

# Package ‘sm’

April 19, 2009

**Version** 2.2-3

**Date** 2008-09-24

**Author** Adrian Bowman and Adelchi Azzalini. Ported to R by B. D. Ripley <ripley@stats.ox.ac.uk> up to version 2.0, version 2.1 by Adrian Bowman and Adelchi Azzalini, version 2.2 by Adrian Bowman.

**Maintainer** Adrian Bowman <adrian@stats.gla.ac.uk>

**Description** This is software linked to the book ‘Applied Smoothing Techniques for Data Analysis: The Kernel Approach with S-Plus Illustrations’ Oxford University Press.

**Title** Smoothing methods for nonparametric regression and density estimation

**Depends** R (>= 2.7.0)

**Suggests** rpanel, tkrplot, rgl, misc3d, akima, gam

**License** GPL-2

**URL** <http://www.stats.gla.ac.uk/~adrian/sm>

**Repository** CRAN

**Date/Publication** 2008-09-28 18:25:44

## R topics documented:

binning	2
geyser	4
h.select	4
hcv	6
hnorm	8
hsj	9
nise	10
nmise	11
nnbr	12
pause	13

provide.data . . . . .	13
sig.trace . . . . .	14
sm . . . . .	15
sm.ancova . . . . .	17
sm.autoregression . . . . .	19
sm.binomial . . . . .	20
sm.binomial.bootstrap . . . . .	21
sm.density . . . . .	22
sm.density.compare . . . . .	24
sm.discontinuity . . . . .	25
sm.monotonicity . . . . .	28
sm.options . . . . .	29
sm.poisson . . . . .	34
sm.poisson.bootstrap . . . . .	35
sm.regression . . . . .	36
sm.regression.autocor . . . . .	39
sm.rm . . . . .	40
sm.script . . . . .	42
sm.sigma . . . . .	43
sm.sigma2.compare . . . . .	44
sm.sphere . . . . .	45
sm.surface3d . . . . .	47
sm.survival . . . . .	48
sm.ts.pdf . . . . .	49
sm.variogram . . . . .	50

## Index 53

---

binning	<i>Construct frequency table from raw data</i>
---------	--

---

### Description

Given a vector or a matrix  $x$ , this function constructs a frequency table associated to appropriate intervals covering the range of  $x$ .

### Usage

```
binning(x, y, breaks, nbins)
```

### Arguments

$x, y$	a vector or a matrix with either one or two columns. If $x$ is a one-dimensional matrix, this is equivalent to a vector.
$breaks$	either a vector or a matrix with two columns (depending on the dimension of $x$ ), assigning the division points of the axis, or the axes in the matrix case. It must not include $Inf, -Inf$ or $NA$ s, and it must span the whole range of the $x$ points. If $breaks$ is not given, it is computed by dividing the range of $x$ into $nbins$ intervals for each of the axes.

`nbins` the number of intervals on the `x` axis (in the vector case), or a vector of two elements with the number of intervals on each axes of `x` (in the matrix case). If `nbins` is not given, a value is computed as `round(log(length(x))/log(2)+1)` or using a similar expression in the matrix case.

### Details

This function is called automatically (under the default settings) by some of the functions of the `sm` library when the sample size is large, to allow handling of datasets of essentially unlimited size. Specifically, it is used by `sm.density`, `sm.regression`, `sm.ancova`, `sm.binomial` and `sm.poisson`.

### Value

In the vector case, a list is returned containing the following elements: a vector `x` of the midpoints of the bins excluding those with 0 frequencies, its associated matrix `x.freq` of frequencies, the coordinates of the `midpoints`, the division points, and the complete vector of observed frequencies `freq.table` (including the 0 frequencies), and the vector `breaks` of division points. In the matrix case, the returned value is a list with the following elements: a two-dimensional matrix `x` with the coordinates of the midpoints of the two-dimensional bins excluding those with 0 frequencies, its associated matrix `x.freq` of frequencies, the coordinates of the `midpoints`, the matrix `breaks` of division points, and the observed frequencies `freq.table` in full tabular form.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

[sm](#), [sm.density](#), [sm.regression](#), [sm.binomial](#), [sm.poisson](#), [cut](#), [table](#)

### Examples

```
# example of 1-d use
x <- rnorm(1000)
xb <- binning(x)
xb <- binning(x, breaks=seq(-4,4,by=0.5))
# example of 2-d use
x <- rnorm(1000)
y <- 2*x + 0.5*rnorm(1000)
x <- cbind(x, y)
xb<- binning(x, nbins=12)
```

---

`geyser`*Old Faithful Geyser Data*

---

**Description**

A version of the eruptions data from the “Old Faithful” geyser in Yellowstone National Park, Wyoming. This version comes from Azzalini and Bowman (1990) and is of continuous measurement from 1st August to 15th August 1985.

Some nocturnal duration measurements were coded as 2, 3 or 4 minutes, having originally been described as ‘short’, ‘medium’ or ‘long’.

**Usage**

```
data(geyser)
```

**Format**

A data frame with 299 observations on 2 variables.

<code>duration</code>	numeric	Eruption time, in minutes
<code>waiting</code>	numeric	Waiting time to next eruption, in minutes

**References**

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

**See Also**

[faithful](#)

---

`h.select`*Selection of the smoothing parameter*

---

**Description**

This function selects a smoothing parameter for density estimation in one or two dimensions and for nonparametric regression with one or two covariates. Several methods of selection are available.

**Usage**

```
h.select(x, y = NA, weights = NA, group = NA, ...)
```

**Arguments**

x	a vector, or two-column matrix.
y	a vector of responses, in regression case.
weights	a vector of integers representing frequencies of individual observations. Use of this parameter is incompatible with <code>binning</code> ; hence <code>nbins</code> must then be set to 0 or left at its default value NA.
group	a vector of groups indicators (numeric or character values) or a factor
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function. There are three which are relevant for this function, namely <code>method</code> , which specifies the method of smoothing parameter selection, <code>df</code> , which specifies the approximate degrees of freedom associated with the selected smoothing parameter, and <code>structure.2d</code> which determines the form of the smoothing parameters in the two-dimensional case. A full description of these arguments are given in the documentation of <code>sm.options</code> .

**Details**

see the two references below for discussion of the methods of smoothing parameter selection.

When the `group` argument is set, the chosen method of smoothing parameter selection is applied to each group and the value returned is the geometric mean of these. This is intended for use in `sm.density.compare` and `sm.ancova`, where the same smoothing parameter is used for all groups so that the principal bias terms cancel when the estimates are compared.

**Value**

the value of the selected smoothing parameter.

**Side Effects**

none

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

Hurvich, C.M., Simonoff, J.S. and Tsai, C.-L. (1998). Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *J. R. Statistic. Soc., Series B*, 60, 271-293.

**See Also**

[sm](#), [hcv](#), [hsj](#), [hnorm](#)

**Examples**

```

x <- rnorm(50)
h.select(x)
h.select(x, method = "sj")

x <- matrix(rnorm(100), ncol = 2)
h.select(x)
sm.density(x, method = "cv")

x <- rnorm(50)
y <- x^2 + rnorm(50)
h.select(x, y)
sm.regression(x, y, method = "aicc")

x <- matrix(rnorm(100), ncol = 2)
y <- x[,1]^2 + x[,2]^2 + rnorm(50)
h.select(x, y, method = "cv", structure.2d = "common")
sm.regression(x, y, df = 8)

```

---

hcv

*Cross-validators choice of smoothing parameter*


---

**Description**

This function uses the technique of cross-validation to select a smoothing parameter suitable for constructing a density estimate or nonparametric regression curve in one or two dimensions.

**Usage**

```
hcv(x, y = NA, hstart = NA, hend = NA, ...)
```

**Arguments**

x	a vector, or two-column matrix of data. If y is missing these are observations to be used in the construction of a density estimate. If y is present, these are the covariate values for a nonparametric regression.
y	a vector of response values for nonparametric regression.
hstart	the smallest value of the grid points to be used in an initial grid search for the value of the smoothing parameter.
hend	the largest value of the grid points to be used in an initial grid search for the value of the smoothing parameter.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: <code>h.weights</code> , <code>ngrid</code> , <code>display</code> , <code>add</code> ; see the documentation of <code>sm.options</code> for their description.

## Details

See Sections 2.4 and 4.5 of the reference below.

The two-dimensional case uses a smoothing parameter derived from a single value, scaled by the standard deviation of each component.

This function does not employ a sophisticated algorithm and some adjustment of the search parameters may be required for different sets of data. An initial estimate of the value of `h` which minimises the cross-validatory criterion is located from a grid search using values which are equally spaced on a log scale between `hstart` and `hend`. A quadratic approximation is then used to refine this initial estimate.

## Value

the value of the smoothing parameter which minimises the cross-validation criterion over the selected grid.

## Side Effects

If the minimising value is located at the end of the grid of search positions, or if some values of the cross-validatory criterion cannot be evaluated, then a warning message is printed. In these circumstances altering the values of `hstart` and `hend` may improve performance.

## Note

As from version 2.1 of the package, a similar effect can be obtained with the new function `h.select`, via `h.select(x, method="cv")`. Users are encouraged to adopt this route, since `hcv` might be not accessible directly in future releases of the package.

## References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

## See Also

[h.select](#), [hsj](#), [hnorm](#)

## Examples

```
# Density estimation

x <- rnorm(50)
par(mfrow=c(1,2))
h.cv <- hcv(x, display="lines", ngrid=32)
sm.density(x, h=hcv(x))
par(mfrow=c(1,1))

# Nonparametric regression

x <- seq(0, 1, length = 50)
y <- rnorm(50, sin(2 * pi * x), 0.2)
```

```
par(mfrow=c(1,2))
h.cv <- hcv(x, y, display="lines", ngrid=32)
sm.regression(x, y, h=hcv(x, y))
par(mfrow=c(1,1))
```

---

hnorm

*Normal optimal choice of smoothing parameter in density estimation*


---

### Description

This functions evaluates the smoothing parameter which is asymptotically optimal for estimating a density function when the underlying distribution is Normal. Data in one, two or three dimensions can be handled.

### Usage

```
hnorm(x, weights)
```

### Arguments

x	a vector, or matrix with two or three columns, containing the data.
weights	an optional vector of integer values which allows the kernel functions over the observations to take different weights when they are averaged to produce a density estimate. This is useful, in particular, for censored data and to construct an estimate from binned data.

### Details

See Section 2.4.2 of the reference below.

### Value

the value of the asymptotically optimal smoothing parameter for Normal case.

