Package ‘locfit’

February 20, 2015

Version 1.5-9.1
Title Local Regression, Likelihood and Density Estimation.
Date 2013-03-22
Author Catherine Loader
Maintainer Andy Liaw <andy.liaw@merck.com>
Description Local regression, likelihood and density estimation.
Depends R (>= 2.0.1)
Imports lattice
Suggests akima, gam
License GPL (>= 2)
Repository CRAN
Date/Publication 2013-04-20 09:01:16
NeedsCompilation yes

R topics documented:

aic .................................................. 4
aicplot ............................................. 4
ais .................................................. 5
ang .................................................. 6
bad .................................................. 7
border ............................................. 7
chemdiab ......................................... 8
claw54 ............................................ 8
cldem ............................................. 9
clttest .......................................... 9
cltrain .......................................... 10
co2 ................................................ 10
cp .................................................. 11
cpar .............................................. 11
cpplot .......................................... 12
crit .............................................. 13
# R topics documented:

- `dat` .......................................................... 14
- `density.lf` ................................................. 14
- `diab` ......................................................... 15
- `ethanol` .................................................... 16
- `expit` ......................................................... 16
- `fitted.locfit` ................................................ 17
- `formula.locfit` ............................................ 18
- `gam.lf` ....................................................... 18
- `gam.slist` .................................................. 19
- `gcv` .......................................................... 19
- `gcvplot` ...................................................... 20
- `geyser` ........................................................ 20
- `geyser.round` ............................................... 21
- `hatmatrix` ................................................... 22
- `heart` ......................................................... 22
- `insect` ....................................................... 23
- `iris` ........................................................... 23
- `kangaroo` ................................................... 24
- `kappa0` ....................................................... 25
- `kdeb` .......................................................... 26
- `km.mrl` ........................................................ 27
- `lcv` ............................................................. 28
- `lcvplot` ...................................................... 28
- `left` ............................................................ 29
- `lf` ............................................................... 30
- `lfeval` ........................................................ 31
- `lfgrid` ........................................................ 31
- `lfknots` ...................................................... 32
- `lflim` .......................................................... 32
- `lfmarg` ........................................................ 33
- `lines.locfit` ................................................ 34
- `livmet` ........................................................ 34
- `locfit` ......................................................... 35
- `locfit.censor` ............................................... 36
- `locfit.matrix` ............................................... 37
- `locfit.quasi` ................................................ 38
- `locfit.raw` ................................................... 39
- `locfit.robust` ............................................... 41
- `lp` .............................................................. 42
- `lscv` ........................................................... 43
- `lscv.exact` .................................................. 44
- `lscvplot` ..................................................... 44
- `mcycc` ........................................................ 45
- `mine` ............................................................ 46
- `mmsamp` ....................................................... 46
- `morths` ........................................................ 47
- `none` ............................................................ 47
- `penny` .......................................................... 48
R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>plot.eval</td>
<td>48</td>
</tr>
<tr>
<td>plot.gcvplot</td>
<td>49</td>
</tr>
<tr>
<td>plot.lfeval</td>
<td>50</td>
</tr>
<tr>
<td>plot.locfit</td>
<td>50</td>
</tr>
<tr>
<td>plot.locfit.1d</td>
<td>52</td>
</tr>
<tr>
<td>plot.locfit.2d</td>
<td>52</td>
</tr>
<tr>
<td>plot.locfit.3d</td>
<td>53</td>
</tr>
<tr>
<td>plot.preplot.locfit</td>
<td>54</td>
</tr>
<tr>
<td>plot.scb</td>
<td>54</td>
</tr>
<tr>
<td>plotbyfactor</td>
<td>55</td>
</tr>
<tr>
<td>points.locfit</td>
<td>56</td>
</tr>
<tr>
<td>predict.locfit</td>
<td>56</td>
</tr>
<tr>
<td>preplot.locfit</td>
<td>57</td>
</tr>
<tr>
<td>preplot.locfit.raw</td>
<td>58</td>
</tr>
<tr>
<td>print.gcvplot</td>
<td>59</td>
</tr>
<tr>
<td>print.lfeval</td>
<td>60</td>
</tr>
<tr>
<td>print.locfit</td>
<td>60</td>
</tr>
<tr>
<td>print.preplot.locfit</td>
<td>61</td>
</tr>
<tr>
<td>print.scb</td>
<td>61</td>
</tr>
<tr>
<td>print.summary.locfit</td>
<td>62</td>
</tr>
<tr>
<td>rbox</td>
<td>62</td>
</tr>
<tr>
<td>regband</td>
<td>63</td>
</tr>
<tr>
<td>residuals.locfit</td>
<td>64</td>
</tr>
<tr>
<td>right</td>
<td>64</td>
</tr>
<tr>
<td>rv</td>
<td>65</td>
</tr>
<tr>
<td>rva</td>
<td>66</td>
</tr>
<tr>
<td>scb</td>
<td>66</td>
</tr>
<tr>
<td>sjpi</td>
<td>67</td>
</tr>
<tr>
<td>smooth.lf</td>
<td>68</td>
</tr>
<tr>
<td>spence.15</td>
<td>69</td>
</tr>
<tr>
<td>spence.21</td>
<td>70</td>
</tr>
<tr>
<td>spencer</td>
<td>71</td>
</tr>
<tr>
<td>stamp</td>
<td>71</td>
</tr>
<tr>
<td>store</td>
<td>72</td>
</tr>
<tr>
<td>summary.gcvplot</td>
<td>72</td>
</tr>
<tr>
<td>summary.locfit</td>
<td>73</td>
</tr>
<tr>
<td>summary.preplot.locfit</td>
<td>74</td>
</tr>
<tr>
<td>trimod</td>
<td>74</td>
</tr>
<tr>
<td>xbar</td>
<td>75</td>
</tr>
</tbody>
</table>

Index 76
aic

Compute Akaike's Information Criterion.

Description

The calling sequence for aic matches those for the locfit or locfit.raw functions. The fit is not returned; instead, the returned object contains Akaike's information criterion for the fit.

