Package ‘vtree’

May 15, 2019

Type Package

Title Display Information About Nested Subsets of a Data Frame

Version 2.0.0

Depends R (>= 2.10)

Author Nick Barrowman

Maintainer Nick Barrowman <nbarrowman@cheo.on.ca>

Description A tool for calculating and drawing ``variable trees''. Variable trees display information about hierarchical subsets of a data frame defined by values of categorical variables.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

VignetteBuilder knitr

Suggests knitr, rmarkdown, ggplot2, testthat (>= 2.1.0)

Imports DiagrammeR, DiagrammeRsvg, rsvg

NeedsCompilation no

Repository CRAN

Date/Publication 2019-05-15 20:10:04 UTC

R topics documented:

crosstabToCases *crosstabToCases*

Description

Convert a crosstabulation into a data frame of cases.

Usage

```
crosstabToCases(x)
```

Arguments

x	a matrix or table of frequencies representing a crosstabulation.

Value

Returns a data frame of cases.

Author(s)

Nick Barrowman, based on the countsToCases function at http://www.cookbook-r.com/Manipulating_data/Converting_between_data_frames_and_contingency_tables/#countstocases-function

Examples

```
# The Titanic data set is in the datasets package.
# Convert it from a 4 x 2 x 2 x 2 crosstabulation
# to a 4-column data frame of 2201 individuals
titanic <- crosstabToCases(Titanic)
```

FakeData *Fake Clinical Dataset*

Description

A dataset consisting of made-up clinical data. Note that some observations are missing (i.e. NAs).

Usage

```
FakeData
```

Format

A small data frame in which the rows represent (imaginary) patients and the columns represent variables of possible clinical relevance.

id Integer: Patient ID number

Group Factor: Treatment Group, A or B

Severity Factor representing severity of condition: Mild, Moderate, or Severe

Sex Factor: M or F

Male Integer: Sex coded as 1=M, 0=F

Age Integer: Age in years, continuous

Score Integer: Score on a test

Category Factor: single, double, or triple

Pre Numeric: initial measurement

Post Numeric: measurement taken after something happened

Post2 Numeric: measurement taken at the very end of the study

Time Numeric: time to event, or time of censoring

Event Integer: Did the event occur? 1=yes, 0=no (i.e. censoring)

Ind1 Integer: Indicator variable for a certain characteristic, 1=present, 0=absent

Ind2 Integer: Indicator variable for a certain characteristic, 1=present, 0=absent

Ind3 Integer: Indicator variable for a certain characteristic, 1=present, 0=absent

Ind4 Integer: Indicator variable for a certain characteristic, 1=present, 0=absent

Viral Logical: Does this patient have a viral illness?

`FakeRCT` *Fake Randomized Controlled Trial (RCT) Data*

Description

A dataset consisting of made-up RCT data.

Usage

```
FakeRCT
```

Format

A small data frame in which the rows represent (imaginary) patients and the columns represent variables of possible clinical relevance.

id String: Patient ID number

eligible Factor: Eligible or Ineligible

randomized Factor: Randomized or Not randomized

group Factor: A or B

followup Factor: Followed up or Not followed up

analyzed Factor: Analyzed or Not analyzed

`grVizToPNG` *grVizToPNG*

Description

`grVizToPNG` Export a grViz object into a PNG file.

Usage

```
grVizToPNG(g, width = NULL, height = NULL, folder = ".")
```

Arguments

`g`	an object produced by the grViz function from the DiagrammmeR package
`width`	the width in pixels of the bitmap
`height`	the height in pixels of the bitmap
`folder`	path to folder where the PNG file should stored

Details

First the grViz object is exported to an SVG file (using `DiagrammeRsvg::export_svg`). Then the SVG file is converted to a bitmap (using `rsvg::rsvg`). Then the bitmap is exported as a PNG file (using `png::writePNG`). Note that the SVG file and the PNG file will be named using the name of the `g` parameter

Value

Returns the full path of the PNG file.

Note

In addition to the DiagrammmeR package, the following packages are used: `DiagrammeRsvg`, `rsvg`

Author(s)

Nick Barrowman

vtree *vtree: Draw a variable tree*

Description

vtree is a tool for drawing variable trees. Variable trees display information about nested subsets of a data frame, in which the subsetting is defined by the values of categorical variables.

