

Package ‘leem’

May 8, 2026

Type Package

Title Laboratory of Teaching to Statistics and Mathematics

Version 0.2.0

Description An educational package for teaching statistics and mathematics in both primary and higher education. The objective is to assist in the teaching/learning process, both for student study planning and teacher teaching strategies. The leem package aims to provide, in a simple yet in-depth manner, knowledge of statistics and mathematics to anyone who wants to study these areas of knowledge.

Depends R (>= 4.1.0)

Imports tcltk, tkRplotR, tkrplot, manipulate, crayon, grDevices, graphics, utils, diagram, methods

License GPL (>= 2)

URL <https://bendeivide.github.io/project/leem/>,
<https://github.com/bendeivide/leem>

BugReports <https://github.com/bendeivide/leem/issues>

RoxygenNote 7.3.2

NeedsCompilation no

Author Ben Deivide [aut, cre] (ORCID: <<https://orcid.org/0000-0001-7019-8794>>),
Andre Barboza [aut] (ORCID: <<https://orcid.org/0000-0003-1190-3822>>),
Alexandre Celestino [ctb] (ORCID:
<<https://orcid.org/0000-0003-3475-3863>>),
Juliane Nassarala [ctb]

Maintainer Ben Deivide <ben.deivide@gmail.com>

Repository CRAN

Date/Publication 2025-03-08 21:50:02 UTC

Contents

amplitude	3
apf	3

barplot.leem	4
boxplot.leem	6
cdf	7
cv	8
Gumbel_distribution	9
hist.leem	10
insert	11
insert.leem	12
leem	14
madev	14
mean.leem	15
medev	16
median.leem	17
mfreq	18
mpos	19
mstde	20
new_leem	21
ogive	21
P	24
piechart	25
polyfreq	26
polyfreq.leem	26
probnormal	28
propofnormal	29
Q	30
Regions_of_probability	31
sdev	33
showboxplot	34
showcdf	34
showci	35
showkur	35
showpar	36
showskew	37
showtabnormal	37
skewness	38
stickchart	39
tabfreq	40
tabfreq.leem	41
variance	43
Index	44

amplitude	<i>Amplitude</i>
-----------	------------------

Description

Compute the sample range

Usage

```
amplitude(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
details	Logical object. Details of data (default FALSE).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data
set.seed(10)
rpois(30, 2.5) |>
  new_leem() |>
  amplitude(grouped = FALSE)
# Example 2: Normal data
rnorm(50, 100, 2.5) |>
  new_leem(variable = 2) |>
  amplitude()
```

apf	<i>Plot of probability function of any discrete variable</i>
-----	--

Description

Help in building the plot of the probability function of any discrete variable

Usage

```
apf(x, p, main = NULL, xlab = NULL, ylab = NULL)
```

Arguments

x	numeric vector of values of X . See Details .
p	numeric vector of $p_X(x)$. See Details .
main	main title for the plot.
xlab	a label for the x axis.
ylab	a label for the y axis.

Details

Consider the X distribution:

$p_X(x)$:	0.23	0.27	0.30	0.12	0.08
x :	1	2	3	4	5

where $p_X(x)$ and x are probability function and values of X . See *Example 1*.

Value

The output is plot of distribution function. See *Example 1*.

Examples

```
# Example 1
x <- 1:5
p <- c(0.23, 0.27, 0.30, 0.12, 0.08)
apf(x, p)
```

barplot.leem

Barplot graph

Description

Class method leem for generic barplot

Usage

```
## S3 method for class 'leem'
barplot(
  height,
  freq = "a",
  bg = TRUE,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  grids = grid(col = "white"),
```

```

    bgcol = "gray",
    bgborder = NA,
    barcol = "yellow",
    barborder = "gray",
    posx1 = 0,
    posx2 = 0,
    xang = 0,
    labels = NULL,
    ...
)

```

Arguments

height	R object (list) of class leem. Use new_leem() function.
freq	Character argument. Type of frequency with options: "a" (absolute and default), "r" relative and "p" percentage.
bg	Logical argument. Default is TRUE, it displays the background, and bg = FALSE otherwise.
main	Character argument. Insert the plot title. The default is NULL.
xlab	Character argument. Insert the title of the x-axis graphic label. The default is NULL.
ylab	Character argument. Insert the title of the y-axis graphic label. The default is NULL.
grids	Insert grids to plot. The default is grid(col = "white").
bgcol	Insert the background color. This argument is only valid when bg = TRUE. The default is bgcol="gray".
bgborder	Character argument. Insert the background border color. This argument is only valid when bg = TRUE. The default is bgborder = NA.
barcol	Character argument. Insert the barplot color. The default is barcol = "yellow". This argument is only valid when bars = TRUE.
barborder	Numeric argument. Insert the barplot border color. This argument is only valid when bars = TRUE. The default is barborder = "gray".
posx1	Numeric argument. Distance of the labels (horizontal) in relation to the x axis.
posx2	Numeric argument. Distance of the labels (vertical) in relation to the x axis.
xang	Numeric argument. Angle of the labels in relation to the x axis
labels	Character argument. Labels name vector.
...	further arguments passed to or from other methods.

Examples

```

library(graphics)
# Example 1 - Simple example
library(leem)
rep(1:5, 5:1) |>
  new_leem() |>

```

```

    barplot()
# Example 2 - Color bars
rep(1:5, 5:1) |>
  new_leem() |>
  barplot(barcol = heat.colors(5))
# Example 3 - Ordered data
library(leem)
school <- rep(c("high", "university", "basic"), 3:5)
sample(school, 30, TRUE) |>
  new_leem() |>
  tabfreq(ordered = c("basic", "high", "university")) |>
  barplot(xang = 15, posx2 = -0.2)
# Example 4 - Coerced to histogram
rnorm(100, 10, 2) |>
  new_leem(variable = 2) |>
  barplot(barcol = heat.colors(10))

```

 boxplot.leem

Box plot

Description

Produce box-and-whisker plot(s) of leem class object and computes the necessary values for the development of the plot.