The definition of AIC used here is -2*log-likelihood + pen*(fitted d.f.). For quasi-likelihood, and local regression, this assumes the scale parameter is one. Other scale parameters can effectively be used by changing the penalty.

The AIC score is exact (up to numerical roundoff) if the ev="data" argument is provided. Otherwise, the residual sum-of-squares and degrees of freedom are computed using locfit's standard interpolation based approximations.

Usage

aic(x, ..., pen=2)

Arguments

x model formula
... other arguments to locfit
pen penalty for the degrees of freedom term

See Also

locfit, locfit.raw, aicplot

aicplot

Compute an AIC plot.

Description

The aicplot function loops through calls to the aic function (and hence to locfit), using a different smoothing parameter for each call. The returned structure contains the AIC statistic for each fit, and can be used to produce an AIC plot.

Usage

aicplot(..., alpha)
ais

Arguments

... arguments to the \texttt{aic, locfit} functions.

\texttt{alpha} Matrix of smoothing parameters. The \texttt{aicplot} function loops through calls to \texttt{aic}, using each row of \texttt{alpha} as the smoothing parameter in turn. If \texttt{alpha} is provided as a vector, it will be converted to a one-column matrix, thus interpreting each component as a nearest neighbor smoothing parameter.

Value

An object with class "gcvplot", containing the smoothing parameters and AIC scores. The actual plot is produced using \texttt{plot.gcvplot}.

See Also

\texttt{locfit, locfit.raw, gcv, aic, plot.gcvplot}

Examples

\begin{verbatim}
data(morths)
plot(aicplot(deaths~age+weights, data=morths, family="binomial",
alpha=seq(0.2,1.0,by=0.05)))
\end{verbatim}

\begin{verbatim}
ais Australian Institute of Sport Dataset
\end{verbatim}

Description

The first two columns are the gender of the athlete and their sport. The remaining 11 columns are various measurements made on the athletes.

Usage

data(ais)

Format

A dataframe.

Source

Cook and Weisberg (1994).

References

**Description**

The `ang()` function is used in a locfit model formula to specify that a variable should be treated as an angular or periodic term. The `scale` argument is used to set the period.

`ang(x)` is equivalent to `lp(x, style="ang")`.

**Usage**

`ang(x,...)`

**Arguments**

- `x` numeric variable to be treated periodically.
- `...` Other arguments to `lp`.

**References**


**See Also**

`locfit`.

**Examples**

```r
# generate an x variable, and a response with period 0.2
x <- seq(0,1,length=200)
y <- sin(10*pi*x)+rnorm(200)/5

# compute the periodic local fit. Note the scale argument is period/(2*pi)
fit <- locfit(y~ang(x, scale=0.2/(2*pi)))

# plot the fit over a single period
plot(fit)

# plot the fit over the full range of the data
plot(fit,xlim=c(0,1))
```
**bad**

*Example dataset for bandwidth selection*

**Description**

Example dataset from Loader (1999).

**Usage**

data(bad)

**Format**

Data Frame with x and y variables.

**References**


---

**border**

*Cricket Batting Dataset*

**Description**

Scores in 265 innings for Australian batsman Allan Border.

**Usage**

data(border)

**Format**

A dataframe with day (decimalized); not out indicator and score. The not out indicator should be used as a censoring variable.

**Source**

Compiled from the Cricinfo archives.

**References**

**Chemical Diabetes Dataset**

**Description**
Numeric variables are *rw*, *fpg*, *ga*, *ina* and *sspg*. Classifier *cc* is the Diabetic type.

**Usage**
data(chemdiab)

**Format**
Data frame with five numeric measurements and categorical response.

**Source**
Reaven and Miller (1979).

**References**

---

**Claw Dataset**

**Description**
A random sample of size 54 from the claw density of Marron and Wand (1992), as used in Figure 10.5 of Loader (1999).

**Usage**
data(claw54)

**Format**
Numeric vector with length 54.

**Source**
Randomly generated.
References

cldem

Example data set for classification

Description
Observations from Figure 8.7 of Loader (1999).

Usage
data(cldem)

Format
Data Frame with x and y variables.

References

cldem

Test dataset for classification

Description
200 observations from a 2 population model. Under population 0, \( x_{1,i} \) has a standard normal distribution, and \( x_{2,i} = (2 - x_{1,i}^2 + z_i)/3 \), where \( z_i \) is also standard normal. Under population 1, \( x_{2,i} = -(2 - x_{1,i}^2 + z_i)/3 \). The optimal classification regions form a checkerboard pattern, with horizontal boundary at \( x_2 = 0 \), vertical boundaries at \( x_1 = \pm \sqrt{2} \).
This is the same model as the cltrain dataset.

Usage
data(cltest)

Format
Data Frame. Three variables x1, x2 and y. The latter indicates class membership.
cltrain

Training dataset for classification

Description

200 observations from a 2 population model. Under population 0, \( x_{1,i} \) has a standard normal distribution, and \( x_{2,i} = (2 - x_{1,i}^2 + z_i)/3 \), where \( z_i \) is also standard normal. Under population 1, \( x_{2,i} = -(2 - x_{1,i}^2 + z_i)/3 \). The optimal classification regions form a checkerboard pattern, with horizontal boundary at \( x_2 = 0 \), vertical boundaries at \( x_1 = \pm \sqrt{2} \).

This is the same model as the cltest dataset.

Usage

data(cltrain)

Format

Data Frame. Three variables x1, x2 and y. The latter indicates class membership.

c02

Carbon Dioxide Dataset

Description

Monthly time series of carbon dioxide measurements at Mauna Loa, Hawaii from 1959 to 1990.

Usage

data(co2)

Format

Data frame with year, month and co2 variables.

Source


References

**cp**

_Compute Mallows’ Cp for local regression models._

**Description**

The calling sequence for cp matches those for the `locfit` or `locfit.raw` functions. The fit is not returned; instead, the returned object contains Cp criterion for the fit.

Cp is usually computed using a variance estimate from the largest model under consideration, rather than $\sigma^2 = 1$. This will be done automatically when the `cpplot` function is used.

