Package ‘CGPfunctions’

March 6, 2020

Title  Powell Miscellaneous Functions for Teaching and Learning Statistics

Version  0.5.9

Description  Miscellaneous functions useful for teaching statistics as well as actually practicing the art. They typically are not “new” methods but rather wrappers around either base R or other packages.

Depends  R (>= 3.6.0)

License  MIT + file LICENSE

Encoding  UTF-8

LazyData  true

Imports  BayesFactor, broom, car, DescTools, dplyr, forcats, ggplot2, ggrepel, grid, methods, paletteer, partykit, purrr, rlang, scales, sjstats, stats, stringr, tidy

Suggests  BSDA, ggthemes, hrbrthemes, janitor, knitr, lsr, magrittr, productplots, pwr, rmarkdown, stringi, tidy, testthat, tidyselect

VignetteBuilder  knitr

RoxygenNote  7.0.2

URL  https://github.com/ibecav/CGPfunctions

BugReports  https://github.com/ibecav/CGPfunctions/issues

NeedsCompilation  no

Author  Chuck Powell [aut, cre] (<https://orcid.org/0000-0002-3606-2188>)

Maintainer  Chuck Powell <ibecav@gmail.com>

Repository  CRAN

Date/Publication  2020-03-06 18:00:10 UTC
CGPfunctions

Description

A package that includes miscellaneous functions useful for teaching statistics as well as actually practicing the art. They typically are not new methods but rather wrappers around either base R or other packages.

Functions included

- `newggslopegraph` creates a "slopegraph" as conceptualized by Edward Tufte.
- `Plot2WayANOVA` which as the name implies conducts a 2 way ANOVA and plots the results using 'ggplot2'
- `PlotXTabs2` which wraps around ggplot2 to provide Bivariate bar charts for categorical and ordinal data.
- `chaid_table` provides tabular summary of CHAID partykit object.
- `cross2_var_vectors` helper function to cross a vector of variables.
- `PlotXTabs` Plots cross tabulated variables using 'ggplot2'
- `Mode` which finds the modal value in a vector of data
- `SeeDist` which wraps around ggplot2 to provide visualizations of univariate data.
- `OurConf` which wraps around ggplot2 to provide visualizations of sampling confidence intervals.
Description

Data from a post-election survey following the year 2000 U.S. presidential elections. This is a subset from package ‘CHAID’.

Usage

chaidUS

Format

A partykit on the following 6 variables:

- vote3 candidate voted for Gore or Bush
- gender gender, a factor with levels male and female
- ager age group, an ordered factor with levels 18-24 < 25-34 < 35-44 < 45-54 < 55-64 < 65+
- empstat status of employment, a factor with levels yes, no or retired
- educr status of education, an ordered factor with levels <HS < HS < >HS < College < Post Coll
- marstat status of living situation, a factor with levels married, widowed, divorced or never married

Source

https://r-forge.r-project.org/R/?group_id=343

Description

Produce CHAID results tables from a partykit CHAID model

Usage

chaid_table(chaidobject)

Arguments

chaidobject An object of type ‘constparty’ or ‘party’ which was produced by ‘CHAID::chaid’ see simple example below.
Value

A tibble containing the results.

Author(s)

Chuck Powell

Examples

library(CGPfunctions)
chaid_table(chaidUS)

cross2_var_vectors
Cross two vectors of variable names from a dataframe

Description

Cross two vectors of variable names from a dataframe

Usage

cross2_var_vectors(data, x, y, verbose = FALSE)

Arguments

data the dataframe or tibble the variables are contained in.

x, y These are either character or integer vectors containing the names, e.g. "am" or the column numbers e.g. 9

verbose the default is FALSE, setting to TRUE will cat additional output to the screen

Value

a list with two sublists ‘lista’ and ‘listb’. Very handy for feeding the lists to ‘purrr’ for further processing.

Author(s)

Chuck Powell
Mode

Derive the modal value(s) for a set of data

Description

This function takes a vector and returns one or mode values that represent the mode point of the data

Usage

Mode(x)

Arguments

x a vector
Value

a vector containing one or more modal values for the input vector

Warning

Be careful the function does some basic error checking but the return to Mode(NA) is NA and a vector where the majority of entries are NA is also NA

Examples

Mode(sample(1:100, 1000, replace = TRUE))
Mode(mtcars$hp)
Mode(iris$Sepal.Length)

newcancer

Tufte dataset on cancer survival rates

Description

A dataset containing cancer survival rates for different types of cancer over a 20 year period.