Usage

```

## S3 method for class 'leem'
boxplot(
  x,
  type = "rawdata",
  details = FALSE,
  horizontal = FALSE,
  coef = 1.5,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  col = rgb(0, 175, 239, maxColorValue = 255),
  ...
)

```

Arguments

x	R object (list) of class leem. Use new_leem() function.
type	character argument. Default is rawdata. If type = "classes", the function returns a boxplot plot for each set of data grouped of classes of x object.
details	Logical argument. Default is FALSE, otherwise, in addition to the plot, the measurements necessary for the development of the plot will be displayed on the console.

horizontal	Logical argument indicating if the boxplots should be horizontal; default FALSE means vertical boxes.
coef	this determines how far the plot whiskers extend out from the box. If coef is positive, the whiskers extend to the most extreme data point which is no more than coef times the interquartile range from the box. A value of zero causes the whiskers to extend to the data extremes.
main	Title name. Defaults is NULL.
xlab	a label for the x axis. Defaults is NULL.
ylab	a label for the y axis. Defaults is NULL.
col	Character vector. Default col = rgb(0, 175, 239, maxColorValue = 255).
...	further arguments passed to or from other methods.

Examples

```
library(leem)
# Example 1
x <- rnorm(30, 100, 2) |>
  new_leem(variable = 2) |>
  tabfreq()
boxplot(x, details = TRUE)
# Example 2
boxplot(x, type = "classes")
```

cdfd

Plot of cumulative distribution function of any discrete variable

Description

Help in building the plot of the cumulative distribution function of any discrete variable

Usage

```
cdfd(x, fda, main = NULL, xlab = NULL, ylab = NULL)
```

Arguments

x	numeric vector of values of X . See Details .
fda	numeric vector of $F_X(x)$. See Details .
main	main title for the plot.
xlab	a label for the x axis.
ylab	a label for the y axis.

Details

Consider the X distribution:

$$\begin{array}{rcccccc} p_X(x): & 0.23 & 0.27 & 0.30 & 0.12 & 0.08 \\ x: & 1 & 2 & 3 & 4 & 5 \end{array}$$

where $p_X(x)$ and x are probability function and values of X . Consider also the X distribution function:

$$F_X(x) = \begin{cases} 0, & \text{if } x < 1; \\ 0.23, & \text{if } 1 \leq x < 2; \\ 0.50, & \text{if } 2 \leq x < 3; \\ 0.80, & \text{if } 3 \leq x < 4; \\ 0.92, & \text{if } 4 \leq x < 5; \\ 1.00 & \text{if } x \geq 5. \end{cases}$$

This way, the `cdfd` function needs to consider only the vectors `x <- 1:5` and `fda <- c(0.23, 0.50, 0.80, 0.92, 1)`, that is, only the equality conditions for x . See *Example 1*.

Value

The output is plot of distribution function. See *Example 1*.

Examples

```
# Example 1
x <- 1:5
fda <- c(0.23, 0.5, 0.8, 0.92, 1)
cdfd(x, fda)
```

 cv

Coefficient of variation

Description

Compute the sample coefficient of variation

Usage

```
cv(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

<code>x</code>	R object (list) of class <code>leem</code> . Use <code>new_leem()</code> function.
<code>rounding</code>	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>details</code>	Logical object. Details of data (default FALSE).
<code>grouped</code>	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data
rpois(30, 2.5) |>
  new_leem() |>
  cv()
# Example 2: Normal data
rnorm(50, 100, 2.5) |>
  new_leem(variable = 2) |>
  cv(grouped = FALSE)
```

Gumbel_distribution *The Gumbel Distribution*

Description

Density, distribution function, quantile function and random generation for the normal distribution with parameters: location and scale

Usage

```
dgumbel(x, location, scale)

pgumbel(q, location, scale, lower.tail = TRUE)

qgumbel(p, location = 0, scale = 1, lower.tail = TRUE)
```

Arguments

<code>x, q</code>	vector of quantiles.
<code>location</code>	numerical. It represents location parameter. See Details.
<code>scale</code>	numerical. It represents scale parameter. See Details.
<code>lower.tail</code>	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$.
<code>p</code>	vector of probabilities.

Details

The CDF of Gumbel distribution is:

$$F(x; \mu, \beta) = e^{-e^{-(x-\mu)/\beta}}, \quad \mu \in \mathbf{R}, \beta > 0,$$

where μ is location parameter (location) and β is scale parameter (scale). The PDF of Gumbel distribution is:

$$\frac{1}{\beta} e^{-(z+e^{-z})},$$

where $z = \frac{x-\mu}{\beta}$. The quantile is:

$$\mu - \beta \ln(-\ln(p)), \quad 0 < p < 1.$$

Examples

```
# PDF
dgumbel(1, 0, 1)
# CDF
pgumbel(1, 0, 1)
# Quantile
qgumbel(0.2, 0, 1)
```

hist.leem

Histogram graph

Description

Class method leem for generic hist

Usage

```
## S3 method for class 'leem'
hist(
  x,
  freq = "a",
  bg = TRUE,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  grids = grid(col = "white"),
  bgcol = "gray",
  bgborder = NA,
  barcol = "yellow",
  barborder = "gray",
  ...
)
```

Arguments

x	R object (list) of class leem. Use <code>new_leem()</code> function.
freq	Character argument. Type of frequency with options: "a" (absolute and default), "r" relative and "p" percentage.
bg	Logical argument. Default is TRUE, it displays the background, and <code>bg = FALSE</code> otherwise.
main	Insert the plot title. The default is NULL.
xlab	Insert the title of the x-axis graphic label. The default is NULL.
ylab	Insert the title of the y-axis graphic label. The default is NULL.
grids	Insert grids to plot. The default is <code>grid(col = "white")</code> .
bgcol	Insert the background color. This argument is only valid when <code>bg = TRUE</code> . The default is <code>bgcol="gray"</code> .
bgborder	Insert the background border color. This argument is only valid when <code>bg = TRUE</code> . The default is <code>bgborder = NA</code> .
barcol	Insert the barplot color. The default is <code>barcol = "yellow"</code> . This argument is only valid when <code>bars = TRUE</code> .
barborder	Numeric argument. Insert the barplot border color. This argument is only valid when <code>bars = TRUE</code> . The default is <code>barborder = "gray"</code> .
...	further arguments passed to or from other methods.