The Cp score is exact (up to numerical roundoff) if the ev="data" argument is provided. Otherwise, the residual sum-of-squares and degrees of freedom are computed using locfit's standard interpolation based approximations.

**Usage**

```r
cp(x, ..., sig2=1)
```

**Arguments**

- `x`  
  model formula or numeric vector of the independent variable.
- `...`  
  other arguments to `locfit` and/or `locfit.raw`.
- `sig2`  
  residual variance estimate.

**See Also**

`locfit, locfit.raw, cpplot`

**cpar**

_Classically parametric term for a Locfit model._

**Description**

A term entered in a `locfit` model formula using cpar will result in a fit that is conditionally parametric. Equivalent to `lp(x, style="cpar")`.

This function is presently almost deprecated. Specifying a conditionally parametric fit as `y~x1+cp(x2)` will no longer work; instead, the model is specified as `y~lp(x1, x2, style=c("n","cpar"))`.

**Usage**

```r
cpar(x, ...)
```

**Arguments**

- `x`  
  numeric variable.
- `...`  
  Other arguments to `link{lp}`().
Description

The cpplot function loops through calls to the cp function (and hence to link(locfit)), using
a different smoothing parameter for each call. The returned structure contains the Cp statistic for
each fit, and can be used to produce an AIC plot.

Usage

cpplot(..., alpha, sig2)

Arguments

... arguments to the cp, locfit functions.
alpha Matrix of smoothing parameters. The cpplot function loops through calls to cp,
using each row of alpha as the smoothing parameter in turn. If alpha is pro-
vided as a vector, it will be converted to a one-column matrix, thus interpreting
each component as a nearest neighbor smoothing parameter.
sig2 Residual variance. If not specified, the residual variance is computed using the
fitted model with the fewest residual degrees of freedom.

Value

An object with class "gcvplot", containing the smoothing parameters and CP scores. The actual
plot is produced using plot.gcvplot.

See Also

locfit, locfit.raw, gcv, aic, plot.gcvplot

Examples

data(ethanol)
plot(cpplot(NOx~E, data=ethanol, alpha=seq(0.2, 1.0, by=0.05)))
crit

Compute critical values for confidence intervals.

Description

Every "locfit" object contains a critical value object to be used in computing and ploting confidence intervals. By default, a 95% pointwise confidence level is used. To change the confidence level, the critical value object must be substituted using crit and crit<-.

Usage

crit(fit, const=c(0, 1), d=1, cov=0.95, rdf=0)
crit(fit) <- value

Arguments

fit "locfit" object. This is optional; if a fit is provided, defaults for the other arguments are taken from the critical value currently stored on this fit, rather than the usual values above. crit(fit) with no other arguments will just return the current critical value.

cost Tube formula constants for simultaneous bands (the default, c(0, 1), produces pointwise coverage). Usually this is generated by the kappa function and should not be provided by the user.

d Dimension of the fit. Again, users shouldn’t usually provide it.

cov Coverage Probability for critical values.

rdf Residual degrees of freedom. If non-zero, the critical values are based on the Student's t distribution. When rdf=0, the normal distribution is used.

value Critical value object generated by crit or kappa.

Value

Critical value object.

See Also

locfit, plot.locfit, kappa0, crit<-

Examples

# compute and plot 99% confidence intervals, with local variance estimate.
data(ethanol)
fit <- locfit(NOx~E, data=ethanol)
crit(fit) <- crit(fit, cov=0.99)
plot(fit, band="local")

# compute and plot 99% simultaneous bands
crit(fit) <- kappa0(NOx~E, data=ethanol, cov=0.99)
plot(fit, band="local")
Description

dat is used to specify evaluation on the given data points for \texttt{locfit.raw()}. 

Usage

dat(cv=FALSE)

Arguments

cv \hspace{1cm} \text{Whether cross-validation should be done.}

density.lf \hspace{1cm} \text{Density estimation using Locfit}

Description

This function provides an interface to Locfit, in the syntax of (a now old version of) the S-Plus \texttt{density} function. This can reproduce density results, but allows additional \texttt{locfit.raw} arguments, such as the degree of fit, to be given.

It also works in double precision, whereas \texttt{density} only works in single precision.

Usage

density.lf(x, n = 50, window = "gaussian", width, from, to,
       cut = if(iwindow == 4.) 0.75 else 0.5,
       ev = lfggrid(mg = n, ll = from, ur = to),
       deg = 0, family = "density", link = "ident", ...)

Arguments

\begin{itemize}
  \item \texttt{x} \hspace{1cm} \text{numeric vector of observations whose density is to be estimated.}
  \item \texttt{n} \hspace{1cm} \text{number of evaluation points. Equivalent to the \texttt{locfit.raw} mg argument.}
  \item \texttt{window} \hspace{1cm} \text{Window type to use for estimation. Equivalent to the \texttt{locfit.raw} kern argument. This includes all the density windows except cosine.}
  \item \texttt{width} \hspace{1cm} \text{Window width. Following density, this is the full width; not the half-width usually used by Locfit and many other smoothers.}
  \item \texttt{from} \hspace{1cm} \text{Lower limit for estimation domain.}
  \item \texttt{to} \hspace{1cm} \text{Upper limit for estimation domain.}
  \item \texttt{cut} \hspace{1cm} \text{Controls default expansion of the domain.}
\end{itemize}
ev

Locfit evaluation structure – default \texttt{lfgrid}().

deg

Fitting degree – default 0 for kernel estimation.

family

Fitting family – default is "density".

link

Link function – default is the "identity".

...

Additional arguments to \texttt{locfit.raw}, with standard defaults.

\textbf{Value}

A list with components \texttt{x} (evaluation points) and \texttt{y} (estimated density).

\textbf{See Also}

density, \texttt{locfit}, \texttt{locfit.raw}

\textbf{Examples}

data(geyser)
density.lf(geyser, window="tria")
# the same result with density, except less precision.
density(geyser, window="tria")

\begin{verbatim}

\end{verbatim}

\textbf{Exhaust emissions}

\textbf{Description}

NOx exhaust emissions from a single cylinder engine. Two predictor variables are E (the engine’s equivalence ratio) and C (Compression ratio).

