Package 'LikertMakeR'

March 2, 2024

Version 0.2.0

Type Package

Title Synthesise and Correlate Rating-Scale Data

Description Synthesise rating-scale data with predefined
first & second moments (mean & standard deviation) and, optionally,
correlate multiple vectors with predefined correlation matrix.
Also generate synthetic rating-scale data with predefined Cronbach's Alpha.

URL https://github.com/WinzarH/LikertMakeR

BugReports https://github.com/WinzarH/LikertMakeR/issues

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

VignetteBuilder knitr

Depends R (>= 4.2.0)

Imports Rcpp

LinkingTo Rcpp, RcppArmadillo

Suggests rmarkdown, knitr, testthat (>= 3.0.0)

RoxygenNote 7.3.1

Config/testthat/edition 3

NeedsCompilation yes

Author Hume Winzar [cre, aut] (<https://orcid.org/0000-0001-7475-2641>)

Maintainer Hume Winzar <winzar@gmail.com>

Repository CRAN

Date/Publication 2024-03-02 22:12:42 UTC

R topics documented:

`alpha` *Calculate Cronbach's Alpha from a correlation matrix or dataframe*

Description

`alpha()` calculate Cronbach's Alpha from a given correlation matrix or a given dataframe.

Usage

```
alpha(cormatrix = NULL, data = NULL)
```

Arguments

`cormatrix`	(real) a square symmetrical matrix with values ranging from -1 to +1 and '1' in the diagonal
`data`	(real) a dataframe or matrix

Value

a single value

Examples

```
## Sample data frame
df <- data.frame(
V1  =  c(4, 2, 4, 3, 2, 2, 2, 1),
V2  =  c(4, 1, 3, 4, 4, 3, 2, 3),
V3  =  c(4, 1, 3, 5, 4, 1, 4, 2),
V4  =  c(4, 3, 4, 5, 3, 3, 3, 3)
)

## example correlation matrix
corMat <- matrix(
  c(
    1.00, 0.35, 0.45, 0.70,
    0.35, 1.00, 0.60, 0.55,
    0.45, 0.60, 1.00, 0.65,
```

```
    0.70, 0.55, 0.65, 1.00
  ),
  nrow = 4, ncol = 4
)

## apply function examples

alpha(cormatrix = corMat)

alpha(, df)

alpha(corMat, df)
```

`eigenvalues` *calculate eigenvalues of a correlation matrix with optional scree plot*

Description

`eigenvalues()` calculate eigenvalues of a correlation matrix and optionally produces a scree plot.

Usage

```
eigenvalues(cormatrix, scree = FALSE)
```

Arguments

`cormatrix`	(real, matrix) a correlation matrix
`scree`	(logical) default = `FALSE`. If `TRUE` (or `1`), then `eigenvalues()` produces a scree plot to illustrate the eigenvalues

Value

a vector of eigenvalues

report on positive-definite status of cormatrix

Examples

```
## define parameters

correlationMatrix <- matrix(
  c(
    1.00, 0.25, 0.35, 0.40,
    0.25, 1.00, 0.70, 0.75,
    0.35, 0.70, 1.00, 0.80,
    0.40, 0.75, 0.80, 1.00
  ),
  nrow = 4, ncol = 4
```

```
)

## apply function

evals <- eigenvalues(cormatrix = correlationMatrix)
evals <- eigenvalues(correlationMatrix, 1)
```

`lcor` *Rearrange columns in a data-frame to fit a predefined correlation matrix*

Description

`lcor_C()` rearranges values in each column of a data-frame so that columns are correlated to match a predefined correlation matrix.

Usage

```
lcor(data, target)
```

Arguments

`data`	data-frame that is to be rearranged
`target`	target correlation matrix - should be a symmetric (square) k*k matrix

Details

Values in a column do not change, so univariate statistics remain the same.

Value

Returns a data-frame whose column-wise correlations approximate a user-specified correlation matrix

Examples

```
n <- 32
lowerbound <- 1
upperbound <- 5
items <- 5
mydat3 <- data.frame(
  x1 = lfast(n, 2.5, 0.75, lowerbound, upperbound, items),
  x2 = lfast(n, 3.0, 1.50, lowerbound, upperbound, items),
  x3 = lfast(n, 3.5, 1.00, lowerbound, upperbound, items)
)
cor(mydat3) |> round(3)
tgt3 <- matrix(
  c(
```

```
    1.00, 0.50, 0.75,
    0.50, 1.00, 0.25,
    0.75, 0.25, 1.00
    ),
  nrow = 3, ncol = 3
)
new3 <- lcor(mydat3, tgt3)
cor(new3) |> round(3)
```

`lexact` *Deprecated. Use lfast() instead*

Description

`lexact` is DEPRECATED. Replaced by `lfast`. Previously, `lexact` used a Differential Evolution (DE) algorithm to find an optimum solution for finding desired mean and standard deviation, but we found that the updated`lfast` function is much faster and just as accurate. Also the package is much less bulky.

Usage

```
lexact(n, mean, sd, lowerbound, upperbound, items = 1)
```

Arguments

`n`	(positive, int) number of observations to generate
`mean`	(real) target mean
`sd`	(real) target standard deviation
`lowerbound`	(positive, int) lower bound (e.g. '1' for a 1-5 rating scale)
`upperbound`	(positive, int) upper bound (e.g. '5' for a 1-5 rating scale)
`items`	(positive, int) number of items in the rating scale. Default = 1

Value

a vector of simulated data approximating user-specified conditions.

