Package 'latentFactoR'

September 9, 2022

Title Data Simulation Based on Latent Factors

Version 0.0.2

Date 2022-09-08

Maintainer Alexander Christensen <alexpaulchristensen@gmail.com>

Description Generates data based on latent factor models. Data can be continuous, polytomous, dichotomous, or mixed. Skews, cross-loadings, wording effects, population errors, and local dependencies can be added. All parameters can be manipulated. Data categorization is based on Garrido, Abad, and Ponsoda (2011) <doi:10.1177/0013164410389489>.

Depends R (>= 3.6.0)

License GPL (>= 3.0)

Imports methods, mvtnorm, psych, rstudioapi

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation no

Author Alexander Christensen [aut, cre]
(<https://orcid.org/0000-0002-9798-7037>),
Maria Dolores Nieto Canaveras [aut],
Hudson Golino [aut] (<https://orcid.org/0000-0002-1601-1447>),
Luis Eduardo Garrido [aut] (<https://orcid.org/0000-0001-8932-6063>)

Repository CRAN

Date/Publication 2022-09-09 07:32:55 UTC

R topics documented:

latentFactoR-package *latentFactoR–package*

Description

Generates data based on latent factor models. Data can be continuous, polytomous, dichotomous, or mixed. Skew, cross-loadings, and population error can be added. All parameters can be manipulated. Data categorization is based on Garrido, Abad, and Ponsoda (2011).

Author(s)

Alexander P. Christensen <alexpaulchristensen@gmail.com>, Maria Dolores Nieto Canaveras <mnietoca@nebrija.es>, Hudson Golino <hfg9s@virginia.edu>, Luis Eduardo Garrido <luisgarrido@pucmm.edu>

References

Christensen, A. P., Garrido, L. E., & Golino, H. (2022).
Unique variable analysis: A network psychometrics method to detect local dependence.
PsyArXiv

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2011).
Performance of Velicer's minimum average partial factor retention method with categorical variables.
Educational and Psychological Measurement, *71*(3), 551-570.

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., ... & Martinez-Molina, A. (2020). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*(3), 292-320.

add_local_dependence *Adds Local Dependence to Factor Model Data*

Description

Adds local dependence to simulated data from `simulate_factors`. See examples to get started

Usage

```
add_local_dependence(
  lf_object,
  method = c("correlate_residuals", "minor_factors", "threshold_shifts"),
  proportion_LD,
  proportion_LD_range = NULL,
  add_residuals = NULL,
  add_residuals_range = NULL,
  allow_multiple = FALSE
)
```

Arguments

`lf_object`	Data object from `simulate_factors`
`method`	Character (length = 1). Method to generate local dependence between variables. Only `"correlate_residuals"` at the moment. Future developments will include minor factor and threshold-shift methods. Description of methods: • `"correlate_residuals"` Adds residuals directly to the population correlation matrix prior to data generation (uses population correlation matrix from `simulate_factors`) • `"minor_factors"` Coming soon... • `"threshold_shifts"` Coming soon...
`proportion_LD`	Numeric (length = 1 or `factors`). Proportion of variables that should be locally dependent across all or each factor. Accepts number of locally dependent values as well
`proportion_LD_range`	Numeric (length = 2). Range of proportion of variables that are randomly selected from a random uniform distribution. Accepts number of locally dependent values as well
`add_residuals`	Numeric (length = 1, `factors`, or total number of locally dependent variables). Amount of residual to add to the population correlation matrix between two variables. Only used when `method = "correlated_residuals"`. Magnitudes are drawn from a random uniform distribution using +/- 0.05 of value input. Can also be specified directly (same length as total number of locally dependent variables). General effect sizes range from small (0.20), moderate (0.30), to large (0.40)
`add_residuals_range`	Numeric (length = 2). Range of the residuals to add to the correlation matrix are randomly selected from a random uniform distribution
`allow_multiple`	Boolean. Whether a variable should be allowed to be locally dependent with more than one other variable. Defaults to `FALSE`. Set to `TRUE` for more complex locally dependence patterns