\textbf{Usage}

data(ethanol)

\textbf{Format}

Data frame with NOx, E and C variables.

\textbf{Source}


\textbf{References}


**Description**

NOx exhaust emissions from a single cylinder engine. Two predictor variables are E (the engine’s equivalence ratio) and C (Compression ratio).

**Usage**

data(ethanol)

**Format**

Data frame with NOx, E and C variables.

**Source**


**References**


**Description**

Computes $e^x/(1 + e^x)$. This is the inverse of the logistic link function, $\log(p/(1 - p))$.

**Usage**

expit(x)

**Arguments**

x numeric vector
**fitted.locfit**

_Fitted values for a "locfit" object._

**Description**

Evaluates the fitted values (i.e. evaluates the surface at the original data points) for a Locfit object. This function works by reconstructing the model matrix from the original formula, and predicting at those points. The function may be fooled; for example, if the original data frame has changed since the fit, or if the model formula includes calls to random number generators.

**Usage**

```r
## S3 method for class 'locfit'
fitted(object, data=NULL, what="coef", cv=FALSE,
       studentize=FALSE, type="fit", tr, ...)
```

**Arguments**

- `object` "locfit" object.
- `data` The data frame for the original fit. Usually, this shouldn’t be needed, especially when the function is called directly. It may be needed when called inside another function.
- `what` What to compute fitted values of. The default, `what="coef"`, works with the fitted curve itself. Other choices include "n1x" for the length of the weight diagram; "infl" for the influence function; "band" for the bandwidth; "degr" for the local polynomial degree; "lik" for the maximized local likelihood; "rdf" for the local residual degrees of freedom and "vari" for the variance function. The interpolation algorithm for some of these quantities is questionable.
- `cv` If TRUE, leave-one-out cross validated fitted values are approximated. Won’t make much sense, unless `what="coef"`.
- `studentize` If TRUE, residuals are studentized.
- `type` Type of fit or residuals to compute. The default is "fit" for `fitted.locfit`, and "dev" for `residuals.locfit`. Other choices include "pear" for Pearson residuals; "raw" for raw residuals, "ldot" for likelihood derivative; "d2" for the deviance residual squared; 1ddot for the likelihood second derivative. Generally, `type` should only be used when `what="coef"`.
- `tr` Back transformation for likelihood models.
- `...` arguments passed to and from methods.

**Value**

A numeric vector of the fitted values.

**See Also**

`locfit, predict.locfit, residuals.locfit`
formula.locfit  
*Formula from a Locfit object.*

**Description**

Extract the model formula from a locfit object.

**Usage**

```r
## S3 method for class 'locfit'
formula(x, ...)  
```

**Arguments**

- `x`: locfit object.
- `...`: Arguments passed to and from other methods.

**Value**

Returns the formula from the locfit object.

**See Also**

`locfit`

---

**gam.lf**  
*Locfit call for Generalized Additive Models*

**Description**

This is a locfit calling function used by `lf()` terms in additive models. It is not normally called directly by users.

**Usage**

```r
gam.lf(x, y, w, xeval, ...)  
```

**Arguments**

- `x`: numeric predictor
- `y`: numeric response
- `w`: prior weights
- `xeval`: evaluation points
- `...`: other arguments to `locfit.raw()`
gam.slist

See Also
locfit, locfit.raw, lf.gam

---

gam.slist

Vector of GAM special terms

Description
This vector adds "lf" to the default vector of special terms recognized by a gam() model formula. To ensure this is recognized, attach the Locfit library with library(locfit, first=T).

Format
Character vector.

See Also
lf, gam

---

gcv

Compute generalized cross-validation statistic.

Description
The calling sequence for gcv matches those for the locfit or locfit.raw functions. The fit is not returned; instead, the returned object contains Wahba's generalized cross-validation score for the fit.

The GCV score is exact (up to numerical roundoff) if the ev="data" argument is provided. Otherwise, the residual sum-of-squares and degrees of freedom are computed using locfit's standard interpolation based approximations.

For likelihood models, GCV is computed uses the deviance in place of the residual sum of squares. This produces useful results but I do not know of any theory validating this extension.

Usage
gcv(x, ...)

Arguments
x, ... Arguments passed on to locfit or locfit.raw.

See Also
locfit, locfit.raw, gcvplot
gcvplot

*Compute a generalized cross-validation plot.*

Description

The `gcvplot` function loops through calls to the `gcv` function (and hence to `link{locfit}`), using a different smoothing parameter for each call. The returned structure contains the GCV statistic for each fit, and can be used to produce an GCV plot.

Usage

```r
gcvplot(..., alpha, df=2)
```

Arguments

- `...` arguments to the `gcv, locfit` functions.
- `alpha` Matrix of smoothing parameters. The `gcvplot` function loops through calls to `gcv`, using each row of `alpha` as the smoothing parameter in turn. If `alpha` is provided as a vector, it will be converted to a one-column matrix, thus interpreting each component as a nearest neighbor smoothing parameter.
- `df` Degrees of freedom to use as the x-axis. 2=trace(L), 3=trace(L' L).

Value

An object with class "gcvplot", containing the smoothing parameters and GCV scores. The actual plot is produced using `plot.gcvplot`.

See Also

`locfit, locfit.raw, gcv, plot.gcvplot, summary.gcvplot`

Examples

```r
data(ethanol)
plot(gcvplot(NOx~E, data=ethanol, alpha=seq(0.2,1.0,by=0.05)))
```

data(geyser)

*Old Faithful Geyser Dataset*

Description

The durations of 107 eruptions of the Old Faithful Geyser.

Usage

```r
data(geyser)
```
**Format**

A numeric vector of length 107.

**Source**

Scott (1992). Note that several different Old Faithful Geyser datasets (including the faithful dataset in R’s base library) have been used in various places in the statistics literature. The version provided here has been used in density estimation and bandwidth selection work.

