Package 'setweaver'

February 4, 2026

Type Package

Title Building Sets of Variables in a Probabilistic Framework

Version 1.0.0

Description Create sets of variables based on a mutual information approach. In this context, a set is a collection of distinct elements (e.g., variables) that can also be treated as a single entity. Mutual information, a concept from probability theory, quantifies the dependence between two variables by expressing how much information about one variable can be gained from observing the other. Furthermore, you can analyze, and visualize these sets in order to better understand the relationships among variables.

License CC BY 4.0

Depends R (>= 4.1.0)

Imports dplyr (>= 1.1.4), igraph (>= 2.1.2), permutes (>= 2.8), pheatmap (>= 1.0.13), splitTools (>= 1.0.1)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

VignetteBuilder knitr

URL https://github.com/nicolasleenaerts/setweaver

NeedsCompilation no

Author Nicolas Leenaerts [aut, cre, cph] (ORCID: <https://orcid.org/0000-0003-2421-6845>), Aaron Fisher [aut, cph] (ORCID: <https://orcid.org/0000-0001-9754-4618>)

Maintainer Nicolas Leenaerts <nicolas.leenaerts@kuleuven.be>

Repository CRAN

Date/Publication 2026-02-04 18:00:08 UTC

Contents

`ce` *ce*

Description

Computes the conditional entropy $H(Y \mid X)$ for two binary vectors ‘y‘ (outcome) and ‘x‘ (predictor).

Usage

```
ce(y, x)
```

Arguments

`y`	A binary outcome vector (0/1 or logical). Must be the same length as ‘x‘.
`x`	A binary predictor vector (0/1 or logical). Must be the same length as ‘y‘.

Value

A numeric scalar giving $H(Y \mid X)$.

Examples

```
ce(misimdata$y,misimdata$x1)
```

cprob *cprob*

Description

Computes the conditional probability $P(Y = 1 \mid X = 1)$ for two binary vectors ‘y‘ and ‘x‘. Rows with missing values in either vector are excluded.

Usage

```
cprob(y, x)
```

Arguments

y	A binary outcome vector (0/1 or logical). Must be the same length as ‘x‘.
x	A binary predictor vector (0/1 or logical). Must be the same length as ‘y‘.

Value

A numeric scalar giving the conditional probability that ‘y = 1‘ given ‘x = 1‘.

Examples

```
cprob(misimdata$y,misimdata$x1)
```

cprob_inv *cprob_inv*

Description

Computes the conditional probability $P(Y = 1 \mid X = 0)$ for two binary vectors ‘y‘ and ‘x‘. Rows with missing values in either vector are excluded.

Usage

```
cprob_inv(y, x)
```

Arguments

y	A binary outcome vector (0/1 or logical). Must be the same length as ‘x‘.
x	A binary predictor vector (0/1 or logical). Must be the same length as ‘y‘.

Value

A numeric scalar giving the conditional probability that ‘y = 1‘ given ‘x = 0‘.

Examples

```
cprob_inv(misimdata$y,misimdata$x1)
```

`entfuns` *entfuns*

Description

Computes a set of descriptive diagnostics for a binary outcome ‘y‘ against one or more predictors in ‘x‘, including marginal probability, conditional probability, absolute and proportional differences between marginal and conditional probabilities, and analogous measures based on . entropy.

Usage

```
entfuns(y, x)
```

Arguments

y	A binary outcome vector (0/1 or logical). Length ‘n‘.
x	A data frame of binary predictors (columns). Must have ‘n‘ rows; each column is analyzed separately against ‘y‘.

Details

Inputs are treated as binary (0/1 or logical). Missing values are removed pairwise for each predictor (rows with ‘NA‘ in either the outcome or the predictor are excluded for that predictor's calculations).

