

Package: HistogramTools (via r-universe)

May 13, 2026

Type Package

Title Utility Functions for R Histograms

Version 0.4.1

Date 2026-05-06

Maintainer Murray Stokely <murray@stokely.org>

Description Provides a number of utility functions useful for manipulating large histograms. This includes methods to trim, subset, merge buckets, merge histograms, convert to CDF, and calculate information loss due to binning. It also provides a protocol buffer representation of R's native histogram class to allow histograms over large data sets to be computed and combined in distributed analytical pipelines. Implements bin-by-bin histogram distance measures described in Rubner, Tomasi and Guibas (2000) <[doi:10.1023/A:1026543900054](https://doi.org/10.1023/A:1026543900054)>, Swain and Ballard (1991) <[doi:10.1007/BF00130487](https://doi.org/10.1007/BF00130487)>, and Puzicha, Hofmann and Buhmann (1997) <[doi:10.1109/CVPR.1997.609331](https://doi.org/10.1109/CVPR.1997.609331)>, and average shifted histograms as in Scott (2015, ISBN:9781118575536).

License Apache License 2.0

URL <https://github.com/murraystokely/histogramtools>

BugReports <https://github.com/murraystokely/histogramtools/issues>

Imports ash, Hmisc

Suggests emdist, gdata, testthat (>= 3.0.0)

Enhances RProtoBuf

Classification/ACM G.3

Config/testthat/edition 3

Copyright Copyright 2011-2015 Google, Inc. Copyright 2026 Murray Stokely

Encoding UTF-8

Config/pak/sysreqs cmake make libicu-dev libuv1-dev

Repository <https://murraystokely.r-universe.dev>
Date/Publication 2026-05-06 18:16:25 UTC
RemoteUrl <https://github.com/murraystokely/histogramtools>
RemoteRef HEAD
RemoteSha 98a1f51dbeb80d3f23d8a8fed26c8b0160dc2836

Contents

HistogramTools-package	2
AddHistograms	4
as.histogram	5
as.Message	7
ASH	8
dtracehistograms	9
HistogramDistance	10
HistToEcdf	12
InformationLoss	13
IntersectHistograms	14
MergeBuckets	15
PlotLog2ByteEcdf	16
PlotRelativeFrequency	18
PreBinnedHistogram	19
quantile	19
ScaleHistogram	20
SubsetHistogram	21
trimspare	22
Index	24

HistogramTools-package
HistogramTools package

Description

This package provides a number of utility functions for manipulating R's native histogram objects. The functions are focused on operations that are particularly useful when dealing with large numbers of histograms with identical buckets, such as those produced from distributed MapReduce computations. This package also provides a 'HistogramTools.HistogramState' protocol buffer representation of the default R histogram class to allow histograms to be very concisely serialized and shared with other systems.

Details

See `library(help=HistogramTools)` for version number, dates, dependencies, and a complete list of functions.

Index (possibly out of date):

<code>AddHistograms</code>	Aggregate histogram objects that have identical breaks.
<code>MergeBuckets</code>	Merge adjacent buckets of a histogram.
<code>ApproxQuantile</code>	Approximate the quantiles of the underlying distribution.
<code>ApproxMean</code>	Approximate the mean of the underlying distribution.
<code>Count</code>	Count of all samples in a histogram.
<code>HistToEcdf</code>	Approximate the ECDF of the underlying distribution.
<code>SubsetHistogram</code>	Subset a histogram by removing some of the buckets.
<code>TrimHistogram</code>	Remove empty buckets from the tails of a histogram.
<code>ScaleHistogram</code>	Scale histogram bucket counts by a numeric value.
<code>PreBinnedHistogram</code>	Generate a histogram from pre-binned data.
<code>AshFromHist</code>	Compute Average Shifted Histogram from a histogram.
<code>KSDCC</code>	Compute maximal KS-statistic of CDFs constructed from histogram.
<code>EMDCC</code>	Compute maximal Earth Mover's Distance of CDFs constructed from histogram.
<code>PlotKSDCC</code>	Plot the KSDCC metric and a CDF from the histogram.
<code>PlotEMDCC</code>	Plot the EMDCC metric and a CDF from the histogram.
<code>PlotLog2ByteEcdf</code>	Plot the CDF from a histogram with log2 scaled byte boundaries.
<code>PlotLogTimeDurationEcdf</code>	Plot the CDF from a histogram with log scaled time duration boundaries.
<code>PlotRelativeFrequency</code>	Plot a relative frequency histogram.
<code>ReadHistogramsFromDtraceOutputFile</code>	Read a list of Histograms from the output of the DTrace tool.
<code>minkowski.dist</code>	Compute the Minkowski difference between two histograms.
<code>intersect.dist</code>	Compute the histogram intersection distance between two histograms.
<code>kl.divergence</code>	Compute the Kullback-Leibler divergence between two histograms.
<code>jeffrey.divergence</code>	Compute the Jeffrey divergence between two histograms.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[hist](#)

