Package ‘silvermantest’

July 26, 2013

Type Package

Title Package which implements the silvermantest

Version 1.0

Date 2013-02-18

Author Florian Schwaiger, Hajo Holzmann

Maintainer <schwaige@mathematik.uni-marburg.de>

Description This package contains the silverman.test, a test concerning the number of modes of an empirical density. Furthermore functions visualizing a number of tests (silverman.plot), plotting a number of densities for given number of modes (densities.plot), calculating a critical bandwidth (h.crit) and determining the number of modes for a given (empirical) density (nr.modes) are available.

License GPL-2

LazyLoad yes

Depends splines , methods

R topics documented:

silvermantest-package *Package to investigate the number of modes*

Description

Package which implements the silvermantest.

Details

Package:	silvermantest
Type:	Package
Version:	1.0
Date:	2013-02-18
Depends:	splines
License:	GPL-2
LazyLoad:	yes

This package contains the `silverman.test`, a test concerning the number of modes of an empirical density. Furthermore functions visualizing a number of tests (`silverman.plot`), plotting a number of densities for given number of modes (`densities.plot`), calculating a critical bandwidth (`h.crit`) and determining the number of modes for a given (empirical) density (`nr.modes`) are available.

Many thanks to Kaspar Rufibach, who created the first version of the test.

Author(s)

Florian Schwaiger, Hajo Holzmann

References

Silverman, B.W. (1981). *Using Kernel Density Estimates to investigate Multimodality*. Journal of the Royal Statistical Society, Series B, 43, 97-99.

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

See Also

`silverman.test`, `densities.plot`, `nr.modes`, `h.crit`, `silverman.plot`

Examples

```
#construct a bimodal dataset
x = c(rnorm(100,mean=-5),rnorm(100,mean=5))
#the H0: number of modes <= 1 should be rejected (small p-value)
s1 = silverman.test(x,1)
s1
#the H0: number of modes <= 2 should not be rejected (big p-value)
s2 = silverman.test(x,2)
```

```
s2
#if you want to replicate the test you can set the saved seed, e.g.
.Random.seed = s2@saved_seed
```

`densities.plot` *Plot densities*

Description

Plots a set of densities of a given dataset for selected numbers of modes.

Usage

```
densities.plot(x,modes=1:4,mark_modes=TRUE,in_one=TRUE*(length(modes)<4))
```

Arguments

`x`	vector of data
`modes`	vector with the number of modes to plot
`mark_modes`	boolean to mark the modes in the densities
`in_one`	boolean to select if all densities should be plotted in one graphic (default is TRUE if length(modes)<=3)

Details

`x` should contain the dataset. In the vector `modes` you can specify the number of modes for each density (e.g. 1,2,3,4). The function calculates the corresponding h.crit and plots the density. Furthermore with the boolean `mark_modes` you can deactivate the marking of the modes of each density.

Examples

```
data=rnorm(100)
#plots densities for modes=1,2,3 in one graphic (default if length(modes)<=3)
densities.plot(data,modes=1:3)
#plots densities for modes=1:4 in different graphics
densities.plot(data)
```

`h.crit` *Bandwidth calculation for density estimation*

Description

Calculates the smallest value so that the gaussian kernel density estimate of the given data `x` has `k` modes.

Usage

```
h.crit(x, k, prec = 6)
```

Arguments

`x`	vector of data
`k`	number of modes
`prec`	number of digits for precision of calculation

Details

The smaller you choose the bandwidth for a kernel density estimate, the larger the number of modes becomes. This function calculates the smallest value leading to a kernel density estimate with `k` number of modes.

See Also

`silverman.test`

Examples

```
par(mfrow=c(1,2))
x = rnorm(1000)
h0 = h.crit(x,1)
plot(density(x,bw=h0),main="k=1")
h1 = h.crit(x,8)
plot(density(x,bw=h1),main="k=8")
```

`landkreise` *Population and gross domestic product of german administrative districts*

Description

This data sets contains gross domestic products (german "BIP") of german administrative districts ("Landkreise") of the years 1992 until 2005 (in columns 3 to 15). Furthermore, you can find the population in the columns 16 to 28 (the number of wage earners in columns 29 to 41) of the corresponding administrative district.

