RECSM: Quantitative Methods in Social Research

Day 3 - 04 07 2025

Burak Sonmez

Measurement of the Unobserved Variables and Scale Construction.

Final Problem Set

Step 1: Open a new script and save it as day3. Clear your workspace.

Step 2: Load the General Social Survey (gss2016.RData) dataset and install "Psych" alongside "psychTools" package.

Step 3: Once you subset the ten variables related to confidence in institutions, you can explore the correlations among those variables in the dataset.

Step 4: Try to create a new variable that combines all 10 likert-scale variables (hint: use rowMeans function).

Step 5: Test the reliability of this scale through Cronbach's alpha (hint: use "alpha" function.

Step 6: Test for the number of factors in your data using parallel analysis.

Step 7: Do the same analysis above with Principle Component Analysis (PCA).

Step 8: Use factor analysis with the data. Then compare the solution to a hierarchical cluster analysis using the "ICLUST" algorithm and function (Revelle, 1979).

Solutions for problem sets

```r
#Remove objects from the environment
rm(list=ls())
#Set your working directory
setwd("~/Downloads/RECSM workshop")
#Load the dataset
load("gss2016.RData")
#Install the package psych
install.packages( "psych" , repos = "http://cran.rstudio.com/" )
```

```
## Installing package into '/Users/buraksonmez/Library/R/arm64/4.3/library'
## (as 'lib' is unspecified)
```

```
##
## The downloaded binary packages are in
##  /var/folders/f6/gb9b_pqd0yg5tl9h5791zmjr0000gp/T//RtmptS3Dbp/downloaded_packages
```

```r
install.packages( "psychTools" , repos = "http://cran.rstudio.com/" )
```

```
## Installing package into '/Users/buraksonmez/Library/R/arm64/4.3/library'
## (as 'lib' is unspecified)

##
## The downloaded binary packages are in
##  /var/folders/f6/gb9b_pqd0yg5tl9h5791zmjr0000gp/T//RtmptS3Dbp/downloaded_packages
```

```r
install.packages( "corrplot" , repos = "http://cran.rstudio.com/" )
```

```
## Installing package into '/Users/buraksonmez/Library/R/arm64/4.3/library'
## (as 'lib' is unspecified)

##
## The downloaded binary packages are in
##  /var/folders/f6/gb9b_pqd0yg5tl9h5791zmjr0000gp/T//RtmptS3Dbp/downloaded_packages
```

```r
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.3.3
```

```r
library(psychTools)
```

```
## Warning: package 'psychTools' was built under R version 4.3.3
```

```r
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 4.3.3

## corrplot 0.95 loaded
```

```r
## Subsetting some variables related to confidence in institutions.
confvars <- c("confinan","conbus", "conclerg", "coneduc",
"conpress", "contv", "conjudge", "consci",
"conlegis", "conarmy")
confdata <- gss[confvars]
lowerCor(confdata)
```

```
##          cnfnn conbs cnclr condc cnprs contv cnjdg consc cnlgs cnrmy
## confinan 1.00
## conbus   0.35  1.00
## conclerg 0.28  0.29  1.00
## coneduc  0.25  0.21  0.23  1.00
## conpress 0.15  0.15  0.11  0.21  1.00
## contv    0.26  0.22  0.14  0.24  0.37  1.00
## conjudge 0.22  0.25  0.13  0.26  0.22  0.20  1.00
## consci   0.07  0.15  0.02  0.13  0.13  0.10  0.31  1.00
## conlegis 0.33  0.24  0.21  0.29  0.21  0.22  0.34  0.13  1.00
## conarmy  0.31  0.22  0.20  0.20  0.12  0.15  0.20  0.17  0.21  1.00
```

```r
## Find out relatively stronger correlations.
round(cor(confdata, use='pairwise'), 2)
```

```
##          confinan conbus conclerg coneduc conpress contv conjudge consci
## confinan     1.00   0.35     0.28    0.25     0.15  0.26     0.22   0.07
## conbus       0.35   1.00     0.29    0.21     0.15  0.22     0.25   0.15
## conclerg     0.28   0.29     1.00    0.23     0.11  0.14     0.13   0.02
## coneduc      0.25   0.21     0.23    1.00     0.21  0.24     0.26   0.13
## conpress     0.15   0.15     0.11    0.21     1.00  0.37     0.22   0.13
## contv        0.26   0.22     0.14    0.24     0.37  1.00     0.20   0.10
## conjudge     0.22   0.25     0.13    0.26     0.22  0.20     1.00   0.31
## consci       0.07   0.15     0.02    0.13     0.13  0.10     0.31   1.00
## conlegis     0.33   0.24     0.21    0.29     0.21  0.22     0.34   0.13
## conarmy      0.31   0.22     0.20    0.20     0.12  0.15     0.20   0.17
##          conlegis conarmy
## confinan     0.33    0.31
## conbus       0.24    0.22
## conclerg     0.21    0.20
## coneduc      0.29    0.20
## conpress     0.21    0.12
## contv        0.22    0.15
## conjudge     0.34    0.20
## consci       0.13    0.17
## conlegis     1.00    0.21
## conarmy      0.21    1.00
```

```r
cor <- cor(confdata)

#At first glance, there appears to be some
#association among confidence in different institutions.

## Creating a variable combining 10 likert scale questions.
confdata$con.scale <- rowMeans(confdata, na.rm=FALSE)
range(confdata$con.scale, na.rm=T)
```

```
## [1] 1 3
```

```r
#Find coefficient alpha as an estimate of reliability. This may be done for a single scale. Interpret t
alpha(confdata)
```