Examples

```
if (requireNamespace("RProtoBuf", quietly = TRUE)) {
  tmp.hist <- hist(c(1,2,4,43,20,33,1,1,3), plot=FALSE)
  # The default R serialization takes a fair number of bytes
  length(serialize(tmp.hist, NULL))

  # Convert to a protocol buffer representation.
  hist.msg <- as.Message(tmp.hist)

  # Which has an ASCII representation like this:
  cat(as.character(hist.msg))
}
```

```

# Or can be serialized and shared with other tools much more
# succinctly than R's built-in serialization format.
length(hist.msg$serialize(NULL))

# And since this isn't even compressed, we can reduce it further
# with in-memory compression:
length(memCompress(hist.msg$serialize(NULL)))

# If we read in the raw.bytes from another tool
raw.bytes <- hist.msg$serialize(NULL)

# We can parse the raw bytes as a protocol buffer
new.hist.proto <- RProtoBuf::P("HistogramTools.HistogramState")$read(raw.bytes)
new.hist.proto

# Then convert back to a native R histogram.
new.hist <- as.histogram(new.hist.proto)

# The new histogram and the old are identical except for xname
}

```

AddHistograms

Aggregate histograms that have identical breaks.

Description

Aggregate histogram objects that have identical breaks.

Usage

```
AddHistograms(..., x=list(...), main=.NewHistogramName(x))
.NewHistogramName(x)
```

Arguments

...	Histogram objects (created by hist).
x	A list of histogram objects. If x is supplied then the ... are ignored.
main	A title to add to the aggregated/merged histogram. By default, if two histograms are provided a title will be created that includes the names of the original histograms. If more histograms are provided the title will simply include the number of aggregated histograms.

Details

This function adds the buckets of the provided histograms to return a single aggregated histogram. `.NewHistogramName` is a utility that takes the list of histogram objects to be aggregated and returns a name for the new merged histogram. It is normally hidden, but can be viewed using `HistogramTools:::NewHistogramName`.

Value

AddHistograms returns an object of class "histogram" whose breaks match those of the inputs and whose counts are the elementwise sum of the counts of the supplied histograms. .NewHistogramName returns a single character string suitable for use as the xname of the merged histogram.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#) and [hist](#).

Examples

```
hist.1 <- hist(c(1,2,3,4), plot=FALSE)
hist.2 <- hist(c(1,2,2,4), plot=FALSE)
hist.sum <- AddHistograms(hist.1, hist.2)
hist.3 <- hist(c(1,2,2,4), plot=FALSE)
hist.sum <- AddHistograms(hist.1, hist.2, hist.3)
```

as.histogram

Convert histogram protocol buffers to histogram objects

Description

This package provides a number of utility functions useful for manipulating large histograms. It provides a 'HistogramTools.HistogramState' protocol buffer representation of the default R histogram class to allow histograms to be very concisely serialized and shared with other systems in a distributed MapReduce environment. It also includes a number of utility functions for manipulating large histograms.

Usage

```
## S3 method for class 'Message'
as.histogram(x, ...)
```

Arguments

x	An RProtoBuf Message of type "HistogramTools.HistogramState" to convert to a histogram object.
...	Not used.

Details

as.histogram reads the provided Protocol Buffer message and extracts the buckets and counts to populate into the standard R histogram class which can be plotted.

Value

An object of class "histogram" (a list with components breaks, counts, density, mids, xname, and equidist) reconstructed from the supplied HistogramTools.HistogramState protocol buffer, suitable for use with `plot.histogram` and the other functions in this package.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [as.Message](#), and [RProtoBuf](#).

