

Package: pscore (via r-universe)

August 27, 2024

Title Standardizing Physiological Composite Risk Endpoints

Version 0.4.0

URL <https://score-project.org>, <https://github.com/JWiley/score-project>

BugReports <https://github.com/JWiley/score-project/issues>

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 (MetSSS) and allostatic load (AL). Provides a wrapper to calculate the MetSSS on new data using the Healthy Hearts formula.

License LGPL-3

Encoding UTF-8

Depends R (>= 4.0.0), methods

Imports stats, ggplot2, reshape2, lavaan, JWileymisc (>= 1.3.0)

Suggests testthat, covr, knitr, rmarkdown

VignetteBuilder knitr

LazyData false

RoxygenNote 7.2.0

Collate 'MetSSS.R' 'MetSSS_model.R' 'methods.R' 'misc.R' 'classes.R' 'composite.R' 'onLoad.R' 'predictions.R'

Repository <https://jwiley.r-universe.dev>

RemoteUrl <https://github.com/jwiley/score-project>

RemoteRef HEAD

RemoteSha f4882a7cb61a37de44a00a14dc403766e8e47e78

Contents

.preparePredict	2
BioDB	3
CompositeData	3
CompositeData-class	4
CompositeReady	5
CompositeReady-class	6
dropData	7
factorComposite	7
ldensity	9
mahalanobisComposite	10
MahalanobisScores-class	11
MetSSS	13
MetSSS_model	13
predictCS	14
prepareComposite	15
sumComposite	16
Index	19

.preparePredict	<i>Internal function to prepare data for prediction</i>
-----------------	---

Description

Internal function to prepare data for prediction

Usage

```
.preparePredict(object, newdata, groups)
```

Arguments

object	An object of S4 class “MahalanobisScores”, “SumScores”, or “FactorScores” containing a model and results to be used to get predictions on new data.
newdata	A data frame with identical variable names as was used to build the initial model.
groups	A vector with the same length as the data frame in newdata, has rows, containing the groups each row belongs to. See CompositeData for more details.

Value

An object of S4 class “CompositeReady”

 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

 CompositeData

An constructor function for the S4 CompositeData class

Description

An constructor function for the S4 CompositeData class

Usage

```
CompositeData(rawdata, groups, thresholds, higherisbetter, k, rawtrans)
```

Arguments

rawdata	A data frame with at least one row and column
groups	an optional character vector. If omitted defaults to a character vector of all “one”s.
thresholds	an optional named list where names match the names in groups. If using defaults for groups, should name it “one”.
higherisbetter	an optional logical vector
k	an optional integer, the number of columns in the raw data
rawtrans	A list of functions to transform the raw data (and thresholds). This list should be in the same order as the variables. For variables that should not be transformed, use: <code>function(x) x</code> .

Value

An S4 object of class “CompositeData”

Examples

```
## no custom raw data transformations
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    qsec = min(qsec)))
),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE))

## square root transform qsec
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    qsec = min(qsec)))
),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  rawtrans = list(
    mpg = function(x) x,
    hp = function(x) x,
    wt = function(x) x,
    qsec = sqrt))

## cleanup
rm(d)
```

CompositeData-class *A S4 class to represent data for creating a composite*

Description

A S4 class to represent data for creating a composite

Slots

rawdata A data frame of the data to be used for the composite scores

groups A character string, the same length as the number of rows of the data indicating to which group each row belong. May be all the same if only one group present in the data.

thresholds A list with as many elements as there are unique groups in the data, and where each element is a vector the same length as the number of columns in the data frame, indicating the reference thresholds for each variable, by group (all the same if only one group).

higherisbetter A logical vector the same length as the number of columns in the data frame, indicating whether higher is better for each variable (if TRUE) and otherwise (if FALSE) that lower is better, indicating that variable should be reversed.

k The number of variables as an integer.

rawtrans A list of functions to transform the raw data (and thresholds).