Value

A data frame with one row per predictor and the following columns:

xvar Predictor name.

yprob Marginal probability $P(Y = 1)$ computed on complete cases for that predictor.

xprob Marginal probability $P(X = 1)$.

cprob Conditional probability $P(Y = 1 \mid X = 1)$.

cpdif Absolute difference $P(Y = 1 \mid X = 1) - P(Y = 1)$.

cpdifper Percent difference relative to $P(Y = 1)$.

yent Entropy $H(Y)$.

ce Conditional entropy $H(Y \mid X)$.

cedif Absolute difference $H(Y) - H(Y \mid X)$.

cedifper Percent difference in entropy relative to $H(Y)$.

Examples

```
entfuns(misimdata$y,misimdata[,2:5])
```

`entropy` *entropy*

Description

Returns marginal entropy for binary variables

Usage

```
entropy(x)
```

Arguments

`x`	A binary vector (numeric coded as 0/1 or logical). Must be length >= 1.

Value

A numeric scalar giving the entropy of ‘x‘.

Examples

```
entropy(misimdata$x1)
```

`find_minimal_sets` *find_minimal_sets*

Description

Given a character vector of sets (each set encoded as variable names joined by a separator), returns the subset of sets that are minimal: no returned set is a strict superset of another. Duplicates and ordering differences are handled according to the implementation.

Usage

```
find_minimal_sets(str_vec, sep = "_")
```

Arguments

`str_vec`	Character vector of set strings for which to find minimally sufficient sets (e.g., ‘c("x1_x2", "x1_x2_x3")‘).
`sep`	Character string used as the separator between variables in each set. Defaults to ‘"_"‘.

Value

A character vector containing the minimally sufficient sets (i.e., sets that are not strict supersets of any other set in ‘str_vec‘).

Examples

```
pairmiresult = pairmi(misimdata[,2:6])
results_probstat <- probstat(misimdata$y,pairmiresult$expanded.data,nfolds=5)
find_minimal_sets(results_probstat$xvars[results_probstat$cprob >= 0.20])
```

`gstat` *gstat*

Description

Computes the likelihood-ratio test statistic (G statistic) from the mutual information and the joint count of two variables:

$$G = 2 \times n \times MI,$$

where n is the joint sample size and MI is the mutual information.

Usage

```
gstat(mi, count)
```

Arguments

`mi`	Numeric scalar; the mutual information between two variables.
`count`	Integer scalar; the joint count (sample size) used in computing `mi`.

Value

A numeric scalar giving the G statistic value.

Examples

```
gstat(mi(misimdata$y,misimdata$x1),jtct(misimdata$y,misimdata$x1))
```

`joint` *joint*

Description

Computes the joint probability $P(X = 1, Y = 1)$ for two binary vectors ‘x‘ and ‘y‘. Rows with missing values in either vector are excluded.

Usage

```
joint(y, x)
```

Arguments

y	A binary outcome vector (0/1 or logical). Must be the same length as ‘x‘.
x	A binary predictor vector (0/1 or logical). Must be the same length as ‘y‘.

Value

A numeric scalar giving the joint probability that both ‘x = 1‘ and ‘y = 1‘, calculated as the joint count divided by the number of complete cases.

Examples

```
joint(misimdata$y,misimdata$x1)
```

`jtct` *jtct*

Description

Counts the number of complete observations where both a binary outcome ‘y‘ and a binary predictor ‘x‘ equal 1. Missing values are excluded pairwise (rows with ‘NA‘ in either ‘x‘ or ‘y‘ are ignored).

Usage

```
jtct(y, x)
```

Arguments

y	Outcome vector (binary: 0/1 or logical). Must be the same length as ‘x‘.
x	Predictor vector (binary: 0/1 or logical). Must be the same length as ‘y‘.

Value

An integer scalar giving the number of observations where ‘x == 1‘ and ‘y == 1‘, after excluding missing values.

Examples

```
cprob_inv(misimdata$y,misimdata$x1)
```

mi *mi*

Description

Computes the mutual information (MI) between an outcome ‘y‘ and a predictor ‘x‘, using the standard definition:

$$MI(X,Y) = H(X) + H(Y) - H(X,Y),$$

Usage

```
mi(y, x)
```

Arguments

y	Outcome vector (binary: 0/1 or logical).
x	Predictor vector (binary: 0/1 or logical). Must be the same length as ‘y‘.