Examples

```
if (requireNamespace("RProtoBuf", quietly = TRUE)) {
  tmp.hist <- hist(c(1,2,4,43,20,33,1,1,3), plot=FALSE)
  # The default R serialization takes a fair number of bytes
  length(serialize(tmp.hist, NULL))

  # Convert to a protocol buffer representation.
  hist.msg <- as.Message(tmp.hist)

  # Which has an ASCII representation like this:
  cat(as.character(hist.msg))

  # Or can be serialized and shared with other tools much more
  # succinctly than R's built-in serialization format.
  length(hist.msg$serialize(NULL))

  # And since this isn't even compressed, we can reduce it further
  # with in-memory compression:
  length(memCompress(hist.msg$serialize(NULL)))

  # If we read in the raw.bytes from another tool
  raw.bytes <- hist.msg$serialize(NULL)

  # We can parse the raw bytes as a protocol buffer
  new.hist.proto <- RProtoBuf::P("HistogramTools.HistogramState")$read(raw.bytes)
  new.hist.proto

  # Then convert back to a native R histogram.
  new.hist <- as.histogram(new.hist.proto)

  # The new histogram and the old are identical except for xname
}
```

`as.Message`*Convert R histograms to Protocol Buffer representation*

Description

This package provides a number of utility functions useful for manipulating large histograms. It provides a 'HistogramTools.HistogramState' protocol buffer representation of the default R histogram class to allow histograms to be very concisely serialized and shared with other systems in a distributed MapReduce environment. It also includes a number of utility functions for manipulating large histograms.

Usage

```
## S3 method for class 'histogram'
as.Message(x)
```

Arguments

x A histogram object (created by [hist](#)).

Details

`as.Message` converts the provided histogram object into a protocol buffer representation that can be compactly serialized and shared with tools written in other languages.

Value

An RProtoBuf "Message" object of type `HistogramTools.HistogramState` containing the histogram's breaks, counts, and name fields. The returned message can be serialized to a compact binary representation with its `$serialize()` method for sharing with tools in other languages.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [as.histogram](#), and [RProtoBuf](#).

Examples

```
if (requireNamespace("RProtoBuf", quietly = TRUE)) {
  tmp.hist <- hist(c(1,2,4,43,20,33,1,1,3), plot=FALSE)
  # The default R serialization takes a fair number of bytes
  length(serialize(tmp.hist, NULL))

  # Convert to a protocol buffer representation.
  hist.msg <- as.Message(tmp.hist)
```

```

# Which has an ASCII representation like this:
cat(as.character(hist.msg))

# Or can be serialized and shared with other tools much more
# succinctly than R's built-in serialization format.
length(hist.msg$serialize(NULL))

# And since this isn't even compressed, we can reduce it further
# with in-memory compression:
length(memCompress(hist.msg$serialize(NULL)))

# If we read in the raw.bytes from another tool
raw.bytes <- hist.msg$serialize(NULL)

# We can parse the raw bytes as a protocol buffer
new.hist.proto <- RProtoBuf::P("HistogramTools.HistogramState")$read(raw.bytes)
new.hist.proto

# Then convert back to a native R histogram.
new.hist <- as.histogram(new.hist.proto)

# The new histogram and the old are identical except for xname
}

```

 ASH

Average Shifted Histograms From a Histogram.

Description

Computes a univariate average shifted histogram (polynomial kernel) given a single input histogram.

Usage

```
HistToASH(h, m=5, kopt=c(2,2))
```

Arguments

<code>h</code>	A histogram object (created by <code>hist</code>) representing a pre-binned dataset.
<code>m</code>	optional integer smoothing parameter, passed to <code>ash1()</code> .
<code>kopt</code>	vector of length 2 specifying the kernel, passed to <code>ash1()</code> .

Details

This function takes a histogram and uses the counts as the input to the `ash1()` function in the `ash` package to compute the average shifted histogram.

Value

A list of class "ash" as returned by `ash1()`, containing components `x` and `y` giving the bin centers and estimated density values for the average shifted histogram, suitable for use with `plot` or `lines`.

Author(s)

Murray Stokely <murray@stokely.org>

References

Scott, David W. *Multivariate density estimation: theory, practice, and visualization*. Vol. 383. Wiley. com, 2009.

See Also

[HistogramTools-package](#), [ash1](#), and [hist](#).