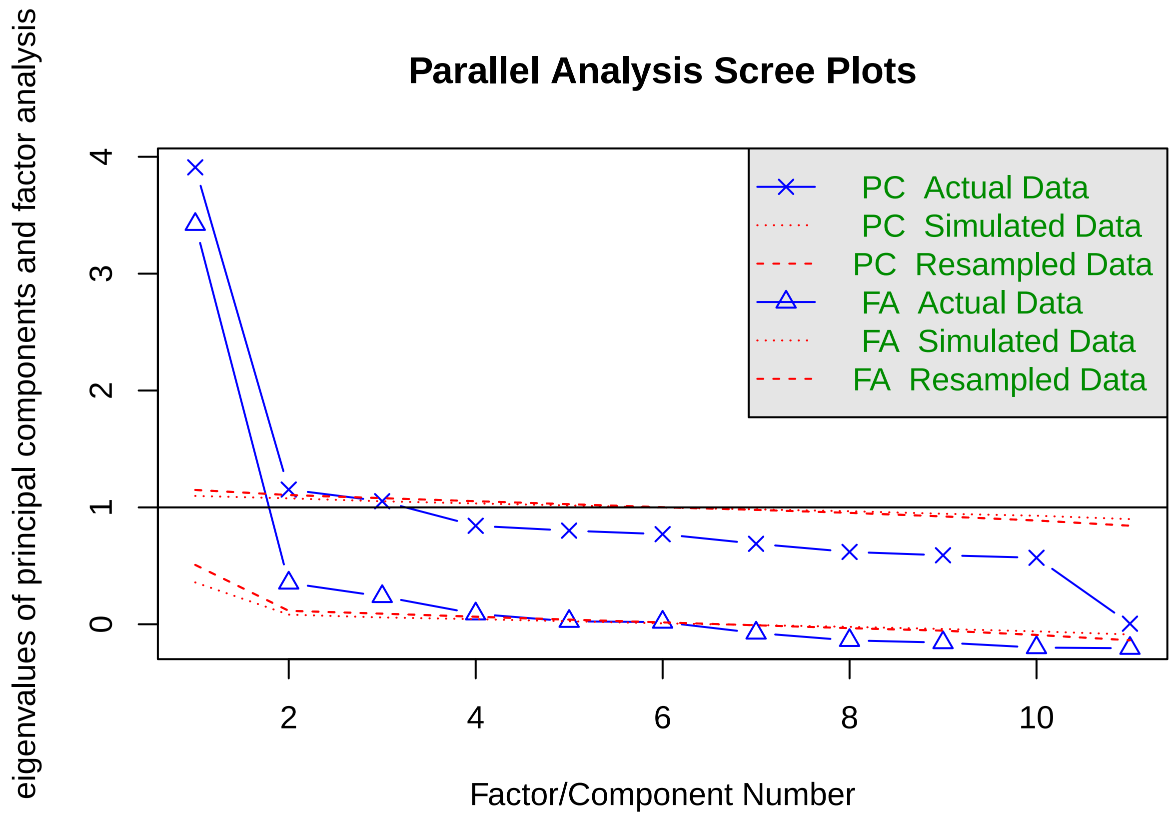
```
## Number of categories should be increased  in order to count frequencies.

##
## Reliability analysis
## Call: alpha(x = confdata)
##
##   raw_alpha std.alpha G6(smc) average_r S/N    ase mean   sd median_r
##       0.77       0.8    0.96      0.27   4 0.0063  2.1 0.35     0.22
##
##     95% confidence boundaries
##          lower alpha upper
```

```
## Feldt     0.76  0.77  0.79
## Duhachek  0.76  0.77  0.79
##
##  Reliability if an item is dropped:
##           raw_alpha std.alpha G6(smc) average_r S/N alpha se  var.r med.r
## confinan       0.75      0.78    0.89      0.26 3.6   0.0070 0.0223  0.22
## conbus         0.75      0.79    0.90      0.27 3.7   0.0069 0.0234  0.22
## conclerg       0.77      0.80    0.89      0.28 3.9   0.0065 0.0221  0.22
## coneduc        0.76      0.79    0.89      0.27 3.7   0.0068 0.0236  0.22
## conpress       0.77      0.79    0.89      0.28 3.9   0.0066 0.0225  0.24
## contv          0.76      0.79    0.89      0.27 3.7   0.0067 0.0228  0.22
## conjudge       0.75      0.78    0.88      0.27 3.6   0.0069 0.0230  0.22
## consci         0.78      0.80    0.90      0.29 4.1   0.0063 0.0208  0.24
## conlegis       0.75      0.78    0.89      0.27 3.6   0.0069 0.0230  0.22
## conarmy        0.76      0.79    0.90      0.28 3.8   0.0066 0.0235  0.24
## con.scale      0.72      0.72    0.72      0.21 2.6   0.0077 0.0058  0.21
##
##  Item statistics
##              n raw.r std.r r.cor r.drop mean   sd
## confinan  1946  0.60  0.60  0.59   0.47  2.2 0.65
## conbus    1926  0.57  0.57  0.56   0.44  2.0 0.59
## conclerg  1914  0.50  0.49  0.47   0.34  2.1 0.68
## coneduc   1948  0.57  0.56  0.55   0.43  1.9 0.66
## conpress  1937  0.50  0.50  0.48   0.35  2.4 0.64
## contv     1938  0.54  0.54  0.53   0.40  2.3 0.65
## conjudge  1915  0.59  0.58  0.58   0.45  1.9 0.67
## consci    1884  0.41  0.41  0.39   0.26  1.6 0.60
## conlegis  1925  0.59  0.59  0.58   0.47  2.5 0.61
## conarmy   1937  0.51  0.51  0.50   0.37  1.5 0.63
## con.scale 1792  1.00  1.00  1.02   0.99  2.1 0.34
```

```r
##Test for the number of factors in your data using parallel analysis
fa.parallel(confdata)
```

```
## 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 score estimation 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 =  4  and the number of components =  2
```

```r
#Let's start with principal component analysis (PCA).
#The default function consists of one component.
