Package ‘random.polychor.pa’

February 15, 2012

Type Package

Title A Parallel Analysis With Polychoric Correlation Matrices

Version 1.1.3.2

Date 2011-09-02

Author Fabio Presaghi & Marta Desimoni

Maintainer Fabio Presaghi <fabio.presaghi@uniroma1.it>

Description The Function performs a parallel analysis using simulated polychoric correlation matrices. The nth-percentile of the eigenvalues distribution obtained from both the randomly generated and the real data polychoric correlation matrices is returned. A plot comparing the two types of eigenvalues (real and simulated) will help determine the number of real eigenvalues that outperform random data. The function is based on the idea that if real data are non-normal and the polychoric correlation matrix is needed to perform a Factor Analysis, then the Parallel Analysis method used to choose a non-random number of factors should also be based on randomly generated polychoric correlation matrices and not on Pearson correlation matrices. Version 1.1.1, fixed a minor bug in the regarding the estimated time needed to complete the simulation. Also in this version, the function is now able to manage supplied data.matrix in which variables representing factors (i.e.,variables with ordered categories) are present and may cause an error when the Pearson correlation matrix is calculated.
Version 1.1.2 simply has updated the function that calculates the polycoric correlation matrix due to changes in the psych() package. Version 1.1.3 tackles two problems signalled by users: 1) the possibility to make available the results of simulation for plotting them in other softwares. Now the random.polychor.pa will show, upon request, all the data used in the scree-plot. 2) The function polichoric() of the psych() package does not handle data matrices that include 0 as possible category and will cause the function to stop with

error. So a check for the detection of the 0 code within the provided data.matrix is now added and will cause the random.polychor.pa function to stop with a warning message.

Depends psych, nFactors, mvtnorm, MASS, boot, lattice, polycor,sfsmisc

LazyLoad yes

LazyData yes

License GPL (>= 2)

Encoding latin1

Repository CRAN

Date/Publication 2011-12-30 11:23:09

R topics documented:

Parallel-Analysis-of-Polychoric-Correlations 2
random.polychor.pa 4

Index **10**

`Parallel-Analysis-of-Polychoric-Correlations`
A Parallel Analysis with Random Polychoric Correlation Matrices

---

Description

The function performs a parallel analysis using simulated polychoric correlation matrices. The function will extract the eigenvalues from each random generated polychoric correlation matrix and from the polychoric correlation matrix of real data. A plot comparing eigenvalues extracted from the specified real data with simulated data will help determine which of real eigenvalue outperform random data. A series of matrices comparing MAP vs PA-Polychoric vs PA-Pearson correlations methods, FA vs PCA solutions are finally presented.

Details

Package:	random.polychor.pa
Type:	Package
Version:	1.1.3.2
Date:	2011-09-02
License:	GPL Version 2 or later
LazyLoad:	yes

The function perform a parallel analysis (Horn, 1965) using randomly simulated polychoric correlations. Generates nrep random samples of the same dimension of the empirical provided data.matrix(i.e, with the same number of participants and of variables). Items are allowed to have varying number of categories. The function will extract the eigenvalues from each randomly generated polychoric matrices and the declared quantile will be extracted. Eigenvalues from polychoric correlation matrix obtained from real data are also computed and compared, in a (scree) plot, with the eigenvalues extracted from the simulation (Polychoric matrices). Recently Cho, Li & Bandalos (2009), showed that in using PA method, it is important to match the type of the correlation matrix used to recover the eigenvalues from real data with the type of correlation matrix used to estimate random eigenvalues. Crossing the type of correlations (using Polychoric correlation matrix to estimate real eigenvalues and random simulated Pearson correlation matrice) may result in a wrong decision (i.e., retaining less non-random factors than the needed). A comparison with eigenvalues extracted from both randomly simulated Pearson correlation matrices and real data is also also included. Finally, for both type of correlation matrix (Polychoric vs Pearson), the two versions (the classic squared coefficient and the 4th power coefficient) of Velicer's MAP criterion are calculated. In version 1.1.1, a minor bug in the regarding the estimated time needed to complete the simulation is fixed. Also in this version, the function is now able to manage supplied data.matrix in which variables representing factors (i.e., variables with ordered categories) are present and may cause an error when the Pearson correlation matrix is calculated. As the poly.mat() function used to calculate the polychoric correlation matrix is going to be deprecated in favour of polychoric() function, the random.polychor.pa was consequently updated (version 1.1.2) to account for changes in psych() package. Version 1.1.3 tackles two problems signalled by users: 1) the possibility to make available the results of simulation for plotting them in other softwares. Now the random.polychor.pa will show, upon request, all the data used in the scree-plot. 2) The function polichoric() of the psych() package does not handle data matrices that include 0 as possible category and will cause the function to stop with error. So a check for the detection of the 0 code within the provided data.matrix is now added and will cause the random.polychor.pa function to stop with a warning message.

