

hist {graphics}
R Documentation

Histograms

Description

The generic function `hist` computes a histogram of the given data values. If `plot=TRUE`, the resulting object of `class` "histogram" is plotted by `plot.histogram`, before it is returned.

Usage

```
hist(x, ...)
```

```
## Default S3 method:
```

```
hist(x, breaks = "Sturges", freq = NULL, probability = !  
freq,  
      include.lowest = TRUE, right = TRUE,  
      density = NULL, angle = 45, col = NULL, border =  
NULL,  
      main = paste("Histogram of" , xname),  
      xlim = range(breaks), ylim = NULL,  
      xlab = xname, ylab,  
      axes = TRUE, plot = TRUE, labels = FALSE,  
      nclass = NULL, ...)
```

Arguments

`x`

a vector of values for which the histogram is desired.

`breaks`

one of:

- a vector giving the breakpoints between histogram cells,
- a single number giving the number of cells for the histogram,
- a character string naming an algorithm to compute the number of cells (see Details),
- a function to compute the number of cells.

In the last three cases the number is a suggestion only.

`freq`

logical; if `TRUE`, the histogram graphic is a representation of frequencies, the `counts` component of the result; if `FALSE`, *relative* frequencies (“probabilities”), component `density`, are plotted. Defaults to `TRUE` *iff* `breaks` are equidistant (and `probability` is not specified).

`probability`

an *alias* for `!freq`, for S compatibility.

`include.lowest`

logical; if `TRUE`, an `x[i]` equal to the `breaks` value will be included in the first (or last, for `right = FALSE`) bar. This will be ignored (with a warning) unless `breaks` is a vector.

`right`

logical; if `TRUE`, the histograms cells are right-closed (left open) intervals.

`density`

the density of shading lines, in lines per inch. The default value of `NULL` means that no shading lines are drawn. Non-positive values of `density` also inhibit the drawing of shading lines.

`angle`

the slope of shading lines, given as an angle in degrees (counter-clockwise).

`col`

a colour to be used to fill the bars. The default of `NULL` yields unfilled bars.

`border`

the color of the border around the bars. The default is to use the standard foreground color.

`main, xlab, ylab`

these arguments to `title` have useful defaults here.

`xlim, ylim`

the range of `x` and `y` values with sensible defaults. Note that `xlim` is *not* used to define the histogram (`breaks`), but only for plotting (when `plot = TRUE`).

`axes`

logical. If `TRUE` (default), axes are drawn if the plot is drawn.

`plot`

logical. If `TRUE` (default), a histogram is plotted, otherwise a list of breaks and counts is returned.

`labels`

logical or character. Additionally draw labels on top of bars, if not `FALSE`; see `plot.histogram`.

`nclass`

numeric (integer). For S(-PLUS) compatibility only, `nclass` is equivalent to `breaks` for a scalar or character argument.

...

further graphical parameters to `title` and `axis`.

Details

The definition of “histogram” differs by source (with country-specific biases). `R`'s default with equi-spaced breaks (also the default) is to plot the counts in the cells defined by `breaks`. Thus the height of a rectangle is proportional to the number of points falling into the cell, as is the area *provided* the breaks are equally-spaced.

The default with non-equi-spaced breaks is to give a plot of area one, in which the *area* of the rectangles is the fraction of the data points falling in the cells.

If `right = TRUE` (default), the histogram cells are intervals of the form $(a, b]$, i.e., they include their right-hand endpoint, but not their left one, with the exception of the first cell when `include.lowest` is `TRUE`.

For `right = FALSE`, the intervals are of the form $[a, b)$, and `include.lowest` really has the meaning of “*include highest*”.

A numerical tolerance of $1e-7$ times the median bin size is applied when counting entries on the edges of bins.

The default for `breaks` is "Sturges": see `nclass.Sturges`. Other names for which algorithms are supplied are "Scott" and "FD" / "Friedman-Diaconis" (with corresponding functions `nclass.scott` and `nclass.FD`). Case is ignored and partial matching is used. Alternatively, a function can be supplied which will compute the intended number of breaks as a function of `x`.

Value

an object of class "histogram" which is a list with components:

`breaks`

the $n+1$ cell boundaries (= `breaks` if that was a vector).

`counts`

n integers; for each cell, the number of `x[]` inside.

`density`

values $f^{\wedge}(x[i])$, as estimated density values. If `all(diff(breaks) == 1)`, they are the relative frequencies `counts/n` and in general satisfy $\sum[i; f^{\wedge}(x[i]) (b[i+1]-b[i])] = 1$, where $b[i] = \text{breaks}[i]$.

`intensities`

same as `density`. Deprecated, but retained for compatibility.

`mids`

the n cell midpoints.

`xname`

a character string with the actual `x` argument name.

`equidist`

logical, indicating if the distances between `breaks` are all the same.

Note

The resulting value does *not* depend on the values of the arguments `freq` (or `probability`) or `plot`. This is intentionally different from `S`.

Prior to R 1.7.0, the element `breaks` of the result was adjusted for numerical tolerances. The nominal values are now returned even

though tolerances are still used when counting.

References

Becker, R. A., Chambers, J. M. and Wilks, A. R. (1988) *The New S Language*. Wadsworth & Brooks/Cole.

Venables, W. N. and Ripley, B. D. (2002) *Modern Applied Statistics with S*. Springer.

See Also

[nclass.Sturges](#), [stem](#), [density](#), [truehist](#) in package MASS.

Examples

```
op <- par(mfrow=c(2, 2))
hist(islands)
utils::str(hist(islands, col="gray", labels = TRUE))

hist(sqrt(islands), br = 12, col="lightblue",
border="pink")
##-- For non-equidistant breaks, counts should NOT be
graphed unscaled:
r <- hist(sqrt(islands), br = c(4*0:5, 10*3:5, 70, 100,
140), col='blue1')
text(r$mids, r$density, r$counts, adj=c(.5, -.5),
col='blue3')
sapply(r[2:3], sum)
sum(r$density * diff(r$breaks)) # == 1
lines(r, lty = 3, border = "purple") # ->
lines.histogram(*)
par(op)

utils::str(hist(islands, br=12, plot= FALSE)) #-> 10 (~=
12) breaks
utils::str(hist(islands,
br=c(12,20,36,80,200,1000,17000), plot = FALSE))
```

```
hist(islands, br=c(12,20,36,80,200,1000,17000), freq =  
TRUE,  
      main = "WRONG histogram") # and warning
```

[Package *graphics* version 2.0.1 [Index](#)]