The Modeest Package

Paul PONCET

May 23, 2009

`asselin` *The Asselin de Beauville Mode Estimator*

Description

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

Usage

```
asselin(x,
        bw = 1,
        ...)
```

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.

Details

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

Value

A numeric value is returned, the mode estimate.

Note

The user should preferentially call `asselin` through `mlv(x, method = "asselin", ...)`. This returns an object of class `mlv`.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Asselin de Beauville J.-P. (1978). Estimation non parametrique de la densite et du mode, exemple de la distribution Gamma. *Revue de Statistique Appliquee*, **26**(3):47-70.

See Also

`mlv` for general mode estimation

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)
M <- mlv(x, method = "asselin")
print(M)
plot(M)
```

`mfv` — *Estimate of the Mode of a Discrete Distribution (Most Frequent Value)*

Description

This function returns the most frequent value(s) in a given numerical vector.

Usage

```
mfv(x, ...)
```

Arguments

`x`	numeric. Vector of observations.
`...`	further arguments, which will be ignored.

Details

Argument `x` is to come from a discrete distribution. This function uses function `tabulate` of R.

Value

The most frequent value(s) found in `x` is (are) returned.

Note

The user should preferentially call `mfv` through `mlv(x, method = "mfv")` (or `mlv(x, method = "discrete")`). This returns an object of class `mlv`.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

See Also

`mlv` for general mode estimation; `geomMode`, `poisMode`, etc. for computation of the mode of the usual discrete distributions

Examples

```
# Unimodal distribution
x <- rbinom(100, size = 10, prob = 0.8)

## True mode
binomMode(size = 10, prob = 0.8)

## Most frequent value
mfv(x)
mlv(x, method = "discrete")

# Bimodal distribution
x <- rpois(100, lambda = 7)

## True mode
poisMode(lambda = 7)

## Most frequent value
mfv(x)
M <- mlv(x, method = "discrete")
print(M)
plot(M)
```

`distribMode` *Computing the Mode of Some Distributions*

Description

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

Usage

```
## Continuous distributions
betaMode(shape1, shape2, ncp = 0) # Beta
cauchyMode(location = 0, ...) # Cauchy
chisqMode(df, ncp = 0) # Chisquare
expMode(...) # Exponentiel
fMode(df1, df2) # F
frechetMode(loc = 0, scale = 1, shape = 1, ...) # Frechet (package 'evd')
gammaMode(shape, rate = 1, scale = 1/rate) # Gamma
normMode(mean = 0, ...) # Normal (Gaussian)
gevMode(loc = 0, scale = 1, shape = 0, ...) # Generalised Extreme Value (package 'evd')
ghMode(alpha = 1, beta = 0, delta = 1, mu = 0,
       lambda = 1, ...) # Generalised Hyperbolic (package 'fBasics')
gpdMode(loc = 0, scale = 1, shape = 0, ...) # Generalised Pareto (package 'evd')
gumbelMode(loc = 0, ...) # Gumbel (package 'evd')
hypMode(alpha = 1, beta = 0, delta = 1, mu = 0,
        pm = c(1, 2, 3, 4)) # Hyperbolic (package 'fBasics')
logisMode(location = 0, ...) # Logistic
lnormMode(meanlog = 0, sdlog = 1) # Lognormal
nigMode(alpha = 1, beta = 0, delta = 1,
        mu = 0, ...) # Normal Inverse Gaussian (package 'fBasics')
stableMode(alpha, beta, gamma = 1, delta = 0, pm = 0, ...) # Stable (package 'fBasics')
symstbMode(...) # Symmetric stable (package 'fBasics')
rweibullMode(loc = 0, scale = 1, shape = 1, ...) # Negative Weibull (package 'evd')
tMode(df, ncp = 0) # T (Student)
unifMode(min = 0, max = 1) # Uniform
weibullMode(shape, scale = 1, ...) # Weibull