Usage

newcancer

Format

A data frame with 96 rows and 3 variables:

Year ordered factor for the 5, 10, 15 and 20 year survival rates
Type factor containing the name of the cancer type
Survival numeric for this data a whole number corresponding to the percent survival rate

Source

**newgdp**

*Tuft dataset on Gross Domestic Product, 1970 and 1979*

**Description**

Current receipts of fifteen national governments as a percentage of gross domestic product

**Usage**

newgdp

**Format**

A data frame with 30 rows and 3 variables:

- **Year** character for 1970 and 1979
- **Country** factor country name
- **GDP** numeric a percentage of gross domestic product

**Source**


**newggslopegraph**

*Plot a Slopegraph a la Tufte using dplyr and ggplot2*

**Description**

Creates a "slopegraph" as conceptualized by Edward Tufte. Slopegraphs are minimalist and efficient presentations of your data that can simultaneously convey the relative rankings, the actual numeric values, and the changes and directionality of the data over time. Takes a dataframe as input, with three named columns being used to draw the plot. Makes the required adjustments to the ggplot2 parameters and returns the plot.

**Usage**

newggslopegraph(
    dataframe,
    Times,
    Measurement,
    Grouping,
    Title = "No title given",
    SubTitle = "No subtitle given",
    Caption = "No caption given",
    XTextSize = 12,
)
Arguments

- **dataframe**: a dataframe or an object that can be coerced to a dataframe. Basic error checking is performed, to include ensuring that the named columns exist in the dataframe. See the `newcancer` dataset for an example of how the dataframe should be organized.

- **Times**: a column inside the dataframe that will be plotted on the x axis. Traditionally this is some measure of time. The function accepts a column of class ordered, factor or character. NOTE if your variable is currently a "date" class you must convert before using the function with `as.character(variableName)`.

- **Measurement**: a column inside the dataframe that will be plotted on the y axis. Traditionally this is some measure such as a percentage. Currently the function accepts a column of type integer or numeric. The slopegraph will be most effective when the measurements are not too disparate.

- **Grouping**: a column inside the dataframe that will be used to group and distinguish measurements.

- **Title**: Optionally the title to be displayed. `Title = NULL` will remove it entirely. `Title = ""` will provide an empty title but retain the spacing.

- **SubTitle**: Optionally the sub-title to be displayed. `SubTitle = NULL` will remove it entirely. `SubTitle = ""` will provide and empty title but retain the spacing.

- **Caption**: Optionally the caption to be displayed. `Caption = NULL` will remove it entirely. `Caption = ""` will provide and empty title but retain the spacing.

- **XTextSize**: Optionally the font size for the X axis labels to be displayed. `XTextSize = 12` is the default must be a numeric. Note that X & Y axis text are on different scales

- **YTextSize**: Optionally the font size for the Y axis labels to be displayed. `YTextSize = 3` is the default must be a numeric. Note that X & Y axis text are on different scales
TitleTextSize
Optionally the font size for the Title to be displayed. TitleTextSize = 14 is the default must be a numeric.

SubTitleTextSize
Optionally the font size for the SubTitle to be displayed. SubTitleTextSize = 10 is the default must be a numeric.

CaptionTextSize
Optionally the font size for the Caption to be displayed. CaptionTextSize = 8 is the default must be a numeric.

TitleJustify
Justification of title can be either a character "L", "R" or "C" or use the hjust = notation from ggplot2 with a numeric value between ‘0’ (left) and ‘1’ (right).

SubTitleJustify
Justification of subtitle can be either a character "L", "R" or "C" or use the hjust = notation from ggplot2 with a numeric value between ‘0’ (left) and ‘1’ (right).

CaptionJustify
Justification of caption can be either a character "L", "R" or "C" or use the hjust = notation from ggplot2 with a numeric value between ‘0’ (left) and ‘1’ (right).

LineThickness
Optionally the thickness of the plotted lines that connect the data points. LineThickness = 1 is the default must be a numeric.