### Note

As from version 2.1 of the package, a similar effect can be obtained with the new function `h.select`, via `h.select(x, method="normal", weights=weights)` or simply `h.select(x)`. Users are encouraged to adopt this route, since `hnorm` might be not accessible directly in future releases of the package.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

[h.select](#), [hcv](#), [hsj](#)

**Examples**

```
x <- rnorm(50)
hnorm(x)
```

---

`hsj`*Sheather-Jones choice of smoothing parameter for density estimation*

---

**Description**

This function uses the Sheather-Jones plug-in method of selecting a smoothing parameter which is suitable for constructing a density estimate in the one-dimensional case.

**Usage**

```
hsj(x)
```

**Arguments**

`x` a vector of data.

**Details**

See Section 2.4.4 of the reference below.

**Value**

the value of the smoothing parameter located by the Sheather-Jones method.

**Note**

As from version 2.1 of the package, a similar effect can be obtained with the new function `h.select`, via `h.select(x, method="sj")`. Users are encouraged to adopt this route, since `hsj` might be not accessible directly in future releases of the package.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[h.select](#), [hnorm](#), [hcv](#)

**Examples**

```
x <- rnorm(50)
hsj(x)
```

---

nise	<i>integrated squared error between a density estimate and a Normal density</i>
------	---

---

### Description

This function evaluates the integrated squared error between a density estimate constructed from a standardised version of the univariate data  $y$  and a standard normal density function.

### Usage

```
nise(y, hmult=1)
```

### Arguments

$y$	a vector of data.
hmult	a factor which can be used to multiply the normal optimal smoothing parameter before construction of the density estimate.

### Details

the data  $y$  are first standardised to have sample mean 0 and sample variance 1. The integrated squared error between a density estimate constructed from these standardised data and a standard normal distribution is then evaluated.

see Section 2.5 of the reference below.

### Value

the integrated squared error.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

[nmise](#)

### Examples

```
x <- rnorm(100)
nise(x)
```

---

nmise	<i>mean integrated squared error for density estimation with normal data</i>
-------	--

---

## Description

This function evaluates the mean integrated squared error of a density estimate which is constructed from data which follow a normal distribution.

## Usage

```
nmise(sd, n, h)
```

## Arguments

sd	the standard deviation of the normal distribution from which the data arise.
n	the sample size of the data.
h	the smoothing parameter used to construct the density estimate.

## Details

see Section 2.4 of the reference below.

## Value

the mean integrated squared error of the density estimate.

## References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

## See Also

[nise](#)

## Examples

```
x <- rnorm(50)
sd <- sqrt(var(x))
n <- length(x)
h <- seq(0.1, 2, length=32)
plot(h, nmise(sd, n, h), type = "l")
```

---

`nnbr`*nearest neighbour distances from data in one or two dimensions*

---

**Description**

This function calculates the  $k$  nearest neighbour distance from each value in  $x$  to the remainder of the data. In two dimensions, Euclidean distance is used after standardising the data to have unit variance in each component.

**Usage**

```
nnbr(x, k)
```

**Arguments**

$x$  the vector, or two-column matrix, of data.  
 $k$  the required order of nearest neighbour.

**Details**

see Section 1.7.1 of the reference below.

**Value**

the vector of nearest neighbour distances.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

none.

**Examples**

```
x <- rnorm(50)
hw <- nnbr(x, 10)
hw <- hw/exp(mean(log(hw)))
sm.density(x, h.weights=hw)
```

---

pause	<i>Pause before continuing execution</i>
-------	--

---

**Description**

If a program produces several plots on the same window, insertion of `pause()` between two plots suspends execution until the <Enter> key is pressed, to allow inspection of the current plot.

**Usage**

```
pause()
```

---

<code>provide.data</code>	<i>Making data available as data.frame</i>
---------------------------	--

---

**Description**

This is a utility function, widely used in the scripts accompanying the book described below. The function provides access to the dataset identified by `name`. For flexibility, the datasets are provided in ASCII form, with the name of each variable listed in the first row of the file. This function reads the files and makes the data available as a data frame.

**Usage**

```
provide.data(data, path, options = list())
```

**Arguments**

<code>data</code>	name of the data to be loaded and attached as <code>data.frame</code>
<code>path</code>	the path where the data and its documentation should be searched for, The default value is an appropriate sub-directory of the <code>sm</code> package.
<code>options</code>	A list of options passed to <code>sm.options</code> . The one used is <code>describe</code> , a logical flag. If <code>describe=TRUE</code> (default), a documentation file of the data is searched and printed, if available.

**Details**

the data file is assumed to be called `data.dat` and the documentation file describing the data (if present) is assumed to be called `data.doc`. If the `data.frame` is already attached, it is re-attached in the second position of the `search` list.

To set `describe=FALSE` for the rest of the current session, use `sm.options(describe=FALSE)`

The function can easily be adapted to play a similar role for other packages.

**Value**

none

**Side Effects**

messages are printed on the command window, describing progress of the operation. If `describe=TRUE` and a documentation file exists, this is printed on the command windows or another windows, depending on the type of platform where the program is executed.

**Author(s)**

Bowman, A.W. and Azzalini, A.

**See Also**

`data.frame`, `attach`, `sm`, `sm.options`

**Examples**

```
provide.data(birth)
```

---

sig.trace

*A significance trace for a hypothesis test*

---

**Description**

This function creates a significance trace for a hypothesis test based on a nonparametric smoothing procedure. The p-value of the test is plotted against a range of smoothing parameters.

**Usage**

```
sig.trace(expn, hvec, display="lines")
```

**Arguments**

<code>expn</code>	an S-Plus expression which should define the hypothesis test to be performed, with the value of the smoothing parameter <code>h</code> omitted from the function call.
<code>hvec</code>	a vector of smoothing parameters for which the test will be performed.
<code>display</code>	any character setting other than "none" will cause the significance trace to be plotted.

**Details**

see Section 5.2 of the reference below.

Only tests involving a univariate smoothing parameter may be used.

**Value**

a list containing vectors with the smoothing parameters and p-values.

**Side Effects**

If the largest p-value is greater than 0.05 then a horizontal line at 0.05 will be superimposed on any plot, for reference.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

none.

**Examples**

```
x <- runif(50, 0, 1)
y <- 5*x^2 + rnorm(50)
sig.trace(sm.regression(x, y, model = "linear", display="none"),
          hvec = seq(0.05, 0.3, length = 10))
```

---

sm

*The sm package: summary information*

---

**Description**

This package implements nonparametric smoothing methods described in the book of Bowman & Azzalini (1997)

**Details**

Missing data are allowed; they are simply removed, together with the associated variates from the same case, if any. Datasets of arbitrary size can be handled by the current version of `sm.density`, `sm.regression` and `sm.ancova`, using binning operations.

**Main Features**

The functions in the package use kernel methods to construct nonparametric estimates of density functions and regression curves in a variety of settings, and to perform some inferential operations.

Specifically, density estimates can be constructed for 1-, 2- and 3-dimensional data. Nonparametric regression for continuous data can be constructed with one or two covariates, and a variety of tests can be carried out. Several other data types can be handled, including survival data, time series, count and binomial data.

## Functions

The main functions are `sm.density` and `sm.regression`; other functions intended for direct access by the user are: `h.select`, `binning`, `sm.ancova`, `sm.autoregression`, `sm.binomial`, `sm.binomial.bootstrap`, `sm.poisson`, `sm.poisson.bootstrap`, `sm.options`, `sm.rm`, `sm.script`, `sm.sphere`, `sm.survival`, `sm.ts.pdf`. There are undocumented functions which are called by these.

## Scripts

The function `sm.script` is used to run a set of examples (called scripts) presented in the book quoted below. These scripts are associated with the package but the package can be used independently of them. The scripts are generally based on the functions of the package `sm`, but a few of them make use of the `gam` package.

## Requirements

R version  $\geq 2.1.0$ . Packages `gam` and `akima` are used by some of the scripts launched via `sm.script`, but they are not used by the functions of this package.

## Version

This is version 2.1. The most recent version of the package can be obtained from either of the web pages: <http://www.stats.gla.ac.uk/~adrian/sm>, [http://azzalini.stat.unipd.it/Book\\_sm](http://azzalini.stat.unipd.it/Book_sm)

## Details

The book by Bowman and Azzalini (1997) provides more detailed and background information. Algorithmic aspects of the software are discussed by Bowman & Azzalini (2003). Differences between the first version of the package, described in the book, and the current one are summarized in the file `history.txt` which is distributed with the package.

## Acknowledgements

Important contributions to prototype versions of functions for some specific techniques included here were made by a succession of students; these include Stuart Young, Eileen Wright, Peter Foster, Angela Diblasi, Mitchum Bock and Adrian Hines. We are grateful for all these interactions. These preliminary version have been subsequently re-written for inclusion in the public release of the package, with the exception of the functions for three-dimensional density estimation, written by Stuart Young. We also thank Luca Scrucca who made useful comments and who has ported the software to XLispStat. We are particularly grateful to Brian Ripley for substantial help in the production of installation files, the creation of MS-Windows versions, initial porting of the software from S-Plus to R and for maintaining the package on CRAN for several years.

## Licence

This package and its documentation are usable under the terms of the "GNU General Public License", a copy of which is distributed with the package.

**Author(s)**

Adrian Bowman (Dept Statistics, University of Glasgow, UK) and Adelchi Azzalini (Dept Statistical Sciences, University of Padua, Italy). Please send comments, error reports, etc. to the authors via the web pages mentioned above.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

Bowman, A.W. and Azzalini, A. (2003). Computational aspects of nonparametric smoothing with illustrations from the `sm` library. *Computational Statistics and Data Analysis*, **42**, 545–560.

---

 sm.ancova

*Nonparametric analysis of covariance*


---

**Description**

This function allows a set of nonparametric regression curves to be compared, both graphically and formally in a hypothesis test. A reference model, used to define the null hypothesis, may be either equality or parallelism. Regression surfaces can also be compared in a test but a graphical display is not produced.

**Usage**

```
sm.ancova(x, y, group, h, model = "none", h.alpha = NA, weights=NA,
          covar = diag(1/weights), ...)
```

**Arguments**

<code>x</code>	a vector or two-column matrix of covariate values.
<code>y</code>	a vector of response values.
<code>group</code>	a vector of group indicators.
<code>h</code>	the smoothing parameter to be used in the construction of each of the regression curves. If this is missing the method of smoothing parameter selection specified by <code>sm.options</code> will be applied.
<code>model</code>	a character variable which defines the reference model. The values "none", "equal" and "parallel" are possible.
<code>h.alpha</code>	the value of the smoothing parameter used when estimating the vertical separations of the curves under the parallelism model. If this is missing, it is set to $2 * r / n$ , where $r$ denotes the range of the data and $n$ the sample size.
<code>weights</code>	case weights; see the documentation of <code>sm.regression</code> for a full description.

