

# Package ‘mvhist’

May 9, 2026

**Type** Package

**Title** Multivariate Histograms

**Version** 1.2

**Date** 2025-10-15

**Author** John P. Nolan [aut, cre, cph]

**Maintainer** John P. Nolan <jpnolan@american.edu>

**Depends** R (>= 4.0), rgl, mvmesh, SimplicialCubature, rccd

**Description** Tabulate and plot directional and other multivariate histograms.

**License** GPL (>= 3)

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2025-10-15 19:20:02 UTC

## Contents

mvhist . . . . .	1
mvhistDirectional . . . . .	5
mvhistMisc . . . . .	10

<b>Index</b>	<b>11</b>
--------------	-----------

---

mvhist	<i>Multivariate Histograms</i>
--------	--------------------------------

---

## Description

Tabulate and plot histograms for multivariate data, including directional histograms. This package used to be part of the [mvmesh](#) package, which works with multivariate meshes and grids. To simplify that package and make these functions more visible, this package was extracted as a self-standing package in Septemeber 2023. The functions provided can tally the number of data points in a list of regions in dimensions two or more. All regions/bins have flat sides.

Version 1.2 (2025-10-15) makes a few changes in this documentation; no functions have changed. Several different plots are available to show 2 and 3 dimensional data; and one can deal with dimension greater than 3. Plots in 3d can be rotated and zoomed in/out with the mouse, as well as resized.

### Usage

```
histRectangular( x, breaks=10, plot.type="default", freq=TRUE, report="summary", ... )
histSimplex( x, S, plot.type="default", freq=TRUE, report="summary", ... )
```

### Arguments

x	data in an (n x d) matrix; rows are d-dimensional data vectors
freq	TRUE for a frequency histogram, FALSE for a relative frequency histogram. See note about <code>normalize.by.area</code>
breaks	specifes the subdivision of the region; see 'breaks' in <a href="#">SolidRectangle</a> in package <a href="#">mvmesh</a> .
plot.type	type of plot, see details below
S	(vps x d x nS) array of simplices in the V-representation, see <a href="#">V2Hrep</a> in package <a href="#">mvmesh</a> . The vector <code>S[,i,j]</code> gives the coordinates of the i-th vertex of the j-th simplex.
report	level of warning messages; one of "summary", "all", "none".
...	Optional arguments to plot, e.g. <code>color="red"</code> , etc.

### Details

Calculate and plot multivariate histograms. `histRectangular` plots histogram based on a rectangular grid, while `histSimplex` plots histogram based on the simplices specified in `S`,

These routines use the functions and conventions of the package [mvmesh](#). In particular, shapes can be described in two formats: vertex representation or half-space representation, respectively called the V-representation or H-representation. In all cases, the bins are simplices are converted to the H-representation and tallied by `TallyHrep`.

'plot.type' values depend on the type of plot being used. Possible values are:

- "none" - does not show a plot, just return the counts. This is useful to summarize data and in higher dimensions where no plot is useful
- "index" - shows a histogram of simplex index number versus count, does not show the geometry, but works in any dimension
- "pillars" - shows a 3D plot with pillars/columns having base the shape of the simplices and height proportional to frequency counts. When the points are 2D, this works for `histRectangular` and `histSimplex`; when the points are 3D, this only works for `histRectangular`. `DrawPillars` is used to plot the pillars.
- "counts" - shows frequency counts as a number in the center of each simplex
- "radial" - `histDirectional` only, shows radial spikes proportional to the counts
- "grayscale" - `histDirectional` only, color codes simplices proportional to the counts
- "orthogonal" - `histDirectional` only, shows radial spikes proportional to the counts
- "default" - type depends on the dimension of the data and type of histogram

**Value**

A plot is drawn (unless `plot.type="none"`). Note that the plots may be underneath/behind other windows; if you don't see a plot, search your desktop and/or the plot tab. A list is returned invisibly, with the following fields:

- `counts` - frequency count in each bin
- `nrejects` - number of x values not in any bin
- `nties` - number of points in more than one bin (if bins are set up to be non-overlapping, this should only occur on a shared edge between two simplices)
- `nx` - total number of data points in x
- `rel.freq` - `counts/nx`
- `rel.rejects` - `nrejects/nx`
- `mesh` - object of type `mvmesh`, see [mvmesh](#)
- `plot.type` - input value
- `report` - input value

While counting data points in the different bins, two issues can arise: (a) a data point is on the boundary of a bin, or (b) a data point is not in any of the specified bins. If `report="none"`, no report is given about these issues. If `report="summary"`, a count is given of the number of ties and the number of rejects. If `report="all"`, the count of number of ties and rejects is given, and the indices (rows of the data matrix) of the rejected points are given.