#However, you can change this by using the argument.
#Please see all the arguments for PCA, using help function.
#In using PCA, the goal can be only data reduction,
#but the interpretation of components is frequently done
#in terms similar to those used when describing
#the latent variables estimated by FA.
#PCA reports the largest n eigen vectors
#rescaled by the square root of their eigen values.
principal(confdata)
```

```
## Principal Components Analysis
## Call: principal(r = confdata)
## Standardized loadings (pattern matrix) based upon correlation matrix
##           PC1   h2     u2 com
## confinan 0.62 0.38 0.6195   1
## conbus   0.58 0.34 0.6625   1
## conclerg 0.48 0.23 0.7696   1
## coneduc  0.57 0.32 0.6765   1
## conpress 0.49 0.24 0.7644   1
## contv    0.54 0.29 0.7059   1
## conjudge 0.59 0.35 0.6518   1
```

```
## consci    0.37 0.14 0.8619   1
## conlegis  0.61 0.37 0.6289   1
## conarmy   0.51 0.26 0.7442   1
## con.scale 1.00 0.99 0.0053   1
##
##                 PC1
## SS loadings    3.91
## Proportion Var 0.36
##
## Mean item complexity =  1
## Test of the hypothesis that 1 component is sufficient.
##
## The root mean square of the residuals (RMSR) is  0.09
##  with the empirical chi square  2403.31  with prob <  0
##
## Fit based upon off diagonal values = 0.92
```

```
## Retaining two factors instead of one. Check the results and identify the differences.
principal(confdata, nfactors = 2)
```

```
## Principal Components Analysis
## Call: principal(r = confdata, nfactors = 2)
## Standardized loadings (pattern matrix) based upon correlation matrix
##             RC1   RC2   h2     u2 com
## confinan   0.72  0.10 0.53 0.4740 1.0
## conbus     0.61  0.18 0.40 0.5984 1.2
## conclerg   0.71 -0.09 0.51 0.4924 1.0
## coneduc    0.44  0.36 0.32 0.6764 1.9
## conpress   0.11  0.62 0.40 0.6008 1.1
## contv      0.29  0.50 0.33 0.6651 1.6
## conjudge   0.22  0.66 0.48 0.5222 1.2
## consci    -0.07  0.66 0.43 0.5667 1.0
## conlegis   0.47  0.38 0.37 0.6288 1.9
## conarmy    0.51  0.18 0.29 0.7083 1.2
## con.scale  0.75  0.66 1.00 0.0048 2.0
##
##                        RC1  RC2
## SS loadings           2.77 2.29
## Proportion Var        0.25 0.21
## Cumulative Var        0.25 0.46
## Proportion Explained  0.55 0.45
## Cumulative Proportion 0.55 1.00
##
## Mean item complexity =  1.4
## Test of the hypothesis that 2 components are sufficient.
##
## The root mean square of the residuals (RMSR) is  0.1
##  with the empirical chi square  2881.53  with prob <  0
##
## Fit based upon off diagonal values = 0.9
```

