Package ‘multimode’

November 13, 2018

Encoding UTF-8

Type Package

Title Mode Testing and Exploring

Version 1.4

Date 2018-11-12

Author Jose Ameijeiras-Alonso [aut,cre],
Rosa M. Crujeiras [aut],
Alberto Rodríguez-Casal [aut],
The R Core Team 1996-2012 [ctb,cph] (C function 'BinDist2' obtained from package 'stats'),
The R Foundation 2005 [ctb,cph] (C function 'BinDist2' obtained from package 'stats')

Maintainer Jose Ameijeiras-Alonso <jose.ameijeirasalonso@kuleuven.be>

Description
Different examples and methods for testing (including different proposals described in Ameijeiras-Alonso et al., 2018 <DOI:10.1007/s11749-018-0611-5>) and exploring (including the mode tree, mode forest and SiZer) the number of modes using nonparametric techniques.

Depends R (>= 2.10.0)

License GPL-3

NeedsCompilation yes

LazyData TRUE

Imports diptest, ks, rootSolve

Suggests nor1mix

URL https://arxiv.org/abs/1803.00472/

Repository CRAN

Date/Publication 2018-11-13 10:50:02 UTC

R topics documented:

`multimode-package` *Mode testing and exploring.*

Description

Different examples and methods for testing (including different proposals described in Ameijeiras-Alonso et al., 2018 <DOI:10.1007/s11749-018-0611-5>) and exploring (including the mode tree, mode forest and SiZer map) the number of modes using nonparametric techniques.

Details

Package:	multimode
Type:	Package
Version:	1.4
Date:	2018-11-12
License:	GPL-3
NeedsCompilation:	yes
LazyData:	yes

This package incorporates the function `modetest` which tests if the number of modes of a sample is equal to a given number (against if it is greater). Functions `bw.crit` and `excessmass` provide the critical bandwidth and the excess mass statistic, respectively. Function `nmodes` computes the number of modes for a given bandwidth. Given a certain number of modes, function `locmodes` provides the estimation of the locations of modes and antimodes and their density value. Functions `modetree` and `modeforest` provide the mode tree and forest, respectively; they represent the estimated mode locations for different bandwidths. Function `sizer` can be used for determining where the smoothed curve is significantly increasing or decreasing. Registries with missing data are removed. For a complete list of functions, use `library(help="multimode")`.

Acknowledgements

This work has been supported by Projects MTM2016–76969–P (Spanish State Research Agency, AEI) and MTM2013–41383–P (Spanish Ministry of Economy, Industry and Competitiveness), both co–funded by the European Regional Development Fund (ERDF), IAP network (Developing crucial Statistical methods for Understanding major complex Dynamic Systems in natural, biomedical and social sciences, StUDyS) from Belgian Science Policy. Work of J. Ameijeiras-Alonso has been supported by the PhD grant BES-2014-071006 from the Spanish Ministry of Economy and Competitiveness.

References

Jose Ameijeiras–Alonso, Rosa M. Crujeiras, Alberto Rodríguez–Casal (2018a). Mode testing, critical bandwidth and excess mass, to appear in *Test*.

Ameijeiras–Alonso, J., Crujeiras, R.M. and Rodríguez–Casal, A. (2018b). multimode: An R Package for Mode Assessment, *arXiv preprint: 1803.00472*.

Chaudhuri, P. and Marron, J. S. (1999). SiZer for exploration of structures in curves, *Journal of the American Statistical Association*, **94**, 807–823.

Minnotte, M. C., Marchette, D. J. and Wegman, E. J. (1998). The bumpy road to the mode forest, *Journal of Computational and Graphical Statistics*, **7**, 239–251.

`acidity` *Acid–neutralizing capacity*

Description

This dataset, analyzed by Crawford (1994), contains the Acid–Neutralizing Capacity (ANC) measured in a sample of 155 lakes in North–Central Wisconsin (USA). ANC describes the capability of a lake to absorb acid, where low ANC values may lead to a loss of biological resources.

Usage

```
data(acidity)
```

Format

`acidity` includes an acidity index of lakes in north–central Wisconsin on the log scale, in particular, it is provided the log(ANC+50) as in Crawford (1994).

Details

This is a classic example for determining the number of modes.

Source

Obtained from the Supplementary material of Richardson and Green (1997), available in `http://www.stats.bris.ac.uk/~peter/mixdata`.

References

Crawford (1994). An application of the Laplace method to finite mixture distributions. *Journal of the American Statistical Association*, **89**, 259–267.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B*, **59**, 731–792.

Examples

```
data("acidity")
# Kernel density estimation with two modes and SiZer
locmodes(acidity,mod0=2,display=TRUE,xlab="log(ANC+50)")
sizer(acidity,bws=c(0.1,1),xlab="log(ANC+50)")
```

bw.crit *Critical bandwidth*

Description

This function computes the critical bandwidth for a specified number of modes.

Usage

```
bw.crit(data,mod0=1,lowsup=-Inf,uppsup=Inf,n=2^15,tol=10^(-5),full.result=F)
```

Arguments

`data` Sample for computing the critical bandwidth.

`mod0` Number of modes for which the critical bandwidth is calculated. Default `mod0=1`.

`lowsup` Lower limit for the random variable support in the computation of the critical bandwidth. Default is `-Inf`.

`uppsup` Upper limit for the random variable support in the computation of the critical bandwidth. Default is `Inf`.

`n` The number of equally spaced points at which the density is estimated. When n > 512, it is rounded up to a power of 2 as in the `density` function. Default `n=2^15`.

`tol` Accuracy requested in the computation of the critical bandwidth. Default value of `tol` is `10^(-5)`.

`full.result` If this argument is TRUE then it returns the full result list, see below. Default `full.result=FALSE`.

Details

With `bw.crit` the critical bandwidth for the number of modes specified in `mod0` is calculated, e.g., the smallest bandwidth such that the kernel density estimator has at most `mod0` modes. If the compact support is unknown, the critical bandwidth introduced by Silverman (1981) is computed and if it is provided that one of Hall and York (2001) is calculated.