**References**


---

<table>
<thead>
<tr>
<th>geyser.round</th>
<th>Discrete Old Faithful Geyser Dataset</th>
</tr>
</thead>
</table>

**Description**

This is a variant of the `geyser` dataset, where each observation is rounded to the nearest 0.05 minutes, and the counts tallied.

**Usage**

`data(geyser.round)`

**Format**

Data Frame with variables `duration` and `count`.

**Source**

Scott (1992). Note that several different Old Faithful Geyser datasets (including the faithful dataset in R’s base library) have been used in various places in the statistics literature. The version provided here has been used in density estimation and bandwidth selection work.

**References**

hatmatrix  

Weight diagrams and the hat matrix for a local regression model.

Description

hatmatrix() computes the weight diagrams (also known as equivalent or effective kernels) for a local regression smooth. Essentially, hatmatrix() is a front-end to locfit(), setting a flag to compute and return weight diagrams, rather than the fit.

Usage

hatmatrix(formula, dc=TRUE, ...)

Arguments

- formula: model formula.
- dc: derivative adjustment (see locfit.raw)
- ...: Other arguments to locfit and locfit.raw.

Value

A matrix with n rows and p columns; each column being the weight diagram for the corresponding locfit fit point. If ev="data", this is the transpose of the hat matrix.

See Also

locfit, plot.locfit.1d, plot.locfit.2d, plot.locfit.3d, lines.locfit, predict.locfit

heart  

Survival Times of Heart Transplant Recipients

Description

The survival times of 184 participants in the Stanford heart transplant program.

Usage

data(heart)

Format

Data frame with surv, cens and age variables.
**Source**
Miller and Halperin (1982). The original dataset includes information on additional patients who never received a transplant. Other authors reported earlier versions of the data.

**References**

---

**insect**

**Insect Dataset**

**Description**
An experiment measuring death rates for insects, with 30 insects at each of five treatment levels.

**Usage**
data(insect)

**Format**
Data frame with `lconc` (dosage), `deaths` (number of deaths) and `nins` (number of insects) variables.

**Source**
Bliss (1935).

**References**

---

**iris**

**Fisher’s Iris Data (subset)**

**Description**
Four measurements on each of fifty flowers of two species of iris (Versicolor and Virginica) – A classification dataset. Fisher’s original dataset contained a third species (Setosa) which is trivially separable.

**Usage**
data(iris)
Format

Data frame with species, petal.wid, petal.len, sepal.wid, sepal.len.

Source


References


Kangaroo skull measurements dataset

Description

Variables are sex (m/f), spec (giganteus, melanops, fuliginosus) and 18 numeric measurements.

Usage

data(kangaroo)

Format

Data frame with measurements on the skulls of 101 kangaroos. (number of insects) variables.

Source


References

Critical Values for Simultaneous Confidence Bands.

Description

The geometric constants for simultaneous confidence bands are computed, as described in Sun and Loader (1994) (bias adjustment is not implemented here). These are then passed to the crit function, which computes the critical value for the confidence bands.

The method requires both the weight diagrams \( l(x) \), the derivative \( l'(x) \) and (in 2 or more dimensions) the second derivatives \( l''(x) \). These are implemented exactly for a constant bandwidth. For nearest neighbor bandwidths, the computations are approximate and a warning is produced.

The theoretical justification for the bands uses normality of the random errors \( e_1, \ldots, e_n \) in the regression model, and in particular the spherical symmetry of the error vector. For non-normal distributions, and likelihood models, one relies on central limit and related theorems.

Computation uses the product Simpson’s rule to evaluate the multidimensional integrals (The domain of integration, and hence the region of simultaneous coverage, is determined by the flim argument). Expect the integration to be slow in more than one dimension. The minp argument controls the precision.

Usage

kappa0(formula = , cov = 0.95, ev = lfgird(20), ...)

Arguments

- **formula**: Local regression model formula. A "locfit" object can also be provided; in this case the formula and other arguments are extracted from this object.
- **cov**: Coverage Probability for critical values.
- **ev**: Locfit evaluation structure. Should usually be a grid – this specifies the integration rule.
- **...**: Other arguments to locfit. Important arguments include flim and alpha.

Value

A list with components for the critical value, geometric constants, e.t.c. Can be passed directly to plot.locfit as the crit argument.

References


See Also

locfit, plot.locfit, crit, crit<-. 
Examples

```r
# compute and plot simultaneous confidence bands
data(ethanol)
fit <- locfit(NOx~E, data=ethanol)
crit(fit) <- kappa0(NOx~E, data=ethanol)
plot(fit, crit=crit, band="local")
```

**Description**

Function to compute kernel density estimate bandwidths, as used in the simulation results in Chapter 10 of Loader (1999).

This function is included for comparative purposes only. Plug-in selectors are based on flawed logic, make unreasonable and restrictive assumptions and do not use the full power of the estimates available in Locfit. Any relation between the results produced by this function and desirable estimates are entirely coincidental.

**Usage**

```r
kdeb(x, h0 = 0.01 * sd, h1 = sd, meth = c("AIC", "LCV", "LSCV", "BCV",
   "SJPI", "GKK"), kern = "gauss", gf = 2.5)
```

**Arguments**

- `x`: One dimensional data vector.
- `h0`: Lower limit for bandwidth selection. Can be fairly small, but h0=0 would cause problems.
- `h1`: Upper limit.
- `meth`: Required selection method(s).
- `kern`: Kernel. Most methods require kern="gauss", the default for this function only.
- `gf`: Standard deviation for the gaussian kernel. Default 2.5, as Locfit’s standard. Most papers use 1.

**Value**

Vector of selected bandwidths.

**References**

Mean Residual Life using Kaplan-Meier estimate

Description

This function computes the mean residual life for censored data using the Kaplan-Meier estimate of the survival function. If $S(t)$ is the K-M estimate, the MRL for a censored observation is computed as $\left(\int_t^{\infty} S(u)du\right)/S(t)$. We take $S(t) = 0$ when $t$ is greater than the largest observation, regardless of whether that observation was censored.