```
#The parallel factors technique compares
#the observed eigen values of a correlation matrix
```

```r
#with those from random data.
#We could fit a one-factor model to these data
#in which all ten indicators are thought to reflect a common
#latent factor. Here, we estimate this factor using maximum likelihood.
conf1 <- fa(confdata, nfactors=1, fm="ml")
conf1
```

```
## Factor Analysis using method =  ml
## Call: fa(r = confdata, nfactors = 1, fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            ML1   h2     u2 com
## confinan  0.60 0.36 0.6399   1
## conbus    0.56 0.32 0.6814   1
## conclerg  0.49 0.24 0.7641   1
## coneduc   0.56 0.32 0.6843   1
## conpress  0.49 0.24 0.7579   1
## contv     0.54 0.29 0.7063   1
## conjudge  0.59 0.35 0.6490   1
## consci    0.39 0.16 0.8442   1
## conlegis  0.59 0.35 0.6468   1
## conarmy   0.50 0.25 0.7487   1
## con.scale 1.00 1.00 0.0049   1
##
##                 ML1
## SS loadings    3.87
## Proportion Var 0.35
##
## Mean item complexity =  1
## Test of the hypothesis that 1 factor is sufficient.
##
## df null model =  55  with the objective function =  6.38 with Chi Square =  18270.06
## df of  the model are 44  and the objective function was  2.96
##
## 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 n.obs is  1885 with the empirical chi square  1535.68  with prob <  2.7e-293
## The total n.obs was  2867  with Likelihood Chi Square =  8454.79  with prob <  0
##
## Tucker Lewis Index of factoring reliability =  0.423
## RMSEA index =  0.258  and the 90 % confidence intervals are  0.254 0.263
## BIC =  8104.51
## Fit based upon off diagonal values = 0.92
## Measures of factor score adequacy
##                                                    ML1
## Correlation of (regression) scores with factors   1.00
## Multiple R square of scores with factors          1.00
## Minimum correlation of possible factor scores     0.99
```

```r
##We see that much of the variation in finance, business,judge,
##legislation, and education is explained by the latent factor.
##However, only 28% and 39% variance in science and press is explained
##with low communality and high uniquenesses. This suggests
```

```r
##a poor factor solution. Let's compare against a 3-factor solution:
confactor <- fa(confdata, nfactors=3, fm="ml", rotate="oblimin")
```

