Package: sm (via r-universe)

October 28, 2026

Type Package

Title Smoothing Methods for Nonparametric Regression and Density Estimation

Version 2.2-6.0

Date 2024-02-17

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.bowman@glasgow.ac.uk>

Depends R (>= 3.1.0)

Suggests rgl, misc3d, interp, gam, tkrplot, rpanel (>= 1.1-4), tcltk

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

License GPL (>= 2)

LazyData TRUE

Encoding UTF-8

RoxygenNote 7.2.1

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2024-02-18 02:34:27 UTC

RemoteUrl https://github.com/cran/sm

RemoteRef HEAD

RemoteSha 403652c39a9f6c19daaaf0fd7d910740d0440d0c

Contents

aircraft . . . 3
airpc . . . 4
binning . . . 4
birth . . . 5
bissell . . . 6
bonions . . . 6
britpts . . . 7
citrate . . . 7
coalash . . . 7
dogs . . . 8
follicle . . . 8
geys3d . . . 9
geyser . . . 9
h.select . . . 10
hcv . . . 11
hnorm . . . 13
hsj . . . 14
lcancer . . . 15
mackerel . . . 16
magrem . . . 16
mildew . . . 17
mosses . . . 17
muscle . . . 19
nile . . . 20
nise . . . 20
nmise . . . 21
nnbr . . . 22
pause . . . 23
poles . . . 23
provide.data . . . 23
radioc . . . 25
sig.trace . . . 25
sm . . . 26
sm.ancova . . . 28
sm.autoregression . . . 29
sm.binomial . . . 31
sm.binomial.bootstrap . . . 32
sm.density . . . 33
sm.density.compare . . . 35
sm.discontinuity . . . 37
sm.monotonicity . . . 39
sm.options . . . 41
sm.pca . . . 45
sm.poisson . . . 48
sm.poisson.bootstrap . . . 50
sm.regression . . . 51

sm.regression.autocor 54
sm.rm 55
sm.script 56
sm.sigma 57
sm.sigma2.compare 59
sm.sphere 60
sm.surface3d 61
sm.survival 63
sm.ts.pdf 64
sm.variogram 65
smacker 69
stanford 70
tephra 70
trawl 71
trout 71
wonions 72
worm 72

Index **73**

`aircraft` *These data record six characteristics of aircraft designs which appeared during the twentieth century*

Description

These data record six characteristics of aircraft designs which appeared during the twentieth century.

The variables are:

`Yr`	year of first manufacture
`Period`	a code to indicate one of three broad time periods
`Power`	total engine power (kW)
`Span`	wing span (m)
`Length`	length (m)
`Weight`	maximum take-off weight (kg)
`Speed`	maximum speed (km/h)
`Range`	range (km)

Source: The data were collected by P. Saviotti and are described in detail in Saviotti (1996), "Technological Evolution, Variety and Economy", Edward Elgar: Cheltenham.

airpc — *These data list the first two principal component scores from the aircraft data, which record six characteristics of aircraft designs throughout the twentieth century*

Description

These data list the first two principal component scores from the aircraft data, which record six characteristics of aircraft designs throughout the twentieth century.

The variables are:

`Comp.1:`	first principal component score
`Comp.2:`	second principal component score
`Yr:`	year of first manufacture
`Period:`	a code to indicate one of three broad time periods

The data were collected by P. Saviotti and are described in detail in Saviotti (1996), "Technological Evolution, Variety and Economy", Edward Elgar: Cheltenham.

binning — *Construct frequency table from raw data in 1, 2 or 3 dimensions.*

Description

Given a vector, or a matrix with 1, 2 or 3 columns, this function constructs a frequency table associated with appropriate intervals covering the range of `x`.

Usage

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

Arguments

`x`	a vector or a matrix with either one, two or three columns, containing the original data.
`y`	a vector of data, for example response data, associated with the data in `x`.
`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 each axis. If `nbins` is not supplied, a value is computed as `round(log(n)/log(2) + 1)`.