Author(s)

Fabio Presaghi <fabio.presaghi@uniroma1.it> and Marta Desimoni <marta.desimoni@uniroma1.it>

References

Cho, S.J., Li, F., & Bandalos, D., (2009). Accuracy of the Parallel Analysis Procedure With Polychoric Correlations. Educational and Psychological Measurement, 69, 748-759.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. Psychometrika, 32, 179-185.

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. Behavior Research Methods, Instrumentation, and Computers, 32, 396-402.

Velicer, W. F. (1976). Determining the number of factors from the matrix of partial correlations. Psychometrika, 41, 321-327.

Velicer, W. F., Eaton, C. A., & Fava, J. L. (2000). Construct explication through factor or component analysis: A review and evaluation of alternative procedures for determining the number of factors or components. In R. D. Goffin & E. Helmes (Eds.), Problems and solutions in human assessment: Honoring Douglas N. Jackson at seventy (pp. 41-72). Norwell, MA: Kluwer Academic.

`random.polychor.pa` *A Parallel Analysis With Randomly Generated Polychoric Correlation Matrices*

Description

The function performs a parallel analysis using simulated polychoric correlation matrices. The eigenvalues (extracted following both FA and PCA methods) from each random generated polychoric correlation matrix and from the polychoric correlation matrix of real data are extraced. A plot comparing eigenvalues extracted from the specified real data and simulated data will help determine which of real eigenvalue outperform random data. Several matrices comparing solutions from Polychorich vs Pearson correlations, FA vs PCA and PA vs MAP are presented.

Usage

```
random.polychor.pa(nvar="NULL", n.ss="NULL", nrep, nstep="NULL",
                         data.matrix, q.eigen, r.seed = "NULL")
```

Arguments

`nvar`	Number of variables (items) in the raw data matrix. From version 1.1 of the function, it is no more needed to specificy nvar as this information is derived from the number of columns of the data.matrix. Default value is set to "NULL" for compatibility with past version of the function.
`n.ss`	Number of participants of the raw data matrix. From version 1.1 of the function, it is no more needed to specify n.ss as this information is derived from the number of rows of the data.matrix. Default value is set to "NULL" for compatibility with past version of the function.
`nrep`	Number of random samples that should be simulated
`nstep`	Number of ordered categories of the item (e.g., Likert-like 3 ordered category item). This information is no longer needed as the new version of the function (1.1) allows also for items with varying number of categories. The number of categories from each item is derived directly from the data.matrix. A table summarizing the different groups of item with different number of categories will be showed. Default value is set to "NULL" for compatibility with past version of the function.
`data.matrix`	the name of raw data matrix. The raw data.matrix should be numeric and none of the ordered category should be coded as 0 (zero). No automatic recode routine is provided within the function to deal with alphanumeric content of the ordered categories of manifest variables. So the user performs all these recodings before running the function.
`q.eigen`	a number comprised within the interval of 0 and 1 and indicating the quantile that is used to choose the number of non-random factors (e.g., .50 or .95 or .99)
`r.seed`	eventually, a preferred number that will be used to initialize the random generator. Default value: 1335031435.