Examples

```
# Example 1
library(leem)
rnorm(36, 100, 50) |> new_leem(variable = "continuous") |> tabfreq() |> hist()

# Example 2
library(leem)
school <- rep(c("high", "university", "basic"), 3:5)
sample(school, 30, TRUE) |>
  new_leem() |>
  tabfreq(ordered = c("basic", "high", "university"))
```

insert

Insert measures of position in plot

Description

Generic function that allows inserting measures of position in plots

Usage

```
insert(x, ...)
```

Arguments

x R object (list) of class leem. Use new_leem() function.
 ... further arguments passed to or from other methods.

Examples

```
# Example 1
library(leem)
set.seed(10)
rnorm(36, 100, 50) |>
new_leem(variable = "continuous") |>
tabfreq() |>
hist() |>
insert(
  lcol = "black",
  tcol = "purple",
  acol = "brown",
  parrow = 0.6,
  larrow = 0.6,
  ptext = 0.4,
  side = "left",
  lwd = 2,
  lwdarrow = 4
)
```

insert.leem

Insert measures of position in plot

Description

Method of insert function

Usage

```
## S3 method for class 'leem'
insert(
  x,
  type = "mean",
  lty = 1,
  lcol = "black",
  tcol = lcol,
  acol = lcol,
  parrow = 0.5,
  larrow = 0.6,
  ptext = 0.06,
  side = "right",
  lwd = 2,
  lwdarrow = lwd,
```

```
    ...
  )
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
type	Type of measure of position. The default is type = "mean". Other options: "median", "mode" or "all".
lty	Line type. The default is lty = 1.
lcol	Vertical line color type. The default is lcol = "black". This argument must be the same length as the type argument.
tcol	Text color type. The default is tcol = lcol.
acol	Arrow color type. The default is acol = lcol.
parrow	Text and arrow height. The default is parrow = 0.5. This argument must be the same length as the type argument.
larrow	Text and arrow length. The default is larrow = 0.6.
ptext	Distance between lines of text. The default is ptext = 0.06.
side	Side to insert the text. The default is side = "right". This argument must be the same length as the type argument.
lwd	numeric argument. The vertical line width. The default is lwd = 2.
lwdarrow	numeric argument. The arrow width. The default is lwdarrow = lwd.
...	further arguments passed to or from other methods.

Value

No return value. This function adds elements to an existing plot.

Examples

```
# Example 1
library(leem)
set.seed(10)
rnorm(36, 100, 50) |>
new_leem(variable = "continuous") |>
tabfreq() |>
hist() |>
insert(
  lcol = "black",
  tcol = "purple",
  acol = "brown",
  parrow = 0.6,
  larrow = 0.6,
  ptext = 0.4,
  side = "left",
  lwd = 2,
  lwdarrow = 4
)
```

leem

Graphical User Interface for leem package

Description

leem A Graphical User Interface (GUI) for the leem package

Usage

```
leem(gui = TRUE)
```

Arguments

gui Logical argument, TRUE or FALSE. The default is TRUE

Value

leem presents GUI with various problems for the teaching of statistics and mathematics. The idea is to use this package to learn these subjects without necessarily programming in R

Examples

```
# Loading package
library(leem)
if (interactive()) {
  leem(gui = FALSE)
}
```

madev*Mean absolute deviation*

Description

Compute the sample mean absolute deviation

Usage

```
madev(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
details	Logical object. Details of data (default FALSE).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data
set.seed(10)
rpois(30, 2.5) |>
  new_leem() |>
  madev(grouped = FALSE)
# Example 2: Normal data
rnorm(50, 100, 2.5) |>
  new_leem(variable = 2) |>
  madev()
```

mean.leem

Arithmetic mean Class method leem for the generic mean function

Description

Arithmetic mean Class method leem for the generic mean function

Usage

```
## S3 method for class 'leem'
mean(
  x,
  trim = 0,
  na.rm = FALSE,
  rounding = 2,
  grouped = TRUE,
  details = FALSE,
  ...
)
```

Arguments

x	R object (list) of class leem. Use <code>new_leem()</code> function. Complex vectors are allowed for <code>trim = 0</code> , only.
trim	The fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).
details	Logical object. Details of data (default FALSE).
...	further arguments passed to or from other methods.

Examples

```
# Example 1
set.seed(10)
x <- rnorm(36, 100, 50)
y <- rbinom(36, 10, 0.8)
y |> new_leem(variable = "discrete") |> tabfreq() |> mean()
x |> new_leem(variable = "continuous") |> tabfreq() |> mean()
```

medev

Median absolute deviation

Description

Compute the sample median absolute deviation

Usage

```
medev(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

x	R object (list) of class leem. Use <code>new_leem()</code> function.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
details	Logical object. Details of data (default FALSE).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data
set.seed(10)
rpois(30, 2.5) |>
  new_leem() |>
  medev(grouped = FALSE)
# Example 2: Normal data
rnorm(50, 100, 2.5) |>
  new_leem(variable = 2) |>
  medev()
```

median.leem	<i>Median value</i>
-------------	---------------------

Description

Class method leem for the generic median function

Usage

```
## S3 method for class 'leem'
median(x, na.rm = FALSE, rounding = 2, grouped = TRUE, details = FALSE, ...)
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).
details	Logical object. Details of data (default FALSE).
...	further arguments passed to or from other methods.

Examples

```
library(leem)
library(stats)
# Examples
rnorm(36, 100, 50) |> new_leem(variable = 2) |> tabfreq() |> median()
```

mfreq	<i>Mode value</i>
-------	-------------------

Description

Compute the sample mode.

Usage

```
mfreq(x, na.rm = FALSE, rounding = 2, grouped = TRUE, details = FALSE)
```

Arguments

<code>x</code>	R object (list) of class <code>leem</code> . Use <code>new_leem()</code> function.
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>rounding</code>	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
<code>grouped</code>	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).
<code>details</code>	Logical object. Details of data (default FALSE).

