Package ‘pscore’

June 24, 2015

Title Standardizing Physiological Composite Risk Endpoints

Version 0.1-2

Date 2015-06-24

Author Joshua F. Wiley <jwiley.psych@gmail.com>

Maintainer Joshua F. Wiley <jwiley.psych@gmail.com>

Description Provides a number of functions to
simplify and automate the scoring, comparison, and evaluation of
different ways of creating composites of data. It is particularly
aimed at facilitating the creation of physiological composites of
metabolic syndrome symptom score (MSSS) and allostatic load (AL).

Depends R (>= 3.1.0)

Imports ggplot2, reshape2, lavaan

License LGPL-3

ByteCompile true

URL https://jwiley.github.io/score-project/

LazyData true

NeedsCompilation no

Repository CRAN

Date/Publication 2015-06-24 15:29:16

R topics documented:

BioDB *Biomarker Thresholds Database*

Description

This data set lists the clinical high risk thresholds for a variety of biomarkers

Format

a `list`.

Source

Various publications

distanceScores *Calculate distance scores on data in preparation for composite scoring*

Description

Calculate distance scores on data in preparation for composite scoring

Usage

```
distanceScores(d, g, thresholds, higherisbetter, winsorize = FALSE,
  better = TRUE, na.rm = TRUE, saveall = FALSE)
```

Arguments

`d`	The data
`g`	A grouping variable
`thresholds`	Thresholds to use when calculating distances (e.g., median, clinical thresholds, etc.). If groups are used, must be thresholds for each group (e.g., to allow separate thresholds for females and males).
`higherisbetter`	A logical vector for each biomarker whether higher scores are better or not.
`winsorize`	Whether to winsorize the data or not. Defaults to `FALSE`. If not `FALSE`, the percentile to winsorize at. For example, .01 would be the .01 and the 1 - .01 percentiles.
`better`	Logical indicating whether “better” values than the threshold are allowed. Defaults to `TRUE`.
`na.rm`	A logical whether missing values should be ommitted. Defaults to `TRUE`.
`saveall`	A logical whether to save all intermediary datasets and graphs. Defaults to `FALSE`.

Value

A list of results.

See Also

Other composite: `factorComposite`; `mahalanobisComposite`; `prepareComposite`; `sumComposite`

Examples

```
# this example creates distances for the built in mtcars data
# see ?mtcars for more details
# The distances are calculated from the "best" in the dataset
# defined by these thresholds
thresholds <- with(mtcars, c(
  mpg = max(mpg),
  hp = max(hp),
  wt = min(wt),
  qsec = min(qsec)))

# higher mpg and hp are better,
# whereas lower wt and qsec are better
dres <- distanceScores(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(thresholds),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  saveall = TRUE)

# see a density plot of the distance scores
dres$Density

# cleanup
rm(thresholds, dres)
```

`factorComposite` *Score Data Using a Factor Model*

Description

Create a composite using a Factor Model

Usage

```
factorComposite(object, type = c("onefactor", "secondorderfactor",
  "bifactor"), factors = "")
```

Arguments

`object` An object ready for use

`type` A character string indicating the type of factor model to use

`factors` A named list where names are the factor names and each element is a character string of the indicator names.

Value

A list of results.

See Also

Other composite: distanceScores; mahalanobisComposite; prepareComposite; sumComposite

Examples

```
# this example creates distances for the built in mtcars data
# see ?mtcars for more details
# The distances are calculated from the "best" in the dataset
# defined by these thresholds
thresholds <- with(mtcars, c(
  mpg = max(mpg),
  hp = max(hp),
  wt = min(wt),
  qsec = min(qsec)))

# higher mpg and hp are better,
# whereas lower wt and qsec are better
dres <- distanceScores(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(thresholds),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  saveall = TRUE)

# see a density plot of the distance scores
dres$Density

# now prepare to create the composite
# covariance matrix will be calculated from the data
# and data will be standardized to unit variance by default
cprep <- prepareComposite(dres)

# now we can create the composite based on summing the (standardized)
# distances from our defined thresholds
# by default, distances are squared, then summed, and then square rooted
# to be back on the original scale
fcomp <- factorComposite(cprep, type = "onefactor")

# view a histogram of the composite scores
fcomp$ScoreHistogram

# summarize the composite scores
summary(fcomp$Scores$Comp)

# cleanup
rm(thresholds, dres, cprep, fcomp)
```

`ldensity` *Density Plot for a Long Dataset*

Description

Internal function only, not meant for general use Simple wrapper around ggplot2 functionaly to create density plots, potentially for many variables and coloured by group.

