EFA Project

Mo’men Mohamed

March 5, 2019

### This is An EFA Project explaining the practice and skills needed

## Steps

The following is the steps of running **EFA**

1- Testing the Assumptions

2- Specifying the number of factors

3- Choosing the factor math and rotation type

4- Exploring the Adequacy of EFA model

#### First of all lets load all libraries which may need to

library(psych) ; library(GPArotation)  
library(bnstruct)  
library(bestNormalize)  
library(knitr)

#### - Now Lets import our data to work with

efadf<-read.csv(file = "3 efa.csv")  
head(efadf)

## o1 o2 o3 o4 o5 o6 o7 o8 o9 o10 o11 o12 o13 o14 o15 o16 o17 o18 o19 o20  
## 1 2 4 3 4 4 4 4 3 4 4 2 2 1 2 3 5 2 4 2 2  
## 2 4 4 3 3 3 4 3 3 4 3 3 4 3 3 4 3 4 3 3 3  
## 3 5 5 2 5 2 1 1 5 2 2 4 4 2 4 1 4 1 1 2 4  
## 4 4 3 4 3 5 4 4 3 4 5 2 2 2 2 NA NA NA NA NA NA  
## 5 1 1 3 5 4 5 1 1 5 1 5 5 1 5 3 5 1 5 1 5  
## 6 5 4 3 5 5 5 4 4 5 5 1 1 1 1 5 1 1 3 1 2  
## condition  
## 1 complimentary  
## 2 benevolent  
## 3 complimentary  
## 4 complimentary  
## 5 neutral  
## 6 neutral

efadf<-efadf[,-21]

#### here we have to take a look on the data structure and attr.for starting testing the assumptions

## 1- testing the assumption

#### it includes :

##### *Accurecy & Missing Value*

str(efadf)

## 'data.frame': 99 obs. of 20 variables:  
## $ o1 : int 2 4 5 4 1 5 4 4 4 5 ...  
## $ o2 : int 4 4 5 3 1 4 4 5 5 5 ...  
## $ o3 : int 3 3 2 4 3 3 3 3 3 4 ...  
## $ o4 : int 4 3 5 3 5 5 4 5 4 4 ...  
## $ o5 : int 4 3 2 5 4 5 4 5 5 5 ...  
## $ o6 : int 4 4 1 4 5 5 3 5 5 5 ...  
## $ o7 : int 4 3 1 4 1 4 3 4 3 3 ...  
## $ o8 : int 3 3 5 3 1 4 2 2 3 5 ...  
## $ o9 : int 4 4 2 4 5 5 4 5 5 5 ...  
## $ o10: int 4 3 2 5 1 5 2 5 5 3 ...  
## $ o11: int 2 3 4 2 5 1 3 3 1 1 ...  
## $ o12: int 2 4 4 2 5 1 2 3 2 2 ...  
## $ o13: int 1 3 2 2 1 1 4 2 2 4 ...  
## $ o14: int 2 3 4 2 5 1 2 3 2 2 ...  
## $ o15: int 3 4 1 NA 3 5 3 3 3 2 ...  
## $ o16: int 5 3 4 NA 5 1 2 1 1 2 ...  
## $ o17: int 2 4 1 NA 1 1 3 1 1 2 ...  
## $ o18: int 4 3 1 NA 5 3 3 2 1 2 ...  
## $ o19: int 2 3 2 NA 1 1 3 4 1 2 ...  
## $ o20: int 2 3 4 NA 5 2 2 2 2 2 ...

#### as we see here we have a problem with the class of numeric value

for(i in 1:(ncol(efadf))){  
 efadf[,i]<-as.numeric(efadf[,i])  
}  
str(efadf)

