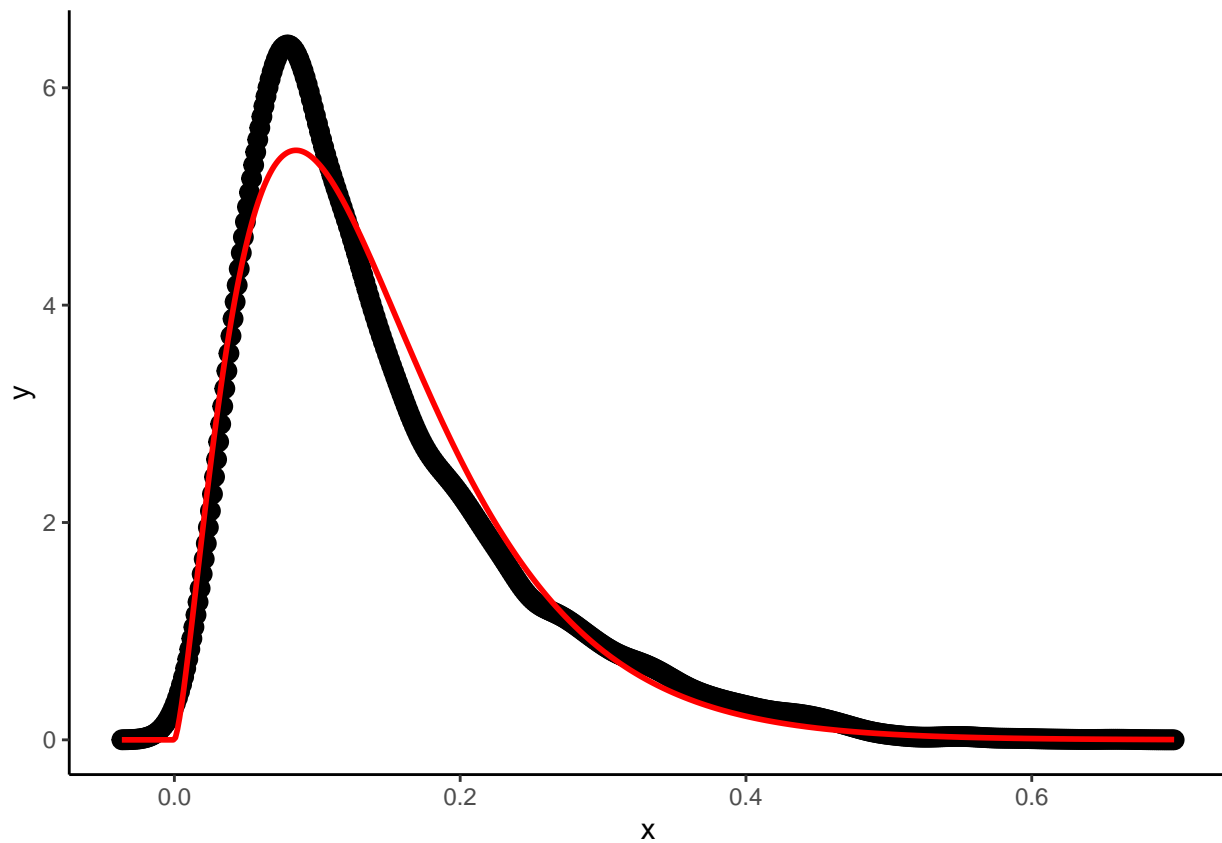
```
hist(frailty_score,freq=F,breaks = 100)
```

Histogram of frailty_score

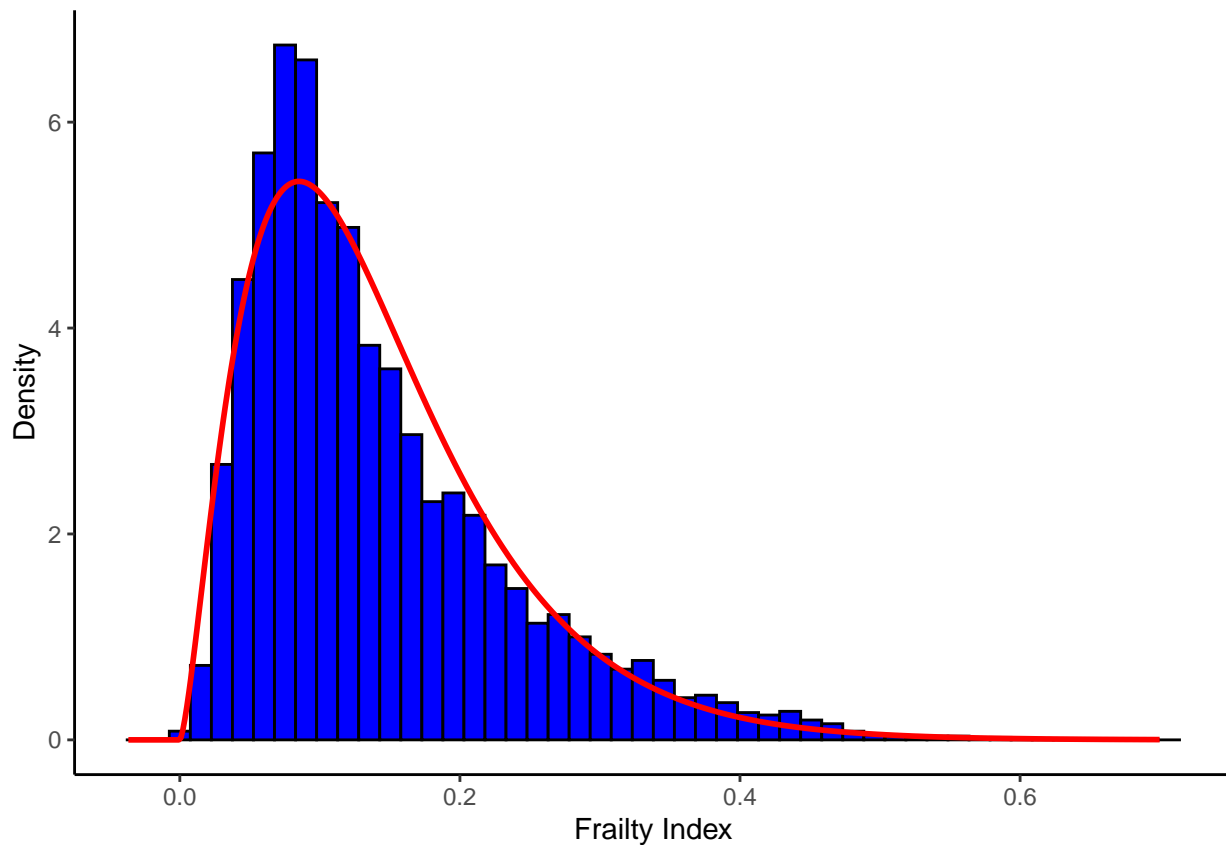


```
library(ggplot2)
library(MASS)
library(goftest)

den=density(frailty_score)
dat <- data.frame(x = den$x, y = den$y)
fit.params= fitdistr(frailty_score,"gamma",lower=c(0,0)) #fit gamma distribution
ggplot(data = dat, aes(x = x,y = y)) +
  geom_point(size = 3) +
  geom_line(aes(x=dat$x, y=dgamma(dat$x,fit.params$estimate["shape"], fit.params$estimate["rate"])), col = "red", size = 1)
  theme_classic()
```



```
ggplot(data = dat) +  
  geom_histogram(data = as.data.frame(frailty_score), aes(x=frailty_score, y=..density..), bins = 50, fill="black", color="black") +  
  geom_line(aes(x=dat$x, y=dgamma(dat$x, fit.params$estimate["shape"], fit.params$estimate["rate"])), color="red") +  
  theme_classic() +  
  labs(x="Frailty Index", y="Density")
```



```
summary(frailty_score)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.004032 0.073871 0.116129 0.142025 0.187258 0.658871
```

```
sd(frailty_score)
```

```
## [1] 0.09413437
```

```
# The scale and shape of "gamma" distriction
scale=sd(frailty_score)^2/mean(frailty_score)
shape=mean(frailty_score)^2/sd(frailty_score)^2
```

Factor analysis is conducted for the items within domains. For all domains, Principal axis method is used to extract factors. For model with more than one factor, factor rotation method is oblique to generate simpler structure.

Note: generally, if we need conduct analysis repeatedly, making code reusable (define functions) is better practice. However, this analysis is pretty exploratory(e.g.,dropping items based on the result of the analysis). Being flexible is optimal.

1. Mobility items

```

var_list=unlist(classified_list[2])
temp=data.frame(mydata[,var_list])
mobility_difficulty=matrix(nrow=length(var_list),ncol=6,dimnames = list(var_list,c("Mean","SD","Min","Max")))

for (j in 1:4){
  for (i in 1: length(var_list)){
    tab=table(temp[,i])
    mobility_difficulty[i,5]=tab[[1]]/sum(tab)
    mobility_difficulty[i,6]=tab[[2]]/sum(tab)
    mobility_difficulty[i,1]=mean(temp[,i])
    mobility_difficulty[i,2]=sd(temp[,i])
    mobility_difficulty[i,3]=min(temp[,i])
    mobility_difficulty[i,4]=max(temp[,i])
  }
}
mobility_difficulty=data.frame(mobility_difficulty)
mobility_difficulty

```

```

##           Mean      SD Min Max percent.of.0. percent.of.1.
## hemobwa4 0.08911429 0.2849344  0  1      0.9108857      0.08911429
## hemobsi4 0.10777033 0.3101182  0  1      0.8922297      0.10777033
## hemobch4 0.23999275 0.4271174  0  1      0.7600072      0.23999275
## hemobcs4 0.35899294 0.4797486  0  1      0.6410071      0.35899294
## hemobcl4 0.11827567 0.3229635  0  1      0.8817243      0.11827567
## hemobst4 0.36243434 0.4807469  0  1      0.6375657      0.36243434
## hemobre4 0.09563485 0.2941165  0  1      0.9043652      0.09563485
## hemobpu4 0.14888607 0.3560084  0  1      0.8511139      0.14888607
## hemobli4 0.21083137 0.4079358  0  1      0.7891686      0.21083137
## hemobpi4 0.04618728 0.2099095  0  1      0.9538127      0.04618728

```

```

for (i in 1:ncol(temp)){
  print(attr(temp[,i],"label"))
}

```

```

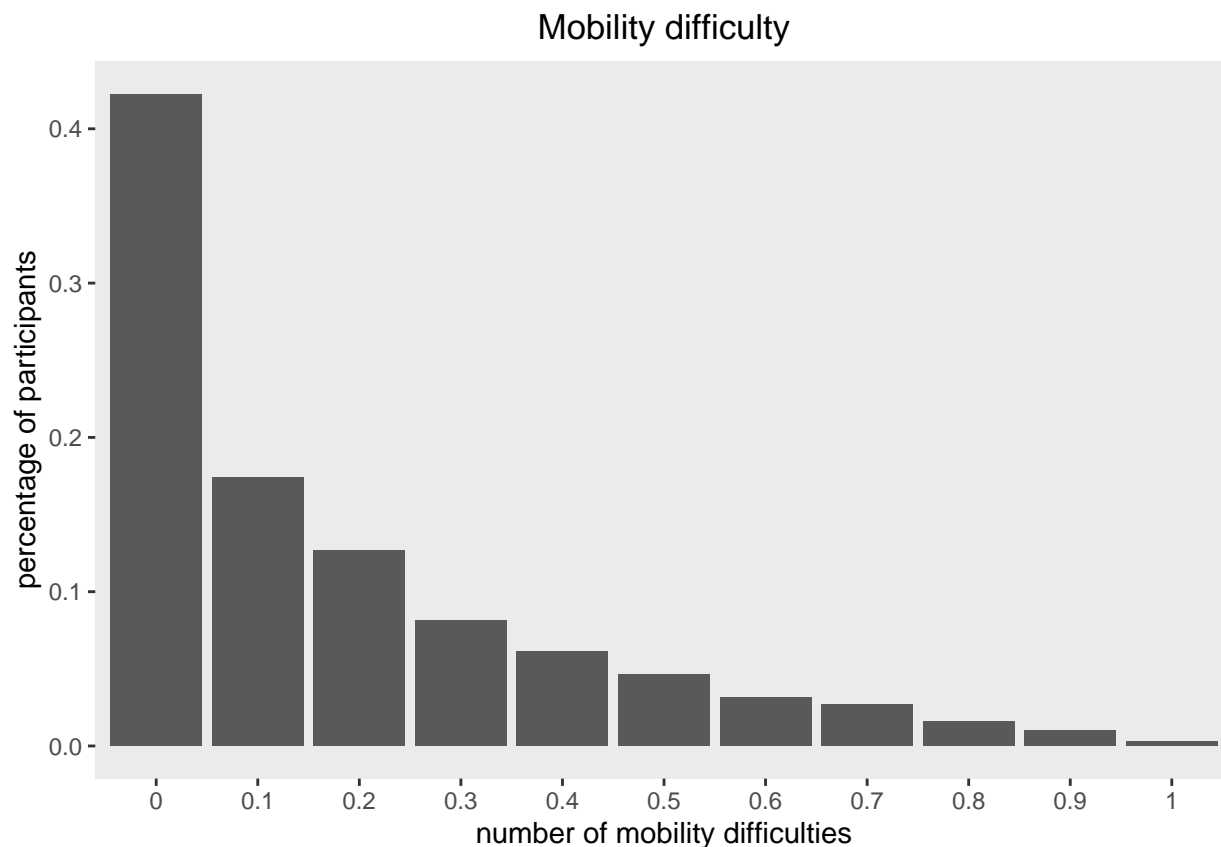
## [1] "hemobwa Mobility: difficulty walking 100 yards "
## [1] "hemobsi Mobility: difficulty sitting 2 hours "
## [1] "hemobch Mobility: difficulty getting up from chair after sitting long periods "
## [1] "hemobcs Mobility: difficulty climbing several flights stairs without resting "
## [1] "hemobcl Mobility: difficulty climbing one flight stairs without resting "
## [1] "hemobst Mobility: difficulty stooping, kneeling or crouching "
## [1] "hemobre Mobility: difficulty reaching or extending arms above shoulder level "
## [1] "hemobpu Mobility: difficulty pulling or pushing large objects "
## [1] "hemobli Mobility: difficulty lifting or carrying weights over 10 pounds "
## [1] "hemobpi Mobility: difficulty picking up 5p coin from table "

```

```

mobility_diff=apply(mydata[,var_list],1,sum)/10
ggplot(NULL,aes(x=as.factor(mobility_diff)))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="number of mobility difficulties",y="percentage of participants",title = "Mobility difficulty")
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())

```



```
# tetrachoric coefficient for binary data
```

```
library(REdaS)
```

```
## Warning: package 'REdaS' was built under R version 4.0.2
```

```
## Loading required package: grid
```

```
corre_mobility=tetrachoric(temp,smooth=T)
```

```
corre_mobility=pmax(corre_mobility$rho,t(corre_mobility$rho))
```

```
print("coefficient of Mobility Items")
```

```
## [1] "coefficient of Mobility Items"
```

```
corre_mobility
```

```
##           hemobwa4 hemobsi4 hemobch4 hemobcs4 hemobcl4 hemobst4 hemobre4
## hemobwa4 1.0000000 0.5537454 0.5645799 0.7461243 0.7768825 0.5882572 0.5229248
## hemobsi4 0.5537454 1.0000000 0.6547672 0.4433167 0.4857053 0.5344240 0.4437488
## hemobch4 0.5645799 0.6547672 1.0000000 0.5865537 0.5474130 0.6883603 0.4737388
## hemobcs4 0.7461243 0.4433167 0.5865537 1.0000000 0.7355800 0.6188949 0.4420761
```



```
## hemobcl4 0.7768825 0.4857053 0.5474130 0.7355800 1.0000000 0.5835148 0.4809174
## hemobst4 0.5882572 0.5344240 0.6883603 0.6188949 0.5835148 1.0000000 0.5039675
## hemobre4 0.5229248 0.4437488 0.4737388 0.4420761 0.4809174 0.5039675 1.0000000
## hemobpu4 0.7199845 0.5581504 0.5612772 0.6708021 0.6729455 0.5792772 0.6045991
## hemobli4 0.7189431 0.5492294 0.5574288 0.6564857 0.6727441 0.5599670 0.6085156
## hemobpi4 0.3976210 0.4360833 0.3878176 0.3477230 0.3564714 0.4674177 0.3711108
##          hemobpu4 hemobli4 hemobpi4
## hemobwa4 0.7199845 0.7189431 0.3976210
## hemobsi4 0.5581504 0.5492294 0.4360833
## hemobch4 0.5612772 0.5574288 0.3878176
## hemobcs4 0.6708021 0.6564857 0.3477230
## hemobcl4 0.6729455 0.6727441 0.3564714
## hemobst4 0.5792772 0.5599670 0.4674177
## hemobre4 0.6045991 0.6085156 0.3711108
## hemobpu4 1.0000000 0.8593260 0.4701617
## hemobli4 0.8593260 1.0000000 0.3683202
## hemobpi4 0.4701617 0.3683202 1.0000000
```

```
print ("factor adequacy test")
```

```
## [1] "factor adequacy test"
```

```
KMO(corre_mobility)
```

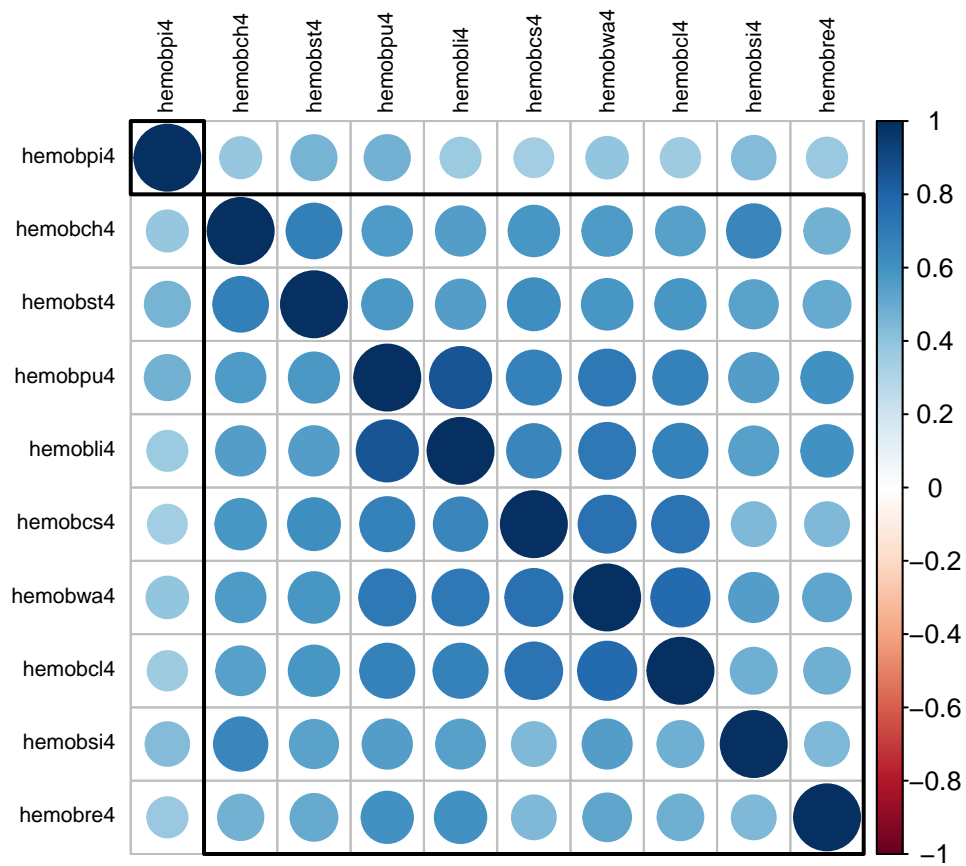
```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = corre_mobility)
## Overall MSA = 0.91
## MSA for each item =
## hemobwa4 hemobsi4 hemobch4 hemobcs4 hemobcl4 hemobst4 hemobre4 hemobpu4
##      0.93      0.91      0.89      0.92      0.93      0.93      0.96      0.89
## hemobli4 hemobpi4
##      0.88      0.90
```

```
bart_spher(corre_mobility)
```

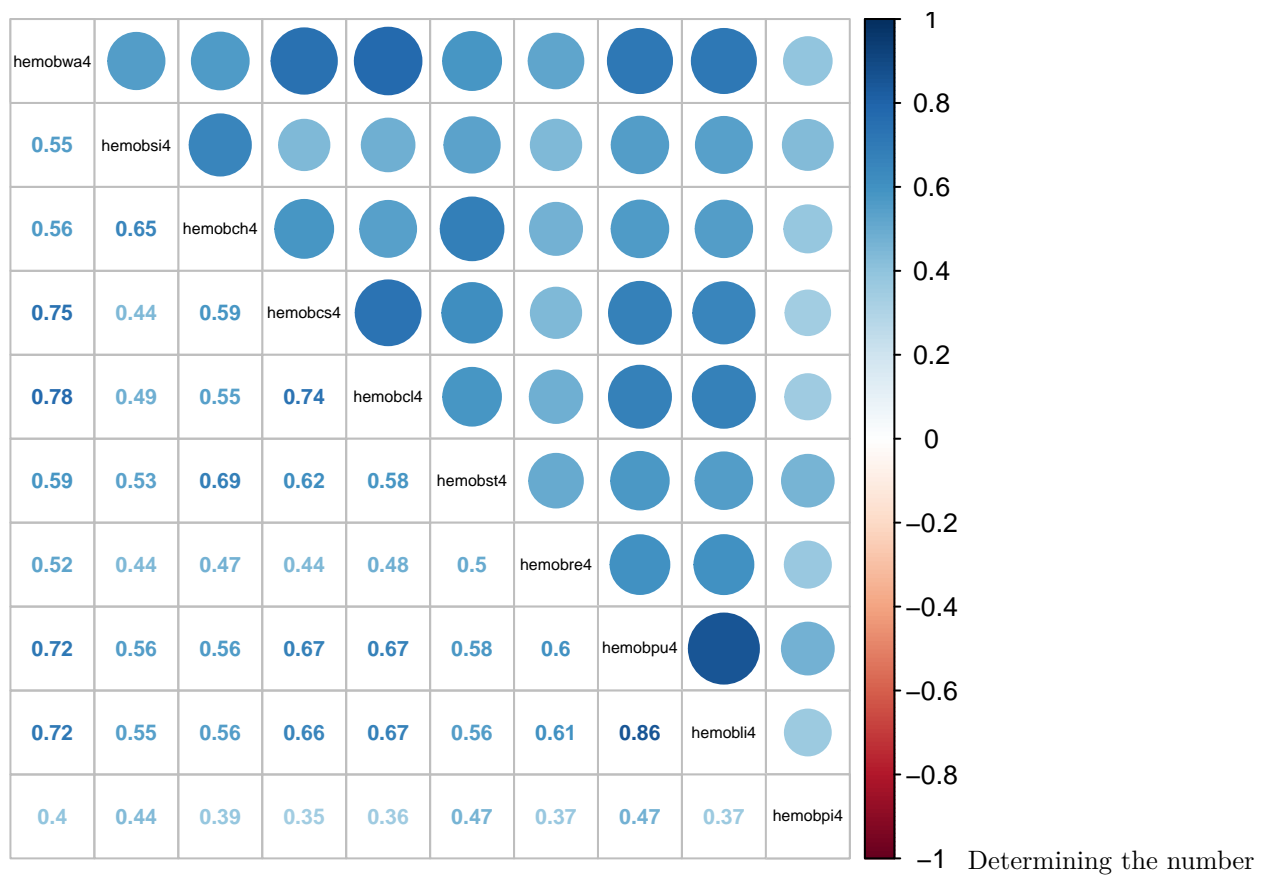
```
## Bartlett's Test of Sphericity
##
## Call: bart_spher(x = corre_mobility)
##
##      X2 = 210.772
##      df = 45
## p-value < 2.22e-16
```

