## Package ‘modeest’

October 30, 2018

<table>
<thead>
<tr>
<th>Type</th>
<th>Package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Title</td>
<td>Mode Estimation</td>
</tr>
<tr>
<td>Version</td>
<td>2.3.2</td>
</tr>
<tr>
<td>Date</td>
<td>2018-10-28</td>
</tr>
<tr>
<td>Description</td>
<td>Provides estimators of the mode of univariate unimodal data or univariate unimodal distributions.</td>
</tr>
<tr>
<td>License</td>
<td>GPL-3</td>
</tr>
<tr>
<td>LazyData</td>
<td>TRUE</td>
</tr>
<tr>
<td>Depends</td>
<td>R (&gt;= 3.1.3)</td>
</tr>
<tr>
<td>Imports</td>
<td>bazar, fBasics, genefilter, stable, stabledist, stats, statip</td>
</tr>
<tr>
<td>biocViews</td>
<td></td>
</tr>
<tr>
<td>Suggests</td>
<td>evd, knitr, mvtnorm, testthat, VGAM</td>
</tr>
<tr>
<td>URL</td>
<td><a href="https://github.com/paulponcet/modeest">https://github.com/paulponcet/modeest</a></td>
</tr>
<tr>
<td>BugReports</td>
<td><a href="https://github.com/paulponcet/modeest/issues">https://github.com/paulponcet/modeest/issues</a></td>
</tr>
<tr>
<td>RoxygenNote</td>
<td>6.1.0</td>
</tr>
<tr>
<td>NeedsCompilation</td>
<td>no</td>
</tr>
<tr>
<td>Author</td>
<td>Paul Poncet [aut, cre]</td>
</tr>
<tr>
<td>Maintainer</td>
<td>Paul Poncet <a href="mailto:paulponcet@yahoo.fr">paulponcet@yahoo.fr</a></td>
</tr>
<tr>
<td>Repository</td>
<td>CRAN</td>
</tr>
<tr>
<td>Date/Publication</td>
<td>2018-10-30 19:00:05 UTC</td>
</tr>
</tbody>
</table>

### R topics documented:

- asselin ........................................... 2
- distrMode ........................................ 3
- grenander ....................................... 8
- hrm .............................................. 10
- hsm .............................................. 11
- lientz .......................................... 13
**asselin**

The Asselin de Beauville mode estimator

**Description**
This mode estimator is based on the algorithm described in Asselin de Beauville (1978).

**Usage**

```r
asselin(x, bw = NULL, ...)
```

**Arguments**

- `x` numeric. Vector of observations.
- `bw` numeric. A number in $(0, 1]$. If $bw = 1$, the selected 'modal chain' may be too long.
- `...` further arguments to be passed to the `quantile` function.

**Value**
A numeric value is returned, the mode estimate.

**Note**
The user should preferentially call `asselin` through `mlv(x, method = "asselin", ...)`.  

**References**


**See Also**

`mlv` for general mode estimation.
distrMode

Examples

x <- rbeta(1000, shape1 = 2, shape2 = 5)

## True mode:
betaMode(shape1 = 2, shape2 = 5)

## Estimation:
asselin(x, bw = 1)
asselin(x, bw = 1/2)
mlv(x, method = "asselin")

distrMode

Mode of some continuous and discrete distributions

Description

These functions return the mode of the main probability distributions implemented in R.

Usage

distrMode(x, ...)

betaMode(shape1, shape2, ncp = 0)

cauhcyMode(location = 0, ...)

chisqMode(df, ncp = 0)

dagumMode(scale = 1, shape1.a, shape2.p)

expMode(...)

fMode(df1, df2)

fiskMode(scale = 1, shape1.a)

frechetMode(location = 0, scale = 1, shape = 1, ...)

gammaMode(shape, rate = 1, scale = 1/rate)
	normMode(mean = 0, ...)

genvMode(location = 0, scale = 1, shape = 0, ...)

gHMode(alpha = 1, beta = 0, delta = 1, mu = 0, lambda = -1/2)
distrMode

ghtMode(beta = 0.1, delta = 1, mu = 0, nu = 10)
gldMode(lambda1 = 0, lambda2 = -1, lambda3 = -1/8, lambda4 = -1/8)
gompertzMode(scale = 1, shape)
gpdMode(location = 0, scale = 1, shape = 0)
gumbelMode(location = 0, ...)
hypMode(alpha = 1, beta = 0, delta = 1, mu = 0, pm = c(1, 2, 3, 4))
koenkerMode(location = 0, ...)
kumarMode(shape1, shape2)
laplaceMode(location = 0, ...)
logisMode(location = 0, ...)
lnormMode(meanlog = 0, sdlog = 1)
lomaxMode(...)
maxwellMode(rate)
mvnormMode(mean, ...)
nakaMode(scale = 1, shape)
nigMode(alpha = 1, beta = 0, delta = 1, mu = 0)
paralogisticMode(scale = 1, shape1, a)
paretoMode(scale = 1, ...)
rayleighMode(scale = 1)
stableMode(alpha, beta, gamma = 1, delta = 0, pm = 0, ...)
stableMode2(loc, disp, skew, tail)
tMode(df, ncp)
unifMode(min = 0, max = 1)
weibullMode(shape, scale = 1)
distrMode

yulesMode(...)  
bernMode(prob)  
binomMode(size, prob)  
geomMode(...)  
hyperMode(m, n, k, ...)  
nbinomMode(size, prob, mu)  
poisMode(lambda)

Arguments

x character. The name of the distribution to consider.