Details

The function perform a parallel analysis (Horn, 1976) using randomly simulated polychoric correlations. Generates nrep random samples of simulated data with the same number of participants and of variables of the provided data.matrix. The function will read the entered data.matrix and extracts: the number of units (i.e., number of rows); the number of variables (i.e., number of columns); and the number of categories of each item. From version 1.1, the function accepts also variables with varying number of categories (e.g., three items with only two categories and two items with three categories, etc.). In version 1.1.1, the function is also able to manage supplied data.matrix in which variables represent factors (i.e., variables with ordered categories) may cause an error when the Pearson correlation matrix is calculated. The information in the supplied data.matrix are used to generate the nrep random raw datasets with the same characteristics of the original real data set. So only three information are needed for the problem to run: the number of replications (nrep), the data matrix (data.matrix) and the percentile to be used (q.eigen). A check for missing values within the real dataset is present and if present will be treated LISTWISE. In this case a warning message will prompt the user signalling how NA were treated (LISTWISE is by now the only treatment considered) and the new sample size. No further checks are made on the raw data, so out-of-range values are not detected and it is on the behalf of the user to make a preliminary check on the reliability of data. A table summarizing the groups of items with different number of categories will be shown along with the main results of the PA. The function will extract the eigenvalues from each randomly generated polychoric matrices and the requested percentile is returned. Eigenvalues from polychoric correlation matrix obtained from real data is also compute and compared, in a (scree) plot, with the eigenvalues extracted from the simulation (Polychoric matrices). Recently, Cho, Li & Bandalos (2009) showed that, in using PA method, it is important to match the type of the correlation matrix used to recover the eigenvalues from real data with the type of correlation matrix used to estimate random eigenvalues. Crossing the type of correlations (using Polychoric correlation matrix to estimate real eigenvalues and random simulated Pearson correlation matrices) may result in a wrong decision (i.e., retaining more non-random factors than the needed). A comparison with eigenvalues extracted from both randomly simulated Pearson correlation matrices and real data is also included. Finally, for both type of correlation matrix (Polychoric vs Pearson), the two versions (the classic squared coefficient and the 4th power coefficient) of Velicer's MAP criterion are calculated (Velicer, 1976; Velicer, Eaton, & Fava, 2000) by implementing under R the code released by O'Connor (2000) for SPSS, SAS and MATLAB. As the poly.mat() function used to calculate the polychoric correlation matrix is going to be deprecated in favour of polychoric() function, the random.polychor.pa was consequently updated (version 1.1.2) to account for changes in psych() package.

Value

The function returns the number of factors for Polychoric and Pearson Correlation PA methods for Factor Analysis and Principal Components Analysis (PCA) methods along with the number of factors chosen by the two Velicer's MAP criteria (original and 4th power) for both Polychoric and Pearson correlation matrices. Furthermore, the function will return the (scree) plot of the eigenvalues for real (Polychoric vs Pearson correlation matrices) and simulated data (Polychoric vs Pearson correlation matrices). Finally the following LIST of matrices will be printed:

`$MAP.selection` Returns a matrix with five columns (variables) and with as many rows as the number of selected factors (by the Velicer's MAP method) plus 1: Factor (i.e., the number of factors); POLY.MAP.squared (classic, squared MAP coefficient calcutated on the polychoric correlation matrix); POLY.MAP.4th (the modern,

4th power, of the MAP coefficient calculated on the polychoric correlation matrix); CORR.MAP.squared (classic, squared MAP coefficient calcutated on the Pearson correlation matrix); CORR.MAP.4th (the modern, 4th power, of the MAP coefficient calculated on the Pearson correlation matrix)

`$POLYCHORIC` Returns a matrix with five columns (variables) and as many rows as the number of selected factors (by the Polychoric PA method) plus 1: Factors (number of factors); Emp.Polyc.Eigen (eigenvalues extracted from the empirical polychoric correlation matrix through the corFA function of nFactors package, i.e. by substituting the item communalities along the main diagonal of the correlation matrix); P.SimMeanEigen (the average n-th eigenvalue, extracted from Polycoric correlation matrix, of the nrep simulated random samples); P.SimSDEigen (the standard deviation for the n-th eigenvalue, extracted from Polycoric correlation matrix, of the nrep simulated random samples); P.SimQuant (the q.eigen*100 Percentile of the distribution of eigenvalues, extracted from the Polychoric correlation matrix, of the nrep simulated random samples)