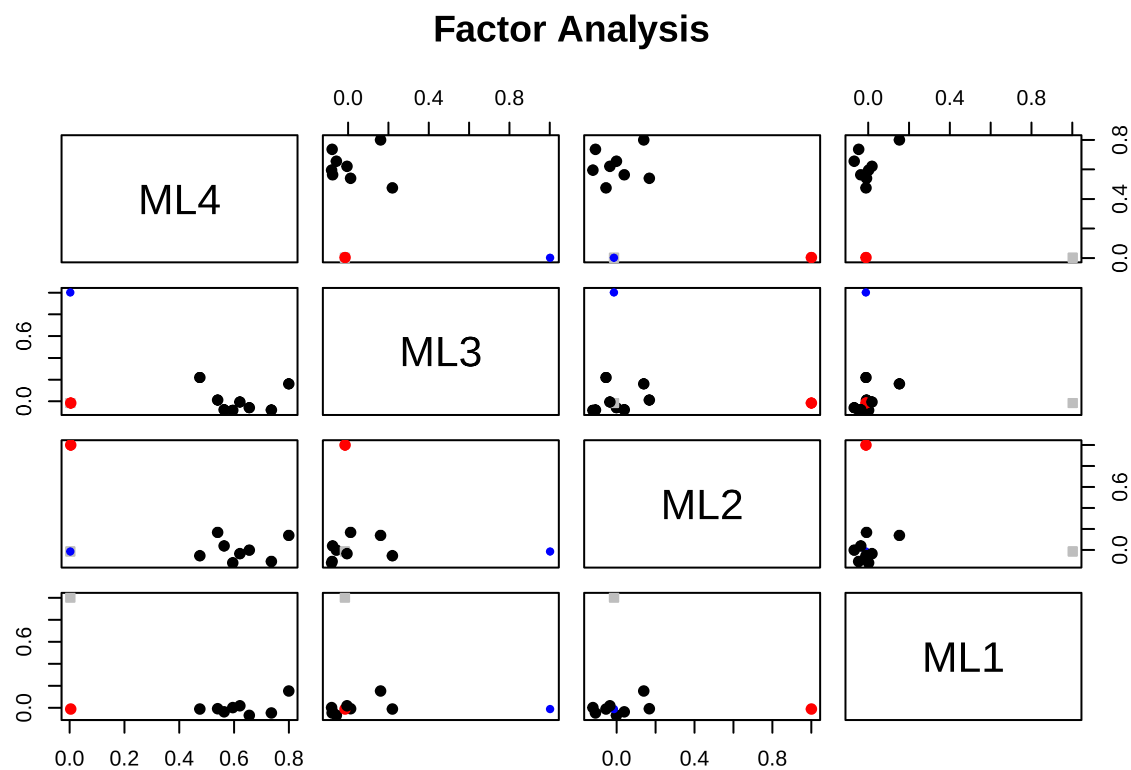
```
## Loading required namespace: GPArotation
```

```r
confactor
```

