Package: vcdinteractive (via r-universe)

November 26, 2026

Type Package

Title Interactive Displays for Visualizing Categorical Data

Version 0.1.0

Description Interactive versions of the displays of the 'vcd' and 'vcdExtra' packages, rendered as 'htmlwidgets'. Mosaic, association, sieve and doubledecker plots share one widget, in which variables can be reordered, added or removed and split directions changed on the fly, tiles report observed and expected counts on hover, several kinds of residuals (Pearson, adjusted standardized, deviance, Freeman-Tukey, Anscombe and residuals from a robust fit) can be used for shading, and log-linear models can be compared. Cells that depart significantly from the model are listed with multiplicity-adjusted p-values. Further widgets provide pairwise mosaic matrices with linked highlighting, fourfold displays with log odds ratio plots, rootograms for discrete distributions, spinograms and conditional density plots, correspondence analysis maps and association graphs. The displays follow the strucplot framework of Meyer, Zeileis and Hornik (2006) <doi:10.18637/jss.v017.i03> and Friendly and Meyer (2016, ISBN: 9781498725835).

License GPL-2 | GPL-3

Encoding UTF-8

Language en-GB

Depends R (>= 4.1.0)

Imports htmlwidgets, stats, grDevices, utils

Suggests vcd, vcdExtra, ca, gnm, robustbase, shiny, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.1

URL https://github.com/Airpino/vcdinteractive

BugReports https://github.com/Airpino/vcdinteractive/issues

Config/pak/sysreqs cmake make libuv1-dev

Repository https://airpino.r-universe.dev

Date/Publication 2026-11-05 22:34:46 UTC

RemoteUrl https://github.com/airpino/vcdinteractive

RemoteRef HEAD

RemoteSha c2d1e75f92b91f0f36530800202f61ca8debbc70

Contents

iassoc_graph 2
ifourfold 3
imcaplot 4
imosaic 6
imosaicOutput 8
ipairs 9
irootogram 10
ispine 11
mosaic_cells 13
residual_types 14
significant_cells 14

Index **16**

`iassoc_graph` *Interactive association graph of a log-linear model*

Description

An interactive version of ‘vcdExtra::assoc_graph()‘ and its ‘plot()‘ method. The variables of a contingency table are the nodes; two variables are joined when they appear together in a term of the log-linear model. The width of an edge is proportional to the strength of the partial association, measured by the increase in G^2 when that association is removed from the model (with its degrees of freedom and p-value).

Usage

```
iassoc_graph(
  x,
  models = c("mutual", "joint", "conditional", "markov", "saturated"),
  initial = NULL,
  main = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

`x`	a contingency table, or anything accepted by [mosaic_cells()] except a fitted model.
`models`	the models to compare, as in [imosaic()]. The default compares the standard sequence of models; the one with the smallest AIC is shown at start-up.
`initial`	the model shown at start-up (position or name); by default the one with the smallest AIC.
`main`	optional title.
`width`, `height`, `elementId`	passed to [htmlwidgets::createWidget()].

Details

Several models can be compared: a side table lists them with G^2, df, p-value and AIC, and clicking a row redraws the graph. The nodes can be dragged; hovering an edge shows its statistics, hovering a node highlights its edges. Optionally the marginal associations of all pairs (Cramér's V of the two-way margins) are drawn as dotted lines.

Value

an 'htmlwidget'.

Examples

```
iassoc_graph(Titanic)
iassoc_graph(UCBAdmissions, models = list(
  "[AD][GD]" = ~ Admit * Dept + Gender * Dept,
  "[AGD]" = ~ Admit * Gender * Dept))
```

---

ifourfold *Interactive fourfold display of 2 x 2 x k tables*

Description

The interactive counterpart of 'vcd::fourfold()'. Each stratum of a $2 \times 2 \times k$ table is drawn as four quarter circles whose areas are proportional to the cell frequencies, after standardizing the table so that both margins are equal (which leaves the odds ratio unchanged). The diagonal quarters are coloured blue for a positive and red for a negative association, in a darker shade when the confidence interval of the odds ratio excludes 1; the rings show the confidence interval.

Usage

```
ifourfold(
  x,
  std = c("margins", "ind.max", "all.max"),
  conf_level = 0.95,
  main = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

x	a 2×2 or $2 \times 2 \times k$ table (further dimensions are crossed into a single stratum variable), or anything accepted by [mosaic_cells()] except a fitted model.
std	initial standardization.
conf_level	initial confidence level, one of 0.90, 0.95, 0.99.
main	optional title.
width, height, elementId	passed to [htmlwidgets::createWidget()].