Examples

```
x <- runif(1000, min=0, max=100)
h <- hist(x, breaks=0:100, plot=FALSE)
plot(h, freq=FALSE)

# Superimpose the Average Shifted Histogram on top of the original.
lines(HistToASH(h), col="red")
```

dtracehistograms *Read Histograms from text DTrace output file.*

Description

Parses the text output of the DTrace command to convert the ASCII representation of aggregate distributions into R histogram objects.

Usage

```
ReadHistogramsFromDtraceOutputFile(filename)
```

Arguments

`filename` A character vector naming a file that is the output of dtrace with aggregate distribution statistics in it.

Details

The DTrace dynamic tracing framework allows users to trace applications and computer operating systems. One of its common outputs is aggregates of a distribution (for example, read request sizes) that are output as a histogram. This function takes the text output from a dtrace command and looks for text distribution representations that can be parsed into R histogram objects.

Value

A list of histogram objects representing the histograms present in the Dtrace output file.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [hist](#).

Examples

```
# A sample DTrace output file is bundled with the package. In a real
# DTrace session the file would be produced by something like:
#   dtrace -n 'syscall::read:return { @[execname] = quantize(arg0); }' \
#     > /tmp/dtraceoutput
filename <- system.file("extdata/buildkernel-readsize-dtrace.txt",
                        package="HistogramTools")
read.size.hists <- ReadHistogramsFromDtraceOutputFile(filename)
plot(read.size.hists[["TOTAL"]])
```

HistogramDistance *Histogram Distance Measures*

Description

The pairs of bins in two histograms with the same bucket boundaries are compared to compute dissimilarity measures.

Usage

```
minkowski.dist(h1, h2, p)
intersect.dist(h1, h2)
kl.divergence(h1, h2)
jeffrey.divergence(h1, h2)
```

Arguments

h1, h2 "histogram" objects (created by [hist](#)) representing a binned dataset.
p Order of the Minkowski distance between two histograms to compute.

Details

The `minkowski.dist` function computes the Minkowski distance of order p between two histograms. $p=1$ is the Manhattan distance and $p=2$ is the Euclidean distance.

The `intersect.dist` function computes the intersection distance of two histograms, as defined in Swain and Ballard 1991, p15. If histograms `h1` and `h2` do not contain the same total of counts, then this metric will not be symmetric.

The `kl.divergence` function computes the Kullback-Leibler divergence between two histograms.

The `jeffrey.divergence` function computes the Jeffrey divergence between two histograms.

Value

Each function returns a single non-negative numeric value giving the computed distance or divergence between the two histograms. Larger values correspond to histograms that are more dissimilar.

Author(s)

Murray Stokely <murray@stokely.org>

References

Rubner, Yossi, Carlo Tomasi, and Leonidas J. Guibas. "The earth mover's distance as a metric for image retrieval." *International Journal of Computer Vision* 40.2 (2000): 99-121.

Puzicha, Jan, Thomas Hofmann, and Joachim M. Buhmann. "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval." *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on. IEEE, 1997.*

Swain, Michael J., and Dana H. Ballard. "Color indexing." *International journal of computer vision* 7.1 (1991): 11-32.

See Also

[HistogramTools-package](#), [ecdf](#), and [hist](#).

Examples

```
h1 <- hist(runif(100), plot=FALSE)
h2 <- hist(runif(100), plot=FALSE)

minkowski.dist(h1, h2, 1)
minkowski.dist(h1, h2, 2)
minkowski.dist(h1, h2, 3)
intersect.dist(h1, h2)
kl.divergence(h1, h2)
jeffrey.divergence(h1, h2)
```

`HistToEcdf`*Empirical Cumulative Distribution Function From a Histogram.*

Description

Computes an approximate empirical cumulative distribution function of a data set given a binned histogram representation of that dataset.

Usage

```
HistToEcdf(h, method="constant", f=0, inverse=FALSE)
```

Arguments

<code>h</code>	A histogram object (created by hist) representing a pre-binned dataset on which we'd like to calculate an ECDF.
<code>method</code>	specifies the interpolation method to be used in call to approxfun (<code>.</code>). Choices are "linear" or "constant".
<code>f</code>	if <code>method="constant"</code> a number between 0 and 1 inclusive, indicating a compromise between left- and right-continuous step functions. See ?approxfun
<code>inverse</code>	if TRUE return the inverse Ecdf.

Details

This function approximates the e.c.d.f. (empirical cumulative distribution function) of a data set given a binned histogram representation of that data set.