The adequacy index of Kaiser-Meryer-Olkin (KMO) method is larger than 0.5. Additionally, Bartlett's Test of Sphericity is significant (the chi square's p-value < 0.01). Both tests indicate that the relationship between variables is strong enough, the data is sufficient to conduct factor analysis.

```
# draw coefficient
library("corrplot")
corrplot(corre_mobility, is.corr = TRUE, number.cex = .5, tl.col = 'black', order = 'hclust', addrect = 1)
```



```
corrplot.mixed(corre_mobility, is.corr = TRUE, number.cex = .7, tl.col = 'black',tl.cex = 0.5)
```



of factors:

There is one eigenvalue larger than 1, which means according to Kaiser's rule, 1 factor should be extracted. This is consistent with the scree plot. However, parallel analysis, the most commonly used method, shows 3 factors can be retained.

```
#eigenvalue
```

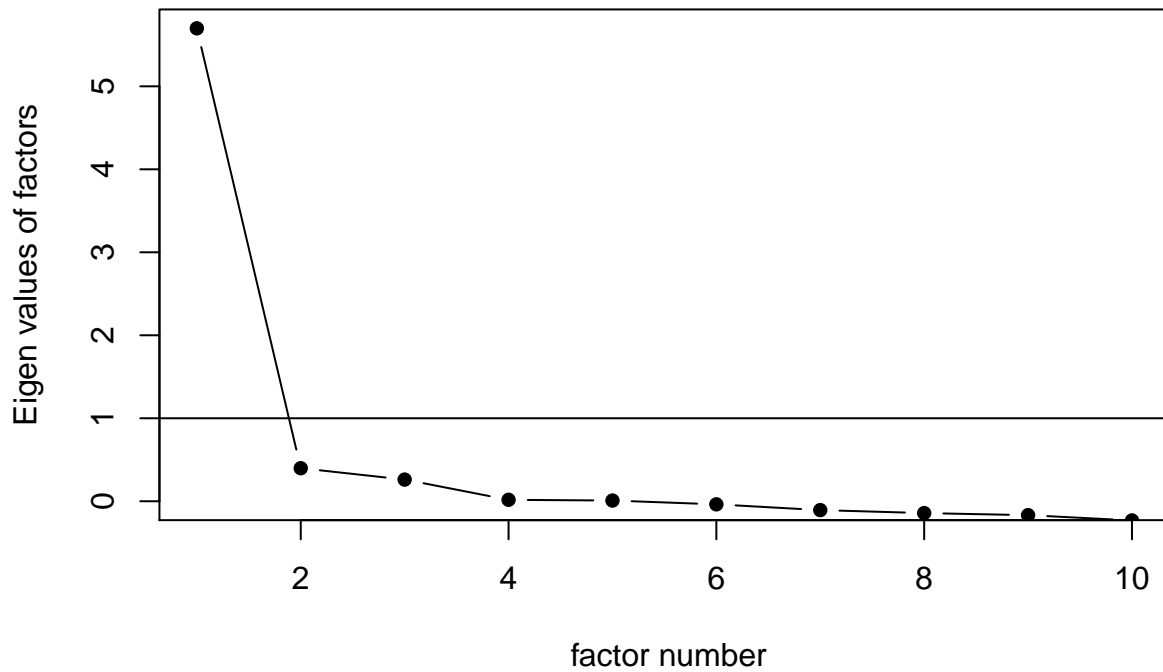
```
eigen.corre <- eigen(corre_mobility)
eigen.corre$values
```

```
## [1] 6.0951045 0.8787582 0.6895782 0.6196821 0.5040498 0.3622503 0.2737411
## [8] 0.2398587 0.2052504 0.1317267
```

```
#scree plot
```

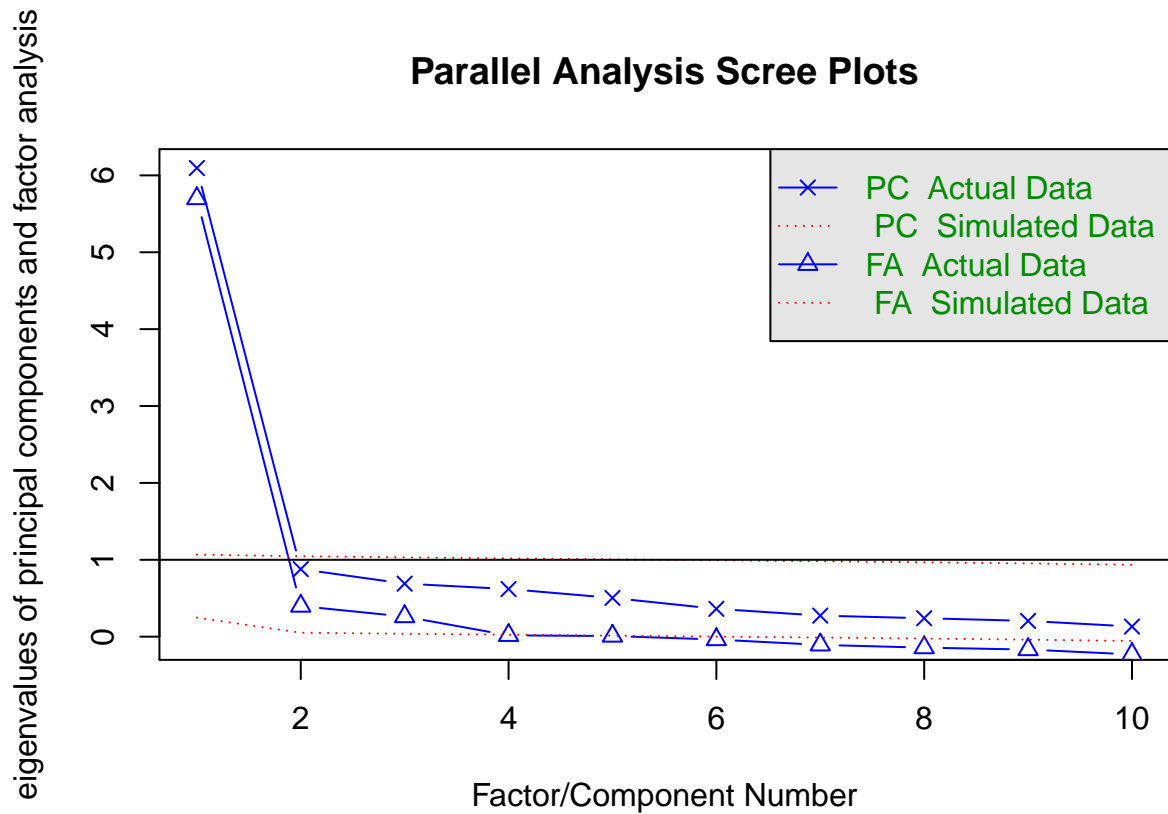
```
scree(corre_mobility,factors = T,pc=F)
```

Scree plot



```
#parallel analysis
fa.parallel(corre_mobility,n.obs= 5500,fm="pa",fa="both",n.iter = 1000)
```

Parallel Analysis Scree Plots



```
## Parallel analysis suggests that the number of factors = 3 and the number of components = 1
```

3 factors, difficulty related to upper limb, difficulty related to lower limb, other difficulty, are retained for the 10 Mobility items. Because all the items measures the same domain and strongly correlated, 1 factor model may also make sense. However one factor model does not add extra information for discovering the underlying structure of this domain since the observed variables have been categorized as mobility. Meanwhile parallel analysis and interpretability also support a 3 factors model and therefore the 3 factors model is preferred. These items are highly loaded on their own underlying factor, showing a good measure for the sub-aspect of mobility.

```
# fit model
```

```
mobility.fa<- fa(corre_mobility, nfactors = 3, n.obs=5520,rotate = 'oblimin',scores = T, fm = 'pa')
```

```
## Loading required namespace: GPArotation
```

```
print(mobility.fa,sort = T)
```

```
## Factor Analysis using method = pa
```

```
## Call: fa(r = corre_mobility, nfactors = 3, n.obs = 5520, rotate = "oblimin",  
## scores = T, fm = "pa")
```

```
## Standardized loadings (pattern matrix) based upon correlation matrix
```

```
##      item  PA1  PA3  PA2  h2  u2 com  
## hemobpu4   8  0.87  0.06  0.03  0.87  0.13  1.0  
## hemobli4   9  0.82  0.13 -0.02  0.83  0.17  1.1  
## hemobre4   7  0.54 -0.07  0.25  0.46  0.54  1.5  
## hemobcs4   4 -0.02  0.84  0.07  0.76  0.24  1.0  
## hemobcl4   5  0.10  0.77  0.02  0.75  0.25  1.0  
## hemobwa4   1  0.23  0.65  0.06  0.77  0.23  1.3  
## hemobch4   3 -0.06  0.12  0.81  0.72  0.28  1.1  
## hemobsi4   2  0.24 -0.11  0.65  0.56  0.44  1.3  
## hemobst4   6 -0.01  0.23  0.64  0.65  0.35  1.3  
## hemobpi4  10  0.25 -0.11  0.44  0.31  0.69  1.7
```

```
##  
##      PA1  PA3  PA2  
## SS loadings      2.36 2.18 2.14  
## Proportion Var    0.24 0.22 0.21  
## Cumulative Var    0.24 0.45 0.67  
## Proportion Explained 0.35 0.33 0.32  
## Cumulative Proportion 0.35 0.68 1.00
```

```
##  
## With factor correlations of
```

```
##      PA1  PA3  PA2  
## PA1 1.00 0.77 0.68  
## PA3 0.77 1.00 0.69  
## PA2 0.68 0.69 1.00
```

```
##  
## Mean item complexity = 1.2  
## Test of the hypothesis that 3 factors are sufficient.  
##
```

```
## The degrees of freedom for the null model are 45 and the objective function was 7.21 with Chi Squ
```

```
## The degrees of freedom for the model are 18 and the objective function was 0.19
```

```
##
```

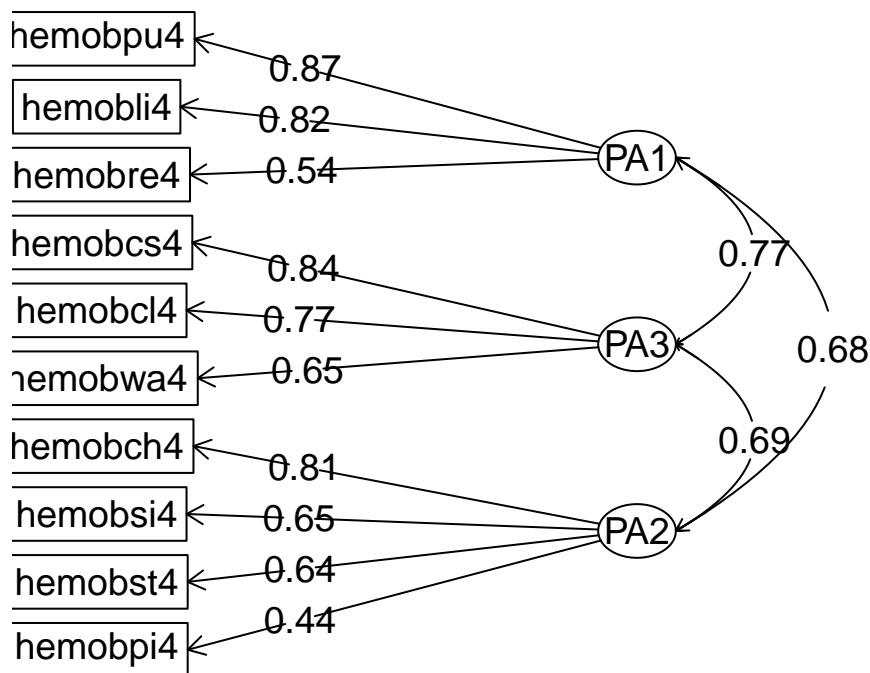
```
## The root mean square of the residuals (RMSR) is 0.02
```

```
## The df corrected root mean square of the residuals is  0.04
##
## The harmonic number of observations is  5520 with the empirical chi square  245.75  with prob <  5.9
## The total number of observations was  5520  with Likelihood Chi Square =  1068.74  with prob <  1.4e
##
## Tucker Lewis Index of factoring reliability =  0.934
## RMSEA index =  0.103  and the 90 % confidence intervals are  0.098 0.108
## BIC =  913.65
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##
## Correlation of (regression) scores with factors    PA1  PA3  PA2
## Multiple R square of scores with factors          0.96 0.95 0.93
## Minimum correlation of possible factor scores      0.93 0.90 0.86
## Minimum correlation of possible factor scores      0.86 0.81 0.72
```

```
#draw diagram of the factor structure and correlation between factors
```

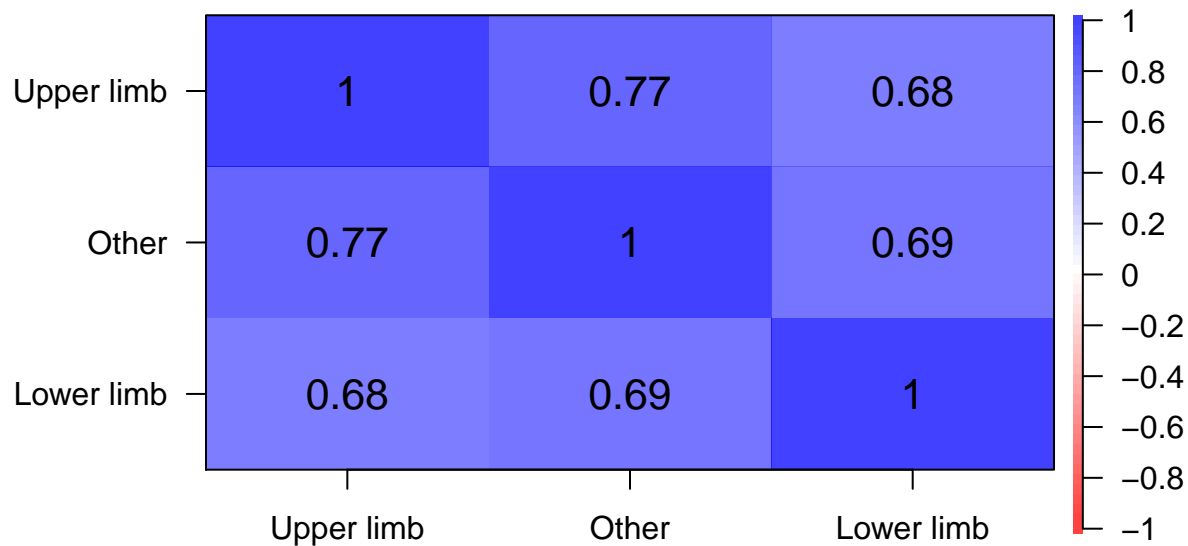
```
fa.diagram(mobility.fa, cut=0.3, labels=names(mobility.fa), digits=2, e.size=0.05, rsize=0.1, cex=1.2, adj=1, l
```

Factor Analysis



```
cor.plot(mobility.fa$Phi, main = "Factor Correlations", labels = c("Upper limb", "Other", "Lower limb"), cex=
```

Factor Correlations



```
# calculate factor score
f1=c(8,9,7)
f2=c(4,5,1)
f3=c(3,2,6,10)
Upper_limb=as.matrix(temp[,f1])%*%mobility.fa$loadings[1:3,1]
Lower_limb=as.matrix(temp[,f2])%*%mobility.fa$loadings[4:6,2]
Other=as.matrix(temp[,f3])%*%mobility.fa$loadings[7:10,3]
score1=factor.scores(temp,mobility.fa)
mobility_factor=score1$scores
```

2. ADLs+IADLs

```
var_list=unlist(classified_list[3:4])
var_list
```

```
## [1] "headldr4" "headlwa4" "headlba4" "headlea4" "headlbe4" "headlwc4"
## [7] "headlma4" "headlpr4" "headlsh4" "headlph4" "headlme4" "headlmo4"
## [13] "headlhg4"
```

```
temp=data.frame(mydata[,var_list])
Disability=matrix(nrow=length(var_list),ncol=6,dimnames = list(var_list,c("Mean","SD","Min","Max","percentile10","percentile90")))

for (j in 1:4){
  for (i in 1: length(var_list)){
    tab=table(temp[,i])
    Disability[i,5]=tab[[1]]/sum(tab)
    Disability[i,6]=tab[[2]]/sum(tab)
    Disability[i,1]=mean(temp[,i])
    Disability[i,2]=sd(temp[,i])
    Disability[i,3]=min(temp[,i])
    Disability[i,4]=max(temp[,i])
  }
}
```

```

}
Disability=data.frame(Disability)
Disability

```

	Mean	SD	Min	Max	percent.of.0.	percent.of.1.
## headldr4	0.118819055	0.32360478	0	1	0.8811809	0.118819055
## headlwa4	0.010324217	0.10109143	0	1	0.9896758	0.010324217
## headlba4	0.075348669	0.26397702	0	1	0.9246513	0.075348669
## headlea4	0.012135483	0.10950061	0	1	0.9878645	0.012135483
## headlbe4	0.034595182	0.18276872	0	1	0.9654048	0.034595182
## headlwc4	0.022459699	0.14818650	0	1	0.9775403	0.022459699
## headlma4	0.033870676	0.18091263	0	1	0.9661293	0.033870676
## headlpr4	0.018112661	0.13337096	0	1	0.9818873	0.018112661
## headlsh4	0.054700235	0.22741479	0	1	0.9452998	0.054700235
## headlph4	0.013584496	0.11576867	0	1	0.9864155	0.013584496
## headlme4	0.007607318	0.08689542	0	1	0.9923927	0.007607318
## headlmo4	0.010686470	0.10283085	0	1	0.9893135	0.010686470
## headlhg4	0.125158486	0.33092850	0	1	0.8748415	0.125158486

```

for (i in 1:ncol(temp)){
  print(attr(temp[,i],"label"))
}

```

```

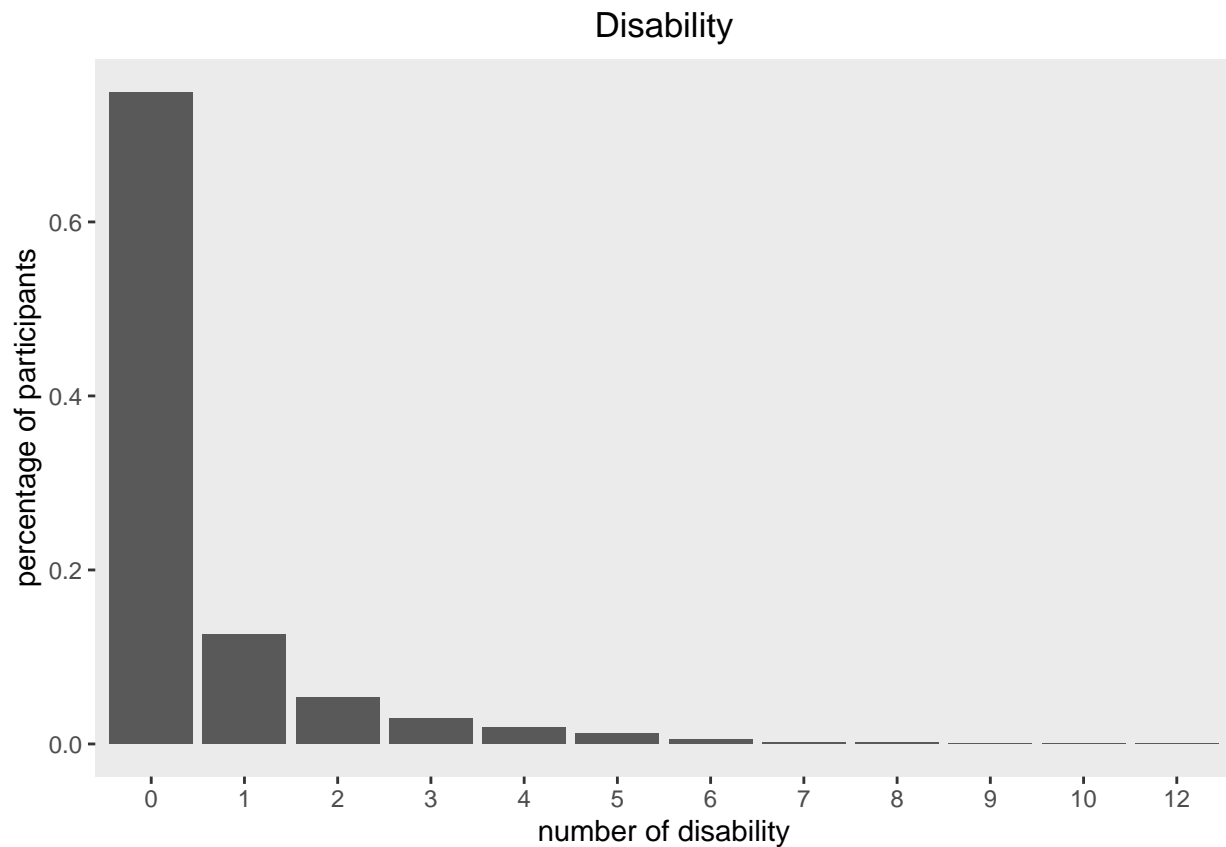
## [1] "headldr ADL: difficulty dressing, including putting on shoes and socks "
## [1] "headlwa ADL: difficulty walking across a room "
## [1] "headlba ADL: difficulty bathing or showering "
## [1] "headlea ADL: difficulty eating, such as cutting up food "
## [1] "headlbe ADL: difficulty getting in and out of bed "
## [1] "headlwc ADL: difficulty using the toilet, including getting up or down "
## [1] "headlma IADL: difficulty using map to figure out how to get around strange place "
## [1] "headlpr IADL: preparing a hot meal "
## [1] "headlsh IADL: shopping for groceries "
## [1] "headlte IADL: making telephone calls "
## [1] "headlme IADL: taking medications "
## [1] "headlmo IADL: managing money, such as bills and expenses "
## [1] "headlho IADL: doing work around the house or garden "

```

```

disability=apply(mydata[,var_list],1,sum)
ggplot(NULL,aes(x=as.factor(disability)))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="number of disability",y="percentage of participants",title = "Disability")+
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())

```

```
#coefficient
#drop=c("headlpr4","headlsh4")
drop=c("headlpr4") #drop prepare hot meal since its cross loading on both factors
temp=temp[,!(names(temp) %in% drop)]
corre_disability=tetrachoric(temp,smooth=T)
```

```
corre_disability=pmax(corre_disability$rho,t(corre_disability$rho))
#corre_disability=cor(temp)
```

```
#factor adequacy
KMO(corre_disability)
```

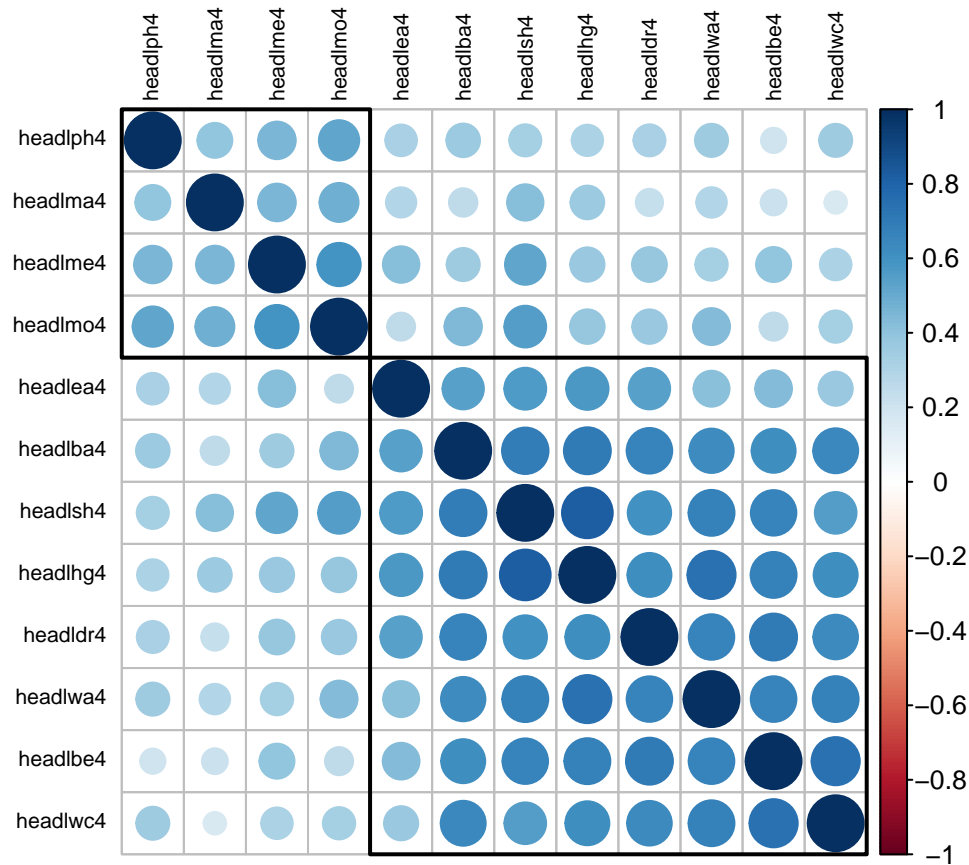
```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = corre_disability)
## Overall MSA = 0.88
## MSA for each item =
## headldr4 headlwa4 headlba4 headlea4 headlbe4 headlwc4 headlma4 headlsh4
## 0.92 0.93 0.94 0.87 0.85 0.89 0.90 0.88
## headlph4 headlme4 headlmo4 headlhg4
## 0.87 0.84 0.77 0.88
```

```
bart_spher(corre_mobility)
```