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 frequecies, 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)
```

---

birth *Low birthweight in babies*

Description

The data refer to a study on low birthweight in babies, defined as "less than 2500 grams", and related factors.

The variables are:

Smoke	indicator of whether the mother is a smoker

`Lwt`	last menstrual weight of the mother
`Low`	indicator variable of low weight

Hosmer & Lemeshow (1989). Applied Logistic Regression. Wiley, NY. The original source contains additional variables; see Appendix 1 of Hosmer & Lemeshow for a full list of the data, pp.29-30 and p.92 for additional information.

`bissell` *Flaws in cloth*

Description

These data refer to the length and the observed number of flaws in rolls of cloth.

The variables are:

`Length`	length of each roll (m)
`Flaws`	number of flaws detected

Source: Bissell (1972). A negative binomial model with varying element sizes. Biometrika 59, 435-41.

`bonions` *Yield-density relationship for Brown Imperial Spanish onions*

Description

These data were collected in a study of the relationship between the yield of Brown Imperial Spanish onion plants and the density of planting.

The variables are:

`Density`	density of planting (plants/m^2)
`Yield`	yield (g/plant)
`Locality`	a code to indicate Mount Gambier (1) or Uraidla (2)

The data were collected by I.S.Rogers (South Australian Dept. of Agriculture & Fisheries). They are listed in Ratkowsky (1983), Nonlinear Regression Modeling. Dekker, New York.

britpts *Coastline of the UK and Ireland*

Description

These data provide the co-ordinates of a set of points lying on the coastlines of the UK and Ireland.

The variables are:

britlong	longitude
brotlat	latitude

citrate *The relationship between plasma citrate and carbohydrate metabolites*

Description

These data were collected in an experiment to study the relationship between possible daily rhythms of plasma citrate and daily rhythms of carbohydrate metabolites during feeding with a citrate-poor diet. During the experiment, plasma citrate concentrations were determined for each of 10 subjects at 14 successive time points during the day. The measurements covered the period 8a.m. to 9p.m. at hourly intervals. Meals were given at 8a.m., noon and 5p.m.

The variables are denoted by C08, ..., C21 and refer to plasma citrate measurements at the indicated hours.

Anderson,A.H., Jensen,E.B. & Schou,G.(1981). Two-way analysis of variance with correlated errors. Int.Stat.Rev. 49,153-67.

The data were taken from a report by T.T.Nielsen, N.S.Sorensen and E.B.Jensen.

coalash *Coal ash in mining samples*

Description

These data record the percentage of coal ash found in mining samples originally reported by Gomez and Hazen (1970) and subsequently used by Cressie (1993).

The variables are:

East	a code for east-west direction
North	a code for north-south direction
Percent	the percentage of coalash

Cressie, N. (1993). Statistics for Spatial Data, revised edition. New York: Wiley.

Gomez, M. and Hazen, K. (1970). Evaluating sulfur and ash distribution in coal seams by statistical response surface regression analysis. Report RI 7377, U.S. Bureau of Mines, Washington, D. C.

dogs *Coronary sinus potassium in dogs*

Description

Measurements of coronary sinus potassium (mil equivalent per litre) were made at (1,3,5,7,9,11,13) minutes after coronary occlusion in a number of different dogs. There are four treatment groups (group 1 is the control). The paper by Grizzle and Allen provides a full description of the treatments. A few subjects develop ventricular fibrillation (see paper for details).

The variables are:

`Group`	a treatment group indicator
`P1, P3, P5, P7, P9, P11, P13`	measurements at indicated times

J.E.Grizzle & D.M.Allen (1969). Analysis of growth and dose response curves. Biometrics vol.25, p.357-381

follicle *Ovarian follicle counts*

Description

These data record the number of ovarian follicles, on a log scale, counted from sectioned ovaries of women of various ages.

The variables are:

`Age`	age of the woman
`Count`	follicle count
`Source`	an indicator of the source of the data

The data were reported by Block (1952; 1953), Richardson et al. (1987) and A Gougeon. They are analysed by Faddy & Gosden (1996) and Faddy & Jones (1997).

geys3d *Duration and the time between eruptions for the Old Faithful Geyser*

Description

These data document the duration of eruptions, and the time between eruptions, for the Old Faithful Geyser in Yellowstone National Park.

The variables are:

`Waiting`	the waiting time before each eruption (minutes)
`Next.waiting`	the waiting time following each eruption (minutes)
`Duration`	the length of an eruption (minutes)

The data were collected by by the Park Geologist, R.A.Hutchinson. An earlier set of data is reported in Weisberg (1990), Applied Linear Regression, Wiley, New York. The later set, used here, was reported by Azzalini & Bowman (1990), "A look at some data on the Old Faithful Geyser", Applied Statistics 39, 357-65.

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.

`duration`	numeric	Eruption time, in minutes
`waiting`	numeric	Waiting time before the eruption, in minutes

Note that in versions of the `sm` package before 2.2-5 the `waiting` variable was incorrectly described as 'waiting time to next eruption'.

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 binning; hence nbins 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 sm.options 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 method, which specifies the method of smoothing parameter selection, df, which specifies the approximate degrees of freedom associated with the selected smoothing parameter, and structure.2d 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 sm.options.

Details

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

If the sample size is large, binning will be employed. In the case of method = "cv" the answer will therefore be different from that obtained through the function hcv where binning is not used.

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-validatory 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 sm.options function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: h.weights, ngrid, display, add; see the documentation of sm.options 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. When the sample size is large hcv uses the raw data while h.select(x, method="cv") uses binning. The latter is likely to produce a more stable choice for h.

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)
```

---

lcancer *Spatial positions of cases of laryngeal cancer*

Description

These data record the spatial positions of cases of laryngeal cancer in the North-West of England between 1974 and 1983, together with the positions of a number of lung cancer patients who were used as controls. The data have been adjusted to preserve anonymity.

The variables are:

`Easting`	a west-east grid reference
`Northing`	a north-south grid reference
`Cancer`	an indicator of laryngeal (1) or lung (2) cancer

Source: Bailey & Gatrell (1995). Interactive Spatial Data Analysis. Longman Scientific and Technical, Harlow. A more extensive set of data is analysed in Kelsall & Diggle, kernel estimation of relative risk, Bernoulli 1, 3-16.

mackerel *The abundance of mackerel eggs*

Description

These data record the abundance of mackerel eggs off the coast of north-western Europe, from a multi-country survey in 1992.

The variables are:

`Density`	egg density
`mack.lat`	latitude of sampling position
`mack.long`	longitude of sampling position
`mack.depth`	bottom depth
`Temperature`	surface temperature
`Salinity`	salinity

Background to the survey and the data are provided by Watson et al. (1992), Priede and Watson (1993) and Priede et al (1995). Borchers et al (1997) describe an analysis of the data.

magrem *Magnetic remanence*

Description

These data record measurements of magnetic remanence in specimens of Precambrian volcanics.

The variables are:

`maglong`	directional component on a longitude scale
`maglat`	directional component on a latitude scale

Schmidt & Embleton (1985) J.Geophys.Res. 90 (B4), 2967-2984.

The data are also listed in Fisher, Lewis & Embleton (1987), Statistical Analysis of Spherical Data, Cambridge University Press, Cambridge, dataset B6.

mildew *Mildew control*

Description

The data refer to study of mildew control sponsored by Bainbridge, Jenkyn and Dyke at Rothamsted Experimental Station. There were four treatments, one of which was a control. There were 36 adjacent plots, with an extra plot at each end. Nine blocks were created by grouping the plots in fours.

The variables are:

`t1, t2, t3`	indicators of the four treatment groups
`p1, ..., p8`	indicators of the nine blocks
`Yield`	tons of grain per hectare

Draper & Guttman (1980). Incorporating overlap effects from neighbouring units into response surface models. Applied Statistics 29, 128-134.

mosses *Heavy metals in mosses in Galicia.*

Description

Mosses are used as a means of measuring levels of heavy metal concentrations in the atmosphere, since most of the nutrient uptake of the mosses is from the air. This technique for large-scale monitoring of long-range transport processes has been used in Galicia, in North-West Spain, over the last decade, as described by Fernandez et al. (2005). In 2006, in both March and September, measurements of different metals were collected at 148 points lying almost in a regular grid over the region with 15 km spacing in north-south and east-west directions. According to the ecologists' expertise, the period between the two samples, passing from a humid to a dry season, is enough time to guarantee the independence of the observed processes.

The dataset consists of a list with six components

`loc.m`	a two-column matrix containing grid locations of the March monitoring sites
`loc.s`	a two-column matrix containing grid locations of the September monitoring sites
`Co.m`	cobalt concentration (log scale) in March
`Co.s`	cobalt concentration (log scale) in September
`Hg.m`	mercury concentration (log scale) in March
`Hg.s`	mercury concentration (log scale) in September

Source: The data were kindly made available by the Ecotoxicology and Vegetal Ecophysiology research group in the University of Santiago de Compostela.