`covar` the (estimated) covariance matrix of  $y$ . The default value assumes the data to be independent. Where appropriate, the covariance structure of  $y$  can be estimated by the user, externally to `sm.ancova`, and passed through this argument. This is used in the hypothesis tests but not in the construction of the reference band for comparing two groups (and so graphics are disabled in this case).

`...` other optional parameters are passed to the `sm.options` function, through a mechanism which limits their effect only to this call of the function. Those relevant for this function are the following: `display`, `ngrid`, `eval.points`, `xlab`, `ylab`; see the documentation of `sm.options` for their description.

### Details

see Sections 6.4 and 6.5 of the book by Bowman & Azzalini, and the papers by Young & Bowman listed below. This function is a developed version of code originally written by Stuart Young.

### Value

a list containing an estimate of the error standard deviation and, where appropriate, a p-value and reference model. If the parallelism model has been selected then a vector of estimates of the vertical separations of the underlying regression curves is also returned.

### Side Effects

a plot on the current graphical device is produced, unless `display="none"`

### References

- Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.
- Young, S.G. and Bowman, A.W. (1995). Nonparametric analysis of covariance. *Biometrics* **51**, 920–931.
- Bowman, A.W. and Young, S.G. (1996). Graphical comparison of nonparametric curves. *Applied Statistics* **45**, 83–98.

### See Also

[sm.regression](#), [sm.density.compare](#), [sm.options](#)

### Examples

```
x <- runif(50, 0, 1)
y <- 4*sin(6*x) + rnorm(50)
g <- rbinom(50, 1, 0.5)
sm.ancova(x, y, g, h = 0.15, model = "equal")
```

---

sm.autoregression *Nonparametric estimation of the autoregression function*

---

### Description

This function estimates nonparametrically the autoregression function (conditional mean given the past values) of a time series  $x$ , assumed to be stationary.

### Usage

```
sm.autoregression(x, h = hnorm(x), d = 1, maxlag = d, lags,
                 se = FALSE, ask = TRUE)
```

### Arguments

<code>x</code>	vector containing the time series values.
<code>h</code>	the bandwidth used for kernel smoothing.
<code>d</code>	number of past observations used for conditioning; it must be 1 (default value) or 2.
<code>maxlag</code>	maximum of the lagged values to be considered (default value is <code>d</code> ).
<code>lags</code>	if <code>d==1</code> , this is a vector containing the lags considered for conditioning; if <code>d==2</code> , this is a matrix with two columns, whose rows contains pair of values considered for conditioning.
<code>se</code>	if <code>se==T</code> , pointwise confidence bands are computed of approximate level 95%.
<code>ask</code>	if <code>ask==TRUE</code> , the program pauses after each plot until <Enter> is pressed.

### Details

see Section 7.3 of the reference below.

### Value

a list with the outcome of the final estimation (corresponding to the last value or pairs of values of lags), as returned by `sm.regression`.

### Side Effects

graphical output is produced on the current device.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

[sm.regression](#), [sm.ts.pdf](#)

**Examples**

```
library(stats)
data(lynx)
sm.autoregression(log(lynx), maxlag=3, se=TRUE)
sm.autoregression(log(lynx), lags=cbind(2:3,4:5))
```

---

sm.binomial

*Nonparametric logistic regression*


---

**Description**

This function estimates the regression curve using the local likelihood approach for a vector of binomial observations and an associated vector of covariate values.

**Usage**

```
sm.binomial(x, y, N = rep(1, length(y)), h, ...)
```

**Arguments**

x	vector of the covariate values
y	vector of the response values; they must be nonnegative integers not larger than those of N.
h	the smoothing parameter; it must be positive.
N	a vector containing the binomial denominators. If missing, it is assumed to contain all 1's.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function; those relevant for this function are the following: <code>add</code> , <code>col</code> , <code>display</code> , <code>eval.points</code> , <code>nbins</code> , <code>ngrid</code> , <code>pch</code> , <code>xlab</code> , <code>ylab</code> ; see the documentation of <code>sm.options</code> for their description.

**Details**

see Sections 3.4 and 5.4 of the reference below.

**Value**

A list containing vectors with the evaluation points, the corresponding probability estimates, the linear predictors, the upper and lower points of the variability bands (on the probability scale) and the standard errors on the linear predictor scale.

**Side Effects**

graphical output will be produced, depending on the value of the `display` parameter.

## References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

## See Also

[sm.binomial.bootstrap](#), [sm.poisson](#), [sm.options](#), [glm](#), [binning](#)

## Examples

```
## Not run:
# the next example assumes that all binomial denominators are 1's
sm.binomial(dose, failure, h=0.5)
# in the next example, (some of) the dose levels are replicated
sm.binomial(dose, failure, n.trials, h=0.5)
## End(Not run)
#
provide.data(birth)
sm.binomial(Lwt[Smoke=="S"], Low[Smoke=="S"], h=20,
            xlab='mother weight[Smoke=="S"]')
#
x<- seq(0,1,length=30)
y<- rbinom(30,10,prob=2*sin(x)/(1+x))
sm.binomial(x,y,N=rep(10,30), h=0.25)
```

---

```
sm.binomial.bootstrap
```

*Bootstrap goodness-of-fit test for a logistic regression model.*

---

## Description

This function is associated with `sm.logit` for the underlying fitting procedure. It performs a Pseudo-Likelihood Ratio Test for the goodness-of-fit of a standard parametric logistic regression of specified degree in the covariate `x`.

## Usage

```
sm.binomial.bootstrap(x, y, N = rep(1, length(x)), h, degree = 1,
                     fixed.disp=FALSE, ...)
```

## Arguments

<code>x</code>	vector of the covariate values
<code>y</code>	vector of the response values; they must be nonnegative integers.
<code>h</code>	the smoothing parameter; it must be positive.
<code>N</code>	a vector containing the binomial denominators. If missing, it is assumed to contain all 1's.

`degree` specifies the degree of the fitted polynomial in  $x$  on the logit scale (default=1).  
`fixed.disp` if TRUE, the dispersion parameter is kept at value 1 across the simulated samples, otherwise the dispersion parameter estimated from the sample is used to generate samples with that dispersion parameter (default=FALSE).  
`...` additional parameters passed to `sm.binomial`.

### Details

see Section 5.4 of the reference below.

### Value

a list containing the observed value of the Pseudo-Likelihood Ratio Test statistic, its observed p-value as estimated via the bootstrap method, and the vector of estimated dispersion parameters when this value is not forced to be 1.

### Side Effects

Graphical output representing the bootstrap samples is produced on the current graphical device. The estimated dispersion parameter, the value of the test statistic and the observed significance level are printed.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

`sm.binomial`, `sm.poisson.bootstrap`

### Examples

```
## Not run: sm.binomial.bootstrap(concentration, dead, N, 0.5, nboot=50)
```

---

`sm.density`

*Nonparametric density estimation in one, two or three dimensions.*

---

### Description

This function creates a density estimate from data in one, two or three dimensions. In two dimensions a variety of graphical displays can be selected, and in three dimensions a contour surface can be plotted. A number of other features of the construction of the estimate, and of its display, can be controlled.

If the `rpanel` package is available, an interactive panel can be activated to control various features of the plot.

If the `rgl` package is also available, rotatable plots are available for the two- and three-dimensional cases. (For three-dimensional data, the `misc3d` package is also required.)

**Usage**

```
sm.density(x, h, model = "none", weights = NA, group=NA, ...)
```

**Arguments**

<code>x</code>	a vector, or a matrix with two or three columns, containing the data.
<code>h</code>	a vector of length one, two or three, defining the smoothing parameter. A normal kernel function is used and <code>h</code> is its standard deviation. If this parameter is omitted, a normal optimal smoothing parameter is used.
<code>model</code>	This argument applies only with one-dimensional data. Its default value is "none". If it is set to "Normal" (or indeed any value other than "none") then a reference band, indicating where a density estimate is likely to lie when the data are normally distributed, will be superimposed on any plot.
<code>weights</code>	a vector of integers representing frequencies of individual observations. Use of this parameter is incompatible with binning; hence <code>nbins</code> must then be set to 0 or left at its default value NA.
<code>group</code>	a vector of groups indicators (numeric or character values) or a factor.
<code>...</code>	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: <code>hmult</code> , <code>h.weights</code> , <code>band</code> , <code>add</code> , <code>lty</code> , <code>display</code> , <code>props</code> , <code>xlab</code> , <code>ylab</code> , <code>zlab</code> , <code>xlim</code> , <code>ylim</code> , <code>yht</code> , <code>nbins</code> , <code>ngrid</code> , <code>eval.points</code> , <code>panel</code> , <code>positive</code> , <code>delta</code> , <code>theta</code> , <code>phi</code> ; see the documentation of <code>sm.options</code> for their description.

**Details**

see Chapters 1, 2 and 6 of the reference below. In the three-dimensional case, the contours of the density estimate are constructed by the `contour3d` function in the `misc3d` package of Feng & Tierney.

**Value**

a list containing the values of the density estimate at the evaluation points, the smoothing parameter, the smoothing parameter weights and the kernel weights. For one- and two-dimensional data, the standard error of the estimate (on the square root scale, where the standard error is approximately constant) and the upper and lower ends of a variability band are also supplied. Less information is supplied when the smoothing parameter weights or kernel weights are not all 1, or when `positive` is set to TRUE.

**Side Effects**

a plot is produced, unless the option `display="none"` is set.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[h.select](#), [hnorm](#), [hsj](#), [hcv](#), [nise](#), [nmise](#), [sm](#), [sm.sphere](#), [sm.regression](#), [sm.options](#)

**Examples**

```
# A one-dimensional example
y <- rnorm(50)
sm.density(y, model = "Normal")
# sm.density(y, panel = TRUE)

# A two-dimensional example
y <- cbind(rnorm(50), rnorm(50))
sm.density(y, display = "image")
# sm.density(y, panel = TRUE)

# A three-dimensional example
# y <- cbind(rnorm(50), rnorm(50), rnorm(50))
# sm.density(y)
```

---

sm.density.compare *Comparison of univariate density estimates*

---

**Description**

This function allows a set of univariate density estimates to be compared, both graphically and formally in a bootstrap hypothesis test of equality.