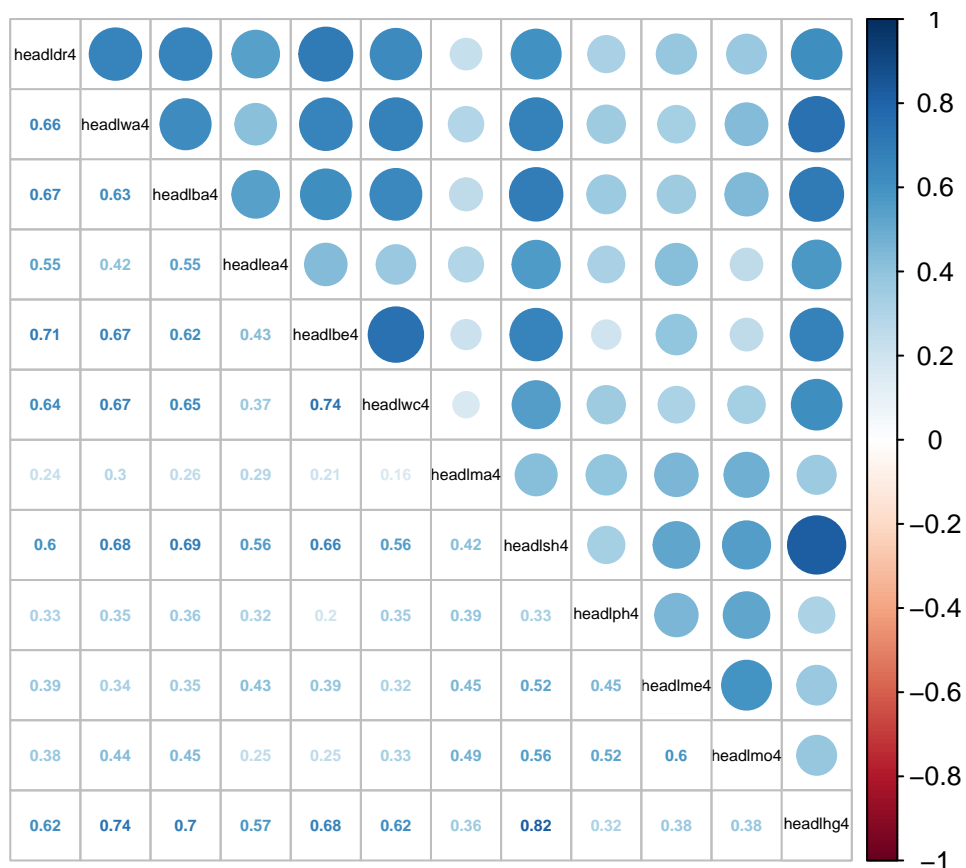
```
## Bartlett's Test of Sphericity
##
```

```
## Call: bart_spher(x = corre_mobility)
##
##      X2 = 210.772
##      df = 45
## p-value < 2.22e-16
```

```
library("corrplot")
corrplot(corre_disability, is.corr = TRUE, number.cex = .5, tl.col = 'black', order = 'hclust', address = )
```



```
corrplot.mixed(corre_disability, is.corr = TRUE, number.cex = .5, tl.col = 'black', tl.cex = 0.5)
```



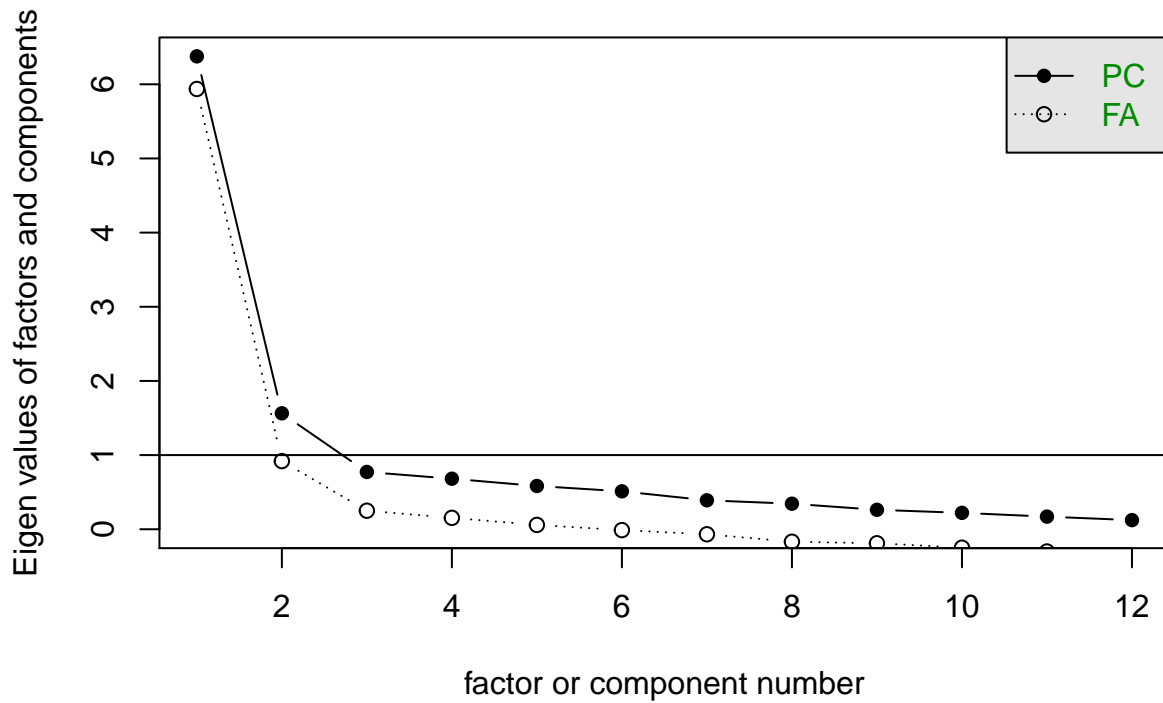
decide number of factors

```
eigen.corre <- eigen(corre_disability)
eigen.corre$values
```

```
## [1] 6.3768552 1.5633073 0.7722669 0.6812745 0.5836075 0.5114630 0.3904702
## [8] 0.3443304 0.2621552 0.2209970 0.1701501 0.1231228
```

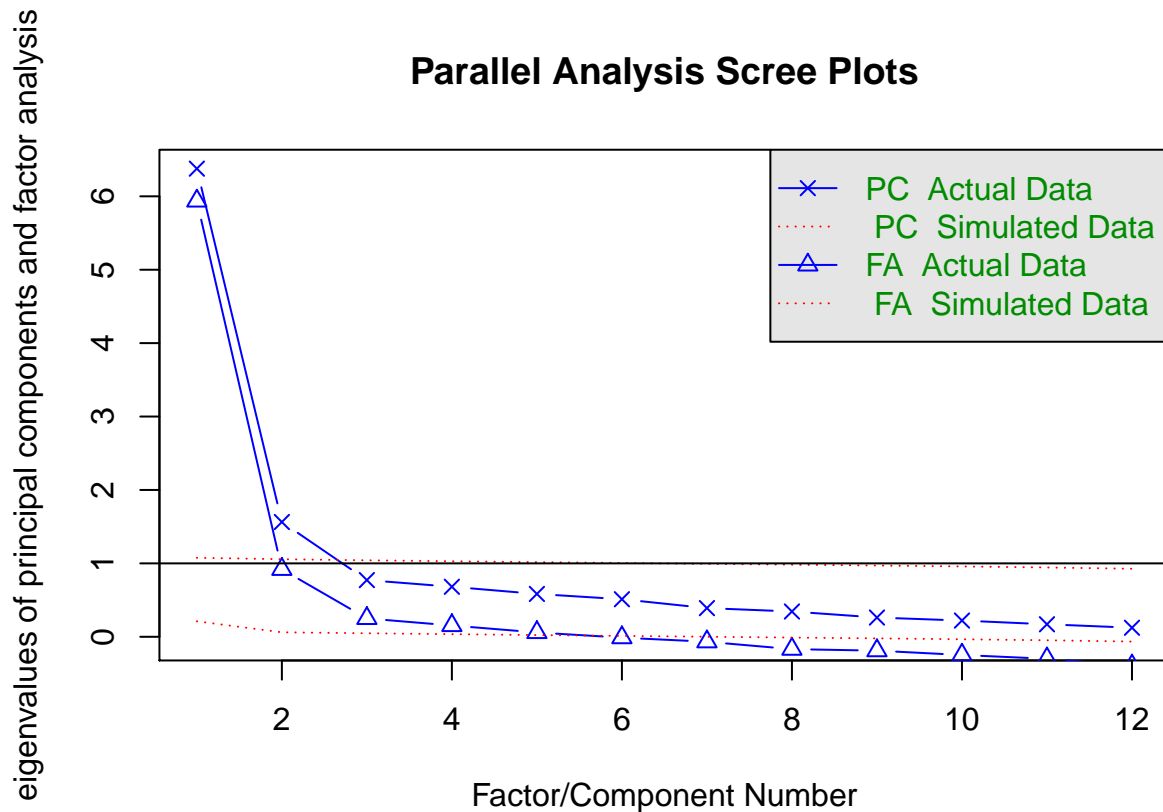
```
scree(corre_disability)
```

Scree plot



```
fa.parallel(corre_disability, n.obs= nrow(mydata),fm="pa",fa="both",n.iter = 1000)
```

Parallel Analysis Scree Plots



```
## Parallel analysis suggests that the number of factors = 5 and the number of components = 2
```

```

disability.fa1<- fa(corre_disability, nfactors = 2, n.obs=5500,rotate = 'none',scores = 'regression', fm = "pa",
#disability.fa1
print(disability.fa1,sort = T)

```

```

## Factor Analysis using method = pa
## Call: fa(r = corre_disability, nfactors = 2, n.obs = 5500, rotate = "none",
##       scores = "regression", fm = "pa", oblique.scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  PA1  PA2  h2  u2 com
## headlsh4   8 0.86 0.03 0.74 0.26 1.0
## headlhg4  12 0.84 -0.16 0.74 0.26 1.1
## headlba4   3 0.80 -0.14 0.66 0.34 1.1
## headlwa4   2 0.80 -0.16 0.66 0.34 1.1
## headldr4   1 0.78 -0.19 0.64 0.36 1.1
## headlbe4   5 0.78 -0.35 0.72 0.28 1.4
## headlwc4   6 0.74 -0.27 0.62 0.38 1.3
## headlea4   4 0.62 0.00 0.38 0.62 1.0
## headlmo4  11 0.60 0.53 0.65 0.35 2.0
## headlme4  10 0.59 0.44 0.53 0.47 1.8
## headlph4   9 0.49 0.38 0.38 0.62 1.9
## headlma4   7 0.45 0.44 0.39 0.61 2.0
##
##
##      PA1  PA2
## SS loadings      6.01 1.11
## Proportion Var      0.50 0.09
## Cumulative Var      0.50 0.59
## Proportion Explained 0.84 0.16
## Cumulative Proportion 0.84 1.00
##
## Mean item complexity = 1.4
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are 66 and the objective function was 8.27 with Chi Square = 8.27
## The degrees of freedom for the model are 43 and the objective function was 1.1
##
## The root mean square of the residuals (RMSR) is 0.05
## The df corrected root mean square of the residuals is 0.06
##
## The harmonic number of observations is 5500 with the empirical chi square 1614.46 with prob < 9.9e-16
## The total number of observations was 5500 with Likelihood Chi Square = 6046.11 with prob < 0
##
## Tucker Lewis Index of factoring reliability = 0.797
## RMSEA index = 0.159 and the 90 % confidence intervals are 0.156 0.163
## BIC = 5675.77
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##
##      PA1  PA2
## Correlation of (regression) scores with factors 0.97 0.85
## Multiple R square of scores with factors 0.95 0.73
## Minimum correlation of possible factor scores 0.89 0.46

```

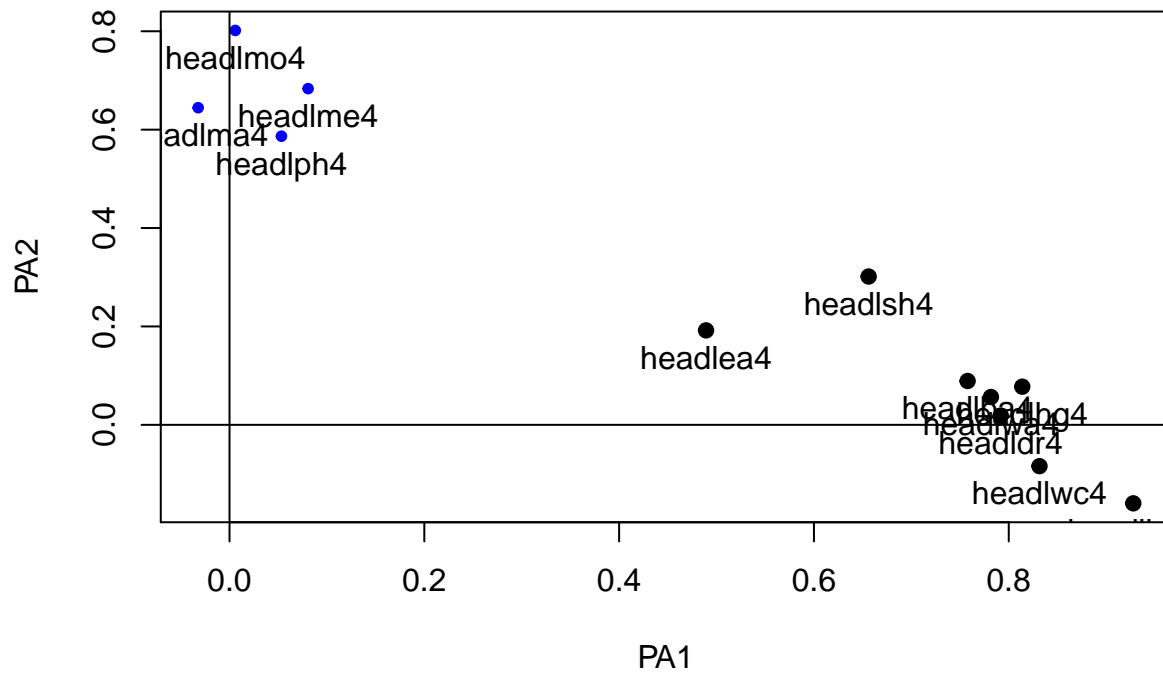
```
disability.fa3<- fa(corre_disability, nfactors = 2,n.obs=5500,rotate = 'oblimin',scores = 'regression',
#disability.fa3
print(disability.fa3,sort = T)
```

```
## Factor Analysis using method = pa
## Call: fa(r = corre_disability, nfactors = 2, n.obs = 5500, rotate = "oblimin",
##       scores = "regression", fm = "pa", oblique.scores = TRUE)
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item    PA1    PA2    h2    u2 com
## headlbe4    5  0.93 -0.16 0.72 0.28 1.1
## headlwc4    6  0.83 -0.08 0.62 0.38 1.0
## headlhg4   12  0.81  0.08 0.74 0.26 1.0
## headldr4    1  0.79  0.02 0.64 0.36 1.0
## headlwa4    2  0.78  0.06 0.66 0.34 1.0
## headlba4    3  0.76  0.09 0.66 0.34 1.0
## headlsh4    8  0.66  0.30 0.74 0.26 1.4
## headlea4    4  0.49  0.19 0.38 0.62 1.3
## headlmo4   11  0.01  0.80 0.65 0.35 1.0
## headlme4   10  0.08  0.68 0.53 0.47 1.0
## headlma4    7 -0.03  0.64 0.39 0.61 1.0
## headlph4    9  0.05  0.59 0.38 0.62 1.0
##
##
##      PA1    PA2
## SS loadings      4.89 2.23
## Proportion Var      0.41 0.19
## Cumulative Var      0.41 0.59
## Proportion Explained 0.69 0.31
## Cumulative Proportion 0.69 1.00
##
## With factor correlations of
##      PA1    PA2
## PA1 1.00 0.55
## PA2 0.55 1.00
##
## Mean item complexity = 1.1
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are 66 and the objective function was 8.27 with Chi Squ
## The degrees of freedom for the model are 43 and the objective function was 1.1
##
## The root mean square of the residuals (RMSR) is 0.05
## The df corrected root mean square of the residuals is 0.06
##
## The harmonic number of observations is 5500 with the empirical chi square 1614.46 with prob < 9.
## The total number of observations was 5500 with Likelihood Chi Square = 6046.11 with prob < 0
##
## Tucker Lewis Index of factoring reliability = 0.797
## RMSEA index = 0.159 and the 90 % confidence intervals are 0.156 0.163
## BIC = 5675.77
## Fit based upon off diagonal values = 0.99
## Measures of factor score adequacy
##
##      PA1    PA2
## Correlation of (regression) scores with factors 0.97 0.91
```

```
## Multiple R square of scores with factors      0.94 0.83
## Minimum correlation of possible factor scores 0.88 0.67
```

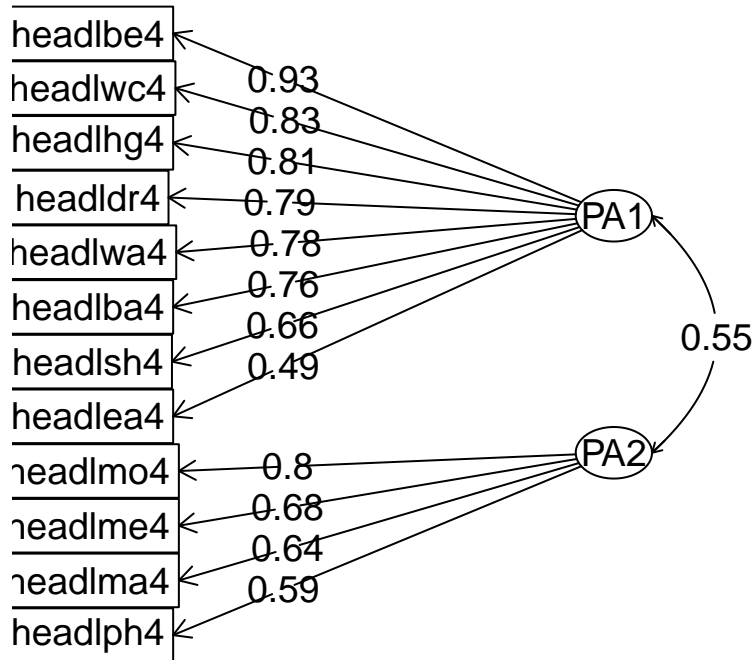
```
factor.plot(disability.fa3, labels = rownames(disability.fa3$loadings))
```

Factor Analysis



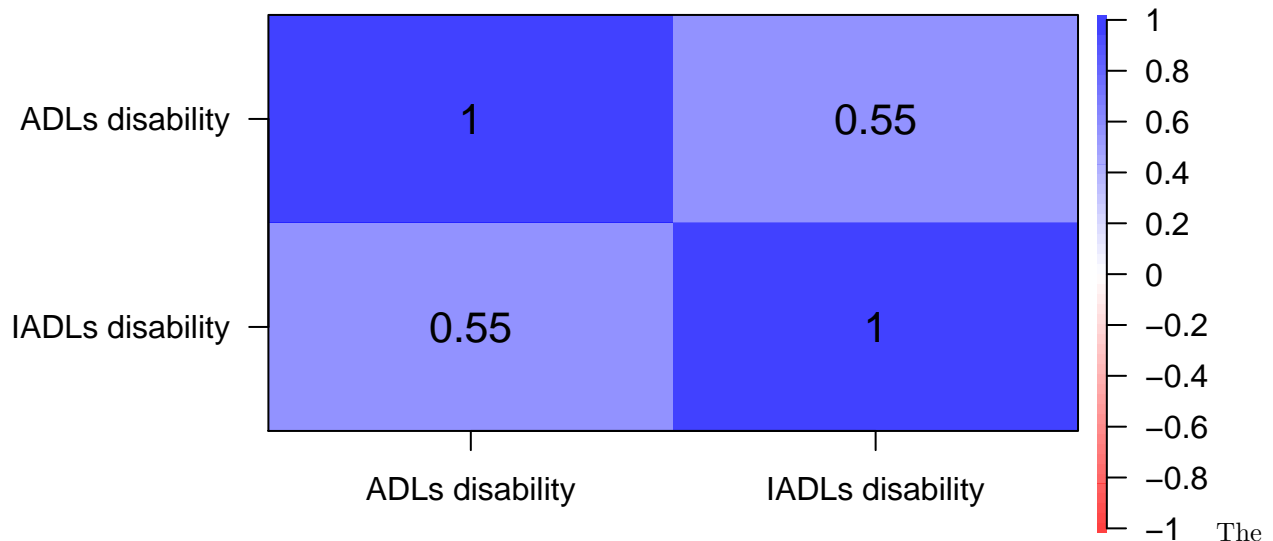
```
fa.diagram(disability.fa3, cut=0.299, labels=names(disability.fa3), digits=2, e.size=0.05, rsize=0.1, cex=1.1)
```

Factor Analysis



```
cor.plot(disability.fa3$Phi, main = "Factor Correlations", labels = c("ADLs disability", "IADLs disability"))
```

Factor Correlations



The factor structure of disability items is clear because it uses well defined ADLs and IADLs to measure disability and this structure also been proved by the statistical approach, methods for determining number of factors and model fitting. ADLs measures more difficulty with physical activities, such as getting in and out of bed, using the toilet, dressing, bathing or showering, eating. IADLs measures more complex activities and are more related to cognitive function than physical function, which is distinct from ADLs.


```
f1=c(5,6,12,1,2,3,8,4)
Physical_disability=as.matrix(temp[,f1])%*%disability.fa3$loadings[1:8,1]
Cognitive_disability=as.matrix(temp[,f1])%*%disability.fa3$loadings[9:12,2]
score2=factor.scores(temp,disability.fa3)
disability_factor=score2$scores
ADLs_disability=Physical_disability
IADLs_disability=Cognitive_disability
```

3. Diagnosed condition: CVD+chronic

Diagnosed conditions include 7 cardiovascular disease (CVD), 9 chronic diseases.

```
var_list=unlist(classified_list[5:6])
temp=data.frame(mydata[,var_list])
diagnosed=matrix(nrow=length(var_list),ncol=6,dimnames = list(var_list,c("Mean","SD","Min","Max","percen

for (i in 1: length(var_list)){
  tab=table(temp[,i])
  diagnosed[i,5]=tab[[1]]/sum(tab)
  diagnosed[i,6]=tab[[2]]/sum(tab)
  diagnosed[i,1]=mean(temp[,i])
  diagnosed[i,2]=sd(temp[,i])
  diagnosed[i,3]=min(temp[,i])
  diagnosed[i,4]=max(temp[,i])
}

diagnosed=data.frame(diagnosed)
diagnosed
```