Since a dichotomy method is employed for computing the critical bandwidth, the parameter `tol` is used to determine a stopping time in such a way that the error committed in the computation of the critical bandwidth is less than `tol`.

The NAs will be automatically removed.

Value

Depending on `full.result` either a number, the critical bandwidth of the sample for `mod0` modes, or an object of class `"estmod"` which is a `list` containing the following components:

`nmodes` The specified hypothesized value of the number of modes.

`sample.size` The number of non-missing observations in the sample used for computing the number of modes.

`bw` Value of the critical bandwidth test statistic.

`lowsup` Lower limit of the support where the critical bandwidth is computed.

`ippsup` Upper limit of the support where the critical bandwidth is computed.

`fnx` The `n` coordinates of the points where the density is estimated for computing the critical bandwidth.

`fny` The estimated density values.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Hall, P. and York, M. (2001). On the calibration of Silverman's test for multimodality, *Statistica Sinica*, **11**, 515–536.

Silverman, B. W. (1981). Using kernel density estimates to investigate multimodality, *Journal of the Royal Statistical Society. Series B*, **43**, 97–99.

Examples

```
# Critical bandwidth of Silverman (1981) for one mode.
set.seed(2016)
data=rnorm(50)
bw.crit(data)
# Critical bandwidth of Hall and York for two modes in the interval (-1.5,1.5).
set.seed(2016)
data=rnorm(50)
bw.crit(data,mod0=2,lowsup=-1.5,uppsup=1.5)
```

chondrite *Percentage of silica in chondrite meteors*

Description

This dataset contains the percentage silica (in %) in 22 chondrite meteors.

Usage

```
data(chondrite)
```

Format

`chondrite` and `chondritegg` include the data provided in Good and Gaskins (1980). In `chondrite`, the typo in the 9th observation was corrected while `chondritegg` includes the original dataset. In `chondritel` is included the scaled data used by Leonard (1978).

Details

This is a classic example for determining the number of modes.

Source

Obtained from the Table 2 of Good and Gaskins (1980).

References

Good, I. J. and Gaskins, R. A. (1980). Density estimation and bump-hunting by the penalized likelihood method exemplified by scattering and meteorite data. *Journal of the American Statistical Association*, **75**, 42–56.

Leonard, T. (1978). Density estimation, stochastic processes and prior information. *Journal of the Royal Statistical Society. Series B (Methodological)*, **40**, 113–146.

Examples

```
data("chondrite")
# SiZer between the critical bandwidths for one and six modes
sizer(chondrite,cbw1=1,cbw2=6)
```

`enzyme` *Blood enzymatic activity*

Description

This dataset concerns the distribution of enzymatic activity in the blood, for an enzyme involved in the metabolism of carcinogenic substances.

Usage

```
data(enzyme)
```

Format

`enzyme` includes the values of the urinary metabolic ratio of 5–acetylamino–6–formylamino–3–methyluracil to 1–methylxanthine (AFMU/1X) after oral administration of caffeine.

Details

This is a classic example for determining the number of modes.

Source

Obtained from the Supplementary material of Richardson and Green (1997), available in `http://www.stats.bris.ac.uk/~peter/mixdata`.

References

Bechtel, Y. C., Bonaiti–Pellie, C., Poisson, N., Magnette, J. and Bechtel, P. R. (1993). A population and family study N–acetyltransferase using caffeine urinary metabolites. *Clinical Pharmacology and Therapeutics*, **54**, 134–141.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B*, **59**, 731–792.

Examples

```
data("enzyme")
# It can be seen that there are two groups in this dataset
# Via exploratory tools
sizer(enzyme,bws=c(0.03,1))
modetree(enzyme,bws=c(0.02,1),logbw=TRUE,addplot=TRUE,col.lines="white")
#Via mode testing
modetest(enzyme)
## Not run:
modetest(enzyme,mod0=2)

## End(Not run)
```

excessmass *Excess mass*

Description

This function computes the excess mass statistic.

Usage

```
excessmass(data,mod0=1,approximate=FALSE,gridsize=NULL,full.result=F)
```

Arguments

`data`	Sample for computing the excess mass.
`mod0`	Number of modes for which the excess mass is calculated. Default `mod0=1`.
`approximate`	If this argument is TRUE then the excess mass value is approximated. Default `approximate=FALSE`.
`gridsize`	When `approximate=TRUE`, number of endpoints at which the $C_m(\lambda)$ sets are estimated (first element) and number of possible values of λ (second element). Default is `gridsize=c(20,20)`.
`full.result`	If this argument is TRUE then it returns the full result list, see below. Default `full.result=FALSE`.

Details

With `excessmass`, the excess mass test statistic, introduced by Müller and Sawitzki (1991), for the integer number of modes specified in `mod0` is computed.

The excess mass test statistic for k modes is defined as $\max_{\lambda}\{D_{n,k+1}(\lambda)\}$, where $D_{n,k+1}(\lambda) = (E_{n,k+1}(P_n,\lambda) - E_{n,k}(P_n,\lambda))$. The empirical excess mass function for k modes is defined as $E_{n,k}(P_n,\lambda) = \sup_{C_1(\lambda),\ldots,C_k(\lambda)}\{\sum_{m=1}^{k} P_n(C_m(\lambda)) - \lambda||C_m(\lambda)||\}$, being the sets $C_m(\lambda)$ closed intervals with endpoints the data points.