When there are ties between censored and uncensored observations, for definiteness our ordering places the censored observations before uncensored.

This function is used by `locfit.censor` to compute censored regression estimates.

Usage

km.mrl(times, cens)

Arguments

- times: Observed survival times.
- cens: Logical variable indicating censoring. The coding is 1 or TRUE for censored; 0 or FALSE for uncensored.

Value

A vector of the estimated mean residual life. For uncensored observations, the corresponding estimate is 0.

References


See Also

- `locfit.censor`

Examples

```r
# censored regression using the Kaplan-Meier estimate.
fit <- locfit.censor(log10(surv+0.5)-age, cens=cens, data=heart, km=TRUE)
plotbyfactor(heart$age, 0.5+heart$surv, heart$cens, ylim=c(0.5,16000), log="y")
lines(fit, tr=function(x)10^x)
```
lcv

Compute Likelihood Cross Validation Statistic.

Description
The calling sequence for lcv matches those for the locfit or locfit.raw functions. The fit is not returned; instead, the returned object contains likelihood cross validation score for the fit.

The LCV score is exact (up to numerical roundoff) if the ev="cross" argument is provided. Otherwise, the influence and cross validated residuals are computed using locfit’s standard interpolation based approximations.

Usage
lcv(x, ...)

Arguments
x model formula
... other arguments to locfit

See Also
locfit, locfit.raw, lcvplot

lcvplot

Compute the likelihood cross-validation plot.

Description
The lcvplot function loops through calls to the lcv function (and hence to link{locfit}), using a different smoothing parameter for each call. The returned structure contains the likelihood cross validation statistic for each fit, and can be used to produce an LCV plot.

Usage
lcvplot(..., alpha)

Arguments
... arguments to the lcv, locfit functions.
alpha Matrix of smoothing parameters. The aicplot function loops through calls to lcv, using each row of alpha as the smoothing parameter in turn. If alpha is provided as a vector, it will be converted to a one-column matrix, thus interpreting each component as a nearest neighbor smoothing parameter.
Value

An object with class "gcvplot", containing the smoothing parameters and LCV scores. The actual plot is produced using \texttt{plot.gcvplot}.

See Also

\texttt{locfit}, \texttt{locfit.raw}, \texttt{gcv}, \texttt{lcv}, \texttt{plot.gcvplot}

Examples

data(ethanol)
plot(lcvplot(NOx~E,data=ethanol,\alpha=seq(0.2,1.0,by=0.05)))

\texttt{left}

\begin{flushright}
\textit{One-sided left smooth for a Locfit model.}
\end{flushright}

Description

The \texttt{left()} function is used in a \texttt{locfit} model formula to specify a one-sided smooth: when fitting at a point \(x\), only data points with \(x_i \leq x\) should be used. This can be useful in estimating points of discontinuity, and in cross-validation for forecasting a time series. \texttt{left(x)} is equivalent to \texttt{lp(x,style="left")}.

When using this function, it will usually be necessary to specify an evaluation structure, since the fit is not smooth and \texttt{locfit}'s interpolation methods are unreliable. Also, it is usually best to use \texttt{deg=0} or \texttt{deg=1}, otherwise the fits may be too variable. If nearest neighbor bandwidth specification is used, it does not recognize \texttt{left()}. 

Usage

\texttt{left(x,...)}

Arguments

\begin{itemize}
\item \texttt{x} numeric variable.
\item \texttt{...} Other arguments to \texttt{lp}.
\end{itemize}

See Also

\texttt{locfit}, \texttt{lp}, \texttt{right}

Examples

\begin{verbatim}
# compute left and right smooths
data(penny)
xev <- (1945:1988)+0.5
fitl <- locfit(thickness~left(year,\h=10,\deg=1), \ev=xev, \data=penny)
fitr <- locfit(thickness~right(year,\h=10,\deg=1),\ev=xev, \data=penny)
# plot the squared difference, to show the change points.
plot( xev, (predict(fitr,where="\ev") - predict(fitl,where="\ev"))^2 )
\end{verbatim}
**Description**

This function is used to specify a smooth term in a `gam()` model formula.

This function is designed to be used with the S-Plus `gam()` function. For R users, there are at least two different `gam()` functions available. Most current distributions of R will include the `mgcv` library by Simon Wood; `lf()` is not compatible with this function.

On CRAN, there is a `gam` package by Trevor Hastie, similar to the S-Plus version. `lf()` should be compatible with this, although it's untested.

**Usage**

```r
lf(..., alpha=0.7, deg=2, scale=1, kern="tcub", ev=rbox(), maxk=100)
```

**Arguments**

- `...`: numeric predictor variable(s)
- `alpha`, `deg`, `scale`, `kern`, `ev`, `maxk`: these are as in `locfit.raw`.

**See Also**

`locfit`, `locfit.raw`, `gam.lf`, `gam`

**Examples**

```r
# fit an additive semiparametric model to the ethanol data.
stopifnot(require(gam))
# The `gam` package must be attached _before_ `locfit`, otherwise
# the following will not work.
data(ethanol, package = "lattice")
fit <- gam(NOx ~ lf(E) + C, data=ethanol)
op <- par(mfrow=c(2, 1))
plot(fit)
par(op)
```
`lfeval`  
*Extract Locfit Evaluation Structure.*

**Description**

Extracts the evaluation structure from a "locfit" object. This object has the class "lfeval", and has its own set of methods for plotting e.t.c.

**Usage**

`lfeval(object)`

**Arguments**

- `object`  
  "locfit" object

**Value**

"lfeval" object.

**See Also**

- `locfit`, `plot.lfeval`, `print.lfeval`  

---

`lfgrid`  
*Locfit - grid evaluation structure.*

**Description**

`lfgrid()` is used to specify evaluation on a grid of points for `locfit.raw()`. The structure computes a bounding box for the data, and divides that into a grid with specified margins.

**Usage**

`lfgrid(mg=10, ll, ur)`

**Arguments**

- `mg`  
  Number of grid points along each margin. Can be a single number (which is applied in each dimension), or a vector specifying a value for each dimension.