Examples

```
library(leem)
# set.seed(10)
x <- rnorm(36, 100, 50)
set.seed(10)
y <- rbinom(36, 10, 0.8)
w <- rep(letters[1:4], 1:4)
(tab1 <- y |> new_leem(variable = "discrete") |> tabfreq())
(tab2 <- x |> new_leem(variable = "continuous") |> tabfreq())
(tab3 <- w |> new_leem(variable = "discrete") |> tabfreq())
y |> new_leem(variable = "discrete") |> tabfreq() |> mfreq()
x |> new_leem(variable = "continuous") |> tabfreq() |> mfreq()
w |> new_leem(variable = "discrete") |> tabfreq() |> mfreq()
```

mpos	<i>Measures of position</i>
------	-----------------------------

Description

Compute all measures of position

Usage

```
mpos(  
  x,  
  trim = 0,  
  na.rm = FALSE,  
  rounding = 2,  
  grouped = TRUE,  
  details = FALSE,  
  ...  
)
```

Arguments

x	R object (list) of class leem. Use <code>new_leem()</code> function. Complex vectors are allowed for <code>trim = 0</code> , only.
trim	The fraction (0 to 0.5) of observations to be trimmed from each end of x before the mean is computed. Values of trim outside that range are taken as the nearest endpoint.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).
details	Logical object. Details of data (default FALSE).
...	further arguments passed to or from other methods.

Details

The measures of position are: average, median and mode.

Examples

```
# Example 1: Poisson data  
rpois(30, 2.5) |>  
  new_leem() |>  
  mpos()  
# Example 2: Normal data
```

```
rnorm(50, 100, 2.5) |>  
  new_leem(variable = 2) |>  
  mpos(grouped = FALSE)
```

mstde	<i>Mean standard error</i>
-------	----------------------------

Description

Compute the sample mean standard error

Usage

```
mstde(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
details	Logical object. Details of data (default FALSE).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data  
set.seed(10)  
rpois(30, 2.5) |>  
  new_leem() |>  
  mstde(rounding = 4)  
# Example 2: Normal data  
rnorm(50, 100, 2.5) |>  
  new_leem(variable = 2) |>  
  mstde(grouped = FALSE)
```

new_leem	<i>Constructor of object of leem class</i>
----------	--

Description

Function that assists other functions of leem package

Usage

```
new_leem(x, variable = "discrete")
```

Arguments

x	R object (vector as data structure).
variable	Type of data. If discrete (default), the data are categorical (numeric or not). If continuous, the data are numeric.

Value

The variable argument also allows using variable = 1 for categorical variable and variable = 2 for continuous variable.

Examples

```
# Example 1
library(leem)
x <- rbinom(36, 10, 0.6)
new_leem(x, variable = 1)

# Example 2 (Pipe operator)
rnorm(36, 100, 4) |> new_leem(variable = 2)
```

ogive	<i>Ogive chart</i>
-------	--------------------

Description

Generic function that plots the culmulative frequency curve.

Usage

```
ogive(x, ...)

## S3 method for class 'leem'
ogive(
  x,
  freq = "a",
  decreasing = FALSE,
  both = FALSE,
  bars = FALSE,
  histogram = FALSE,
  bg = TRUE,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  grids = grid(col = "white"),
  bgcol = "gray",
  bgborder = NA,
  barcol = "yellow",
  histcol = barcol,
  barborder = "gray",
  histborder = barborder,
  type = "b",
  lpcol = "black",
  lwd = 2,
  pch = 19,
  lty = 2,
  ...
)
```

Arguments

<code>x</code>	R object (list) of class <code>leem</code> . Use <code>new_leem()</code> function.
<code>...</code>	further arguments passed to or from other methods.
<code>freq</code>	Character argument. Type of frequency with options: "a" (absolute and default), "r" relative and "p" percentage.
<code>decreasing</code>	Logical argument. Default is <code>FALSE</code> . If <code>decreasing = FALSE</code> , it represents the "ogive larger than", if <code>decreasing = TRUE</code> , it represents the "ogive less than".
<code>both</code>	Logical argument. Default is <code>FALSE</code> . If <code>both = TRUE</code> , both o will be plotted. If <code>both = FALSE</code> otherside.
<code>bars</code>	Logical argument. Default is <code>FALSE</code> . If <code>bars = TRUE</code> , the bars of the accumulated frequency will be inserted to plot, according to the <code>decreasing</code> argument. If <code>bars = FALSE</code> otherside.
<code>histogram</code>	Logical argument. Default is <code>FALSE</code> . If <code>histogram = TRUE</code> , the histogram will be inserted to plot.
<code>bg</code>	Logical argument. Default is <code>TRUE</code> , it displays the background, and <code>bg = FALSE</code> otherwise.

main	Insert the plot title. The default is NULL.
xlab	Insert the title of the x-axis graphic label. The default is NULL.
ylab	Insert the title of the y-axis graphic label. The default is NULL.
grids	Insert grids to plot. The default is <code>grid(col = "white")</code> .
bgcol	Insert the background color. This argument is only valid when <code>bg = TRUE</code> . The default is <code>bgcol="gray"</code> .
bgborder	Insert the background border color. This argument is only valid when <code>bg = TRUE</code> . The default is <code>bgborder = NA</code> .
barcol	Insert the barplot color. The default is <code>barcol = "yellow"</code> . This argument is only valid when <code>bars = TRUE</code> .
histcol	Insert the histogram color. The default is <code>histcol = barcol</code> . This argument is only valid when <code>histogram = TRUE</code> .
barborder	Insert the barplot border color. This argument is only valid when <code>bars = TRUE</code> . The default is <code>barborder = "gray"</code> .
histborder	Insert the histogram border color. This argument is only valid when <code>histogram = TRUE</code> . The default is <code>histborder = barborder</code> .
type	Type of plot. The default is <code>type = "b"</code> , i.e., line and points. See graphical parameter for details.
lpcol	Type of line color. The default is <code>lpcol = "black"</code> .
lwd	numeric argument. The line width. The default is <code>lwd = 2</code> .
pch	Type of point. The default is <code>pch = 19</code> .
lty	Type of line. The default is <code>lty = 2</code> .

Value

Ogive plot.

