Package 'RulesTools'

July 21, 2025

Title Preparing, Analyzing, and Visualizing Association Rules

Version 0.1.1

Description Streamlines data preprocessing, analysis, and visualization for association rule mining. Designed to work with the 'arules' package, features include discretizing data frames, generating rule set intersections, and visualizing rules with heatmaps and Euler diagrams. 'RulesTools' also includes a dataset on Brook trout detection from Nolan et al. (2022) <doi:10.1007/s13412-022-00800-x>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports mice, arules, ggplot2, tidyr, eulerr, magrittr

Suggests testthat (>= 3.0.0), knitr, rmarkdown, roxygen2 (>= 7.1.0)

RoxygenNote 7.3.2

URL https://github.com/nikolett0203/RulesTools

BugReports https://github.com/nikolett0203/RulesTools/issues

NeedsCompilation no

Author Nikolett Toth [aut, cre],
Jarrett Phillips [ctb]

Maintainer Nikolett Toth <rules.tools.pkg@gmail.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2025-01-28 15:50:06 UTC

Contents

`BrookTrout` *Brook Trout eDNA and Environmental Data*

Description

This dataset contains information on brook trout detections using environmental DNA (eDNA) and environmental parameters collected from various sites in Ontario, Canada. The data was sourced from a scientific study comparing eDNA sampling methods with electrofishing to detect Brook trout populations.

Usage

```
BrookTrout
```

Format

A dataframe with 10 variables and multiple rows (one row per sample):

Backpack Character. The type of eDNA sampler: "OSMOS" or "ANDe".

Site Integer. The site number where the sample was taken.

eFishCatch Integer. The number of fish caught via electrofishing.

AirTemp Numeric. Air temperature in degrees Celsius.

WaterTemp Numeric. Water temperature in degrees Celsius.

pH Numeric. pH level of the water sample.

DissolvedOxygen Numeric. Dissolved oxygen concentration in mg/L.

Conductivity Numeric. Conductivity in uS/cm.

VolumeFiltered Numeric. Volume of water filtered in litres.

eDNAConc Numeric. eDNA concentration in copies per microlitre.

Source

Adapted from Nolan, K. P., Loeza-Quintana, T., Little, H. A., et al. (2022). Detection of brook trout in spatiotemporally separate locations using validated eDNA technology. *Journal of Environmental Studies and Sciences*, 13, 66-82. doi:10.1007/s13412-022-00800-x

Examples

```
data(BrookTrout)
summary(BrookTrout)
plot(eDNAConc ~ Site, data = BrookTrout)
```

compare_rules *Compare Association Rule Sets and Find Their Intersections*

Description

Compares multiple sets of association rules, identifies intersections, and optionally displays the results or writes them to a CSV file.

Usage

```
compare_rules(..., display = TRUE, filename = NULL)
```

Arguments

`...`	Named association rule sets (objects of class `rules`).
`display`	Logical. If `TRUE`, prints the intersection results. Default is `TRUE`.
`filename`	Character string. If provided, writes the results to a CSV file. Default is `NULL`.

Value

A list containing the intersections of the provided rule sets.

Examples

```
library(arules)
data(BrookTrout)

# Discretize the BrookTrout dataset
discrete_bt <- dtize_df(BrookTrout, cutoff = "mean")

# Generate the first set of rules with a confidence threshold of 0.5
rules1 <- apriori(
  discrete_bt,
  parameter = list(supp = 0.01, conf = 0.5, target = "rules")
)

# Generate the second set of rules with a higher confidence threshold of 0.6
rules2 <- apriori(
  discrete_bt,
  parameter = list(supp = 0.01, conf = 0.6, target = "rules")
)

# Compare the two sets of rules and display the intersections
compare_rules(
  r1 = rules1,
  r2 = rules2,
  display = TRUE
)
```

```
# If `filename = "intersections.csv"`, the data is saved in a .csv file
```

`dtize_col` *Discretize a Numeric Column*

Description

Discretizes a numeric vector into categories based on specified cutoff points. The function handles missing values, allows for infinite bounds, and supports predefined cutoffs such as the mean or median.

Usage

```
dtize_col(
  column,
  cutoff = "median",
  labels = c("low", "high"),
  include_right = TRUE,
  infinity = TRUE,
  include_lowest = TRUE,
  na_fill = "none"
)
```

Arguments

`column`	A numeric vector to discretize.
`cutoff`	A numeric vector specifying cutoff points, or a string ("mean" or "median").
`labels`	A character vector specifying labels for the resulting categories.
`include_right`	Logical. If `TRUE`, intervals are closed on the right (default `TRUE`).
`infinity`	Logical. If `TRUE`, extends cutoffs to `-Inf` and `Inf` (default `TRUE`).
`include_lowest`	Logical. If `TRUE`, the lowest interval is closed on the left (default `TRUE`).
`na_fill`	A string specifying the method to impute missing values: "none", "mean", or "median" (default "none").