- `ll`  
  Lower left limits for the grid. Length should be the number of dimensions of the data provided to `locfit.raw()`.

- `ur`  
  Upper right limits for the grid. By default, `ll` and `ur` are generated as the bounding box for the data.
Examples

```r
data(ethanol, package="locfit")
plot.eval(locfit(NOx ~ lp(E, C, scale=TRUE), data=ethanol, ev=lfgrid()枀
```

**lfknots**  
*Extraction of fit-point information from a Locfit object.*

**Description**
Extracts information, such as fitted values, influence functions from a "locfit" object.

**Usage**

```r
lfknots(x, tr, what = c("x", "coef", "h", "nlx"), delete.pv = TRUE)
```

**Arguments**
- `x`: Fitted object from `locfit()`.
- `tr`: Back transformation. Default is the inverse link function from the Locfit object.
- `what`: What to return; default is `c("x", "coef", "h", "nlx")`. Allowed fields are `x` (fit points); `coef` (fitted values); `f1` (local slope); `nlx` (length of the weight diagram); `nlx1` (estimated derivative of `nlx`); `se` (standard errors); `infl` (influence function); `infla` (slope of influence function); `lik` (maximised local log-likelihood and local degrees of freedom); `h` (bandwidth) and `deg` (degree of fit).
- `delete.pv`: If `T`, pseudo-vertices are deleted.

**Value**
A matrix with one row for each fit point. Columns correspond to the specified `what` vector; some fields contribute multiple columns.

**lflim**  
*Construct Limit Vectors for Locfit fits.*

**Description**
This function is used internally to interpret `xlim` and `flim` arguments. It should not be called directly.

**Usage**

```r
lflim(limits, nm, ret)
```
Arguments

- limits: Limit argument.
- nm: Variable names.
- ret: Initial return vector.

Value

Vector with length 2*dim.

See Also

locfit

Description

This function is usually called by plot.locfit.

Usage

lfmarg(xlim, m = 40)

Arguments

- xlim: Vector of limits for the grid. Should be of length 2*d; the first d components represent the lower left corner, and the next d components the upper right corner. Can also be a "locfit" object.
- m: Number of points for each grid margin. Can be a vector of length d.

Value

A list, whose components are the d grid margins.

See Also

locfit, plot.locfit
lines.locfit  
Add locfit line to existing plot

Description

Adds a Locfit line to an existing plot. llines is for use within a panel function for Lattice.

Usage

```r
## S3 method for class 'locfit'
llines(x, m=100, tr=x$trans, ...)
## S3 method for class 'locfit'
llines(x, m=100, tr=x$trans, ...)
```

Arguments

- `x`  
  locfit object. Should be a model with one predictor.
- `m`  
  Number of points to evaluate the line at.
- `tr`  
  Transformation function to use for plotting. Default is the inverse link function, or the identity function if derivatives are required.
- `...`  
  Other arguments to the default `lines` function.

See Also

- `locfit`, `plot.locfit`, `lines`

livmet  
liver Metastases dataset

Description

Survival times for 622 patients diagnosed with Liver Metastases. Beware, the censoring variable is coded as 1 = uncensored, so use `cens=1-z` in `locfit()` calls.

Usage

```r
data(livmet)
```

Format

Data frame with survival times (t), censoring indicator (z) and a number of covariates.

Source

Haupt and Mansmann (1995)
locfit

References


locfit

Local Regression, Likelihood and Density Estimation.

Description

locfit is the model formula-based interface to the Locfit library for fitting local regression and likelihood models.

locfit is implemented as a front-end to locfit.raw. See that function for options to control smoothing parameters, fitting family and other aspects of the fit.

Usage

locfit(formula, data=sys.frame(sys.parent()), weights=1, cens=0, base=0, subset, geth=FALSE, ..., lfproc=locfit.raw)

Arguments

formula Model Formula; e.g. y~lp(x) for a regression model; ~lp(x) for a density estimation model. Use of lp() on the RHS is recommended, especially when non-default smoothing parameters are used.
data Data Frame.
weights Prior weights (or sample sizes) for individual observations. This is typically used where observations have unequal variance.
cens Censoring indicator. 1 (or TRUE) denotes a censored observation. 0 (or FALSE) denotes uncensored.
base Baseline for local fitting. For local regression models, specifying a base is equivalent to using y-base as the reponse. But base also works for local likelihood.
subset Subset observations in the data frame.
geth Don’t use.
... Other arguments to locfit.raw() (or the lfproc).
lfproc A processing function to compute the local fit. Default is locfit.raw(). Other choices include locfit.robust(), locfit.censor() and locfit.quasi().

Value

An object with class "locfit". A standard set of methods for printing, plotting, etc. these objects is provided.
References

See Also
locfit.raw

Examples

# fit and plot a univariate local regression
data(ethanol, package="locfit")
fit <- locfit(NOx ~ E, data=ethanol)
plot(fit, get.data=TRUE)

# a bivariate local regression with smaller smoothing parameter
fit <- locfit(NOx~lp(E,C,nn=0.5,scale=0), data=ethanol)
plot(fit)

# density estimation
data(geyser, package="locfit")
fit <- locfit(~ lp(geyser, nn=0.1, h=0.8))
plot(fit, get.data=TRUE)

locfit.censor Censored Local Regression

Description
locfit.censor produces local regression estimates for censored data. The basic idea is to use an
EM style algorithm, where one alternates between estimating the regression and the true values of
censored observations.

locfit.censor is designed as a front end to locfit.raw with data vectors, or as an intermediary
between locfit and locfit.raw with a model formula. If you can stand the syntax, the second
calling sequence above will be slightly more efficient than the third.

Usage
locfit.censor(x, y, cens, ..., iter=3, km=FALSE)

Arguments

x Either a locfit model formula or a numeric vector of the predictor variable.
y If x is numeric, y gives the response variable.
cens Logical variable indicating censoring. The coding is 1 or TRUE for censored; 0
or FALSE for uncensored.
... Other arguments to locfit.raw
iter Number of EM iterations to perform
Reconstruct a Locfit model matrix.