Value

A function of class `c("ecdf", "stepfun", "function")`, as returned by [ecdf](#). Calling it with a numeric vector `x` returns the approximate cumulative probability at each value of `x`. If `inverse=TRUE` the returned function instead maps a probability in $[0, 1]$ to the corresponding approximate quantile.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [ecdf](#), [approxfun](#), and [hist](#).

Examples

```
h <- hist(runif(100), plot=FALSE)
plot(HistToEcdf(h))
```

Description

Computes a metric between 0 and 1 of the amount of information lost about the underlying distribution of data for a given histogram.

Usage

```
KSDCC(h)
EMDCC(h)
PlotKSDCC(h, arrow.size.scale=1, main=paste("KSDCC =", KSDCC(h)), ...)
PlotEMDCC(h, main=paste("EMDCC =", EMDCC(h)), ...)
```

Arguments

<code>h</code>	A "histogram" object (created by hist) representing a pre-binned dataset on which we'd like to calculate the information loss due to binning.
<code>arrow.size.scale</code>	specifies a size scaling factor for the arrow illustrating the point of Kolmogorov-Smirnov distance between the two e.c.d.fs
<code>main</code>	if 'method="constant"' a number between 0 and 1 inclusive, indicating a compromise between left- and right-continuous step functions. See <code>?approxfun</code>
<code>...</code>	Any other arguments to pass to plot

Details

The `KSDCC` (Kolmogorov-Smirnov Distance of the Cumulative Curves) function provides the Kolmogorov-Smirnov distance between the empirical distribution functions of the smallest and largest datasets that could be represented by the binned data in the provided histogram. This quantity is also called the Maximum Displacement of the Cumulative Curves (MDCC) in the computer science performance evaluation community (see references).

The `EMDCC` (Earth Mover's Distance of the Cumulative Curves) function is like the Kolmogorov-Smirnov statistic, but uses an integral to capture the difference across all points of the curve rather than just the maximum difference. This is also known as Mallows distance, or Wasserstein distance with $p=1$.

The `PlotKSDCC` and `PlotEMDCC` functions take a histogram and generate a plot showing a geometric representation of the information loss metrics for the provided histogram.

Value

`KSDCC` and `EMDCC` each return a single numeric value between 0 and 1 quantifying the information loss for the supplied histogram. `PlotKSDCC` and `PlotEMDCC` have no return value and are called for the side effect of drawing an annotated ECDF plot on the current graphics device.

Author(s)

Murray Stokely <murray@stokely.org>

References

Douceur, John R., and William J. Bolosky. "A large-scale study of file-system contents." *ACM SIGMETRICS Performance Evaluation Review* **27.1** (1999): 59-70.

See Also

[HistogramTools-package](#), [ecdf](#), and [hist](#).

Examples

```
x <- rexp(1000)
h <- hist(x, breaks=c(0,1,2,3,4,8,16,32), plot=FALSE)
KSDCC(h)

# For small enough data sets we can construct the two extreme data sets
# that can be constructed from a histogram. One assuming every data point
# is on the left boundary of its bucket, and another assuming every data
# point is on the right boundary of its bucket. Our KSDCC metric for
# histograms is equivalent to the ks.test statistics for these two
# extreme data sets.

x.min <- rep(head(h$breaks, -1), h$counts)
x.max <- rep(tail(h$breaks, -1), h$counts)
ks.test(x.min, x.max, exact=FALSE)

PlotKSDCC(h)

EMDCC(h)
PlotEMDCC(h)
```

IntersectHistograms *Intersect Histograms*

Description

Takes two histograms with identical bucket boundaries and returns a histogram of the intersection of the two. Each bucket of the returned histogram contains the minimum of the equivalent buckets in the two provided histograms.

Usage

```
IntersectHistograms(h1, h2)
```

Arguments

h1, h2 "histogram" objects (created by [hist](#)) representing a dataset summarized by binning.

Value

An object of class "histogram" whose breaks match those of the input histograms and whose counts are the elementwise minimum of the counts in h1 and h2.

Author(s)

Murray Stokely <murray@stokely.org>

References

Swain, Michael J., and Dana H. Ballard. "Color indexing." International journal of computer vision 7.1 (1991): 11-32.

See Also

[HistogramTools-package](#), [hist](#).