Details

A second panel plots the log odds ratio of each stratum with its confidence interval, together with the Mantel-Haenszel estimate of the common odds ratio. Woolf's test of homogeneity of the odds ratios across strata is reported in the header. In the widget you can change the standardization (‘"margins"‘, ‘"ind.max"‘, ‘"all.max"‘ as in ‘vcd::fourfold()‘), the confidence level and the display of the rings; hovering a stratum in either panel highlights it in both.

Value

an ‘htmlwidget‘.

Examples

```
ifourfold(UCBAdmissions)
ifourfold(aperm(UCBAdmissions, c(2, 1, 3)), std = "ind.max")
```

---

imcaplot *Interactive correspondence analysis map*

Description

An interactive version of the plots of ‘vcdExtra::mcaplot()‘ and of ‘plot()‘ for correspondence analysis. For a two-way table a simple correspondence analysis is computed; for three or more variables a multiple correspondence analysis (from the Burt matrix), whose percentages of inertia are Greenacre's adjusted ones. Category points are in principal coordinates of the indicator analysis. Objects returned by ‘ca::ca()‘ and ‘ca::mjca()‘ can also be passed.

Usage

```
imcaplot(
  x,
  dims = c(1, 2),
  nd = 5,
  lines = TRUE,
  main = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

x	a contingency table (or anything accepted by [mosaic_cells()] except a fitted model), or a 'ca' or 'mjca' object.
dims	dimensions shown at start-up.
nd	maximum number of dimensions made available in the widget.
lines	join the categories of each variable at start-up.
main	optional title.
width, height, elementId	passed to [htmlwidgets::createWidget()].

Details

In the widget you choose the dimensions on the axes, show or hide each variable (click its legend entry), join the categories of each variable with lines, zoom with the mouse wheel, pan by dragging and reset with a double click. Labels that would overlap are hidden and appear on hover; the tooltip reports mass, coordinates, contributions and the quality of representation (squared cosines) on the two displayed dimensions.

Value

an 'htmlwidget'.

Examples

```
imcaplot(margin.table(HairEyeColor, 1:2))
imcaplot(HairEyeColor)
```

`imosaic` *Interactive mosaic plot*

Description

Draws a mosaic plot of a contingency table as an interactive 'htmlwidget'. Compared with 'vcd::mosaic()':

- the order of the variables can be changed by dragging them in the toolbar, and the direction of each split (horizontal/vertical) can be toggled; the plot is recomputed and animated immediately, in the browser, without re-running R;
- hovering a tile shows the combination of categories, the observed and expected counts, their ratio and difference, the residual and its p-value; hovering a category label highlights all its tiles;
- the residuals used for shading can be switched among those listed in [residual_types] (Pearson, adjusted standardized, deviance, Freeman-Tukey, Anscombe, robust);
- variables can be added to or removed from the plot with the check box of each variable: the sub-tables (margins of the full table) and their models are precomputed, so that the mosaic can be built up one variable at a time;
- a side panel lists the cells that are significantly far from the model, with a choice of multiplicity correction; hovering a row highlights the corresponding tile.

Usage

```
imosaic(x, ...)