Examples

```
library(leem)
# Example 1 - Both ogives
rnorm(36, 100, 50) |> new_leem(variable = 2) |> tabfreq() |> ogive(both = TRUE)

# Example 2 - Insert barplot
rnorm(36, 100, 50) |> new_leem(variable = 2) |> tabfreq() |> ogive(both = TRUE, bars = TRUE)
# Example 3 - Insert histogram
rnorm(36, 100, 50) |> new_leem(variable = 2) |> tabfreq() |> ogive(both = TRUE, histogram = TRUE)
```

P *Cumulative distribution function*

Description

P Compute the cumulative distribution function for multiple distributions

Usage

```
P(
  q,
  dist = "normal",
  lower.tail = TRUE,
  rounding = 5,
  percentage = FALSE,
  gui = "plot",
  main = NULL,
  ...
)
```

Arguments

q	quantile. The q argument can have length 1 or 2. See Details.
dist	distribution to use. The default is 'normal'. Options: 'normal', 't-student', 'gumbel', 'binomial', 'poisson', and
lower.tail	logical; if TRUE (default), probabilities are $P[X \leq x]$ otherwise, $P[X > x]$. This argument is valid only if q has length 1.
rounding	numerical; it represents the number of decimals for calculating the probability.
percentage	logical; if FALSE (default), the result in decimal. Otherwise, probability is given in percentage.
gui	default is 'plot'; it graphically displays the result of the probability. Others options are: 'none', 'rstudio' or 'tcltk'.
main	default is NULL; it represents title of plot.
...	additional arguments according to the chosen distribution.

Details

The argument that can have length 2, when we use the functions that give us the probability regions, given by: $\%<X<\%$, $\%<=X<\%$, $\%<X<=\%$, $\%<=X<=\%$, $\%>X>\%$, $\%>X=>\%$, $\%>X=>\%$ and $\%>=X=>\%$. The additional arguments represent the parameters of the distributions, that is:

Value

P returns the probability and its graphical representation. The result can be given as a percentage or not.

Examples

```
# Loading package
library(leem)
# Example 1 - t-Student distribution
## Not run:
P(q = 2, dist = "t-student", df = 10)
P(q = 2, dist = "t-student", df = 10, gui = 'rstudio')
P(q = 2, dist = "t-student", df = 10, gui = 'tcltk')
P(-1 %<X<% 1, dist = "t-student", df = 10)

## End(Not run)
# Example 2 - Normal distribution
P(-2, dist = "normal", mean = 3, sd = 2,
  main = expression(f(x) == (1 / sqrt(n * sigma^2)) *
    exp(-1/2 * (x - mu)^2/sigma^2)))
```

 piechart

Pie Chart

Description

Draw a pie chart.

Usage

```
piechart(
  x,
  labels = NULL,
  col = heat.colors(5, 1),
  border = FALSE,
  main = NULL,
  ...
)
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
labels	One or more expressions or character strings giving names for the slices
col	Character vector. Default col = heat.colors(5).
border	Logical argument (default FALSE).
main	Title name.
...	further arguments passed to or from other methods.

Examples

```

library(leem)
# Example 1
school <- rep(c("high", "university", "basic"), 3:5)
x <- sample(school, 30, TRUE) |>
  new_leem() |>
  tabfreq(ordered = c("basic", "high", "university"))
# Example 2
x <- rbinom(36, 10, 0.6)
x <- new_leem(x, variable = "discrete")
x <- tabfreq(x)
piechart(x)

```

polyfreq

Frequency polygon Graph

Description

Generic function that plots the frequency polygon curve.

Usage

```
polyfreq(x, ...)
```

Arguments

`x` R object (list) of class leem. Use `new_leem()` function.
`...` further arguments passed to or from other methods.

Examples

```

# Example 1
library(leem)
rnorm(36, 100, 50) |> new_leem(variable = "continuous") |> tabfreq() |> polyfreq()

```

polyfreq.leem

Frequency polygon Graph

Description

Plot the frequency polygon curve.

Usage

```
## S3 method for class 'leem'
polyfreq(
  x,
  freq = "a",
  type = "b",
  bars = TRUE,
  bg = TRUE,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  grids = grid(col = "white"),
  bgcol = "gray",
  bgborder = NA,
  barcol = "yellow",
  barborder = "gray",
  lpcol = "black",
  lwd = 2,
  pch = 19,
  lty = 2,
  ...
)
```

Arguments

x	R object (list) of class leem. Use new_leem() function.
freq	Character argument. Type of frequency with options: "a" (absolute and default), "r" relative and "p" percentage.
type	Type of plot. The default is type = "b", i.e., line and points. See graphical parameter for details.
bars	Logical argument. Default is FALSE. If bars = TRUE, the histogram will be inserted to plot.
bg	Logical argument. Default is TRUE, it displays the background, and bg = FALSE otherwise.
main	Insert the plot title. The default is NULL.
xlab	Insert the title of the x-axis graphic label. The default is NULL.
ylab	Insert the title of the y-axis graphic label. The default is NULL.
grids	Insert grids to plot. The default is grid(col = "white").
bgcol	Insert the background color. This argument is only valid when bg = TRUE. The default is bgcol="gray".
bgborder	Insert the background border color. This argument is only valid when bg = TRUE. The default is bgborder = NA.
barcol	Insert the barplot color. The default is barcol = "yellow". This argument is only valid when bars = TRUE.

barborder	Insert the barplot border color. This argument is only valid when bars = TRUE. The default is barborder = "gray".
lpcol	Type of line color. The default is lpcol = "black".
lwd	numeric argument. The line width. The default is lwd = 2.
pch	Type of point. The default is pch = 19.
lty	Type of line. The default is lty = 2.
...	further arguments passed to or from other methods.

Examples

```
# Example 1
library(leem)
rnorm(36, 100, 50) |> new_leem(variable = "continuous") |> tabfreq() |> polyfreq()
```

probnormal

Understanding the probability of the normal distribution

Description

Using a graphical visualization, it is possible to understand the probabilities involved in a normal distribution.

Usage

```
probnormal(
  a = 1,
  b = 2,
  col = "lightblue",
  mean = 0,
  sd = 1,
  type = 1,
  rounding = 4,
  zang = 0,
  xang = 0
)
```

Arguments

a	lower limit. The default is 1.
b	upper limit. The default is 2, and b must be greater than a.
col	plot color. The default is col = "lightblue".
mean	parameter. The default is 0.
sd	parameter. The default is 1.
type	type of visualization of the probability region plot. Default is 1, others: 2, 3, 4, 5, 6. See Details.

rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 4).
zang	Angle of the values on the Z-axis. Default is zang = 0.
xang	Angle of the values on the X-axis. Default is xang = 0.

Details

- type = 1, 2: a and b must be greater than mean;
- type = 3, 4: a and b must be less than mean;
- type = 5, 6: a and b can be any real value.

Examples