`$PEARSON` Returns a matrix with five columns (variables) and as many rows as the number of selected factors (by the Pearson correlation PA method) plus 1: Factors (number of factors); Emp.Pears.Eigen (eigenvalues extracted from the empirical Pearson correlation matrix through the corFA function of nFactors package, i.e. by substituting the item communalities along the main diagonal of the correlation matrix); C.SimMeanEigen (the average n-th eigenvalue, extracted from Pearson correlation matrix, of the nrep simulated random samples); C.SimSDEigen (the standard deviation for the n-th eigenvalue, extracted from Pearson correlation matrix, of the nrep simulated random samples); C.SimQuant (the q.eigen*100 Quantile eigenvalue, extracted from the Pearson correlation matrix, of the nrep simulated random samples)

`$POLYCHORIC.PCA`

Returns a matrix with five columns (variables) and as many rows as the number of selected factors (by the Polychoric PA method) plus 1: Factors (number of factors); Emp.Polyc.Eigen.PCA (eigenvalues extracted from the empirical polychoric correlation matrix through the Principal Components Analysis); P.SimMeanEigen.PCA (the average n-th eigenvalue (PCA method), extracted from Polycoric correlation matrix, of the nrep simulated random samples); P.SimSDEigen (the standard deviation for the n-th eigenvalue (PCA method), extracted from Polycoric correlation matrix, of the nrep simulated random samples); P.SimQuant (the q.eigen*100 Percentile of the distribution of eigenvalues (PCA method), extracted from the Polychoric correlation matrix, of the nrep simulated random samples)

`$PEARSON.PCA` Returns a matrix with five columns (variables) and as many rows as the number of selected factors (by the Pearson correlation PA method) plus 1: Factors (number of factors); Emp.Pears.Eigen (eigenvalues extracted from the empirical Pearson correlation matrix through the PCA method); C.SimMeanEigen (the average n-th eigenvalue (PCA method), extracted from Pearson correlation matrix, of the nrep simulated random samples); C.SimSDEigen (the standard deviation for the n-th eigenvalue (PCA method), extracted from Pearson correlation matrix, of the nrep simulated random samples); C.SimQuant (the q.eigen*100 Quantile eigenvalue (PCA method), extracted from the Pearson correlation matrix, of the nrep simulated random samples)

Note

In running the random.polychor.pa function it should be reminded that it may take a lot of time to complete the simulation. This is due in part to the fact that the estimation of the polychoric correlation matrix is cumbersome and in part to the fact that the code is not optimized, but simply it does the work.

Occasionally, in calculating the polychoric correlation matrix it may occur an error when the matrix is non-positive definite. In this case you have to re-run the simulation.

A note should be made concerning the method used (from version 1.1) to read the raw data.matrix supplied by the user and used to retrieve the three basic information needed to build the random matrices (number of rows, number of columns and the number of categories for each manifest variable). The number of categories for each variable is derived from the raw data.matrix, so if the possible number of categories for a specific item is for example 5, but subjects endorse only three out of the five categories then the random.polychor.pa function will simulate a variable with only three categories. This means that the function guarantees that the empirical and the simulated data matrix are similar, but this also means that by changing the sample of participants the simulated data will change (even if slightly).

Author(s)

Fabio Presaghi <fabio.presaghi@uniroma1.it> and Marta Desimoni <marta.desimoni@uniroma1.it>

References

Cho, S.J., Li, F., & Bandalos, D., (2009). Accuracy of the Parallel Analysis Procedure With Polychoric Correlations. Educational and Psychological Measurement, 69, 748-759.

Horn, J. L. (1965). A rationale and test for the number of factors in factor analysis. Psychometrika, 32, 179-185.

O'Connor, B. P. (2000). SPSS and SAS programs for determining the number of components using parallel analysis and Velicer's MAP test. Behavior Research Methods, Instrumentation, and Computers, 32, 396-402.

Velicer, W. F. (1976). Determining the number of factors from the matrix of partial correlations. Psychometrika, 41, 321-327.

Velicer, W. F., Eaton, C. A., & Fava, J. L. (2000). Construct explication through factor or component analysis: A review and evaluation of alternative procedures for determining the number of factors or components. In R. D. Goffin & E. Helmes (Eds.), Problems and solutions in human assessment: Honoring Douglas N. Jackson at seventy (pp. 41-72). Norwell, MA: Kluwer Academic.

See Also

nFactors, psych, paran.