Usage

```
data(landkreise)
```

Format

A data frame with 439 observations on the following 41 variables.

Examples

```
data(landkreise)
#BIP per wage earner
bippe<-landkreise[,3:15]/landkreise[,29:41] *1000
#logarithmised values
lbippe<-log(bippe)/log(10)
#logarithmised values of 1992
lbippe_1992<-na.omit(lbippe[,1])
plot(density(lbippe_1992), main="", xlab="", ylab="",cex.axis=1.5)
```

`nr.modes` *Number of modes*

Description

Calculates the number of modes for given y-values of a density function.

Usage

```
nr.modes(y)
```

Arguments

`y`	vector of y-values of a density function

See Also

`silverman.test`

Examples

```

x=rnorm(1000)
par(mfrow=c(1,2))

#with preset value for bandwidth of density function
nr.modes(density(x)$y)
plot(density(x),main="preset bw")

#with very small value for bandwidth
nr.modes(density(x,bw=0.05)$y)
plot(density(x,bw=0.05),main="small bw")
```

silverman.plot *Visualisation of a sequence of silvermantests*

Description

Plots the p-values of a sequence of `silverman.test`'s. This visualisation is an easy way to find the smallest number of modes where the null hypothesis of the `silverman.test` can be rejcted.

Usage

```
silverman.plot(x, kmin=1, kmax=5, alpha=0.05, adjust=FALSE)
```

Arguments

`x`	vector of data
`kmin`	smallest value for number of modes to test
`kmax`	biggest value for number of modes to test
`alpha`	level of significance
`adjust`	boolean to adjust p-value for k=1

Details

The null hypothesis of one silvermantest is that the underlying density has at most `k` modes (H0: number of modes <= k). This function calculates a sequence of silvermantests by increasing the parameter `k` of the hypothesises. The first test, which returns a p-value greater than the level of significance `alpha` shows that one cannot reject this null hypothesis.

Examples

```
#simulate a dataset with three modes
x = c(rnorm(100,mean=-5),rnorm(100),rnorm(100,mean=5))
#H0:number of modes <=1 and H0:number of modes <=2 are rejected
silverman.plot(x)
```

silverman.test *Silvermantest*

Description

The silvermantest tests the null hypothesis that an underlying density has at most `k` modes.

Usage

```
silverman.test(x, k, M = 999, adjust = FALSE, digits = 6)
```

Arguments

`x`	vector of data
`k`	number of modes for the null hypothesis
`M`	number of bootstrap replications
`adjust`	boolean to activate the adjusting of the p-value (valid if k=1) (see Hall and York)
`digits`	number of digits of the p-value

Details

The null hypothesis of a silvermantest is that the underlying density has at most `k` modes (H0: number of modes <= k). By rejecting this hypothesis (returned p-value is smaller than given level of significance) one can verify that the underlying density has **more** than `k` modes. The test uses `M` bootstrap replications and calculates the frequency that the critical bandwidth of one bootstrap sample data set (h_crit) is greater than the critical bandwidth of the given data (h_0). This frequency is interesting, because this event (hcrit of sample data > h_0) is equivalent to the event that the sample data has more than `k` modes. The frequency is the returned p-value.

Value

An object of the class Silvermantest (see: `Silvermantest-class`).

References

Silverman, B.W. (1981). *Using Kernel Density Estimates to investigate Multimodality*. Journal of the Royal Statistical Society, Series B, 43, 97-99.

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

Examples

```
#construct a bimodal dataset
x = c(rnorm(100,mean=0),rnorm(100,mean=3))
#silvermantest with adjusted p-value (only valid if k=1 is to test)
silverman.test(x,1,adjust=TRUE)
#the H0: number of modes <= 2 should not be rejected (big p-value)
silverman.test(x,2)
```

`Silvermantest-class` *Class "Silvermantest"*

Description

Objects of this class are the returned value of a `silverman.test`.

Slots

- `data`: the given dataset of a test
- `p_value`: p-value of the test (reject H0 for p-value less significance level)
- `saved_seed` the seed of the random generator before the test started
- `k` the silvermantest tests if the number of modes is less or equal to `k`

Methods

We provide the following methods for a Bimadalitytest object:

the visualisation of an object

Index