**Usage**

```
sm.density.compare(x, group, h, model = "none", ...)
```

**Arguments**

x	a vector of data.
group	a vector of group labels.
h	the smoothing parameter to be used in the construction of each density estimate. Notice that the same smoothing parameter is used for each group. If this value is omitted, the mean of the normal optimal values for the different groups is used.
model	the default value is "none" which restricts comparison to plotting only. The alternative value "equal" can produce a bootstrap hypothesis test of equality and the display of an appropriate reference band.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: <code>method</code> , <code>df</code> , <code>band</code> , <code>test</code> , <code>nboot</code> , plus those controlling graphical display (unless <code>display="none"</code> is set); see the documentation of <a href="#">sm.options</a> for their description.

**Details**

see Section 6.2 of the reference below.

**Value**

When `model` is set to "none", nothing is returned. When "model" is set to "equal", a list containing the smoothing parameter and the p-value of the test is returned. When `band` takes the value TRUE, and there are only two groups to compare, the list contains in addition the upper and lower end-points of the reference band for equality.

**Side Effects**

a plot on the current graphical device is produced, unless `display="none"`.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[sm.density](#), [sm.ancova](#), [sm.options](#)

**Examples**

```
y <- rnorm(100)
g <- rep(1:2, rep(50,2))
sm.density.compare(y, g, model="equal")
```

---

sm.discontinuity    *The detection of discontinuities in a regression curve or surface.*

---

**Description**

This function uses a comparison of left and right handed nonparametric regression curves to assess the evidence for the presence of one or more discontinuities in a regression curve or surface. A hypothesis test is carried out, under the assumption that the errors in the data are approximately normally distributed. A graphical indication of the locations where the evidence for a discontinuity is strongest is also available.

**Usage**

```
sm.discontinuity(x, y, h, hd, ...)
```

**Arguments**

x	a vector or two-column matrix of covariate values.
y	a vector of responses observed at the covariate locations.
h	a smoothing parameter to be used in the construction of the nonparametric regression estimates. A normal kernel function is used and h is its standard deviation(s). However, if this argument is omitted h will be selected by an approximate degrees of freedom criterion, controlled by the df parameter. See sm.options for details.
hd	a smoothing parameter to be used in smoothing the differences of the left and right sided nonparametric regression estimates. A normal kernel function is used and hd is its standard deviation(s). However, if this argument is omitted hd will be set to $h * \sqrt{0.25}$ , and h reset to $h * \sqrt{0.25}$ , when x is a vector. When x is a matrix, hd will be set to $h * \sqrt{0.5}$ and h will be reset to the same value.
...	other optional parameters are passed to the sm.options function, through a mechanism which limits their effect only to this call of the function; those relevant for this function are add, eval.points, ngrid, se, band, xlab, ylab, xlim, ylim, lty, col; see the documentation of sm.options for their description.

**Details**

The reference below describes the statistical methods used in the function. There are minor differences in some computational details of the implementation.

**Value**

a list containing the following items

p	the p-value for the test of the null hypothesis that no discontinuities are present.
sigma	the estimated standard deviation of the errors.
eval.points	the evaluation points of the nonparametric regression estimates. When x is a matrix, eval.points is also a matrix whose columns define the evaluation grid of each margin of the evaluation rectangle.
st.diff	a vector or matrix of standardised differences between the left and right sided estimators at the evaluation points.
diffmat	when x is a vector, this contains the locations and standardised differences where the latter are greater than 2.5.
angle	when x is a matrix, this contains the estimated angles at which the standardised differences were constructed.
h	the principal smoothing parameter.
hd	the smoothing parameter used for double-smoothing (see the reference below).

**Side Effects**

a plot on the current graphical device is produced, unless the option display="none" is set.

## References

Bowman, A.W., Pope, A. and Ismail, B. (2006). Detecting discontinuities in nonparametric regression curves and surfaces. *Statistics & Computing*, 16, 377–390.

## See Also

[sm.regression](#), [sm.options](#)

## Examples

```
par(mfrow = c(3, 2))

provide.data(nile)
sm.discontinuity(Year, Volume, hd = 0)
sm.discontinuity(Year, Volume)

ind <- (Year > 1898)
plot(Year, Volume)
h <- h.select(Year, Volume)
sm.regression(Year[!ind], Volume[!ind], h, add = TRUE)
sm.regression(Year[ind], Volume[ind], h, add = TRUE)

hvec <- 1:15
p <- numeric(0)
for (h in hvec) {
  result <- sm.discontinuity(Year, Volume, h, display = "none", verbose = 0)
  p <- c(p, result$p)
}
plot(hvec, p, type = "l", ylim = c(0, max(p)), xlab = "h")
lines(range(hvec), c(0.05, 0.05), lty = 2)

provide.data(trawl)
Position <- cbind(Longitude, Latitude)
ind <- (Longitude < 143.8)
sm.regression(Position[ind,], Score1[ind], theta = 35, phi = 30)
sm.discontinuity(Position[ind,], Score1[ind], col = "blue")

par(mfrow = c(1, 1))

# The following example takes longer to run.
# Alternative values for nside are 32 and 64.
# Alternative values of yjump are 1 and 0.5.
# nside <- 16
# yjump <- 2
# x1 <- seq(0, 1, length = nside)
# x2 <- seq(0, 1, length = nside)
# x <- expand.grid(x1, x2)
# x <- cbind(x1 = x[, 1], x2 = x[, 2])
# y <- rnorm(nside * nside)
# ind <- (sqrt((x[, 1] - 0.5)^2 + (x[, 2] - 0.5)^2) <= 0.25)
# y[ind] <- y[ind] + yjump
# image(x1, x2, matrix(y, ncol = nside))
```

```
# sm.discontinuity(x, y, df = 20, add = TRUE)
```

---

```
sm.monotonicity    A test of monotonicity in a regression curve.
```

---

## Description

This function uses the idea of a ‘critical bandwidth’ to assess the evidence that a regression curve is non-monotonic. A hypothesis test is carried out by bootstrap methods and the empirical p-value is reported. Response variables on a continuous scale or with binomial variation can be handled.

## Usage

```
sm.monotonicity(x, y, N = rep(1, length(y)), h, type = "continuous", ...)
```

## Arguments

x	a vector of covariate values.
y	a vector of responses observed at the covariate locations.
N	a vector of sample sizes at the covariate locations, when the responses have a binomial error structure.
h	a smoothing parameter to be used in the construction of the nonparametric regression estimates. A normal kernel function is used and h is its standard deviation(s). However, if this argument is omitted h will be selected automatically, using the method which is currently active. See <code>sm.options</code> and <code>h.select</code> for details.
type	an indicator of whether the response variable is on a "continuous" or "binomial" scale.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function; some of those relevant for this function are <code>add</code> , <code>ngrid</code> , <code>xlab</code> , <code>ylab</code> , <code>xlim</code> , <code>ylim</code> , <code>lty</code> , <code>col</code> ; see the documentation of <code>sm.options</code> for their description.

## Details

The first reference below describes the statistical methods used in the function. The test is an extension of one by Silverman (1986) for density estimation.

## Value

a list containing the following items

p	the p-value for the test of the null hypothesis that the true curve is monotonic.
hcrit	the ‘critical’ smoothing parameter. This is the smallest value which, when applied to the observed data, makes the curve monotonic.
h	the smoothing parameter used for double-smoothing (see the reference below).

**Side Effects**

a plot of the curves generated by the bootstrap procedure is produced, unless the option `display="none"` is set. Those curves which are non-monotonic, and therefore contribute to the empirical p-value, are drawn in red.

**References**

Bowman, A.W., Jones, M.C. and Gijbels, I. (1998). Testing monotonicity of regression. *J.Comp.Graph.Stat.* 7, 489-500.

**See Also**

[sm.regression](#), [sm.options](#)

**Examples**

```
#      Radiocarbon dating data

provide.data(radioc)
ind      <- (Cal.age>5000 & Cal.age<6000)
cal.age <- Cal.age[ind]
rc.age  <- Rc.age[ind]
sm.monotonicity(cal.age, rc.age, method = "aicc", nboot = 200)

#      Hosmer & Lemeshow birth data

provide.data(birth)
sm.monotonicity(Lwt[Smoke == "N"], Low[Smoke == "N"], type = "binomial")
```

---

sm.options                      *Set or return options of sm library*

---

**Description**

Provides a means to control the behaviour of the `sm` library such as the colour of the plotted lines, the size of the grid in 2-D estimation, the set of evaluations points, and many others.

**Usage**

```
sm.options(...)
```

**Arguments**

...                      a list may be given as the only argument, or any number of arguments may be in the `name=value` form, or no argument at all may be given. See the `Value` and `Details` sections for explanation.

Arguments which are set by a function call will remain in effect until the end of the current S-plus session, unless overwritten by a subsequent call. In addition,

they can be added as optional parameters of calls to specific functions of the `sm` library; in this case, their effect is limited to that function call. See the documentation of specific functions for the list of options which are recognised by that function. Notice that some options are relevant only to some functions.

- `add` a logical value which controls whether the estimate is added to the current plot. Its default value is `FALSE`, which creates a new plot. This argument applies only with one-dimensional data or to contour and `rgl` created from two-dimensional data.
- `alpha` an parameter, lying between 0 and 1, which controls the transparency of the surface used to construct an `rgl` plot for regression with two covariates. The default value is 0.7.
- `alpha.mesh` an parameter, lying between 0 and 1, which controls the transparency of the mesh lines used to construct an `rgl` plot for regression with two covariates. The default value is 1.
- `band` a logical value which controls whether the distance between the nonparametric estimate and a reference model should be indicated as a band (one covariate), or through colour painting of the regression surface (two covariates). This is activated only when a model has been nominated through the `model` parameter. In the case of two covariates, the setting of the argument `col` has priority over `band`. The setting `se = TRUE` can also activate this feature.
- `col` the colour used for plotting observed points and estimated curves. Where groups are used, with one covariate, `col` may be set to a vector of colours associated with the groups. In regression with two covariates using an `rgl` display, `col` may be set to a single colour, or to the values "height" or "se". These letter two setting cause the surface to be painted according to its height or to standard error information; see the information on the parameters `se`, `se.breaks` and `model`. Default: `col=1`.
- `col.band` the colour used for the reference band when a `model` is specified in regression with one covariate. Default: `col.band="cyan"`.
- `col.mesh` the colour used for the 'wire mesh' representation plotting observed points in an `rgl` display for regression with two covariates. This can also be set by the second component of a vector of length two which is set for `col`. Default: `col.mesh="black"`.
- `col.palette` the colours used for shading an image plot, or for surface painting in an `rgl` display, for regression with two covariates. Default: `col.palette=topo.colors(12)`.
- `col.points` the colour used for plotting observed points in a regression with one covariate or an `rgl` display for regression with two covariates. Default: `col.points="black"`.
- `delta` in `sm.density`, a value which will be added to the data before they are log transformed in the procedure to handle positive data. The value of `delta` is used only when `positive` takes the value `TRUE`. The default value is the smallest value observed in each dimension. This argument does not apply with three-dimensional data. Default: `delta=NA`
- `describe` logical flag which affects the behaviour of `sm.script` and `provide.data`. If `describe=TRUE` (default), a data documentation file is printed.