CompositeReady *An constructor function for the S4 CompositeReady class*

Description

An constructor function for the S4 CompositeReady class

Usage

```
CompositeReady(
  data,
  covmat,
  sigma,
  standardize,
  use.prethreshold,
  distances,
  distanceDensity,
  winsorizedValues,
  better,
  rawdata,
  groups,
  thresholds,
  higherisbetter,
  k,
  rawtrans
)
```

Arguments

data	A data frame of the ready to use data
covmat	A covariance matrix of the data
sigma	A vector of the standard deviations of each variable
standardize	A logical whether the data were standardized or not
use.prethreshold	A logical value whether to calculate covariance matrix based on the data after winsorizing, but before applying the threshold. Defaults to FALSE, so that covariances are calculated after thresholds (if any) are applied.
distances	A data frame of the distance scores
distanceDensity	A ggplot2 graph of the densities of each distance score. If not passed, generated automatically from the data.

winsorizedValues	A data frame indicating the values used (if any) for winsorization. Should have one row for each variable in the dataset.
better	A logical vector the same length as the number of columns in the distance scores indicating whether higher or lower values are better for each.
rawdata	A data frame with at least one row and column
groups	an optional character vector. If omitted defaults to a character vector of all "one"s.
thresholds	an optional named list where names match the names in groups. If using defaults for groups, should name it "a".
higherisbetter	an optional logical vector
k	an optional integer, the number of columns in the raw data
rawtrans	A list of functions to transform the raw data (and thresholds).

Value

An S4 object of class "CompositeReady"

Examples

```
#make me!
```

CompositeReady-class *An S4 class to represent composite ready data*

Description

An S4 class to represent composite ready data

Slots

data A data frame ready for use to generate composite scores

covmat A covariance matrix

sigma The standard deviation of each variable

standardize A logical value whether standardization was applied

distances A data frame of the distance scores

distanceDensity A ggplot2 density graph

winsorizedValues A data frame of the values at which each variable was winsorized. If percentile is 0, that means no winsorization, and values will simply be min and max.

better A logical value whether better scores than threshold were allowed

use.prethreshold A logical value whether covariance matrix was calculated from pre threshold but post winsorizing data.

dropData	<i>Drop unnecessary data from a MahalanobisComposite object.</i>
----------	--

Description

This function removes graphs and other sensitive data.

Usage

```
dropData(object)
```

Arguments

object A MahalanobisComposite object

Value

A MahalanobisComposite object with some slots replaced with missing values

Examples

```
# make me!!!
```

factorComposite	<i>Score Data Using a Factor Model</i>
-----------------	--

Description

Create a composite using a Factor Model

Usage

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

Arguments

object An object of class CompositeReady
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

An S4 object of class FactorScores.

See Also

Other composite: [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
# First we create an appropriate CompositeData class object
# higher mpg & hp are better and lower wt & qsec are better
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    qsec = min(qsec)))
),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE))

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

# see a density plot of the distance scores
dres@distanceDensity
# regular summary of distance scores
summary(dres@distances)

# examine covariance matrix
round(dres@covmat,2)

# 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(dres, type = "onefactor")

# view a histogram of the composite scores
fcomp@scoreHistogram

# summarize the composite scores
summary(fcomp@scores)

## Not run:
# we can also fit a second-order factor model
# there are not enough indicators to identify the factor
```

```
# and so lavaan gives us warning messages
fcomp2 <- factorComposite(dres, type = "secondorderfactor",
  factors = list(speed = c("hp", "qsec")))

# view a histogram of the composite scores
fcomp2@scoreHistogram

# summarize the composite scores
summary(fcomp2@scores)

# compare one and second-order factor model scores
plot(fcomp@scores, fcomp2@scores)

# cleanup
rm(d, dres, fcomp, fcomp2)

## End(Not run)
```

Idensity

Density Plot for a Long Dataset

Description

Internal function only, not meant for general use Simple wrapper around ggplot2 functionality 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 faceted plot
pscore::ldensity(mtcars, TRUE)
# simple coloured plot
pscore::ldensity(mtcars, x = "mpg", g = "cyl")
```

mahalanobisComposite *Score Data Using the Mahalanobis Distance*

Description

Create a composite using the Mahalanobis Distance

Usage

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

Arguments

object	An object of class CompositeReady
ncomponents	the number of components to use from the principal component analysis. If missing, defaults to the number of columns in the data.
pca	An optional PCA object from princomp to use. If not passed, will be calculated from the data.

Value

An S4 object of class MahalanobisScores.