LineColor
Optionally the color of the plotted lines. By default it will use the ggplot2 color palette for coloring by Grouping. The user may override with one valid color of their choice e.g. "black" (see colors() for choices) OR they may provide a vector of colors such as c("gray", "red", "green", "gray", "blue") OR a named vector like c("Green" = "gray", "Liberal" = "red", "NDP" = "green", "Others" = "gray", "PC" = "blue"). Any input must be character, and the length of a vector should equal the number of levels in Grouping. If the user does not provide enough colors they will be recycled.

DataTextSize
Optionally the font size of the plotted data points. DataTextSize = 2.5 is the default must be a numeric.

DataTextColor
Optionally the font color of the plotted data points. "black" is the default can be either 'colors()' or hex value e.g. "#FF00FF".

DataLabelPadding
Optionally the amount of space between the plotted data point numbers and the label "box". By default very small = 0.05 to avoid overlap. Must be a numeric. Too large a value will risk "hiding" datapoints.

DataLabelLineWidth
Optionally how wide a line to plot around the data label box. By default = 0 to have no visible border line around the label. Must be a numeric.

DataLabelFillColor
Optionally the fill color or background of the plotted data points. "white" is the default can be any of the 'colors()' or hex value e.g. "#FF00FF".

WiderLabels
logical, set this value to TRUE if your "labels" or Grouping variable values tend to be long as they are in the newcancer dataset. This setting will give them more room in the same plot size.

ReverseYAxis
logical, set this value to TRUE if you want to reverse the Y scale, especially useful for rankings when you want #1 on top.

ReverseXAxis
logical, set this value to TRUE if you want to reverse the **factor levels** on the X scale.
RemoveMissing logical, by default set to TRUE so that if any Measurement is missing all rows for that Grouping are removed. If set to FALSE then the function will try to remove and graph what data it does have. N.B. missing values for Times and Grouping are never permitted and will generate a fatal error with a warning.

ThemeChoice character, by default set to "bw" the other choices are "ipsum", "econ", "wsj", "gdocs", and "tufte".

Value

a plot of type ggplot to the default plot device

Author(s)

Chuck Powell

References


See Also

newcancer and newgdp

Examples

# the minimum command to generate a plot
newggslopegraph(newcancer, Year, Survival, Type)

# adding a title which is always recommended
newggslopegraph(newcancer, Year, Survival, Type,
    Title = "Estimates of Percent Survival Rates",
    SubTitle = NULL,
    Caption = NULL
)

# simple formatting changes
newggslopegraph(newcancer, Year, Survival, Type,
    Title = "Estimates of Percent Survival Rates",
    LineColor = "darkgray",
    LineThickness = .5,
    SubTitle = NULL,
    Caption = NULL
)

# complex formatting with recycling and wider labels see vignette for more examples
newggslopegraph(newcancer, Year, Survival, Type,
    Title = "Estimates of Percent Survival Rates",
    SubTitle = "Based on: Edward Tufte, Beautiful Evidence, 174, 176."
    LineColor = c("black", "red", "grey"),
    LineThickness = .5,
WiderLabels = TRUE

# not a great example but demonstrating functionality
newgdp$rGDP <- round(newgdp$GDP)

newggslopegraph(newgdp,
  Year,
  rGDP,
  Country,
  LineColor = c(rep("grey", 3), "red", rep("grey", 11)),
  DataTextSize = 3,
  DataLabelFillColor = "gray",
  DataLabelPadding = .2,
  DataLabelLineSize = .5
)

---

**OurConf**  
*Plotting random samples of confidence intervals around the mean*

**Description**

This function takes some parameters and simulates random samples and their confidence intervals.

**Usage**

```
OurConf(samples = 100, n = 30, mu = 0, sigma = 1, conf.level = 0.95)
```

**Arguments**

- `samples`  
The number of times to draw random samples
- `n`  
The sample size we draw each time
- `mu`  
The population mean mu
- `sigma`  
The population standard deviation
- `conf.level`  
What confidence level to compute 1 - alpha (significance level)

**Value**

A ggplot2 object

**Author(s)**

Chuck Powell

**See Also**

`qnorm, rnorm, CIsim`
Examples

OurConf(samples = 100, n = 30, mu = 0, sigma = 1, conf.level = 0.95)
OurConf(samples = 2, n = 5)
OurConf(samples = 25, n = 25, mu = 100, sigma = 20, conf.level = 0.99)

Description

Takes a formula and a dataframe as input, conducts an analysis of variance prints the results (AOV summary table, table of overall model information and table of means) then uses ggplot2 to plot an interaction graph (line or bar). Also uses Brown-Forsythe test for homogeneity of variance. Users can also choose to save the plot out as a png file.