When `mod0>1` and the sample size is *large*, a two–steps approximation (`approximate=TRUE`) can be performed in order to improve the computing time efficiency. First, since the possible λ candidates to maximize $D_{n,k+1}(\lambda)$ can be directly obtained from the sets that maximize $E_{n,k+1}$ and $E_{n,k}$ (see Appendix E in Ameijeiras–Alonso *et al.*, 2016), the possible values of λ are computed by looking to the empirical excess mass function in `gridsize[1]` endpoints candidates for $C_m(\lambda)$ and also in the λ values associated to the empirical excess mass for one mode. Once a λ maximizing the approximated values of $D_{n,k+1}(\lambda)$ is chosen, in order to obtain the approximation of the excess mass test statistic, in its neighborhood, a grid of possible values of λ is created, being its length equal to `gridsize[2]`, and the exact value of $D_{n,k+1}(\lambda)$ is calculated for these values of λ (using the algorithm proposed by Müller and Sawitzki, 1991).

If there are repeated data in the sample or the distance between different pairs of data points shows ties, a data perturbation is applied. This modification is made in order to avoid the discretization of the data which has important effects on the computation of the test statistic. The perturbed sample is obtained by adding a sample from the uniform distribution in minus/plus a half of the minimum of the positive distances between two sample points.

The NAs will be automatically removed.

Value

Depending on `full.result` either a number, the excess mass statistic for `mod0` modes, or an object of class `"estmod"` which is a `list` containing the following components:

`nmodes` The specified hypothesized value of the number of modes.

`sample.size` The number of non-missing observations in the sample used for computing the excess mass.

`excess.mass` Value of the excess mass test statistic.

`approximate` A logical value indicating if the excess mass was approximated.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Jose Ameijeiras–Alonso, Rosa M. Crujeiras, Alberto Rodríguez–Casal (2016). Mode testing, critical bandwidth and excess mass, *arXiv preprint: 1609.05188*.

Müller, D. W. and Sawitzki, G. (1991). Excess mass estimates and tests for multimodality, *The Annals of Statistics*, **13**, 70–84.

Examples

```
# Excess mass statistic for one mode
set.seed(2016)
data=rnorm(50)
excessmass(data)
```

`galaxy` *Velocities of galaxies diverging away from our own galaxy*

Description

This dataset contains the velocities in km/sec of different galaxies from the unfilled survey of the Corona Borealis region.

Usage

```
data(galaxy)
```

Format

galaxy includes the original measures in Roeder (1990). galaxyrg includes the data provided in the Supplementary material of Richardson and Green (1997), where the velocities are divided by 1000 and the 78th observation was replaced by 26690 km/sec. galaxyp add a measurement of 5607 km/sec included in Postman *et al.* (1986).

Details

This is a classic example for determining the number of modes.

Source

Obtained from the Table 1 of Postman *et al.* (1986), Table 1 of Roeder (1990) and the Supplementary material of Richardson and Green (1997), available in http://www.stats.bris.ac.uk/~peter/mixdata.

References

Postman, M., Huchra, J. P. and Geller, M. J. (1986). Probes of large–scale structures in the Corona Borealis region. *Astronomical Journal*, **92**, 1238–1247.

Richardson, S. and Green, P. J. (1997). On Bayesian analysis of mixtures with unknown number of components (with discussion). *Journal of the Royal Statistical Society, Series B*, **59**, 731–792.

Roeder, K. (1990). Density estimation with confidence sets exemplified by superclusters and voids in the galaxies. *Journal of the American Statistical Association*, **85**, 617–624.

Examples

```
data("galaxy")
# Mode tree between the critical bandwidths for one and six modes
modetree(galaxy,cbw1=1,cbw2=6)
```

geyser *Waiting time between geyser eruptions*

Description

This dataset contains the interval times between the starts of the geyser eruptions on the Old Faithful Geyser in Yellowstone National Park, Wyoming, USA.

Usage

```
data(geyser)
data(geyserab)
data(geyserh)
data(geyserw)
```

Format

geyser, geyserh and geyserw include data collected in October 1980. geyser and geyserh were obtained from Table 3 of Härdle (2012), in geyser some repeated data are removed. geyserw was obtained from the Supplementary material of Weisberg (2005). geyserab include data collected in August 1985 from Table 1 in Azzalini and Bowman (1990).

Details

This is a classic example for estimating the density.

Source

Obtained from the original tables in Azzalini and Bowman (1990), Härdle (2012) and Supplementary material of Weisberg (2005).

References

Azzalini, A. and Bowman, A. W. (1990). A look at some data on the Old Faithful geyser. *Applied Statistics*, **39**, 357-365.

Härdle, W. (1991). *Smoothing techniques: with implementation in S*. New York: Springer-Verlag.

Weisberg, S. (2005). *Applied Linear Regression*. New York: Wiley.

Examples

```
data("geyser")
# Kernel density estimation with two modes
locmodes(geyser,mod0=2,display=TRUE)
```

---

locmodes *Location of modes and antimodes*

Description

Given a certain number of modes, this function provides the estimation of the location of modes and antimodes and their density value.

Usage

```
locmodes(data,mod0=1,lowsup=-Inf,uppsup=Inf,n=2^15,tol=10^(-5),display=F,...)

## S3 method for class 'locmod'
plot(x,addplot=NULL,xlab=NULL,ylab=NULL,addLegend=NULL,posLegend=NULL,...)

## S3 method for class 'locmod'
print(x,digits=getOption("digits"), ...)
```

Arguments

`data` Sample in which the critical bandwidth is computed.

`mod0` Number of modes for which the critical bandwidth is calculated. Default `mod0=1`.

`lowsup` Lower limit for the random variable support in the computation of the critical bandwidth. Default is `-Inf`.

`uppsup` Upper limit for the random variable support in the computation of the critical bandwidth. Default is `Inf`.

`n` The number of equally spaced points at which the density is to be estimated. When `n` > 512, it is rounded up to a power of 2 as in the `density` function. Default `n=2^15`.

`tol` Accuracy requested in the computation of the critical bandwidth. Default value `tol=10^(-5)`.

`display` Logical, if `TRUE` the kernel density estimation is plotted adding the estimated location of the modes and the antimodes. Default is `FALSE`.

`...` Arguments to be passed to subsequent methods, `plot.default` for the `plot` method and `formatC` for the `print` method.

