Package 'CGPfunctions'

June 14, 2019

Title Powell Miscellaneous Functions for Teaching and Learning Statistics

Version 0.5.7

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 (>= 3.5.0)

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports broomExtra, car, DescTools, devtools, dplyr, ggplot2, ggrepel, methods, pwr, rlang, scales, sjstats, tibble

Suggests BSDA, lsr, janitor, knitr, productplots, rmarkdown, stringi

VignetteBuilder knitr

RoxygenNote 6.1.1

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 2019-06-14 19:45:36 UTC

R topics documented:

`CGPfunctions` *CGPfunctions: A package of miscellaneous functions for teaching statistics.*

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

- `Plot2WayANOVA` which as the name implies conducts a 2 way ANOVA and plots the results using 'ggplot2'
- `PlotXTabs` Plots cross tabulated variables using 'ggplot2'
- `neweta` which is a helper function that appends the results of a Type II eta squared calculation onto a classic ANOVA table
- `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.

`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

https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0003nk

`neweta` *Calculates eta squared for an AOV model using the Type II method*

Description

Takes an aov object and returns a standard AOV table with eta squared computed

Usage

```
neweta(MyAOV)
```

Arguments

`MyAOV`	a valid aov object such as those produced by `aov(dv~iv1*iv2)`

Details

There are three ways to compute eta squared this function only uses Type II

This function is deprecated as of version 0.5 please consider using `eta_sq` instead

Value

Returns a tibble containing the AOV output similar to `summary(aov(MyAOV))` but with eta squared computed and appended as an additional column

Author(s)

Chuck Powell

References

neweta function is a shortened and simplified verion of Dani Navarro's `etaSquared`

See Also

`Plot2WayANOVA`

Examples

```
neweta(aov(mpg~am*cyl, mtcars))
```

newgdp *Tufte 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

Edward Tufte. *Beautiful Evidence*. Graphics Press, 174-176.

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, YTextSize = 3,
  TitleTextSize = 14, SubTitleTextSize = 10, CaptionTextSize = 8,
  LineThickness = 1, LineColor = "ByGroup", DataTextSize = 2.5,
  DataTextColor = "black", DataLabelPadding = 0.05,
  DataLabelLineSize = 0, DataLabelFillColor = "white",
  WiderLabels = FALSE, RemoveMissing = TRUE)
```

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.
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.

`DataLabelLineSize` 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.

`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.

Value

a plot of type ggplot to the default plot device

Author(s)

Chuck Powell

References

Based on: Edward Tufte, Beautiful Evidence (2006), pages 174-176.

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.",
             Caption = "https://www.edwardtufte.com/bboard/q-and-a-fetch-msg?msg_id=0003nk",
                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)
```

Plot2WayANOVA *Plot a 2 Way ANOVA using dplyr and ggplot2*

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,
              mean.label.color = "black",
```

```
                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
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 confdence 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

```
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

```
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)
```

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

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")
```

Index