Value

A numeric scalar giving the mutual information between ‘x‘ and ‘y‘

Examples

```
mi(misimdata$y,misimdata$x1)
```

misimdata *misimdata*

Description

A data set with 10 predictors and 1 outcome that can be used to try out the functions of the setweaver package

Usage

```
misimdata
```

Format

A data frame with 2500 rows and 11 variables:

y Outcome

x1 First binary predictor

x2 Second binary predictor

x3 Third binary predictor

x4 Fourth binary predictor

x5 Fifth binary predictor

x6 Sixth binary predictor

x7 Seventh binary predictor

x8 Eighth binary predictor

x9 Ninth binary predictor

x10 Tenth binary predictor

`pairmi` *pairmi*

Description

A function that calculates the mutual information for sets of variables, calculates the G statistic, determines the significance of the sets, and only keeps those that are significant.

Usage

```
pairmi(data, alpha = 0.05, MI.threshold = NULL, n_elements = 5, sep = "_")
```

Arguments

`data`	A data frame containing the variables to be paired/combined. Columns should be binary.
`alpha`	Numeric p-value threshold for significance (default used by the implementation if not supplied).
`MI.threshold`	Numeric mutual information threshold. If provided, it overrides ‘alpha‘-based filtering.
`n_elements`	Integer giving the maximum size of sets to evaluate (e.g., ‘2‘ for pairs, ‘3‘ for triplets). Must be >= 2.
`sep`	String used to join variable names when forming set identifiers (e.g., ‘"_"‘).

Value

A list with the following components:

expanded.data A data frame containing the original variables and the columns for significant sets (e.g., pair/triplet indicators).

original.variables Character vector of the original variable names.

sets A data frame describing significant sets, including their members, size, MI, G statistic, p-value, and constructed name.

Examples

```
pairmi(misimdata[,2:6])
```

`plot_prob` *plot_prob*

Description

Creates a network-style graph showing how a set of predictors (‘x_vars‘) are related to an outcome (‘y_var‘). Relationships can be displayed either as conditional probabilities or as effects estimated by logistic regression.

Usage

```
plot_prob(
  data,
  y_var,
  x_vars,
  var_labels = NULL,
  prob_digits = 2,
  method = "conditional",
  title = NULL,
  vertex_color = "lightblue",
  vertex_frame_color = "darkblue",
  vertex_label_color = "black",
  edge_color = "darkgrey",
  edge_label_color = "black",
  min_arrow_width = 1,
  max_arrow_width = 10,
  node_size = 45,
  label_cex = 0.8
)
```

Arguments

data	A data frame containing the outcome (‘y_var‘) and predictors (‘x_vars‘).
y_var	Character string giving the name of the outcome variable in ‘data‘.
x_vars	Character vector of predictor variable names in ‘data‘.
var_labels	Optional character vector of display labels for the predictors. Must match the length of ‘x_vars‘.
prob_digits	Integer; number of decimal places to round conditional probabilities. Defaults to ‘2‘.
method	Character string indicating how to quantify associations: ‘"prob"‘ for conditional probabilities or ‘"logistic"‘ for logistic regression effects.
title	Character string; title of the plot.
vertex_color	Character string giving the fill color of nodes.
vertex_frame_color	Character string giving the color of node borders.
vertex_label_color	Character string giving the color of node labels.
edge_color	Character string giving the color of edges.
edge_label_color	Character string giving the color of edge labels.
min_arrow_width	Numeric value for the minimum edge width.
max_arrow_width	Numeric value for the maximum edge width.
node_size	Numeric value controlling the size of nodes.
label_cex	Numeric value controlling the size of node labels.

Value

A graph object (typically an [‘igraph::igraph‘] object or similar) is returned and plotted. Nodes represent variables and edges represent associations. Node labels include variable names and marginal probabilities. Edge labels display either conditional probabilities or logistic regression effects.