## Discrete distributions
bernMode(prob) # Bernoulli
binomMode(size, prob) # Binomial
geomMode(...) # Geometric
hyperMode(m, n, k, ...) # Hypergeometric
nbinomMode(size, prob, mu) # Negative Binomial
poisMode(lambda) # Poisson
```

Arguments

`shape1, shape2, ncp, location, df, df1, df2, loc, scale, shape,`

`rate, mean, alpha, beta, delta, mu, lambda, pm, meanlog, sdlog,`

`gamma, min, max, prob, size, m, n, k`
: The different arguments are those of the corresponding distribution functions.

`...`
: Further arguments, which will be ignored.

Value

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

Note

Some functions like `normMode` or `cauchyMode`, which are related to symmetric distributions, are trivial, but are implemented for exhaustivity.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩, except for `hypMode` and `stableMode` written by Diethelm Wuertz, see package **fBasics**.

See Also

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

Examples

```
layout(mat = matrix(1:2,1,2))

## 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(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)

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

layout(mat = matrix(1,1,1))
```

`grenander` *The Grenander Mode Estimator*

Description

This function computes the Grenander mode estimator.

Usage

```
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. Paramater 'k' in Grenander's mode estimate.
- `p` numeric. Paramater 'p' in Grenander's mode estimate. If `p = Inf`, function `venter` is used.
- `...` further arguments to be passed to `link{venter}`

Details

The Grenander estimate is defined by

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

If p tends to infinity, this estimate tends to the Venter mode estimate; this justifies to call `venter` if `p = Inf`.

The user should either give the bandwidth `bw` or the argument `k`, `k` being taken equal to `ceiling(bw*ny) - 1` if missing.

Value

A numeric value is returned, the mode estimate. If `p = Inf`, the `Venter` mode estimator is returned.

Note

The user should preferentially call `grenander` through `mlv(x, method = "grenander", bw, k, p, ...)`. This returns an object of class `mlv`.

Author(s)

D.R. Bickel for the original code,
Paul Poncet ⟨paulponcet@yahoo.fr⟩ for the slight modifications introduced.

References

- Grenander U. (1965). Some direct estimates of the mode. *Ann. Math. Statist.*, **36**:131-138.
- Dalenius T. (1965). The Mode - A Negleted Statistical Parameter. *J. Royal Statist. Soc. A*, *128*:110-117.
- Adriano K.N., Gentle J.E. and Sposito V.A. (1977). On the asymptotic bias of Grenander's mode estimator. *Commun. Statist.-Theor. Meth. A*, **6**:773-776.
- Hall P. (1982). Asymptotic Theory of Grenander's Mode Estimator. *Z. Wahrsch. Verw. Gebiete*, **60**:315-334.

See Also

`mlv` for general mode estimation; `venter` for the Venter mode estimate

Examples

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

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

## Parameter 'k'
k <- 5

## Many values of parameter 'p'
p <- seq(0.1, 4, 0.01)

## Estimate of the mode with these parameters
M <- sapply(p, function(pp) grenander(x, p = pp, k = k))

## Distribution obtained
plot(density(M), xlim = c(22.5, 23.5))
```

`hrm` *Half-range Mode*

Description

This function computes Bickel's half range mode estimator described in Bickel (2002).

Usage

```
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.
`...`	further 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)`. This returns an object of class `mlv`.

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

- Bickel D.R. (2002). Robust estimators of the mode and skewness of continuous data. *Computational Statistics and Data Analysis*, **39**:153-163.
- Hedges S.B. and Shah P. (2003). Comparison of mode estimation methods and application in molecular clock analysis. *BMC Bioinformatics*, **4**:31-41.
- Bickel D.R. et Fruehwirth R. (2006). On a Fast, Robust Estimator of the Mode: Comparisons to Other Robust Estimators with Applications. *Computational Statistics and Data Analysis*, **50**(12):3500-3530.