Examples

```
h1 <- hist(runif(100), plot=FALSE)
h2 <- hist(runif(100), plot=FALSE)
plot(IntersectHistograms(h1, h2))
```

MergeBuckets

Merge adjacent buckets in a histogram to create a new histogram.

Description

Merge adjacent buckets in a histogram and return a new histogram.

Usage

```
MergeBuckets(x, adj.buckets=NULL, breaks=NULL, FUN=sum)
```

Arguments

x A histogram object (created by [hist](#)).

adj.buckets The number of adjacent buckets to merge together.

breaks If adj.buckets is equal to NULL, then this argument either specifies a vector giving the breakpoints between cells, or a single number giving the total number of cells in the new histogram. The vector of new buckets must have the same range as the original break list. If a single number is provided it must be less than length(x\$breaks).

FUN The user defined function that should be run to merge the counts of adjacent buckets in the histogram.

Details

Many data analysis pipelines write out histogram protocol buffers with thousands of buckets so as to be applicable in a wide range of contexts. This function provides a way to transform the histogram into one with fewer buckets.

Value

An object of class "histogram" with bucket boundaries derived from `adj.buckets` or `breaks` and counts produced by applying `FUN` to the counts of the corresponding adjacent buckets in `x`. The returned object covers the same range as `x` and is suitable for plotting with `plot.histogram`.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#) and [hist](#).

Examples

```
hist.1 <- hist(c(1,2,3), breaks=c(0,1,2,3,4,5,6,7,8,9), plot=FALSE)
hist.2 <- MergeBuckets(hist.1, adj.buckets=2)
```

```
hist.1
hist.2
```

PlotLog2ByteEcdf *Plot Binned Histogram and ECDF Data.*

Description

Produces aesthetically pleasing ECDF plots for two common classes of non-equiwidth histograms. Specifically, (1) histograms with power of two bucket boundaries commonly used in computer science for measuring resource usage, and (2) histograms with log-scaled time duration buckets with a range from 1 second to 10 years.

Usage

```
PlotLog2ByteEcdf(x, xlab="Bytes (log)",
                 ylab="Cumulative Fraction", with.grid=TRUE, ...)
PlotLogTimeDurationEcdf(x, with.grid=TRUE,
                        xlab="Age (log)",
                        ylab="Cumulative Fraction",
                        cex.lab=1.6,
                        cex.axis=1.6, ...)
```

Arguments

<code>x</code>	A "histogram" object (created by hist) representing a pre-binned dataset or an "ecdf" object (created by ecdf or HistToEcdf).
<code>xlab</code>	x-axis label for the Ecdf plot.
<code>ylab</code>	y-axis label for the Ecdf plot.
<code>with.grid</code>	Logical. If TRUE, draw faint grid lines on the ECDF plot.
<code>cex.lab</code>	Graphical parameters for plot and axes.
<code>cex.axis</code>	Graphical parameters for plot and axes.
<code>...</code>	Additional parameters are passed to plot() .

Details

The `PlotLog2ByteEcdf` function takes a "histogram" or "ecdf" which has power of 2 bucket boundaries representing bytes and creates an Ecdf plot.

The `PlotLogTimeDurationEcdf` function takes a "histogram" or "ecdf" with exponential bucket boundaries representing seconds of age or duration and creates an Ecdf plot.

Value

No return value; both functions are called for their side effect of drawing an ECDF plot on the current graphics device.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [hist](#), [ecdf](#).

Examples

```
filename <- system.file("extdata/buildkernel-readsize-dtrace.txt",
                        package="HistogramTools")
dtrace.hists <- ReadHistogramsFromDtraceOutputFile(filename)
x <- SubsetHistogram(dtrace.hists[["TOTAL"]], minbreak=1)
PlotLog2ByteEcdf(x, cex.lab=1.4)

x <- rexp(100000)
x <- x*(86400*300)/diff(range(x))

n <- as.integer(1+log2(max(x)))

h <- hist(x, breaks=c(0, unique(as.integer(2^seq(from=0, to=n, by=.25)))))
PlotLogTimeDurationEcdf(h)
```

PlotRelativeFrequency *Plot Relative Frequency Histogram*

Description

Produces a relative frequency histogram.

Usage

```
PlotRelativeFrequency(x, ylab="Relative Frequency", ...)
```

Arguments

<code>x</code>	A "histogram" object (created by hist) representing a pre-binned dataset.
<code>ylab</code>	y-axis label for the plot.
<code>...</code>	Additional parameters are passed to plot() .