## 'data.frame': 99 obs. of 20 variables:  
## $ o1 : num 2 4 5 4 1 5 4 4 4 5 ...  
## $ o2 : num 4 4 5 3 1 4 4 5 5 5 ...  
## $ o3 : num 3 3 2 4 3 3 3 3 3 4 ...  
## $ o4 : num 4 3 5 3 5 5 4 5 4 4 ...  
## $ o5 : num 4 3 2 5 4 5 4 5 5 5 ...  
## $ o6 : num 4 4 1 4 5 5 3 5 5 5 ...  
## $ o7 : num 4 3 1 4 1 4 3 4 3 3 ...  
## $ o8 : num 3 3 5 3 1 4 2 2 3 5 ...  
## $ o9 : num 4 4 2 4 5 5 4 5 5 5 ...  
## $ o10: num 4 3 2 5 1 5 2 5 5 3 ...  
## $ o11: num 2 3 4 2 5 1 3 3 1 1 ...  
## $ o12: num 2 4 4 2 5 1 2 3 2 2 ...  
## $ o13: num 1 3 2 2 1 1 4 2 2 4 ...  
## $ o14: num 2 3 4 2 5 1 2 3 2 2 ...  
## $ o15: num 3 4 1 NA 3 5 3 3 3 2 ...  
## $ o16: num 5 3 4 NA 5 1 2 1 1 2 ...  
## $ o17: num 2 4 1 NA 1 1 3 1 1 2 ...  
## $ o18: num 4 3 1 NA 5 3 3 2 1 2 ...  
## $ o19: num 2 3 2 NA 1 1 3 4 1 2 ...  
## $ o20: num 2 3 4 NA 5 2 2 2 2 2 ...

#### Great !!

#### Now lets look at the summary of the data

desc<-apply(efadf,2,function(x){  
 (  
 round(c(  
 min=min(x,na.rm = T),  
 max=max(x,na.rm = T),  
 mean=mean(x,na.rm = T),  
 na=sum(is.na(x))/nrow(efadf)  
 ),2)  
 )  
})  
  
print(desc)

## o1 o2 o3 o4 o5 o6 o7 o8 o9 o10 o11 o12 o13 o14  
## min 1.00 1.00 1.00 1.00 2.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00 1.00  
## max 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00 5.00  
## mean 3.68 3.96 2.96 3.75 4.24 4.23 3.69 3.56 4.26 3.44 2.47 2.32 2.60 2.78  
## na 0.01 0.00 0.00 0.00 0.00 0.02 0.02 0.02 0.02 0.02 0.02 0.02 0.03 0.01  
## o15 o16 o17 o18 o19 o20  
## min 1.00 1.00 1.00 1.00 1.00 1.00  
## max 5.00 5.00 5.00 5.00 5.00 5.00  
## mean 3.15 2.76 2.13 2.72 2.64 2.54  
## na 0.02 0.02 0.02 0.03 0.02 0.04

#### is there are any item has missing more than 5% ?

any(desc["na",]>.05)

## [1] FALSE

#### great there is no item with more than 5% missing value

#### now test the missing for each individual

ind.desc<-apply(efadf,1,function(x){  
 round(c(sum(is.na(x))/ncol(efadf)),2)  
})  
  
any(ind.desc>.05)

## [1] TRUE

which(ind.desc>.05)

## [1] 4 39 68 99

ind.desc[which(ind.desc>.05)]

## [1] 0.30 0.45 0.35 0.40

#### woow it is so high percent

#### we will delete these members from the data

efadf<-efadf[-which(ind.desc>.05),]

#### Now we have to impute the missing value

efadf<-knn.impute(as.matrix(efadf),k = 5)%>%data.frame()

#### finally lets reverse the some items !!

key<-rep(1,ncol(efadf))  
  
key[c(1,2,11,13,15,17,19,20)]<-key[c(1,2,11,13,15,17,19,20)]\*-1  
  
efadf<-reverse.code(keys = key,items = efadf)

#### congrats: we finished this part **Acurecy and missing**

Now I am going to test some assumption related to factor analysis like:

1- **Additivity**

2- **enough correlation between items**

3- **adequacy of sample number**

#Additivity   
  
corr<-cor(efadf,method = "p")  
  
apply(corr,2,function(x){  
 any(x<1&x>=abs(.9))  
})

## o1- o2- o3 o4 o5 o6 o7 o8 o9 o10 o11- o12   
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE   
## o13- o14 o15- o16 o17- o18 o19- o20-   
## FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE

#Ok !! there is no additivity   
  
#Now test the enough correlation using correlation bartelett test  
  
cortest.bartlett(efadf)

## R was not square, finding R from data

## $chisq  
## [1] 870.0704  
##   
## $p.value  
## [1] 1.435168e-87  
##   
## $df  
## [1] 190

#significant result indicate enough correlation   
  
#Finally KMO test for sample size adequacy   
  
KMO(efadf)