See Also

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

Examples

```
# Unimodal distribution
x <- rgamma(1000, shape = 31.9)

## True mode
```

```
gammaMode(shape = 31.9)

## Estimate of the mode
hrm(x, bw = 0.4)
M <- mlv(x, method = "hrm", bw = 0.4)
print(M)
plot(M)
```

`hsm` *Half Sample Mode*

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

```
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.
`...`	further 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 = 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 = ceiling(bw(n)) - 1` instead.

Value

A numeric value is returned, the mode estimate.

Note

The user should preferentially call `hsm` through `mlv(x, method = "hsm", ...)`. This returns an object of class `mlv`.

Author(s)

D.R. Bickel for the original code,
Paul Poncet ⟨paulponcet@yahoo.fr⟩ for the slight modifications introduced.

References

- Robertson T. and Cryer J.D. (1974). An iterative procedure for estimating the mode. *J. Amer. Statist. Assoc.*, **69**(348):1012-1016.
- Bickel D.R. et Fruehwirth R. (2006). On a Fast, Robust Estimator of the Mode: Comparisons to Other Robust Estimators with Applications. *Computational Statistics and Data Analysis*, **50**(12):3500-3530.

See Also

`mlv` for general mode estimation; `venter` for the Venter mode estimate

Examples

```
# 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)
M <- mlv(x, method = "hsm", bw = bandwidth, alpha = 2)
print(M)
plot(M)
```

`lientz` *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':
mlv(x,
    bw = NULL,
    biau = FALSE,
    par = shorth(x),
    optim.method = "BFGS",
    ...)

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

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

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.
`biau`	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 `biau = FALSE`, the method used in `optim`.
`zoom`	logical. If `TRUE`, one can zoom on the graph created.
`digits`	numeric. Number of digits to be printed.
`...`	if `biau = FALSE`, further arguments to be passed to `optim`, or further arguments to be passed to `plot.default`.

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

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

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

`mlv.lientz` returns a numeric value, the mode estimate. If `biau = 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.lientz` through `mlv(x, method = "lientz", ...)`. This returns an object of class `mlv`.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Lientz B.P. (1969). On estimating points of local maxima and minima of density functions. *Nonparametric Techniques in Statistical Inference (ed. M.L. Puri, Cambridge University Press*, p.275-282.
- Lientz B.P. (1970). Results on nonparametric modal intervals. *SIAM J. Appl. Math.*, **19**:356-366.
- Lientz B.P. (1972). Properties of modal intervals. *SIAM J. Appl. Math.*, **23**:1-5.

See Also

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

Examples

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

## True mode
betaMode(23, 4)

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

## Estimate of the mode
mlv(f)              # optim(shorth(x), fn = f)
```

```
mlv(f, biau = TRUE) # x[which.min(f(x))]
M <- mlv(x, method = "lientz", bw = 0.2)
print(M)
plot(M)

# Bimodal distribution
x <- c(rnorm(1000,5,1), rnorm(1500, 22, 3))
f <- lientz(x, 0.1)
plot(f)
```

---

mlv *Estimation of the Mode*

Description

mlv is a generic function which enables to compute an estimate of the mode of a univariate distribution. Many 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.

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

A 'plot' and a 'print' methods are provided.

Usage

```
mlv(x,
    ...)

## Default S3 method:
mlv(x,
    bw = NULL,
    method,
    na.rm = FALSE,
    boot = FALSE,
```

```
    R = 100,
    B = length(x),
    ...)

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

## S3 method for class 'integer':
mlv(x,
    na.rm = FALSE,
    ...)

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

## S3 method for class 'density':
mlv(x,
    all = TRUE,
    abc = FALSE,
    ...)

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

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

## S3 method for class 'mlv':
as.numeric(x,
           ...)
```

Arguments

`x`	numeric (vector of observations), or an object of class `"factor"`, `"integer"`, etc. For the function `as.numeric`, an object of class `"mlv"`.
`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'.
`na.rm`	logical. Should missing values be removed?
`boot`	logical. Should bootstrap resampling be done?
`R`	numeric. If `boot = TRUE`, the number of bootstrap resampling rounds to use.