Details

The default [plot.histogram](#) function supports Frequency or Density plots, but does not provide a way to produce a relative frequency histogram. This function plots this type of histogram.

Value

No return value; called for the side effect of drawing a relative frequency histogram on the current graphics device.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [hist](#), [plot.histogram](#).

Examples

```
x <- runif(100)
h <- hist(x, plot=FALSE)
PlotRelativeFrequency(h)
```

PreBinnedHistogram *PreBinnedHistogram*

Description

Takes a set of already binned data represented by a vector of $n+1$ breaks and a vector of n counts and returns a normal R histogram object.

Usage

```
PreBinnedHistogram(breaks, counts, xname="")
```

Arguments

breaks	A numeric vector of $n+1$ breakpoints for the histogram.
counts	A numeric vector of n counts for each bucket of the histogram.
xname	A character string with the name for this histogram.

Value

An object of class "histogram" (a list with components breaks, counts, density, mids, xname, and equidist) constructed from the supplied breaks and counts, suitable for plotting with [plot.histogram](#) or for use with the other functions in this package.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [hist](#).

quantile *Histogram Approximate Quantiles.*

Description

Approximate the quantiles of the underlying distribution for which only a histogram is available.

Usage

```
ApproxQuantile(x, probs, ...)  
ApproxMean(x)  
Count(x)
```

Arguments

x	A histogram object (created by hist).
probs	Numeric vector of probabilities with values in [0,1].
...	Any other arguments to pass to wtd.quantile .

Details

Many data analysis pipelines write out histogram protocol buffers with thousands of buckets so as to be applicable in a wide range of contexts. This function provides a way to transform the histogram into approximations of the quantile, median, mean, etc of the underlying distribution. `Count(x)` returns the number of observations in the histogram.

Value

`Count` returns a single numeric value giving the total number of observations summarized by the histogram (the sum of `x$counts`). `ApproxMean` returns a single numeric value giving an approximation of the mean of the underlying distribution computed as a weighted mean of the bucket midpoints. `ApproxQuantile` returns a numeric vector the same length as `probs` containing approximate quantiles of the underlying distribution; if `probs` is named the result is named accordingly.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#) and [hist](#).

Examples

```
x <- hist(c(1,2,3), breaks=c(0,1,2,3,4,5,6,7,8,9), plot=FALSE)
Count(x)
ApproxMean(x)
ApproxQuantile(x, .5)
ApproxQuantile(x, c(.05, .95))
```

ScaleHistogram

Scale Histogram Counts

Description

Scales the counts of the provided histograms by the provided factor and returns a new histogram.

Usage

```
ScaleHistogram(x, factor)
```

Arguments

x	A "histogram" object (created by hist) representing a dataset summarized by binning.
factor	A number value to scale the bucket counts by. Defaults to 1/Count(x) to normalize the sum of counts of the histogram to 1.

Value

An object of class "histogram" with the same breaks as x and counts multiplied by factor; the density and mids components are recomputed to remain consistent.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#), [hist](#).

Examples

```
x <- runif(100)
h <- hist(x, plot=FALSE)
plot(ScaleHistogram(h, 10))
```

SubsetHistogram

Subset a histogram by removing some of the buckets.

Description

SubsetHistogram creates a histogram by zooming in on a portion of a larger histogram and discarding tail buckets.

Usage

```
SubsetHistogram(x, minbreak=NULL, maxbreak=NULL)
```

Arguments

x	A histogram object (created by hist).
minbreak	If non-NULL, specifies a new minimum breakpoint for the returned histogram. Must be one of the existing breakpoints of x.
maxbreak	If non-NULL, specifies a new maximum breakpoint for the returned histogram. Must be one of the existing breakpoints of x.

Details

This function provides a way to "zoom-in" on a histogram by setting new minimum and maximum breakpoints and returning a histogram of only the interior part of the distribution. At least one of `minbreak` or `maxbreak` should be set, otherwise the original histogram is returned unmodified.

Value

An object of class "histogram" containing only the buckets of `x` whose boundaries lie within `[minbreak, maxbreak]`; the remaining buckets, breaks, and other components are recomputed to be internally consistent.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#) and [hist](#).