References

Fernandez, J., Real, C., Couto, J., Aboal, J., Carballeira, A. (2005). The effect of sampling design on extensive biomonitoring surveys of air pollution. Science of the Total Environment, 337, 11-21.

Examples

```
## Not run:
# Comparison of Co in March and September

with(mosses, {

   nbins <- 12
   vgm.m <- sm.variogram(loc.m, Co.m, nbins = nbins, original.scale = TRUE,
                         ylim = c(0, 1.5))
   vgm.s <- sm.variogram(loc.s, Co.s, nbins = nbins, original.scale = TRUE,
                         add = TRUE, col.points = "blue")

   trns <- function(x) (x / 0.977741)^4
   del <- 1000
   plot(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean), type = "b",
         ylim = c(0, 1.5), xlab = "Distance", ylab = "Semi-variogram")
   points(vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean), type = "b",
         col = "blue", pch = 2, lty = 2)

   plot(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean), type = "b",
         ylim = c(0, 1.5), xlab = "Distance", ylab = "Semi-variogram")
   points(vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean), type = "b",
         col = "blue", pch = 2, lty = 2)
   segments(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean - 2 * vgm.m$se),
         vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean + 2 * vgm.m$se))
   segments(vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean - 2 * vgm.s$se),
         vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean + 2 * vgm.s$se),
         col = "blue", lty = 2)

   mn <- (vgm.m$sqrtdiff.mean + vgm.s$sqrtdiff.mean) / 2
   se <- sqrt(vgm.m$se^2 + vgm.s$se^2)
   plot(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean), type = "n",
        ylim = c(0, 1.5), xlab = "Distance", ylab = "Semi-variogram")
   polygon(c(vgm.m$distance.mean, rev(vgm.m$distance.mean)),
        c(trns(mn - se), rev(trns(mn + se))),
        border = NA, col = "lightblue")
   points(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean))
   points(vgm.s$distance.mean, trns(vgm.s$sqrtdiff.mean), col = "blue", pch = 2)

   vgm1 <- sm.variogram(loc.m, Co.m, nbins = nbins, varmat = TRUE,
                        display = "none")
   vgm2 <- sm.variogram(loc.s, Co.s, nbins = nbins, varmat = TRUE,
                        display = "none")

   nbin  <- length(vgm1$distance.mean)
   vdiff <- vgm1$sqrtdiff.mean - vgm2$sqrtdiff.mean
   tstat <- c(vdiff %*% solve(vgm1$V + vgm2$V) %*% vdiff)
```

```
   pval  <- 1 - pchisq(tstat, nbin)
   print(pval)
})

# Assessing isotropy for Hg in March

with(mosses, {
   sm.variogram(loc.m, Hg.m, model = "isotropic")
})

# Assessing stationarity for Hg in September

with(mosses, {
   vgm.sty <- sm.variogram(loc.s, Hg.s, model = "stationary")
   i <- 1
   image(vgm.sty$eval.points[[1]], vgm.sty$eval.points[[2]], vgm.sty$estimate[ , , i],
         col = topo.colors(20))
   contour(vgm.sty$eval.points[[1]], vgm.sty$eval.points[[2]], vgm.sty$sdiff[ , , i],
         col = "red", add = TRUE)
})


## End(Not run)
```

`muscle` *Rat skeletal muscles*

Description

The data refer to the counts of fibres of different types in groups of fibres taken from rat skeletal muscles.

The variables are:

`row.labels`	indicators of the four treatment groups
`TypeI.R`	indicators of the nine blocks
`TypeI.P`	tons of grain per hectare
`TypeI.B`	tons of grain per hectare
`TypeII`	tons of grain per hectare

Hand, Daly, Lunn, McConway and Ostrowski (1994). A handbook of Small Data Sets. Chapman & Hall: London. The data were collected by M. Khan and M. Khan.

nile *Water level of the River Nile*

Description

These data record historical data on the water level of the River Nile.

The variables are:

`Volume`	Annual volume of the Nile River (discharge at Aswan, 10^8 m^3)
`Year`	1871-1970

Cobb, G. (1978). The problem of the Nile: conditional solution to a change-point problem. Biometrika 65, 243-251.

nise *Integrated squared error between a density estimate and a Normal density*

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, ...)
```

Arguments

`y`	a vector of data.
`...`	further arguments which are to be passed to `sm.options`.

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 *mean integrated squared error for density estimation with normal data*

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 *Pause before continuing execution*

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()
```

poles *Positions of the south pole*

Description

These data refer to positions of the south pole determined from the palaeomagnetic study of New Caledonian laterites.

The variables are:

```
Latitude
 Longitude
```

The data were collected by Falvey and Musgrave. They are listed in Fisher, Lewis & Embleton (1987), Statistical Analysis of Spherical Data, Cambridge University Press, Cambridge, dataset B1.

provide.data *Making data available as data.frame*

Description

This function is no longer available in the sm package. It should be replaced by the use of `attach`, if necessary. Each dataset now also has its own help file.

It was a utility function, widely used in the scripts accompanying the book described below. The function provided access to the dataset identified by `name`. For flexibility, the datasets were 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

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

radioc *Radiocarbon in Irish oak*

Description

These data record high precision measurements of radiocarbon on Irish oak, used to construct a calibration curve.

The variables are:

`Rc.age`	age predicted from the radiocarbon dating process
`Precision`	a measure of precision of the dating process
`Cal.age`	true calendar age

Pearson & Qua (1993). High precision 14C measurement of Irish oaks to show the natural 14C variations from AD 1840 - 5000 BC: a correction. Radiocarbon 35, 105-123.