`B`	numeric. If `boot = TRUE`, the size of the bootstrap samples drawn from `x`. Default is to use a sample which is the same size as data. For large data sets, this may be slow and unnecessary.
`all`	logical.
`abc`	logical. If `FALSE` (the default), the estimate of the density function is maximised using `optim`.
`digits`	numeric. Number of digits to be printed.
`...`	Further arguments to be passed to the function called for computation. This function is related to the `method` argument.

Details

For the function `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 `x` is of class `"factor"` or `"integer"`, the most frequent value found in `x` is returned.
If `x` is of class `"character"`, `x` should be one of `"beta"`, `"cauchy"`, `"gev"`, etc. i.e. a character for which a function `'x'Mode` exists (for instance `betaMode`, `cauchyMode`, etc.). See `distribMode` for the available functions. The mode of the corresponding distribution is returned.
If `x` is of class `"density"`, the value where the density is maximised is returned.
For the S3 function `mlv.lientz`, see `Lientz` for more details.

Value

`mlv` returns an object of `class` `"mlv"`.
An object of class `"mlv"` is a list containing at least the following components:

`M`	the value of the mode
`skewness`	Bickel's measure of `skewness`
`x`	the argument `x`
`method`	the argument `method`
`bw`	the bandwidth
`boot`	the argument `boot`
`boot.M`	if `boot = TRUE`, the resampled values of the mode
`call`	the call which produced the result

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

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

```
# 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 = "hrm", bw = 0.3)
mlv(x, method = "hsm")
mlv(x, method = "parzen", kernel = "gaussian")
mlv(x, method = "tsybakov", kernel = "gaussian")
mlv(x, method = "asselin", bw = 2/3)

## Bootstrap
M <- mlv(x, method = "kernel", boot = TRUE, R = 150)
print(M)
plot(M)
print(mean(M[["boot.M"]]))
```

`modeest` *Mode Estimation*

Description

This package intends to provide 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()`.

Details

Package:	modeest
Type:	Package
Version:	1.09
Date:	2009-05-23
License:	GPL version 2 or newer

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

Maintainer: Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Parzen E. (1962). On estimation of a probability density function and mode. *Ann. Math. Stat.*, **33**(3):1065-1076.
- Chernoff H. (1964). Estimation of the mode. *Ann. Inst. Statist. Math.*, **16**:31-41.
- Huber P.J. (1964). Robust estimation of a location parameter. *Ann. Math. Statist.*, **35**:73-101.
- Dalenius T. (1965). The Mode - A Negleted Statistical Parameter. *J. Royal Statist. Soc. A*, **128**:110-117.
- Grenander U. (1965). Some direct estimates of the mode. *Ann. Math. Statist.*, **36**:131-138.
- Venter J.H. (1967). On estimation of the mode. *Ann. Math. Statist.*, **38**(5):1446-1455.
- Lientz B.P. (1969). On estimating points of local maxima and minima of density functions. *Nonparametric Techniques in Statistical Inference (ed. M.L. Puri, Cambridge University Press)*, p.275-282.
- Lientz B.P. (1970). Results on nonparametric modal intervals. *SIAM J. Appl. Math.*, **19**:356-366.
- Wegman E.J. (1971). A note on the estimation of the mode. *Ann. Math. Statist.*, **42**(6):1909-1915.
- Yamato H. (1971). Sequential estimation of a continuous probability density function and mode. *Bull. Math. Statist.*, **14**:1-12.
- Ekblom H. (1972). A Monte Carlo investigation of mode estimators in small samples. *Applied Statistics*, **21**:177-184.
- Lientz B.P. (1972). Properties of modal intervals. *SIAM J. Appl. Math.*, **23**:1-5.
- Konakov V.D. (1973). On the asymptotic normality of the mode of multidimensional distributions. *Theory Probab. Appl.*, **18**:794-803.
- Robertson T. and Cryer J.D. (1974). An iterative procedure for estimating the mode. *J. Amer. Statist. Assoc.*, **69**(348):1012-1016.
- Kim B.K. and Van Ryzin J. (1975). Uniform consistency of a histogram density estimator and modal estimation. *Commun. Statist.*, **4**:303-315.
- Sager T.W. (1975). Consistency in nonparametric estimation of the mode. *Ann. Statist.*, **3**(3):698-706.
- Stone C.J. (1975). Adaptive maximum likelihood estimators of a location parameter. *Ann. Statist.*, **3**:267-284.
- Mizoguchi R. and Shimura M. (1976). Nonparametric Learning Without a Teacher Based on Mode Estimation. *IEEE Transactions on Computers*, **C25**(11):1109-1117.
- Adriano K.N., Gentle J.E. and Sposito V.A. (1977). On the asymptotic bias of Grenander's mode estimator. *Commun. Statist.-Theor. Meth. A*, **6**:773-776.