See Also

Other composite: [factorComposite\(\)](#), [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
# First we create an appropriate CompositeData class object
# higher mpg & hp are better and lower wt & qsec are better
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    qsec = min(qsec)))
),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE))
```

```
# create the distance scores
# and the composite
# covariance matrix will be calculated from the data
# and data will be standardized to unit variance by default
dres <- prepareComposite(d)

# see a density plot of the distance scores
dres@distanceDensity
# regular summary of distance scores
summary(dres@distances)

# examine covariance matrix
round(dres@covmat,2)

# now we can create the composite based on mahalanobis distances
# from our defined thresholds
mcomp <- mahalanobisComposite(dres, 1)

# 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(dres, 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(d, dres, mcomp, mcomp2)
```

MahalanobisScores-class

An S4 class to represent composite scores based on Mahalanobis distance

Description

An S4 class to represent composite scores based on Mahalanobis distance

An S4 class to represent composite scores based on summing

An S4 class to represent composite scores based on (confirmatory) factor analysis

Slots

scores A vector of the final scores

scoreHistogram A histogram of the final scores

screePlot A screeplot from the PCA

loadingGraph A graph of the component loadings

loadingTable A table of all the component loadings

pca Principal component analysis results. A list (coerced from princomp output).

ncomponents The number of components of the PCA to be used.

CompositeReady The original CompositeReady class object passed in

scores A vector of the final scores

scoreHistogram A histogram of the final scores

transform A character string of the type of transformation applied

type A character string indicating whether data were summed or averaged

trans A list of functions to transform data and backtransform it

systems A list with as many elements as there are systems where each element is a character vector containing the variable names for each system. If all variables belong to one system, a list with one element that is missing.

CompositeReady The original CompositeReady class object passed in

scores A vector of the final scores

scoreHistogram A histogram of the final scores

factorScores A data frame with all factor score predictions

type A character string indicating the type of factor model used

factors A list with as many elements as there are specific factors, where each element is a character vector of the variables for a specific factor

Fit A fitted model object from lavaan.

CompositeReady The original CompositeReady class object passed in

MetSSS	<i>Score the MetSSS</i>
--------	-------------------------

Description

Function requires systolic and diastolic blood pressure, triglycerides, waist circumference, HDL cholesterol, blood glucose, and sex.

Usage

```
MetSSS(input)
```

Arguments

input data passed on as newdata to [predictCS](#)

Examples

```
mydata <- data.frame(
  sbp = c(122, 102.5),
  dbp = c(76.5, 64),
  trigs = c(1.47, 1.27),
  hdl = c(2.22, 1.59),
  waist = c(71, 91),
  glucose = c(5.16, 5.82),
  sex = c("Female", "Male"))
```

```
MetSSS(mydata)
```

MetSSS_model	<i>MetSSS Model</i>
--------------	---------------------

Description

Simple example model that predicts hours of watching TV from age and marital status

Usage

```
MetSSS_model
```

Format

an object from [mahalanobisComposite](#)

Source

Healthy Hearts Study

predictCS *Internal function to prepare data for prediction*

Description

Internal function to prepare data for prediction

Usage

```
predictCS(object, newdata, groups)
```

Arguments

object	An object of S4 class “MahalanobisScores”, “SumScores”, or “FactorScores” containing a model and results to be used to get predictions on new data.
newdata	A data frame with identical variable names as was used to build the initial model.
groups	A vector with the same length as the data frame in newdata, has rows, containing the groups each row belongs to. See CompositeData for more details.

Value

An object of S4 class “CompositeReady”

Examples

```
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "disp")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    disp = min(disp)))),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE))
## create the distance scores
## and prepare to create the composite
dres <- prepareComposite(d)

## create composite based on summing the (standardized)
scomp <- sumComposite(dres, "square", "sum")
## use model to generate predictions on new data
predictCS(scomp,
  newdata = mtcars[1, c("mpg", "hp", "wt", "qsec")],
  groups = "one")

## create composite based on mahalanobis distances
mcomp <- mahalanobisComposite(dres)
## use model to generate predictions on new data
predictCS(mcomp,
  newdata = mtcars[1, c("mpg", "hp", "wt", "qsec")],
  groups = "one")
```