## Kaiser-Meyer-Olkin factor adequacy  
## Call: KMO(r = efadf)  
## Overall MSA = 0.81  
## MSA for each item =   
## o1- o2- o3 o4 o5 o6 o7 o8 o9 o10 o11- o12 o13- o14 o15-   
## 0.79 0.81 0.43 0.69 0.84 0.86 0.60 0.76 0.85 0.88 0.87 0.84 0.82 0.91 0.45   
## o16 o17- o18 o19- o20-   
## 0.86 0.87 0.81 0.75 0.94

#here we can see that the data roughly adequate as it is >.8

Now it s the time to dive more in EFA

## 2- taking decision about the number of factors

by following the next method

1- **Parallel test**

2- **Kaiser criterion**

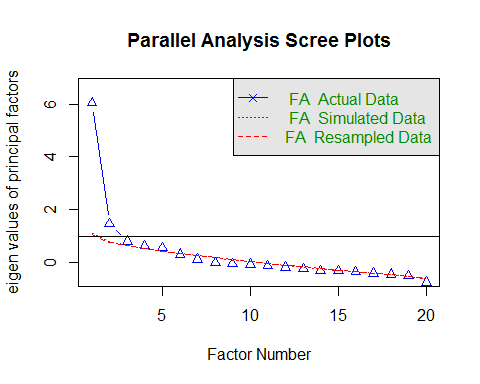
3- **scree plot**

nf<-fa.parallel(x = efadf,fa = "fa")#suggestion of parallel test is 5 factors

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : A loading greater than abs(1) was detected. Examine the loadings  
## carefully.

## Warning in fa.stats(r = r, f = f, phi = phi, n.obs = n.obs, np.obs  
## = np.obs, : The estimated weights for the factor scores are probably  
## incorrect. Try a different factor extraction method.

## Warning in fac(r = r, nfactors = nfactors, n.obs = n.obs, rotate =  
## rotate, : An ultra-Heywood case was detected. Examine the results carefully

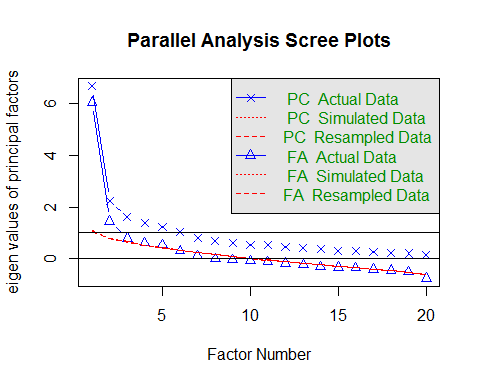


## Parallel analysis suggests that the number of factors = 5 and the number of components = NA

sum(nf$fa.values>.7) #Kaiser criterion suggest 3 factors

## [1] 3

plot(nf) #scree plot suggest 2-3 factor



#### lets try 3 factors and look at the adequacy of the model

efafit<-fa(efadf,nfactors = 3,rotate = "oblimin")  
print(efafit)