... Additional parameters.

shape1 non-negative parameters of the Beta distribution.
shape2 non-negative parameters of the Beta distribution.
cp non-centrality parameter.
location location and scale parameters.
df degrees of freedom (non-negative, but can be non-integer).
scale location and scale parameters.
shape1.a shape parameters.
shape2.p shape parameters.
df1 degrees of freedom. Inf is allowed.
df2 degrees of freedom. Inf is allowed.
shape the location parameter \(a\), scale parameter \(b\), and shape parameter \(s\).
rate vector of rates.
mean vector of means.
alpha shape parameter \(\alpha\); skewness parameter \(\beta\), abs(\(\beta\)) is in the range (0, \(\alpha\)); scale parameter \(\delta\). \(\delta\) must be zero or positive; location parameter \(\mu\), by default 0. These is the meaning of the parameters in the first parameterization \(\text{pm}=1\) which is the default parameterization selection. In the second parameterization, \(\text{pm}=2\) \(\alpha\) and \(\beta\) take the meaning of the scale parameters (usually named) \(\zeta\) and \(\rho\). In the third parameterization, \(\text{pm}=3\) \(\alpha\) and \(\beta\) take the meaning of the shape parameters (usually named) \(\xi\) and \(\chi\). In the fourth parameterization, \(\text{pm}=4\) \(\alpha\) and \(\beta\) take the meaning of the scale parameters (usually named) \(\alpha.\bar{a}\) and \(\beta.\bar{b}\).

beta shape parameter \(\alpha\); skewness parameter \(\beta\), abs(\(\beta\)) is in the range (0, \(\alpha\)); scale parameter \(\delta\). \(\delta\) must be zero or positive; location parameter \(\mu\), by default 0. These is the meaning of the parameters in the first
parameterization \( p m = 1 \) which is the default parameterization selection. In the second parameterization, \( p m = 2 \) alpha and beta take the meaning of the shape parameters (usually named) \( \zeta \) and \( \rho \). In the third parameterization, \( p m = 3 \) alpha and beta take the meaning of the shape parameters (usually named) \( \xi \) and \( \chi \). In the fourth parameterization, \( p m = 4 \) alpha and beta take the meaning of the shape parameters (usually named) \( a Nbar \) and \( bNbar \).

\( \delta \) shape parameter alpha; skewness parameter beta, \( \text{abs}(\beta) \) is in the range \((0, \alpha)\); scale parameter delta, delta must be zero or positive; location parameter mu, by default 0. These is the meaning of the parameters in the first parameterization \( p m = 1 \) which is the default parameterization selection. In the second parameterization, \( p m = 2 \) alpha and beta take the meaning of the shape parameters (usually named) \( \zeta \) and \( \rho \). In the third parameterization, \( p m = 3 \) alpha and beta take the meaning of the shape parameters (usually named) \( \xi \) and \( \chi \). In the fourth parameterization, \( p m = 4 \) alpha and beta take the meaning of the shape parameters (usually named) \( a Nbar \) and \( bNbar \).

\( \mu \) shape parameter alpha; skewness parameter beta, \( \text{abs}(\beta) \) is in the range \((0, \alpha)\); scale parameter delta, delta must be zero or positive; location parameter mu, by default 0. These is the meaning of the parameters in the first parameterization \( p m = 1 \) which is the default parameterization selection. In the second parameterization, \( p m = 2 \) alpha and beta take the meaning of the shape parameters (usually named) \( \zeta \) and \( \rho \). In the third parameterization, \( p m = 3 \) alpha and beta take the meaning of the shape parameters (usually named) \( \xi \) and \( \chi \). In the fourth parameterization, \( p m = 4 \) alpha and beta take the meaning of the shape parameters (usually named) \( a Nbar \) and \( bNbar \).

\( \lambda \) a numeric value, the number of degrees of freedom. Note, alpha takes the limit of \( \text{abs}(\beta) \), and \( \lambda = -\nu/2 \).

\( \lambda_1 \) are numeric values where \( \lambda_1 \) is the location parameter, \( \lambda_2 \) is the location parameter, \( \lambda_3 \) is the first shape parameter, and \( \lambda_4 \) is the second shape parameter.

\( \lambda_2 \) are numeric values where \( \lambda_1 \) is the location parameter, \( \lambda_2 \) is the location parameter, \( \lambda_3 \) is the first shape parameter, and \( \lambda_4 \) is the second shape parameter.