`x` An object inheriting from class `"locmod"`.

`addplot` Logical, if `TRUE` the plot is added to the current one. Default is `FALSE`.

`xlab` A title for the x axis. See `title`.

`ylab` A title for the y axis. See `title`.

`addLegend` Logical, if `TRUE` the legend is added in the plot. Default is `TRUE`.

`posLegend` The vector of two elements of coordinates to be used to position the legend. It can be specified by keyword as in the function `legend`. Default is `"topright"`.

`digits` Number of significant digits to use, see `formatC`.

Details

Given a certain number of modes, `mod0`, with `locmodes` the estimation of the location of modes and antimodes, their density value and the corresponding critical bandwidth is provided. To obtain these estimates, the kernel density estimation with gaussian kernel and the critical bandwidth for `mod0` modes is employed. If the compact support is unknown, the critical bandwidth of Silverman (1981) is computed and if such a support is provided, then the one proposed by Hall and York (2001) is calculated. Note that when the support is unknown the critical bandwidth may create artificial modes in the tails.

Since a dichotomy method is employed for computing the critical bandwidth, the parameter `tol` is used to determine a stopping time in such a way that the error committed in the computation of the critical bandwidth is less than `tol`.

If `display=TRUE`, then the kernel density estimation using the critical bandwidth for `mod0` modes is plotted. Additionally, the estimated location of modes (dashed lines), antimodes (point lines) and support (solid lines) can be also plotted. If `addLegend=TRUE`, a legend (in the position `posLegend`) with this information is included.

The NAs will be automatically removed.

Value

A list with class `"locmod"` containing the following components:

`locations` Vector with the estimated locations of modes (odd positions of the vector) and antimodes (even positions).

`fvalue` Vector with estimated density values at modes (odd positions of the vector) and antimodes (even positions).

`cbw` A list with class `"estmod"` which contains the critical bandwidth of the sample for `mod0` modes, see `bw.crit`.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Jose Ameijeiras–Alonso, Rosa M. Crujeiras, Alberto Rodríguez–Casal (2018). Mode testing, critical bandwidth and excess mass, to appear in *Test*.

Hall, P. and York, M. (2001). On the calibration of Silverman's test for multimodality, *Statistica Sinica*, **11**, 515–536.

Silverman, B. W. (1981). Using kernel density estimates to investigate multimodality, *Journal of the Royal Statistical Society. Series B*, **43**, 97–99.

Examples

```
# Testing for unimodality
set.seed(2016)
data=rnorm(50)
modetest(data)
#There is no evidence to reject the null hypothesis of unimodality
#Estimated location of the mode and its density value
locmodes(data)
## Not run:
#Estimated locations of the five modes in the claw of Marron and Wand (1992)
library(nor1mix)
set.seed(2016)
n<-200
data<-nor1mix::rnorMix(n,MW.nm10)
#Adding the plot of the estimated locations
locmodes(data,5,display=T)
#Assuming that the compact support is [-1.5,1.5]
locmodes(data,5,-1.5,1.5,display=T)

## End(Not run)
```

modeforest *Mode forest*

Description

This function provides the mode forest.

Usage

```
modeforest(data,bws=NULL,gridsize=NULL,B=99,n=512,cbw1=NULL,cbw2=NULL,
display=TRUE,logbw=FALSE,from=NULL,to=NULL,logbw.regulargrid=NULL,...)
```

Arguments

`data` Sample in which the mode forest is computed.

`bws` Vector or range of bandwidths. If it is a vector of size two, then it is used a grid of bandwidths between the given values. Default lower bandwidth is twice the distance between the grid points used for estimating the density and upper bandwidth equal to the range of the data. Unless it is specified a vector of size greater than two, the number of bandwidths employed is equal to the second element of `gridsize`.

`gridsize` Number of grid points in the horizontal (values of the variable, first element) and vertical (bandwidths, second element) axis. Default is `c(100,151)`.

`B` Number of replicates used for generating the mode forest. Default `B=99`.

`n` The number of equally spaced points at which the density is to be estimated. When `n` > 512, it is rounded up to a power of 2 as in the `density` function. Default `n=512`.

`cbw1` Number of modes for which the first critical bandwidth is calculated. This is the first bandwidth used to compute the mode tree when `bws` is not specified.

`cbw2` Number of modes for which the second critical bandwidth is calculated. This is the last bandwidth used to compute the mode tree when `bws` is not specified.

`display` Logical, if `TRUE` the mode tree plot is plotted. Default `TRUE`.

`logbw` Logical, if `TRUE` the plot displays and returns the log10 bandwidths. Default `logbw=FALSE`.

`from` First plotted value of the variable. Default is below the data minimum.

`to` Last plotted value of the variable. Default is above the data maximum.

`logbw.regulargrid`
Logical, if `TRUE` a regular grid of bandwidths is created over the log10 scale. Default `logbw.regulargrid=FALSE`.

`...` Arguments to be passed to subsequent methods, `plot.gtmod` for the `plot`, `print` and `summary` methods.

Details

The mode forest for the sample given in `data` is computed. For this calculation, a kernel density estimator with Gaussian kernel and bandwidths `bws` is used. The mode forest is generated by looking simultaneously at a collection of mode trees generated by the original sample and `B` random resamples drawn with replacement from `data`. When the mode forest is plotted, this tool represents the number of times an estimated mode falls in each location-bandwidth (horizontal-vertical axis) pixel. The pixels are then shaded proportionally to counts (large counts corresponding to darker pixels and low counts to lighter ones).

The NAs will be automatically removed.

Value

A list with class `"gtmod"` containing the following components:

`modeforest` Matrix including the percentage of times that a mode tree falls in each location-bandwidth pixel.

`range.x` Employed location values to represent the mode forest.

`range.bws` Employed bandwidths to compute the different mode trees.

`logbw` Logical value indicating if the bandwidths are given in the log10 scale.