- `df` approximate degrees-of-freedom of the smoothing parameter used in `sm.regression`, when a numerical value of `h` is not specified. In this case, the equivalent value of `h` will be computed and included in the list returned on exit from `sm.regression`. Default value is 6 if `x` is a vector and 12 if `x` is a matrix.
- `diff.ord` in `sm.regression`, an integer defining the degree of differencing to be applied in the estimation process. When this argument is set to 1, the method of Rice, based on the squared differences of pairs of neighbouring observations, is used. When the argument is set to 2 (default), the method of Gasser, Sroka and Jennen-Steinmetz, based on differences between each observation and a linear interpolation from its two neighbours, is used.
- `display` This argument applies only with one- or two-dimensional data. The setting "none" will prevent any graphical output from being produced. In one dimensions, the default setting "line" will produce the estimate. (For compatibility with earlier versions of the package, the setting "se" will produce a variability band to show the variability, but not the bias, of the estimate. This should now be controlled by setting the separate parameter `se` to TRUE.) In two dimensions, the default setting "persp" will produce a perspective plot of the estimate, while the settings "slice", "image" and "rgl" will produce slice (contour), image or rgl plots.
- `eval.grid` logical flag which controls how the options `eval.points` are used for two-dimensional data. If `eval.grid=TRUE` (default), evaluation is performed at points obtained by the cross-product of the two columns of `eval.points`. If `eval.grid=FALSE` then evaluation is performed at points with coordinates specified by the coordinates in `eval.points`
- `eval.points` the points at which the density or the regression curve or surface estimate should be evaluated, for the values returned in the result of the function. This should be a vector for one-dimensional data and a two-column matrix for two-dimensional data. This argument does not apply with three-dimensional data.
- `h.weights` a vector of weights which multiply the smoothing parameter used in the kernel function at each observation. This argument does not apply with three-dimensional data. Default value: 1.
- `hmult` a factor which can be used to multiply the normal smoothing parameter before construction of the density estimate. Default value: 1.
- `hull` a logical value which controls whether the estimate is evaluated and plotted only on grid points which fall within the convex hull of the data. When this argument is set to FALSE, evaluation and plotting take place at all grid points where the contribution from at least one kernel function is non-negligible. Both of these settings ensure that the estimate is not evaluated at points where there are no observations nearby. This argument applies only to `sm.regression` and `sm.discontinuity` in the case of two covariates.
- `lty` the line type used to plot the estimate. This argument applies only when the estimate is displayed as a curve or a contour. Default value: 1.
- `method` the method used to select smoothing parameters. In density estimation the default is "normal" which uses a value which is asymptotically opti-

- mal for the normal distribution. Other possibilities are "cv" for cross-validation and "sj" for the Sheather-Jones method. In nonparametric regression, the default is "df" which selects a smoothing parameters associated with the approximate degrees of freedom given in the df option. Other possibilities are "cv" for cross-validation and "aicc" for an AIC-based method proposed by Hurvich, Simonoff and Tsai.
- nbins** the number of bins used in one-dimensional binning operations; in two-dimensional cases, nbins refers to the number of bins formed along each axis. Bins with 0 observed frequencies are ignored. If nbins=0, binning is not performed; if nbins=NA (default), binning is switched on when the number of observations exceeds a certain threshold, which depends on the function.
- nboot** number of samples generated in bootstraps. Default value: 100.
- ngrid** the number of points in the regular grid used to plot the estimate. For two- and three-dimensional data, ngrid refers to the number of points along the axis in each dimension. The same parameter is also used by a few other functions which perform some form of search (e.g. hcv). Default value for sm.regression: 50 and 20 for 1-, 2-dimensional data, respectively. Default value for sm.density: 100, 50 and 20 for 1-, 2- and 3-dimensional data, respectively.
- panel** a logical value which, when set to true, creates a panel which allows interactive control of sm.regression or sm.density plots for one- or two-dimensional data. The panel can be used to alter the value of the smoothing parameter and control a variety of other settings.
- panel.plot** a logical value which, when set to true (the default), places the plot inside the control panel (see the panel argument above), This creates a neater screen arrangement.
- pch** the standard plotting character identified for data plotting. Default value: 1.
- period** a vector of length one or two identifying the period for covariates which are on a periodic scale. Non-periodic covariates are identified by NA. Default value: NA.
- phi** the vertical rotation (in degrees) of perspective plots of estimate in the form of surfaces. Default value: 40.
- poly.index** an integer defining local constant (0) or local linear (1) smoothing. Default value: 1.
- positive** a logical value which indicates whether the data should be assumed to take positive values only, in sm.density. When this argument is set to TRUE, a log transformation is applied to the data before construction of a density estimate. The result is transformed back to the original scale. This argument does not apply with three-dimensional data. Default value: FALSE.
- props** a vector defining the proportions of the data to be included within each contour in a slice plot, from two-dimensional data, or a contour surface plot, from three-dimensional data. In the three-dimensional case only the first element of the vector will be used. This argument does not apply to one-dimensional data. Default value: c(75, 50, 25).
- rugplot** logical flag which regulates whether a rugplot is superimposed to the density estimate, in the univariate case. Default value: TRUE.

- se logical flag which regulates whether a standard error information is added to the plot produced by `sm.regression`. If a `model` is specified, then these standard errors refer to the difference between this fitted model and the nonparametric regression estimate. Default value: `TRUE`.
- se.breaks a numerical vector which defines the cut-points, on a standard error scale, for the assignment of colours when painting a regression surface with standard error information. Default value: `c(-3, -2, 3, 3)`.
- show.script logical flag which affects the behaviour of `sm.script` when this is called with non-empty argument. If `show.script=TRUE` (default) a window is opened to display the source code of the script.
- size an integer which defines the size of plotted points in `rgl` displays. The default value is 2.
- structure.2d the structure of the smoothing parameter in two-dimensional settings. The default is `"scaled"`, which uses the structure `(h*sd(x[,1]), h*sd(x[,2]))`. Other possibilities are `"separate"`, which uses `(h1, h2)`, and `"common"` which uses `(h, h)`. The `"common"` option may be particularly appropriate when the data have a spatial origin, where distances in each variable have the same meaning. Note that the `"separate"` option is not available when `"method"` is set to `"df"`.
- test a logical flag controlling the production of a formal test, using the reference model as the null hypothesis. Default value: `TRUE`.
- theta the horizontal rotation (in degrees) of perspective plots of estimates in the form of surfaces. Default value: `-30`.
- verbose regulates the amount of messages and other output printed out. If `verbose=0` only errors produce messages; if `verbose=1` (default value) warnings and the more relevant numerical output are printed; if `verbose=2` more messages and more numerical output are printed.
- xlab the label attached to the x-axis.
- xlim the range of the horizontal axis of the plot. This argument does not apply with three-dimensional data.
- yht the upper limit of the vertical axis in a plot of a one-dimensional density estimate. The lower limit is always set to 0. This argument does not apply with two- or three-dimensional data.
- ylab the label attached to the y-axis.
- ylim the range of the vertical axis of the plot. This argument does not apply with three-dimensional data.
- zlab the label attached to the z-axis (three-dimensional plots only).
- zlim the range of the vertical axis when estimates are displayed as perspective plots.

## Details

The parameter values set via a call to this function will remain in effect for the rest of the session, affecting the subsequent behaviour of the other functions of the `sm` library for which the given parameters are relevant.

This offers the possibility of customizing the functioning of the `sm` library, for instance by insertions of appropriate calls to `sm.options` in a load hook for package **sm**.

**Value**

a list with the updated values of the parameters. If the argument list is not empty, the returned list is invisible.

**Examples**

```
## Not run:
sm.options(poly.index = 0)
# subsequent regression estimations will be performed using local means
# instead of local regression
#
sm.options(describe = FALSE)
# turns off typing documentation files of data loaded by `sm.script`
# (works from command-line)
#
## End(Not run)
```

---

sm.poisson	<i>Nonparametric Poisson regression</i>
------------	---

---

**Description**

This function estimates the regression curve using the local likelihood approach for a vector of Poisson observations and an associated vector of covariate values.

**Usage**

```
sm.poisson(x, y, h, ...)
```

**Arguments**

x	vector of the covariate values
y	vector of the response values; they must be nonnegative integers.
h	the smoothing parameter; it must be positive.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: <code>add</code> , <code>col</code> , <code>display</code> , <code>eval.points</code> , <code>nbins</code> , <code>ngrid</code> , <code>pch</code> , <code>xlab</code> , <code>ylab</code> ; see the documentation of <a href="#">sm.options</a> for their description.

**Details**

see Sections 3.4 and 5.4 of the reference below.

**Value**

A list containing vectors with the evaluation points, the corresponding probability estimates, the linear predictors, the upper and lower points of the variability bands and the standard errors on the linear predictor scale.

**Side Effects**

graphical output will be produced, depending on the value of the `display` option.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

`sm.binomial`, `sm.binomial.bootstrap`, `binning`, `glm`

**Examples**

```
provide.data(muscle)
TypeI <- TypeI.R+ TypeI.P+TypeI.B
sm.poisson(x=log(TypeI), y=TypeII, h=0.25, display="se")
sm.poisson(x=log(TypeI), y=TypeII, h=0.75, col=2, add=TRUE)
```

---

```
sm.poisson.bootstrap
```

*Bootstrap goodness-of-fit test for a Poisson regression model*

---

**Description**

This function is associated with `sm.poisson` for the underlying fitting procedure. It performs a Pseudo-Likelihood Ratio Test for the goodness-of-fit of a standard parametric Poisson regression of specified `degree` in the covariate `x`.

**Usage**

```
sm.poisson.bootstrap(x, y, h, degree = 1,
                    fixed.disp = FALSE, intercept = TRUE, ...)
```

**Arguments**

<code>x</code>	vector of the covariate values
<code>y</code>	vector of the response values; they must be nonnegative integers.
<code>h</code>	the smoothing parameter; it must be positive.
<code>degree</code>	specifies the degree of the fitted polynomial in <code>x</code> on the logit scale (default=1).