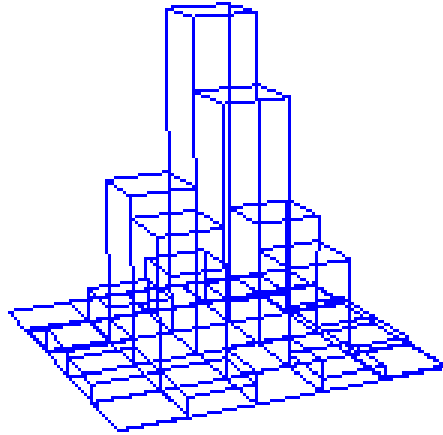
**Warning**

These functions use double precision numbers by default, and most real numbers cannot be expressed exactly as doubles. So testing for being on a boundary is subject to the usual issues with floating point numbers. This is why the message "If you want correct answers, use rational arithmetic." is given when the package `rcdd` is loaded.

It is possible, but takes some work, to specify regions using only rational numbers as coordinates, and if the data is rational, you will be able to exactly specify regions and possibly boundaries. See the help for packages [mvmesh](#) and `rcdd`. Using rational coordinates has not been tested in this package.

**Example of histRectangular**

## 2d, default plot.type

**Examples**

```

# isotropic data in 2 and 3 dimensions
x2d <- matrix( rnorm(8000), ncol=2 )
x3d <- matrix( rnorm(9000), ncol=3 )

# 3d plots are in separate windows opened by the rgl package; you may have
#   to search on the desktop to find those windows
if( interactive() ){

# save graphical parameters; restore to original value at end of examples
oldpar <- par()

# simple histogram of 2d data
histRectangular( x2d, breaks=5); title3d("2d, default plot.type" )

# simple histogram of 3d data: slices of data are stacked
histRectangular( x3d, breaks=4 ); title3d("3d, default plot.type" )

histRectangular( x2d, breaks=5, col='blue', plot.type="pillars" )
histRectangular( x2d, breaks=5, plot.type="counts" )
histRectangular( x2d, breaks=5, plot.type="index" )

# count number of data points in a triangle, using mvmesh function to define the partition

```

```

S1 <- 4*SolidSimplex( n=2, k=3 )$S
histSimplex( x2d, S1, plot.type="counts" ) # note many rejects
histSimplex( x2d, S1, col="green", lwd=3 ) # default plot.type="pillars"

# partition a ball
S2 <- 4*UnitBall( n=2, k=2 )$S
histSimplex( x2d, S2, plot.type="counts", col="purple" )
histSimplex( x2d, S2, col="red" )

# Specify simplices explicitly to get specific region, e.g. restrict to x[1] >= 0
S1 <- matrix( c(0,0, 10,0, 0,10, 10,10), ncol=2, byrow=TRUE ) # first quadrant (bounded)
S2 <- matrix( c(0,0, 10,0, 0,-10, 10,-10), ncol=2, byrow=TRUE ) # fourth quadrant (bounded)
S <- array( c(S1,S2), dim=c(4,2,2) )
simp <- histSimplex( x2d, S, plot.type="counts" )
text(2,9, paste("nrejects=", simp$nrejects), col='red' )

# check behavior with rejects and ties
r <- histSimplex( x2d, S, plot.type="counts" )
str(r) # see list of returned values
sum(c(r$counts, r$nrejects))

# restore user's graphical parameters
par(oldpar)
}

```

---

mvhistDirectional      *Directional Histograms*

---

## Description

Tabulate and plot directional histograms

## Usage

```

histDirectional( x, k, p=2, plot.type="default", freq=TRUE, positive.only=FALSE,
  report="summary", label.orthants=TRUE, normalize.by.area=FALSE, ... )
histDirectionalQuantileThreshold( x, probs=1, p=2, k=3, positive.only=FALSE, ... )
histDirectionalAbsoluteThreshold( x, thresholds=0, p=2, k=3, positive.only=FALSE, ... )

```

## Arguments

x	data in an (n x d) matrix; rows are d-dimensional data vectors
k	number of subdivisions, i.e. directional cones
p	power of p-norm
freq	TRUE for a frequency histogram, FALSE for a relative frequency histogram. See note about normalize.by.area

<code>normalize.by.area</code>	if TRUE, then the counts are normalized by the surface area of the corresponding simplex on the sphere. This is useful since in general the surface area varies with the region and counts will vary accordingly. In particular, isotropic data will not appear isotropic without setting this to TRUE. If TRUE, the value of <code>freq</code> is ignored: the histogram always shows the ratio <code>count/surface area</code>
<code>plot.type</code>	type of plot, see details below
<code>positive.only</code>	If TRUE, look only in the first orthant
<code>report</code>	level of warning messages; one of "summary", "all", "none".
<code>label.orthants</code>	If <code>plot.type="index"</code> , this controls whether or not the orthants are labeled on the plot.
<code>probs</code>	vector of probabilities specifying what fraction of the extremes to keep
<code>thresholds</code>	vector of thresholds specifying cutoff for extremes to keep
<code>...</code>	Optional arguments to plot

## Details

Calculate and plot multivariate directional histograms.