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, ...)
```

Arguments

`expn`	an S-Plus expression which should define the hypothesis test to be performed, with the value of the smoothing parameter `h` omitted from the function call.
`hvec`	a vector of smoothing parameters for which the test will be performed.
`...`	further arguments which will be passed to `sm.options`.

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`, `sm.variogram`, `sm.pca`. 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 used of the `gam` package.

Requirements

R version >= 3.1.0. The `gam` package is used by some of the scripts launched via `sm.script`, but it is not used by the functions of this package.

Version

This is version 2.2. The most recent version of the package can be obtained from the CRAN archive.

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 (School of Mathematics and Statistics, University of Glasgow, UK) and Adelchi Azzalini (Dept Statistical Sciences, University of Padua, Italy). Please send comments, error reports, etc. to the authors.

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

`x`	a vector or two-column matrix of covariate values.
`y`	a vector of response values.
`group`	a vector of group indicators.
`h`	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 `sm.options` will be applied.
`model`	a character variable which defines the reference model. The values `"none"`, `"equal"` and `"parallel"` are possible.
`h.alpha`	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.
`weights`	case weights; see the documentation of `sm.regression` 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. If a reference band has been requested, the upper and lower boundaries and their common evaluation points are 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

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

```
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 `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: `add`, `col`, `display`, `eval.points`, `nbins`, `ngrid`, `pch`, `xlab`, `ylab`; see the documentation of `sm.options` 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)

with(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.binomial 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

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.
N	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

x	a vector, or a matrix with two or three columns, containing the data.
h	a vector of length one, two or three, defining the smoothing parameter. A normal kernel function is used and h is its standard deviation. If this parameter is omitted, a normal optimal smoothing parameter is used.
model	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.
weights	a vector of integers representing frequencies of individual observations. Use of this parameter is incompatible with binning; hence nbins 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 sm.options function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: hmult, h.weights, band, add, lty, display, props, xlab, ylab, zlab, xlim, ylim, yht, nbins, ngrid, eval.points, panel, positive, delta, theta, phi; see the documentation of sm.options 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 permutation test of equality.

Usage

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

Arguments

`x`	a vector of data.
`group`	a vector of group labels. If this is not already a factor it will be converted to a factor.
`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 `sm.options` function, through a mechanism which limits their effect only to this call of the function. Relevant parameters for this function are: `method`, `df`, `band`, `test`, `nboot`, plus those controlling graphical display (unless `display = "none"` is set) such as `col`, `col.band`, `lty` and `lwd`; see the documentation of `sm.options` for their description. The parameter `nboot` controls the number of permutations used in the permutation test.

Details

For a general description of the methods involved, see Section 6.2 of the reference below.

The colours and linetypes of the density estimates are set by `col` and `lty` which can be passed as additional arguments. By default these are set to `1 + 1:ngroup`, where `ngroup` is the number of groups represented in the `group` variable.

Value

A list is returned containing:

`estimate`	a matrix whose rows contain the estimates for each group.
`eval.points`	the grid of common evaluation points for the estimates.
`h`	the common smoothing parameter used in the construction of the estimates.
`levels`	the levels of the group factor.
`col`, `lty`, `lwd`	plotting details which can be useful in constructing a legend for the plot; see the examples below.

When `"model"` is set to `"equal"`, the output list also contains the p-value (`p`) of the test.

When `band = TRUE`, and there are only two groups to compare, the output list also contains the upper (`upper`) and lower (`lower`) edges 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)

comp <- sm.density.compare(y, g, model = "equal")
legend("topleft", comp$levels, col = comp$col, lty = comp$lty, lwd = comp$lwd)
```

`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(0.75)`, 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.

Currently duplicated rows of `x` cause a difficulty in the two covariate case. Duplicated rows should be removed.

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))

with(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)
})

with(trawl, {
   Position  <- cbind(Longitude, Latitude)
   ind <- (Longitude < 143.8)
   # Remove a repeated point which causes difficulty with sm.discontinuity
   ind[54] <- FALSE
   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 sm.options and h.select for details.
type	an indicator of whether the response variable is on a "continuous" or "binomial" scale.
...	other optional parameters are passed to the sm.options function, through a mechanism which limits their effect only to this call of the function; some of those relevant for this function are add, ngrid, xlab, ylab, xlim, ylim, lty, col; see the documentation of sm.options 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

```
 ## Not run:
#     Radiocarbon dating data

with(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

with(birth, {
   sm.monotonicity(Lwt[Smoke == "N"], Low[Smoke == "N"],
          type = "binomial")
})

## End(Not run)
```

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

Description

This function 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. A list may be given as the only argument, or any number of arguments may be in the `name=value` form. If no arguments are specified then the function returns the current settings of all the arguments.

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. The valid names of the arguments are given below.

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 a value, or a vector of values, lying between 0 and 1, which controls the transparency of the surfaces used to construct an `rgl` plot, when this form if display is requested. In the case of regression with two covariates, a single value for `alpha` is used and the default is `0.7`. In the case of density estimation with three variables, `alpha` should be set to a vector whose length matches the number of contours to be drawn (specified by the `props` argument). In this case the default is `seq(1, 0.5, length = length(props))`.

alpha.mesh a 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 latter 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`. In the case of density estimation with three variables, `col` can be set to a vector whose length matches the number of contours to be drawn (specified by the `props` argument).

col.band the colour used for the reference band when a `model` is specified in regression with one covariate or in density estimation with a single variable. 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 optimal for the normal distribution. Other possibilities are `"cv"` for cross-validation and, for one-dimensional data only, `"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. Periodic smoothing is implemented by local mean estimation, using a von Mises kernel function. 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

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 package; 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.

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.pca *Smooth principal components analysis*

Description

This function calculates principal components in a manner which changes smoothly with a covariate. The smooth eigenvalues and eigenvector loadings can be plotted. A permutation test of equality of the components, both eigenvalues and eigenvectors, can be carried out.

Usage