<code>fixed.disp</code>	if TRUE, the dispersion parameter is kept at value 1 across the simulated samples, otherwise the dispersion parameter estimated from the sample is used to generate samples with that dispersion parameter (default=FALSE).
<code>intercept</code>	TRUE (default) if an intercept is to be included in the fitted model.
<code>...</code>	additional parameters passed to <code>sm.poisson</code> .

### Details

see Section 5.4 of the reference below.

### Value

a list containing the observed value of the Pseudo-Likelihood Ratio Test statistic, its observed p-value as estimated via the bootstrap method, and the vector of estimated dispersion parameters when this value is not forced to be 1.

### Side Effects

Graphical output representing the bootstrap samples is produced on the current graphical device. The estimated dispersion parameter, the value of the test statistic and the observed significance level are printed.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

`sm.poisson`, `sm.binomial.bootstrap`

### Examples

```
## takes a while: extend sm.script(muscle)
provide.data(muscle, options=list(describe=FALSE))
TypeI <- TypeI.P + TypeI.R + TypeI.B
sm.poisson.bootstrap(log(TypeI), TypeII, h = 0.5)
```

---

`sm.regression`      *Nonparametric regression with one or two covariates.*

---

### Description

This function creates a nonparametric regression estimate from data consisting of a single response variable and one or two covariates. In two dimensions a perspective, image (`image`), contour (`slice`) or `rgl` plot of the estimated regression surface is produced. A number of other features of the construction of the estimate, and of its display, can be controlled.

If the `rpanel` package is available, an interactive panel can be activated to control various features of the plot.

**Usage**

```
sm.regression(x, y, h, design.mat = NA, model = "none", weights = NA,
             group = NA, ...)
```

**Arguments**

<code>x</code>	a vector, or two-column matrix, of covariate values.
<code>y</code>	a vector of reponses.
<code>h</code>	a vector of length 1 or 2 giving the smoothing parameter. A normal kernel function is used and <code>h</code> is its standard deviation.
<code>design.mat</code>	the design matrix used to produce <code>y</code> when these are assumed to be the residuals from a linear model.
<code>model</code>	a character variable which defines a reference model. The values "none", "no effect" and "linear" and are possible.
<code>weights</code>	a vector which allows the kernel functions associated with the observations to take different weights. This is useful, in particular, when different observations have different precisions. The normal usage of this parameter is to associate observations with frequencies; if the <code>weights</code> are not integers, they are converted to integers, but in this case the standard errors and tests which are computed cannot be considered. This argument applies only to the case of one covariate. Use of this parameter is incompatible with <code>binning</code> ; hence <code>nbins</code> must then be set to 0 or left at its default value <code>NA</code> .
<code>group</code>	a vector of groups indicators (numeric or character values) or a factor
<code>...</code>	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function; those relevant for this function are the following: <code>display</code> , <code>hmult</code> , <code>h.weights</code> , <code>poly.index</code> , <code>band</code> , <code>add</code> , <code>ngrid</code> , <code>eval.points</code> , <code>se</code> , <code>se.breaks</code> , <code>xlab</code> , <code>ylab</code> , <code>zlab</code> , <code>hull</code> , <code>panel</code> , <code>panel.plot</code> , <code>lty</code> , <code>col</code> , <code>col.band</code> , <code>col.mesh</code> , <code>col.points</code> , <code>col.palette</code> ; see the documentation of <code>sm.options</code> for their description.

**Details**

When `display` is set to "persp" or "rgl", a number of graphical options are available. By setting the `col` parameter to "height" or "se", the surface will be painted by colours to reinforce the perception of height or indicate the relative sizes of the standard errors respectively. When `model` is not "none", the colour coding refers to the number of standard errors which separate the smooth regression surface and the nominated model at each position. The parameter "se.breaks", whose default value is `c(-3, -2, 3, 3)` can then be used to set the colour ranges. In this case, `col.palette` must be set to a list of colours whose length is one greater than the length of the cut-points in "se.breaks". If this is not the case, the default colour palette `rev(rainbow(length(opt$se.breaks) + 1, start = 0/6, end = 4/6))`.

If the argument `col` is not set then surface painting will be determined by the setting of `se`. If neither is set then colour painting will be activated by default if `model != "none"`. (In this latter case, the argument `band`, retained from earlier versions for compatibility, will also be examined.)

When `display` is set to `"rgl"`, some additional parameters can be used to control details of the plot. Transparency can be set by `alpha`, which lies between 0 and 1. When `alpha` is set to a vector of length two, the first component refers to the surface panels and the second to the surface mesh. Setting a component of `alpha` to 0 will remove the corresponding feature from the plot. `col.mesh`, whose valid values match those of `col`, controls the colour of the surface mesh. The logical parameter `lit` has the same meaning as in the `rgl` package; see `material3d`.

When `panel` is set to `"TRUE"`, an interactive control panel is created if the `rpanel` package is available.

If a covariate is on a cyclical scale, this can be incorporated by setting the `period` argument to a vector (of length 1 or 2) whose components give the values of the periods, or `NA` if the covariate is not periodic.

See Chapters 3, 4 and 5 of the first reference below for the details of the construction of the estimate and its standard error. The second reference gives further details and examples of surface painting.

### Value

a list containing the values of the estimate at the evaluation points, the smoothing parameter and the smoothing parameter weights. If a reference model has been specified and `test` set to `TRUE`, then the p-value of the test is also returned. When there is only one covariate, the weights associated with different observations, an estimate of the error standard deviation and the standard error of the estimate are also returned. If a reference model has been specified, this standard error refers to the comparison between the estimate and the reference model, and the values defining the reference model are also returned. If an `rgl` display is used, then the indices of the surface and lines used to create the display are returned.

### Side Effects

a plot on the current graphical device is produced, unless the option `display="none"` is set.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

Bowman, A.W. (2006). Comparing nonparametric surfaces. *Statistical Modelling*, 6, 279-299.

### See Also

[hcv](#), [sm](#), [sm.ancova](#), [sm.binomial](#), [sm.poisson](#), [sm.regression.autocor](#), [sm.survival](#), [sm.options](#), [sm.surface3d](#)

### Examples

```
provide.data(trawl)
Zone92 <- (Year == 0 & Zone == 1)
Position <- cbind(Longitude - 143, Latitude)
dimnames(Position)[[2]][1] <- "Longitude - 143"

par(mfrow = c(2, 2))
sm.regression(Longitude, Score1, method = "aicc", col = "red",
```

```

    model = "linear")
sm.regression(Position[Zone92, ], Score1[Zone92], display = "image",
  theta = 120)
sm.regression(Position[Zone92, ], Score1[Zone92], df = 12, col = "se",
  theta = 120)
sm.regression(Position[Zone92, ], Score1[Zone92], df = 12, col = "se",
  model = "linear", theta = 120)
par(mfrow = c(1, 1))

# sm.regression(Position[Zone92, 2:1], Score1[Zone92], display = "rgl", df = 12)
# sm.regression(Position[Zone92, 2:1], Score1[Zone92], display = "rgl", df = 12,
#   alpha = c(0.9, 1), col = "se", model = "linear")

# sm.regression(Position[Zone92, 1], Score1[Zone92], panel = TRUE)
# sm.regression(Position[Zone92, ], Score1[Zone92], panel = TRUE)
# sm.regression(Position[Zone92, ], Score1[Zone92], panel = TRUE, display = "rgl")

```

---

```
sm.regression.autocor
```

*Nonparametric regression with autocorrelated errors*

---

## Description

This function estimates nonparametrically the regression function of  $y$  on  $x$  when the error terms are serially correlated.

## Usage

```
sm.regression.autocor(x = 1:n, y, h.first, minh, maxh, method = "direct", ...)
```

## Arguments

<code>y</code>	vector of the response values
<code>h.first</code>	the smoothing parameter used for the initial smoothing stage.
<code>x</code>	vector of the covariate values; if unset, it is assumed to be <code>1:length(y)</code> .
<code>minh</code>	the minimum value of the interval where the optimal smoothing parameter is searched for (default is 0.5).
<code>maxh</code>	the maximum value of the interval where the optimal smoothing parameter is searched for (default is 10).
<code>method</code>	character value which specifies the optimality criterium adopted; possible values are "no.cor", "direct" (default), and "indirect".
<code>...</code>	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function. Those relevant for this function are the following: <code>ngrid</code> , <code>display</code> ; see the documentation of <code>sm.options</code> for their description.

**Details**

see Section 7.5 of the reference below.

**Value**

a list as returned from `sm.regression` called with the new value of smoothing parameter, with an additional term `$aux` added which contains the initial value `h.first`, the estimated curve using `h.first`, the autocorrelation function of the residuals from the initial fit, and the residuals.

**Side Effects**

a new suggested value for `h` is printed; also, if the parameter `display` is not equal to "none", graphical output is produced on the current graphical device.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[sm.regression](#), [sm.autoregression](#)

---

sm.rm

*Nonparametric analysis of repeated measurements data*

---

**Description**

This function estimates nonparametrically the mean profile from a matrix `y` which is assumed to contain repeated measurements (i.e. longitudinal data) from a set of individuals.

**Usage**

```
sm.rm(Time, y, minh = 0.1, maxh = 2, optimize = FALSE,
      rice.display = FALSE, ...)
```

**Arguments**

<code>y</code>	matrix containing the values of the response variable, with rows associated to individuals and columns associated to observation times.
<code>Time</code>	a vector containing the observation times of the response variable, assumed to be the same for all individuals of matrix <code>y</code> . If <code>Time</code> is not given, this is assumed to be <code>1:ncol(y)</code> .
<code>minh</code>	the minimum value of the interval where the optimal value of the smoothing parameter is searched according to the modified Rice criterion. See reference below for details.
<code>maxh</code>	the maximum value of the above interval.