## S3 method for class 'formula'
imosaic(formula, data, ...)

## S3 method for class 'glm'
imosaic(x, ...)

## Default S3 method:
imosaic(
  x,
  expected = NULL,
  residuals = "standardized",
  direction = "h",
  alpha = 0.05,
  adjust = "holm",
  robust = c("auto", "medpolish", "glmrob", "none"),
  breaks = c(2, 4),
  colors = NULL,
  main = NULL,
  display = c("mosaic", "assoc", "sieve", "doubledecker"),
  models = NULL,
  include = NULL,
```

```
  subsets = c("auto", "all", "sequential", "none"),
  width = NULL,
  height = NULL,
  elementId = NULL,
  ...
)
```

Arguments

x	a contingency table (‘table‘, ‘xtabs‘, ‘ftable‘, ‘vcd::structable‘, array), a data frame either in case form or in frequency form (with a ‘Freq‘ column), or a fitted Poisson log-linear model (‘glm‘ with ‘family = poisson‘, or ‘gnm::gnm‘) for frequency data, whose fitted values are then used as expected counts.
...	further arguments (currently ignored).
formula	optional one-sided formula selecting (and ordering) the variables of ‘data‘, e.g. ‘~ Class + Sex + Survived‘.
data	a table or data frame, used with the formula interface.
expected	the log-linear model giving the expected counts: ‘NULL‘ (mutual independence), a one-sided formula such as ‘~ Class * Sex + Survived‘, or a list of margins as in [stats::loglin()].
residuals	residual type used for shading at start-up.
direction	initial split directions, a vector of ‘"h"‘/‘"v"‘ recycled over variables. The default ‘"h"‘ alternates ‘h, v, h, ...‘ as in ‘vcd::mosaic()‘.
alpha, adjust	significance level and multiplicity correction used for the list of significant cells (see [significant_cells()]).
robust	method for the robust fit: ‘"auto"‘ (‘robustbase::glmrob()‘ if ’robustbase’ is installed, otherwise median polish for the mutual independence model), ‘"glmrob"‘, ‘"medpolish"‘ or ‘"none"‘.
breaks	absolute residual cut-offs for the shading (as ‘interpolate‘ in ‘vcd::shading_hcl()‘).
colors	optional named list with ‘pos‘, ‘neg‘ (vectors of ‘length(breaks)‘ colours from light to dark) and ‘neutral‘.
main	optional title.
display	initial display: ‘"mosaic"‘, ‘"assoc"‘ (association plot, as ‘vcd::assoc()‘), ‘"sieve"‘ (sieve diagram, as ‘vcd::sieve()‘) or ‘"doubledecker"‘ (as ‘vcd::doubledecker()‘, the last displayed variable being the response). It can be changed in the widget.
models	optional set of log-linear models to compare in the widget, replacing ‘expected‘: a character vector with some of ‘"mutual"‘, ‘"joint"‘, ‘"conditional"‘, ‘"markov"‘, ‘"saturated"‘ (as in ‘vcdExtra::seq_loglm()‘, built on the order of the variables of the table, the last one playing the role of response), and/or a named list of one-sided formulas or lists of margins. The first model is shown at start-up; a panel lists them all with G^2, df, p-value and AIC.
include	initially displayed variables (names or positions); the others can be added from the toolbar. Default: all.

`subsets` which sub-tables are precomputed, so that variables can be added and removed in the browser: ‘"all"‘ (every non-empty subset of the variables, $2^k - 1$ tables), ‘"sequential"‘ (only the leading variables of the initial order, as in ‘vcdExtra::seq_mosaic()‘: variables can then be added or removed only at the end), or ‘"none"‘. The default ‘"auto"‘ is ‘"all"‘ for up to 6 variables and ‘"sequential"‘ otherwise. With a fitted model as input only the full table is available.

`width`, `height`, `elementId`
passed to [htmlwidgets::createWidget()].

Details

Since expected counts and residuals depend on the cells and not on the order in which variables are split, everything is computed once in R by [mosaic_cells()] and the layout is recomputed in JavaScript.

When a variable is removed, the table is collapsed over it and the model is projected on the remaining variables: each term of ‘expected‘ is restricted to the variables still displayed. For instance, removing ‘Dept‘ from the model ‘~ Admit * Dept + Gender * Dept‘ gives the model of independence ‘~ Admit + Gender‘ for the collapsed ‘Admit x Gender‘ table, which is the comparison behind Simpson’s paradox.

Value

an object of class ‘htmlwidget‘.

Examples