Value

A factor with the same length as `column`, where each value is categorized based on the cutoffs.

Examples

```
data(BrookTrout)

# Example with predefined cutoffs
discrete_water_temp <- dtize_col(
  BrookTrout$eDNAConc, cutoff=13.3,
```

```
  labels=c("low", "high"),
  infinity=TRUE
)

# Example with median as cutoff
discrete_pH <- dtize_col(BrookTrout$pH, cutoff="median")

# Example with missing value imputation
filled_col <- dtize_col(
  c(1, 2, NA, 4, 5),
  cutoff = "mean",
  include_right=FALSE,
  na_fill = "mean"
)
```

`dtize_df` *Discretize Dataframe Columns*

Description

Discretizes numeric columns of a dataframe based on specified splitting criteria, and handles missing values using specified imputation methods.

Usage

```
dtize_df(
  data,
  cutoff = "median",
  labels = c("low", "high"),
  include_right = TRUE,
  infinity = TRUE,
  include_lowest = TRUE,
  na_fill = "none",
  m = 5,
  maxit = 5,
  seed = NULL,
  printFlag = FALSE
)
```

Arguments

`data`	A dataframe containing the data to be discretized.
`cutoff`	A character string specifying the splitting method for numeric columns. Options are `"median"` (default), `"mean"` or a custom numeric vector of split points.
`labels`	A character vector of labels for the discretized categories. Default is `c("low", "high")`.

include_right	A logical value indicating if the intervals should be closed on the right. Default is TRUE.
infinity	A logical value indicating if the split intervals should extend to infinity. Default is TRUE.
include_lowest	A logical value indicating if the lowest value should be included in the first interval. Default is TRUE.
na_fill	A character string specifying the imputation method for handling missing values. Options are "none" (default), "mean", "median", or "pmm" (predictive mean matching).
m	An integer specifying the number of multiple imputations if na_fill = "pmm". Default is 5.
maxit	An integer specifying the maximum number of iterations for the mice algorithm. Default is 5.
seed	An integer seed for reproducibility of the imputation process. Default is NULL.
printFlag	A logical value indicating if mice should print logs during imputation. Default is FALSE.

Value

A dataframe with numeric columns discretized and missing values handled based on the specified imputation method.

Examples

```
data(BrookTrout)

# Example with median as cutoff
med_df <- dtize_df(
  BrookTrout,
  cutoff="median",
  labels=c("below median", "above median")
)

# Example with mean as cutoff
mean_df <- dtize_df(
  BrookTrout,
  cutoff="mean",
  include_right=FALSE
)

# Example with missing value imputation
air <- dtize_df(
  airquality,
  cutoff="mean",
  na_fill="pmm",
  m=10,
  maxit=10,
  seed=42
)
```

rule_euler *Create an Euler Diagram for Association Rules*

Description

Generates an Euler diagram visualization for up to 4 sets of association rules. The function displays the relationships between rule sets with customizable colors, transparency, and labels.

Usage

```
rule_euler(
  rules,
  fill_color = NULL,
  fill_alpha = 0.5,
  stroke_color = "black",
  stroke_size = 1,
  title = NULL,
  name_color = "black",
  name_size = 12,
  text_color = "black",
  text_size = 11,
  show_legend = FALSE,
  legend_position = "bottom",
  nrow = NULL,
  ncol = NULL
)
```

Arguments

rules	A list of rules objects from the arules package. The list must contain between 2 and 4 rules objects.
fill_color	A character vector of valid R color names or hex color codes for filling the sets. If NULL, default colors c("red", "blue", "green", "purple") will be used. Defaults to NULL.
fill_alpha	A numeric value between 0 and 1 specifying the transparency of the fill colors. Defaults to 0.5.
stroke_color	A character string specifying the color of the set borders. Defaults to "black".
stroke_size	A positive numeric value specifying the size of the set borders. Defaults to 1.
title	A character string specifying the title of the Euler diagram. Defaults to NULL.
name_color	A character string specifying the color of the set names. Defaults to "black".
name_size	A positive numeric value specifying the font size of the set names. Defaults to 12.

`text_color` A character string specifying the color of the quantity labels (counts) in the diagram. Defaults to `"black"`.

`text_size` A positive numeric value specifying the font size of the quantities (counts). Defaults to 11.

`show_legend` A logical value indicating whether to display a legend for the sets rather than labels. Defaults to `FALSE`.

`legend_position` A character string specifying the position of the legend. Must be one of `"top"`, `"bottom"`, `"left"`, or `"right"`. Defaults to `"bottom"`.

