Package 'RGenData'

November 14, 2018

Type Package

Title Generates Multivariate Nonnormal Data and Determines How Many Factors to Retain

Version 1.0

Author John Ruscio

Maintainer John Ruscio <ruscio@tcnj.edu>

Description The GenDataSample() and GenDataPopulation() functions create, respectively, a sample or population of multivariate nonnormal data using methods described in Ruscio and Kaczetow (2008). Both of these functions call a FactorAnalysis() function to reproduce a correlation matrix. The EFACompData() function allows users to determine how many factors to retain in an exploratory factor analysis of an empirical data set using a method described in Ruscio and Roche (2012). The latter function uses populations of comparison data created by calling the GenDataPopulation() function.
<DOI: 10.1080/00273170802285693>.
<DOI: 10.1037/a0025697>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2018-11-14 15:00:09 UTC

R topics documented:

EFACompData 2
FactorAnalysis 3
GenDataPopulation 4
GenDataSample 5

Index **7**

EFACompData *EFACompData*

Description

Comparison data

Usage

```
EFACompData(data, f.max, n.pop = 10000, n.samples = 500, alpha = .30, graph = FALSE,
corr.type = "pearson")
```

Arguments

data	Matrix to store the simulated data (matrix).
f.max	Largest number of factors to consider (scalar).
n.pop	Size of finite populations of comparison data (scalar, default is 10,000 cases).
n.samples	Number of samples drawn from each population (scalar, default is 500).
alpha	Alpha level when testing statistical significance of improvement with additional factor (scalar, default is .30)
graph	Whether to plot the fit of eigenvalues to those for comparison data (default is FALSE).
corr.type	Type of correlation (character, default is "pearson", user can also call "spearman").

Value

Nothing, displays number of factors on screen.

Author(s)

John Ruscio

References

Ruscio & Roche (2011)

Examples

```
# create data matrix x with n = 200 cases, k = 9 variables
# 3 variables load onto each of 3 orthogonal factors
# all marginal distributions are highly skewed
x <- matrix(nrow = 200, ncol = 9)
for (i in 1:3) {
  shared <- rchisq(200, 1)
  for (j in 1:3) {
    x[, (i - 1) * 3 + j] <- shared + rchisq(200, 1)
```

```
  }
}
# empirically determine number of factors in data matrix x
EFACompData(x, f.max = 5)
```

FactorAnalysis *FactorAnalysis*

Description

Analyzes comparison data with known factorial structures

Usage

```
FactorAnalysis(data, corr.matrix = FALSE, max.iteration = 50,n.factors = 0,
corr.type = "pearson")
```

Arguments

data	Matrix to store the simulated data (matrix).
corr.matrix	Correlation matrix (default is FALSE)
max.iteration	Maximum number of iterations (scalar, default is 50).
n.factors	Number of factors (scalar, default is 0).
corr.type	Type of correlation (character, default is "pearson", user can also call "spearman").

Value

$loadings	Factor loadings (vector, if one factor. matrix, if multiple factors)
$factors	Number of factors (scalar).

Author(s)

John Ruscio

References

Ruscio & Roche (2011)

Examples

```
# create data matrix x with n = 200 cases, k = 9 variables
# 3 variables load onto each of 3 orthogonal factors
# all marginal distributions are highly skewed
x <- matrix(nrow = 200, ncol = 9)
for (i in 1:3) {
  shared <- rchisq(200, 1)
  for (j in 1:3) {
    x[, (i - 1) * 3 + j] <- shared + rchisq(200, 1)
  }
}
# perform factor analysis of data matrix x
FactorAnalysis(x)
```

`GenDataPopulation` *GenDataPopulation*

Description

Simulates multivariate nonnormal data using an iterative algorithm

Usage

```
GenDataPopulation(supplied.data, n.factors, n.cases, max.trials = 5,
                  initial.multiplier = 1, corr.type = "pearson", seed = 0)
```

Arguments

`supplied.data`	Data supplied by user.
`n.factors`	Number of factors (scalar).
`n.cases`	Number of cases (scalar).
`max.trials`	Maximum number of trials (scalar, default is 5).
`initial.multiplier`	Value of initial multiplier (scalar, default is 1).
`corr.type`	Type of correlation (character, default is "pearson", user can also call "spearman").
`seed`	seed value (scalar, default is 0).

Value

dataPopulation of data

Author(s)

John Ruscio

References

Ruscio & Roche (2011)

Examples

```
# create data matrix x with n = 200 cases, k = 9 variables
# 3 variables load onto each of 3 orthogonal factors
# all marginal distributions are highly skewed
x <- matrix(nrow = 200, ncol = 9)
for (i in 1:3) {
  shared <- rchisq(200, 1)
  for (j in 1:3) {
    x[, (i - 1) * 3 + j] <- shared + rchisq(200, 1)
  }
}
# generate (finite) population of data reproducing distributions and correlations in x
GenDataPopulation(x, n.factors = 3, n.cases = 10000)
```

`GenDataSample` *GenDataSample*

Description

Bootstraps each variable's score distribution from a supplied data set.

Usage

```
GenDataSample(supplied.data, n.factors = 0, max.trials = 5, initial.multiplier = 1,
corr.type = "pearson", seed = 0)
```

Arguments

`supplied.data`	Data supplied by user.
`n.factors`	Number of factors (scalar, default is 0).
`max.trials`	Maximum number of trials (scalar, default is 5).
`initial.multiplier`	Value of initial multiplier (scalar, default is 1).
`corr.type`	Type of correlation (character, default is "pearson", user can also call "spearman").
`seed`	seed value (scalar, default is 0).

Value

dataSample of data

Author(s)

John Ruscio

References

Ruscio & Kaczetow (2008)

Examples

```
# create data matrix x with n = 200 cases, k = 9 variables
# 3 variables load onto each of 3 orthogonal factors
# all marginal distributions are highly skewed
x <- matrix(nrow = 200, ncol = 9)
for (i in 1:3) {
  shared <- rchisq(200, 1)
  for (j in 1:3) {
    x[, (i - 1) * 3 + j] <- shared + rchisq(200, 1)
  }
}
# generate sample of data reproducing distributions and correlations in x
GenDataSample(x)
```

Index

EFACompData, 2

FactorAnalysis, 3

GenDataPopulation, 4
GenDataSample, 5