```
## Not run:
probnormal(type = 2)
probnormal(-1, 0, type = 3)
probnormal(-1, 0, type = 4)
probnormal(-1, 0, type = 5)
probnormal(-1, 2, type = 5)
probnormal(1, 2, type = 5)
probnormal(1, 2, type = 6)

## End(Not run)
```

propofnormal

Properties of the normal distribution

Description

Graphically it is possible to observe some properties of the normal distribution

Usage

```
propofnormal(col = "lightblue2", type = 1)
```

Arguments

col	color type.
type	numerical. Type of properties. Options: 1, 2, 3, 4 and 5. Default code = 1.

Q

*Quantile distribution function.***Description**

Q Quantile function for multiple distributions.

Usage

```
Q(
  p,
  dist = "normal",
  lower.tail = TRUE,
  two.sided = FALSE,
  rounding = 2,
  gui = "plot",
  mfrow = c(1, 2),
  type = "both",
  ...
)
```

Arguments

p	probability. The p argument need have length 1 and value lower then 1.
dist	distribution to use. The default is 'normal'. Options: 'normal', 't-student', 'gumbel', 'binomial', 'poisson', and
lower.tail	logical; if TRUE (default), the quantile function is computed; otherwise, the complement of the quantile function (survival function) will be computed. The lower.tail argument will only be valid for two-sided = FALSE.
two.sided	logical. if TRUE (default), the calculation of the quantile function and survival will be presented; otherwise the Q() function will be based according to the lower.tail argument.
rounding	numerical; it represents the number of decimals for calculating the probability.
gui	default is 'plot'; it graphically displays the result of the probability. Others options are: "plot" and "rstudio" and "tcltk".
mfrow	numerical vector. Considering the arguments two.sided = TRUE and type = "both", the default will be to present two graphs (based on CDF and PDF) horizontally for the quantile function, that is, mfrow = c(1, 2) (default).
type	character argument. The default is "both"; the output will display two plots (based on CDF and PDF) to present the result of Q(). The other options are: "cdf" and "pdf".
...	additional parameters according to the chosen distribution.

Details

The expression of quantile function is given by:

$$Q(p) = \inf x \in \mathbb{R} : p \leq F(x),$$

where p is the first argument of $Q()$ and x its return value;

Value

Q returns the quantile and its graphical representation for a given distribution. The output is a vector.

Examples

```
# Attaching package
library(leem)
## Not run:
Q(p = 0.8, dist = "normal", mean = 200, sd=30)

## End(Not run)
```

Regions_of_probability
Regions of probability

Description

These binary operators return a vector of length 2, describing the desired probability region.

Usage

```
a %>x>% b
a %>X>% b
a %<X<% b
a %<x<% b
a %>=X>=% b
a %>=x>=% b
a %<=X<=% b
a %<=x<=% b
a %>=X>% b
```

a $\%>=x>\%$ b

a $\%>X>=\%$ b

a $\%>x>=\%$ b

a $\%<=X<\%$ b

a $\%<=x<\%$ b

a $\%<X<=\%$ b

a $\%<x<=\%$ b

Arguments

- a scalar. when referring to a discrete random variable, use the syntax "L" after the number.
- b scalar. when referring to a discrete random variable, use the syntax "L" after the number.

Value

A vector of length 2.

Examples

```
# Example 1 - Discrete
```

```
2L  $\%>x>\%$  5L
```

```
2L  $\%>X>=\%$  5L
```

```
2L  $\%<X<\%$  5L
```

```
2L  $\%<x<\%$  5L
```

```
2L  $\%>=X>=\%$  5L
```

```
2L  $\%>=x>=\%$  5L
```

```
2L  $\%<=X<=\%$  5L
```

```
2L  $\%<=x<=\%$  5L
```

```
2L  $\%>=X>\%$  5L
```

```
2L  $\%>=x>\%$  5L
```

```
2L  $\%>x>=\%$  5L
```

```
2L  $\%>X>=\%$  5L
```

```
2L  $\%<=X<\%$  5L
```

```
2L  $\%<=x<\%$  5L
```

```
2L  $\%<X<=\%$  5L
```

```
2L  $\%<x<=\%$  5L
```

```
# Example 2 - Continuous
```

```
2  $\%>x>\%$  5
```

```
2  $\%>X>\%$  5
```

```
2  $\%<X<\%$  5
```

```
2  $\%<x<\%$  5
```

```
2  $\%>=X>=\%$  5
```

```

2 %>=x>=% 5
2 %<=X<=% 5
2 %<=x<=% 5
2 %>=X>% 5
2 %>=x>% 5
2 %>x>=% 5
2 %>X>=% 5
2 %<=X<% 5
2 %<=x<% 5
2 %<X<=% 5
2 %<x<=% 5

```

sdev

Standard Deviation

Description

Compute the sample standard deviation

Usage

```
sdev(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

x	R object (list) of class leem. Use <code>new_leem()</code> function.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
details	Logical object. Details of data (default FALSE).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```

# Example 1: Poisson data
rpois(30, 2.5) |>
  new_leem() |>
  sdev()
# Example 2: Normal data
rnorm(50, 100, 2.5) |>
  new_leem(variable = 2) |>
  sdev(grouped = FALSE)

```

showboxplot *Understanding a box plot*

Description

Detailing of a box plot, showing the main information contained in this type of graph

Usage

```
showboxplot(horizontal = FALSE, col = rgb(0, 175, 239, maxColorValue = 255))
```

Arguments

horizontal Logical argument indicating if the boxplots should be horizontal; default FALSE means vertical boxes.

col Character vector. Default col = rgb(0, 175, 239, maxColorValue = 255).

Examples

```
library(leem)
# Example 1
showboxplot()
```

showcdf *Distribution Function Properties*

Description

Graphic presentation of properties for distribution function

Usage