Usage

```
vtree(z, vars, splitspaces = TRUE, prune = list(),
  prunebelow = list(), keep = list(), follow = list(),
  prunelone = NULL, pruneNA = FALSE, labelnode = list(),
  tlabelnode = NULL, labelvar = NULL, varminwidth = NULL,
  varminheight = NULL, varlabelloc = NULL, fillcolor = NULL,
  fillnodes = TRUE, NAfillcolor = "white", rootfillcolor = "#EFF3FF",
  palette = NULL, gradient = TRUE, revgradient = FALSE,
  singlecolor = 2, colorvarlabels = TRUE, title = "",
  sameline = FALSE, Venn = FALSE, check.is.na = FALSE, seq = FALSE,
  pattern = FALSE, ptable = FALSE, showroot = TRUE, text = list(),
  ttext = list(), plain = FALSE, squeeze = 1,
  shownodelabels = TRUE, showvarnames = TRUE, showlevels = TRUE,
  showpct = TRUE, showlpct = TRUE, showcount = TRUE,
  showlegend = FALSE, varnamepointsize = 18, HTMLtext = FALSE,
  digits = 0, cdigits = 1, splitwidth = 20, lsplitwidth = 15,
  getscript = FALSE, nodesep = 0.5, ranksep = 0.5, margin = 0.2,
  vp = TRUE, horiz = TRUE, summary = "", runsummary = NULL,
  retain = NULL, width = NULL, height = NULL, graphattr = "",
  nodeattr = "", edgeattr = "", color = c("blue", "forestgreen",
  "red", "orange", "pink"), colornodes = FALSE, showempty = FALSE,
  rounded = TRUE, nodefunc = NULL, nodeargs = NULL,
  choicechecklist = TRUE, parent = 1, last = 1, root = TRUE)
```

Arguments

z	Required: Data frame, or a single vector.
vars	Required (unless z is a vector): Either a character string of whitespace-separated variable names or a vector of variable names.
splitspaces	When vars is a character string, split it by spaces to get variable names? It is only rarely necessary to use this parameter. This should only be FALSE when a single variable name that contains spaces is specified.
prune	List of vectors that specifies nodes to prune. The name of each element of the list must be one of the variable names in vars. Each element is a vector of character strings that identifies the values of the variable (i.e. the nodes) to prune.
prunebelow	Like prune but instead of pruning the specified nodes, their descendants are pruned.

`keep`	Like `prune` but specifies which nodes to *keep*. The other nodes will be pruned.
`follow`	Like `keep` but specifies which nodes to "follow", i.e. which nodes' *descendants* to keep.
`prunelone`	A vector of values specifying "lone nodes" (of *any* variable) to prune. A lone node is a node that has no siblings.
`pruneNA`	Prune all missing values? This should be used carefully because "valid" percentages are hard to interpret when NAs are pruned.
`labelnode`	List of vectors used to change how values of variables are displayed. The name of each element of the list is one of the variable names in `vars`. Each element of the list is a vector of character strings, representing the values of the variable. The names of the vector represent the labels to be used in place of the values.
`tlabelnode`	A list of vectors, each of which specifies a particular node, as well as a label for that node (a "targeted" label). The names of each vector specify variable names, except for an element named `label`, which specifies the label to use.
`labelvar`	A named vector of labels for variables.
`varminwidth`	A named vector of minimum initial widths for nodes of each variable. (Sets the Graphviz `width` attribute.)
`varminheight`	A named vector of minimum initial heights for nodes of each variable. (Sets the Graphviz `height` attribute.)
`varlabelloc`	A named vector of vertical label locations ("t", "c", or "b" for top, center, or bottom, respectively) for nodes of each variable. (Sets the Graphviz `labelloc` attribute.)
`fillcolor`	A named vector of colors for filling the nodes of each variable. If an unnamed, scalar color is specified, all nodes will have this color.
`fillnodes`	Fill the nodes with color?
`NAfillcolor`	Fill-color for missing-value nodes. If `NULL`, fill colors of missing value nodes will be consistent with the fill colors in the rest of the tree.
`rootfillcolor`	Fill-color for the root node.
`palette`	A vector of palette numbers (which can range between 1 and 9). The names of the vector indicate the corresponding variable. See **Palettes** below for more information.
`gradient`	Use gradients of fill color across the values of each variable? A single value (with no names) specifies the setting for all variables. A logical vector of `TRUE` values for named variables is interpreted as `TRUE` for those variables and `FALSE` for all others. A logical vector of `FALSE` values for named variables is interpreted as `FALSE` for those variables and `TRUE` for all others.
`revgradient`	Should the gradient be reversed (i.e. dark to light instead of light to dark)? A single value (with no names) specifies the setting for all variables. A logical vector of `TRUE` values for named variables is interpreted as A logical vector of `FALSE` values for named variables is interpreted as `FALSE` for those variables and `TRUE` for all others.
`singlecolor`	When a variable has a single value, this parameter is used to specify whether nodes should have a (1) light shade, (2) a medium shade, or (3) a dark shade. specify `singlecolor=1` to assign a light shade.