`sample.size` The number of non-missing observations in the sample used for obtaining the mode forest.

`call` The unevaluated expression, which consists of the named function applied to the given arguments.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Minnotte, M. C., Marchette, D. J. and Wegman, E. J. (1998). The bumpy road to the mode forest, *Journal of Computational and Graphical Statistics*, **7**, 239–251.

Examples

```
#Mode forest using a grid of bandwidths between 0.2 and 0.5 and 29 bootstrap replicas
set.seed(2016)
data=rnorm(50)
modeforest(data,bws=c(0.2,0.5),B=29)
#Original mode tree for this sample
modetree(data,bws=c(0.2,0.5),addplot=TRUE,col.lines="red")
```

modetest *Test for the number of modes*

Description

This function tests the number of modes.

Usage

```
modetest(data,mod0=1,method="ACR",B=500,lowsup=-Inf,uppsup=Inf,
submethod=NULL,n=NULL,tol=NULL,tol2=NULL,gridsize=NULL,alpha=NULL,
nMC=NULL,BMC=NULL)
```

Arguments

`data` Sample to be tested.

`mod0` The maximum number of modes in the null hypothesis. Default `mod0=1` (unimodality vs. multimodality test).

`method` The method employed for testing the number of modes. Available methods are: SI (Silverman, 1981), HY (Hall and York, 2001), FM (Fisher and Marron, 2001), HH (Hartigan and Hartigan, 1985), CH (Cheng and Hall,1998), ACR (Ameijeiras–Alonso *et al.*, 2018). Default `method="ACR"`.

`B` Number of replicates used in the test. Default `B=500`.

`lowsup` Lower limit for the random variable support in the computation of the critical bandwidth. Default is `-Inf`.

`uppsup` Upper limit for the random variable support in the computation of the critical bandwidth. Default is `Inf`.

`submethod` Different approaches when using `method` SI, HY or ACR. Available submethods are: 1 and 2, see Details below for more information. Default `submethod=1`, except when `mod0>1`, `method` is ACR and the sample size is greater than 200.

`n` The number of equally spaced points at which the density is to be estimated. When n > 512, it is rounded up to a power of 2 as for `density` function. Default is `n=2^10` when `method` is SI, HY or FM and `n=2^15` when the `method` is CH or ACR. Argument not used for other methods.

`tol` Accuracy for computing the critical bandwidth. Default `tol=10^(-5)`.

`tol2` Accuracy for integration of the calibration function in the `method` ACR when the support is known. Default `tol2=10^(-5)`.

`gridsize` When the approximated version of the excess mass is employed in `method` ACR, number of endpoints at which the $C_m(\lambda)$ sets are estimated (first element) and number of possible values of λ (second element). Default is `gridsize=c(20,20)`.

`alpha` Significance level employed for testing unimodality when method 1 of Hall and York (2001) is used. Default `alpha=0.05`.

`nMC` Number of Monte Carlo replicates used to approximate the p-value in the method 2 of Hall and York (2001). Default `nMC=100`.

BMC Number of bootstrap replicas used for computing the p–value in each Monte Carlo replicate of the Hall and York (2001) method 2. Default BMC=100.

Details

The number of modes for the underlying density of a sample given by data can be tested with modetest. The null hypothesis states that the sample has mod0 modes, and the alternative hypothesis is if it has more modes. The test used for calculating the p-value is specified in method. All the available proposals require bootstrap or Monte Carlo resamples (number specified in B).

Except when the support is employed, the typical usages are

```
modetest(data,...)
modetest(data,mod0=1,method="ACR",B=500,...)
```

Since a dichotomy algorithm is employed for computing the critical bandwidth (methods SI, HY, FM, ACR), the parameter tol is used to determine a stopping time in such a way that the error committed in the computation of the critical bandwidth is less than tol.

The sample data can be perturbed in the methods using the excess mass or the dip statistic (HH, CH and ACR) in order to avoid important effects on the computation of the test statistic. In general, the exact excess mass/dip value is employed, but also its approximated version can be used by setting submethod=2 in method ACR. See excessmass Details for more information.

Typical usages are

```
modetest(data,mod0=1,method="ACR",B=500,submethod=1,n=NULL,tol=NULL)
modetest(data,mod0=1,method="ACR",B=500,submethod=2,n=NULL,tol=NULL,
gridsize=NULL)
```

When employing SI method, two ways of computing the resamples are available. If submethod=1, the resamples are generated from the rescaled bootstrap resamples as proposed by Silverman (1981). If submethod=2, as in method HY, the resamples are generated from the distribution that is associated to the kernel density estimation with the critical bandwidth.

Typical usage is

```
modetest(data,mod0=1,method="SI",B=500,submethod=NULL,n=NULL,tol=NULL)
```

If a compact support containing a mode is known, it can be used to compute the Hall and York (2001) critical bandwidth. Note that in the case of the test proposed by Hall and York (2001), this support should be known, unless the support of the density function is bounded. For their proposal, two methods are implemented. submethod 1 is an asymptotic correction of Silverman (1981) test based on the limiting distribution of the test statistic. When submethod=1, the significance level must be previously determined with alpha. submethod 2 is based on Monte Carlo techniques. For this reason, when submethod=2, the number of replicates (nMC) and the number of bootstrap replicates (BMC) used for computing the p–value in each Monte Carlo replicate are needed.

Typical usages are

```
modetest(data,method="HY",lowsup=-1.5,uppsup=1.5,...)
modetest(data,method="HY",B=500,lowsup=-1.5,uppsup=1.5,submethod=1,
n=NULL,tol=NULL,alpha=NULL)
modetest(data,method="HY",B=500,lowsup=-1.5,uppsup=1.5,submethod=2,
n=NULL,tol=NULL,nMC=NULL,BMC=NULL)
```

A modification of the proposal of Ameijeiras–Alonso *et al.* (2018) can be also applied, by setting `method=ACR` and including a known compact support for detecting the modes. The parameter `tol2` is the accuracy required in the integration of the calibration function. For more information, see Ameijeiras–Alonso *et al.* (2018), the default approach, when the support is unknown, is given in Section 2.3 and, when it is provided, the approach shown in the Appendix B is employed.