Usage

Plot2WayANOVA(formula,
  dataframe = NULL,
  confidence=.95,
  plottype = "line",
  xlab = NULL,
  ylab = NULL,
  title = NULL,
  subtitle = NULL,
  interact.line.size = 2,
  ci.line.size = 1,
  mean.label = FALSE,
  mean.ci = TRUE,
  mean.size = 4,
  mean.shape = 23,
  mean.color = "darkred",
  mean.label.size = 3,
  offset.style = "none",
  overlay.type = NULL,
  posthoc.method = "scheffe",
  show.dots = FALSE,
  PlotSave = FALSE)

Arguments

formula a formula with a numeric dependent (outcome) variable, and two independent (predictor) variables e.g. mpg ~ am * vs. The independent variables are coerced to factors (with warning) if possible.

dataframe a dataframe or an object that can be coerced to a dataframe
Plots2WayANOVA

- **confidence**: what confidence level for confidence intervals
- **plottype**: bar or line (quoted)
- **xlab, ylab**: Labels for 'x' and 'y' axis variables. If 'NULL' (default), variable names for 'x' and 'y' will be used.
- **title**: The text for the plot title. A generic default is provided.
- **subtitle**: The text for the plot subtitle. If 'NULL' (default), key model information is provided as a subtitle.
- **interact.line.size**: Line size for the line connecting the group means (Default: ‘2’).
- **ci.line.size**: Line size for the confidence interval bracketing the group means (Default: ‘1’).
- **mean.label**: Logical that decides whether the value of the group mean is to be displayed (Default: ‘FALSE’).
- **mean.ci**: Logical that decides whether the confidence interval for group means is to be displayed (Default: ‘TRUE’).
- **mean.size**: Point size for the data point corresponding to mean (Default: ‘4’).
- **mean.shape**: Shape of the plot symbol for the mean (Default: ‘23’ which is a diamond).
- **mean.color**: Color for the data point corresponding to mean (Default: ‘darkred’).
- **mean.label.size, mean.label.color**: Aesthetics for the label displaying mean. Defaults: ‘3’, ‘black’, respectively.
- **offset.style**: A character string (e.g., ‘"wide"’ or ‘"narrow"’, or ‘"none"’) which controls whether items are offset from the centerline for clarity. Useful when you want to add individual datapoints or confidence interval lines overlap. (Default: ‘"none"’).
- **overlay.type**: A character string (e.g., ‘"box"’ or ‘"violin"’), if you wish to overlay that information on factor1
- **posthoc.method**: A character string, one of "hsd", "bonf", "lsd", "scheffe", "newmankeuls", defining the method for the pairwise comparisons. (Default: ‘"scheffe"’).
- **show.dots**: Logical that decides whether the individual data points are displayed (Default: ‘FALSE’).
- **PlotSave**: a logical indicating whether the user wants to save the plot as a png file

**Details**

Details about how the function works in order of steps taken.

1. Some basic error checking to ensure a valid formula and dataframe. Only accepts fully *crossed* formula to check for interaction term
2. Ensure the dependent (outcome) variable is numeric and that the two independent (predictor) variables are or can be coerced to factors – user warned on the console
3. Remove missing cases – user warned on the console
4. Calculate a summarized table of means, sds, standard errors of the means, confidence intervals, and group sizes.
5. Use `aov` function to execute an Analysis of Variance (ANOVA)
6. Use `anova_stats` to calculate eta squared and omega squared values per factor. If the design is unbalanced warn the user and use Type II sums of squares
7. Produce a standard ANOVA table with additional columns
8. Use the `PostHocTest` for producing a table of post hoc comparisons for all effects that were significant
9. Use the `leveneTest` for testing Homogeneity of Variance assumption with Brown-Forsythe
10. Use the `PostHocTest` for conducting post hoc tests for effects that were significant
11. Use the `shapiro.test` for testing normality assumption with Shapiro-Wilk
12. Use `ggplot2` to plot an interaction plot of the type the user specified.