Examples

```
### EXAMPLE 1:
### example data
raw.data<-data.frame(ss=1:20, v1=c(1,5,2,1,4,3,2,1,2,5,1,5,2,4,2,2,2,5,4,3),
v2=c(5,3,3,2,3,1,1,2,3,5,2,5,5,4,4,5,3,4,2,1),
```

```
v3=c(2,4,2,3,3,2,1,3,2,4,1,2,2,2,4,4,5,1,2,1),
v4=c(3,1,3,2,5,2,3,5,2,3,5,5,5,4,3,3,2,3,3,1),
v5=c(3,1,4,5,3,4,3,4,2,5,1,2,1,2,1,4,2,2,4,3))

raw.item.data <- (raw.data[,2:6])
summary (raw.item.data)
cor(raw.item.data)
eigen(cor(raw.item.data))

random.polychor.pa(nrep=5, data.matrix=raw.item.data, q.eigen=.99)

### EXAMPLE 2a:
### this example is particularly instructive on how the solution may
### change by changing the type of correlation, method of extraction and
### method of selection.
### Before launching the example consider that the
### ESTIMATED TIME TO COMPLETE THE SIMULATION IS ABOUT: 10 MIN.
require(psych)
data(bfi)
raw.data<-as.matrix(bfi)
raw.data <- (raw.data[1:100,2:6])
test.1<-random.polychor.pa(nrep=3, data.matrix=raw.data, q.eigen=.99)
test.1

### EXAMPLE 2b:
### in this example one of the categories of item1 is recoded: 2=1
### so this item has 5 categories: 1 (2) 3 4 5 6
### category 1 is within brackets as it has frequency=0
### so this is a case where empirical data (0 2 3 4 5 6) diverge from
### theorethical data (0 1 2 3 4 5 6)
require(psych)
data(bfi)
raw.data.1<-as.matrix(bfi)
raw.data.1 <- (raw.data.1[1:100,1:25])
for(i in 1:nrow(raw.data.1)) { if(raw.data.1[i,1]==2) raw.data.1[i,1]<-1}
test.2<-random.polychor.pa(nrep=3, data.matrix=raw.data.1, q.eigen=.99)
test.2

### EXAMPLE 2c:
### in this example one of the categories of item1 is recoded: 1=0
### so this item has one of its categories coded as 0
### this will cause polychoric() function to stop with error
### and the random.polychoric.pa will prompt a warning message
#require(psych)
#data(bfi)
#raw.data.2<-as.matrix(bfi)
#raw.data.2 <- (raw.data.2[1:100,1:25])
#for(i in 1:nrow(raw.data.2)) { if(raw.data.2[i,1]==1) raw.data.2[i,1]<-0} # recode 1=0
# random.polychor.pa(nrep=3, data.matrix=raw.data.2, q.eigen=.99)

### EXAMPLE 3:
######## for SPSS users ####
### the following instructions can used to load a SPSS data file (.sav).
```

```
### 1) load the library to read external datafile (e.g., SPSS datafile)
### 2) choose the SPSS datafile by pointing directly in the folder on your hard-disk
### 3) select only the variables (i.e., the items) needed to for Parallel Analysis
# library(foreign) ### load the needed library
# raw.data <- read.spss(choose.files(), use.value.labels=TRUE,
#                       max.value.labels=Inf, to.data.frame=TRUE)
# raw.spss.item <- na.exclude(raw.data[,2:4])
# summary (raw.spss.item)
# random.polychor.pa(nrep=5, data.matrix=raw.spss.item, q.eigen=.99)
```

Index

∗Topic **EXPLORATORY FACTOR ANALYSIS**
Parallel-Analysis-of-Polychoric-Correlations, 2
random.polychor.pa, 4
∗Topic **PARALLEL ANALISYS**
random.polychor.pa, 4
∗Topic **PARALLEL ANALYSIS**
Parallel-Analysis-of-Polychoric-Correlations, 2
∗Topic **POLYCHORIC CORRELATION MATRIX**
Parallel-Analysis-of-Polychoric-Correlations, 2
∗Topic **POLYCHORIC CORRELATION**
random.polychor.pa, 4

Parallel-Analysis-of-Polychoric-Correlations, 2

random.polychor.pa, 4
random.polychor.pa-package *(*Parallel-Analysis-of-Polychoric-Correlations*)*, 2