<code>optimize</code>	Logical value, default is <code>optimize=FALSE</code> . If <code>optimize=TRUE</code> , then a full optimization is performed after searching the interval <code>(minh, maxh)</code> using the optimizer <code>optim</code> .
<code>rice.display</code>	If this set to <code>TRUE</code> (default is <code>FALSE</code> ), a plot is produced of the curve representing the modified Rice criterion for bandwidth selection. See reference below for details.
<code>...</code>	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function; those relevant for this function are the following: <ul style="list-style-type: none"> <li><code>add</code> logical value, default is <code>add=FALSE</code>. If <code>add=TRUE</code> and <code>display</code> is not set to <code>"none"</code>, then graphical output added to the existing plot, rather than starting a new one.</li> <li><code>display</code> character value controlling the amount of graphical output of the estimated regression curve. It has the same meaning as in <code>sm.regression</code>. Default value is <code>display="lines"</code>.</li> <li><code>ngrid</code> the number of divisions of the above interval to be considered. Default: <code>ngrid=20</code>.</li> <li><code>poly.index</code> overall degree of locally-fitted polynomial, as used by <code>sm.regression</code>. Default: <code>ngrid=1</code>.</li> </ul>

### Details

see Section 7.4 of the reference below.

### Value

a list containing the returned value produced by `sm.regression` when smoothing the mean response value at each given observation time, with an extra component `$aux` added to the list. This additional component is a list itself containing the mean value at each observation time, the residual variance of the residuals from the estimated regression curve, the autocorrelation function of the residuals, and the value `h` of the chosen smoothing parameter.

### Side Effects

if the parameter `display` is not set to `"none"`, a plot of the estimated regression curve is produced; other aspects are controlled by the optional parameters (`...`). If `rice.display=TRUE`, a plot of the modified Rice criterion is shown.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

[sm.regression](#), [sm.regression.autocor](#), [optim](#)

**Examples**

```

provide.data(citrate, options=list(describe=FALSE))
provide.data(dogs, options=list(describe=FALSE))
a <- sm.rm(y=citrate, display.rice=TRUE)
#
Time <- seq(1,13,by=2)
gr1 <- as.matrix(dogs[dogs$Group==1,2:8])
plot(c(1,13), c(3,6),xlab="time", ylab="potassium", type="n")
sm1 <- sm.rm(Time, gr1, display="se", add=TRUE)

```

---

sm.script

*Running a script associated to the sm library*


---

**Description**

This is a utility function to run scripts, usually those associated with the book described below.

**Usage**

```
sm.script(name, path)
```

**Arguments**

name	the name of the file containing the code; a .q suffix will be appended to the name. If name is missing, a list of the scripts associated to the sm library will be displayed.
path	the name of the path where to look for the script. If path is missing, it is assumed to be the appropriate location for the scripts of the sm library.

**Details**

This utility allows the illustrations of the reference below to be reproduced easily, since each of them has an associated script. The display of the script file itself is controlled by the setting of the logical variable `show.script` in `sm.options`. This is set to `TRUE` by default.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[sm](#),

**Examples**

```

sm.script()
sm.script(speed)

```

---

sm.sigma	<i>Estimation of the error standard deviation in nonparametric regression.</i>
----------	--

---

### Description

This function estimates the error standard deviation in nonparametric regression with one or two covariates.

### Usage

```
sm.sigma(x, y, rawdata = NA, weights = rep(1, length(y)),
         diff.ord = 2, ci = FALSE, model = "none", h = NA, ...)
```

### Arguments

x	a vector or two-column matrix of covariate values.
y	a vector of responses.
rawdata	a list containing the output from a binning operation. This argument is used by <code>sm.regression</code> and it need not be set for direct calls of the function.
weights	a list of frequencies associated with binned data. This argument is used by <code>sm.regression</code> and it need not be set for direct calls of the function.
diff.ord	an integer value which determines first (1) or second (2) differencing in the estimation of sigma.
ci	a logical value which controls whether a confidence interval is produced.
model	a character variable. If this is set to "constant" then a test of constant variance over the covariates is performed (only in the case of two covariates)
h	a vector of length two defining a smoothing parameter to be used in the test of constant variance.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function; the only one relevant for this function is <code>nbins</code> .

### Details

see the reference below.

### Value

a list containing the estimate and, in the two covariate case, a matrix which can be used by the function `sm.sigma2.compare`, pseudo-residuals and, if appropriate, a confidence interval and a p-value for the test of constant variance.

**Side Effects**

none.

**References**

Bock, M., Bowman, A.W. & Ismail, B. (2007). Estimation and inference for error variance in bivariate nonparametric regression. *Statistics & Computing*, to appear.

**See Also**

[sm.sigma2.compare](#)

**Examples**

```
data(airquality)
attach(airquality)
x     <- cbind(Wind, Temp)
y     <- Ozone^(1/3)
group <- (Solar.R < 200)
sig1 <- sm.sigma(x[ group, ], y[ group], ci = TRUE)
sig2 <- sm.sigma(x[!group, ], y[!group], ci = TRUE)
print(c(sig1$estimate, sig1$ci))
print(c(sig2$estimate, sig2$ci))
print(sm.sigma(x[ group, ], y[ group], model = "constant", h = c(3, 5))$p)
print(sm.sigma(x[!group, ], y[!group], model = "constant", h = c(3, 5))$p)
print(sm.sigma2.compare(x[group, ], y[group], x[!group, ], y[!group]))
```

---

sm.sigma2.compare *Comparison across two groups of the error standard deviation in non-parametric regression with two covariates.*

---

**Description**

This function compares across two groups, in a hypothesis test, the error standard deviation in nonparametric regression with two covariates.

**Usage**

```
sm.sigma2.compare(x1, y1, x2, y2)
```

**Arguments**

x1	a two-column matrix of covariate values for group 1.
y1	a vector of responses for group 1.
x2	a two-column matrix of covariate values for group 2.
y2	a vector of responses for group 2.

**Details**

see the reference below.

**Value**

a p-value for the test of equality of standard deviations.

**Side Effects**

none.

**References**

Bock, M., Bowman, A.W. & Ismail, B. (2007). Estimation and inference for error variance in bivariate nonparametric regression. *Statistics & Computing*, to appear.

**See Also**

[sm.sigma](#)

**Examples**

```
data(airquality)
attach(airquality)
x      <- cbind(Wind, Temp)
y      <- Ozone^(1/3)
group <- (Solar.R < 200)
sig1 <- sm.sigma(x[ group, ], y[ group], ci = TRUE)
sig2 <- sm.sigma(x[!group, ], y[!group], ci = TRUE)
print(c(sig1$estimate, sig1$ci))
print(c(sig2$estimate, sig2$ci))
print(sm.sigma(x[ group, ], y[ group], model = "constant", h = c(3, 5))$p)
print(sm.sigma(x[!group, ], y[!group], model = "constant", h = c(3, 5))$p)
print(sm.sigma2.compare(x[group, ], y[group], x[!group, ], y[!group]))
```

---

sm.sphere

*Nonparametric density estimation for spherical data.*


---

**Description**

This function creates a density estimate from data which can be viewed as lying on the surface of a sphere. Directional data form a principal example. The data are displayed in spherical form and a density estimate may be superimposed. The angle of view may be altered. An interactive panel is available to control some features of the estimate and the display. Only modest amounts of data may be used. The limit will depend on the memory available.

**Usage**

```
sm.sphere(lat, long, kappa = 20, hidden = FALSE, sphim = FALSE,
          addpoints = FALSE, ...)
```

**Arguments**

lat	a vector giving the latitude component of the data in degrees from the equator.
long	a vector giving the longitude component of the data in degrees east.
kappa	the smoothing parameter used to construct the density estimate. The kernel function is a Fisher distribution and kappa is its scale parameter. Larger values of kappa will produce smaller amounts of smoothing.
hidden	a logical value which, when set to TRUE, will display the points which lie on the rear side of the displayed sphere. This argument will be ignored if sphim is set to TRUE.
sphim	a logical value which controls whether a density estimate is constructed and displayed on the sphere in image form.
addpoints	a logical value which controls whether the data points are added to the plot of the density estimate.
...	arguments for <a href="#">sm.options</a> .

**Details**

see Section 1.5 of the reference below.

**Value**

a list containing the value of the smoothing parameter and the rotation angles of the displayed plot.

**Side Effects**

none.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[sm.density](#)

**Examples**

```
lat <- rnorm(50, 10, 15)
long <- c(rnorm(25, 300, 15), rnorm(25, 240, 15))
par(mfrow=c(1,2))
sm.sphere(lat, long)
sm.sphere(lat, long, sphim=TRUE, kappa=15)
par(mfrow=c(1,1))
```

---

sm.surface3d      *Adding a regression surface to an rgl plot.*

---

### Description

This function adds a regression surface, defined by a matrix of heights at a regular grid of values of two covariates, to an `rgl` plot. Missing values can be accommodated.

### Usage

```
sm.surface3d(eval.points, surf, scaling,
             col = "green", col.mesh = "black",
             alpha = 0.7, alpha.mesh = 1, lit = TRUE, ...)
```

### Arguments

<code>eval.points</code>	a two-column matrix defining the marginal grids of covariate values.
<code>surf</code>	a matrix of heights corresponding to the grid of covariate values. NAs are allowed.
<code>scaling</code>	a function to define the scaling for the <code>rgl</code> plot. This function is returned by an initial call to <code>rp.plot3d</code> in the <code>rpanel</code> package.
<code>col</code>	the colour of the surface. If <code>col</code> is set to a single value, this is replicated across the two components. However, a matrix of values corresponding to the entries of <code>surf</code> can also be supplied.
<code>col.mesh</code>	the colour of the surface mesh. If <code>col.mesh</code> is set to a single value, this is replicated across the two components. However, a matrix of values corresponding to the entries of <code>surf</code> can also be supplied.
<code>alpha</code>	the transparency of the filled triangles defining the surface. Setting this to 0 will remove the filled triangles from the plot.
<code>alpha.mesh</code>	the transparency of the lines drawn across the regular grid of covariate values. Setting this to 0 will remove the lines from the plot.
<code>lit</code>	a logical variable which controls whether the <code>rgl</code> plot is lit or not.
<code>...</code>	other optional parameters which are passed to <code>material3d</code> in the <code>rgl</code> package.

### Details

the principal motivation for this function is that it can handle missing data in regression surfaces. In particular, it can be used to plot the results of applying `sm.regression`. In addition, the function can be used to build up more complex plots by adding successive surfaces.

### Value

a vector of length 2 containing the ids of the filled surface and lines added to the `rgl` plot.

**Side Effects**

a surface is added to the rgl plot.