`colorvarlabels`	Color the variable labels?
`title`	Optional title for the root node of the tree.
`sameline`	Display node labels on the same line as the count and percentage?
`Venn`	Display multi-way set membership information? This provides an alternative to a Venn diagram. This sets `showpct=FALSE` and `shownodelabels=FALSE`. Assumption: all of the specified variables are logicals or 0/1 numeric variables.
`check.is.na`	Replace each variable named in `vars` with a logical vector indicating whether or not each of its values is missing?
`seq`	Display the variable tree using "sequences"? Each unique sequence (i.e. pattern) of values will be shown separately. The sequences are sorted from least frequent to most frequent.
`pattern`	Same as `seq`, but lines without arrows are drawn, and instead of a `sequence` variable, a `pattern` variable is shown.
`ptable`	Generate a pattern table instead of a variable tree? Only applies when `pattern=TRUE`.
`showroot`	Show the root node? When `seq=TRUE`, it may be useful to set `showroot=FALSE`.
`text`	A list of vectors containing extra text to add to nodes corresponding to specified values of a specified variable. The name of each element of the list must be one of the variable names in `vars`. Each element is a vector of character strings. The names of the vector identify the nodes to which the text should be added. (See **Formatting codes** below for information on how to format text.)
`ttext`	A list of vectors, each of which specifies a particular node, as well as text to add to that node ("targeted" text). The names of each vector specify variable names, except for an element named `text`, which specifies the text to add.
`plain`	Use "plain" settings? These settings are as follows: for each variable all nodes are the same color, namely a shade of blue (with each successive variable using a darker shade); all variable labels are black; and the `squeeze` parameter is set to 0.6.
`squeeze`	The degree (between 0 and 1) to which the tree will be "squeezed". This controls two Graphviz parameters: `margin` and `nodesep`.
`shownodelabels`	Show node labels? A single value (with no names) specifies the setting for all variables. A logical vector of `TRUE` values for named variables is interpreted as `TRUE` for those variables and `FALSE` for all others. A logical vector of `FALSE` values for named variables is interpreted as `FALSE` for those variables and `TRUE` for all others.
`showvarnames`	Show the name of the variable next to each level of the tree?
`showlevels`	(Deprecated) Same as showvarnames.
`showpct`	Show percentage in each node? A single value (with no names) specifies the setting for all variables. A logical vector of `TRUE` for named variables is interpreted as A logical vector of `FALSE` for named variables is interpreted as `FALSE` for those variables and TRUE for all others.
`showlpct`	Show percentages (for the marginal frequencies) in the legend?
`showcount`	Show count in each node? A single value (with no names) specifies the setting for all variables. A logical vector of `TRUE` for named variables is interpreted as A logical vector of `FALSE` for named variables is interpreted as `FALSE` for those variables and `TRUE` for all others.

`showlegend`	Show legend (including marginal frequencies) for each variable?
`varnamepointsize`	Font size (in points) to use when displaying variable names.
`HTMLtext`	Is the text formatted in HTML?
`digits`	Number of decimal digits to show in percentages.
`cdigits`	Number of decimal digits to show in continuous values displayed via the summary parameter.
`splitwidth`	The minimum number of characters before an automatic linebreak is inserted.
`lsplitwidth`	In legends, the minimum number of characters before an automatic linebreak is inserted.
`getscript`	Instead of displaying the variable tree, return the DOT script as a character string?
`nodesep`	Graphviz attribute: Node separation amount.
`ranksep`	Graphviz attribute: Rank separation amount.
`margin`	Graphviz attribute: node margin.
`vp`	Use "valid percentages"? Valid percentages are computed by first excluding any missing values, i.e. restricting attention to the set of "valid" observations. The denominator is thus the number of non-missing observations. When `vp=TRUE`, nodes for missing values show the number of missing values but do not show a percentage; all the other nodes how valid percentages. When `vp=FALSE`, all nodes (including nodes for missing values) show percentages of the total number of observations.
`horiz`	Should the tree be drawn horizontally? (i.e. parent node on the left, with the tree growing to the right)
`summary`	A character string used to specify summary statistics to display in the nodes. The first word in the character string is the name of the variable to be summarized. The rest of the character string is the text that will be displayed, along with special codes specifying the information to display (see **Summary codes** below). A vector of character strings can also be specified, if more than one variable is to be summarized.
`runsummary`	A list of functions, with the same length as `summary`. Each function must take a data frame as its sole argument, and return a logical value. Each string in `summary` will only be interpreted if the corresponding logical value is `TRUE`. the corresponding string in `summary` will be evaluated.
`retain`	Vector of names of additional variables in the data frame that need to be available to execute the functions in `runsummary`.
`width`	Width (in pixels) to be passed to `DiagrammeR::grViz`.
`height`	Height (in pixels) to be passed to `DiagrammeR::grViz`.
`graphattr`	Character string: Additional attributes for the Graphviz graph.
`nodeattr`	Character string: Additional attributes for Graphviz nodes.
`edgeattr`	Character string: Additional attributes for Graphviz edges.
`color`	A vector of color names for the *outline* of the nodes at each level.