Examples

```
plot_prob(misimdata,'y',colnames(misimdata[,3:6]),method='logistic')
```

prob *prob*

Description

Computes the marginal probability $P(X = 1)$ for a binary vector ‘x‘, ignoring missing values.

Usage

```
prob(x)
```

Arguments

x	A numeric or logical vector coded as 0/1 (or ‘FALSE‘/‘TRUE‘). Values other than 0, 1, ‘FALSE‘, ‘TRUE‘, or ‘NA‘ will be ignored.

Value

A numeric scalar giving the proportion of entries equal to 1 among the non-missing values of ‘x‘.

Examples

```
prob(c(0, 1, 1, 0, 1))
```

probstat *probstat*

Description

Computes marginal, conditional, and information-theoretic summaries for a binary outcome ‘y‘ against one or more predictors in ‘x‘. Performs either Fisher’s exact test or a generalized linear mixed model (GLMM) for inference.

Usage

```
probstat(y, x, test = "Fisher", ri, nfolds, seed = 10101)
```

Arguments

y	A binary outcome vector (logical or numeric coded as 0/1). Length ‘n‘.
x	A data frame of predictors (typically the expanded data returned by [pairmi()]). Must have ‘n‘ rows; columns are treated as candidate predictors.
test	Character string selecting the inferential method; one of ‘c("fisher", "glmm")‘. Defaults to ‘"fisher"‘ if missing.

`ri`	Optional vector/factor giving the grouping variable for a random intercept in the GLMM. Must be length ‘n‘. Ignored if ‘test = "fisher"‘.
`nfolds`	Integer; number of folds used for cross-validation.
`seed`	Integer seed for fold randomization.

Value

A data frame with one row per evaluated predictor (or pair) and the following columns:

xprob Marginal probability of $X = 1$.

yprob Marginal probability of $Y = 1$.

cprob Conditional probability $P(Y = 1 \mid X = 1)$.

cprobx Conditional probability $P(X = 1 \mid Y = 1)$.

cprobi Inverse conditional probability $P(Y = 1 \mid X = 0)$.

cpdif Difference $P(Y = 1 \mid X = 1) - P(Y = 1)$.

cpdifper Percent difference relative to $P(Y = 1)$.

xent Entropy of X.

yent Entropy of Y.

ce Conditional entropy of $Y \mid X$.

cedif Difference between marginal and conditional entropy of Y.

cedifper Percent difference in entropy.

p p-value from Fisher’s exact test or the GLMM (as applicable).

Examples

```
pairmiresult = pairmi(misimdata[,2:6])
probstat(misimdata$y,pairmiresult$expanded.data,nfolds=5)
```

`setmapmi` *setmapmi*

Description

Creates a set map visualization from the output of [pairmi()], showing which original variables compose the derived sets at a specified depth.

Usage

```
setmapmi(original_variables = NULL, sets = NULL, n_elements = NULL)
```

Arguments

original_variables	Character vector of names for the original variables that were paired (typically 'pairmi_result$original.variables').
sets	A data frame returned by [pairmi()] describing the sets. Must contain the columns required by 'setmapmi()' (e.g., identifiers for sets and their constituent variables).
n_elements	Integer scalar giving the set size (depth) to visualize (e.g., '2' for pairs, '3' for triplets). Must be >= 1 and present in 'sets'.

Value

A setmap showing which original variables make up the sets at a certain depth

Examples

```
pairmiresult = pairmi(misimdata[,2:6])
setmapmi(pairmiresult$original.variables,pairmiresult$sets,2)
```

---

zprob *zprob*

Description

Computes the z-score for testing whether the proportion (probability) of successes in 'x' differs from zero.

Usage

```
zprob(x)
```

Arguments

x	A numeric or logical vector representing binary outcomes (e.g., 0/1 or TRUE/FALSE), from which the proportion is calculated.

Value

A numeric value giving the z-score for the observed proportion.

Examples

```
zprob(misimdata$x1)
```

Index