Typical usage is

```
modetest(data,mod0=1,method="ACR",B=500,lowsup=-1.5,uppsup=1.5,
submethod=NULL,n=NULL,tol=NULL,tol2=NULL,gridsize=NULL)
```

The NAs will be automatically removed.

Value

A list with class `"htest"` containing the following components:

`p.value` P-value obtained after applying the test.

`statistic` Value of the test statistic. Critical bandwidth if the `method` is SI or HY; Cramer-von Mises statistic if the `method` is FM; the dip statistic if the `method` is HH, and the excess mass when the `method` is CH or ACR.

`null.value` The specified hypothesized value of the number of modes.

`alternative` A character string describing the alternative hypothesis, which is always `"greater"`.

`method` A character string indicating what type of multimodality test was performed.

`sample.size` The number of non-missing observations in the sample used for the hypothesis test.

`data.name` A character string giving the name of the data.

`bad.obs` The number of missing values that were removed from the data object prior to performing the hypothesis test.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Jose Ameijeiras–Alonso, Rosa M. Crujeiras, Alberto Rodríguez–Casal (2018). Mode testing, critical bandwidth and excess mass, to appear in *Test*.

Cheng, M. Y. and Hall, P. (1998). Calibrating the excess mass and dip tests of modality, *Journal of the Royal Statistical Society. Series B*, **60**, 579–589.

Fisher, N.I. and Marron, J. S. (2001). Mode testing via the excess mass estimate, *Biometrika*, **88**, 419–517.

Hall, P. and York, M. (2001). On the calibration of Silverman's test for multimodality, *Statistica Sinica*, **11**, 515–536.

Hartigan, J. A. and Hartigan, P. M. (1985). The Dip Test of Unimodality, *Journal of the American Statistical Association*, **86**, 738–746.

Silverman, B. W. (1981). Using kernel density estimates to investigate multimodality, *Journal of the Royal Statistical Society. Series B*, **43**, 97–99.

Examples

```
# Testing for unimodality
data(geyser)
data=geyser
modetest(data)
## Not run:
# Testing bimodality using B=100 bootstrap replicas
modetest(data,mod0=2,B=100)
#There is no evidence to reject the null hypothesis of bimodality
locmodes(data,mod0=2,display=TRUE)

## End(Not run)
```

`modetree` *Mode tree*

Description

This function provides the mode tree.

Usage

```
modetree(data,bws=NULL,gridsize=NULL,cbw1=NULL,cbw2=NULL,display=TRUE,
logbw=FALSE,logbw.regulargrid=NULL,...)
```

Arguments

`data` Sample in which the mode tree is computed.

`bws` Vector or range of bandwidths. If it is a vector of size two, then it is used a grid of bandwidths between the given values. Default lower bandwidth is twice the distance between the grid points used for estimating the density and upper bandwidth equal to the range of the data. Unless it is specified a vector of size greater than two, the number of bandwidths employed is equal to the second element of `gridsize`.

`gridsize` Number of equally spaced points at which the density is to be estimated (first element) and bandwidths used to compute the mode tree (second element). Default is `gridsize=c(512,151)`.

`cbw1` Number of modes for which the first critical bandwidth is calculated. This is the first bandwidth used to compute the mode tree when `bws` is not specified.

`cbw2` Number of modes for which the second critical bandwidth is calculated. This is the last bandwidth used to compute the mode tree when `bws` is not specified.

`display` Logical, if `TRUE` the mode tree plot is plotted. Default `TRUE`.

`logbw` Logical, if `TRUE` the plot displays and returns the log10 bandwidths. Default `logbw=FALSE`.

`logbw.regulargrid`
Logical, if `TRUE` a regular grid of bandwidths is created over the log10 scale. Default `logbw.regulargrid=FALSE`.

`...` Arguments to be passed to subsequent methods, `plot.gtmod` for the `plot`, `print` and `summary` methods.

Details

The mode tree for the sample given in `data` is computed. For this calculation, a kernel density estimator with Gaussian kernel and bandwidths `bws` is used. When it is plotted, it shows with the continuous lines the estimated mode locations at each bandwidth. The horizontal dashed lines indicate the splitting of a mode in more modes.

The NAs will be automatically removed.

Value

A list with class `"gtmod"` containing the following components:

`locations` Estimated mode locations for the bandwidths given in the row names.

`range.x` The data range, employed to represent the mode tree.

`range.bws` Employed bandwidths to compute the mode tree.

`logbw` Logical value indicating if the bandwidths are given in the log10 scale.

`sample.size` The number of non-missing observations in the sample used for obtaining the mode tree.

`call` The unevaluated expression, which consists of the named function applied to the given arguments.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Minnotte, M. C. and Scott, D. W. (1993). The mode tree: A tool for visualization of nonparametric density features, *Journal of Computational and Graphical Statistics*, **2**, 51–68.

Examples

```
#Mode tree using a grid of bandwidths between 0.2 and 0.5
set.seed(2016)
data=rnorm(50)
modetree(data,bws=c(0.2,0.5))
## Not run:
#Estimated locations of the five modes in the claw of Marron and Wand (1992)
library(nor1mix)
set.seed(2016)
n<-200
data<-nor1mix::rnorMix(n,MW.nm10)
#Mode tree between the critical bandwidths for 1 and 8 modes
modetree(data,cbw1=1,cbw2=8)
abline(v=1.5);abline(v=-1.5)

## End(Not run)
```

nmodes *Number of modes*

Description

This function computes the number of modes in a kernel density estimator using the Gaussian kernel and a given bandwidth parameter.