##		Mean	SD	Min	Max	percen.of.0.	percen.of.1.
##	hedimbp4	0.4309001992	0.49524705	0	1	0.5690998	0.4309001992
##	hediman4	0.0961782286	0.29486221	0	1	0.9038218	0.0961782286
##	hedimmi4	0.0820503532	0.27446628	0	1	0.9179496	0.0820503532
##	hedimhf4	0.0041659120	0.06441513	0	1	0.9958341	0.0041659120
##	hedimar4	0.0851295055	0.27909959	0	1	0.9148705	0.0851295055
##	hedimdi4	0.1045100525	0.30594878	0	1	0.8954899	0.1045100525
##	hedimst4	0.0429270060	0.20271093	0	1	0.9570730	0.0429270060
##	hediblu4	0.0532512226	0.22455437	0	1	0.9467488	0.0532512226
##	hedibas4	0.1113928636	0.31464651	0	1	0.8886071	0.1113928636
##	hedibar4	0.3946748777	0.48882502	0	1	0.6053251	0.3946748777
##	hedibos4	0.0778844412	0.26801393	0	1	0.9221156	0.0778844412
##	hedibca4	0.0585038942	0.23471508	0	1	0.9414961	0.0585038942
##	hedibpd4	0.0070639377	0.08375744	0	1	0.9929361	0.0070639377
##	hedibps4	0.0697337439	0.25472083	0	1	0.9302663	0.0697337439
##	hedibad4	0.0003622532	0.01903123	0	1	0.9996377	0.0003622532
##	hedibde4	0.0012678863	0.03558803	0	1	0.9987321	0.0012678863

```
for (i in 1:ncol(temp)){
  print(attr(temp[,i],"label"))
}
```

```
## [1] "hedimbp CVD: high blood pressure diagnosis newly reported (merged) "
```

```
## [1] "hediman CVD: angina diagnosis newly reported (merged) "
```

```
## [1] "hedimmi CVD: heart attack diagnosis newly reported (merged) "
```

```
## [1] "hedimhf CVD: congestive heart failure diagnosis newly reported (merged) "
```

```
## [1] "hedimar CVD: abnormal heart rhythm diagnosis newly reported (merged) "
```

```
## [1] "hedimdi CVD: diabetes or high blood sugar diagnosis newly reported (merged) "
```

```
## [1] "hedimst CVD: stroke diagnosis newly reported (merged) "
```

```
## [1] "hediblu Chronic: lung disease diagnosis newly reported "
```

```
## [1] "hedibas Chronic: asthma diagnosis newly reported "
```

```
## [1] "hedibar Chronic: arthritis diagnosis newly reported "
```

```
## [1] "hedibos Chronic: osteoporosis diagnosis newly reported "
```

```
## [1] "hedibca Chronic: cancer diagnosis newly reported "
```

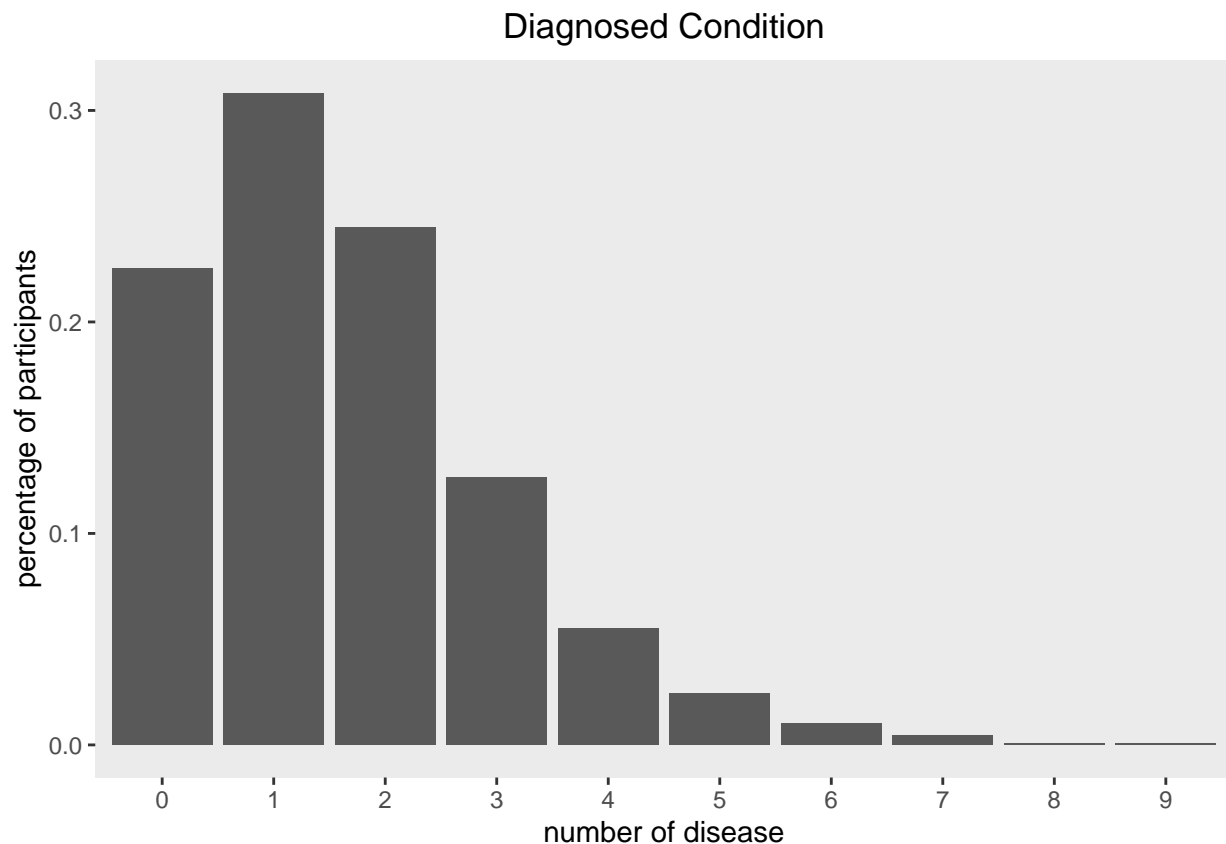
```
## [1] "hedibpd Chronic: Parkinsons Disease diagnosis newly reported "
```

```
## [1] "hedibps Chronic: psychiatric condition newly reported "
```

```
## [1] "hedibad Chronic: Alzheimers Disease diagnosis newly reported "
```

```
## [1] "hedibde Chronic: dementia diagnosis newly reported "
```

```
diagnosed_N=apply(mydata[,var_list],1,sum)
ggplot(NULL,aes(x=as.factor(diagnosed_N)))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="number of disease",y="percentage of participants",title = "Diagnosed Condition")+
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```



Since CVD and chronic diseases are different type of diseases, the correlation of the items in the two groups are also low, factor analysis is considered to be conducted within the two groups separately.

The factor adequacy index of chronic diseases is lower than 0.5, so EFA is not plausible.

As for the number of factors for CVD items, Kaiser's rule and and scree plot (2 factors) is contradict with parallel analysis (4 factors). Similar procedure is conducted as previously, performance of models with factor numbers of 2 to 4 are compared, the two factors structure is optimal.

```
#drop=c("hedimar4")
#temp=temp[,!(names(temp) %in% drop)]
#corre_diagnoed=cor(temp)
corre_diagnoed=tetrachoric(temp[,1:7],smooth=T)
```

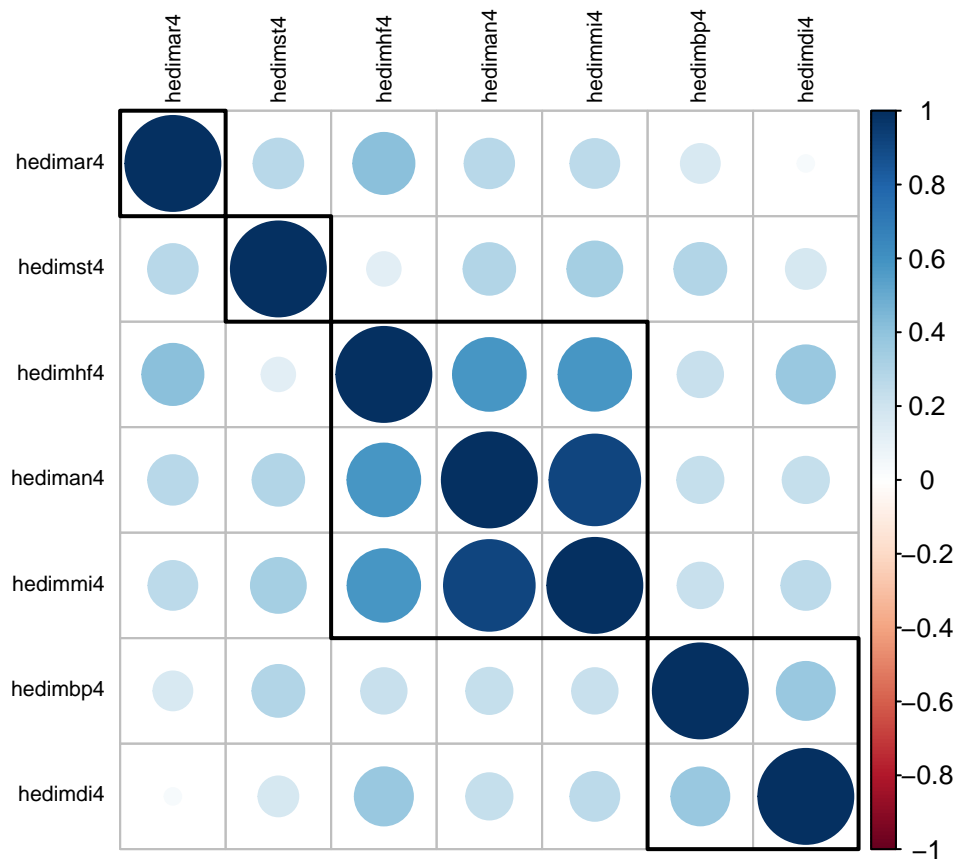
```
corre_diagnoed=pmax(corre_diagnoed$rho,t(corre_diagnoed$rho))
KMO(corre_diagnoed)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = corre_diagnoed)
## Overall MSA = 0.67
## MSA for each item =
## hedimbp4 hediman4 hedimmi4 hedimhf4 hedimar4 hedimdi4 hedimst4
##      0.74      0.65      0.65      0.76      0.62      0.63      0.68
```

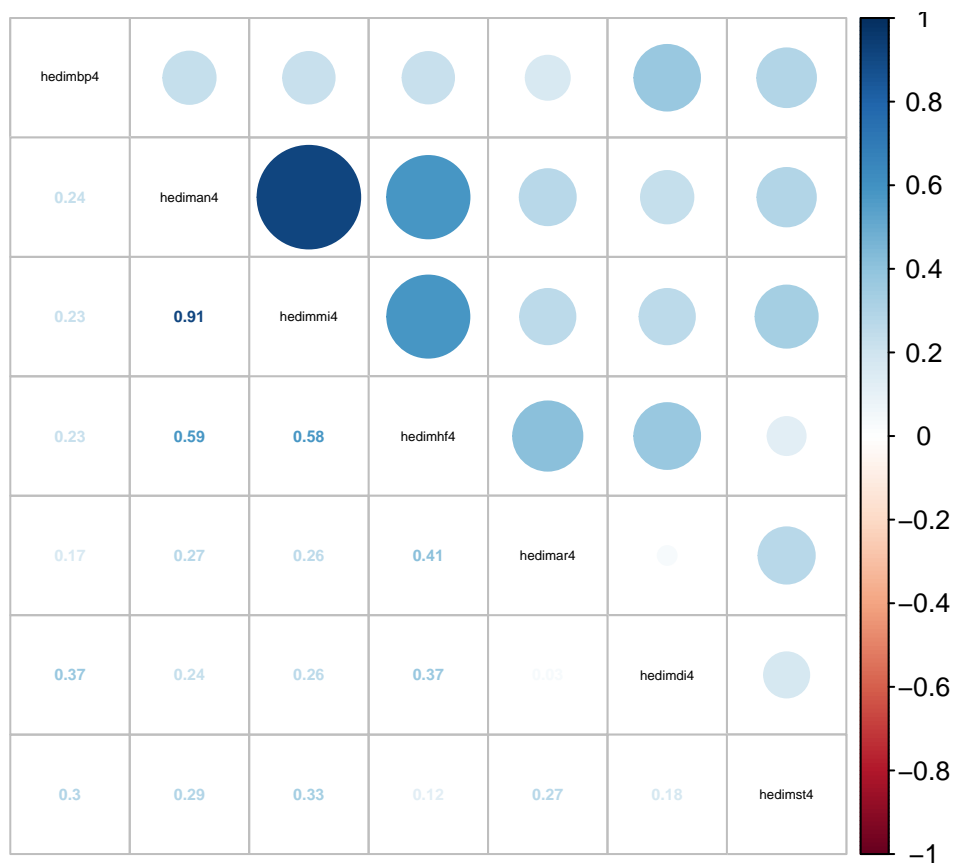
```
bart_spher(corre_diagnoed)
```

```
## Bartlett's Test of Sphericity
##
## Call: bart_spher(x = corre_diagnoed)
##
##      X2 = 122.688
##      df = 21
## p-value < 2.22e-16
```

```
library("corrplot")
corrplot(corre_diagnoed, is.corr = TRUE, number.cex = .5, tl.col = 'black',order = 'hclust', addrect = 4)
```



```
corrplot.mixed(corre_diagnoed, is.corr = TRUE, number.cex = .5, tl.col = 'black',tl.cex = 0.4)
```

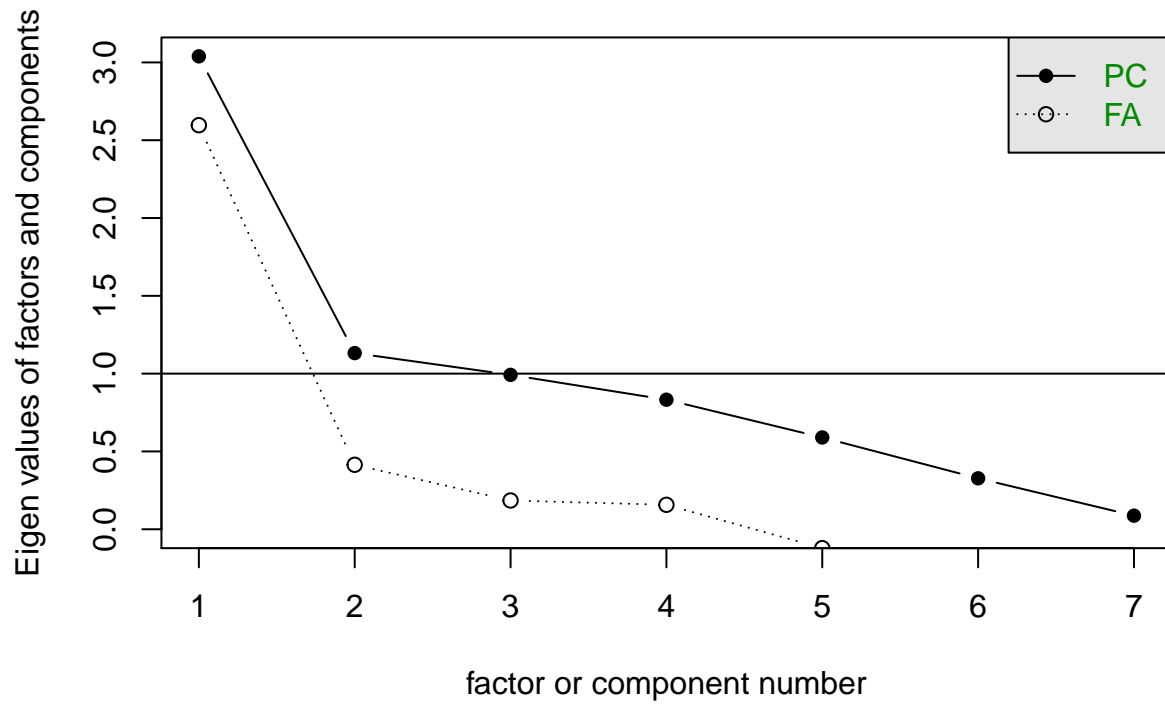


```
eigen.corre <- eigen(corre_diagnoed)
eigen.corre$values
```

```
## [1] 3.03895654 1.13200823 0.99218458 0.83239480 0.58983047 0.32729972 0.08732567
```

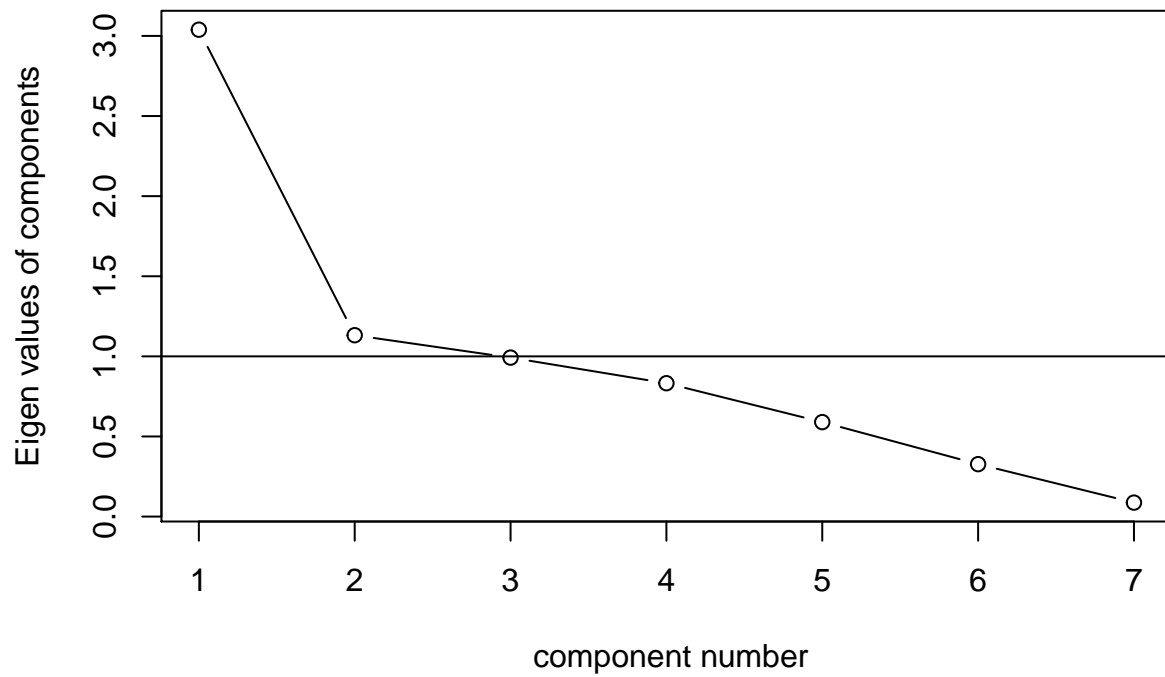
```
scree(corre_diagnoed)
```

Scree plot

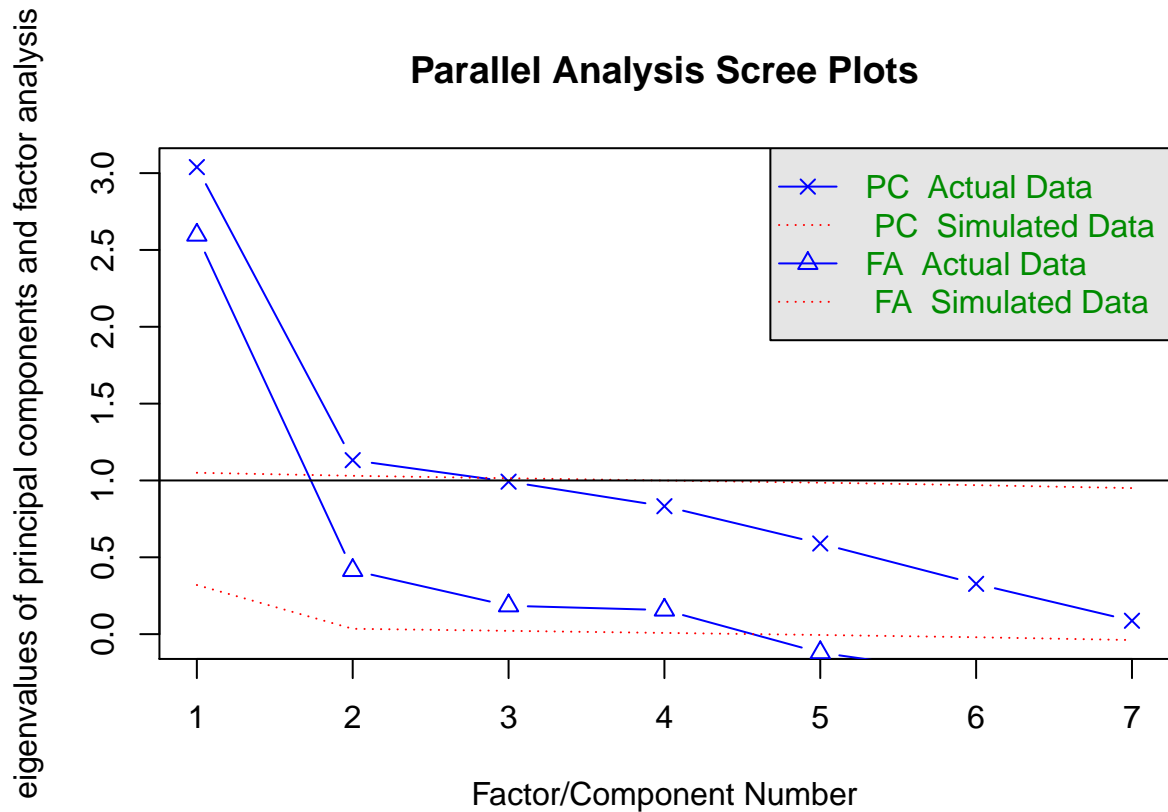


```
VSS.scree(corre_diagnoed)
```

scree plot



```
fa.parallel(corre_diagnoed, n.obs= nrow(mydata),fm = "pa",fa="both",n.iter = 1000)
```



```
## Parallel analysis suggests that the number of factors = 4 and the number of components = 2
```

```
#VSS(corre_diagnoed,n=3,plot = T,n.obs= nrow(mydata))
```

```
diagnoed.fa3<- fa(corre_diagnoed, nfactors = 2,n.obs=4800,rotate = 'oblimin',scores = 'regression', fm = "pa")
print(diagnoed.fa3,sort = T)
```

```
## Factor Analysis using method = pa
## Call: fa(r = corre_diagnoed, nfactors = 2, n.obs = 4800, rotate = "oblimin",
##       scores = "regression", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  PA1  PA2  h2  u2 com
## hediman4   2  0.95 -0.03 0.88 0.12 1.0
## hedimmi4   3  0.95 -0.01 0.89 0.11 1.0
## hedimhf4   4  0.57  0.20 0.45 0.55 1.2
## hedimar4   5  0.28  0.16 0.14 0.86 1.6
## hedimbp4   1 -0.03  0.73 0.51 0.49 1.0
## hedimdi4   6  0.10  0.47 0.27 0.73 1.1
## hedimst4   7  0.20  0.30 0.18 0.82 1.7
##
##
##      SS loadings    PA1  PA2
## Proportion Var    0.33 0.14
```

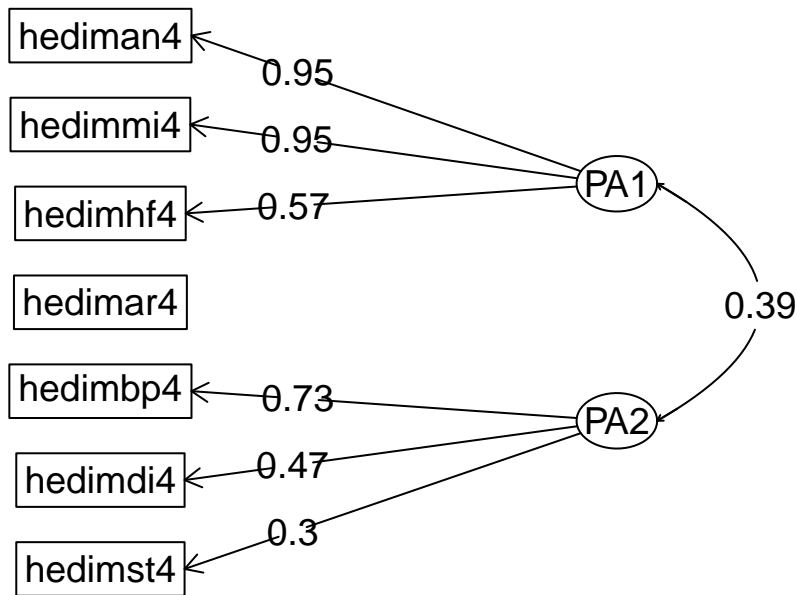
```