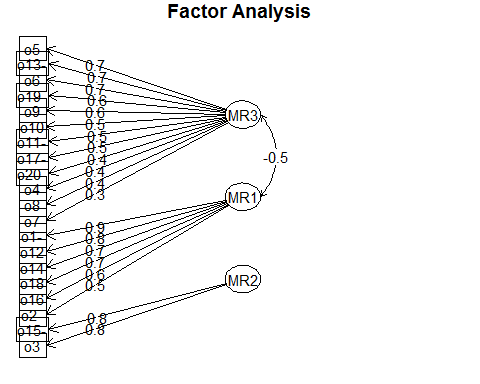
## Factor Analysis using method = minres  
## Call: fa(r = efadf, nfactors = 3, rotate = "oblimin")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## MR3 MR1 MR2 h2 u2 com  
## o1- 0.13 0.88 -0.01 0.69 0.31 1.0  
## o2- -0.12 0.52 0.10 0.34 0.66 1.2  
## o3 0.06 0.06 0.83 0.67 0.33 1.0  
## o4 0.43 0.07 -0.09 0.17 0.83 1.1  
## o5 0.68 -0.01 0.03 0.47 0.53 1.0  
## o6 0.65 0.08 -0.05 0.39 0.61 1.0  
## o7 0.31 -0.09 -0.04 0.13 0.87 1.2  
## o8 0.36 -0.14 0.17 0.23 0.77 1.8  
## o9 0.64 -0.04 0.19 0.47 0.53 1.2  
## o10 0.54 -0.16 -0.03 0.40 0.60 1.2  
## o11- 0.53 -0.36 0.05 0.60 0.40 1.8  
## o12 -0.04 0.82 0.04 0.69 0.31 1.0  
## o13- 0.66 0.11 -0.04 0.38 0.62 1.1  
## o14 -0.17 0.69 0.00 0.63 0.37 1.1  
## o15- -0.05 -0.05 0.83 0.71 0.29 1.0  
## o16 -0.23 0.57 0.11 0.50 0.50 1.4  
## o17- 0.52 -0.18 -0.04 0.39 0.61 1.2  
## o18 0.02 0.65 -0.24 0.52 0.48 1.3  
## o19- 0.64 0.08 -0.04 0.37 0.63 1.0  
## o20- 0.44 -0.29 -0.06 0.40 0.60 1.8  
##   
## MR3 MR1 MR2  
## SS loadings 4.01 3.58 1.56  
## Proportion Var 0.20 0.18 0.08  
## Cumulative Var 0.20 0.38 0.46  
## Proportion Explained 0.44 0.39 0.17  
## Cumulative Proportion 0.44 0.83 1.00  
##   
## With factor correlations of   
## MR3 MR1 MR2  
## MR3 1.00 -0.50 -0.03  
## MR1 -0.50 1.00 -0.15  
## MR2 -0.03 -0.15 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 190 and the objective function was 10.06 with Chi Square of 870.07  
## The degrees of freedom for the model are 133 and the objective function was 2.57   
##   
## The root mean square of the residuals (RMSR) is 0.07   
## The df corrected root mean square of the residuals is 0.08   
##   
## The harmonic number of observations is 95 with the empirical chi square 163.95 with prob < 0.035   
## The total number of observations was 95 with Likelihood Chi Square = 217.21 with prob < 5.6e-06   
##   
## Tucker Lewis Index of factoring reliability = 0.818  
## RMSEA index = 0.093 and the 90 % confidence intervals are 0.062 0.101  
## BIC = -388.46  
## Fit based upon off diagonal values = 0.96  
## Measures of factor score adequacy   
## MR3 MR1 MR2  
## Correlation of (regression) scores with factors 0.93 0.95 0.91  
## Multiple R square of scores with factors 0.87 0.90 0.83  
## Minimum correlation of possible factor scores 0.74 0.80 0.66

#extracting the fit measure   
  
fitm<-c(RMS=efafit$rms,  
RMSEA=efafit$RMSEA,  
TLI=efafit$TLI,  
CFI=(  
 1-(efafit$STATISTIC-efafit$dof)/(efafit$null.chisq-efafit$null.dof)  
))  
print(fitm)

## RMS RMSEA.RMSEA RMSEA.lower RMSEA.upper   
## 0.06739189 0.09321294 0.06178499 0.10141980   
## RMSEA.confidence TLI CFI   
## 0.90000000 0.81772318 0.87618059

#### the model here is poor but lets take a look on the loading and drawing the diagram

fa.diagram(efafit)



efafit$loadings

##   
## Loadings:  
## MR3 MR1 MR2   
## o1- 0.126 0.882   
## o2- -0.117 0.523 0.104  
## o3 0.828  
## o4 0.427   
## o5 0.681   
## o6 0.653   
## o7 0.307   
## o8 0.358 -0.139 0.174  
## o9 0.639 0.185  
## o10 0.539 -0.157   
## o11- 0.527 -0.356   
## o12 0.816   
## o13- 0.658 0.107   
## o14 -0.174 0.694   
## o15- 0.832  
## o16 -0.234 0.566 0.112  
## o17- 0.519 -0.176   
## o18 0.652 -0.244  
## o19- 0.642   
## o20- 0.443 -0.285   
##   
## MR3 MR1 MR2  
## SS loadings 3.704 3.270 1.552  
## Proportion Var 0.185 0.164 0.078  
## Cumulative Var 0.185 0.349 0.426