Usage

```
nmodes(data,bw,lowsup=-Inf,uppsup=Inf,n=2^15,full.result=F)
```

Arguments

data	Sample for computing a kernel density estimator and determine the number of modes.
bw	Bandwidth parameter for kernel density estimation.
lowsup	Lower limit for the random variable support. Just the number of modes greater than lowsup are taken into account. Default is -Inf.
uppsup	Upper limit for the random variable support. Just the number of modes greater than lowsup are taken into account. Default is Inf.
n	The number of equally spaced points at which the density is to be estimated. When n > 512, it is rounded up to a power of 2 as in the density function. Default n=2^15.
full.result	If this argument is TRUE then it returns the full result list, see below. Default full.result=FALSE.

Details

The number of modes in the interval provided by `lowsup` and `uppsup` is computed. For this calculation, a kernel density estimator with Gaussian kernel and bandwidth `bw` is used.

The NAs will be automatically removed.

Value

Depending on `full.result` either a number, the number of modes for the bandwidth provided in `bw`, or an object of class `"estmod"` which is a `list` containing the following components:

`nmodes` The number of modes for the bandwidth provided in `bw`.

`sample.size` The number of non-missing observations in the sample used for computing the number of modes.

`bw` Employed bandwidth for kernel density estimation.

`lowsup` Lower limit of the support where the number of modes are computed.

`ippsup` Upper limit of the support where the number of modes are computed.

`fnx` The `n` coordinates of the points where the density is estimated for computing the number of modes.

`fny` The estimated density values.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

Examples

```
# Number of modes in the interval (-1.5,1.5), using the bandwidth 0.5.
set.seed(2016)
data=rnorm(50)
nmodes(data,0.5,-1.5,1.5)
```

`plot.gtmod` *Plot, print and summarize a* `"gtmod"` *object*

Description

Plot, print and summarize methods for `"gtmod"` objects, i.e., the results of the graphical tools for exploring the number of modes.

Usage

```
## S3 method for class 'gtmod'
plot(x, addplot = FALSE, xlab = NULL, ylab = NULL,
col.lines = "black", col.sizer = NULL, addlegend = TRUE,
poslegend = "topright", ...)

## S3 method for class 'gtmod'
print(x,digits=getOption("digits"), ...)

## S3 method for class 'gtmod'
summary(object, bandwidths=TRUE, digits = getOption("digits"), width=1, levelmf=NULL, ...)
```

Arguments

`x` An object inheriting from class `"gtmod"`.

`object` An object of class `"gtmod"` for which a summary is desired.

`addplot` Logical, if `TRUE` the `modetree` plot is added to the current one. Default is `FALSE`.

`xlab` A title for the x axis. See `title`.

`ylab` A title for the y axis. See `title`.

`col.lines` Colors employed in the `modetree`. If the length is two, the first color is employed for representing the location of the modes and the second one for the splitting of the modes. Default is `col.lines="black"`.

`col.sizer` Colors employed in the `sizer` for indicating the behaviour of the smoothed curve. The first color indicates where it is significantly increasing, the second where it is not significantly different from zero, the third where it is significantly negative and the forth where the data are too sparse for meaningful inference. Default `col.sizer=c("red","orchid","blue","grey")`.

`addlegend` Logical, if `TRUE` the legend is displayed in the `sizer`. Default `TRUE`.

`poslegend` Position where the legend in the `sizer` should be displayed. Default `posLegend="topright"`.

`digits` Number of significant digits to use, see `formatC`.

`bandwidths` Logical, if `TRUE` the bandwidths range for which the modes are estimated is showed. Default `TRUE`.

`width` The total field width in each showed number, see `formatC`.

`levelmf` In the `modeforest`, just the pixels with a percentage (of times that an estimated mode falls on them) greater than this value will be considered. Default is `levelmf=0.5`.

`...` Arguments to be passed to subsequent methods, `plot.default` (`modetree`) or `image.default` (`modeforest` and `sizer`) for the `plot` method and `formatC` for the `print` and the `summary` method.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

See Also

modetree, modeforest and sizer, also for examples.

sizer *SIgnificant ZERo crossing*

Description

This function provides the SiZer map.

Usage

```
sizer(data,method=2,bws=NULL,gridsize=NULL,alpha=0.05,B=NULL,n0=NULL,
cbw1=NULL,cbw2=NULL,display=TRUE,logbw=TRUE,from=NULL,to=NULL,logbw.regulargrid=NULL,...)
```

Arguments

`data` Sample in which the SiZer map is computed.

`method` The method employed for computing the SiZer map. Available methods are: 1 (q1, pointwise Gaussian quantiles), 2 (q2, approximate simultaneous over x Gaussian quantiles), 3 (q3, bootstrap quantile simultaneous over x) and 4 (q4, bootstrap quantile simultaneous over x and h). Default `method=2`.

`bws` Vector or range of bandwidths. If it is a vector of size two, then it is used a grid of bandwidths between the given values. Default lower bandwidth is twice the grid size used for estimating the density and upper bandwidth equal to the range of the data. Unless it is specified a vector of size greater than two, the number of bandwidths employed is equal to the second element of `gridsize`.

`gridsize` Number of grid points in the horizontal (values of the variable, first element) and vertical (bandwidths, second element) axis. Default is `c(512,151)`.

`alpha` Significance level employed for determining the significant features. Default `alpha=0.05`.

`B` Number of replicates used for generating the SiZer map when method q3 or q4 are used. Default `B=100`.

`n0` When the effective sample size is below this quantity, the pixel in the SiZer map is shaded grey. Default `n0=5`.

`cbw1` Number of modes for which the first critical bandwidth is calculated. This is the first bandwidth used to compute the SiZer map when `bws` is not specified.

`cbw2` Number of modes for which the second critical bandwidth is calculated. This is the last bandwidth used to compute the SiZer map when `bws` is not specified.

`display` Logical, if `TRUE` the SiZer map is plotted. Default `TRUE`.

`logbw` Logical, if `TRUE` the plot displays and returns the log10 bandwidths. Default `logbw=TRUE`.