## Cumulative Var          0.33 0.47
## Proportion Explained   0.70 0.30
## Cumulative Proportion 0.70 1.00
##
## With factor correlations of
##      PA1  PA2
## PA1 1.00 0.39
## PA2 0.39 1.00
##
## Mean item complexity = 1.2
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are 21 and the objective function was 3.04 with Chi Squ
## The degrees of freedom for the model are 8 and the objective function was 0.32
##
## The root mean square of the residuals (RMSR) is 0.07
## The df corrected root mean square of the residuals is 0.11
##
## The harmonic number of observations is 4800 with the empirical chi square 1000.05 with prob < 1.5
## The total number of observations was 4800 with Likelihood Chi Square = 1526.7 with prob < 0
##
## Tucker Lewis Index of factoring reliability = 0.726
## RMSEA index = 0.199 and the 90 % confidence intervals are 0.191 0.207
## BIC = 1458.89
## Fit based upon off diagonal values = 0.96
## Measures of factor score adequacy
##
## Correlation of (regression) scores with factors    PA1  PA2
## Multiple R square of scores with factors          0.97 0.80
## Minimum correlation of possible factor scores      0.94 0.64
## Minimum correlation of possible factor scores      0.88 0.28

fa.diagram(diagnoed.fa3, cut=0.299, labels=names(diagnoed.fa), digits=2, e.size=0.05, rsize=0.1, cex=1.2, adj

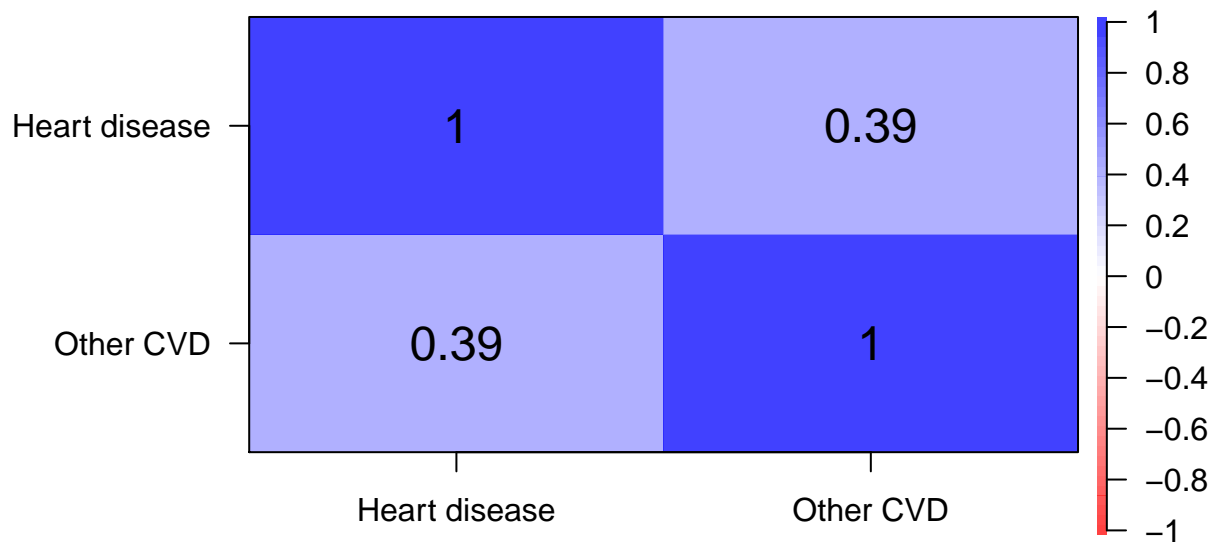
```


Factor Analysis



```
cor.plot(diagnoed.fa3$Phi,main = "Factor Correlations",labels = c("Heart disease","Other CVD"),cex = 1.5)
```

Factor Correlations



```
f1=c(2,3,4,5)
f2=c(1,6,7)
Heart_diseases=as.matrix(temp[,f1])%*%diagnoed.fa3$loadings[1:4,1]
Other_CVD=as.matrix(temp[,f2])%*%diagnoed.fa3$loadings[5:7,2]
score3=factor.scores(temp[,1:7],diagnoed.fa3)
cvd_factor=score3$scores
```

The two factors are heart diseases including items: Angina heart attack, Congestive heart failure and Abnormal heart rhythm. Diabetes or high blood sugar, stroke and high blood pressure are factorised as other CVD.

The total observed variance is 47%, which is relatively low compared with previous models for other items. Consistently, TLI is lower than the cutoff value 0.8 and RMSEA is also high, p-value of likelihood chi square suggests that the model is sufficient, though it dose not fit the data very well. Sole reliance on one or two fit index seems imprudent, the chi squared goodness of fit and the two factors is meaningful, so the two factors solution is acceptable.

4. self-reported items (only EDA)

```
var_list=c(unlist(classified_list[1]),unlist(classified_list[7]))
temp=data.frame(mydata[,var_list])
self_reported_item=matrix(nrow=length(var_list),ncol=7,dimnames = list(var_list,c("Mean","SD","%oflow "

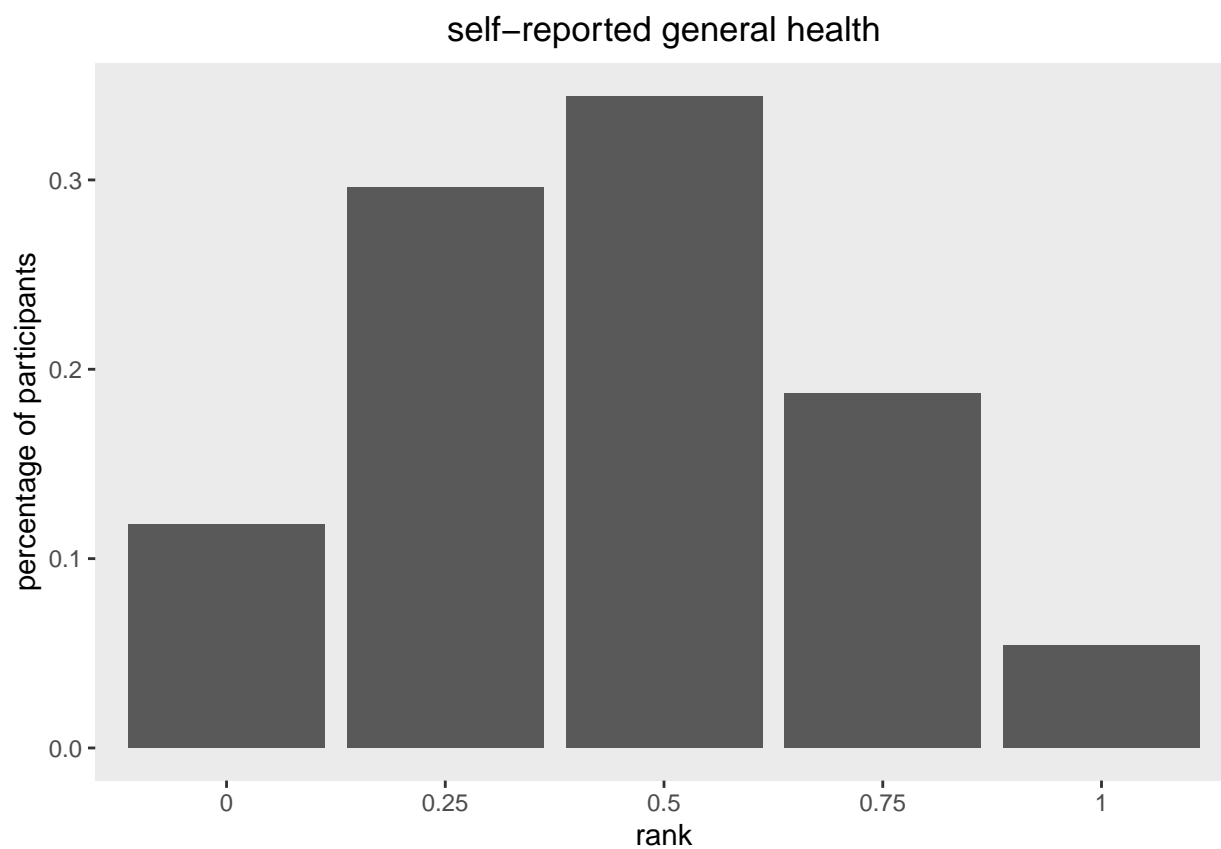
for (i in 1: length(var_list)){
  tab=table(temp[,i])
  self_reported_item[i,3]=tab[[1]]/sum(tab)
  self_reported_item[i,4]=tab[[2]]/sum(tab)
  self_reported_item[i,1]=mean(temp[,i])
  self_reported_item[i,2]=sd(temp[,i])
  self_reported_item[i,5]=tab[[3]]/sum(tab)
  self_reported_item[i,6]=tab[[4]]/sum(tab)
  self_reported_item[i,7]=tab[[5]]/sum(tab)
}
self_reported_item=data.frame(self_reported_item)
self_reported_item
```

```
##           Mean      SD X.oflow.  X.of2nd  X.of3rd  X.of4th
## hehelf4 0.4408622 0.2642468 0.1182757 0.2957798 0.3443217 0.18746604
## heeye4 0.2905633 0.1740389 0.1470748 0.3562760 0.3963050 0.09889513
## hehear4 0.4107951 0.2668053 0.1660931 0.2745879 0.3479442 0.17279478
##           X.ofhigh
## hehelf4 0.054156856
## heeye4 0.001449013
## hehear4 0.038579967
```

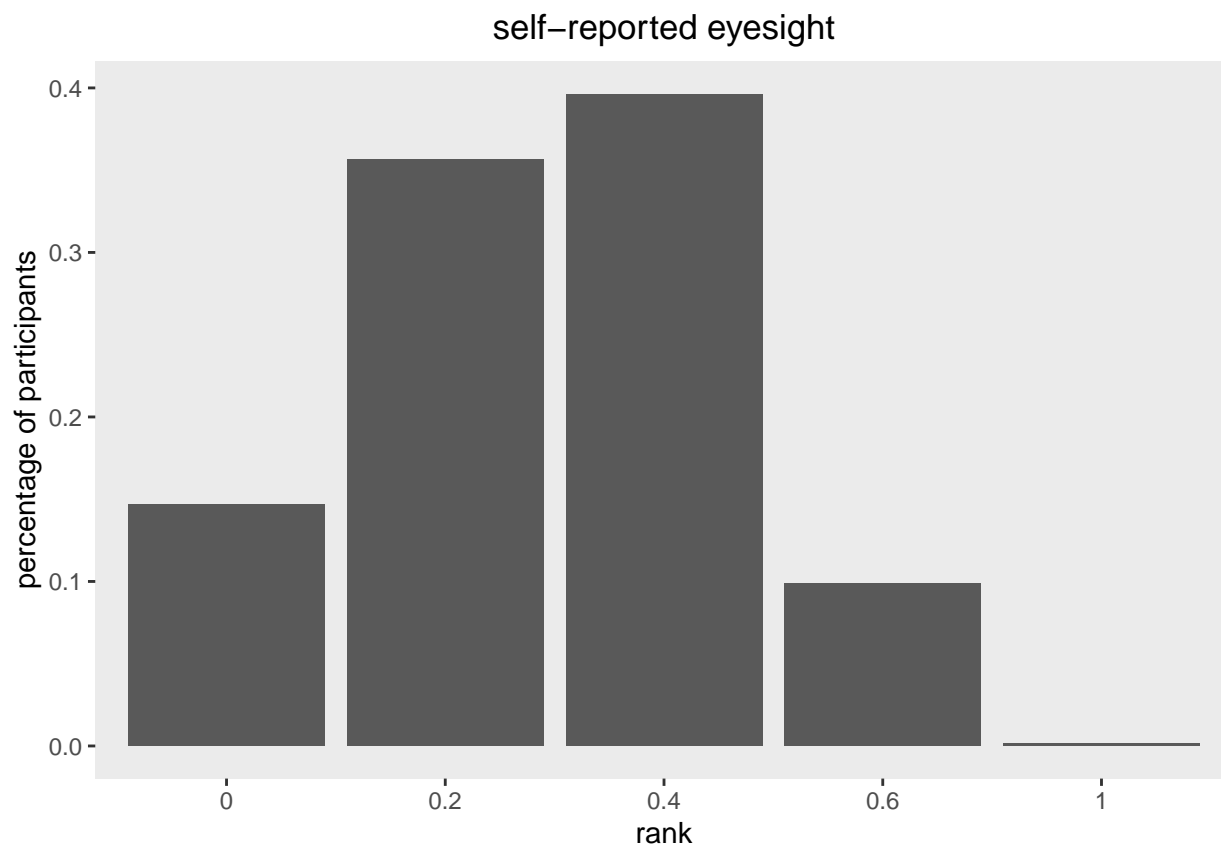
```
for (i in 1:ncol(temp)){
  print(attr(temp[,i],"label"))
}
```

```
## [1] "hehelf Self-reported general health "
## [1] "heeye Self-reported eyesight (while using lenses, if appropriate) "
## [1] "hehear Self-reported hearing (while using hearing aid if appropriate) "
```

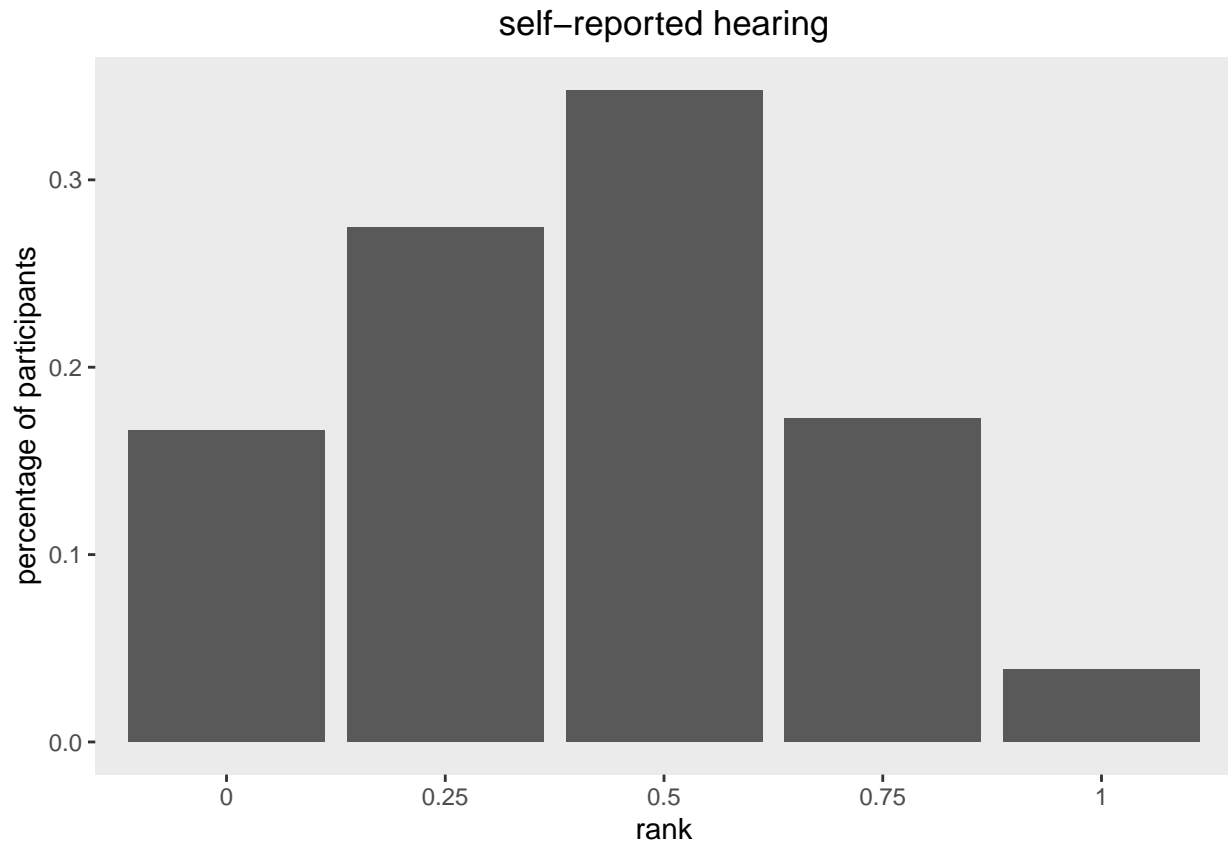
```
ggplot(NULL,aes(x=as.factor(temp[,1]))) +
  geom_bar(aes(y=(..count..)/sum(..count..))) +
  labs(x="rank",y="percentage of participants",title = "self-reported general health") +
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```



```
ggplot(NULL,aes(x=as.factor(temp[,2]))) +  
  geom_bar(aes(y=(..count..)/sum(..count..))) +  
  labs(x="rank",y="percentage of participants",title = "self-reported eyesight") +  
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```



```
ggplot(NULL,aes(x=as.factor(temp[,3]))) +  
  geom_bar(aes(y=(..count..)/sum(..count..))) +  
  labs(x="rank",y="percentage of participants",title = "self-reported hearing") +  
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```



5. Movement items

```
var_list=c(unlist(classified_list[8]))
temp=data.frame(mydata[,var_list])
fall_=matrix(nrow=length(var_list),ncol=6,dimnames = list(var_list,c("Mean","SD","Min","Max","percen of

for (i in 1: length(var_list)){
  tab=table(temp[,i])
  fall_[i,5]=tab[[1]]/sum(tab)
  fall_[i,6]=tab[[2]]/sum(tab)
  fall_[i,1]=mean(temp[,i])
  fall_[i,2]=sd(temp[,i])
  fall_[i,3]=min(temp[,i])
  fall_[i,4]=max(temp[,i])
}

fall_=data.frame(fall_)
fall_
```

##		Mean	SD	Min	Max	percen.of.No.	percen.of.Yes.
##	hefla4	0.237094729	0.4253394	0	1	0.7629053	0.237094729
##	hefrac4	0.004528165	0.0671452	0	1	0.9954718	0.004528165
##	heji4	0.044194892	0.2055465	0	1	0.9558051	0.044194892
##	mmpain4	0.103604419	0.3047743	0	1	0.8963956	0.103604419

```

for (i in 1:ncol(temp)){
  print(attr(temp[,i],"label"))
}

## [1] "hefla Whether fallen down in the last year "
## [1] "hefrac Whether fractured hip in last two years "
## [1] "heji Whether had joint replacement "
## [1] "mmpain Whether had pain whilst walking "

corre_diagnoed=tetrachoric(temp)

corre_diagnoed=pmax(corre_diagnoed$rho,t(corre_diagnoed$rho))
KMO(corre_diagnoed)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = corre_diagnoed)
## Overall MSA = 0.58
## MSA for each item =
##   hefla4 hefrac4   heji4 mmpain4
##    0.57    0.55    0.57    0.77

bart_spher(corre_diagnoed)

## Bartlett's Test of Sphericity
##
## Call: bart_spher(x = corre_diagnoed)
##
##      X2 = 30.218
##      df = 6
## p-value = .00004

pc=fa(corre_diagnoed, nfactors = 1,n.obs=5500,rotate = 'none',scores = 'regression', fm = 'pa')
print(pc,sort=T)

## Factor Analysis using method = pa
## Call: fa(r = corre_diagnoed, nfactors = 1, n.obs = 5500, rotate = "none",
##   scores = "regression", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      V PA1    h2    u2 com
## hefrac4 2 0.92 0.848 0.15  1
## heji4   3 0.46 0.216 0.78  1
## hefla4  1 0.41 0.164 0.84  1
## mmpain4 4 0.26 0.067 0.93  1
##
##              PA1
## SS loadings    1.30
## Proportion Var 0.32
##
## Mean item complexity = 1
## Test of the hypothesis that 1 factor is sufficient.

```

```
##
## The degrees of freedom for the null model are 6 and the objective function was 0.44 with Chi Square
## The degrees of freedom for the model are 2 and the objective function was 0.02
##
## The root mean square of the residuals (RMSR) is 0.04
## The df corrected root mean square of the residuals is 0.07
##
## The harmonic number of observations is 5500 with the empirical chi square 114.27 with prob < 1.5e-
## The total number of observations was 5500 with Likelihood Chi Square = 116.79 with prob < 4.4e-
##
## Tucker Lewis Index of factoring reliability = 0.857
## RMSEA index = 0.102 and the 90 % confidence intervals are 0.087 0.118
## BIC = 99.56
## Fit based upon off diagonal values = 0.98
## Measures of factor score adequacy
##
## Correlation of (regression) scores with factors PA1 0.93
## Multiple R square of scores with factors 0.86
## Minimum correlation of possible factor scores 0.72
```

```
f1=c(2,3,1)
Movement=as.matrix(temp[,f1])%*%pc$loadings[1:3,1]
score4=factor.scores(temp,pc)
Movement_factor=score4$scores
```

Both factor adequacy methods suggest that the correlation of the 4 movement items is sufficient to conduct factor analysis, though 0.58 may be a low value of factor adequacy index. There are only 4 items, a 1 factor is supposed to be extracted based on Kaiser's rule, scree plot and interpretability. Item pain whilst walking is cut off in the model because of the low factor loading.