```
imosaic(HairEyeColor)
imosaic(Titanic, residuals = "robust")
imosaic(~ Class + Survived, data = Titanic, expected = ~ Class + Survived)
# start from two variables and add the others interactively
imosaic(Titanic, include = c("Class", "Survived"))
```

---

imosaicOutput *Shiny bindings for imosaic*

Description

Shiny bindings for imosaic

Usage

```
imosaicOutput(outputId, width = "100%", height = "620px")

renderImosaic(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

outputId	output variable to read from.
width, height	widget size.
expr	an expression that generates an [imosaic()].
env	the environment in which to evaluate ‘expr‘.
quoted	is ‘expr‘ a quoted expression?

Value

a Shiny output or render function.

---

ipairs *Interactive matrix of pairwise mosaic plots*

Description

The interactive counterpart of ‘pairs()‘ for tables in ’vcd’: a matrix with one mosaic plot for each pair of variables and a bar chart of each variable on the diagonal. Each pairwise mosaic is shaded by the residuals of the independence model for that two-way margin (‘type = "pairwise"‘ in ‘vcd::pairs.table()‘).

Usage

```
ipairs(
  x,
  residuals = "standardized",
  alpha = 0.05,
  adjust = "holm",
  robust = c("auto", "medpolish", "glmrob", "none"),
  breaks = c(2, 4),
  colors = NULL,
  main = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

x	a contingency table with at least two variables, or anything accepted by [mosaic_cells()] except a fitted model.
residuals	residual type used for shading at start-up.
alpha, adjust	significance level and multiplicity correction used for the list of significant cells (see [significant_cells()]).

`robust`	method for the robust fit: ‘"auto"‘ (‘robustbase::glmrob()‘ if ’robustbase’ is installed, otherwise median polish for the mutual independence model), ‘"glmrob"‘, ‘"medpolish"‘ or ‘"none"‘.
`breaks`	absolute residual cut-offs for the shading (as ‘interpolate‘ in ‘vcd::shading_hcl()‘).
`colors`	optional named list with ‘pos‘, ‘neg‘ (vectors of ‘length(breaks)‘ colours from light to dark) and ‘neutral‘.
`main`	optional title.
`width, height, elementId`	passed to [htmlwidgets::createWidget()].

Details

Interactions: hovering a tile shows its counts, residuals and p-values and highlights, in all the other panels, the tiles with the same categories (linked highlighting); hovering a bar of the diagonal highlights that category everywhere; clicking a panel enlarges it, with category labels, and the "Matrix" button goes back. The side panel switches the residuals, and lists the significant cells of all panels, with a multiplicity correction applied to all the cells of all the panels.

Value

an ‘htmlwidget‘.

Examples

```
ipairs(HairEyeColor)
ipairs(Titanic, residuals = "pearson")
```

`irootogram` *Interactive rootogram for discrete distributions*

Description

The interactive counterpart of ‘vcd::rootogram()‘ applied to ‘vcd::goodfit()‘. The observed frequencies of a count variable are drawn as bars hanging from (or standing on) the curve of the frequencies expected under a fitted discrete distribution, on the square-root scale (Tukey’s rootogram), which stabilizes the variance so that departures are comparable along the whole range.

Usage

```
irootogram(
  x,
  dists = c("poisson", "binomial", "nbinomial"),
  size = NULL,
  type = c("hanging", "standing", "deviation"),
  scale = c("sqrt", "raw"),
  xlab = NULL,
  main = NULL,
```