```
showcdf(variable = "discrete", prop = NULL)
```

Arguments

variable Variable type. Defaults variable = "discrete". The options are: discrete or "1" and continuous or "2".

prop Properties for distribution function. See Details.

Details

- prop = 1: $\lim_{x \rightarrow \infty} F_X(x) = 1$ and $\lim_{x \rightarrow -\infty} F_X(x) = 0$;
- prop = 2: $F_X(x) \leq F_X(y)$, $x \leq y \forall x, y \in \mathbb{R}$;
- prop = 3: $\lim_{x_n \downarrow x} F_X(x_n) \downarrow F_X(x)$.

Examples

```
library(leem)
# Example 1
showcdf()
```

showci

Understanding the Confiance Indice

Description

Detailing the confiance indice plot, showing the main information contained in this type of graph.

Usage

```
showci(dist = "normal", ci = "two.sided", main = NULL)
```

Arguments

dist	Parameter to indicate the distribution of the graphic, fixed for now.
ci	Parameter to indicate the region of the confiance indice.
main	Parameter to indicate the title of the graphic.

Examples

```
library(leem)
# Example 1
showci()
```

showkur

Plot of interpretation about Kurtosis

Description

showkur Interpretation of kutosis

Usage

```
showkur()
```

Value

showkur returns a plot with the kurtosis characteristics.

Examples

```
# Loading package
library(leem)
## Not run:
showkur()

## End(Not run)
```

showpar

Interpretation of location and scale parameters

Description

showpar Function that exemplifies the interpretation of location and scale parameters

Usage

```
showpar(gui = "rstudio")
```

Arguments

gui character argument. The options are: "rstudio" (default) and "tcltk".

Details

The result of the showpar() call will interactively present a plot of the normal distribution showing the behavior of the location and scale parameters via RStudio. For showpar(gui = "tcltk") the result will be displayed in a tcltk interface.

Value

showpar returns an interactive plot.

Examples

```
# Loading package
library(leem)
## Not run:
showpar()

## End(Not run)
```

showskew	<i>Plot of interpretation about skewness</i>
----------	--

Description

showskew Interpretation of asymmetry based on frequency distributions

Usage

```
showskew(mpos = FALSE)
```

Arguments

mpos Logical. It shows the measures of position or not (default FALSE).

Examples

```
# Loading package
library(leem)
## Not run:
showskew()

## End(Not run)
```

showtabnormal	<i>showtabnormal</i>
---------------	----------------------

Description

Detailing of the Ztable, showing the main information contained in this type of table.

Usage

```
showtabnormal(z)
```

Arguments

z Parameter for locate the z value on the table; default NULL means error if not insert a number.

Examples

```
library(leem)
# Example 1
showtabnormal(1)
```

skewness	<i>Skewness value</i>
----------	-----------------------

Description

Compute the skewness

Usage

```
skewness(  
  x,  
  type = "pearson",  
  rounding = 2,  
  na.rm = FALSE,  
  details = FALSE,  
  grouped = TRUE  
)
```

Arguments

<code>x</code>	R object (list) of class leem. Use <code>new_leem()</code> function.
<code>type</code>	character. methodology addressed. For now, we only have the "pearson" option (default).
<code>rounding</code>	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
<code>na.rm</code>	a logical value indicating whether NA values should be stripped before the computation proceeds.
<code>details</code>	Logical object. Details of data (default FALSE).
<code>grouped</code>	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data  
rpois(30, 2.5) |>  
  new_leem() |>  
  skewness()  
# Example 2: Normal data  
rnorm(50, 100, 2.5) |>  
  new_leem(variable = 2) |>  
  skewness(grouped = TRUE)
```

`stickchart`*Stick chart*

Description

Stick chart for discrete data

Usage

```
stickchart(  
  x,  
  freq = "a",  
  bg = TRUE,  
  main = NULL,  
  xlab = NULL,  
  ylab = NULL,  
  grids = grid(col = "white"),  
  bgcol = "gray",  
  bgborder = NA,  
  posx1 = 0,  
  posx2 = 0,  
  xang = 0,  
  labels = NULL,  
  lcol = "black",  
  pcol = lcol,  
  pty = 19,  
  pwd = 3,  
  lty = 1,  
  lwd = 2,  
  ...  
)
```

Arguments

<code>x</code>	R object (list) of class <code>leem</code> . Use <code>new_leem()</code> function.
<code>freq</code>	Character argument. Type of frequency with options: "a" (absolute and default), "r" relative and "p" percentage.
<code>bg</code>	Logical argument. Default is TRUE, it displays the background, and <code>bg = FALSE</code> otherwise.
<code>main</code>	Insert the plot title. The default is NULL.
<code>xlab</code>	Insert the title of the x-axis graphic label. The default is NULL.
<code>ylab</code>	Insert the title of the y-axis graphic label. The default is NULL.
<code>grids</code>	Insert grids to plot. The default is <code>grid(col = "white")</code> .
<code>bgcol</code>	Insert the background color. This argument is only valid when <code>bg = TRUE</code> . The default is <code>bgcol="gray"</code> .

<code>bgborder</code>	Insert the background border color. This argument is only valid when <code>bg = TRUE</code> . The default is <code>bgborder = NA</code> .
<code>posx1</code>	Numeric argument. Distance of the labels (horizontal) in relation to the x axis.
<code>posx2</code>	Numeric argument. Distance of the labels (vertical) in relation to the x axis.
<code>xang</code>	Numeric argument. Angle of the labels in relation to the x axis
<code>labels</code>	Character argument. Labels name vector.
<code>lcol</code>	Line color. The default is <code>lcol = "black"</code> .
<code>pcol</code>	Point color. The default is <code>pcol = lcol</code> .
<code>pty</code>	Point type. The default is <code>pty = 19</code> .
<code>pwd</code>	Point width. The default is <code>pwd = 3</code> .
<code>lty</code>	Line type. The default is <code>lty = 2</code> .
<code>lwd</code>	Line width. The default is <code>lwd = 2</code> .
<code>...</code>	further arguments passed to or from other methods.

Value

The result of `stickchart()` is x object.