`colornodes`	Color the node outlines?
`showempty`	Show nodes that do not contain any observations?
`rounded`	Use rounded boxes for nodes?
`nodefunc`	A node function (see **Node functions** below).
`nodeargs`	A list containing named arguments for the node function specified by `nodefunc`.
`choicechecklist`	When REDCap checklists are specified using the `stem:` syntax, automatically extract the names of choices and use them as variable names?
`parent`	Parent node number (Internal use only.)
`last`	Last node number (Internal use only.)
`root`	Is this the root node of the tree? (Internal use only.)

Value

If `getscript=TRUE`, returns a character string of DOT script that describes the variable tree. If `getscript=FALSE`, returns an object of class `htmlwidget` that will intelligently print itself into HTML in a variety of contexts including the R console, within R Markdown documents, and within Shiny output bindings.

Summary codes

- `%mean%` mean
- `%SD%` standard deviation
- `%min%` minimum
- `%max%` maximum
- `%pX%` Xth percentile, e.g. p50 means the 50th percentile
- `%median%` median, i.e. p50
- `%IQR%` interquartile range, i.e. p25, p75
- `%npct%` number and percentage of `TRUE` values
- `%list%` list of the individual values
- `%mv%` the number of missing values
- `%v%` the name of the variable
- `%noroot%` flag: Do not show summary in the root node.
- `%leafonly%` flag: Only show summary in leaf nodes.
- `%var=V%` flag: Only show summary in nodes of variable V.
- `%node=N%` flag: Only show summary in nodes with value N.
- `%trunc=n%` flag: Truncate the summary to the first n characters.

Node functions

Node functions provide a mechanism for running a function within each subset representing a node of the tree. The `summary` parameter uses node functions. A node functions is a function takes as arguments a data frame subset, the name of the subsetting variable, the value of the subsetting variable, and a list of named arguments.

Formatting codes

Formatting codes for the `text` argument. Also used by `labelnode` and `labelvar`.

- `\n` line break
- `*...*` italics
- `**...**` bold
- `^...^` superscript (using 10 point font)
- `~...~` subscript (using 10 point font)
- `%%red ...%%` display text in red (or whichever color is specified)

Palettes

Sequential palettes from Color Brewer:

1. Reds
2. Blues
3. Greens
4. Oranges
5. Purples
6. YlGn
7. PuBu
8. PuRd
9. YlOrBr

Author(s)

Nick Barrowman <nbarrowman@cheo.on.ca>

Examples

```
# A single-level hierarchy
vtree(FakeData,"Severity")

# A two-level hierarchy
vtree(FakeData,"Severity Sex")

# A two-level hierarchy with pruning of some values of Severity
vtree(FakeData,"Severity Sex",prune=list("Severity"=c("Moderate","NA")))

# Rename some nodes
vtree(FakeData,"Severity Sex",labelnode=list(Sex=(c("Male"="M","Female"="F"))))

# Rename a variable
vtree(FakeData,"Severity Sex",labelvar=c(Severity="How bad?"))

# Show legend. Put labels on the same line as counts and percentages
vtree(FakeData,"Severity Sex Viral",sameline=TRUE,showlegend=TRUE)
```

```
# Using the summary parameter to list ID numbers (truncated to 40 characters) in specified nodes
vtree(FakeData,"Severity Sex",summary="id \nid = %list% %var=Severity% %trunc=40%")

# Adding text to specified nodes of a tree
vtree(FakeData,"Severity Sex",ttext=list(
  c(Severity="Severe",Sex="M",text="\nMales with Severe disease"),
  c(Severity="NA",text="\nUnknown severity")))
```

Index