```
  width = NULL,
  height = NULL,
  elementId = NULL
)
```

Arguments

x	the frequency distribution: a one-way table or named numeric vector whose names are the count values (as ‘vcd::HorseKicks‘), a data frame with a value column and a frequency column (as ‘vcd::Federalist‘ converted with ‘as.data.frame()‘), or a vector of raw counts.
dists	distributions to fit; the first one is shown at start-up.
size	number of trials of the binomial distribution (default: the largest observed value).
type, scale	initial type of rootogram and scale.
xlab	label of the count variable.
main	optional title.
width, height, elementId	passed to [htmlwidgets::createWidget()].

Details

The distributions (Poisson, binomial, negative binomial) are fitted by maximum likelihood. In the widget you can switch the distribution, the type of rootogram (‘"hanging"‘, ‘"standing"‘, ‘"deviation"‘) and the scale (square root or raw); the header reports the parameters and the Pearson and likelihood-ratio goodness-of-fit statistics, with $df = K - 1 -$ number of estimated parameters, K being the number of count values. Hovering a bar shows the observed and expected frequencies and the Pearson residual.

Value

an ‘htmlwidget‘.

Examples

```
horse <- c("0" = 109, "1" = 65, "2" = 22, "3" = 3, "4" = 1)
irootogram(horse, xlab = "Deaths by horse kicks")
irootogram(rpois(200, 3) + rbinom(200, 1, 0.3) * 5, dists = c("nbinomial", "poisson"))
```

---

ispine *Interactive spinogram and conditional density plot*

Description

Interactive counterparts of ‘vcd::spine()‘ and ‘vcd::cd_plot()‘ for a categorical response and a numeric predictor. Both show how the conditional distribution of the response changes along the predictor: the spinogram groups the predictor into bins whose widths are proportional to their counts, the conditional density plot smooths $P(y = k \mid x)$ with kernel density estimates.

Usage

```
ispine(
  formula,
  data,
  display = c("spine", "cdplot"),
  breaks = NULL,
  bw = 1,
  main = NULL,
  width = NULL,
  height = NULL,
  elementId = NULL
)

icdplot(formula, data, ...)
```

Arguments

`formula`	a formula ‘y ~ x‘ with a factor (or logical) response and a numeric predictor.
`data`	a data frame.
`display`	initial display, ‘"spine"‘ or ‘"cdplot"‘.
`breaks`	initial number of bins of the spinogram (default: Sturges).
`bw`	initial bandwidth multiplier for the conditional density.
`main`	optional title.
`width`, `height`, `elementId`	passed to [htmlwidgets::createWidget()].
`...`	further arguments passed to ‘ispine()‘.

Details

The widget switches between the two displays; a slider sets the number of bins of the spinogram (equal-width or quantile bins) or the bandwidth of the kernel, as a multiple of the default ‘stats::bw.nrd0()‘ bandwidth, and the plot is recomputed in the browser. Hovering shows the bin counts and proportions, or the estimated conditional probabilities at that value.

The conditional density uses the same Gaussian kernel and bandwidth for all the classes, so that $\hat{P}(y = k \mid x) = \sum_{i:y_i=k} K_h(x - x_i) / \sum_i K_h(x - x_i)$, which is what ‘vcd::cd_plot()‘ computes from ‘density()‘ estimates.

Value

an ‘htmlwidget‘.

Examples

```
ispine(am ~ mpg, data = transform(mtcars, am = factor(am, labels = c("automatic", "manual"))))
icdplot(Species ~ Sepal.Length, data = iris)
```

`mosaic_cells` *Cell statistics for a mosaic plot*

Description

Fits a Poisson log-linear model to a contingency table (by default the model of mutual independence, as in ‘vcd::mosaic(..., shade = TRUE)‘) and returns, for every cell, the observed and expected counts, several kinds of residuals (see [residual_types]) and two-sided p-values.

Usage