**See Also**

[sm.regression](#)

**Examples**

```
provide.data(trawl)
Zone93 <- (Year == 1 & Zone == 1)
Position <- cbind(Longitude - 143, Latitude)
model1 <- sm.regression(Position[Zone93,], Score1[Zone93],
  h= c(0.1, 0.1), display = "rgl", xlab="Longitude - 143")
model2 <- sm.regression(Position[Zone93,], Score1[Zone93],
  h= c(0.2, 0.2), display = "none")
sm.surface3d(model2$eval.points, model2$est, model1$scaling, col = "red")
```

---

sm.survival

*Nonparametric regression with survival data.*

---

**Description**

This function creates a smooth, nonparametric estimate of the quantile of the distribution of survival data as a function of a single covariate. A weighted product-limit estimate of the survivor function is obtained by smoothing across the covariate scale. A small amount of smoothing is then also applied across the survival time scale in order to achieve a smooth estimate of the quantile.

**Usage**

```
sm.survival(x, y, status, h , hv = 0.05, p = 0.5, status.code = 1, ...)
```

**Arguments**

x	a vector of covariate values.
y	a vector of survival times.
status	an indicator of a complete survival time or a censored value. The value of <code>status.code</code> defines a complete survival time.
h	the smoothing parameter applied to the covariate scale. A normal kernel function is used and <code>h</code> is its standard deviation.
hv	a smoothing parameter applied to the weighted to the product-limit estimate derived from the smoothing procedure in the covariate scale. This ensures that a smooth estimate is obtained.
p	the quantile to be estimated at each covariate value.
status.code	the value of <code>status</code> which defines a complete survival time.

... other optional parameters are passed to the `sm.options` function, through a mechanism which limits their effect only to this call of the function; those relevant for this function are `add`, `eval.points`, `ngrid`, `display`, `xlab`, `ylab`, `lty`; see the documentation of `sm.options` for their description.

### Details

see Section 3.5 of the reference below.

### Value

a list containing the values of the estimate at the evaluation points and the values of the smoothing parameters for the covariate and survival time scales.

### Side Effects

a plot on the current graphical device is produced, unless the option `display="none"` is set.

### References

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

### See Also

`sm.regression`, `sm.options`

### Examples

```
x <- runif(50, 0, 10)
y <- rexp(50, 2)
z <- rexp(50, 1)
status <- rep(1, 50)
status[z<y] <- 0
y <- pmin(z, y)
sm.survival(x, y, status, h=2)
```

---

sm.ts.pdf

*Nonparametric density estimation of stationary time series data*

---

### Description

This function estimates the density function of a time series `x`, assumed to be stationary. The univariate marginal density is estimated in all cases; bivariate densities of pairs of lagged values are estimated depending on the parameter `lags`.

### Usage

```
sm.ts.pdf(x, h = hnorm(x), lags, maxlag = 1, ask = TRUE)
```

**Arguments**

x	a vector containing a time series
h	bandwidth
lags	for each value, k say, in the vector lags a density estimate is produced of the joint distribution of the pair $(x(t-k), x(t))$ .
maxlag	if lags is not given, it is assigned the value 1:maxlag (default=1).
ask	if ask=TRUE, the program pauses after each plot, until <Enter> is pressed.

**Details**

see Section 7.2 of the reference below.

**Value**

a list of two elements, containing the outcome of the estimation of the marginal density and the last bivariate density, as produced by [sm.density](#).

**Side Effects**

plots are produced on the current graphical device.

**References**

Bowman, A.W. and Azzalini, A. (1997). *Applied Smoothing Techniques for Data Analysis: the Kernel Approach with S-Plus Illustrations*. Oxford University Press, Oxford.

**See Also**

[sm.density](#), [sm.autoregression](#)

**Examples**

```
data(geyser)
a <- sm.ts.pdf(geyser$duration, lags=1:2)
```

---

sm.variogram	<i>A test of spatial independence based on smoothing an empirical variogram.</i>
--------------	--

---

**Description**

This function constructs an empirical variogram, using the robust form of construction based on square-root absolute value differences of the data. Nonparametric regression is used to assess whether the mean value of the empirical variogram changes with distance. A hypothesis test and a graphical reference band are both available, under the assumption that the errors in the data are approximately normally distributed.

**Usage**

```
sm.variogram(x, y, h, ...)
```

**Arguments**

x	a vector or two-column matrix of spatial location values.
y	a vector of responses observed at the spatial locations.
h	a smoothing parameter to be used on the distance scale. A normal kernel function is used and h is its standard deviation. However, if this argument is omitted h will be selected by an approximate degrees of freedom criterion, controlled by the df parameter. See <code>sm.options</code> for details.
...	other optional parameters are passed to the <code>sm.options</code> function, through a mechanism which limits their effect only to this call of the function; those relevant for this function are <code>add</code> , <code>eval.points</code> , <code>ngrid</code> , <code>se</code> , <code>xlab</code> , <code>ylab</code> , <code>xlim</code> , <code>ylim</code> , <code>lty</code> ; see the documentation of <code>sm.options</code> for their description. See the details section below for a discussion of the <code>display</code> and <code>se</code> parameters in this setting.

**Details**

The reference below describes the statistical methods used in the function.

The `display` argument has a special meaning for this function. Its default value is "binned", which plots the binned version of the empirical variogram. As usual, the value "none" will suppress the graphical display. Any other value will lead to a plot of the individual differences between all observations. This will lead to a very large number of plotted points, unless the dataset is small.

Setting `se = TRUE` produces standard errors for the binned values of the empirical variogram. They are constructed to account for the correlation which arises from the differencing involved in the variogram. However, the calculations assume spatial independence and they therefore simply give an indication of the variability under this assumption.

**Value**

a list containing the raw (`sqrtdiff`, `distance`) and binned (`sqrtdiff.mean`, `distance.mean`) differences and distances, the frequencies of the bins (`weights`), the values of the estimate (`estimate`) at the evaluation points (`eval.points`) the value of the smoothing parameter (`h`) used, an indicator (`ibin`) of the bin in which the distance between each pair of observations was placed and the indices (`ipair`) of the original observations used to construct each pair. If a test of spatial independence has been performed, the p-value (`p`) is returned. If `se` was set to `TRUE` then the standard errors (`se`) of the binned values, under the assumption of spatial independence, are returned. If a reference band has been requested, the standard error (`se.band`) of the difference between the smooth estimate and the mean of all the data points is also returned.

**Side Effects**

a plot on the current graphical device is produced, unless the option `display="none"` is set.

**References**

Dibiasi, A. and Bowman, A.W. (2001). On the use of the variogram for checking independence in a Gaussian spatial process. *Biometrics*, 57, 211-218.

**See Also**

[sm.regression](#), [sm.options](#)

**Examples**

```
provide.data(coalash)
Position <- cbind(East, North)
sm.options(df = 6)

par(mfrow=c(2,2))
sm.variogram(Position, Percent, band = FALSE)
sm.variogram(Position, Percent)
sm.variogram(East,      Percent)
sm.variogram(North,    Percent)
par(mfrow=c(1,1))
```

# Index

## \*Topic **datasets**

geyser, 3

## \*Topic **htest**

sm.binomial.bootstrap, 20

sm.poisson.bootstrap, 34

## \*Topic **misc**

pause, 12

## \*Topic **models**

sm.binomial, 19

sm.binomial.bootstrap, 20

sm.poisson.bootstrap, 34

## \*Topic **nonparametric**

binning, 2

h.select, 4

hcv, 5

hnorm, 7

hsj, 8

nise, 9

nmise, 10

nnbr, 11

sig.trace, 13

sm, 14

sm.ancova, 16

sm.autoregression, 18

sm.binomial, 19

sm.binomial.bootstrap, 20

sm.density, 21

sm.density.compare, 23

sm.options, 28

sm.poisson, 33

sm.poisson.bootstrap, 34

sm.regression, 35

sm.regression.autocor, 38

sm.rm, 39

sm.sigma, 42

sm.sigma2.compare, 43

sm.sphere, 44

sm.surface3d, 46

sm.survival, 47

sm.ts.pdf, 48

## \*Topic **regression**

h.select, 4

sm.discontinuity, 24

sm.monotonicity, 27

sm.regression, 35

sm.regression.autocor, 38

sm.surface3d, 46

sm.variogram, 49

## \*Topic **smooth**

h.select, 4

hcv, 5

hnorm, 7

hsj, 8

nise, 9

nmise, 10

nnbr, 11

sig.trace, 13

sm, 14

sm.ancova, 16

sm.autoregression, 18

sm.binomial, 19

sm.binomial.bootstrap, 20

sm.density, 21

sm.density.compare, 23

sm.discontinuity, 24

sm.monotonicity, 27

sm.options, 28

sm.poisson, 33

sm.poisson.bootstrap, 34

sm.regression, 35

sm.regression.autocor, 38

sm.rm, 39

sm.sigma, 42

sm.sigma2.compare, 43

sm.sphere, 44

sm.surface3d, 46

sm.survival, 47

sm.ts.pdf, 48

- sm.variogram, 49
- \*Topic **spatial**
  - sm.variogram, 49
- \*Topic **survival**
  - sm.survival, 47
- \*Topic **ts**
  - sm.autoregression, 18
  - sm.ts.pdf, 48
- \*Topic **utilities**
  - provide.data, 12
  - sm.script, 41
- attach, 13
- binning, 2, 20, 34
- cut, 3
- data.frame, 13
- faithful, 3
- geyser, 3
- glm, 20, 34
- h.select, 4, 7–9, 23
- hcv, 5, 5, 8, 9, 23, 37
- hnorm, 5, 7, 7, 9, 23
- hsj, 5, 7, 8, 8, 23
- nise, 9, 10, 23
- nmise, 10, 10, 23
- nnbr, 11
- optim, 40
- pause, 12
- provide.data, 12
- sig.trace, 13
- sm, 3, 5, 13, 14, 23, 37, 41
- sm.ancova, 4, 16, 24, 37
- sm.autoregression, 18, 39, 49
- sm.binomial, 3, 19, 21, 34, 37
- sm.binomial.bootstrap, 20, 20, 34, 35
- sm.density, 3, 21, 24, 45, 49
- sm.density.compare, 4, 17, 23
- sm.discontinuity, 24
- sm.monotonicity, 27
- sm.options, 4, 6, 12, 13, 17, 19, 20, 22–27, 28, 28, 33, 36–38, 45, 48, 50, 51
- sm.poisson, 3, 20, 33, 35, 37
- sm.poisson.bootstrap, 21, 34
- sm.regression, 3, 16–18, 23, 26, 28, 35, 39, 40, 47, 48, 51
- sm.regression.autocor, 37, 38, 40
- sm.rm, 39
- sm.script, 41
- sm.sigma, 42, 44
- sm.sigma2.compare, 43, 43
- sm.sphere, 23, 44
- sm.surface3d, 37, 46
- sm.survival, 37, 47
- sm.ts.pdf, 18, 48
- sm.variogram, 49
- table, 3