- Asselin de Beauville J.-P. (1978). Estimation non parametrique de la densite et du mode, exemple de la distribution Gamma. *Revue de Statistique Appliquee*, **26**(3):47-70.
- Sager T.W. (1978). Estimation of a multivariate mode. *Ann. Statist.*, **6**:802-812.
- Devroye L. (1979). Recursive estimation of the mode of a multivariate density. *Canadian J. Statist.*, **7**(2):159-167.
- Sager T.W. (1979). An iterative procedure for estimating a multivariate mode and isopleth. *J. Amer. Statist. Assoc.*, **74**(366):329-339.
- Eddy W.F. (1980). Optimum kernel estimators of the mode. *Ann. Statist.*, **8**(4):870-882.
- Eddy W.F. (1982). The Asymptotic Distributions of Kernel Estimators of the Mode. *Z. Wahrsch. Verw. Gebiete*, **59**:279-290.
- Hall P. (1982). Asymptotic Theory of Grenander's Mode Estimator. *Z. Wahrsch. Verw. Gebiete*, **60**:315-334.
- Sager T.W. (1983). Estimating modes and isopleths. *Commun. Statist.-Theor. Meth.*, **12**(5):529-557.
- Hartigan J.A. and Hartigan P.M. (1985). The Dip Test of Unimodality. *Ann. Statist.*, **13**:70-84.
- Hartigan P.M. (1985). Computation of the Dip Statistic to Test for Unimodality. *Appl. Statist. (JRSS C)*, **34**:320-325.
- Romano J.P. (1988). On weak convergence and optimality of kernel density estimates of the mode. *Ann. Statist.*, **16**(2):629-647.
- Tsybakov A. (1990). Recursive estimation of the mode of a multivariate distribution. *Probl. Inf. Transm.*, **26**:31-37.
- Hyndman R.J. (1996). Computing and graphing highest density regions. *Amer. Statist.*, **50**(2):120-126.
- Leclerc J. (1997). Comportement limite fort de deux estimateurs du mode : le shorth et l'estimateur naif. *C. R. Acad. Sci. Paris, Serie I*, **325**(11):1207-1210.
- Leclerc J. (2000). Strong limiting behavior of two estimates of the mode: the shorth and the naive estimator. *Statistics and Decisions*, **18**(4).
- Groeneboom P.and Wellner J.A. (2001). Computing Chernoff's distribution. *J. Comput. Graph. Statist.*, **10**:388-400.
- Shoung J.M. and Zhang C.H. (2001). Least squares estimators of the mode of a unimodal regression function. *Ann. Statist.*, **29**(3):648-665.
- Bickel D.R. (2002). Robust estimators of the mode and skewness of continuous data. *Computational Statistics and Data Analysis*, **39**:153-163.
- Abraham C., Biau G. and Cadre B. (2003). Simple Estimation of the Mode of a Multivariate Density. *Canad. J. Statist.*, **31**(1):23-34.
- Bickel D.R. (2003). Robust and efficient estimation of the mode of continuous data: The mode as a viable measure of central tendency. *J. Statist. Comput. Simul.*, **73**:899-912.
- Djeddour K., Mokkadem A. et Pelletier M. (2003). Sur l'estimation recursive du mode et de la valeur modale d'une densite de probabilite. *Technical report 105.*