Value

Returns a list containing:

`correlated_residuals`	A data frame with the first two columns specifying the variables that are locally dependent and the third column specifying the magnitude of the added residual for each locally dependent pair
`data`	Simulated data from the specified factor model
`population_correlation`	Population correlation matrix with local dependence added
`original_correlation`	Original population correlation matrix *before* local dependence was added
`parameters`	A list containing the parameters used to generate the data:

- `factors` Number of factors
- `variables` Variables on each factor
- `loadings` Loading matrix
- `factor_correlations` Correlations between factors
- `categories` Categories for each variable
- `skew` Skew for each variable

Author(s)

Alexander P. Christensen <alexpaulchristensen@gmail.com>, Hudson Golino <hfg9s@virginia.edu>, Luis Eduardo Garrido <luisgarrido@pucmm.edu>

References

Christensen, A. P., Garrido, L. E., & Golino, H. (2022).
Unique variable analysis: A network psychometrics method to detect local dependence.
PsyArXiv

Examples

```
# Generate factor data
two_factor <- simulate_factors(
  factors = 2, # factors = 2
  variables = 6, # variables per factor = 6
  loadings = 0.55, # loadings between = 0.45 to 0.65
  cross_loadings = 0.05, # cross-loadings N(0, 0.05)
  correlations = 0.30, # correlation between factors = 0.30
  sample_size = 1000 # number of cases = 1000
)

# Add local dependence
two_factor_LD <- add_local_dependence(
  lf_object = two_factor,
  proportion_LD = 0.25,
  add_residuals = 0.20,
  allow_multiple = FALSE
)

# Randomly vary proportions
two_factor_LD <- add_local_dependence(
  lf_object = two_factor,
  proportion_LD_range = c(0.10, 0.50),
  add_residuals = 0.20,
  allow_multiple = FALSE
)

# Randomly vary residuals
two_factor_LD <- add_local_dependence(
  lf_object = two_factor,
  proportion_LD = 0.25,
  add_residuals_range = c(0.20, 0.40),
```

```
  allow_multiple = FALSE
)

# Randomly vary proportions, residuals, and allow multiple
two_factor_LD <- add_local_dependence(
  lf_object = two_factor,
  proportion_LD_range = c(0.10, 0.50),
  add_residuals_range = c(0.20, 0.40),
  allow_multiple = TRUE
)
```

`categorize` *Categorize Continuous Data*

Description

Categorizes continuous data based on Garrido, Abad and Ponsoda (2011; see references). Categorical data with 2 to 6 categories can include skew between -2 to 2 in increments of 0.05

Usage

```
categorize(data, categories, skew_value = 0)
```

Arguments

`data`	Numeric (length = n). A vector of continuous data with *n* values. For matrices, use `apply`
`categories`	Numeric (length = 1). Number of categories to create. Between 2 and 6 categories can be used with skew
`skew_value`	Numeric (length = 1). Value of skew. Ranges between -2 to 2 in increments of 0.05. Skews not in this sequence will be converted to the nearest value in this sequence. Defaults to `0` or no skew

Value

Returns a numeric vector of the categorize data

Author(s)

Maria Dolores Nieto Canaveras <mnietoca@nebrija.es>, Luis Eduardo Garrido <luisgarrido@pucmm.edu>, Hudson Golino <hfg9s@virginia.edu>, Alexander P. Christensen <alexpaulchristensen@gmail.com>

References

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2011).
Performance of Velicer's minimum average partial factor retention method with categorical variables.
Educational and Psychological Measurement, *71*(3), 551-570.

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., ... & Martinez-Molina, A. (2020). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*(3), 292-320.