Each orthant is subdivided by a  $k$ -wise subdivision and cones are constructed from the origin through each subdivision (and continuing to infinity). The tally then counts how many data points are in each (infinite) cone.

`histDirectional` plots a directional histogram for all the data, `histDirectionalQuantileThreshold` plots  $m=\text{length}(\text{probs})$  directional histograms, with plot  $i$  using the top `probs[i]` fraction of the data, `histDirectionalAbsoluteThreshold` plots  $m=\text{length}(\text{cut.off})$  directional histograms, with plot  $i$  using the data above `cut.off[i]`.

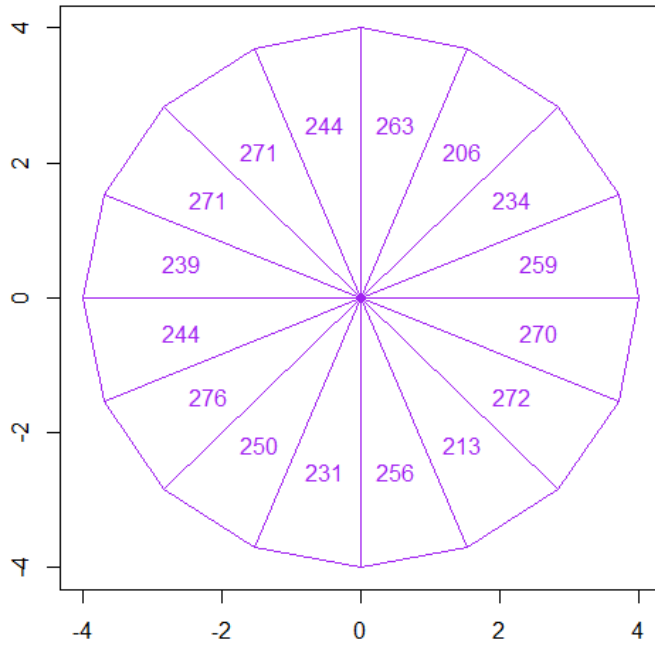
When thresholding is being done, distance from the origin is computed by using the  $p$ -norm;  $p=2$  is Euclidean distance.

'`plot.type`' values depend on the type of plot being used. See possible values in `mvhist`.

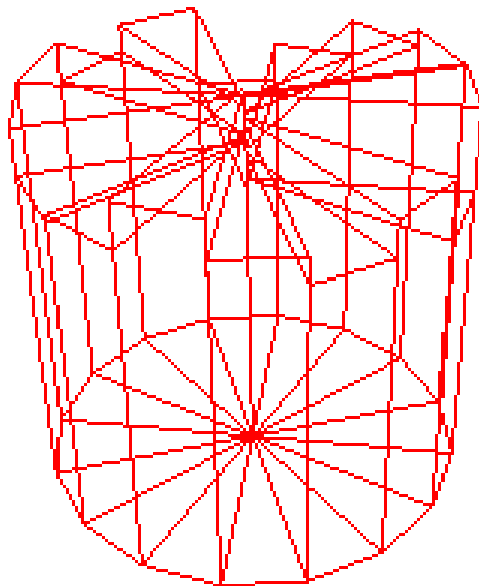
## Value

A plot is drawn (unless `plot.type="none"`) and a list is returned invisibly. See function `mvhist` for the contents of that list. When the dimension of the data is bigger than 2, a flat 2d graph is drawn, with bins grouped by orthant. This may be useful to show is data is concentrated in certain directions. Orthants are labeled with a string of pluses and minuses, e.g. `++-` means the first two coordinates are positive and the last two are negative.

**Examples of directional histograms**



histSimplex example with plot.type="counts"



histSimplex example with plot.type="pillars"