\( \lambda_3 \) are numeric values where \( \lambda_1 \) is the location parameter, \( \lambda_2 \) is the location parameter, \( \lambda_3 \) is the first shape parameter, and \( \lambda_4 \) is the second shape parameter.

\( \lambda_4 \) are numeric values where \( \lambda_1 \) is the location parameter, \( \lambda_2 \) is the location parameter, \( \lambda_3 \) is the first shape parameter, and \( \lambda_4 \) is the second shape parameter.
distrMode

pm an integer value between 1 and 4 for the selection of the parameterization. The default takes the first parameterization.

meanlog mean and standard deviation of the distribution on the log scale with default values of 0 and 1 respectively.

sdlog mean and standard deviation of the distribution on the log scale with default values of 0 and 1 respectively.

gamma value of the index parameter alpha in the interval= (0, 2]; skewness parameter beta, in the range [-1, 1]; scale parameter gamma; and location (or 'shift') parameter delta.

loc vector of (real) location parameters.

disp vector of (positive) dispersion parameters.

skew vector of skewness parameters (in [-1,1]).

tail vector of parameters (in [1,2]) related to the tail thickness.

min lower and upper limits of the distribution. Must be finite.

max lower and upper limits of the distribution. Must be finite.

prob Probability of success on each trial.

size number of trials (zero or more).

m the number of white balls in the urn.

n number of observations. If length(n) > 1, the length is taken to be the number required.

k the number of balls drawn from the urn.

Value

A numeric value is returned, the (true) mode of the distribution.

Note

Some functions like normMode or cauchyMode, which relate to symmetric distributions, are trivial, but are implemented for the sake of exhaustivity.

Author(s)

ghMode and ghtMode are from package fBasics; hypMode was written by David Scott; gldMode, nigMode and stableMode were written by Diethelm Wuertz.

See Also

mlv for the estimation of the mode; the documentation of the related distributions Beta, GammaDist, etc.
Examples

```r
## Beta distribution
curve(dbeta(x, shape1 = 2, shape2 = 3.1),
xlim = c(0,1), ylab = "Beta density")
M <- betaMode(shape1 = 2, shape2 = 3.1)
abline(v = M, col = 2)
mlv("beta", shape1 = 2, shape2 = 3.1)

## Lognormal distribution
curve(stats::dlnorm(x, meanlog = 3, sdlog = 1.1),
xlim = c(0, 10), ylab = "Lognormal density")
M <- lnormMode(meanlog = 3, sdlog = 1.1)
abline(v = M, col = 2)
mlv("lnorm", meanlog = 3, sdlog = 1.1)

curve(VGAM::dpareto(x, scale = 1, shape = 1), xlim = c(0, 10))
abline(v = paretoMode(scale = 1), col = 2)

## Poisson distribution
poisMode(lambda = 6)
poisMode(lambda = 6.1)
mlv("poisson", lambda = 6.1)
```

---

grenander

The Grenander mode estimator

Description

This function computes the Grenander mode estimator.

Usage

```r
grenander(x, bw = NULL, k, p, ...)
```

Arguments

- **x**: numeric. Vector of observations.
- **bw**: numeric. The bandwidth to be used. Should belong to (0, 1].
- **k**: numeric. Parameter 'k' in Grenander's mode estimate, see below.
- **p**: numeric. Parameter 'p' in Grenander's mode estimate, see below. If p = Inf, the function `venter` is used.
- **...**: Additional arguments to be passed to `venter`. 
Details

The Grenander estimate is defined by

\[
\sum_{j=1}^{n-k} \frac{(x_{j+k}+x_j)}{2(x_{j+k}-x_j)p} \quad \sum_{j=1}^{n-k} \frac{1}{(x_{j+k}-x_j)p}
\]

If \( p \) tends to infinity, this estimate tends to the Venter mode estimate; this justifies to call \texttt{venter} if \( p = \text{Inf} \).

The user should either give the bandwidth \( bw \) or the argument \( k \), \( k \) being taken equal to \( \text{ceiling}(bw*n) - 1 \) if missing.

Value

A numeric value is returned, the mode estimate. If \( p = \text{Inf} \), the \texttt{venter} mode estimator is returned.

Note

The user should preferentially call \texttt{grenander} through \texttt{mlv(x, method = \"grenander\", bw, k, p, ...)}.

Author(s)

D.R. Bickel for the original code, P. Poncet for the slight modifications introduced.

References


See Also

\texttt{mlv} for general mode estimation; \texttt{venter} for the Venter mode estimate.

Examples

```r
# Unimodal distribution
x <- rnorm(1000, mean = 23, sd = 0.5)

## True mode
normMode(mean = 23, sd = 0.5) # (!)

## Parameter 'k'
k <- 5
```
## Description

This function computes Bickel’s half range mode estimator described in Bickel (2002). It is a wrapper around the function `half.range.mode` from package `genefilter`.