6. mental health: CESD

```
var_list=unlist(classified_list[9])
temp=data.frame(mydata[,var_list])
CESD=matrix(nrow=length(var_list),ncol=6,dimnames = list(var_list,c("Mean","SD","Min","Max","percent of 0","percent of 1")))

for (i in 1:length(var_list)){
  tab=table(temp[,i])
  CESD[i,5]=tab[[1]]/sum(tab)
  CESD[i,6]=tab[[2]]/sum(tab)
  CESD[i,1]=mean(temp[,i])
  CESD[i,2]=sd(temp[,i])
  CESD[i,3]=min(temp[,i])
  CESD[i,4]=max(temp[,i])
}

CESD=data.frame(CESD)
CESD
```

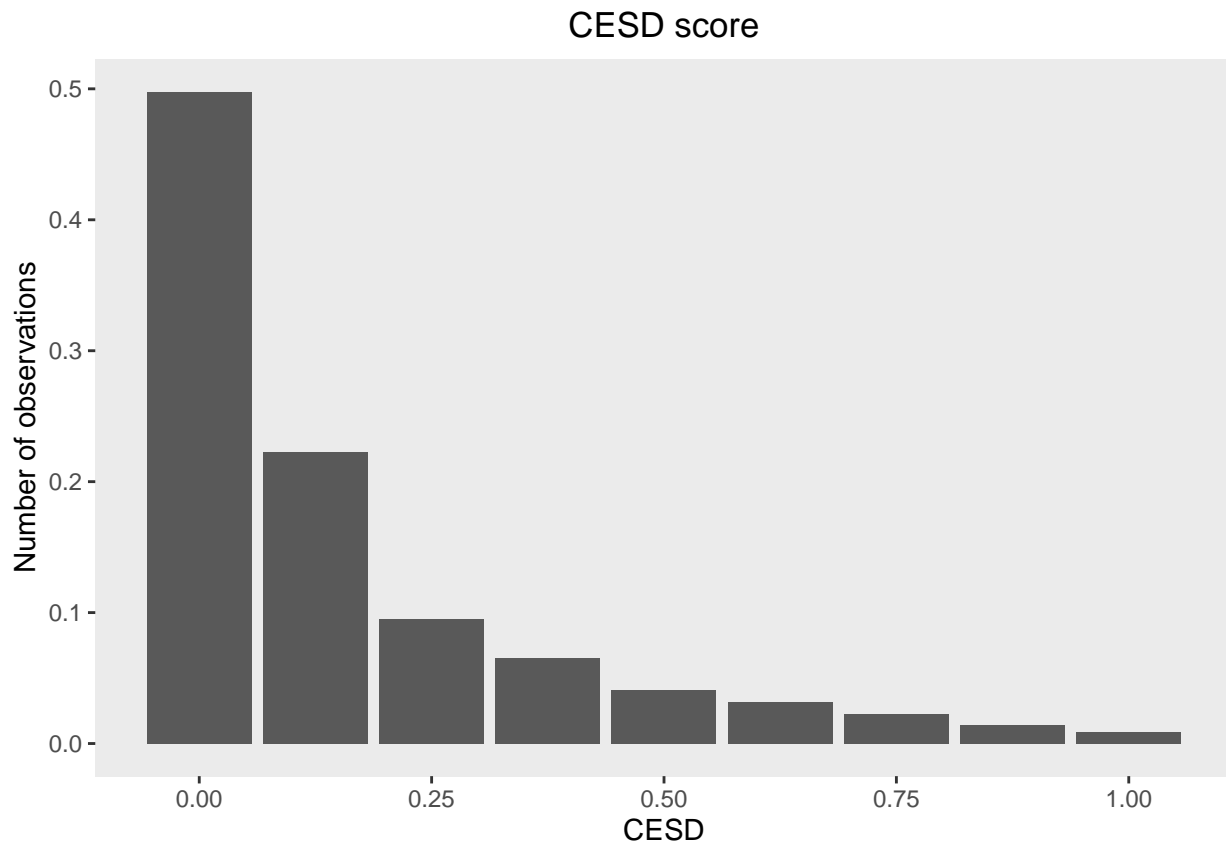
```
##
##          Mean          SD Min Max percent.of.0. percent.of.1.
## psceda4 0.12660750 0.3325629  0  1    0.8733925    0.12660750
## pscedb4 0.16138381 0.3679179  0  1    0.8386162    0.16138381
```

```
## pscedc4 0.32095635 0.4668863 0 1 0.6790437 0.32095635
## pscedd4 0.08476725 0.2785603 0 1 0.9152327 0.08476725
## pscede4 0.11338526 0.3170919 0 1 0.8866147 0.11338526
## pscedf4 0.07299402 0.2601503 0 1 0.9270060 0.07299402
## pscedg4 0.18800942 0.3907551 0 1 0.8119906 0.18800942
## pscedh4 0.17134577 0.3768450 0 1 0.8286542 0.17134577
```

```
for (i in 1:ncol(temp)){
  print(attr(temp[,i],"label"))
}
```

```
## [1] "Psceda Whether felt depressed much of the time during past week "
## [1] "pscedb Whether felt everything they did during past week was an effort "
## [1] "pscedc Whether felt their sleep was restless during past week "
## [1] "pscedd Whether was happy much of the time during past week "
## [1] "pscede Whether felt lonely much of the time during past week "
## [1] "pscedf Whether enjoyed life much of the time during past week "
## [1] "pscedg Whether felt sad much of the time during past week "
## [1] "pscedh Whether could not get going much of the time during past week "
```

```
mean_CESD=apply(mydata[,var_list],1,mean)
ggplot(NULL,aes(x=mean_CESD))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="CESD",y="Number of observations",title = "CESD score")+
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```




```
corre_mental=tetrachoric(mydata[,var_list],smooth=T)
```

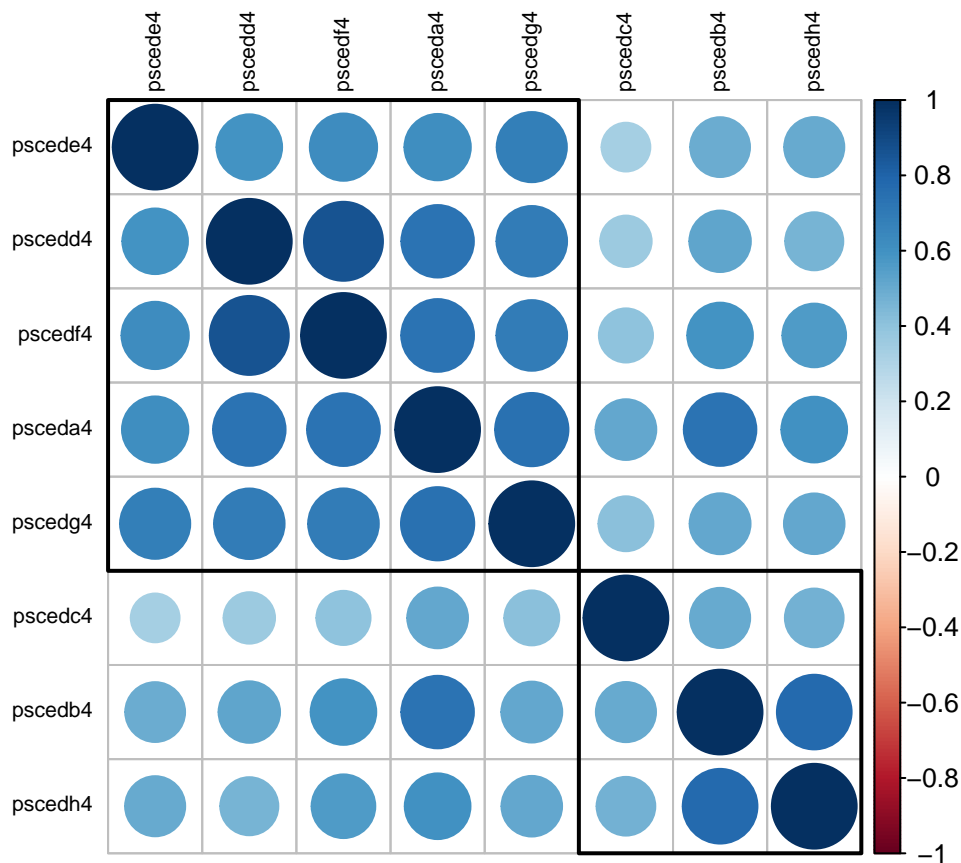
```
corre_mental=pmax(corre_mental$rho,t(corre_mental$rho))
KMO(corre_mental)
```

```
## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = corre_mental)
## Overall MSA = 0.87
## MSA for each item =
## psceda4 pscedb4 pscedc4 pscedd4 pscede4 pscedf4 pscedg4 pscedh4
## 0.88 0.81 0.96 0.84 0.93 0.85 0.89 0.84
```

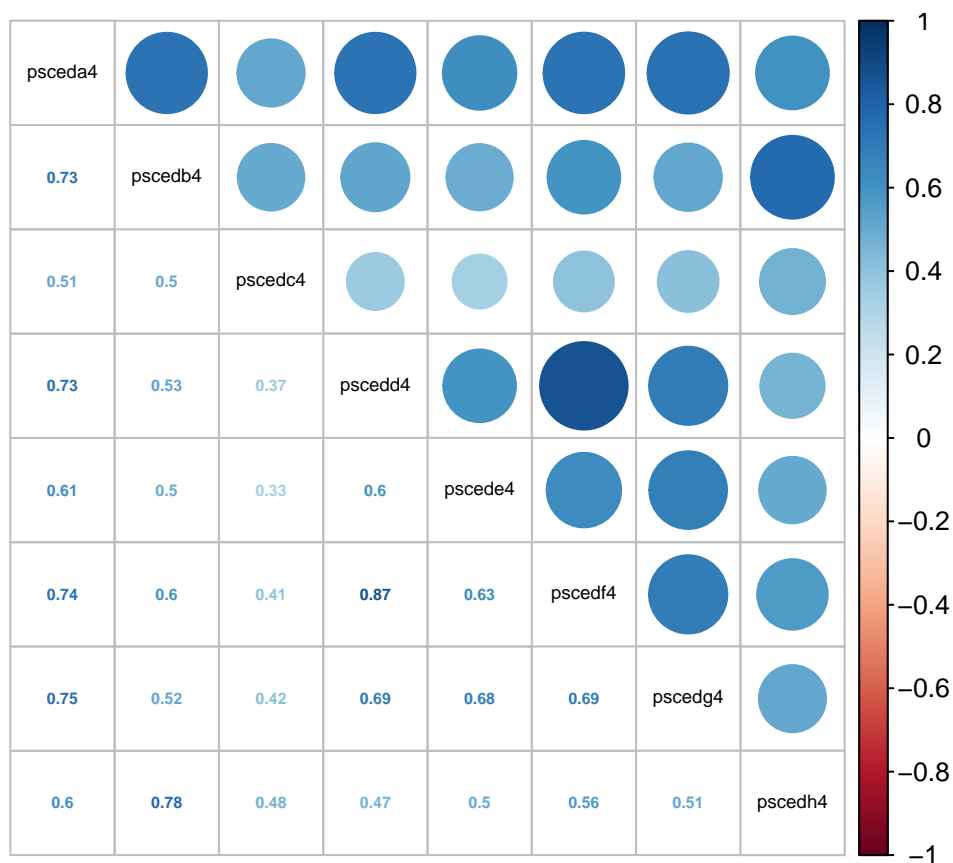
```
bart_spher(corre_mental)
```

```
## Bartlett's Test of Sphericity
##
## Call: bart_spher(x = corre_mental)
##
## X2 = 148.634
## df = 28
## p-value < 2.22e-16
```

```
corrplot(corre_mental, is.corr = TRUE, number.cex = .5, tl.col = 'black',order = 'hclust', addrect = 2,
```



```
corrplot.mixed(corre_mental, is.corr = TRUE, number.cex = .5, tl.col = 'black',tl.cex = 0.6)
```

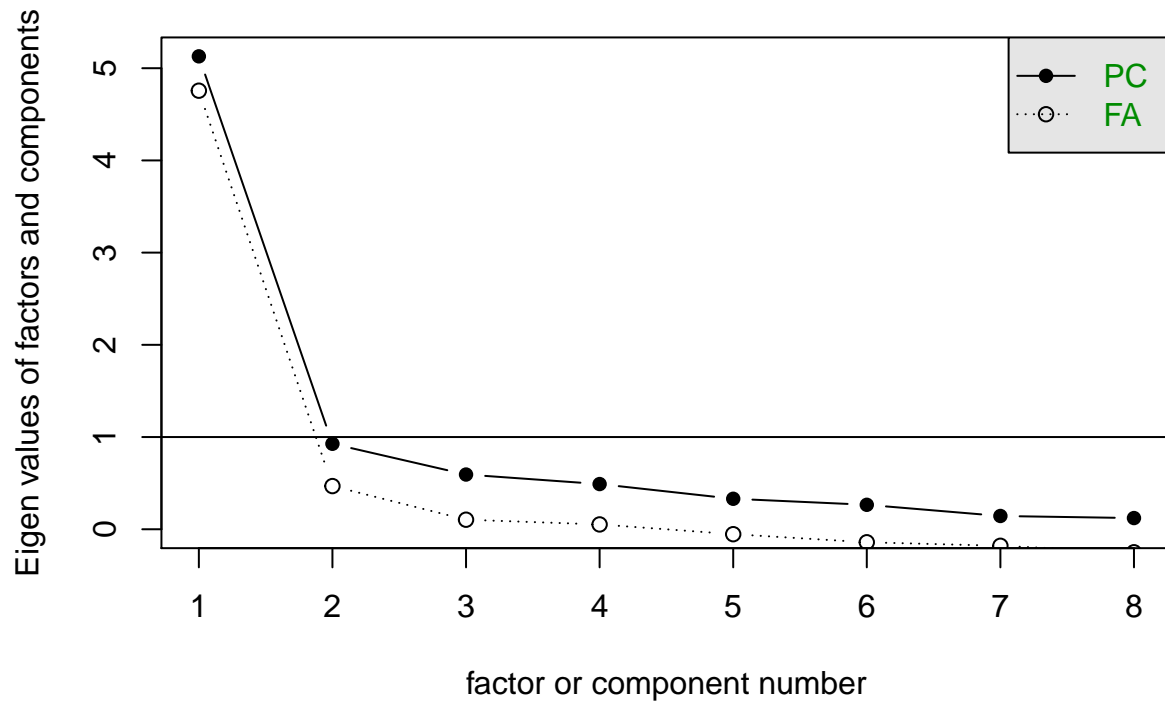


```
eigen.corre <- eigen(corre_mental)
eigen.corre$values
```

```
## [1] 5.1285424 0.9268439 0.5934229 0.4901758 0.3305126 0.2652109 0.1438090
## [8] 0.1214826
```

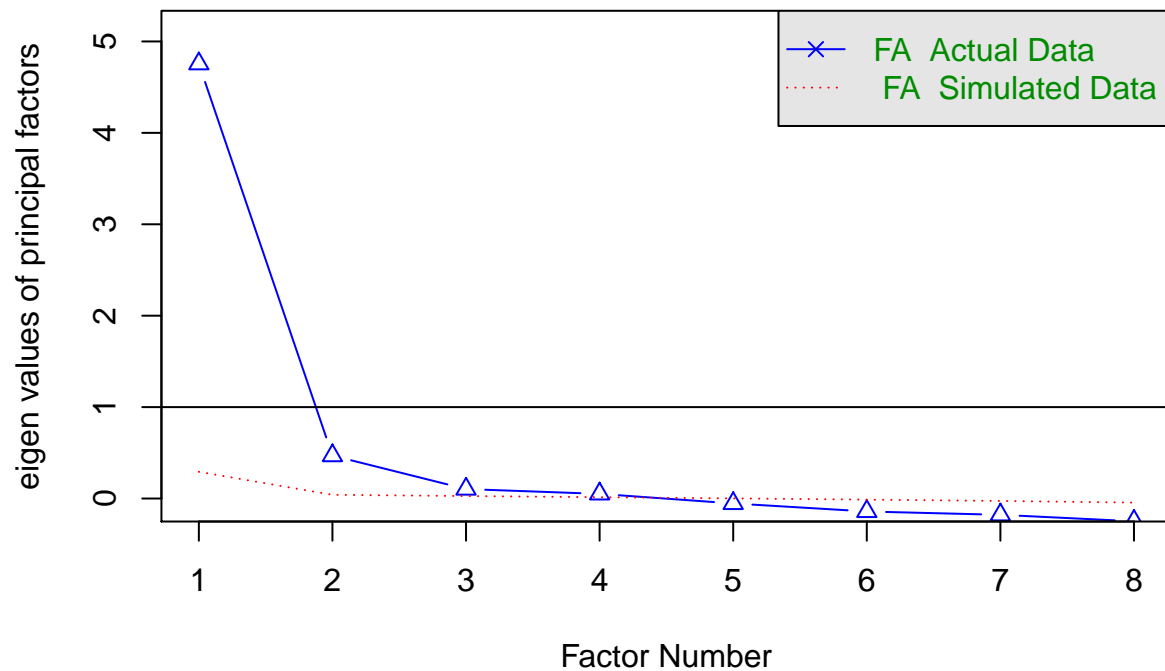
```
scree(corre_mental)
```

Scree plot



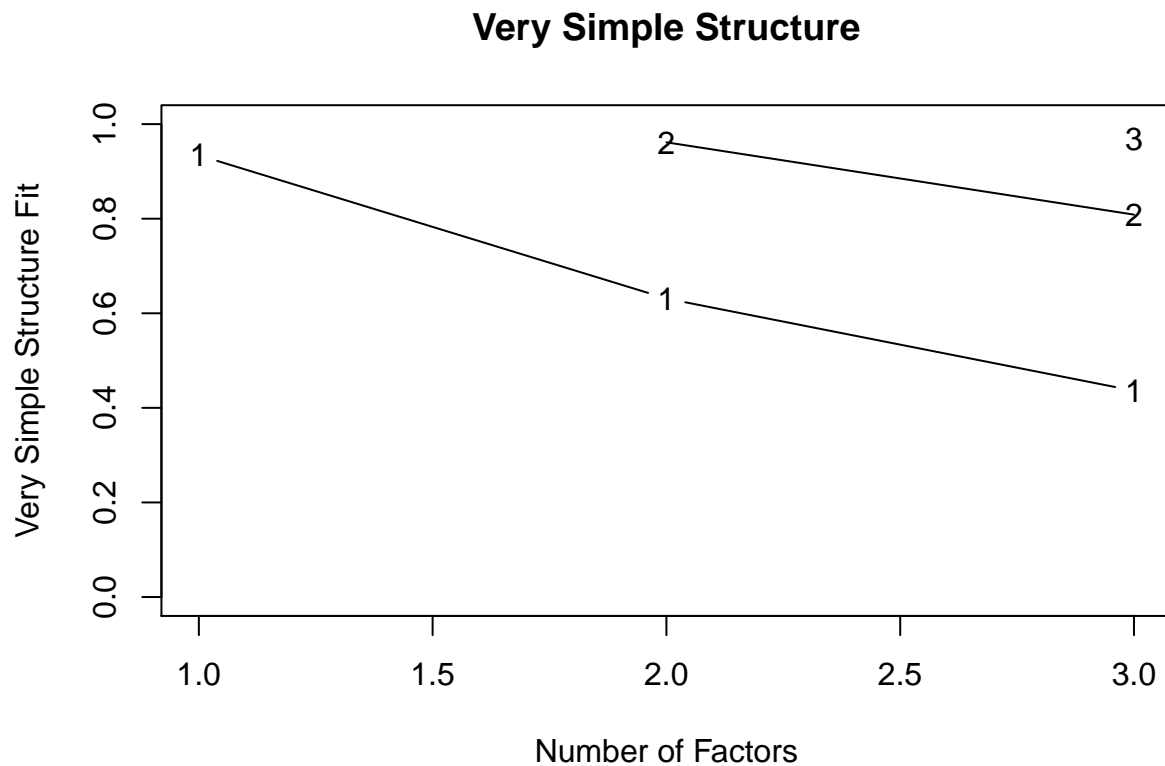
```
fa.parallel(corre_mental, fm="pa", fa="fa", n.obs= nrow(mydata), n.iter = 1000)
```

Parallel Analysis Scree Plots



```
## Parallel analysis suggests that the number of factors = 4 and the number of components = NA
```

```
VSS(corre_mental,n.obs = 5500,n=3,plot = T)
```



```
##
## Very Simple Structure
## Call: vss(x = x, n = n, rotate = rotate, diagonal = diagonal, fm = fm,
##       n.obs = n.obs, plot = plot, title = title, use = use, cor = cor)
## VSS complexity 1 achieves a maximum of 0.93 with 1 factors
## VSS complexity 2 achieves a maximum of 0.96 with 2 factors
##
## The Velicer MAP achieves a minimum of 0.08 with 1 factors
## BIC achieves a minimum of 445.31 with 3 factors
## Sample Size adjusted BIC achieves a minimum of 467.55 with 3 factors
##
## Statistics by number of factors
##   vss1 vss2  map dof chisq  prob sqresid  fit RMSEA  BIC  SABIC complex eChisq
## 1 0.93 0.00 0.075  20  6225  0e+00   1.84 0.93  0.24 6053  6117    1.0  1925
## 2 0.63 0.96 0.083  13  2161  0e+00   1.08 0.96  0.17 2049  2090    1.5   332
## 3 0.44 0.81 0.137   7   506 5e-105   0.89 0.97  0.11  445   468    1.9    82
##   SRMR eCRMS eBIC
## 1 0.079 0.094 1753
## 2 0.033 0.048  220
## 3 0.016 0.033   22
```

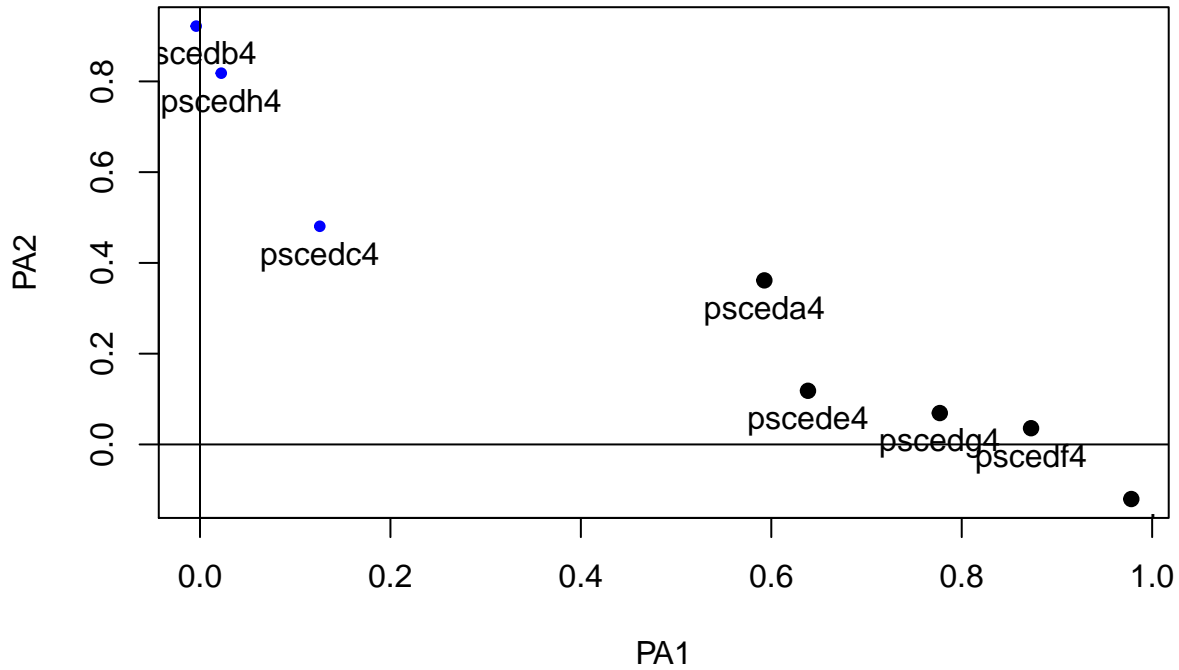
Although there is only one eigenvalues of the correlation coefficient matrix larger than 1, and 1 point above the elbow of the scree plot, parallel analysis have 2 eigenvalues of the actual data above the random data, suggesting a 2 factors model.