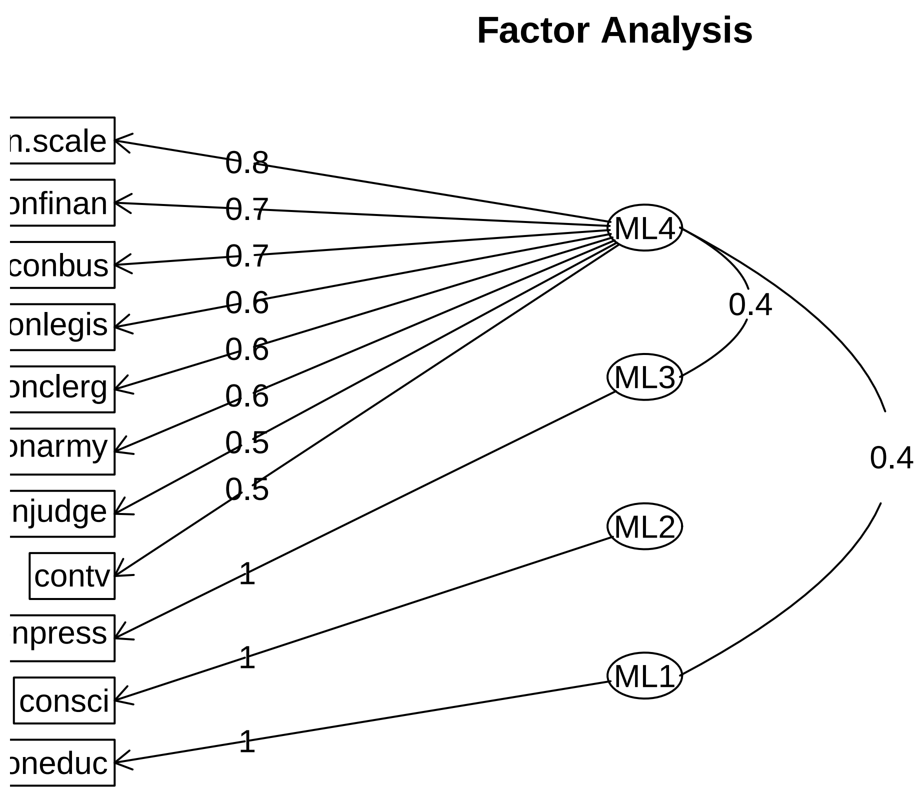
```
## Factor Analysis using method =  ml
## Call: fa(r = confdata, nfactors = 3, rotate = "oblimin", fm = "ml")
## Standardized loadings (pattern matrix) based upon correlation matrix
##            ML1   ML3   ML2   h2     u2 com
## confinan  0.71 -0.10 -0.12 0.43 0.5713 1.1
## conbus    0.62 -0.08 -0.01 0.35 0.6531 1.0
## conclerg  0.59 -0.10 -0.13 0.30 0.7021 1.2
## coneduc   0.59  0.00 -0.04 0.33 0.6672 1.0
## conpress  0.01  1.00 -0.01 1.00 0.0050 1.0
## contv     0.47  0.21 -0.06 0.32 0.6794 1.4
## conjudge  0.54  0.00  0.16 0.36 0.6373 1.2
## consci    0.01 -0.02  1.00 1.00 0.0050 1.0
## conlegis  0.63 -0.02 -0.04 0.38 0.6230 1.0
## conarmy   0.54 -0.09  0.03 0.28 0.7243 1.1
## con.scale 0.89  0.14  0.12 1.00 0.0049 1.1
##
##                        ML1  ML3  ML2
## SS loadings           3.58 1.08 1.07
## Proportion Var        0.33 0.10 0.10
## Cumulative Var        0.33 0.42 0.52
## Proportion Explained  0.62 0.19 0.19
## Cumulative Proportion 0.62 0.81 1.00
##
##  With factor correlations of
##      ML1  ML3  ML2
## ML1 1.00 0.37 0.28
## ML3 0.37 1.00 0.16
## ML2 0.28 0.16 1.00
##
## Mean item complexity =  1.1
## Test of the hypothesis that 3 factors are sufficient.
##
## df null model =  55  with the objective function =  6.38 with Chi Square =  18270.06
## df of  the model are 25  and the objective function was  2.55
##
## The root mean square of the residuals (RMSR) is  0.07
## The df corrected root mean square of the residuals is  0.1
##
## The harmonic n.obs is  1885 with the empirical chi square  960.32  with prob <  1.5e-186
## The total n.obs was  2867  with Likelihood Chi Square =  7304.54  with prob <  0
##
## Tucker Lewis Index of factoring reliability =  0.12
## RMSEA index =  0.319  and the 90 % confidence intervals are  0.313 0.325
## BIC =  7105.51
## Fit based upon off diagonal values = 0.95
## Measures of factor score adequacy
```

```
##                                                      ML1  ML3  ML2
## Correlation of (regression) scores with factors    1.00 1.00 1.00
## Multiple R square of scores with factors           0.99 0.99 0.99
## Minimum correlation of possible factor scores      0.99 0.99 0.99
```

```r
##The paralel analysis suggests a 4-factor solution.
confactor <- fa(confdata, nfactors=4, fm="ml", rotate="oblimin")
##fa.plot will plot the loading from a factor, principal components,
##or cluster analysis.If there are more than two factors, then a SPLOM
##of the loadings is generated.
fa.plot(confactor)
```