Usage

```
ldensity(data, melt = FALSE, x, facet, g, hist = FALSE)
```

Arguments

`data`	A dataset (or melt()ed dataset)
`melt`	Logical whether to melt() dataset
`x`	name of variable for density
`facet`	A variable to use for facetting
`g`	A variable to use for grouping/colouring. If `melt=TRUE`, this is used as id.var as well.
`hist`	Logical whether to make a density plot or histogram (if TRUE).

Value

A ggplot2 graph.

Examples

```
# simple facetted plot
pscore:::ldensity(mtcars, TRUE)
# simple coloured plot
pscore:::ldensity(mtcars, x = "mpg", g = "factor(cyl)")
```

`mahalanobisComposite` *Score Data Using the Mahalanobis Distance*

Description

Create a composite using the Mahalanobis Distance

Usage

```
mahalanobisComposite(object, ncomponents)
```

Arguments

`object` An object of class `compositedata` ready for use

`ncomponents` the number of components to use from the principal component analysis. If missing, defaults to the number of columns in the data.

Value

A list of results.

See Also

Other composite: `distanceScores`; `factorComposite`; `prepareComposite`; `sumComposite`

Examples

```
# this example creates distances for the built in mtcars data
# see ?mtcars for more details
# The distances are calculated from the "best" in the dataset
# defined by these thresholds
thresholds <- with(mtcars, c(
  mpg = max(mpg),
  hp = max(hp),
  wt = min(wt),
  qsec = min(qsec)))

# higher mpg and hp are better,
# whereas lower wt and qsec are better
dres <- distanceScores(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(thresholds),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  saveall = TRUE)

# see a density plot of the distance scores
dres$Density

# now prepare to create the composite
# covariance matrix will be calculated from the data
# and data will be standardized to unit variance by default
cprep <- prepareComposite(dres)

# now we can create the composite based on mahalanobis distances
# from our defined thresholds
mcomp <- mahalanobisComposite(cprep)

# view a histogram of the composite scores
mcomp$ScoreHistogram

# summarize the composite scores
summary(mcomp$Scores)

# check the screeplot and loadings
```

```
mcomp$Screeplot
mcomp$LoadingGraph
# examine the loadings as a table
mcomp$LoadingTable

# one component is adequate to explain these data
# to be safe can pick first two and re-run model

# use only first two components
mcomp2 <- mahalanobisComposite(cprep, ncomponents = 2)

# view a histogram of the updated composite scores
mcomp2$ScoreHistogram

# summarize the composite scores
summary(mcomp2$Scores)

# compare using all versus two components
plot(mcomp$Scores, mcomp2$Scores)

# cleanup
rm(thresholds, dres, cprep, mcomp, mcomp2)
```

`prepareComposite` *Prepare data to have a composite calculated*

Description

Prepare data to have a composite calculated

Usage

```
prepareComposite(object, covmat, standardize = TRUE)
```

Arguments

`object` An object ready for use

`covmat` The covariance matrix to use. If missing, austomatically calculated from the data.

`standardize` A logical value whether to standardize the data or not. Defaults to `TRUE`.

Value

A list of results.