#### as we see here that all items are good regarding its loading to factors

#### So, lets try to repeat the analysis with 2 factors instead of 3

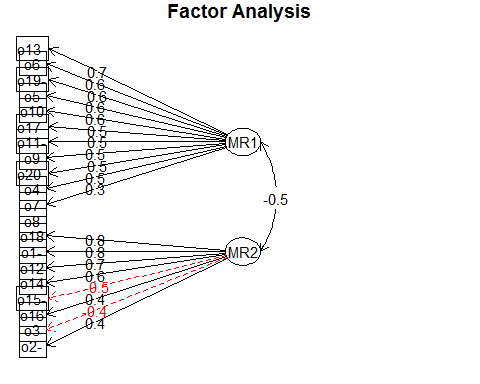
efafit2<-fa(efadf,nfactors = 2,rotate = "oblimin")  
  
print(efafit2)

## Factor Analysis using method = minres  
## Call: fa(r = efadf, nfactors = 2, rotate = "oblimin")  
## Standardized loadings (pattern matrix) based upon correlation matrix  
## MR1 MR2 h2 u2 com  
## o1- 0.01 0.78 0.60 0.40 1.0  
## o2- -0.23 0.41 0.30 0.70 1.6  
## o3 -0.29 -0.41 0.15 0.85 1.8  
## o4 0.46 0.11 0.17 0.83 1.1  
## o5 0.63 -0.06 0.44 0.56 1.0  
## o6 0.64 0.08 0.37 0.63 1.0  
## o7 0.32 -0.07 0.13 0.87 1.1  
## o8 0.26 -0.26 0.20 0.80 2.0  
## o9 0.51 -0.19 0.39 0.61 1.3  
## o10 0.55 -0.14 0.40 0.60 1.1  
## o11- 0.53 -0.38 0.60 0.40 1.8  
## o12 -0.17 0.69 0.61 0.39 1.1  
## o13- 0.65 0.11 0.37 0.63 1.1  
## o14 -0.25 0.63 0.61 0.39 1.3  
## o15- -0.37 -0.49 0.21 0.79 1.9  
## o16 -0.35 0.44 0.46 0.54 1.9  
## o17- 0.54 -0.15 0.39 0.61 1.2  
## o18 0.09 0.78 0.56 0.44 1.0  
## o19- 0.63 0.08 0.36 0.64 1.0  
## o20- 0.49 -0.24 0.40 0.60 1.4  
##   
## MR1 MR2  
## SS loadings 4.19 3.54  
## Proportion Var 0.21 0.18  
## Cumulative Var 0.21 0.39  
## Proportion Explained 0.54 0.46  
## Cumulative Proportion 0.54 1.00  
##   
## With factor correlations of   
## MR1 MR2  
## MR1 1.00 -0.45  
## MR2 -0.45 1.00  
##   
## Mean item complexity = 1.3  
## Test of the hypothesis that 2 factors are sufficient.  
##   
## The degrees of freedom for the null model are 190 and the objective function was 10.06 with Chi Square of 870.07  
## The degrees of freedom for the model are 151 and the objective function was 3.5   
##   
## The root mean square of the residuals (RMSR) is 0.09   
## The df corrected root mean square of the residuals is 0.1   
##   
## The harmonic number of observations is 95 with the empirical chi square 271.44 with prob < 6.6e-09   
## The total number of observations was 95 with Likelihood Chi Square = 297.66 with prob < 1.2e-11   
##   
## Tucker Lewis Index of factoring reliability = 0.723  
## RMSEA index = 0.112 and the 90 % confidence intervals are 0.085 0.119  
## BIC = -389.97  
## Fit based upon off diagonal values = 0.93  
## Measures of factor score adequacy   
## MR1 MR2  
## Correlation of (regression) scores with factors 0.93 0.94  
## Multiple R square of scores with factors 0.87 0.88  
## Minimum correlation of possible factor scores 0.74 0.75

#extracting the fit measure   
  
fitm2<-c(RMS=efafit2$rms,  
RMSEA=efafit2$RMSEA,  
TLI=efafit2$TLI,  
CFI=(  
 1-(efafit2$STATISTIC-efafit2$dof)/(efafit2$null.chisq-efafit2$null.dof)  
))  
  
print(fitm2)