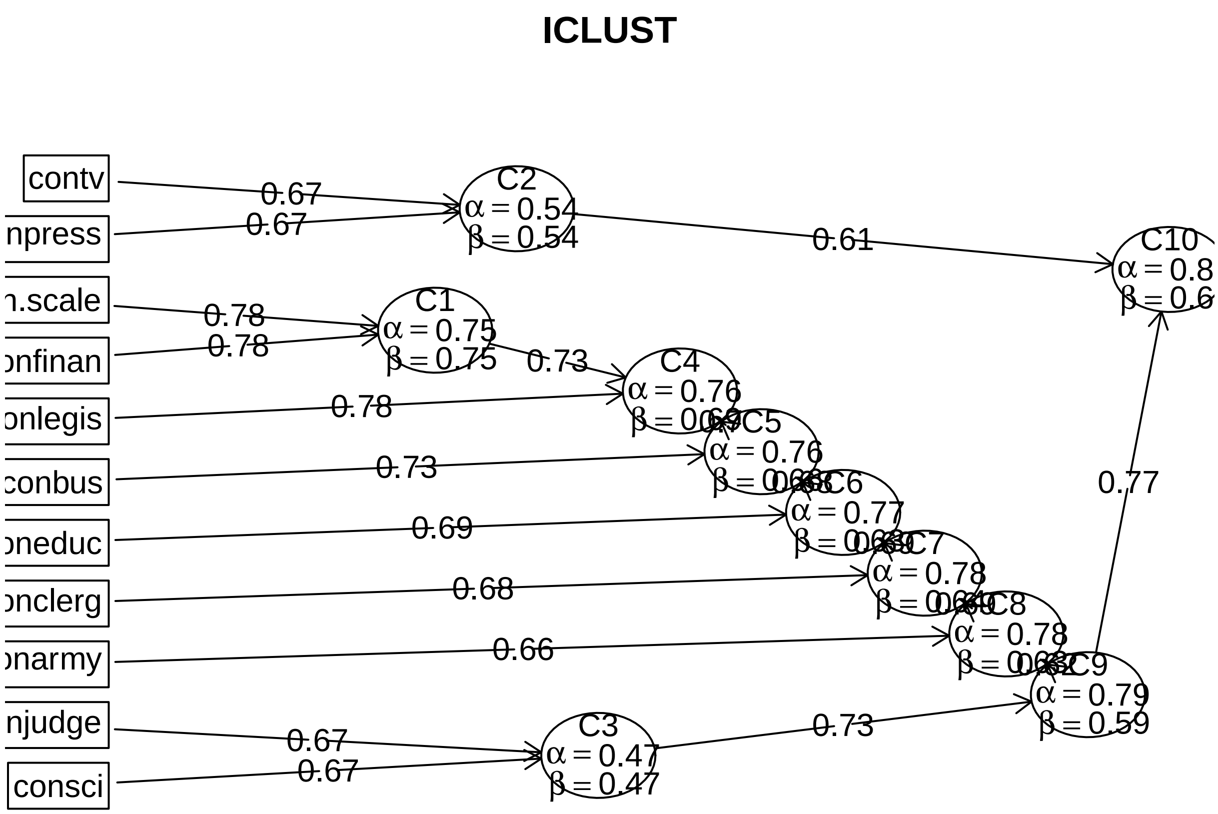



```r
##fa.diagram replaces fa.graph and will draw a path diagram representing
##the factor structure. It does not require Rgraphviz and
##thus is probably preferred.
fa.diagram(confactor)
```




```r
#Item Cluster Analysis
iclust(confdata)
```

```
## 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 score estimation method.
```




```
## ICLUST (Item Cluster Analysis)
## Call: iclust(r.mat = confdata)
##
## Purified Alpha:
## C10
## 0.8
##
## G6* reliability:
## C10
##   1
##
## Original Beta:
##  C10
## 0.64
##
## Cluster size:
## C10
##  11
##
## Item by Cluster Structure matrix:
##             O   P  C10
## confinan  C10 C10 0.59
## conbus    C10 C10 0.56
## conclerg  C10 C10 0.47
## coneduc   C10 C10 0.55
## conpress  C10 C10 0.48
```

```
## contv     C10 C10 0.53
## conjudge  C10 C10 0.58
## consci    C10 C10 0.39
## conlegis  C10 C10 0.58
## conarmy   C10 C10 0.50
## con.scale C10 C10 1.02
##
## With Sums of squares of:
## C10
## 3.8
##
## Purified scale intercorrelations
##  reliabilities on diagonal
##  correlations corrected for attenuation above diagonal:
##     C10
## C10 0.8
##
## Cluster fit =  0.72   Pattern fit =  0.97  RMSR =  0.08
```