```
mosaic_cells(
  x,
  expected = NULL,
  robust = c("auto", "medpolish", "glmrob", "none")
)
```

Arguments

`x`	a contingency table (‘table‘, ‘xtabs‘, ‘ftable‘, ‘vcd::structable‘, array), a data frame either in case form or in frequency form (with a ‘Freq‘ column), or a fitted Poisson log-linear model (‘glm‘ with ‘family = poisson‘, or ‘gnm::gnm‘) for frequency data, whose fitted values are then used as expected counts.
`expected`	the log-linear model giving the expected counts: ‘NULL‘ (mutual independence), a one-sided formula such as ‘~ Class * Sex + Survived‘, or a list of margins as in [stats::loglin()].
`robust`	method for the robust fit: ‘"auto"‘ (‘robustbase::glmrob()‘ if ’robustbase’ is installed, otherwise median polish for the mutual independence model), ‘"glmrob"‘, ‘"medpolish"‘ or ‘"none"‘.

Details

P-values are obtained by comparing with the standard normal distribution the *adjusted* version of each residual (divided by $\sqrt{1-h}$, h being the leverage of the cell), which is asymptotically N(0, 1). For the robust residuals no leverage correction is applied. P-values are unadjusted: multiplicity corrections are applied in [significant_cells()] and in the widget.

Value

a data frame of class ‘mosaic_cells‘ with one row per cell. The attribute ‘model‘ holds the global goodness-of-fit statistics.

Examples

```
cells <- mosaic_cells(HairEyeColor)
head(cells)
attr(cells, "model")
```

residual_types *Residual types available in vcdinteractive*

Description

Character vector with the names of the residuals computed by [mosaic_cells()]. They are:

'pearson' $(o - e)/\sqrt{e}$, the residuals used by default in 'vcd::mosaic()'. Their variance is smaller than one, so they are conservative when compared with normal quantiles.

'standardized' adjusted (Haberman) residuals $(o - e)/\sqrt{e(1 - h)}$, where h is the leverage of the cell in the Poisson log-linear model. Asymptotically N(0, 1): the recommended choice for cell-wise testing.

'deviance' signed square roots of the cell contributions to G^2.

'freeman_tukey' $\sqrt{o} + \sqrt{o + 1} - \sqrt{4e + 1}$, variance-stabilized; better behaved than Pearson residuals with small expected counts.

'anscombe' Anscombe residuals for the Poisson, $1.5(o^{2/3} - e^{2/3})/e^{1/6}$, closest to normality.

'robust' Pearson-type residuals $(o - \tilde{e})/\sqrt{\tilde{e}}$ where $\tilde{e}$ comes from a robust fit of the same model: by default a Poisson GLM fitted by robust quasi-likelihood ('robustbase::glmrob(method = "Mqle")', Cantoni and Ronchetti, 2001); alternatively a median polish of the log-counts (mutual independence only; fast but unreliable for sparse tables). Outlying cells do not pull the fitted values towards themselves, which avoids the masking of deviations that affects the classical fit.

Usage

```
residual_types
```

Format

An object of class `character` of length 6.

significant_cells *Cells that depart significantly from the fitted model*

Description

Returns the combinations of categories whose residual is significantly different from zero, after a multiple-testing correction across all cells of the table, sorted by decreasing absolute residual.

Usage

```
significant_cells(x, type = "standardized", alpha = 0.05, adjust = "holm", ...)
```

Arguments

x	a table (or anything accepted by [mosaic_cells()]) or an object returned by [mosaic_cells()].
type	residual type, one of [residual_types].
alpha	significance level.
adjust	multiplicity correction, passed to [stats::p.adjust()] (‘"holm"‘, ‘"bonferroni"‘, ‘"BH"‘, ‘"none"‘, ...).
...	further arguments passed to [mosaic_cells()].

Value

a data frame with the categories, observed and expected counts, the residual, the raw and adjusted p-values and the direction of the departure (‘"over"‘/‘"under"‘-represented).

Examples

```
significant_cells(HairEyeColor)
significant_cells(HairEyeColor, type = "robust", adjust = "BH")
```

Index

∗ **datasets**
residual_types, 14

iassoc_graph, 2
icdplot *(*ispine*)*, 11
ifourfold, 3
imcaplot, 4
imosaic, 6
imosaicOutput, 8
ipairs, 9
irootogram, 10
ispine, 11

mosaic_cells, 13

renderImosaic *(*imosaicOutput*)*, 8
residual_types, 14

significant_cells, 14