## Usage

```r
hrm(x, bw = NULL, ...)
```

## Arguments

- `x` numeric. Vector of observations.
- `bw` numeric. The bandwidth to be used. Should belong to (0, 1]. This gives the fraction of the observations to consider at each step of the iterative algorithm.
- `...` Additional arguments.

## Details

The mode estimator is computed by iteratively identifying densest half ranges. A densest half range is an interval whose width equals half the current range, and which contains the maximal number of observations. The subset of observations falling in the selected densest half range is then used to compute a new range, and the procedure is iterated.

## Value

A numeric value is returned, the mode estimate.

## Note

The user should preferentially call `hrm` through `mlv(x, method = "hrm", bw, ...)`.

## Author(s)

The C and R code are due to Richard Bourgon <bourgon@stat.berkeley.edu>, see package `genefilter`. The algorithm is described in Bickel (2002).
References


See Also

*mlv* for general mode estimation; *hsm* for the half sample mode; *venter* for the Venter mode estimate.

Examples

```r
# Unimodal distribution
x <- rgamma(1000, shape = 31.9)
## True mode
gammaMode(shape = 31.9)

## Estimate of the mode
hrm(x, bw = 0.4)
mlv(x, method = "hrm", bw = 0.4)
```

---

**Description**

This function computes the Robertson-Cryer mode estimator described in Robertson and Cryer (1974), also called half sample mode (if bw = 1/2) or fraction sample mode (for some other bw) by Bickel (2006).

**Usage**

```r
hsm(x, bw = NULL, k, tie.action = "mean", tie.limit = 0.05, ...)
```

**Arguments**

- **x** numeric. Vector of observations.
- **bw** numeric or function. The bandwidth to be used. Should belong to (0, 1].
- **k** numeric. See 'Details'.
- **tie.action** character. The action to take if a tie is encountered.
- **tie.limit** numeric. A limit deciding whether or not a warning is given when a tie is encountered.
- **...** Additional arguments.
Details

The modal interval, i.e. the shortest interval among intervals containing \( k+1 \) observations, is computed iteratively, until only one value is found, the mode estimate. At each step \( i \), one takes \( k = \text{ceiling}(bw*n) - 1 \), where \( n \) is the length of the modal interval computed at step \( i-1 \). If \( bw \) is of class "function", then \( k = \text{ceiling}(bw(n)) - 1 \) instead.

Value

A numeric value is returned, the mode estimate.

Note

The user should preferentially call \texttt{hsm} through \texttt{mlv(x, method = "hsm", ...)}.

Author(s)

D.R. Bickel for the original code, P. Poncet for the slight modifications introduced.

References


See Also

\texttt{mlv} for general mode estimation; \texttt{venter} for the Venter mode estimate.

Examples

```r
# Unimodal distribution
x <- rweibull(10000, shape = 3, scale = 0.9)

## True mode
weibullMode(shape = 3, scale = 0.9)

## Estimate of the mode
bandwidth <- function(n, alpha) {1/n^alpha}

hsm(x, bw = bandwidth, alpha = 2)
mlv(x, method = "hsm", bw = bandwidth, alpha = 2)
```
The empirical Lientz function and the Lientz mode estimator

Description

The Lientz mode estimator is nothing but the value minimizing the empirical Lientz function. A 'plot' and a 'print' methods are provided.

Usage

lientz(x, bw = NULL)

## S3 method for class 'lientz'
plot(x, zoom = FALSE, ...)

## S3 method for class 'lientz'
print(x, digits = NULL, ...)

## S3 method for class 'lientz'
mlv(x, bw = NULL, abc = FALSE, par = shorth(x),
    optim.method = "BFGS", ...)  

Arguments

x numeric (vector of observations) or an object of class "lientz".
bw numeric. The smoothing bandwidth to be used. Should belong to (0, 1). Parameter 'beta' in Lientz (1970) function.
zoom logical. If TRUE, one can zoom on the graph created.
... if abc = FALSE, further arguments to be passed to optim, or further arguments to be passed to plot.
digits numeric. Number of digits to be printed.
abc logical. If FALSE (the default), the Lientz empirical function is minimised using optim.
par numeric. The initial value used in optim.
optim.method character. If abc = FALSE, the method used in optim.

Details

The Lientz function is the smallest non-negative quantity $S(x, \beta)$, where $\beta = bw$, such that

$$ F(x + S(x, \beta)) - F(x - S(x, \beta)) \geq \beta. $$

Lientz (1970) provided a way to estimate $S(x, \beta)$; this estimate is what we call the empirical Lientz function.
Value

`llientz` returns an object of class c("llientz", "function"); this is a function with additional attributes:

- `x` the `x` argument
- `bw` the `bw` argument
- `call` the call which produced the result

`mlv.llientz` returns a numeric value, the mode estimate. If `abc = TRUE`, the `x` value minimizing the Lientz empirical function is returned. Otherwise, the `optim` method is used to perform minimization, and the attributes: 'value', 'counts', 'convergence' and 'message', coming from the `optim` method, are added to the result.