Examples

```
library(leem)
# Example 1
rbinom(30, 10, 0.4) |>
  new_leem() |>
  tabfreq() |>
  stickchart()
# Example 2
school <- rep(c("high", "university", "basic"), 3:5)
sample(school, 30, TRUE) |>
  new_leem() |>
  tabfreq(ordered = c("basic", "high", "university")) |>
  stickchart(xang = 15, posx2 = -0.5)
```

<code>tabfreq</code>	<i>Frequency distribution table</i>
----------------------	-------------------------------------

Description

Generic function that allows you to tabulate continuous and categorical data (quantitative or qualitative) in frequency distribution. Depending on the nature of the data, they can be grouped into class ranges or not.

Usage

```
tabfreq(data, ...)
```

Arguments

`data` R object (data structure vector) of class leem. Use `new_leem()` function.
`...` further arguments passed to or from other methods.

Value

The result of `tabfreq()` is a list. This list has two elements: `table` and `statistics`. The first is the data frequency table, and the second represents some useful statistics for methods of leem class.

Examples

```
# Example 1
library(leem)
x <- rbinom(36, 10, 0.6)
x <- new_leem(x, variable = "discrete")
tabfreq(x)

# Example 2 (Pipe operator)
rnorm(36, 100, 4) |>
  new_leem(variable = "continuous") |> tabfreq()

# Example 3
x <- rbinom(36, 10, 0.6)
# Constructor (object of leem class)
x <- new_leem(x, variable = "discrete")
tab <- tabfreq(x)
# Details
tab$table
tab$statistics

# Example 3 - ordered categories ("d","a", "b", "c")
w <- rep(letters[1:4], 1:4)
w |> new_leem(variable = "discrete") |> tabfreq(ordered = c("d", "a", "b", "c"))
```

tabfreq.leem

Frequency distribution table

Description

Allows you to tabulate continuous and categorical data (quantitative or qualitative) in frequency distribution. Depending on the nature of the data, they can be grouped into class ranges or not.

Usage

```
## S3 method for class 'leem'
tabfreq(
  data,
```

```

    k = NULL,
    na.rm = FALSE,
    ordered = NULL,
    namereduction = TRUE,
    ...
  )

```

Arguments

<code>data</code>	R object (data structure vector) of class <code>leem</code> . Use <code>new_leem()</code> function.
<code>k</code>	Number of classes. Default is <code>NULL</code> .
<code>na.rm</code>	a logical evaluating to <code>TRUE</code> or <code>FALSE</code> indicating whether NA values should be stripped before the computation proceeds.
<code>ordered</code>	Ordered vector of the same length and elements of data object. Default is <code>NULL</code> .
<code>namereduction</code>	Logical argument. If <code>TRUE</code> (default), the group names are reduced the 10 characters. If <code>FALSE</code> , otherwise.
<code>...</code>	further arguments passed to or from other methods.

Value

The result of `tabfreq()` is a list. This list has two elements: `table` and `statistics`. The first is the data frequency table, and the second represents some useful statistics for methods of `leem` class.

Examples

```

# Example 1
library(leem)
x <- rbinom(36, 10, 0.6)
x <- new_leem(x, variable = "discrete")
tabfreq(x)

# Example 2 (Pipe operator)
rnorm(36, 100, 4) |>
  new_leem(variable = "continuous") |> tabfreq()

# Example 3
x <- rbinom(36, 10, 0.6)
# Constructor (object of leem class)
x <- new_leem(x, variable = "discrete")
tab <- tabfreq(x)
# Details
tab$table
tab$statistics

# Example 3 - ordered categories ("d", "a", "b", "c")
w <- rep(letters[1:4], 1:4)
w |> new_leem(variable = "discrete") |> tabfreq(ordered = c("d", "a", "b", "c"))

```

variance	<i>Variance value</i>
----------	-----------------------

Description

Compute the sample variance

Usage

```
variance(x, rounding = 2, na.rm = FALSE, details = FALSE, grouped = TRUE)
```

Arguments

x	R object (list) of class leem. Use <code>new_leem()</code> function.
rounding	Numerical object. Rounds the values in its first argument to the specified number of decimal places (default 2).
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
details	Logical object. Details of data (default FALSE).
grouped	Logical object. Determines whether the measure of position result will be based on grouped data or not (default TRUE).

Examples

```
# Example 1: Poisson data
rpois(30, 2.5) |>
  new_leem() |>
  variance()
# Example 2: Normal data
rnorm(50, 100, 2.5) |>
  new_leem(variable = 2) |>
  variance(grouped = FALSE)
```

Index

`%<=X<=%` (Regions_of_probability), 31
`%<=X<%` (Regions_of_probability), 31
`%<=x<=%` (Regions_of_probability), 31
`%<=x<%` (Regions_of_probability), 31
`%<X<=%` (Regions_of_probability), 31
`%<X<%` (Regions_of_probability), 31
`%<x<=%` (Regions_of_probability), 31
`%<x<%` (Regions_of_probability), 31
`%>=X>=%` (Regions_of_probability), 31
`%>=X>%` (Regions_of_probability), 31
`%>=x>=%` (Regions_of_probability), 31
`%>=x>%` (Regions_of_probability), 31
`%>X>=%` (Regions_of_probability), 31
`%>X>%` (Regions_of_probability), 31
`%>x>=%` (Regions_of_probability), 31
`%>x>%` (Regions_of_probability), 31

`amplitude`, 3
`apf`, 3

`barplot.leem`, 4
`boxplot.leem`, 6

`cdfd`, 7
`cv`, 8

`dgumbel` (Gumbel_distribution), 9

Gumbel_distribution, 9

`hist.leem`, 10

`insert`, 11
`insert.leem`, 12

`leem`, 14

`madev`, 14
`mean.leem`, 15
`medev`, 16
`median.leem`, 17

`mfreq`, 18
`mpos`, 19
`mstde`, 20

`new_leem`, 21

`ogive`, 21

P, 24
`pgumbel` (Gumbel_distribution), 9
`piechart`, 25
`polyfreq`, 26
`polyfreq.leem`, 26
`probnormal`, 28
`propofnormal`, 29

Q, 30
`qgumbel` (Gumbel_distribution), 9

Regions_of_probability, 31

`sdev`, 33
`showboxplot`, 34
`showcdf`, 34
`showci`, 35
`showkur`, 35
`showpar`, 36
`showskew`, 37
`showtabnormal`, 37
`skewness`, 38
`stickchart`, 39

`tabfreq`, 40
`tabfreq.leem`, 41

`variance`, 43