Package ‘Rmisc’
February 19, 2015

Type Package
Title Rmisc: Ryan Miscellaneous
Version 1.5
Date 2013-10-21
Author Ryan M. Hope <rmh3093@gmail.com>
Maintainer Ryan M. Hope <rmh3093@gmail.com>
Description The Rmisc library contains many functions useful for data analysis and utility operations.
License GPL-3
Suggests latticeExtra, Hmisc, stats4
Depends lattice, plyr
Collate ‘CI.R’ 'STDERR.R' 'group.UCL.R’ 'group.CI.R’ 'group.STDERR.R’
 'lr.glover.R' 'multiplot.R' 'panel.circle.R' 'rounder.R’
 'rsi.R' 'summarySE.R'
NeedsCompilation no
Repository CRAN
Date/Publication 2013-10-22 00:46:56

R topics documented:

CI .......................................................... 2
group.CI .................................................. 2
group.STDERR  .......................................... 3
group.UCL .................................................. 4
lr.glover ................................................. 4
multiplot .................................................. 5
normDataWithin ......................................... 6
panel.circle ............................................... 7
rounder ..................................................... 7
rsi .......................................................... 8
STDERR .................................................... 8
summarySE ............................................... 9
summarySEwithin ....................................... 10
### CI

**Confidence Interval**

**Description**
Calculates the confidence interval of a vector of data.

**Usage**

\[ \text{CI}(x, \text{ci} = 0.95) \]

**Arguments**

- **x**: a vector of data
- **ci**: the confidence interval to be calculated

**Value**

- **upper**: Upper bound of interval.
- **mean**: Mean of data.
- **lower**: Lower bound of interval.

**Examples**

\[ \text{CI}(\text{rnorm}(100)) \]

---

### group.CI

**Group Confidence Interval**

**Description**

Calculates the confidence interval of grouped data.

**Usage**

\[ \text{group.CI}(x, \text{data}, \text{ci} = 0.95) \]

**Arguments**

- **x**: an ‘aggregate’ compatible formula
- **data**: a data frame (or list) from which the variables in formula should be taken
- **ci**: the confidence interval to be calculated
**Value**

A data frame consisting of one column for each grouping factor plus three columns for the upper bound, mean and lower bound of the confidence interval for each level of the grouping factor.

**Examples**

```r
require(latticeExtra)
with(group.CI(weight~feed,chickwts),
    segplot(feed~weight.lower+weight.upper,center=weight.mean)
)

require(Hmisc)
with(group.CI(Temp~Month.airquality),
    xyplot(Cbind(Temp.mean,Temp.lower,Temp.upper)~numericScale(Month),type="b",ylim=c(60,90))
)
```

---

**Description**

Calculates the standard error interval of grouped data.

**Usage**

```r
group.STDERR(x, data)
```

**Arguments**

- `x` an `aggregate` compatible formula
- `data` a data frame (or list) from which the variables in formula should be taken.

**Value**

A data frame consisting of one column for each grouping factor plus three columns for the upper bound, mean and lower bound of the standard error interval for each level of the grouping factor.

**Examples**

```r
require(latticeExtra)
with(group.STDERR(weight~feed,chickwts),
    segplot(feed~weight.lower+weight.upper,center=weight.mean)
)

require(Hmisc)
with(group.STDERR(Temp~Month.airquality),
    xyplot(Cbind(Temp.mean,Temp.lower,Temp.upper)~numericScale(Month),type="b",ylim=c(60,90))
)
```
**group.UCL**  
*Group Upper-Center-Lower*

**Description**

Applies a function which calculates a parameter with lower/upper bounds to groups of data.

**Usage**

```
group.UCL(x, data, FUN, ...)
```

**Arguments**

- `x`  
an 'aggregate' compatible formula
- `data`  
a data frame (or list) from which the variables in formula should be taken.
- `FUN`  
the function to apply to each group
- `...`  
extra params passed on to aggregate

**Value**

A data frame consisting of one column for each grouping factor plus three columns for the upper bound, mean and lower bound of the standard error interval for each level of the grouping factor.

**Examples**

```
require(latticeExtra)
with(group.UCL(weight~feed, chickwts, FUN=CI),
    segplot(feed~weight.upper+weight.lower+weight.mean, center=weight.mean)
)

require(Hmisc)
with(group.UCL(Temp~Month, airquality, FUN=STDERR),
    xyplot(Cbind(Temp.mean, Temp.lower, Temp.upper)~numericScale(Month), type="b", ylim=c(60,90))
)
```

---

**lr.glover**  
*Likelihood Ratio Test*

**Description**

Computes a likelihood ratio statistic which reflects the relative likelihood of the data given two competing models.

**Usage**

```
lr.glover(object, ..., name = NULL)
```
multiplot

Arguments

- **object**: an object. See below for details.
- **...**: further object specifications passed to methods. See below for details.
- **name**: a function for extracting a suitable name/description from a fitted model object. By default the name is queried by calling `formula`.

Value

An object of class "anova" which contains the log-likelihood, degrees of freedom, the difference in degrees of freedom, likelihood ratio, and AIC/BIC corrected likelihood ratios.

Details

lr.glover performs comparisons of models via likelihood ratio tests. The default method consecutively compares the fitted model object `object` with the models passed in `...`. Subsequently, a likelihood ratio test for each two consecutive models is carried out.