The defaults are deliberately constructed to emphasize the nature of the interaction rather than focusing on distributions. So while a violin plot of the first factor by level is displayed along with dots for individual data points shaded by the second factor, the emphasis is on the interaction lines.

**Value**

A list with 5 elements which is returned invisibly. These items are always sent to the console for display but for user convenience the function also returns a named list with the following items in case the user desires to save them or further process them - $ANOVATable, $ModelSummary, $MeansTable, $PosthocTable, $BFTest, and $SWTest. The plot is always sent to the default plot device

**Author(s)**

Chuck Powell

**References**

: ANOVA: Delacre, Leys, Mora, & Lakens, *PsyArXiv*, 2018

**See Also**

`aov, leveneTest, anova_stats, replications, shapiro.test`

**Examples**

```r
Plot2WayANOVA(mpg ~ am * cyl, mtcars, plottype = "line")
Plot2WayANOVA(mpg ~ am * cyl,  
              mtcars,  
              plottype = "line",  
              overlay.type = "box",  
              mean.label = TRUE  
)  
Plot2WayANOVA(mpg ~ am * vs, mtcars, confidence = .99)

# Create a new dataset
library(dplyr)
library(ggplot2)
```
library(stringi)
newmpg <- mpg %>%
  filter(cyl != 5) %>%
  mutate(am = stringi::stri_extract(trans, regex = "auto|manual"))
Plot2WayANOVA(
  formula = hwy ~ am * cyl,
  dataframe = newmpg,
  ylab = "Highway mileage",
  xlab = "Transmission type",
  plottype = "line",
  offset.style = "wide",
  overlay.type = "box",
  mean.label = TRUE,
  mean.shape = 20,
  mean.size = 5,
  mean.label.size = 5,
  show.dots = TRUE
)

PlotXTabs
Plot a Cross Tabulation of two variables using dplyr and ggplot2

Description
Takes a dataframe and at least two variables as input, conducts a crosstabulation of the variables using dplyr. Removes NAs and then plots the results as one of three types of bar (column) graphs using ggplot2. The function accepts either bare variable names or column numbers as input (see examples for the possibilities)

Usage
PlotXTabs(dataframe, xwhich, ywhich, plottype = "side")

Arguments
  dataframe an object that is of class dataframe
  xwhich either a bare variable name that is valid in the dataframe or one or more column
            numbers. An attempt will be made to coerce the variable to a factor but odd
            plots will occur if you pass it a variable that is by rights continuous in nature.
  ywhich either a bare variable name that is valid in the dataframe or one or more column
           numbers that exist in the dataframe. An attempt will be made to coerce the
           variable to a factor but odd plots will occur if you pass it a variable that is by
           rights continuous in nature.
  plottype one of three options "side", "stack" or "percent"

Value
One or more ggplots to the default graphics device as well as advisory information in the console
**Author(s)**

Chuck Powell

**See Also**

`janitor`

**Examples**

```r
PlotXTabs(mtcars, am, vs)
PlotXTabs(mtcars, am, vs, "stack")
PlotXTabs(mtcars, am, vs, "percent")
PlotXTabs(mtcars, am, 8, "side")
PlotXTabs(mtcars, 8, am, "stack")
PlotXTabs(mtcars, am, c(8, 10), "percent")
PlotXTabs(mtcars, c(10, 8), am)
PlotXTabs(mtcars, c(2, 9), c(10, 8), "mispelled")
## Not run:
PlotXTabs(happy, happy, sex) # baseline
PlotXTabs(happy, 2, 5, "stack") # same thing using column numbers
PlotXTabs(happy, 2, c(5:9), plottype = "percent") # multiple columns RHS
PlotXTabs(happy, c(2, 5), 9, plottype = "side") # multiple columns LHS
PlotXTabs(happy, c(2, 5), c(6:9), plottype = "percent")
PlotXTabs(happy, happy, c(6, 7, 9), plottype = "percent")
PlotXTabs(happy, c(6, 7, 9), happy, plottype = "percent")
## End(Not run)
```

**PlotXTabs2**

*Bivariate bar (column) charts with statistical tests*

**Description**

Bivariate bar charts for categorical and ordinal data with (optionally) statistical details included in the plot as a subtitle.