```
sm.pca(x, Y, h, cor = TRUE, nperm = 100, pc = 1, ...)
```

Arguments

x	either a vector of covariate values or a list object which is the output of a previous call to sm.pca. In the latter case, previously computed information is used to create plots and tests and the arguments Y, h, cor and nperm are not required.
Y	a matrix of responses whose rows correspond to the covariate values.
h	the smoothing parameter which controls the smoothness of estimation with respect to the covariate x.
cor	a logical value indicating whether the correlation, rather than covariance, matrix should be used.
nperm	the number of permutations used in the permutation test and graphical reference band.
pc	an integer value indicating the component to be plotted against the covariate.
...	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 (here set to "eigevalues" or "eigenvectors") ngrid, xlab; see the documentation of sm.options for their description.

Details

Several further arguments may be set and these are passed to sm.options. Relevant arguments for this function are display ("eigenvalues", "eigenvectors"), ngrid and df. See link{sm.options} for further details.

The smoothing is performed by the local constant kernel method and the smoothing parameter corresponds to the standard deviation of a normal kernel function. If h is left unspecified then it is selected to correspond to the degrees of freedom set by the parameter df.

The reference band for a constant eigenvalue is constructed from the upper and lower pointwise 2.5 percentiles of the smooth eigenvalue curves from the data with permuted covariate values. The p-value compares the observed value of the difference between the smoothed and constant eigenvalues, summed over the covariate grid in eval.points, with the values generated from the permuted data.

In the eigenvector case, a reference band is computed from the percentiles of the curves from the permuted data, for each of the loadings. In order to plot all the loadings curves simultaneously, the locations where each curve lies inside its respective reference band are indicated by pale colour. The p-value compares the observed value of 1 - sum(e*e0)^2, where e and e0 are the eigenvectors under the smooth and constant scenarios (summed over the covariate grid), with the values generated from the permuted data. This test statistic differs from the one described in the Miller and Bowman (2012) reference below. It has been used as it conveniently handles the arbitrary sign of principal components.

When some components explain similar proportions of variance, the eigenvalues and eigenvectors can easily interchange, causing apparent sharp changes in the eigenvalue and eigenvector curves. It is difficult to track the components to avoid this.

Value

a list with the following components:

xgrid a vector of values on the covariate scale at which the smooth estimates are constructed.

evals a matrix whose columns give the smooth eigenvalues for each component at the covariate values.

evecs a three-dimensional array whose third dimension corresponds to the covariate values and whose second dimension indexes the smooth components.

mhat a matrix whose columns give the estimated smooth means for each dimension of `Y` at the covariate values.

var.explained a matrix whose rows give the proportions of variance explained by each component at each covariate value.

xlab the label attached to the x-axis.

h the smoothing parameter used.

x the covariate values, after removal of missing cases.

Y the matrix of response values, after removal of missing cases.

cor a logical indicator of whether the correlation, rather than covariance, matrix is used in the construction of the eigenvalues and eigenvectors.

When a test or reference band is computed, the list has the additional components:

nperm the number of permutations used.

evals.perm the eigenvalues computed from the permuted data.

evecs.perm the eigenvectors computed from the permuted data.

When display contains `"eigenvalues"` or `"eigenvectors"`, the list has the additional components:

p.values the p-value for a test of constant eigenvalue for the component identified by `pc`.

p.vectors the p-value for a test of constant eigenvectors for the component identified by `pc`.

When display contains `"eigenvalues"`, the list has the additional component:

band a matrix whose two columns contain the boundaries of a reference band which indicates where the smooth eigenvalue curve should like if the hypothesis of no change in the eigenvalues with the covariate is correct.

When display contains `"eigenvectors"`, the list has the additional components:

xgrid.plot a vector of values used for plotting the smooth eigenvectors.

evecs.plot a matrix whose rows contain the smooth eigenvectors at each value of `xgrid.plot`.

evecs.plot a matrix whose columns contain the colours for the line segments in each smooth eigenvector component.

Side Effects

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

References

Miller, C. and Bowman, A.W. (2012). Smooth principal components for investigating changes in covariances over time. *Applied Statistics* **61**, 693–714.

See Also

sm.regression, sm.options

Examples

```
## Not run:
Y    <- log(as.matrix(aircraft[ , -(1:2)]))
year <- aircraft$Yr
h    <- h.select(year, Y[ , 1], method = "df", df = 4)
spca <- sm.pca(year, Y, h, display = "none")
sm.pca(year, Y, h, display = "eigenvalues")
sm.pca(year, Y, h, display = "eigenvectors", ylim = c(-1, 1))

# The following code shows how the plots can be redrawn from the returned object

spca <- sm.pca(year, Y, h, display = "eigenvalues")
spca <- sm.pca(year, Y, h, display = "eigenvectors", ylim = c(-1, 1))

with(spca, {
   ylim <- range(evals[ , 1], band)
   plot(xgrid, evals[ , 1], type = "n", ylab = "Variance", ylim = ylim)
   polygon(c(xgrid, rev(xgrid)), c(band[ , 1], rev(band[ , 2])),
           col = "lightgreen", border = NA)
   lines(xgrid, evals[ , 1], col = "red")
})

with(spca, {
   pc <- 1
   plot(range(xgrid.plot), range(evecs.plot), type = "n",
        xlab = "x", ylab = "PC loadings")
   for (i in 1:ncol(Y))
      segments(xgrid.plot[-length(xgrid.plot)],
               evecs.plot[-nrow(evecs.plot), i],
               xgrid.plot[-1], evecs.plot[-1, i],
               col = col.plot[ , i], lty = i)
})

## End(Not run)
```

---

sm.poisson *Nonparametric Poisson regression*

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 sm.options function, through a mechanism which limits their effect only to this call of the function. Those specifically relevant for this function are the following: add, col, display, eval.points, nbins, ngrid, pch, xlab, ylab; see the documentation of sm.options 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

```
with(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

`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.
`degree`	specifies the degree of the fitted polynomial in `x` on the logarithm 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`).
`intercept`	`TRUE` (default) if an intercept is to be included in the fitted model.
`...`	additional parameters passed to `sm.poisson`.

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)
with(muscle, {
   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

x	a vector, or two-column matrix, of covariate values.
y	a vector of responses.
h	a vector of length 1 or 2 giving the smoothing parameter. A normal kernel function is used and h is its standard deviation.
design.mat	the design matrix used to produce y when these are assumed to be the residuals from a linear model.
model	a character variable which defines a reference model. The settings "none", "no effect" and "linear" and all valid. Note that when a group argument is used then model should be set to "equal" or "parallel", as befits an analysis of covariance model.
weights	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 weights 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 binning; hence nbins 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. If this argument is used then the data are passed to the sm.ancova function. See the details of the model argument above in that 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, hmult, h.weights, poly.index, band, add, ngrid, eval.points, se, se.breaks, period, xlab, ylab, zlab, hull, panel, panel.plot, lty, col, col.band, col.mesh, col.points, col.palette; see the documentation of sm.options 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

```
with(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

`y`	vector of the response values
`h.first`	the smoothing parameter used for the initial smoothing stage.
`x`	vector of the covariate values; if unset, it is assumed to be `1:length(y)`.
`minh`	the minimum value of the interval where the optimal smoothing parameter is searched for (default is 0.5).
`maxh`	the maximum value of the interval where the optimal smoothing parameter is searched for (default is 10).
`method`	character value which specifies the optimality criterium adopted; possible values are `"no.cor"`, `"direct"` (default), and `"indirect"`.
`...`	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: `ngrid`, `display`; see the documentation of `sm.options` 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

y	matrix containing the values of the response variable, with rows associated to individuals and columns associated to observation times.
Time	a vector containing the observation times of the response variable, assumed to be the same for all individuals of matrix y. If Time is not given, this is assumed to be 1:ncol(y).
minh	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.
maxh	the maximum value of the above interval.
optimize	Logical value, default is optimize=FALSE. If optimize=TRUE, then a full optimization is performed after searching the interval (minh,maxh) using the optimizer optim.
rice.display	If this set to TRUE (default is FALSE), a plot is produced of the curve representing the modified Rice criterion for bandwidth selection. See reference below for details.
...	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:

add logical value, default is add=FALSE. If add=TRUE and display is not set to "none", then graphical output added to the existing plot, rather than starting a new one.

display character value controlling the amount of graphical output of the estimated regression curve. It has the same meaning as in sm.regression. Default value is display="lines".

ngrid the number of divisions of the above interval to be considered. Default: ngrid=20.

poly.index overall degree of locally-fitted polynomial, as used by sm.regression. Default: ngrid=1.

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