`nrow` An optional numeric value specifying the number of rows in the legend layout. If `NULL`, the number of rows is calculated automatically. Defaults to `NULL`.

`ncol` An optional numeric value specifying the number of columns in the legend layout. If `NULL`, the number of columns is calculated automatically. Defaults to `NULL`.

Value

A `plot` object displaying the Euler diagram visualization.

Examples

```
library(arules)
data(BrookTrout)

# Discretize the BrookTrout dataset
discrete_bt <- dtize_df(BrookTrout, cutoff = "median")

# Generate the first set of rules with a confidence threshold of 0.5
rules1 <- apriori(
  discrete_bt,
  parameter = list(supp = 0.01, conf = 0.5, target = "rules")
)

# Generate the second set of rules with a higher confidence threshold of 0.6
rules2 <- apriori(
  discrete_bt,
  parameter = list(supp = 0.01, conf = 0.6, target = "rules")
)

# Create an Euler diagram to visualize the intersections between the rule sets
rule_euler(
  rules = list(conf0.5 = rules1, conf0.6 = rules2),
  title = "Euler Diagram of BrookTrout Rule Sets",
  fill_color = c("#7832ff", "lightgreen"),
  stroke_color = "darkblue"
)
```

rule_heatmap *Create a Heatmap for Association Rules*

Description

Generates a heatmap visualization of association rules, showing relationships between antecedents and consequents based on a specified metric.

Usage

```
rule_heatmap(
  rules,
  metric = "confidence",
  graph_title = "",
  graph_title_size = 14,
  x_axis_title = "Antecedents",
  x_axis_title_size = 12,
  x_axis_text_size = 11,
  x_axis_text_angle = 45,
  y_axis_title = "Consequents",
  y_axis_title_size = 12,
  y_axis_text_size = 11,
  y_axis_text_angle = 0,
  legend_title = metric,
  legend_text_size = 8,
  legend_position = "right",
  low_color = "lightblue",
  high_color = "navy",
  include_zero = FALSE
)
```

Arguments

rules	An object of class rules from the arules package.
metric	A character string specifying the metric to use for coloring the heatmap. Must be one of "confidence", "support", or "lift". Defaults to "confidence".
graph_title	A character string specifying the title of the graph. Defaults to an empty string ("").
graph_title_size	A numeric value specifying the size of the graph title text. Defaults to 14.
x_axis_title	A character string specifying the title for the x-axis. Defaults to "Antecedents".
x_axis_title_size	A numeric value specifying the size of the x-axis title text. Defaults to 12.
x_axis_text_size	A numeric value specifying the size of the x-axis text. Defaults to 11.

x_axis_text_angle
: A numeric value specifying the angle of the x-axis text. Defaults to 45.

y_axis_title
: A character string specifying the title for the y-axis. Defaults to "Consequents".

y_axis_title_size
: A numeric value specifying the size of the y-axis title text. Defaults to 12.

y_axis_text_size
: A numeric value specifying the size of the y-axis text. Defaults to 11.

y_axis_text_angle
: A numeric value specifying the angle of the y-axis text. Defaults to 0.

legend_title
: A character string specifying the title of the legend. Defaults to the value of metric.

legend_text_size
: A numeric value specifying the size of the legend text. Defaults to 8.

legend_position
: A character string specifying the position of the legend. Possible values are "right" (default), "left", "top", "bottom", or "none".

low_color
: A valid R color or hex color code for the lower bound of the gradient. Defaults to "lightblue".

high_color
: A valid R color or hex color code for the upper bound of the gradient. Defaults to "navy".

include_zero
: A logical value indicating whether to include zero values for missing antecedent-consequent combinations. Defaults to FALSE.

Value

A ggplot object representing the heatmap visualization of the association rules.

Examples

```
library(arules)
library(tidyr)
data(BrookTrout)

# Discretise data
discrete_bt <- dtize_df(BrookTrout, cutoff="median")

# Generate rules
rules <- apriori(
  discrete_bt,
  parameter = list(supp = 0.01, conf = 0.5, target = "rules"),
  appearance = list(rhs="eDNAConc=high")
)

# Subset ruleset (too many rules won't fit on the heatmap)
rules <- rules %>%
  subset(!is.redundant(., measure = "confidence")) %>%
  subset(is.significant(., alpha = 0.05)) %>%
  sort(by = c("confidence", "lift", "support"))
```

```
# Create a heatmap of the rules using confidence as the metric
rule_heatmap(
  rules,
  metric = "confidence",
  graph_title = "Confidence Heatmap"
)

# Create a heatmap of the rules using lift as the metric
rule_heatmap(
  rules,
  metric = "lift",
  graph_title = "Lift Heatmap",
  low_color = "#D4A221",
  high_color = "darkgreen"
)
```

Index