Note

The user should preferentially call `mlv.llientz` through `mlv(x, method = "llientz", ...)`.

References


See Also

`mlv` for general mode estimation; `shorth` for the shorth estimate of the mode

Examples

```r
# Unimodal distribution
x <- rbeta(1000, 23, 4)

## True mode
betaMode(23, 4)

## Lientz object
f <- llientz(x, 0.2)
print(f)
plot(f)

## Estimate of the mode
mlv(f)  # optim(shorth(x), fn = f)
mlv(f, abc = TRUE)  # x[which.min(f(x))]
mlv(x, method = "llientz", bw = 0.2)

# Bimodal distribution
x <- c(rnorm(1000, 5, 1), rnorm(1500, 22, 3))
```
**meanshift**

\[
f <- l entail(x, 0.1) 
plot(f)
\]

---

**meanshift**  
*The Meanshift mode estimator*

**Description**

The Meanshift mode estimator.

**Usage**

meanshift(x, bw = NULL, kernel = "gaussian", par = shorth(x),
iter = 1000, tolerance = sqrt(.Machine$double.eps))

**Arguments**

- **x**  
  numeric. Vector of observations.

- **bw**  
  numeric. The smoothing bandwidth to be used.

- **kernel**  
  character. The kernel to be used. Available kernels are "biweight", "cosine", "eddy", "epanechnikov", "gaussian", "optcosine", "rectangular", "triangular", "uniform". See `density` for more details on some of these kernels.

- **par**  
  numeric. The initial value used in the meanshift algorithm.

- **iter**  
  numeric. Maximal number of iterations.

- **tolerance**  
  numeric. Stopping criteria.

**Value**

meanshift returns a numeric value, the mode estimate, with an attribute "iterations". The number of iterations can be less than iter if the stopping criteria specified by eps is reached.

**Note**

The user should preferentially call meanshift through `mlv(x, method = "meanshift", ...).

**References**


**See Also**

`mlv`, `tsybakov`. 
Examples

# Unimodal distribution
x <- rweibull(100, shape = 12, scale = 0.8)

## True mode
weibullMode(shape = 12, scale = 0.8)

## Estimate of the mode
mlv(x, method = "meanshift", par = mean(x))

mlv

Estimation of the Mode(s) or Most Likely Value(s)

Description

mlv is a generic function for estimating the mode of a univariate distribution. Different estimates (or methods) are provided:

- `mfv`, which returns the most frequent value(s) in a given numerical vector,
- the `Lientz` mode estimator, which is the value minimizing the Lientz function estimate,
- the Chernoff mode estimator, also called naive mode estimator, which is defined as the center of the interval of given length containing the most observations,
- the `Venter` mode estimator, including the `shortH`, i.e. the midpoint of the modal interval,
- the `Grenander` mode estimator,
- the half sample mode (`HSM`) and the half range mode (`HRM`), which are iterative versions of the Venter mode estimator,
- Parzen's kernel mode estimator, which is the value maximizing the kernel density estimate,
- the `Tsybakov` mode estimator, based on a gradient-like recursive algorithm,
- the Asselin de Beauville mode estimator, based on an algorithm detecting chains and holes in the sample,
- the `Vieu` mode estimator,
- the `meanshift` mode estimator.

mlv can also be used to compute the mode of a given distribution, with mlv.character.

Usage

mlv(x, ...)

## S3 method for class 'character'
mlv(x, ...)

## S3 method for class 'factor'
mlv(x, ...)

### S3 method for class 'logical'
\texttt{mlv(x, ...)}

### S3 method for class 'integer'
\texttt{mlv(x, na.rm = FALSE, ...)}

### Default S3 method:
\texttt{mlv(x, bw = NULL, method, na.rm = FALSE, ...)}

### S3 method for class 'density'
\texttt{mlv(x, all = TRUE, abc = FALSE, ...)}

\texttt{mlvl(x, ...)}

**Arguments**

- **x**: numeric (vector of observations), or an object of class "factor", "integer", etc. For the function \texttt{as.numeric}, an object of class "mlv".
- **...**: Further arguments to be passed to the function called for computation; this function is related to the method argument.
- **na.rm**: logical. Should missing values be removed?
- **bw**: numeric. The bandwidth to be used. This may have different meanings regarding the method used.
- **method**: character. One of the methods available for computing the mode estimate. See 'Details'.
- **all**: logical.
- **abc**: logical. If FALSE (the default), the estimate of the density function is maximised using \texttt{optim}.

**Details**

For the function \texttt{mlv.default}, available methods are "mfv", "lientz", "naive", "venter", "grenander", "hsm", "hrm", "parzen", "tsybakov", and "asselin". See the description above and the associated links.

If \texttt{x} is of class "factor" or "integer", the most frequent value found in \texttt{x} is returned.

If \texttt{x} is of class "character", \texttt{x} should be one of "beta", "cauchy", "gev", etc. i.e. a character for which a function \texttt{mode} exists (for instance betaMode, cauchyMode, etc.). See \texttt{distrMode} for the available functions. The mode of the corresponding distribution is returned.

