

# Package ‘Rmisc’

May 2, 2022

**Type** Package

**Title** Ryan Miscellaneous

**Version** 1.5.1

**Date** 2013-10-21

**Author** Ryan M. Hope <rmh3093@gmail.com>

**Maintainer** Ryan M. Hope <rmh3093@gmail.com>

**Description** 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** 2022-05-02 13:01:44 UTC

## 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

**Index****11**


---

CI	<i>Confidence Interval</i>
----	----------------------------

---

**Description**

Calculates the confidence interval of a vector of data.

**Usage**

```
CI(x, 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**

```
CI(rnorm(100))
```

---

group.CI	<i>Group Confidence Interval</i>
----------	----------------------------------

---

**Description**

Calculates the confidence interval of grouped data

**Usage**

```
group.CI(x, data, 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**

```
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))
)
```

---

group.STDERR

*Group Standard Error Interval*


---

**Description**

Calculates the standard error interval of grouped data.

**Usage**

```
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**

```
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	<i>Group Upper-Center-Lower</i>
-----------	---------------------------------

---

**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.lower+weight.upper,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	<i>Likelihood Ratio Test</i>
-----------	------------------------------

---

**Description**

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

**Usage**

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

**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**

Glover, S. & Dixon, P. (2004). Likelihood ratios: A simple and flexible statistic for empirical psychologists. *Psychonomic Bulletin & Review*, 11(5), 791-806.

**Examples**

```
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**

```
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

**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\\_\(ggplot2\)](http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2))

---

normDataWithin	<i>Normalize within-group data</i>
----------------	------------------------------------

---

**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**

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

**Arguments**

<code>data</code>	a data frame.
<code>idvar</code>	the name of a column that identifies each subject (or matched subjects)
<code>measurevar</code>	the name of a column that contains the variable to be summarized
<code>betweenvars</code>	a vector containing names of columns that are between-subjects variables
<code>na.rm</code>	a boolean that indicates whether to ignore NA's
<code>.drop</code>	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\\_\(ggplot2\)](http://www.cookbook-r.com/Graphs/Plotting_means_and_error_bars_(ggplot2))

---

panel.circle	<i>Circle Drawing</i>
--------------	-----------------------

---

**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	<i>Round to Increment</i>
---------	---------------------------

---

**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	<i>Run Start Indices</i>
-----	--------------------------

---

**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	<i>Standard Error</i>
--------	-----------------------

---

**Description**

Calculates the standard error interval of a vector of data

**Usage**

```
STDERR(x)
```

**Arguments**

x                    a vector of data.



**Value**

upper	Upper bound of interval.
mean	Mean of data.
lower	Lower bound of interval.

**Examples**

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

---

summarySE	<i>Summarizes data</i>
-----------	------------------------

---

**Description**

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

**Usage**

```
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\\_\(ggplot2\)](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\\_\(ggplot2\)](http://www.cookbook-r.com/Graphs/Plotting_means_and_error_bars_(ggplot2))

# Index

\* **multivariate**

group.CI, 2

group.STDERR, 3

group.UCL, 4

\* **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