Examples

```
hist.1 <- hist(c(1,2,3), breaks=c(0,1,2,3,4,5,6,7,8,9), plot=FALSE)
hist.2 <- SubsetHistogram(hist.1, maxbreak=6)
```

```
hist.1
hist.2
```

trimsparse

Trim the tails of a sparse histogram.

Description

Removes empty consecutive buckets at the tails of a histogram.

Usage

```
TrimHistogram(x, left=TRUE, right=TRUE)
```

Arguments

<code>x</code>	A histogram object (created by hist).
<code>left</code>	If TRUE, consecutive buckets in the left tail of the histogram without any elements will be removed from the returned histogram.
<code>right</code>	If TRUE, consecutive buckets in the right tail of the histogram without any elements will be removed from the returned histogram.

Details

Many data analysis pipelines write out histogram protocol buffers with thousands of buckets so as to be applicable in a wide range of contexts. This function provides a way to transform the histogram into one with fewer buckets by removing sparseness in the tails.

Value

An object of class "histogram" identical to `x` except with consecutive zero-count buckets removed from the requested tails. If all buckets of `x` are empty the histogram is returned unchanged with a warning.

Author(s)

Murray Stokely <murray@stokely.org>

See Also

[HistogramTools-package](#) and [hist](#).

Examples

```
hist.1 <- hist(c(1,2,3), breaks=c(0,1,2,3,4,5,6,7,8,9), plot=FALSE)
length(hist.1$counts)
sum(hist.1$counts)
hist.trimmed <- TrimHistogram(hist.1)
length(hist.trimmed$counts)
sum(hist.trimmed$counts)
```

Index

- * **datagen**
 - dtracehistograms, 9
- * **manip**
 - AddHistograms, 4
 - as.histogram, 5
 - as.Message, 7
 - dtracehistograms, 9
 - HistToEcdf, 12
 - MergeBuckets, 15
 - quantile, 19
 - SubsetHistogram, 21
 - trimsparse, 22
- * **methods**
 - MergeBuckets, 15
 - quantile, 19
 - SubsetHistogram, 21
 - trimsparse, 22
- * **nonparametric**
 - ASH, 8
- * **package**
 - HistogramTools-package, 2
- * **utilities**
 - AddHistograms, 4
 - as.histogram, 5
 - as.Message, 7
 - dtracehistograms, 9
 - HistToEcdf, 12
 - MergeBuckets, 15
 - quantile, 19
 - SubsetHistogram, 21
 - trimsparse, 22
- .NewHistogramName (AddHistograms), 4
- AddHistograms, 4
- approxfun, 12
- ApproxMean (quantile), 19
- ApproxQuantile (quantile), 19
- as.histogram, 5, 7
- as.Message, 6, 7
- ASH, 8
- ash1, 8, 9
- Count (quantile), 19
- dtrace (dtracehistograms), 9
- dtracehistograms, 9
- ecdf, 11, 12, 14, 17
- EMDCC (InformationLoss), 13
- hist, 3–5, 7–23
- histogram (HistogramTools-package), 2
- HistogramDistance, 10
- HistogramTools
 - (HistogramTools-package), 2
- HistogramTools-package, 2
- HistToASH (ASH), 8
- HistToEcdf, 12, 17
- InformationLoss, 13
- intersect.dist (HistogramDistance), 10
- IntersectHistograms, 14
- jeffrey.divergence (HistogramDistance), 10
- kl.divergence (HistogramDistance), 10
- KSDCC (InformationLoss), 13
- lines, 9
- MergeBuckets, 15
- minkowski.dist (HistogramDistance), 10
- plot, 9, 13, 17, 18
- plot.histogram, 6, 16, 18, 19
- PlotEMDCC (InformationLoss), 13
- PlotKSDCC (InformationLoss), 13
- PlotLog2ByteEcdf, 16
- PlotLogTimeDurationEcdf
 - (PlotLog2ByteEcdf), 16

PlotRelativeFrequency, [18](#)

PreBinnedHistogram, [19](#)

quantile, [19](#)

ReadHistogramsFromDtraceOutputFile
(dtracehistograms), [9](#)

RProtoBuf, [6, 7](#)

ScaleHistogram, [20](#)

SubsetHistogram, [21](#)

TrimHistogram(trimsparse), [22](#)

trimhistogram(trimsparse), [22](#)

trimsparse, [22](#)

wtd.quantile, [20](#)