- Djeddour K., Mokkadem A. et Pelletier M. (2003). Application du principe de moyennisation a l'estimation recursive du mode et de la valeur modale d'une densite de probabilite. *Technical report 106*.
- Hedges S.B. and Shah P. (2003). Comparison of mode estimation methods and application in molecular clock analysis. *BMC Bioinformatics*, **4**:31-41.
- Herrmann E. and Ziegler K. (2004). Rates of consistency for nonparametric estimation of the mode in absence of smoothness assumptions. *Statistics and Probability Letters*, **68**:359-368.
- Abraham C., Biau G. and Cadre B. (2004). On the Asymptotic Properties of a Simple Estimate of the Mode. *ESAIM Probab. Stat.*, **8**:1-11.
- Mokkadem A. et Pelletier M. (2005). Adaptive Estimation of the Mode of a Multivariate Density. *J. Nonparametr. Statist.*, **17**(1):83-105.
- Bickel D.R. et Fruehwirth R. (2006). On a Fast, Robust Estimator of the Mode: Comparisons to Other Robust Estimators with Applications. *Computational Statistics and Data Analysis*, **50**(12):3500-3530.

See Also

`mlv` for general mode estimation

`naive` *The Chernoff Mode Estimator*

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

```
naive(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 `naive` through `mlv(x, method = "naive", bw)`. This returns an object of class `mlv`.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Chernoff H. (1964). Estimation of the mode. *Ann. Inst. Statist. Math.*, **16**:31-41.
- Leclerc J. (1997). Comportement limite fort de deux estimateurs du mode : le shorth et l'estimateur naif. *C. R. Acad. Sci. Paris, Serie I*, **325**(11):1207-1210.
- Leclerc J. (2000). Strong limiting behavior of two estimates of the mode: the shorth and the naive estimator. *Statistics and Decisions*,**18**(4).

See Also

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

Examples

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

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

## Estimate of the mode
mean(naive(x, bw = 1/4))
M <- mlv(x, method = "naive", bw = 1/4)
print(M)
plot(M, xlim = c(0,2))
```

`parzen` *Parzen's Kernel Mode Estimator*

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.default` 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", ...)`. This returns an object of class `mlv`.

Presently, `parzen` is quite slow.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Parzen E. (1962). On estimation of a probability density function and mode. *Ann. Math. Stat.*, **33**(3):1065–1076.
- Konakov V.D. (1973). On the asymptotic normality of the mode of multidimensional distributions. *Theory Probab. Appl.*, **18**:794-803.
- Eddy W.F. (1980). Optimum kernel estimators of the mode. *Ann. Statist.*, **8**(4):870-882.
- Eddy W.F. (1982). The Asymptotic Distributions of Kernel Estimators of the Mode. *Z. Wahrsch. Verw. Gebiete*, **59**:279-290.
- Romano J.P. (1988). On weak convergence and optimality of kernel density estimates of the mode. *Ann. Statist.*, **16**(2):629-647.

- Abraham C., Biau G. and Cadre B. (2003). Simple Estimation of the Mode of a Multivariate Density. *Canad. J. Statist.*, **31**(1):23-34.
- Abraham C., Biau G. and Cadre B. (2004). On the Asymptotic Properties of a Simple Estimate of the Mode. *ESAIM Probab. Stat.*, **8**:1-11.

See Also

mlv, naive

Examples

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

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

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

---

skewness *Skewness*

Description

The `skewness.default` function from package **fBasics** is completed in order to implement Bickel's measure of skewness, based on the mode of the distribution considered.

Usage

```
skewness(x,
          ...)
## Default S3 method:
skewness(x,
          na.rm = FALSE,
          method = c("moment", "fisher", "bickel"),
          M = shorth(x),
          ...)
```

Arguments

`x`	numeric. Vector of observations.
`na.rm`	logical. Should missing values be removed?