Examples

```
# Dichotomous data (no skew)
dichotomous <- categorize(
  data = rnorm(1000),
  categories = 2
)

# Dichotomous data (with positive skew)
dichotomous_skew <- categorize(
  data = rnorm(1000),
  categories = 2,
  skew_value = 1.25
)

# 5-point Likert scale (no skew)
five_likert <- categorize(
  data = rnorm(1000),
  categories = 5
)

# 5-point Likert scale (negative skew)
five_likert <- categorize(
  data = rnorm(1000),
  categories = 5,
  skew_value = -0.45
)
```

`simulate_factors` *Simulates Latent Factor Data*

Description

Simulates data from a latent factor model based on many manipulable parameters. Parameters do not have default values and must each be set. See examples to get started

Usage

```
simulate_factors(
  factors,
  variables,
  variables_range = NULL,
  loadings,
  loadings_range = NULL,
  cross_loadings,
  cross_loadings_range = NULL,
  correlations,
  correlations_range = NULL,
  sample_size,
  variable_categories = Inf,
  categorical_limit = 6,
  skew = 0,
  skew_range = NULL
)
```

Arguments

`factors`	Numeric (length = 1). Number of factors
`variables`	Numeric (length = 1 or `factors`). Number of variables per factor. Can be a single value or as many values as there are factors. Minimum three variables per factor
`variables_range`	Numeric (length = 2). Range of variables to randomly select from a random uniform distribution. Minimum three variables per factor
`loadings`	Numeric or matrix (length = 1, `factors`, total number of variables (`factors` x `variables`), or `factors` x total number of variables. Loadings drawn from a random uniform distribution using +/- 0.10 of value input. Can be a single value or as many values as there are factors (corresponding to the factors). Can also be a loading matrix. Columns must match factors and rows must match total variables (`factors` x `variables`) General effect sizes range from small (0.40), moderate (0.55), to large (0.70)
`loadings_range`	Numeric (length = 2). Range of loadings to randomly select from a random uniform distribution. General effect sizes range from small (0.40), moderate (0.55), to large (0.70)
`cross_loadings`	Numeric or matrix(length = 1, `factors`, or `factors` x total number of variables. Cross-loadings drawn from a random normal distribution with a mean of 0 and standard deviation of value input. Can be a single value or as many values as there are factors (corresponding to the factors). Can also be a loading matrix. Columns must match factors and rows must match total variables (`factors` x `variables`)
`cross_loadings_range`	Numeric (length = 2). Range of cross-loadings to randomly select from a random uniform distribution

`correlations` Numeric (length = 1 or `factors` x `factors`). Can be a single value that will be used for all correlations between factors. Can also be a square matrix (`factors` x `factors`). General effect sizes range from orthogonal (0.00), small (0.30), moderate (0.50), to large (0.70)

`correlations_range` Numeric (length = 2). Range of correlations to randomly select from a random uniform distribution. General effect sizes range from orthogonal (0.00), small (0.30), moderate (0.50), to large (0.70)

`sample_size` Numeric (length = 1). Number of cases to generate from a random multivariate normal distribution using `rmvnorm`

`variable_categories` Numeric (length = 1 or total variables (`factors` x `variables`)). Number of categories for each variable. `Inf` is used for continuous variables; otherwise, values reflect number of categories

`categorical_limit` Numeric (length = 1). Values greater than input value are considered continuous. Defaults to `6` meaning that 7 or more categories are considered continuous (i.e., variables are *not* categorized from continuous to categorical)

`skew` Numeric (length = 1 or categorical variables). Skew to be included in categorical variables. It is randomly sampled from provided values. Can be a single value or as many values as there are (total) variables. Current skew implementation is between -2 and 2 in increments of 0.05. Skews that are not in this sequence will be converted to their nearest value in the sequence. Not recommended to use with `variables_range`. Future versions will incorporate finer skews

`skew_range` Numeric (length = 2). Randomly selects skews within in the range. Somewhat redundant with `skew` but more flexible. Compatible with `variables_range`