```
mental.fa3<- fa(corre_mental, nfactors = 2, rotate = 'oblimin',n.obs=5500,scores = 'regression', fm = 'pa')
print(mental.fa3,sort = T)
```

```
## Factor Analysis using method = pa
## Call: fa(r = corre_mental, nfactors = 2, n.obs = 5500, rotate = "oblimin",
##       scores = "regression", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      item  PA1  PA2  h2  u2 com
## pscedd4    4 0.98 -0.12 0.81 0.19 1.0
## pscedf4    6 0.87  0.04 0.81 0.19 1.0
## pscedg4    7 0.78  0.07 0.68 0.32 1.0
## pscede4    5 0.64  0.12 0.53 0.47 1.1
## psceda4    1 0.59  0.36 0.78 0.22 1.7
## pscedb4    2 0.00  0.92 0.84 0.16 1.0
## pscedh4    8 0.02  0.82 0.70 0.30 1.0
## pscedc4    3 0.13  0.48 0.33 0.67 1.1
##
##
##      PA1  PA2
## SS loadings      3.33 2.14
## Proportion Var    0.42 0.27
## Cumulative Var    0.42 0.68
## Proportion Explained 0.61 0.39
## Cumulative Proportion 0.61 1.00
##
## With factor correlations of
##      PA1  PA2
## PA1 1.00 0.69
## PA2 0.69 1.00
##
## Mean item complexity = 1.1
## Test of the hypothesis that 2 factors are sufficient.
##
## The degrees of freedom for the null model are 28 and the objective function was 6.16 with Chi Square = 4.3
## The degrees of freedom for the model are 13 and the objective function was 0.39
##
## The root mean square of the residuals (RMSR) is 0.03
## The df corrected root mean square of the residuals is 0.05
##
## The harmonic number of observations is 5500 with the empirical chi square 332.18 with prob < 4.3
## The total number of observations was 5500 with Likelihood Chi Square = 2165.99 with prob < 0
##
## Tucker Lewis Index of factoring reliability = 0.863
## RMSEA index = 0.174 and the 90 % confidence intervals are 0.167 0.18
## BIC = 2054.03
## Fit based upon off diagonal values = 1
## Measures of factor score adequacy
##
##      PA1  PA2
## Correlation of (regression) scores with factors 0.97 0.95
## Multiple R square of scores with factors 0.93 0.91
## Minimum correlation of possible factor scores 0.87 0.81
```

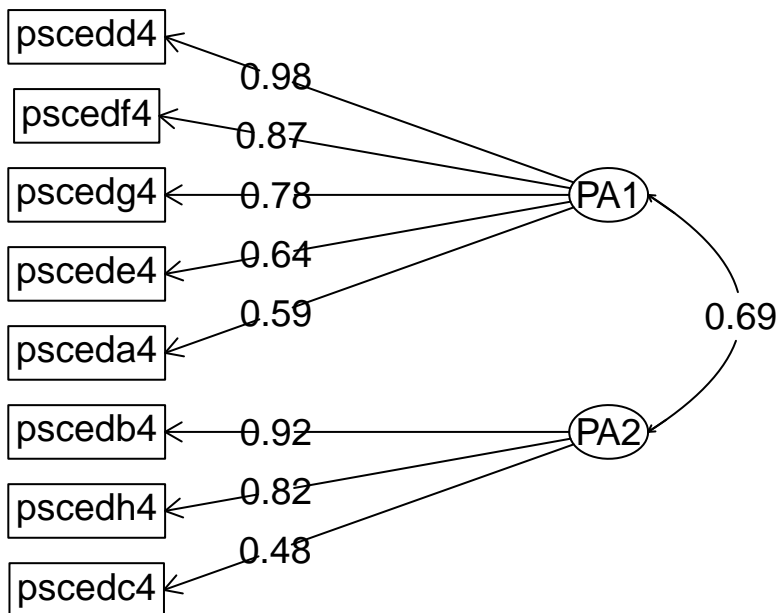
```
factor.plot(mental.fa3, labels = rownames(mental.fa3$loadings))
```

Factor Analysis

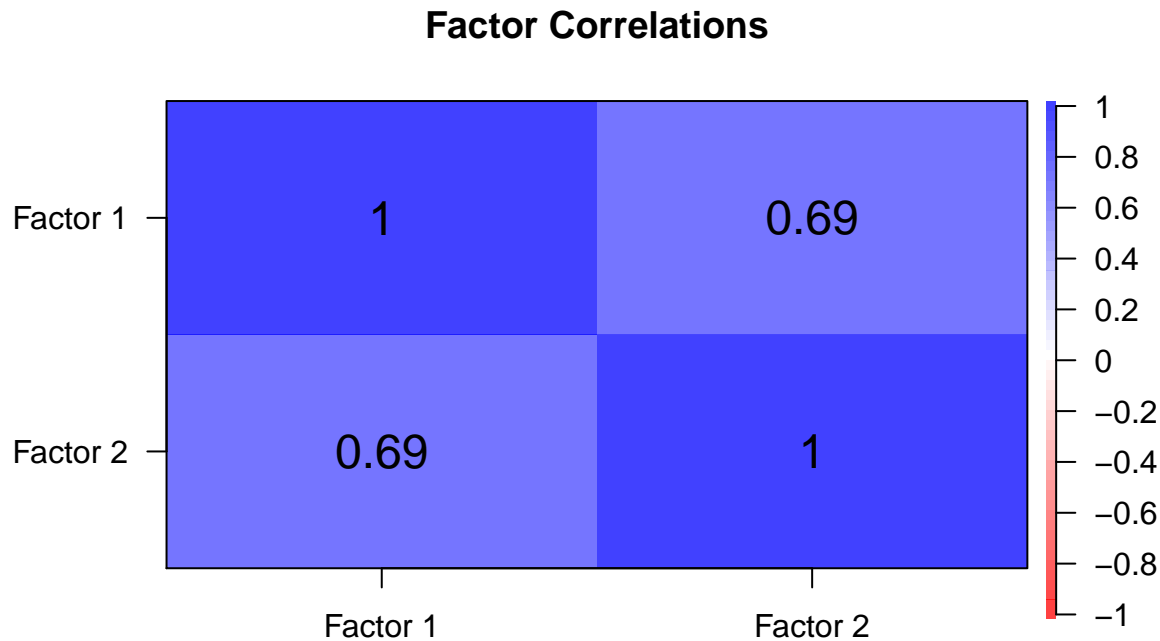


```
fa.diagram(mental.fa3, cut=0.299, labels=names(mental.fa3), digits=2, e.size=0.05, rsize=0.1, cex=1.2, adj=1, ...)
```

Factor Analysis



```
cor.plot(mental.fa3$Phi,main = "Factor Correlations",labels = c("Factor 1","Factor 2"),cex = 1.5)
```

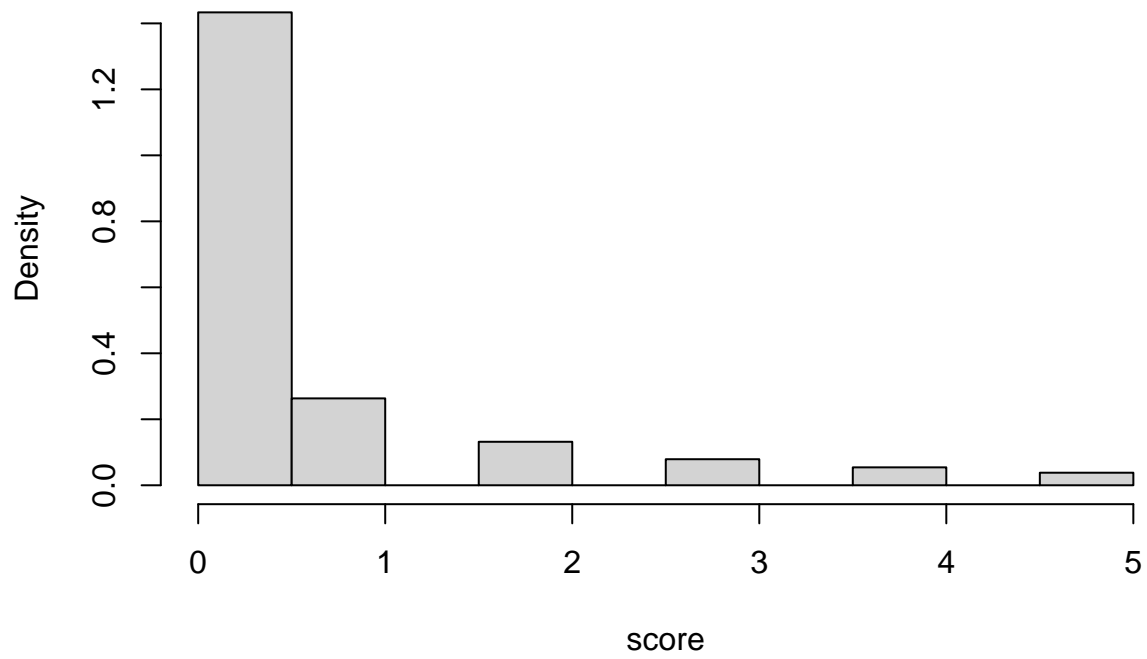


```
f1=c(4,6,7,5,1)
Absence_positive_anhedonia=as.matrix(temp[,f1])%*%mental.fa3$loadings[1:5,1]
Somatic_activity=as.matrix(temp[,~f1])%*%mental.fa3$loadings[6:8,2]
score5=factor.scores(temp,mental.fa3)
mental_factor=score5$scores
```

CES-D depression scales have two constructs: absence of positive affect or anhedonia, somatic activity or inactivity. Absence of positive affect or anhedonia is measured by “happy”, “enjoyed life”, “felt sad”, “felt lonely”, “felt depressed”. These items are distinct from the other items (felt everything was an effort, could not get going, felt restless) measuring the other factor of CES-D. The underlying structure is clear. It is noticeable that the two factors are significantly related though they are conceptually distinctly defined. The interaction between positive or negative affect (PA/NA) and somatic activity (SA) has been studied by researchers.

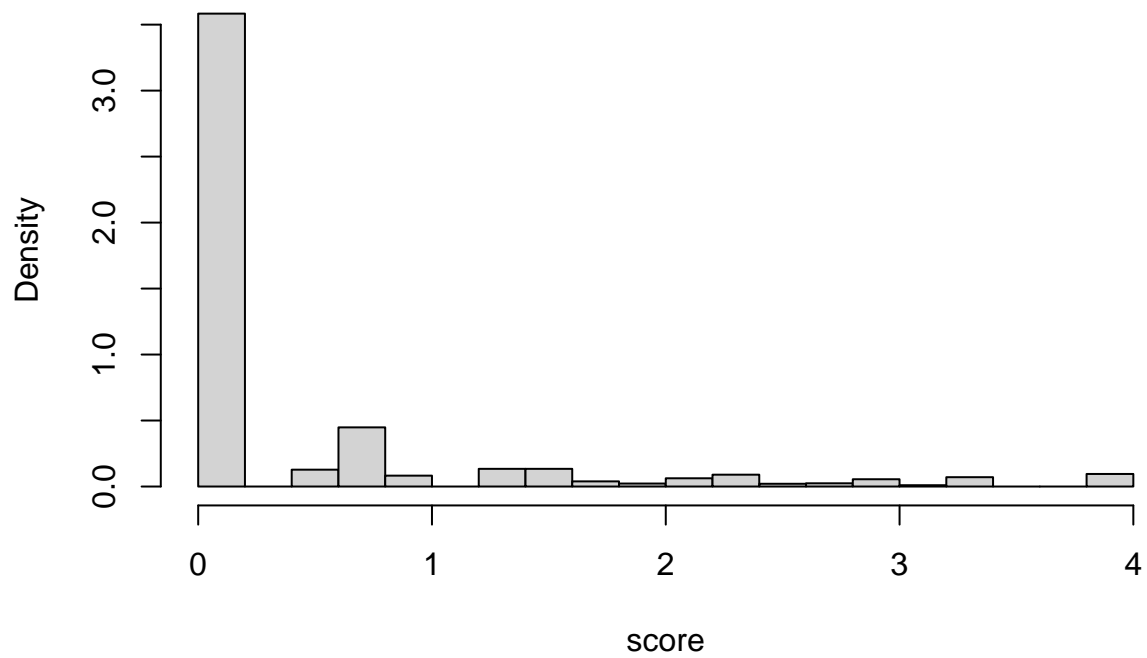
```
f1=c(1,4:7)
score=apply(temp[,f1],1,sum)
hist(score,freq=F)
```

Histogram of score



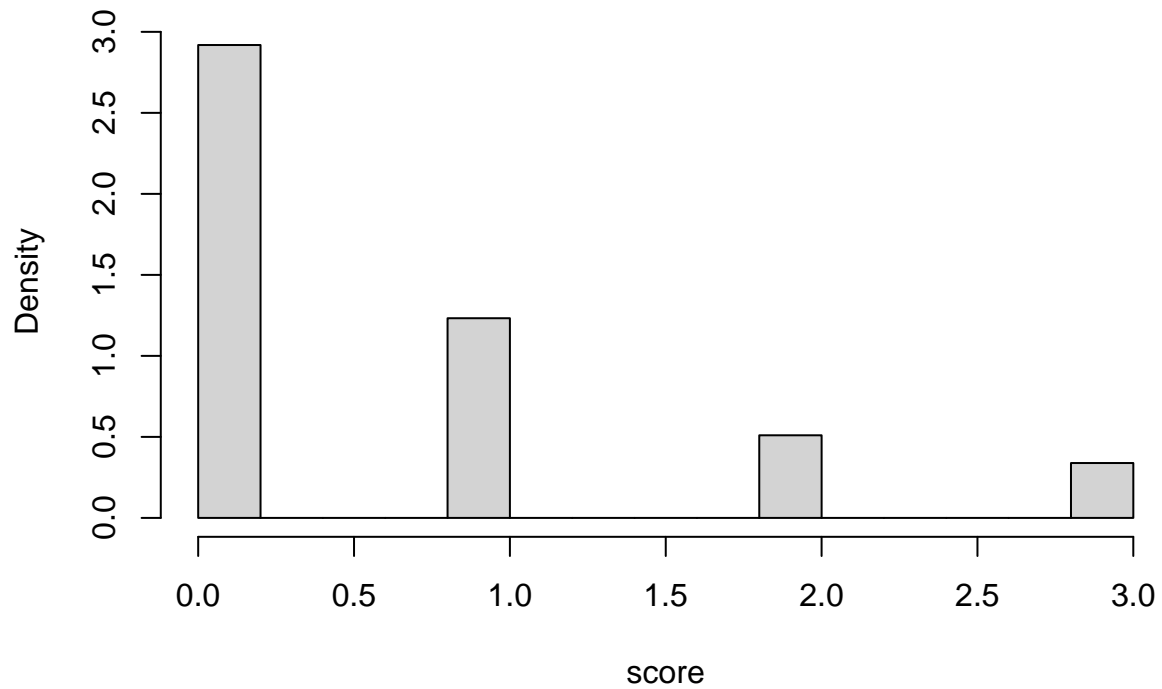
```
score=as.matrix(temp[,f1])%*%mental.fa3$loadings[f1,1]  
hist(score,freq=F)
```

Histogram of score

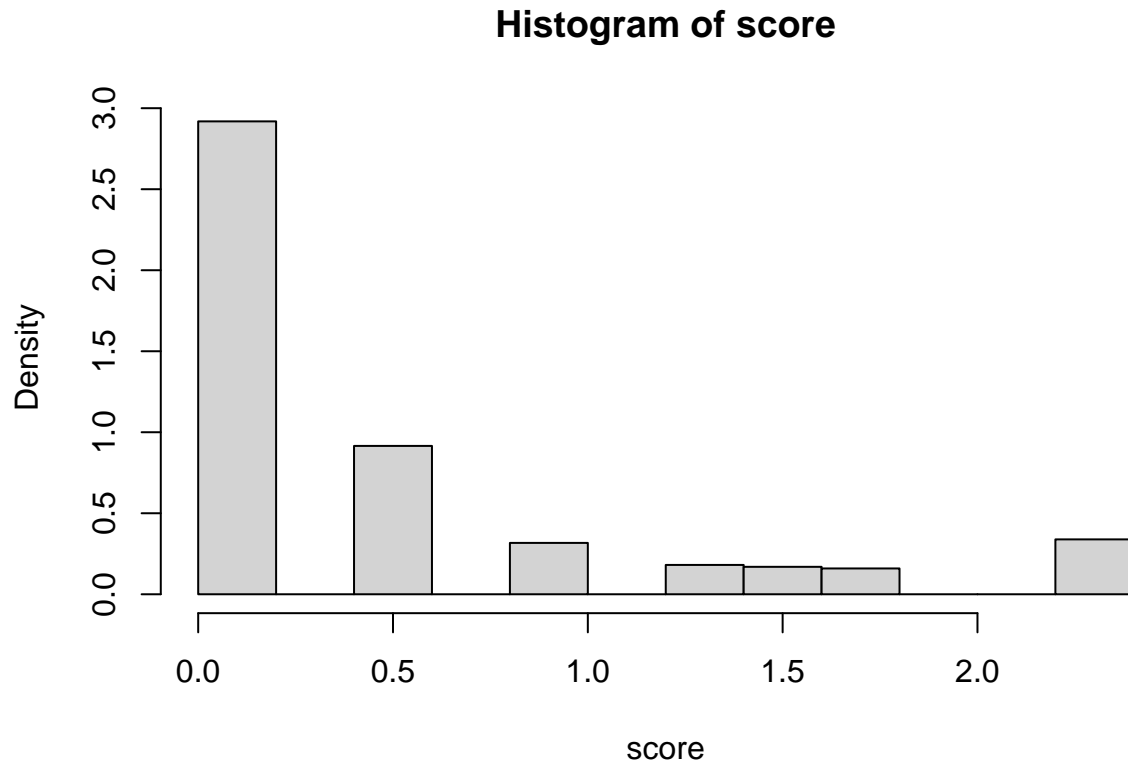



```
score=apply(temp[,-f1],1,sum)
hist(score,freq=F)
```

Histogram of score



```
score=as.matrix(temp[,-f1])%*%mental.fa3$loadings[-f1,2]
hist(score,freq=F)
```



7. Cognitive function

Cognitive loss is a major issue for older people, having huge impact on their daily life.

```
var_list=c(unlist(classified_list[10]))
temp=data.frame(mydata[,var_list])
time_pro=matrix(nrow=5,ncol=6,dimnames = list(var_list[1:5],c("Mean","SD","Min","Max","percent of'0'","percent of'1'")))
for (i in 1: 5){
  tab=table(temp[,i])
  time_pro[i,5]=tab[[1]]/sum(tab)
  time_pro[i,6]=1-time_pro[i,5]
  time_pro[i,1]=mean(temp[,i])
  time_pro[i,2]=sd(temp[,i])
  time_pro[i,3]=min(temp[,i])
  time_pro[i,4]=max(temp[,i])
}

time_pro=data.frame(time_pro)
time_pro
```

##		Mean	SD	Min	Max	percent.of.0.	percent.of.1.
##	cfdatd4	0.00000000	0.0000000	0	0	1.0000000	0.00000000
##	cfdatm4	0.01285999	0.1126806	0	1	0.9871400	0.01285999
##	cfdaty4	0.01086760	0.1036891	0	1	0.9891324	0.01086760
##	cfday4	0.01159210	0.1070505	0	1	0.9884079	0.01159210
##	cfmem4	0.31552255	0.4647657	0	1	0.6844774	0.31552255

```
for (i in 1:ncol(temp)){
  print(attr(temp[,i], "label"))
}
```

```
## [1] "cfdatd Whether correct day of month given "
## [1] "cfdatm Whether correct month given "
## [1] "cfdaty Whether correct year given "
## [1] "cfday Whether correct day given "
## [1] "cfmem Whether prompt given for prospective memory test (remembering to write initials) "
## [1] "cflisenqs (cflisen Number of words recalled immediately) "
## [1] "Cflisdqs( cflisd Number of words recalled after delay) "
## [1] "Cfaniqs (cfani Number of animals mentioned (fluency)) "
```

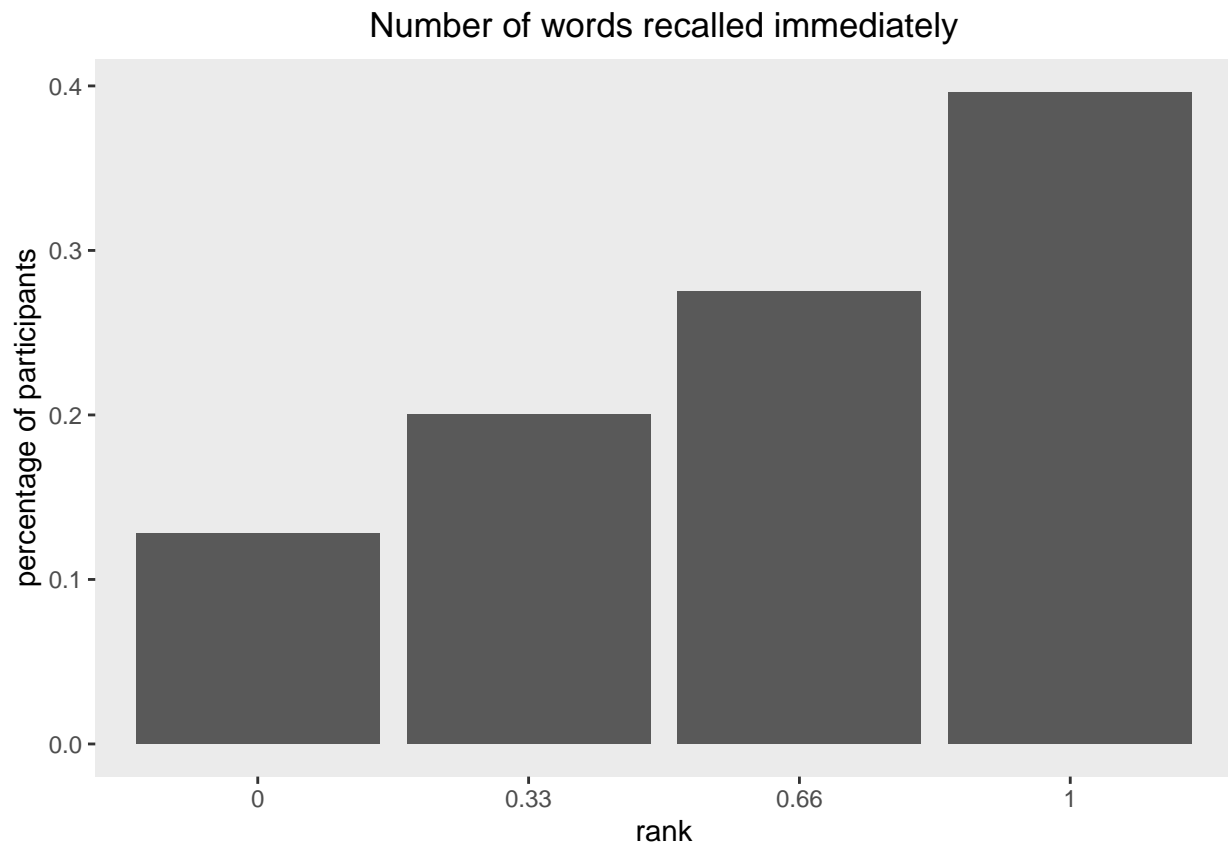
```
temp=mydata[,var_list[6]]
attr(mydata[,var_list[6]], "label")
```

```
## [1] "cflisenqs (cflisen Number of words recalled immediately) "
```

```
table(temp)/5521
```

```
## temp
##      0      0.33      0.66      1
## 0.1282376 0.2001449 0.2753124 0.3963050
```

```
ggplot(NULL, aes(x=as.factor(temp)))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="rank", y="percentage of participants", title = "Number of words recalled immediately")+
  theme(plot.title = element_text(hjust = 0.5), panel.grid = element_blank())
```



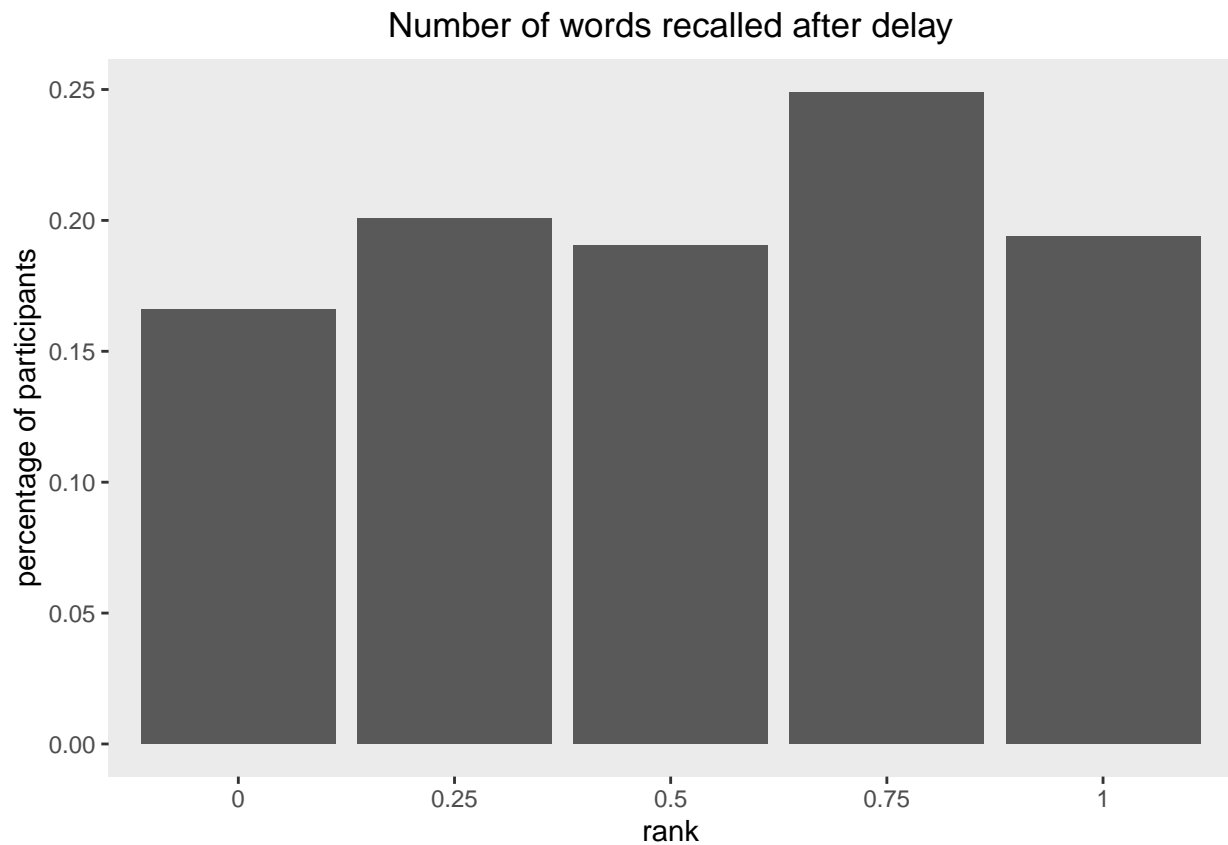
```
temp=mydata[,var_list[8]]
attr(mydata[,var_list[8]],"label")
```