```
sm.rm(y=as.matrix(citrate), display.rice=TRUE)
#
with(dogs, {
   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 *Estimation of the error standard deviation in nonparametric regression.*

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 sm.regression 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 sm.regression 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 sm.options function, through a mechanism which limits their effect only to this call of the function; the only one relevant for this function is nbins.

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

```
## Not run:
with(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]))
})

## End(Not run)
```

`sm.sigma2.compare` *Comparison across two groups of the error standard deviation in nonparametric 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

```
## Not run:
with(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]))
})

## End(Not run)
```

---

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 sm.options.

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

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

Details

the principal motivation for this function is that is 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

```
with(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 `status.code` defines a complete survival time.
`h`	the smoothing parameter applied to the covariate scale. A normal kernel function is used and `h` 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 `status` 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

```
with(geyser, {
   sm.ts.pdf(geyser$duration, lags=1:2)
})
```

---

sm.variogram *Confidence intervals and tests based on smoothing an empirical variogram.*

Description

This function constructs an empirical variogram, using the robust form of construction based on square-root absolute value differences of the data. Flexible regression is used to assess a variety of questions about the structure of the data used to construct the variogram, including independence, isotropy and stationarity. Confidence bands for the underlying variogram, and reference bands for the independence, isotropy and stationarity models, can also be constructed under the assumption that the errors in the data are approximately normally distributed.

Usage

```
sm.variogram(x, y, h, df.se = "automatic", max.dist = NA, n.zero.dist = 1,
             original.scale = TRUE, varmat = FALSE, ...)
```

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 `sm.options` for details.

`df.se` the degrees of freedom used when smoothing the empirical variogram to estimate standard errors. The default value of "automatic" selects the degrees of smoothing described in the Bowman and Crujeiras (2013) reference below.

`max.dist` this can be used to constrain the distances used in constructing the variogram. The default is to use all distances.

`n.zero.dist` an integer value which sets the lower limit on the number of pairs of data points with distance zero (in other words the number of pairs whose positions are identical) to trigger a separate variogram bin for zero distance. Repeat observations at the same location can be informative about the variance of the underlying process. The default for `n.zero.dist` is 1, which will create a zero bin for any identical pairs. However, with small numbers of identical pairs the estimated variance associated with the zero bin may be large, so a higher setting of the `n.zero.dist` argument can be used to amalgamate the zero distances into the adjacent variogram bin.

This parameter has no effect when `model = "isotropic"` or `model = "stationary"` as the number of bins generated by the multiple dimensions becomes large in these cases, and so zero distances are not allocated a special bin.

`original.scale` a logical value which determines whether the plots are constructed on the original variogram scale (the default) or on the square-root absolute value scale on which the calculations are performed.

`varmat` a logical value which determines whether the variance matrix of the estimated variogram is returned.

`...` other optional parameters are passed to the `sm.options` function, through a mechanism which limits their effect only to this call of the function.

An important parameter here is `model` which, for `sm.variogram`, can be set to `"none"`, `"independent"`, `"isotropic"` or `"stationary"`. The latter two cases are appropriate only for 2-d data.

Other relevant parameters are `add`, `eval.points`, `ngrid`, `se`, `xlab`, `ylab`, `xlim`, `ylim`, `lty`; see the documentation of `sm.options` for their description. See the details section below for a discussion of the `display` and `se` parameters in this setting.

Details

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

Note that, apart from the simple case of the independence model, the calculations required are extensive and so the function can be slow.

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.

Value

A list with the following components:

`sqrtdiff`, `distance`
: the raw differences and distances

`sqrtdiff.mean`, `distance.mean`
: the binned differences and distances

`weights`
: the frequencies of the bins

`estimate`
: the values of the estimate at the evaluation points

`eval.points`
: the evaluation points

`h`
: the value of the smoothing parameter used

`ibin`
: an indicator of the bin in which the distance between each pair of observations was placed

`ipair`
: the indices of the original observations used to construct each pair

When `model = "isotropic"` or `model = "stationary"` the following components may also be returned, depending on the arguments passed in ... or the settings in `sm.options`:

`p`
: the p-value of the test

`se`
: the standard errors of the binned values (if the argument `se` was set to `TRUE`)

`se.band`
: when an independence model is examined, this gives the standard error of the difference between the smooth estimate and the mean of all the data points, if a reference band has been requested

`V`
: the variance matrix of the binned variogram. When `model` is set to `"isotropic"` or `"stationary"`, the variance matrix is computed under those assumptions.

`sdiff`
: the standardised difference between the estimate of the variogram and the reference model, evaluated at `eval.points`

`levels`
: the levels of standardised difference at which contours are drawn in the case of `model = "isotropy"`.

Side Effects

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

References

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

Bowman, A.W. and Crujeiras, R.M. (2013). Inference for variograms. *Computational Statistics and Data Analysis*, 66, 19-31.

See Also

`sm.regression`, `sm.options`

Examples

```
## Not run:
with(coalash, {
   Position <- cbind(East, North)
   sm.options(list(df = 6, se = TRUE))

   par(mfrow=c(2,2))
   sm.variogram(Position, Percent, original.scale = FALSE, se = FALSE)
   sm.variogram(Position, Percent, original.scale = FALSE)
   sm.variogram(Position, Percent, original.scale = FALSE, model = "independent")
   sm.variogram(East,     Percent, original.scale = FALSE, model = "independent")
   par(mfrow=c(1,1))
})