`from` First plotted value of the variable. Default is below the data minimum.

`to` Last plotted value of the variable. Default is above the data maximum.

`logbw.regulargrid` Logical, if `TRUE` a regular grid of bandwidths is created over the log10 scale. Default `logbw.regulargrid=FALSE`.

`...` Arguments to be passed to subsequent methods, `plot.gtmod` for the `plot`, `print` and `summary` methods.

Details

With this function the assessment of SIgnificant ZERo crossing of the derivative of the smoothed curve are computed for the sample given in `data`. For this calculation, a kernel density estimator with Gaussian kernel and bandwidths `bws` is used. When it is plotted, at a given location (horizontal axis) and using a specified bandwidth parameter (vertical axis), the SiZer map represents where the curve is significantly increasing (blue color by default), decreasing (red by default) or not significantly different from zero (orchid, a light tone of purple, by default). Thus, for a given bandwidth, a region significantly increasing followed by a region significantly decreasing (blue-red pattern by default) indicates where there is a significant peak.

For methods q2, q3 and q4, it is calculated where the data are too sparse for meaningful inference (grey color by default). A location-bandwidth pixel is classified in this last category when the estimated Effective Sample Size is less than `n0`. For more information, see Chaudhuri and Marron (1999).

For methods q3 and q4, the bootstrap quantiles are computed generating `B` random samples drawn with replacement from `data`.

The NAs will be automatically removed.

Value

A list with class `"gtmod"` containing the following components:

`sizer` Matrix indicating the significant behavior of the smoothed curve in each location-bandwidth pixel. One indicates significantly decreasing; two, not significantly different from zero; three, significantly increasing and four where the data are too sparse for meaningful inference.

`method` A number indicating what type of quantile was performed.

`lower.CI` Matrix containig the lower limit of the confidence interval in each location-bandwidth pixel.

`estimate` Matrix containing the derivative values of the kernel density estimation in each location-bandwidth pixel.

`upper.CI` Matrix containig the upper limit of the confidence interval in each location-bandwidth pixel.

`ESS` Matrix containing the Effective Sample Size in each location-bandwidth pixel.

`range.x` Employed location values to represent the SiZer map.

`range.bws` Employed bandwidths to compute the SiZer map.

`logbw` Logical value indicating if the bandwidths are given in the log10 scale.

`sample.size` The number of non-missing observations in the sample used for obtaining the SiZer map.

`call` The unevaluated expression, which consists of the named function applied to the given arguments.

Author(s)

Jose Ameijeiras–Alonso, Rosa M. Crujeiras and Alberto Rodríguez–Casal

References

Chaudhuri, P. and Marron, J. S. (1999). SiZer for exploration of structures in curves, *Journal of the American Statistical Association*, **94**, 807–823.

Examples

```
#SiZer map using a grid of bandwidths between 1 and 10
data(geyser)
data=geyser
sizer(data,bws=c(1,10))

## Not run:
#Different methods for calculating the confidence limits

#Pointwise Gaussian quantiles
sizer(data,method=1,bws=c(1,10))

#Approximate simultaneous over x Gaussian quantiles
sizer(data,method=2,bws=c(1,10))

#Bootstrap quantile simultaneous over x
sizer(data,method=3,bws=c(1,10))

#Bootstrap quantile simultaneous over x and h
sizer(data,method=4,bws=c(1,10))

## End(Not run)

#Adding the original mode tree for this sample
modetree(data,bws=c(0.8,10),logbw=TRUE,addplot=TRUE,col.lines="white")
```

`stamps` *Stamps thickness*

Description

This dataset, analysed in Izenman and Sommer (1988) and Ameijeiras–Alonso *et al.* (2018), consists of thickness measurements (in millimeters) of 485 unwatermarked used white wove stamps of the 1872 Hidalgo stamp issue of Mexico. All of them had an overprint with the year (1872 or either an 1873 or 1874) and some of them were watermarked (Papel Sellado or LA+-F).

Usage

```
data(stamps)
data(stamps1)
data(stamps2)
data(stampstable)
```

Format

`stamps` includes the thickness (in millimeters) of the different stamps in the 1872 Hidalgo stamp issue. `stampstable` reproduces the original frequency table in Izenman and Sommer (1988), including the `thickness`, the overprinted years (`1872` or `1873-1874`) and the watermarks (`Papel Sellado` or `LA+-F`). `stamps1` includes both the `thickness` and the overprinted `year`. `stamps2` includes both the `thickness` and the `watermark`. Note that the stamp with a thickness equal to 0.118 and the watermark `"Papel Sellado"` was removed in `stamps2` since it does not have a corresponding year, probably this watermark is associated with the observation of `thickness` 0.117 or 0.119.

Details

This is a classic example for determining the number of modes.

Source

Obtained from the original table in Izenman and Sommer (1988).

References

Jose Ameijeiras–Alonso, Rosa M. Crujeiras, Alberto Rodríguez–Casal (2018). Mode testing, critical bandwidth and excess mass, to appear in *Test*.

Izenman, A. J., and Sommer, C. J. (1988). Philatelic mixtures and multimodal densities. *Journal of the American Statistical association*, **83**, 941–953.

Examples

```
data("stamps")
# Histogram of Wilson (bin width 0.008)
seqx=seq(0.0585,0.1385,by=0.008)
hist(stamps,breaks=seqx)
# Histogram of Figure 1, Izenman and Sommer (bin width 0.002)
seqx=seq(0.0585,0.1385,by=0.002)
hist(stamps,breaks=seqx)

# Kernel density estimation of Izenman and Sommer (number of modes: 7)
locmodes(stamps,mod0=7,lowsup=0.04,uppsup=0.15,display=TRUE)
# Kernel density estimation of Ameijeiras-Alonso et al. (number of modes: 4)
locmodes(stamps,mod0=4,lowsup=0.04,uppsup=0.15,display=TRUE)
```

Index