If \texttt{x} is of class "density", the value where the density is maximised is returned. For the S3 function \texttt{mlvl.lientz}, see \texttt{Lientz} for more details.

**Value**

A vector of the same type as \texttt{x}. Be aware that the length of this vector can be > 1.
References

See the references on mode estimation on the modeest-package’s page.

See Also

mfv, Lientz, naive, venter, grenander, hrm, hsm, parzen, tsybakov, skewness

Examples

```r
# Unimodal distribution
x <- rbeta(1000, 23, 4)

## True mode
betaMode(23, 4)
# or
mlv("beta", 23, 4)

## Estimate of the mode
mlv(x, method = "lientz", bw = 0.2)
mlv(x, method = "naive", bw = 1/3)
mlv(x, method = "venter", type = "shorth")
mlv(x, method = "grenander", p = 4)
mlv(x, method = "hsm")
mlv(x, method = "parzen", kernel = "gaussian")
mlv(x, method = "tsybakov", kernel = "gaussian")
mlv(x, method = "asselin", bw = 2/3)
mlv(x, method = "vieu")
mlv(x, method = "meanshift")
```

modeest  

Mode Estimation

Description

This package provides estimators of the mode of univariate unimodal (and sometimes multimodal) data, and values of the modes of usual probability distributions.

For a complete list of functions, use `library(help = "modeest")` or `help.start()`.

References


See Also

`mlv` for general mode estimation.
**Description**

This estimator, also called the "naive" mode estimator, is defined as the center of the interval of given length containing the most observations. It is identical to Parzen's kernel mode estimator, when the kernel is chosen to be the uniform kernel.

**Usage**

```r
native(x, bw = 1/2)
```

**Arguments**

- `x`: numeric. Vector of observations.
- `bw`: numeric. The smoothing bandwidth to be used. Should belong to (0, 1). See below.

**Value**

A numeric vector is returned, the mode estimate, which is the center of the interval of length 2*bw containing the most observations.

**Note**

The user should preferentially call `native` through `mlv(x, method = "native", bw)`.

**References**


**See Also**

- `mlv` for general mode estimation; `parzen` for Parzen's kernel mode estimation.

**Examples**

```r
# Unimodal distribution
x <- rf(10000, df1 = 40, df2 = 30)

## True mode
fMode(df1 = 40, df2 = 30)

## Estimate of the mode
mean(native(x, bw = 1/4))
```
Description
Parzen’s kernel mode estimator is the value maximizing the kernel density estimate.

Usage
parzen(x, bw = NULL, kernel = "gaussian", abc = FALSE,
       par = shorth(x), optim.method = "BFGS", ...)

Arguments
x numeric. Vector of observations.
bw numeric. The smoothing bandwidth to be used.
kernel character. The kernel to be used. Available kernels are "biweight", "cosine", "eddy", "epanechnikov", "gaussian", "optcosine", "rectangular", "triangular", "uniform". See density for more details on some of these kernels.
abc logical. If FALSE (the default), the kernel density estimate is maximised using optim.
par numeric. The initial value used in optim.
optim.method character. If abc = FALSE, the method used in optim.
... If abc = FALSE, further arguments to be passed to optim.

Details
If kernel = "uniform", the naive mode estimate is returned.

Value
parzen returns a numeric value, the mode estimate. If abc = TRUE, the x value maximizing the density estimate is returned. Otherwise, the optim method is used to perform maximization, and the attributes: 'value', 'counts', 'convergence' and 'message', coming from the optim method, are added to the result.

Note
The user should preferentially call parzen through mlv(x, method = "kernel", ...) or mlv(x, method = "parzen", ...).
Presently, parzen is quite slow.
skewness

References


See Also

`mlv, naive`

Examples

```r
# Unimodal distribution
x <- rlnorm(10000, meanlog = 3.4, sdlog = 0.2)

## True mode
lnormMode(meanlog = 3.4, sdlog = 0.2)

## Estimate of the mode
mlv(x, method = "kernel", kernel = "gaussian", bw = 0.3, par = shorth(x))
```

<table>
<thead>
<tr>
<th>skewness</th>
<th>Skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Description

This function encodes different methods to calculate the skewness from a vector of observations.