# Comparison of Co in March and September

with(mosses, {

   nbins <- 12
   vgm.m <- sm.variogram(loc.m, Co.m, nbins = nbins, original.scale = TRUE,
                         ylim = c(0, 1.5))
   vgm.s <- sm.variogram(loc.s, Co.s, nbins = nbins, original.scale = TRUE,
                         add = TRUE, col.points = "blue")

   trns <- function(x) (x / 0.977741)^4
   del <- 1000
   plot(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean), type = "b",
         ylim = c(0, 1.5), xlab = "Distance", ylab = "Semi-variogram")
   points(vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean), type = "b",
         col = "blue", pch = 2, lty = 2)

   plot(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean), type = "b",
         ylim = c(0, 1.5), xlab = "Distance", ylab = "Semi-variogram")
   points(vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean), type = "b",
         col = "blue", pch = 2, lty = 2)
   segments(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean - 2 * vgm.m$se),
         vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean + 2 * vgm.m$se))
   segments(vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean - 2 * vgm.s$se),
         vgm.s$distance.mean - del, trns(vgm.s$sqrtdiff.mean + 2 * vgm.s$se),
         col = "blue", lty = 2)

   mn <- (vgm.m$sqrtdiff.mean + vgm.s$sqrtdiff.mean) / 2
   se <- sqrt(vgm.m$se^2 + vgm.s$se^2)
   plot(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean), type = "n",
        ylim = c(0, 1.5), xlab = "Distance", ylab = "Semi-variogram")
   polygon(c(vgm.m$distance.mean, rev(vgm.m$distance.mean)),
        c(trns(mn - se), rev(trns(mn + se))),
        border = NA, col = "lightblue")
   points(vgm.m$distance.mean, trns(vgm.m$sqrtdiff.mean))
   points(vgm.s$distance.mean, trns(vgm.s$sqrtdiff.mean), col = "blue", pch = 2)

   vgm1 <- sm.variogram(loc.m, Co.m, nbins = nbins, varmat = TRUE,
```

```
                        display = "none")
   vgm2 <- sm.variogram(loc.s, Co.s, nbins = nbins, varmat = TRUE,
                        display = "none")

   nbin  <- length(vgm1$distance.mean)
   vdiff <- vgm1$sqrtdiff.mean - vgm2$sqrtdiff.mean
   tstat <- c(vdiff %*% solve(vgm1$V + vgm2$V) %*% vdiff)
   pval  <- 1 - pchisq(tstat, nbin)
   print(pval)
})

# Assessing isotropy for Hg in March

with(mosses, {
   sm.variogram(loc.m, Hg.m, model = "isotropic")
})

# Assessing stationarity for Hg in September

with(mosses, {
   vgm.sty <- sm.variogram(loc.s, Hg.s, model = "stationary")
   i <- 1
   image(vgm.sty$eval.points[[1]], vgm.sty$eval.points[[2]], vgm.sty$estimate[ , , i],
         col = topo.colors(20))
   contour(vgm.sty$eval.points[[1]], vgm.sty$eval.points[[2]], vgm.sty$sdiff[ , , i],
         col = "red", add = TRUE)
})