```
## note in this too simple example, there are negative variance estimates
## create composite based on factor scores
fcomp <- factorComposite(dres, type = "onefactor")
## use model to generate predictions on new data
predictCS(fcomp,
          newdata = mtcars[1:5, c("mpg", "hp", "wt", "disp")],
          groups = rep("one", 5))
```

prepareComposite	<i>Prepare distance scores on data in preparation for composite scoring</i>
------------------	---

Description

Prepare distance scores on data in preparation for composite scoring

Usage

```
prepareComposite(
  object,
  winsorize = 0,
  values,
  better = TRUE,
  covmat,
  standardize = TRUE,
  use.prethreshold = FALSE
)
```

Arguments

object	An object of class ‘CompositeData’.
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.
values	The values to use for winsorization. Optional. If specified, preempts the percentiles given by winsorize.
better	Logical indicating whether “better” values than the threshold are allowed. Defaults to TRUE. #' @param object An DistanceScores class object
covmat	The covariance matrix to use. If missing, automatically calculated from the data.
standardize	A logical value whether to standardize the data or not. Defaults to TRUE.
use.prethreshold	A logical value whether to calculate covariance matrix based on the data after winsorizing, but before applying the threshold. Defaults to FALSE, so that covariances are calculated after thresholds (if any) are applied.

Value

An S4 object of class “CompositeReady”.

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
# First we create an appropriate CompositeData class object
# higher mpg & hp are better and lower wt & qsec are better
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    qsec = min(qsec)))
),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE),
  rawtrans = list(
    mpg = function(x) x,
    hp = function(x) x,
    wt = function(x) x,
    qsec = sqrt))

# create the distance scores
dres <- prepareComposite(d)

# see a density plot of the distance scores
dres@distanceDensity
# regular summary of distance scores
summary(dres@distances)

# examine covariance matrix
round(dres@covmat,2)
# cleanup
rm(d, dres)
```

Description

Create a composite using summation

Usage

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

Arguments

object	An object of class CompositeReady
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

An S4 object of class SumScores.

See Also

Other composite: [factorComposite\(\)](#), [mahalanobisComposite\(\)](#)

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
# First we create an appropriate CompositeData class object
# higher mpg & hp are better and lower wt & qsec are better
d <- CompositeData(mtcars[, c("mpg", "hp", "wt", "qsec")],
  thresholds = list(one = with(mtcars, c(
    mpg = max(mpg),
    hp = max(hp),
    wt = min(wt),
    qsec = min(qsec)))
),
  higherisbetter = c(TRUE, TRUE, FALSE, FALSE))

# create the distance scores
# and the composite
# covariance matrix will be calculated from the data
# and data will be standardized to unit variance by default
dres <- prepareComposite(d)
```

```
# see a density plot of the distance scores
dres@distanceDensity
# regular summary of distance scores
summary(dres@distances)

# examine covariance matrix
round(dres@covmat,2)

# 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(dres, "square", "sum")

# view a histogram and summary of the composite scores
scomp@scoreHistogram
summary(scomp@scores)

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

# view a histogram and summary of the composite scores
scomp2@scoreHistogram
summary(scomp2@scores)

# scores are still the same (just different scaling)
plot(scomp@scores, scomp2@scores)

# first average scores within a system, then sum
# within a system, scores are always averaged, never summed
scomp3 <- sumComposite(dres, "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(d, dres, scomp, scomp2, scomp3)
```

Index

- * **composite**
 - factorComposite, [7](#)
 - mahalanobisComposite, [10](#)
 - sumComposite, [16](#)
- * **prepare**
 - prepareComposite, [15](#)
 - .preparePredict, [2](#)
- BioDB, [3](#)
- CompositeData, [3](#)
- CompositeData-class, [4](#)
- CompositeReady, [5](#)
- CompositeReady-class, [6](#)
- dropData, [7](#)
- factorComposite, [7](#), [10](#), [17](#)
- FactorScores-class
 - (MahalanobisScores-class), [12](#)
- ldensity, [9](#)
- mahalanobisComposite, [8](#), [10](#), [13](#), [17](#)
- MahalanobisScores-class, [11](#)
- MetSSS, [13](#)
- MetSSS_model, [13](#)
- predictCS, [13](#), [14](#)
- prepareComposite, [15](#)
- sumComposite, [8](#), [10](#), [16](#)
- SumScores-class
 - (MahalanobisScores-class), [12](#)