See Also

Other composite: `distanceScores`; `factorComposite`; `mahalanobisComposite`; `sumComposite`

Examples

```
# this example creates distances for the built in mtcars data
# see ?mtcars for more details
# The distances are calculated from the "best" in the dataset
# defined by these thresholds
thresholds <- with(mtcars, c(
  mpg = max(mpg),
  hp = max(hp),
  wt = min(wt),
  qsec = min(qsec)))

# higher mpg and hp are better,
# whereas lower wt and qsec are better
dres <- distanceScores(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(thresholds),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  saveall = TRUE)

# see a density plot of the distance scores
dres$Density

# now prepare to create the composite
# covariance matrix will be calculated from the data
# and data will be standardized to unit variance by default
cprep <- prepareComposite(dres)
# cleanup
rm(thresholds, dres, cprep)
```

sumComposite *Score Data Using a simple sum*

Description

Create a composite using summation

Usage

```
sumComposite(object, transform = c("square", "abs", "none"), type = c("sum",
  "mean"), systems)
```

Arguments

`object` An object ready for use

`transform` A character string indicating the type of transformation to use. One of “square”, “abs”, or “none”, which either sums the raw data, sums the squared data and then takes the square root, or sums the absolute values of the data.

`type` A character string indicating the type of aggregation to use. One of “sum” or “mean”.

`systems` An optional list where each element is a character vector of the variable names within a particular system. If given, scores are first averaged within a system, before being aggregated across systems.

Value

A list of results.

See Also

Other composite: `distanceScores`; `factorComposite`; `mahalanobisComposite`; `prepareComposite`

Examples

```
# this example creates distances for the built in mtcars data
# see ?mtcars for more details
# The distances are calculated from the "best" in the dataset
# defined by these thresholds
thresholds <- with(mtcars, c(
  mpg = max(mpg),
  hp = max(hp),
  wt = min(wt),
  qsec = min(qsec)))

# higher mpg and hp are better,
# whereas lower wt and qsec are better
dres <- distanceScores(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(thresholds),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  saveall = TRUE)

# see a density plot of the distance scores
dres$Density

# now prepare to create the composite
# covariance matrix will be calculated from the data
# and data will be standardized to unit variance by default
cprep <- prepareComposite(dres)

# now we can create the composite based on summing the (standardized)
# distances from our defined thresholds
# by default, distances are squared, then summed, and then square rooted
# to be back on the original scale
scomp <- sumComposite(cprep, "square", "sum")

# view a histogram and summary of the composite scores
scomp$ScoreHistogram
summary(scomp$Scores)

# calculate average (mean) instead of sum
scomp2 <- sumComposite(cprep, "square", "mean")
```

```
# view a histogram and summary of the composite scores
scomp2$ScoreHistogram
summary(scomp2$Scores)

# scores are still the same
plot(scomp$Scores, scomp2$Scores)

# first average scores within a system, then sum
# note that within a system, scores are always averaged,
# never summed.
scomp3 <- sumComposite(cprep, "square", "sum",
  systems = list(
    environment = c("mpg"),
    performance = c("hp", "qsec", "wt")))

# view a histogram and summary of the composite scores
scomp3$ScoreHistogram
summary(scomp3$Scores)

# compare all three scores
# because of the different number of indicators within each system
# there is a re-weighting for S3
plot(data.frame(S1 = scomp$Scores, S2 = scomp2$Scores, S3 = scomp3$Scores))

# cleanup
rm(thresholds, dres, cprep, scomp, scomp2, scomp3)
```

winsorizor *Winsorize at specified percentiles*

Description

Simple function winsorizes data at the specified percentile.

Usage

```
winsorizor(d, percentile, na.rm = TRUE)
```

Arguments

d	A vector, matrix, or data frame to be winsorized
percentile	The percentile bounded by [0, 1] to winsorize data at
na.rm	A logical whether to remove NAs.

Value

winsorized data. Attributes are included to list the exact values (for each variable, if a data frame or matrix) used to winsorize at the lower and upper ends.

Examples

```
dev.new(width = 10, height = 5)
par(mfrow = c(1, 2))
hist(as.vector(eurodist), main = "Eurodist")
hist(winsorizor(as.vector(eurodist), .05), main = "Eurodist with lower and upper\n5% winsorized")
```

Index