**Usage**

```r
PlotXTabs2(
  data,
  x,
  y,
  counts = NULL,
  results.subtitle = TRUE,
  title = NULL,
  subtitle = NULL,
  caption = NULL,
)```
plottype = "percent",
xlab = NULL,
ylab = "Percent",
legend.title = NULL,
legend.position = "right",
labels.legend = NULL,
sample.size.label = TRUE,
data.label = "percentage",
label.text.size = 4,
label.fill.color = "white",
label.fill.alpha = 1,
bar.outline.color = "black",
x.axis.orientation = NULL,
conf.level = 0.95,
k = 2,
perc.k = 0,
bf.details = FALSE,
bf.display = "regular",
sampling.plan = "jointMulti",
fixed.margin = "rows",
prior.concentration = 1,
paired = FALSE,
ggtheme = ggplot2::theme_bw(),
package = "RColorBrewer",
palette = "Dark2",
direction = 1,
ggplot.component = NULL
)

Arguments

data A dataframe or tibble containing the 'x' and 'y' variables.
x The variable to plot on the X axis of the chart.
y The variable to segment the **columns** and test for independence.
counts If the dataframe is based upon counts rather than individual rows for observations, 'counts' must contain the name of variable that contains the counts. See 'HairEyeColor' example.
results.subtitle Decides whether the results of statistical tests are displayed as a subtitle (Default: TRUE). If set to FALSE, no subtitle.
title The text for the plot title.
subtitle The text for the plot subtitle. **N.B** if statistical results are requested through 'results.subtitle = TRUE' the results will have precedence.
caption The text for the plot caption. Please note the interaction with 'bf.details'.
plottype one of three options "side", "stack" or "percent"
xlab Custom text for the 'x' axis label (Default: 'NULL', which will cause the 'x' axis label to be the 'x' variable).
ylab  Custom text for the ‘y’ axis label (Default: ‘"Percent"’). Set to ‘NULL’ for no label.

legend.title  Title text for the legend.

legend.position  The position of the legend ‘"none"’, ‘"left"’, ‘"right"’, ‘"bottom"’, ‘"top"’ (Default: ‘"right"’).

labels.legend  A character vector with custom labels for levels of the ‘y’ variable displayed in the legend.

sample.size.label  Logical that decides whether sample size information should be displayed for each level of the grouping variable ‘y’ (Default: ‘TRUE’).

data.label  Character decides what information needs to be displayed on the label in each bar segment. Possible options are ‘"percentage"’ (default), ‘"counts"’, ‘"both"’.

label.text.size  Numeric that decides size for bar labels (Default: ‘4’).

label.fill.color  Character that specifies fill color for bar labels (Default: ‘white’).

label.fill.alpha  Numeric that specifies fill color transparency or ‘"alpha"’ for bar labels (Default: ‘0’ range ‘0’ to ‘1’).

bar.outline.color  Character specifying color for bars (default: ‘"black"’).

x.axis.orientation  The orientation of the ‘x’ axis labels one of “slant” or “vertical” to change from the default horizontal orientation (Default: ‘NULL’ which is horizontal).

conf.level  Scalar between 0 and 1. If unspecified, the defaults return lower and upper confidence intervals (0.95).

k  Number of digits after decimal point (should be an integer) (Default: k = 2) for statistical results.

perc.k  Numeric that decides number of decimal places for percentage labels (Default: ‘0’).

bf.details  Logical that decides whether to display additional information from the Bayes Factor test in the caption (default: ‘FALSE’). This will take precedence over any text you enter as a ‘caption’.

bf.display  Character that determines how the Bayes factor value is is displayed. The default is simply the number rounded to ‘k’. Other options include "sensible", "log" and "support".

sampling.plan  the sampling plan (see details in ?contingencyTableBF).

fixed.margin  (see details in ?contingencyTableBF).

prior.concentration  (see details in ?contingencyTableBF).

paired  Not used yet.
ggtheme  A function, ggplot2 theme name. Default value is ggplot2::theme_bw(). Any of the ggplot2 themes, or themes from extension packages are allowed (e.g., hrbrthemes::theme_ipsum(), etc.).

package  Name of package from which the palette is desired as string or symbol.

palette  Name of palette as string or symbol.

direction  Either ‘1’ or ‘-1’. If ‘-1’ the palette will be reversed.

ggplot.component  A ggplot component to be added to the plot prepared by ggstatsplot. Default is NULL. The argument should be entered as a function. If the given function has an argument axes.range.restrict and if it has been set to TRUE, the added ggplot component might not work as expected.

Author(s)

Chuck Powell, Indrajeet Patil

Examples