## RMS RMSEA.RMSEA RMSEA.lower RMSEA.upper   
## 0.08671335 0.11183752 0.08455481 0.11859502   
## RMSEA.confidence TLI CFI   
## 0.90000000 0.72318726 0.78434514

fa.diagram(efafit2)



efafit2$loadings

##   
## Loadings:  
## MR1 MR2   
## o1- 0.779  
## o2- -0.226 0.409  
## o3 -0.291 -0.411  
## o4 0.457 0.113  
## o5 0.634   
## o6 0.641   
## o7 0.320   
## o8 0.264 -0.263  
## o9 0.509 -0.195  
## o10 0.554 -0.144  
## o11- 0.526 -0.375  
## o12 -0.166 0.694  
## o13- 0.653 0.110  
## o14 -0.254 0.630  
## o15- -0.372 -0.486  
## o16 -0.352 0.442  
## o17- 0.544 -0.151  
## o18 0.783  
## o19- 0.635   
## o20- 0.493 -0.236  
##   
## MR1 MR2  
## SS loadings 3.904 3.260  
## Proportion Var 0.195 0.163  
## Cumulative Var 0.195 0.358

#### As we see here that there are a great distortion in the loading with a huge drop in CFI

#### so we will back to 3 factor model

#### The following codes for reporting in tables

fa.load<-matrix(efafit$loadings,ncol = 3,dimnames=list(c(paste0("q",1:20)),c("F1","F2","F3"))  
)  
  
fa.var<-matrix(efafit$Vaccounted,ncol = 3,dimnames = list(c("SS loading","Prop var","Cum var","Prop explain","Cum Prop"),c(paste0("MR",1:3))))  
  
fa.cor<-matrix(efafit$score.cor,ncol = 3,dimnames = list(c(paste0("MR",1:3)),c(paste0("MR",1:3))))  
  
efa.fit<-data.frame(fitm)  
  
#Tables  
kable(fa.var,format = "markdown",align = "c",digits = 3,caption = "Table1 : Variance Explanation for factors")

|  |  |  |  |
| --- | --- | --- | --- |
|  | MR1 | MR2 | MR3 |
| SS loading | 4.006 | 3.579 | 1.559 |
| Prop var | 0.200 | 0.179 | 0.078 |
| Cum var | 0.200 | 0.379 | 0.457 |
| Prop explain | 0.438 | 0.391 | 0.170 |
| Cum Prop | 0.438 | 0.830 | 1.000 |

kable(fa.cor,format = "markdown",align = "c",digits = 3,caption = "Table2: the correlation between factors")

|  |  |  |  |
| --- | --- | --- | --- |
|  | MR1 | MR2 | MR3 |
| MR1 | 1.000 | -0.577 | 0.007 |
| MR2 | -0.577 | 1.000 | -0.117 |
| MR3 | 0.007 | -0.117 | 1.000 |

kable(fa.load,format = "markdown",align = "c",digits = 3,caption = "Table3: Item Factor Loading")

|  |  |  |  |
| --- | --- | --- | --- |
|  | F1 | F2 | F3 |
| q1 | 0.126 | 0.882 | -0.015 |
| q2 | -0.117 | 0.523 | 0.104 |
| q3 | 0.065 | 0.056 | 0.828 |
| q4 | 0.427 | 0.069 | -0.086 |
| q5 | 0.681 | -0.010 | 0.027 |
| q6 | 0.653 | 0.079 | -0.049 |
| q7 | 0.307 | -0.086 | -0.043 |
| q8 | 0.358 | -0.139 | 0.174 |
| q9 | 0.639 | -0.042 | 0.185 |
| q10 | 0.539 | -0.157 | -0.026 |
| q11 | 0.527 | -0.356 | 0.047 |
| q12 | -0.039 | 0.816 | 0.042 |
| q13 | 0.658 | 0.107 | -0.043 |
| q14 | -0.174 | 0.694 | 0.004 |
| q15 | -0.046 | -0.047 | 0.832 |
| q16 | -0.234 | 0.566 | 0.112 |
| q17 | 0.519 | -0.176 | -0.039 |
| q18 | 0.023 | 0.652 | -0.244 |
| q19 | 0.642 | 0.083 | -0.036 |
| q20 | 0.443 | -0.285 | -0.060 |

kable(efa.fit,format = "latex",align = "c",digits = 3,caption = "Table4: Fit measures")