`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.
`M`	numeric. (An estimate of) the mode of the observations `x`. Default value is `shorth(x)`.
`...`	arguments to be passed.

Value

`skewness` returns a numeric value. An attribute which reports the method used is added.

Author(s)

Diethelm Wuertz and other authors for the original `skewness` function from package **fBasics**; Paul Poncet ⟨paulponcet@yahoo.fr⟩ for the slight modification introduced.

References

- Bickel D.R. (2002). Robust estimators of the mode and skewness of continuous data. *Computational Statistics and Data Analysis*, **39**:153-163.
- Bickel D.R. et Fruehwirth R. (2006). On a Fast, Robust Estimator of the Mode: Comparisons to Other Robust Estimators with Applications. *Computational Statistics and Data Analysis*, **50**(12):3500-3530.

See Also

`015A-BasicStatistics` from package **fBasics**; `mlv` for general mode estimation; `shorth` for the shorth estimate of the mode;

Examples

```
## 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))
```

tsybakov	*The Tsybakov Mode Estimator*

Description

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

Usage

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

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 are under investigation.

Note

The user should preferentially call `tsybakov` through `mlv(x, method = "tsybakov", ...)`. This returns an object of class `mlv`.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Mizoguchi R. and Shimura M. (1976). Nonparametric Learning Without a Teacher Based on Mode Estimation. *IEEE Transactions on Computers*, **C25**(11):1109-1117.
- Tsybakov A. (1990). Recursive estimation of the mode of a multivariate distribution. *Probl. Inf. Transm.*, **26**:31-37.
- Djeddour K., Mokkadem A. et Pelletier M. (2003). Sur l'estimation recursive du mode et de la valeur modale d'une densite de probabilite. *Technical report 105*.
- Djeddour K., Mokkadem A. et Pelletier M. (2003). Application du principe de moyennisation a l'estimation recursive du mode et de la valeur modale d'une densite de probabilite. *Technical report 106*.

See Also

`mlv` for general mode estimation

Examples

```
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)
M <- mlv(x, method = "tsybakov", kernel = "gaussian", alpha = 0.99)
print(M)
plot(M)
```

---

venter — *The Venter / Dalenius / LMS Mode Estimator*

Description

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

Usage

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

shorth(x,
       ...)
```

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 `bw` or the argument `k`, `k` being taken equal to `ceiling(bw*ny) - 1` if missing.
If `type = 1`, the midpoint of the modal interval is returned. If `type = 2`, the `floor((k+1)/2)`th element of the modal interval is returned. If `type = 3` or `type = "dalenius"`, the median of the modal interval is returned. If `type = 4` or `type = "shorth"`, the mean of the modal interval is returned. If `type = 5` or `type = "ekblom"`, Ekblom's $L_{-\infty}$ estimate is returned, see Ekblom (1972). If `type = 6` or `type = "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 `mlv(x, method = "venter", ...)`. This returns an object of class `mlv`.

Author(s)

Paul Poncet ⟨paulponcet@yahoo.fr⟩

References

- Dalenius T. (1965). The Mode - A Negleted Statistical Parameter. *J. Royal Statist. Soc. A*, *128*:110-117.
- Venter J.H. (1967). On estimation of the mode. *Ann. Math. Statist.*, **38**(5):1446-1455.
- Ekblom H. (1972). A Monte Carlo investigation of mode estimators in small samples. *Applied Statistics*, **21**:177-184.
- Leclerc J. (1997). Comportement limite fort de deux estimateurs du mode : le shorth et l'estimateur naif. *C. R. Acad. Sci. Paris, Serie I*, **325**(11):1207-1210.
- Leclerc J. (2000). Strong limiting behavior of two estimates of the mode: the shorth and the naive estimator. *Statistics and Decisions*, **18**(4).

See Also

`mlv` for general mode estimation, `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)
M <- mlv(x, method = "venter", bw = 1/3)
print(M)
plot(M, xlim = c(20, 30))
```