References


Examples

```r
m1 <- lm(mpg ~ ., mtcars)
m2 <- step(m1, ~ ., trace=0)
m3 <- step(m1, ~ . + .^2, trace=0)
lr.glover(m1, m2, m3)
```

---

multiplot  

Multiple plot function

Description

Renders multiple ggplot plots in one image

Usage

```r
multiplot(..., plotlist = NULL, cols = 1, layout = NULL)
```

Arguments

- **...**: ggplot objects
- **plotlist**: a list of ggplot objects
- **cols**: Number of columns in layout
- **layout**: A matrix specifying the layout. If present, 'cols' is ignored
**normDataWithin**

**Note**

If the layout is something like matrix(c(1,2,3,3), nrow=2, byrow=TRUE), then plot 1 will go in the upper left, 2 will go in the upper right, and 3 will go all the way across the bottom.

**References**

http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)

---

normDataWithin Normalize within-group data

**Description**

Norms the data within specified groups in a data frame; it normalizes each subject (identified by idvar) so that they have the same mean, within each group specified by betweenvars.

**Usage**

```r
normDataWithin(data = NULL, idvar, measurevar, 
    betweenvars = NULL, na.rm = FALSE, .drop = TRUE)
```

**Arguments**

- **data**: a data frame.
- **idvar**: the name of a column that identifies each subject (or matched subjects)
- **measurevar**: the name of a column that contains the variable to be summarized
- **betweenvars**: a vector containing names of columns that are between-subjects variables
- **na.rm**: a boolean that indicates whether to ignore NA’s
- **.drop**: should combinations of variables that do not appear in the input data be preserved (FALSE) or dropped (TRUE, default)

**Value**

a data frame with normalized data

**References**

http://www.cookbook-r.com/Graphs/Plotting_means_and_error_bars_(ggplot2)
panel.circle  Circle Drawing

Description
A panel function for drawing circles.

Usage
panel.circle(x, y, r, segments = 50L, groups = NULL, ...)

Arguments
- x: The x coordinate of the circle center
- y: The y coordinate of the circle center
- r: The radius of the circle
- segments: The number of polygon segments used to create the circle
- groups: A factor defining groups
- ...: Additional arguments passed to panel.polygon

Examples
panel.circle(0, 0, 10)

rounder  Round to Increment

Description
Rounds a value to nearest increment

Usage
rounder(x, inc, fun = "round")

Arguments
- x: The value to be rounded
- inc: The increment to round to
- fun: The rounding function. Valid options are 'floor', 'round' and 'ceiling'.

Value
an object of class numeric
Examples
rounder(.92, .05)
rounder(.93, .05)
rounder(.93, .05, "floor")
rounder(.93, .05, "ceiling")

rsi  Run Start Indices

Description
Find the starting indices of runs in a vector.

Usage
rsi(x)

Arguments
x a vector of data.

Value
a vector of indices indicating starting points for runs

Examples
rsi(c(0,0,0,1,2,2,3,3,3,3,3,4))

STDERR Standard Error

Description
Calculates the standard error interval of a vector of data

Usage
STDERR(x)

Arguments
x a vector of data.
**summarySE**

**Value**

- **upper**: Upper bound of interval.
- **mean**: Mean of data.
- **lower**: Lower bound of interval.

**Examples**

```r
STDERR(rnorm(100))
```

---

**SummarySE**  **Summarizes data**

**Description**

Gives count, mean, standard deviation, standard error of the mean, and confidence interval (default 95%).

**Usage**

```r
summarySE(data = NULL, measurevar, groupvars = NULL, na.rm = FALSE, conf.interval = 0.95, .drop = TRUE)
```

**Arguments**

- **data**: a data frame
- **measurevar**: the name of a column that contains the variable to be summarized
- **groupvars**: a vector containing names of columns that contain grouping variables
- **na.rm**: a boolean that indicates whether to ignore NA’s
- **conf.interval**: the percent range of the confidence interval (default is 95%)
- **.drop**: should combinations of variables that do not appear in the input data be preserved (FALSE) or dropped (TRUE, default)

**Value**

A data frame with count, mean, standard deviation, standard error of the mean, and confidence interval (default 95%).

**References**

http://www.cookbook-r.com/Graphs/Plotting_means_and_error_bars_ggplot2
**summarySEwithin**  
*Summarize within-subjects data*

**Description**
Summarizes data, handling within-subjects variables by removing inter-subject variability. It will still work if there are no within-S variables. Gives count, mean, standard deviation, standard error of the mean, and confidence interval (default 95%). If there are within-subject variables, calculate adjusted values using method from Morey (2008).

**Usage**
```
summarySEwithin(data = NULL, measurevar,  
betweenvars = NULL, withinvars = NULL, idvar = NULL,  
na.rm = FALSE, conf.interval = 0.95, drop = TRUE)
```

**Arguments**
- **data** a data frame
- **measurevar** the name of a column that contains the variable to be summarized
- **betweenvars** a vector containing names of columns that are between-subjects variables
- **withinvars** a vector containing names of columns that are within-subjects variables
- **idvar** the name of a column that identifies each subject (or matched subjects)
- **na.rm** a boolean that indicates whether to ignore NA’s
- **conf.interval** the percent range of the confidence interval (default is 95%)
- **drop** should combinations of variables that do not appear in the input data be preserved (FALSE) or dropped (TRUE, default)

**Value**
a data frame with count, mean, standard deviation, standard error of the mean, and confidence interval (default 95%).

**References**
http://www.cookbook-r.com/Graphs/Plotting_means_and_error_bars_(ggplot2)
Index

*Topic multivariate
   group.CI, 2
   group.STDERR, 3
   group.UCL, 4

*Topic univar
   CI, 2
   STDERR, 8

CI, 2

group.CI, 2

group.STDERR, 3

group.UCL, 4

lr.glover, 4

multiplot, 5

normDataWithin, 6

panel.circle, 7

rounder, 7

rsi, 8

STDERR, 8

summarySE, 9

summarySEwithin, 10