Usage

```r
skewness(x, na.rm = FALSE, method = c("moment", "fisher", "bickel"), M, ...)```
Arguments

\begin{itemize}
\item \texttt{x} numeric. Vector of observations.
\item \texttt{na.rm} logical. Should missing values be removed?
\item \texttt{method} character. Specifies the method of computation. These are either "moment", "fisher" or "bickel". The "moment" method is based on the definition of skewness for distributions; this form should be used when resampling (bootstrap or jackknife). The "fisher" method corresponds to the usual "unbiased" definition of sample variance, although in the case of skewness exact unbiasedness is not possible.
\item \texttt{M} numeric. (An estimate of) the mode of the observations \texttt{x}. Default value is \texttt{shorth(x)}.
\item \ldots Additional arguments.
\end{itemize}

Value

\texttt{skewness} returns a numeric value. An attribute reports the method used.

Author(s)

Diethelm Wuertz and contributors for the original \texttt{skewness} function from package \texttt{fBasics}.

References

\begin{itemize}
\end{itemize}

See Also

\texttt{mlv} for general mode estimation; \texttt{shorth} for the shorth estimate of the mode

Examples

\begin{verbatim}
## Skewness = 0
x <- rnorm(1000)
skewness(x, method = "bickel", M = shorth(x))

## Skewness > 0 (left skewed case)
x <- rbeta(1000, 2, 5)
skewness(x, method = "bickel", M = betaMode(2, 5))

## Skewness < 0 (right skewed case)
x <- rbeta(1000, 7, 2)
skewness(x, method = "bickel", M = hsm(x, bw = 1/3))
\end{verbatim}
The Tsybakov mode estimator

Description

This mode estimator is based on a gradient-like recursive algorithm, more adapted for online estimation. It includes the Mizoguchi-Shimura (1976) mode estimator, based on the window training procedure.

Usage

```r
tsybakov(x, bw = NULL, a, alpha = 0.9, kernel = "triangular",
        dmp = TRUE, par = shorth(x))
```

Arguments

- `x` numeric. Vector of observations.
- `bw` numeric. Vector of length `length(x)` giving the sequence of smoothing bandwidths to be used.
- `a` numeric. Vector of length `length(x)` used in the gradient algorithm
- `alpha` numeric. An alternative way of specifying `a`. See 'Details'.
- `kernel` character. The kernel to be used. Available kernels are "biweight", "cosine", "eddy", "epanechnikov", "gaussian", "optcosine", "rectangular", "triangular", "uniform". See `density` for more details on some of these kernels.
- `dmp` logical. If `TRUE`, Djeddour et al. version of the estimate is used.
- `par` numeric. Initial value in the gradient algorithm. Default value is `shorth(x)`.

Details

If `bw` or `a` is missing, a default value advised by Djeddour et al (2003) is used: `bw = (1:length(x))^(-1/7)` and `a = (1:length(x))^(-alpha)`. (with `alpha = 0.9` if `alpha` is missing).

Value

A numeric value is returned, the mode estimate.

Warning

The Tsybakov mode estimate as it is presently computed does not work very well. The reasons of this inefficiency should be further investigated.

Note

The user should preferentially call `tsybakov` through `mlv(x, method = "tsybakov", ...).`
References


See Also

mlv for general mode estimation.

Examples

```r
x <- rbeta(1000, shape1 = 2, shape2 = 5)

## True mode:
betaMode(shape1 = 2, shape2 = 5)

## Estimation:
tsybakov(x, kernel = "triangular")
tsybakov(x, kernel = "gaussian", alpha = 0.99)
mlv(x, method = "tsybakov", kernel = "gaussian", alpha = 0.99)
```

Description

This function computes the Venter mode estimator, also called the Dalenius, or LMS (Least Median Square) mode estimator.

Usage

```r
venter(x, bw = NULL, k, iter = 1, type = 1, tie.action = "mean",
       tie.limit = 0.05)

shorth(x, ...)
```
venter

Arguments

- **x** numeric. Vector of observations.
- **bw** numeric. The bandwidth to be used. Should belong to (0, 1]. See ‘Details’.
- **k** numeric. See ‘Details’.
- **iter** numeric. Number of iterations.
- **type** numeric or character. The type of Venter estimate to be computed. See ‘Details’.
- **tie.action** character. The action to take if a tie is encountered.
- **tie.limit** numeric. A limit deciding whether or not a warning is given when a tie is encountered.
- ... Further arguments.

Details

The modal interval, i.e. the shortest interval among intervals containing \( k + 1 \) observations, is first computed. The user should either give the bandwidth \( \text{bw} \) or the argument \( \text{k} \), \( \text{k} \) being taken equal to \( \text{ceiling(} \text{bw}*n) - 1 \) if missing.

If \( \text{type} = 1 \), the midpoint of the modal interval is returned. If \( \text{type} = 2 \), the \( \text{floor((k+1)/2)} \)th element of the modal interval is returned. If \( \text{type} = 3 \) or \( \text{type} = \text{"dalenius"} \), the median of the modal interval is returned. If \( \text{type} = 4 \) or \( \text{type} = \text{"shorth"} \), the mean of the modal interval is returned. If \( \text{type} = 5 \) or \( \text{type} = \text{"ekblom"} \), Ekblom’s \( L_{-\infty} \) estimate is returned, see Ekblom (1972). If \( \text{type} = 6 \) or \( \text{type} = \text{"hsm"} \), the half sample mode (hsm) is computed, see hsm.

Value

A numeric value is returned, the mode estimate.

Note

The user should preferentially call venter through \text{mlv(x, method = "venter", ...)}. 

References


See Also

\text{mlv} for general mode estimation, \text{hsm} for the half sample mode.
Examples

library(evd)

# Unimodal distribution
x <- rgev(1000, loc = 23, scale = 1.5, shape = 0)

## True mode
gevMode(loc = 23, scale = 1.5, shape = 0)

## Estimate of the mode
venter(x, bw = 1/3)
mlv(x, method = "venter", bw = 1/3)

---

vieu Vieu's mode estimator

Description

Vieu's mode estimator is the value at which the kernel density derivative estimate is null.

Usage

vieu(x, bw = NULL, kernel = "gaussian", abc = FALSE, ...)

Arguments

x numeric. Vector of observations.
bw numeric. The smoothing bandwidth to be used.
kernel character. The kernel to be used. Available kernels are "biweight", "cosine", 
"eddy","epanechnikov","gaussian","optcosine","rectangular","triangular", 
"uniform". See density for more details on some of these kernels.
abc logical. If FALSE (the default), the root of the density derivate estimate is searched 
with uniroot.
...
If abc = FALSE, further arguments to be passed to uniroot.

Value

vieu returns a numeric value, the mode estimate. If abc = TRUE, the x value at which the density 
derivative estimate is null is returned. Otherwise, the uniroot method is used.

Note

The user should preferentially call vieu through mlv(x, method = "vieu", ...).
Presently, vieu is quite slow.
References


See Also

`mlv`, `parzen`.

Examples