```
## [1] "Cfaniqs (cfani Number of animals mentioned (fluency)) "
```

```
table(temp)/5521
```

```
## temp
##      0      0.25      0.5      0.75      1
## 0.1660931 0.2006883 0.1903641 0.2490491 0.1938055
```

```
ggplot(NULL,aes(x=as.factor(temp)))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="rank",y="percentage of participants",title = "Number of words recalled after delay")+
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```



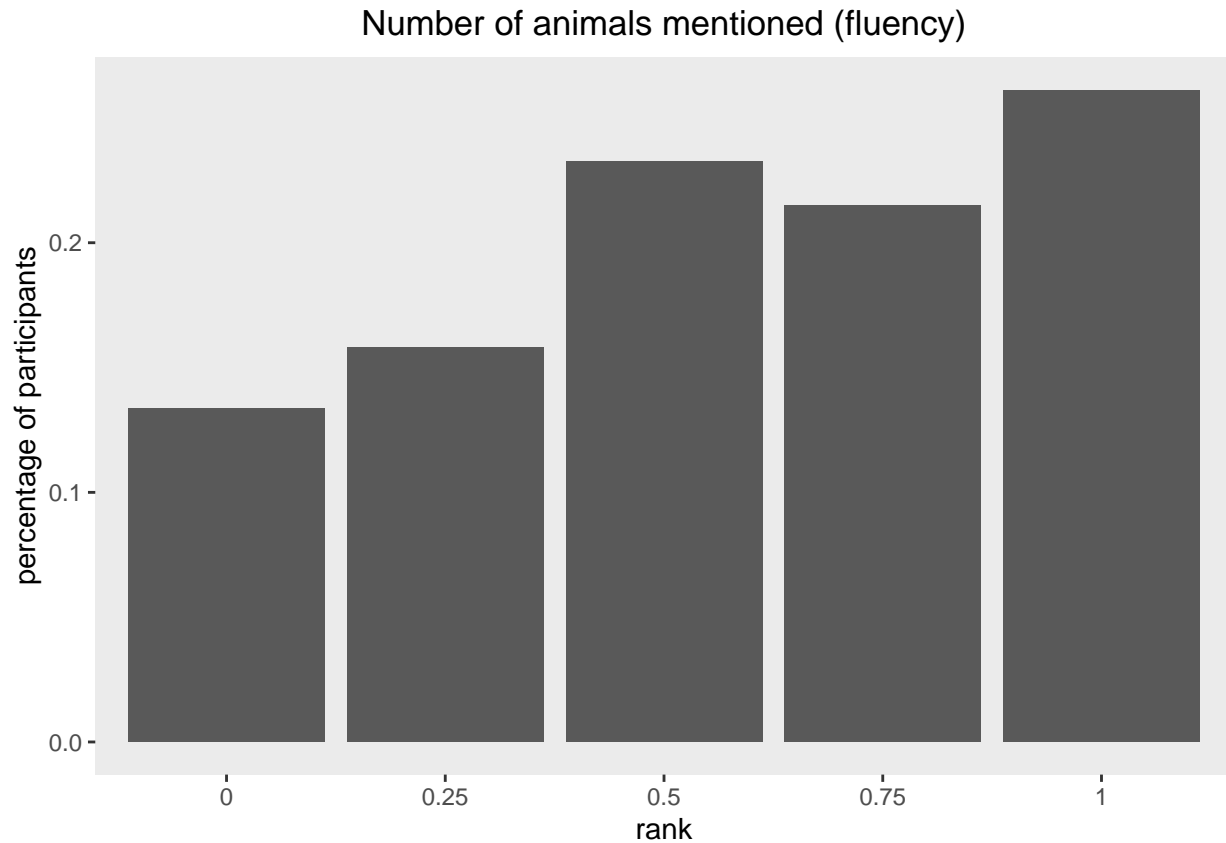
```
temp=mydata[,var_list[7]]
attr(mydata[,var_list[7]],"label")
```

```
## [1] "Cflisdqs( cflisd Number of words recalled after delay) "
```

```
table(temp)/5521
```

```
## temp
##      0      0.25      0.5      0.75      1
## 0.1334903 0.1579424 0.2325666 0.2148162 0.2611846
```

```
ggplot(NULL,aes(x=as.factor(temp)))+
  geom_bar(aes(y=(..count..)/sum(..count..)))+
  labs(x="rank",y="percentage of participants",title = "Number of animals mentioned (fluency)")+
  theme(plot.title = element_text(hjust = 0.5),panel.grid =element_blank())
```



```

name_cognitive=unlist(classified_list[10])
con=mydata[,name_cognitive]
con$cfdatd4[1]=1
con=con[,1:4]
#drop=c("cfdatd4")
#drop=c("cfdatd4","cfmem4")
#con=con[,!(names(con) %in% drop)]
#corre_cognitive=cor(con,method = "pearson")
#con=apply(con,as.factor)
#corre_cognitive=polychoric(con)
corre_cognitive=tetrachoric(con)

#corre_cognitive=corre_cognitive$correlations
corre_cognitive=pmax(corre_cognitive$rho,t(corre_cognitive$rho))
KMO(corre_cognitive)

## Kaiser-Meyer-Olkin factor adequacy
## Call: KMO(r = corre_cognitive)
## Overall MSA = 0.67
## MSA for each item =
## cfdatd4 cfdatm4 cfdaty4 cfday4
## 0.74 0.63 0.66 0.65

bart_spher(corre_cognitive)

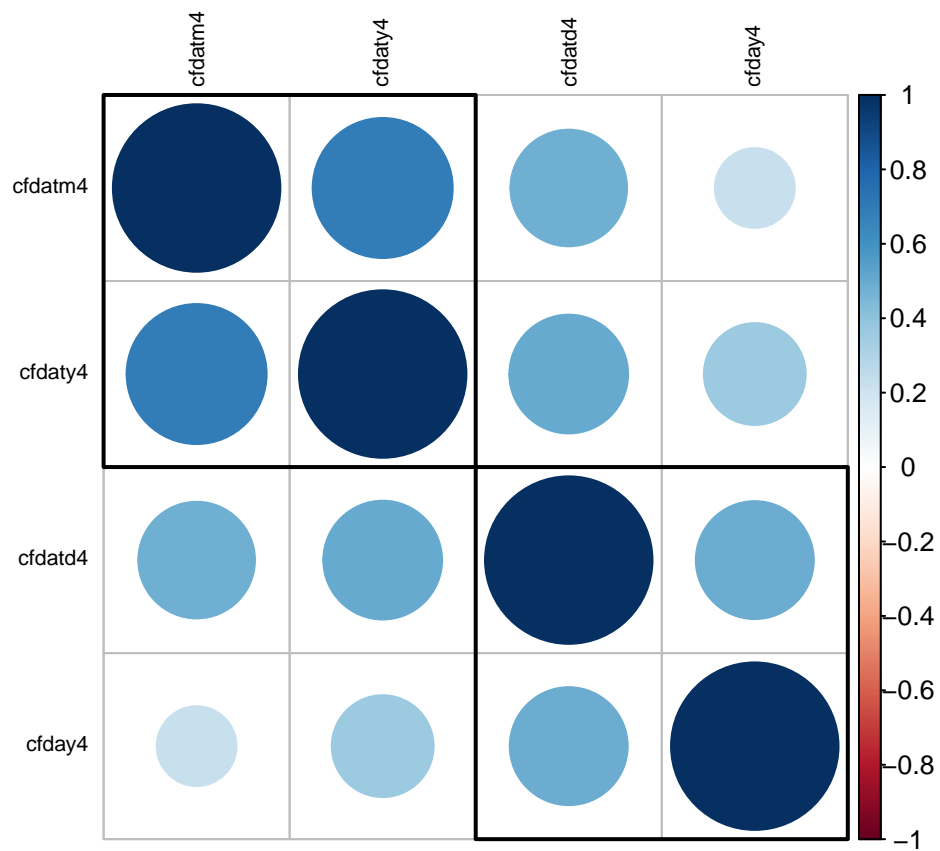
## Bartlett's Test of Sphericity

```

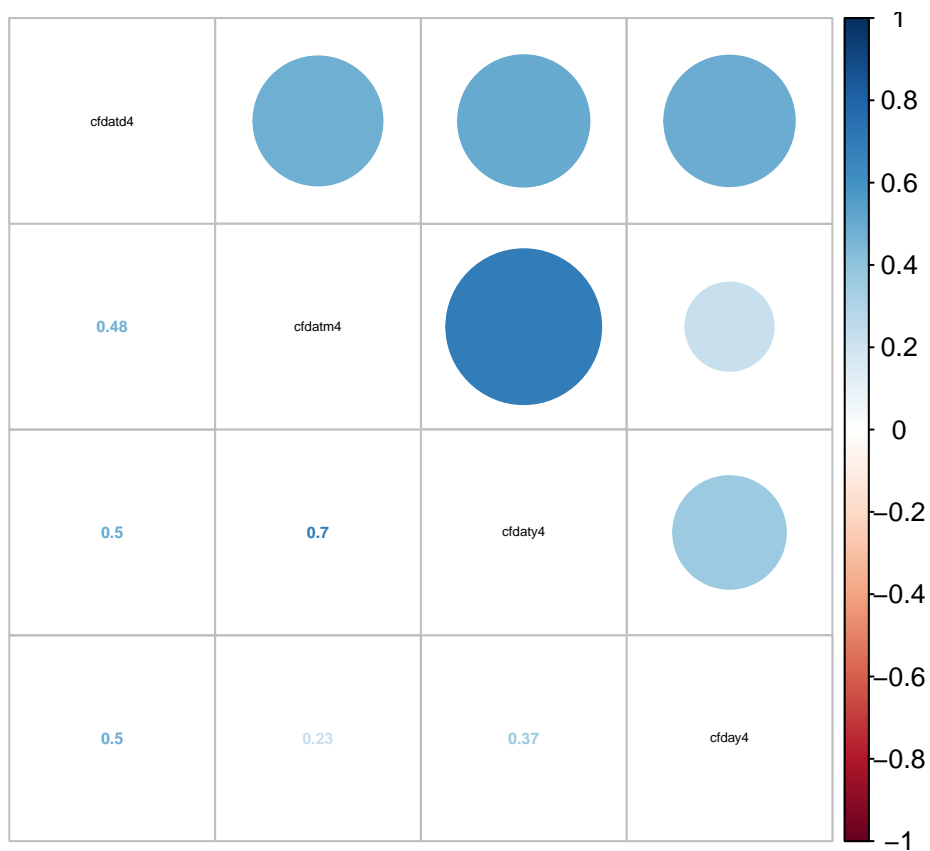
```
##
## Call: bart_spher(x = corre_cognitive)
##
##      X2 = 30.268
##      df = 6
## p-value = .00003
```

```
#cor.plot(corre_cognitive,main = "Corrlation of cognitive function items",labels = #c("Day of month","M
```

```
corrplot(corre_cognitive, is.corr = TRUE, number.cex = .5, tl.col = 'black',order = 'hclust', addrect =
```



```
corrplot.mixed(corre_cognitive, is.corr = TRUE, number.cex = .5, tl.col = 'black',tl.cex = 0.4)
```

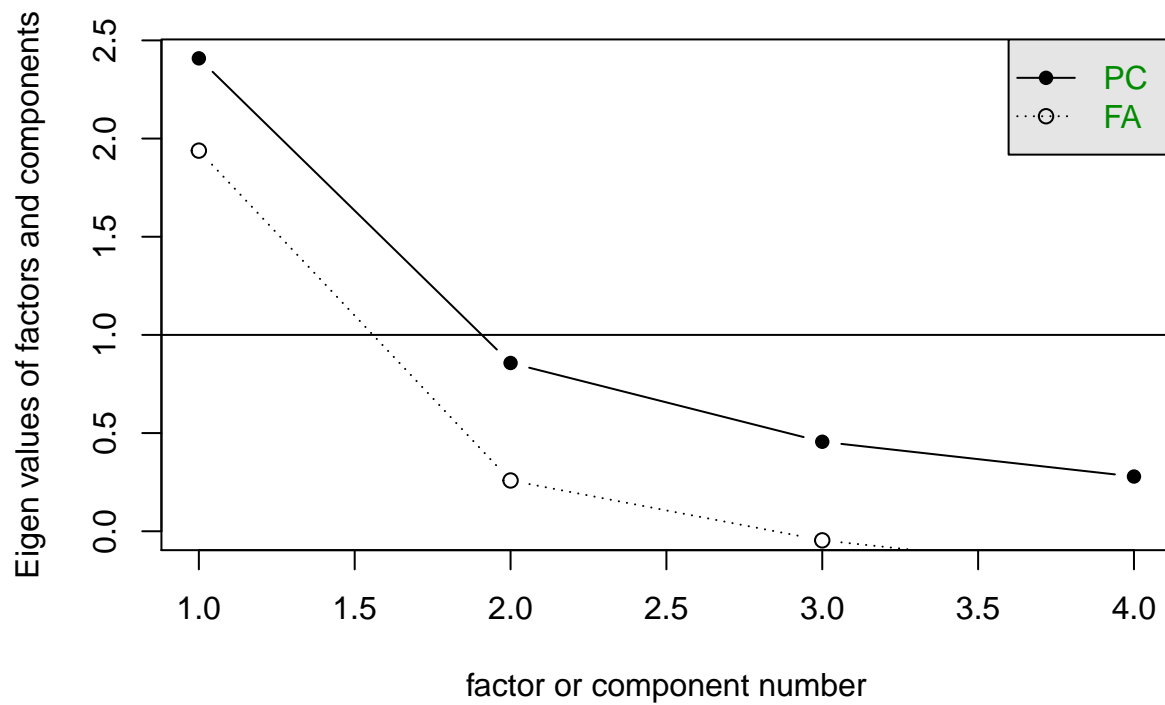


```
eigen.corre <- eigen(corre_cognitive)
eigen.corre$values
```

```
## [1] 2.4085435 0.8570412 0.4557348 0.2786806
```

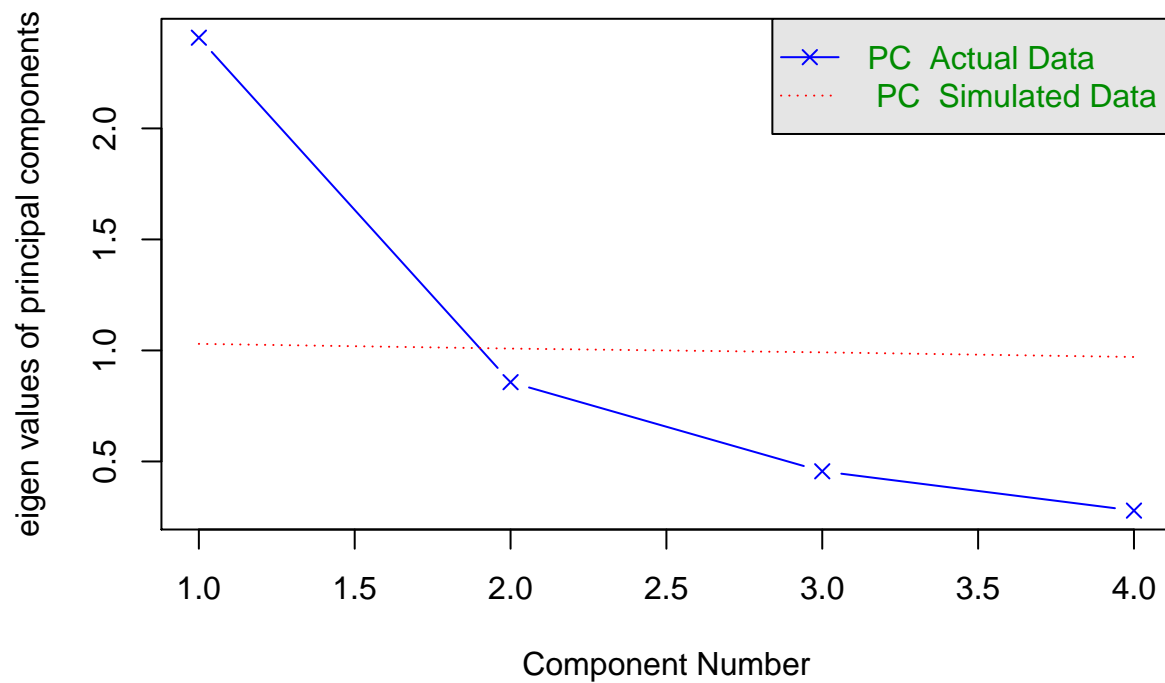
```
scree(corre_cognitive)
```


Scree plot



```
fa.parallel(corre_cognitive, fa="pc", n.obs= nrow(mydata), n.iter = 1000)
```

Parallel Analysis Scree Plots



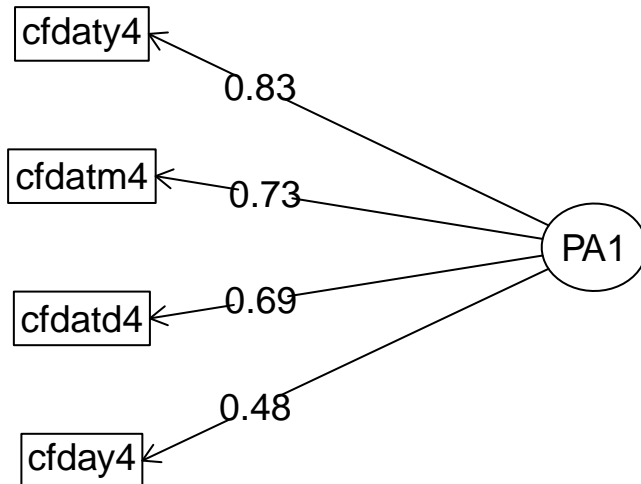
```
## Parallel analysis suggests that the number of factors = NA and the number of components = 1
```

```
cognitive.fal<- fa(corre_cognitive, nfactors = 1, rotate = 'none',n.obs=4800,scores = 'regression', fm = "pa")
print(cognitive.fal,sort = T)
```

```
## Factor Analysis using method = pa
## Call: fa(r = corre_cognitive, nfactors = 1, n.obs = 4800, rotate = "none",
##       scores = "regression", fm = "pa")
## Standardized loadings (pattern matrix) based upon correlation matrix
##      V  PA1   h2   u2 com
## cfdaty4 3 0.83 0.69 0.31   1
## cfdatm4 2 0.73 0.54 0.46   1
## cfdatd4 1 0.69 0.48 0.52   1
## cfday4  4 0.48 0.23 0.77   1
##
##              PA1
## SS loadings    1.94
## Proportion Var 0.48
##
## Mean item complexity = 1
## Test of the hypothesis that 1 factor is sufficient.
##
## The degrees of freedom for the null model are 6 and the objective function was 1.34 with Chi Square = 547.28
## The degrees of freedom for the model are 2 and the objective function was 0.19 with Chi Square = 925.03
##
## The root mean square of the residuals (RMSR) is 0.1
## The df corrected root mean square of the residuals is 0.17
##
## The harmonic number of observations is 4800 with the empirical chi square 547.28 with prob < 1.4e-10
## The total number of observations was 4800 with Likelihood Chi Square = 925.03 with prob < 1.4e-10
##
## Tucker Lewis Index of factoring reliability = 0.568
## RMSEA index = 0.31 and the 90 % confidence intervals are 0.293 0.327
## BIC = 908.07
## Fit based upon off diagonal values = 0.96
## Measures of factor score adequacy
##
##              PA1
## Correlation of (regression) scores with factors 0.91
## Multiple R square of scores with factors        0.82
## Minimum correlation of possible factor scores    0.64
```

```
fa.diagram(cognitive.fal, cut=0.299,labels=names(cognitive.fal),digits=2,e.size=0.05,rsiz=0.1,cex=1.2,
```

Factor Analysis



```

score6=factor.scores(con,cognitive.fa1)
cognitive_factor=score6$scores

Orientation_time=as.matrix(con)%*%cognitive.fa1$loadings[1:4,1]
  
```

1 factor is extracted for date naming items and named as orientation in time.

8. Correlation between factors

```

# factor_matrix=data.frame(Upper_limb,Lower_limb,Other,ADLs_disability,IADLs_disability,Heart_diseases,
#
# cor.plot(cor(factor_matrix), main= "Correlation between all factors")

factor_name=c( "Upper_limb", "Lower_limb", "Other", "ADLs_disability", "IADLs_disability", "Heart_diseases"
factor_score=data.frame(mobility_factor,disability_factor,cvd_factor,Movement_factor,mental_factor,cogn
names(factor_score)=factor_name
cor.plot(cor(factor_score), main= "Correlation between all factors",show.legend = F, cex.axis=0.7,xlas :
  
```

Correlation between all factors

Upper_limb	1	0.77	0.68	0.6	0.34	0.21	0.19	0.17	0.24	0.41	0.03
Lower_limb	0.77	1	0.69	0.58	0.32	0.24	0.24	0.16	0.22	0.41	0.04
Other	0.68	0.69	1	0.53	0.27	0.14	0.17	0.18	0.22	0.36	0.02
ADLs_disability	0.6	0.58	0.53	1	0.55	0.16	0.17	0.17	0.19	0.34	0.05
IADLs_disability	0.34	0.32	0.27	0.55	1	0.13	0.14	0.09	0.12	0.19	0.09
Heart_diseases	0.21	0.24	0.14	0.16	0.13	1	0.39	0.04	0.07	0.12	0.02
Other_CVD	0.19	0.24	0.17	0.17	0.14	0.39	1	0.01	0.06	0.13	0.02
Movement	0.17	0.16	0.18	0.17	0.09	0.04	0.01	1	0.07	0.09	0.02
Positive_anhodonina	0.24	0.22	0.22	0.19	0.12	0.07	0.06	0.07	1	0.69	0.02
Somatic_activity	0.41	0.41	0.36	0.34	0.19	0.12	0.13	0.09	0.69	1	0.02
Orientation_time	0.03	0.04	0.02	0.05	0.09	0.02	0.02	0.02	0.02	0.02	1
	Upper_limb	Lower_limb	Other	ADLs_disability	IADLs_disability	Heart_diseases	Other_CVD	Movement	Positive_anhodonina	Somatic_activity	Orientation_time

11 factors are identified under domains of the FI. Correlation of factors across domains are assessed. 3 factors of mobility: upper limb difficulty, lower limb difficulty and other difficulty, ADLs and IADLs in disability and somatic activity or inactivity are related. First, 3 factors of mobility and the 2 factors of disability are both symptoms or the result of conditions or diseases of physical health. In fact, it is common to assess disability with mobility items along with ADLs and IADLs, though for the case of ELSA, they are considered separately. Mobility and disability may be a result of common health conditions or diseases, such as the loss of muscle mass, quality, strength, Osteoporosis, neural weakness etc. Consequently, in this study, domain mobility and domain disability are correlated.

The third domain involving correlation is CES-D measured mental health, specifically, somatic activity or inactivity. There is evidence showing depression is associated with disability in elderly people, meanwhile the association between somatic symptom has also been identified. It therefore makes sense that somatic activity or inactivity are correlated with disability and mobility.