## End(Not run)
```

`smacker` *Mackerel data from a Spanish survey*

Description

These data were recorded by a Spanish survey, as part of a multi-country survey of the abundance of mackerel eggs off the coast of north-western Europe, in 1992.

The variables are:

`Density`	egg density
`smack.lat`	latitude of sampling position
`smack.long`	longitude of sampling position
`smack.depth`	bottom depth
`Temperature`	surface temperature
`D200`	distance from the 200m depth contour line

Background to the survey and the data are provided by Watson et al. (1992), Priede and Watson (1993) and Priede et al (1995). Borchers et al (1997) describe an analysis of the data.

stanford *Survival times from the Stanford Heart Transplant Study*

Description

These data refer to the survival times of patients of different ages from the Stanford Heart Transplant Study.

The variables are:

`Log.time`	egg density
`Status`	latitude of sampling position
`Age`	longitude of sampling position

Source: Miller & Halpern (1980). Regression with censored data. Biometrika 69, 521-531.

tephra *Tephra layer*

Description

These data record the percentages of aluminium oxide found in samples from a tephra layer resulting from a volcanic eruption in Iceland around 3500 years ago.

The variables are:

`Al2O3`	percentage of aluminium oxide

The data were collected by A.Dugmore. The geological background to the data is given by Dugmore et al (1992), Geochemical stability of finegrained silicic tephra in Iceland and Scotland, J.Quatern.Sci. 7, 173-83.

trawl *Trawl data from the Great Barrier Reef*

Description

These data refer to a survey of the fauna on the sea bed lying between the coast of northern Queensland and the Great Barrier Reef. The sampling region covered a zone which was closed to commercial fishing, as well as neighbouring zones where fishing was permitted.

The variables are:

`Zone`	an indicator for the closed (1) and open (0) zones
`Year`	an indicator of 1992 (0) or 1993 (1)
`Latitude`	latitude of the sampling position
`Longitude`	longitude of the sampling position
`Depth`	bottom depth
`Score1`	catch score 1
`Score2`	catch score 2

The details of the survey and an analysis of the data are provided by Poiner et al. (1997), The effects of prawn trawling in the far northern section of the Great Barrier Reef, CSIRO Division of Marine Research, Queensland Dept. of Primary Industries.

trout *Potassium cyanate and trout eggs*

Description

These data were collected in a toxocological experiment conducted at the University of Waterloo. Different concentrations of potassium cyanate were applied to vials of trout eggs. The eggs in half of the vials were allowed to water-harden before the toxicant was applied.

The variables are:

`Concentr`	toxicant concentration
`Trouts`	number of trout eggs
`Dead`	number of eggs which died
`Insert`	an indicator of whether the eggs were allowed to water-harden

Source: O'Hara Hines & Carter (1993). Improved added variable and partial residual plots for the detection of influential observations in generalized linear models. Applied Statistics 42, 3-20.

The data are also reported by Hand et al. (1994), A Handbook of Small Data Sets, data set no.418.

wonions *Yield-density relationship for White Imperial Spanish onion plants*

Description

These data were collected in a study of the relationship between the yield of White Imperial Spanish onion plants and the density of planting.

The variables are:

`Density`	density of planting (plants/m^2)
`Yield`	yield (g/plant)
`Locality`	a code to indicate Purnong Landing (1) or Virginia (2)

The data were collected by I.S.Rogers (South Australian Dept. of Agriculture & Fisheries). They are listed in Ratkowsky (1983), Nonlinear Regression Modeling. Dekker, New York.

worm *Human parasitic worm infections*

Description

These data record the occurrence of a human parasitic worm infection in residents of a rural community in China.

The variables are:

`Age`	age of the resident
`Infection`	presence (1) or absence (0) of infection
`Sex`	male (1) or female (2)

The background to the data, and an analysis, are described by Weidong et al. (1996), Ascaris, people and pigs in a rural community of Jiangxi province, China, Parasitology 113, 545-57.

Index

∗ **datasets**
geyser, 9
∗ **htest**
sm.binomial.bootstrap, 32
sm.poisson.bootstrap, 50
∗ **misc**
pause, 23
∗ **models**
sm.binomial, 31
sm.binomial.bootstrap, 32
sm.poisson.bootstrap, 50
∗ **nonparametric**
binning, 4
h.select, 10
hcv, 11
hnorm, 13
hsj, 14
nise, 20
nmise, 21
nnbr, 22
sig.trace, 25
sm, 26
sm.ancova, 28
sm.autoregression, 29
sm.binomial, 31
sm.binomial.bootstrap, 32
sm.density, 33
sm.density.compare, 35
sm.options, 41
sm.pca, 45
sm.poisson, 48
sm.poisson.bootstrap, 50
sm.regression, 51
sm.regression.autocor, 54
sm.rm, 55
sm.sigma, 57
sm.sigma2.compare, 59
sm.sphere, 60
sm.surface3d, 61
sm.survival, 63
sm.ts.pdf, 64
∗ **regression**
aircraft, 3
airpc, 4
birth, 5
bissell, 6
bonions, 6
britpts, 7
citrate, 7
coalash, 7
dogs, 8
follicle, 8
geys3d, 9
h.select, 10
lcancer, 15
mackerel, 16
magrem, 16
mildew, 17
mosses, 17
muscle, 19
nile, 20
poles, 23
radioc, 25
sm.discontinuity, 37
sm.monotonicity, 39
sm.regression, 51
sm.regression.autocor, 54
sm.surface3d, 61
sm.variogram, 65
smacker, 69
stanford, 70
tephra, 70
trawl, 71
trout, 71
wonions, 72
worm, 72
∗ **smooth**
aircraft, 3

airpc, 4
birth, 5
bissell, 6
bonions, 6
britpts, 7
citrate, 7
coalash, 7
dogs, 8
follicle, 8
geys3d, 9
h.select, 10
hcv, 11
hnorm, 13
hsj, 14
lcancer, 15
mackerel, 16
magrem, 16
mildew, 17
mosses, 17
muscle, 19
nile, 20
nise, 20
nmise, 21
nnbr, 22
poles, 23
radioc, 25
sig.trace, 25
sm, 26
sm.ancova, 28
sm.autoregression, 29
sm.binomial, 31
sm.binomial.bootstrap, 32
sm.density, 33
sm.density.compare, 35
sm.discontinuity, 37
sm.monotonicity, 39
sm.options, 41
sm.pca, 45
sm.poisson, 48
sm.poisson.bootstrap, 50
sm.regression, 51
sm.regression.autocor, 54
sm.rm, 55
sm.sigma, 57
sm.sigma2.compare, 59
sm.sphere, 60
sm.surface3d, 61
sm.survival, 63
sm.ts.pdf, 64
sm.variogram, 65
smacker, 69
stanford, 70
tephra, 70
trawl, 71
trout, 71
wonions, 72
worm, 72
∗ **spatial**
sm.variogram, 65
∗ **survival**
sm.survival, 63
∗ **ts**
sm.autoregression, 29
sm.ts.pdf, 64
∗ **utilities**
provide.data, 23
sm.script, 56

aircraft, 3
airpc, 4
attach, *24*

binning, 4, *32*, *49*
birth, 5
bissell, 6
bonions, 6
britpts, 7

citrate, 7
coalash, 7
cut, *5*

data.frame, *24*
dogs, 8

faithful, *10*
follicle, 8

geys3d, 9
geyser, 9
glm, *32*, *49*

h.select, 10, *13–15*, *34*
hcv, 11, *11*, *14, 15*, *34*, *53*
hnorm, *11*, 13, *13*, *15*, *34*
hsj, *11*, *13*, 14, *14*, *34*

lcancer, 15

mackerel, 16
magrem, 16
mildew, 17
mosses, 17
muscle, 19

nile, 20
nise, 20, *21*, *34*
nmise, 21, *21*, *34*
nnbr, 22

optim, *56*

pause, 23
poles, 23
provide.data, 23

radioc, 25

sig.trace, 25
sm, *5*, *11*, *24*, 26, *34*, *53*, *57*
sm.ancova, *10*, 28, *36*, *53*
sm.autoregression, 29, *55*, *65*
sm.binomial, *5*, 31, *32, 33*, *49*, *53*
sm.binomial.bootstrap, 32, *32*, *49*, *51*
sm.density, *5*, 33, *36*, *61*, *65*
sm.density.compare, *10*, *29*, 35
sm.discontinuity, 37
sm.monotonicity, 39
sm.options, *10*, *12*, *24*, *28, 29*, *31, 32*, *34–38*, *40*, 41, *46*, *48, 49*, *52–54*, *61*, *63, 64*, *66, 67*
sm.pca, 45
sm.poisson, *5*, *32*, 48, *50, 51*, *53*
sm.poisson.bootstrap, *33*, 50
sm.regression, *5*, *28–30*, *34*, *38*, *40*, *48*, 51, *55, 56*, *62*, *64*, *67*
sm.regression.autocor, *53*, 54, *56*
sm.rm, 55
sm.script, 56
sm.sigma, 57, *60*
sm.sigma2.compare, *58*, 59
sm.sphere, *34*, 60
sm.surface3d, *53*, 61
sm.survival, *53*, 63
sm.ts.pdf, *30*, 64
sm.variogram, 65
smacker, 69
stanford, 70

table, *5*
tephra, 70
trawl, 71
trout, 71

wonions, 72
worm, 72