```r
# Unimodal distribution
x <- rlnorm(10000, meanlog = 3.4, sdlog = 0.2)

## True mode
lnormMode(meanlog = 3.4, sdlog = 0.2)

## Estimate of the mode
mlv(x, method = "vieu", kernel = "gaussian")
```
Index

*Topic **distribution**
  modeest, 18

*Topic **nonparametric**
  modeest, 18

*Topic **robust**
  modeest, 18

*Topic **univar**
  modeest, 18

Asselin, 16
Asselin (asselin), 2
asselin, 2

bernMode (distrMode), 3
Beta, 7
betaMode (distrMode), 3
binomMode (distrMode), 3

cauchyMode (distrMode), 3
Chernoff (naive), 21
chernoff (naive), 21
chisqMode (distrMode), 3

dagumMode (distrMode), 3
density, 15, 22, 25, 28
distrMode, 3, 17

expMode (distrMode), 3

fiskMode (distrMode), 3
fMode (distrMode), 3
frechetMode (distrMode), 3

GammaDist, 7
gammaMode (distrMode), 3
ggeomMode (distrMode), 3
gevMode (distrMode), 3
ghMode, 7
ghmMode (distrMode), 3
ghtMode, 7
ghtMode (distrMode), 3

gldMode, 7
gldMode (distrMode), 3
gompertzMode (distrMode), 3
gpdMode (distrMode), 3
Grenander, 16
Grenander (grenander), 8
grenander, 8, 18
gumbelMode (distrMode), 3

half-range.mode, 10
HRM, 16
HRM (hrm), 10
hrm, 10, 18
HSM, 16
HSM (hsm), 11
hsm, 11, 11, 18, 27
hyperMode (distrMode), 3
hypoMode, 7
hypoMode (distrMode), 3

koenkerMode (distrMode), 3
kumarMode (distrMode), 3

laplaceMode (distrMode), 3
Lientz, 16–18
Lientz (lientz), 13
lientz, 13
lnormMode (distrMode), 3
logisMode (distrMode), 3
lomaxMode (distrMode), 3

maxwellMode (distrMode), 3
meanshift, 15, 16
mfv, 16, 18
mlv, 2, 7, 9, 11, 12, 14, 15, 16, 20, 21, 23, 24, 26, 27, 29
mlv.lientz (lientz), 13
mlv1 (mlv), 16
modeest, 18
modeest-package (modeest), 18

30
mvnormMode (distrMode), 3

naive, 16, 18, 21, 22, 23
nakaMode (distrMode), 3
nbinomMode (distrMode), 3
nigMode, 7
nigMode (distrMode), 3
normMode (distrMode), 3

optim, 13, 14, 17, 22

paralogisticMode (distrMode), 3
paretoMode (distrMode), 3
Parzen, 16
Parzen (parzen), 22
parzen, 18, 21, 22, 29
plot, 13
plot.lentz (lentz), 13
poisMode (distrMode), 3
print.lentz (lentz), 13

quantile, 2

rayleighMode (distrMode), 3

shorth, 14, 16, 24, 25
shorth (venter), 26
skewness, 18, 23
stableMode, 7
stableMode (distrMode), 3
stableMode2 (distrMode), 3

tMode (distrMode), 3
Tsybakov, 16
Tsybakov (tsybakov), 25
tsybakov, 15, 18, 25

unifMode (distrMode), 3
unifroot, 28

Venter, 16
Venter (venter), 26
venter, 8, 9, 11, 12, 18, 26
Vieu, 16
Vieu (vieu), 28
vieu, 28

weibullMode (distrMode), 3

yulesMode (distrMode), 3