Examples

```
x <- lexact(
  n = 256,
  mean = 4.0,
  sd = 1.0,
  lowerbound = 1,
  upperbound = 7,
  items = 6
```

```
)

x <- lexact(256, 2, 1.8, 0, 10)
```

`lfast` *Rating scale data (e.g. Likert scale) from a Scaled Beta Distribution*

Description

`lfast()` generates random discrete values from a (scaled Beta distribution) so the data replicate a rating scale - for example,a 1-5 scale made from 5 items (questions) or 0-10 likelihood-of-purchase scale. `lfast()` takes repeated samples selecting a vector that best fits the desired moments, while `lfast()` takes just one sample. `lfast()` is slightly slower than `lfast()`.

Usage

```
lfast(n, mean, sd, lowerbound, upperbound, items = 1)
```

Arguments

`n`	(positive, int) number of observations to generate
`mean`	(real) target mean
`sd`	(real) target standard deviation
`lowerbound`	(positive, int) lower bound (e.g. '1' for a 1-5 rating scale)
`upperbound`	(positive, int) upper bound (e.g. '5' for a 1-5 rating scale)
`items`	(positive, int) number of items in the rating scale. Default = 1

Value

a vector of simulated data approximating user-specified conditions.

Examples

```
x <- lfast(
  n = 256,
  mean = 4.0,
  sd = 1.0,
  lowerbound = 1,
  upperbound = 7,
  items = 6
)

x <- lfast(256, 2, 1.8, 0, 10)
```

`makeCorrAlpha` *Construct a random correlation matrix of given dimensions from predefined Cronbach's Alpha*

Description

`makeCorrAlpha()` generates a random correlation matrix of given dimensions and predefined Cronbach's Alpha

Usage

```
makeCorrAlpha(items, alpha, variance = 0.5)
```

Arguments

`items`	(positive, int) number of rows & columns to generate
`alpha`	(real) target Cronbach's Alpha (usually positive, must be between -1 and +1)
`variance`	(positive, real) Default = 0.5. User-provided standard deviation of values sampled from a normally-distributed log transformation.

Value

a correlation matrix

Note

Random values generated by `makeCorrAlpha()` are highly volatile. `makeCorrAlpha()` may not generate a feasible (positive-definite) correlation matrix, especially when

- variance is high relative to
  - desired Alpha, and
  - desired correlation dimensions

`makeCorrAlpha()` will inform the user if the resulting correlation matrix is positive definite, or not.

If the returned correlation matrix is not positive-definite, a feasible solution may still be possible. The user is encouraged to try again, possibly several times, to find one.

Examples

```
items <- 4
alpha <- 0.85
variance <- 0.5

set.seed(42)
cor_matrix <- makeCorrAlpha(items, alpha, variance)

print(cor_matrix)
```

```
alpha(cor_matrix)
eigenvalues(cor_matrix,1)

cor_matrix2 <- makeCorrAlpha(items = 6, alpha = 0.95)

print(cor_matrix2)
alpha(cor_matrix2)
eigenvalues(cor_matrix2,1)
```

`makeItems` *Generate synthetic rating-scale data with predefined first and second moments and a predefined correlation matrix*

Description

`makeItems()` generates a dataframe of random discrete values from a (scaled Beta distribution) so the data replicate a rating scale, and are correlated close to a predefined correlation matrix. `makeItems()` is wrapper function for:

- `lfast()`, which takes repeated samples selecting a vector that best fits the desired moments, and
- `lcor()`, which rearranges values in each column of the dataframe so they closely match the desired correlation matrix.

Usage

```
makeItems(n, means, sds, lowerbound, upperbound, cormatrix)
```

Arguments

`n`	(positive, int) number of observations to generate
`means`	(real) target means: a vector of length k of mean values for each scale item
`sds`	(positive, real) target standard deviations: a vector of length k of standard deviation values for each scale item
`lowerbound`	(positive, int) a vector of length k (same as rows & columns of correlation matrix) of values for lower bound of each scale item (e.g. '1' for a 1-5 rating scale)
`upperbound`	(positive, int) a vector of length k (same as rows & columns of correlation matrix) of values for upper bound of each scale item (e.g. '5' for a 1-5 rating scale)
`cormatrix`	(real, matrix) the target correlation matrix: a square symmetric matrix of values renging between -1 and +1, and '1' in the diagonal.

Value

a dataframe of rating-scale values

Examples

```
## define parameters

n <- 16

lowerbound <- rep(1, 4)

upperbound <- rep(5, 4)

corMat <- matrix(
  c(
    1.00, 0.25, 0.35, 0.40,
    0.25, 1.00, 0.70, 0.75,
    0.35, 0.70, 1.00, 0.80,
    0.40, 0.75, 0.80, 1.00
  ),
  nrow = 4, ncol = 4
)

dfMeans <- c(2.5, 3.0, 3.0, 3.5)

dfSds <- c(1.0, 1.0, 1.5, 0.75)

## apply function

df <- makeItems(n = n, means = dfMeans, sds = dfSds,
lowerbound = lowerbound, upperbound = upperbound, cormatrix = corMat)

## test function

print(df)

apply(df, 2, mean) |> round(3)

apply(df, 2, sd) |> round(3)

cor(df) |> round(3)
```

Index