### Examples

```
# some directional, heavy tailed 2-dim data
n <- 1000
A <- matrix( c(1,2, 4,1), nrow=2,ncol=2)
x2 <- matrix( 0.0, nrow=n, ncol=2 )
for (i in 1:n) {
  x2[i,] <- c( sum( A[1,] * (1/runif(1))), sum(A[2,] * (1/runif(1))) )
}

# three dimensional positive data
x3 <- matrix( abs(rnorm(9000)), ncol=3 )

if( interactive() ){

# save graphical parameters; restore to original value at end of examples
oldpar <- par()

histDirectional( x3, k=3, positive.only=TRUE, col='blue', lwd=3 )

# show scatter plot of all daat, then directional histogram of 100%, then top 25%,
# and finally top 10%
dev.new(); par(mfrow=c(2,2))
plot(x2,main="Raw data",col='red')
```

```

histDirectionalQuantileThreshold( x2, probs=c(1,0.25,0.1), p=1,
    positive.only=TRUE, col='green',lwd=3)

dev.new(); par(mfrow=c(2,2))
histDirectionalAbsoluteThreshold( x2, thresholds=c(0,50,100,200), p=1,
    positive.only=TRUE, col='blue',lwd=3)

# two dimensional, isotropic
x <- matrix( rnorm(8000), ncol=2 )
histDirectional( x, k=2 ) # default plot.type="radial"
histDirectional( x, k=2, col='red',lwd=4 ) # tinker with color and line width
histDirectional( x, k=2, plot.type="index" )

# three dimensional positive data
x <- matrix( abs(rnorm(9000)), ncol=3 )
histDirectional( x, k=3, positive.only=TRUE, col='blue', lwd=3 )
histDirectional( x, k=2, plot.type="orthogonal", positive.only=TRUE, p=1 )

# three dimensional isotropic/radially symmetric
x <- matrix( rnorm(3000), ncol=3, nrow=1000 )

histDirectional( x, k=2 ) # default plot.type="radial"
histDirectional( x, k=2, plot.type="grayscale" )
histDirectional( x, k=2, plot.type="index" )

# compare frequency, relative freq. and normalized histograms
n <- 20000; d <- 3; k <- 2
x <- matrix( rnorm( n*d ), nrow=n, ncol=d )
dev.new(); par(mfrow=c(3,1),mar=c(4,4,2,1))
histDirectional( x, k, plot.type="index" )
title("omnidirectional data: frequency")
histDirectional( x, k, freq=FALSE, plot.type="index" )
title("omnidirectional data: relative frequency")
histDirectional( x, k, plot.type="index", normalize.by.area=TRUE )
title("omnidirectional data: frequency/surface.area")

# 3d data, first octant only
# second plot is a multiple of first; third plot normalizes by the area of
# the partition elements
dev.new(); par(mfrow=c(3,1),mar=c(4,4,2,1))
histDirectional( abs(x), k=3, positive.only=TRUE, plot.type="index" )
title("positive data: frequency")
histDirectional( abs(x), k=3, positive.only=TRUE, freq=FALSE, plot.type="index" )
title("positive data: relative frequency")
histDirectional( abs(x), k=3, positive.only=TRUE, plot.type="index", normalize.by.area=TRUE )
title("positive data: frequency/surface.area")

# 4 dimensional data
d <- 4; n <- 10000
x <- matrix( rnorm(d*n), nrow=n, ncol=d )
histDirectional( x, k=1 ) # orthants are identified by + and - signs

```

```

histDirectional( x, k=1, normalize.by.area=TRUE )
histRectangular( x, breaks=2 )

# restore user's graphical parameters
par(oldpar)
}

```

---

mvhistMisc

*Miscellaneous routines for multivariate histograms*


---

## Description

Utility functions for multivariate histograms

## Usage

```

TallyHrep( x, H, report="summary" )
DrawPillars( S, height, shift=rep(0.0,3), ... )
HrepCones( S )

```

## Arguments

x	data in an (n x d) matrix; rows are d-dimensional data vectors
S	(vps x d x nS) array of simplices in V representation, see <a href="#">V2Hrep</a>
H	array of simplices in H representation, see <a href="#">V2Hrep</a>
report	what information to return for tally; one of "summary", "all", "none"
height	height of the pillars
shift	(x,y,z) shift of the pillars
...	Optional arguments to plot, e.g. color="red", etc.

## Details

Internal functions for multivariate histograms.

TallyHrep tallies the number of data points in each simplex of the H representation.

DrawPillars plot 3d pillars/columns to show histogram height.

HrepCones computes an (m x (d+2) x ncones) array H, with H[,k] giving the H-representation of the k-th cone,

## Value

TallyHrep returns a list containing the number of data points in each simplex.

DrawPillars draws 3d pillars/columns to show histogram height.

HrepCones an (m x (d+2) x ncones) array H, with H[,k] giving the H-representation of the k-th cone.

# Index

DrawPillars (mvhistMisc), [10](#)

histDirectional (mvhistDirectional), [5](#)  
histDirectionalAbsoluteThreshold  
    (mvhistDirectional), [5](#)  
histDirectionalQuantileThreshold  
    (mvhistDirectional), [5](#)  
histRectangular (mvhist), [1](#)  
histSimplex (mvhist), [1](#)  
HrepCones (mvhistMisc), [10](#)

mvhist, [1](#)  
mvhist-package (mvhist), [1](#)  
mvhistDirectional, [5](#)  
mvhistMisc, [10](#)  
mvmesh, [1–3](#)

SolidRectangle, [2](#)

TallyHrep (mvhistMisc), [10](#)

V2Hrep, [2, 10](#)