```r
# for reproducibility
set.seed(123)

# simplest possible call with the defaults
PlotXTabs2(
  data = mtcars,
  y = vs,
  x = cyl
)

# more complex call
PlotXTabs2(
  data = datasets::mtcars,
  y = vs,
  x = cyl,
  bf.details = TRUE,
  labels.legend = c("0 = V-shaped", "1 = straight"),
  legend.title = "Engine Style",
  legend.position = "right",
  title = "The perennial mtcars example",
  palette = "Pastel1"
)

PlotXTabs2(
  data = as.data.frame(HairEyeColor),
  y = Eye,
  x = Hair,
  counts = Freq
)
```
SeeDist

See The Distribution

Description
This function takes a vector of numeric data and returns one or more ggplot2 plots that help you visualize the data.

Usage
SeeDist(
  qqq,
  numbins = 0,
  whatvar = "Unspecified",
  whatplots = c("d", "b", "h")
)

Arguments
- `qqq`: the data to be visualized must be numeric.
- `numbins`: the number of bins to use for any plots that bin. If nothing is specified the function will calculate a rational number using Freedman-Diaconis via the `nclass.FD` function.
- `whatvar`: additional contextual information about the variable as a string such as "Miles Per Gallon".
- `whatplots`: what type of plots? The default is `whatplots = c("d","b","h")` for a density, a boxplot, and a histogram.

Value
from 1 to 3 plots depending on what the user specifies as well as a base R summary printed to the console

Warning
If the data has more than 3 modal values only the first three of them are plotted. The rest are ignored and the user is warned on the console.
Missing values are removed with a warning to the user.

Author(s)
Chuck Powell

See Also
`nclass`
USvoteS

Examples

SeeDist(rnorm(100, mean = 100, sd = 20), numbins = 15, whatvar = "A Random Sample")
SeeDist(mtcars$hp, whatvar = "Horsepower", whatplots = c("d", "b"))
SeeDist(iris$Sepal.Length, whatvar = "Sepal Length", whatplots = "d")

USvoteS

U.S. 2000 Election Data (short)

Description

Data from a post-election survey following the year 2000 U.S. presidential elections. This is a subset from package ‘CHAID’.

Usage

USvoteS

Format

A data frame with 1000 observations on the following 6 variables:

- vote3 candidate voted for Gore or Bush
- gender gender, a factor with levels male and female
- ager age group, an ordered factor with levels 18-24 < 25-34 < 35-44 < 45-54 < 55-64 < 65+
- empstat status of employment, a factor with levels yes, no or retired
- educr status of education, an ordered factor with levels <HS < HS < >HS < College < Post Coll
- marstat status of living situation, a factor with levels married, widowed, divorced or never married

Source

https://r-forge.r-project.org/R/?group_id=343
Index

*Topic datasets
  chaidUS, 3
  newcancer, 6
  newgdp, 7
  USvoteS, 21

anova_stats, 14
aov, 13, 14

CGPfunctions, 2
chaid_table, 2, 3
chaidUS, 3
CISim, 11
cross2_var_vectors, 2, 4

janitor, 16
leveneTest, 14

Mode, 2, 5

nclass, 20
newcancer, 6, 8, 10
newgdp, 7, 10
newggslopegraph, 2, 7

OurConf, 2, 11

Plot2WayANOVA, 2, 12
PlotXTabs, 2, 15
PlotXTabs2, 2, 16
PostHocTest, 14

qnorm, 11

replications, 14
rnorm, 11

SeeDist, 2, 20
shapiro.test, 14

USvoteS, 21