Value

Returns a list containing:

`data` Simulated data from the specified factor model

`population_correlation` Population correlation matrix

`parameters` A list containing the parameters used to generate the data:

- `factors` Number of factors
- `variables` Variables on each factor
- `loadings` Loading matrix
- `factor_correlations` Correlations between factors
- `categories` Categories for each variable
- `skew` Skew for each variable

Author(s)

Maria Dolores Nieto Canaveras <mnietoca@nebrija.es>, Alexander P. Christensen <alexpaulchristensen@gmail.com>, Hudson Golino <hfg9s@virginia.edu>, Luis Eduardo Garrido <luisgarrido@pucmm.edu>

References

Garrido, L. E., Abad, F. J., & Ponsoda, V. (2011).
Performance of Velicer's minimum average partial factor retention method with categorical variables.
Educational and Psychological Measurement, *71*(3), 551-570.

Golino, H., Shi, D., Christensen, A. P., Garrido, L. E., Nieto, M. D., Sadana, R., ... & Martinez-Molina, A. (2020). Investigating the performance of exploratory graph analysis and traditional techniques to identify the number of latent factors: A simulation and tutorial. *Psychological Methods*, *25*(3), 292-320.

Examples

```
# Generate factor data
two_factor <- simulate_factors(
  factors = 2, # factors = 2
  variables = 6, # variables per factor = 6
  loadings = 0.55, # loadings between = 0.45 to 0.65
  cross_loadings = 0.05, # cross-loadings N(0, 0.05)
  correlations = 0.30, # correlation between factors = 0.30
  sample_size = 1000 # number of cases = 1000
)

# Randomly vary loadings
two_factor_loadings <- simulate_factors(
  factors = 2, # factors = 2
  variables = 6, # variables per factor = 6
  loadings_range = c(0.30, 0.80), # loadings between = 0.30 to 0.80
  cross_loadings = 0.05, # cross-loadings N(0, 0.05)
  correlations = 0.30, # correlation between factors = 0.30
  sample_size = 1000 # number of cases = 1000
)

# Generate dichotomous data
two_factor_dichotomous <- simulate_factors(
  factors = 2, # factors = 2
  variables = 6, # variables per factor = 6
  loadings = 0.55, # loadings between = 0.45 to 0.65
  cross_loadings = 0.05, # cross-loadings N(0, 0.05)
  correlations = 0.30, # correlation between factors = 0.30
  sample_size = 1000, # number of cases = 1000
  variable_categories = 2 # dichotomous data
)

# Generate dichotomous data with skew
two_factor_dichotomous_skew <- simulate_factors(
  factors = 2, # factors = 2
  variables = 6, # variables per factor = 6
  loadings = 0.55, # loadings between = 0.45 to 0.65
  cross_loadings = 0.05, # cross-loadings N(0, 0.05)
  correlations = 0.30, # correlation between factors = 0.30
  sample_size = 1000, # number of cases = 1000
```

```
  variable_categories = 2, # dichotomous data
  skew = 1 # all variables with have a skew of 1
)

# Generate dichotomous data with variable skew
two_factor_dichotomous_skew <- simulate_factors(
  factors = 2, # factors = 2
  variables = 6, # variables per factor = 6
  loadings = 0.55, # loadings between = 0.45 to 0.65
  cross_loadings = 0.05, # cross-loadings N(0, 0.05)
  correlations = 0.30, # correlation between factors = 0.30
  sample_size = 1000, # number of cases = 1000
  variable_categories = 2, # dichotomous data
  skew_range = c(-2, 2) # skew = -2 to 2 (increments of 0.05)
)
```

---

skew_tables *Skew Tables*

Description

Tables for skew based on the number of categories (2, 3, 4, 5, or 6) in the data

Usage

```
data(skew_tables)
```

Format

A list (length = 5)

Examples

```
data("skew_tables")
```

Index