Description
Reconstructs the model matrix, and associated variables such as the response, prior weights and censoring indicators, from a locfit object. This is used by functions such as fitted.locfit; it is not normally called directly. The function will only work properly if the data frame has not been changed since the fit was constructed.

Usage
locfit.matrix(fit, data)

Arguments
  fit      Locfit object
  data     Data Frame.

Examples
data(heart, package="locfit")
fit <- locfit.censor(log10(surv+0.5) ~ age, cens=cens, data=heart)
## Can also be written as:
## Not run: fit <- locfit(log10(surv + 0.5) ~ age, cens=cens, data=heart, lfproc=locfit.censor)
with(heart, plotbyfactor(age, 0.5 + surv, cens, ylim=c(0.5, 16000), log="y")
lines(fit, tr=function(x) 10^x)
locfit.quasi

Value

A list with variables x (the model matrix); y (the response); w (prior weights); sc (scales); ce (censoring indicator) and base (baseline fit).

See Also

locfit, fitted.locfit, residuals.locfit

locfit.quasi

Local Quasi-Likelihood with global reweighting.

Description

locfit.quasi assumes a specified mean-variance relation, and performs iterative reweighted local regression under this assumption. This is appropriate for local quasi-likelihood models, and is an alternative to specifying a family such as “qpoisson”.

locfit.quasi is designed as a front end to locfit.raw with data vectors, or as an intermediary between locfit and locfit.raw with a model formula. If you can stand the syntax, the second calling sequence above will be slightly more efficient than the third.

Usage

locfit.quasi(x, y, weights, ..., iter=3, var=abs)

Arguments

- x: Either a locfit model formula or a numeric vector of the predictor variable.
- y: If x is numeric, y gives the response variable.
- weights: Case weights to use in the fitting.
- ...: Other arguments to locfit.raw
- iter: Number of EM iterations to perform
- var: Function specifying the assumed relation between the mean and variance.

Value

"locfit" object.

See Also

locfit, locfit.raw
Description

locfit.raw is an interface to Locfit using numeric vectors (for a model-formula based interface, use locfit). Although this function has a large number of arguments, most users are likely to need only a small subset.

The first set of arguments (x, y, weights, cens, and base) specify the regression variables and associated quantities.

Another set (scale, alpha, deg, kern, kt, acri and basis) control the amount of smoothing: bandwidth, smoothing weights and the local model. Most of these arguments are deprecated - they’ll currently still work, but should be provided through the lp() model term instead.

deriv and dc relate to derivative (or local slope) estimation.

family and link specify the likelihood family.

xlim and renorm may be used in density estimation.

ev specifies the evaluation structure or set of evaluation points.

maxk, itype, mint, maxit and debug control the Locfit algorithms, and will be rarely used.

geth and sty are used by other functions calling locfit.raw, and should not be used directly.

Usage

locfit.raw(x, y, weights=1, cens=0, base=0,
           scale=FALSE, alpha=0.7, deg=2, kern="tricube", kt="sph",
           acri="none", basis=list(NULL),
           deriv=numeric(0), dc=FALSE,
           family, link="default",
           xlim, renorm=FALSE,
           ev=rbox(),
           maxk=100, itype="default", mint=20, maxit=20, debug=0,
           geth=FALSE, sty="none")

Arguments

x  Vector (or matrix) of the independent variable(s). Can be constructed using the lp() function.

y  Response variable for regression models. For density families, y can be omitted.

weights  Prior weights for observations (reciprocal of variance, or sample size).

cens  Censoring indicators for hazard rate or censored regression. The coding is 1 (or TRUE) for a censored observation, and 0 (or FALSE) for uncensored observations.

base  Baseline parameter estimate. If provided, the local regression model is fitted as 

       \[ Y_i = b_i + m(x_i) + \epsilon_i \]

    with Locfit estimating the \( m(x) \) term. For regression models, this effectively subtracts \( b_i \) from \( Y_i \). The advantage of the base formulation is that it extends to likelihood regression models.
scale  Deprecated - see $lp()$.
alpha  Deprecated - see $lp()$. A single number (e.g. $\alpha=0.7$) is interpreted as a nearest neighbor fraction. With two components (e.g. $\alpha=c(0.7,1.2)$), the first component is a nearest neighbor fraction, and the second component is a fixed component. A third component is the penalty term in locally adaptive smoothing.
deg   Degree of local polynomial. Deprecated - see $lp()$.
kern  Weight function, default = "tcub". Other choices are "rect", "trwt", "tria", "epan", "bisq" and "gauss". Choices may be restricted when derivatives are required; e.g. for confidence bands and some bandwidth selectors.
kt    Kernel type, "sph" (default); "prod". In multivariate problems, "prod" uses a simplified product model which speeds up computations.
acri  Deprecated - see $lp()$.
basis User-specified basis functions.
deriv Derivative estimation. If deriv=1, the returned fit will be estimating the derivative (or more correctly, an estimate of the local slope). If deriv=c(1,1) the second order derivative is estimated. deriv=2 is for the partial derivative, with respect to the second variable, in multivariate settings.
dc    Derivative adjustment.
family Local likelihood family; "gaussian"; "binomial"; "poisson"; "gamma" and "geom". Density and rate estimation families are "dens", "rate" and "hazard" (hazard rate). If the family is preceded by a 'q' (for example, family="qbinomial"), quasi-likelihood variance estimates are used. Otherwise, the residual variance ($rv$) is fixed at 1. The default family is "gauss" if a response $y$ is provided; "density" if no response is provided.
link Link function for local likelihood fitting. Depending on the family, choices may be "ident", "log", "logit", "inverse", "sqrt" and "arcsin".
xlim  For density estimation, Locfit allows the density to be supported on a bounded interval (or rectangle, in more than one dimension). The format should be c(l1,u1) where $l1$ is a vector of the lower bounds and $u1$ the upper bounds. Bounds such as $[0,\infty)$ are not supported, but can be effectively implemented by specifying a very large upper bound.
renorm Local likelihood density estimates may not integrate exactly to 1. If renorm=T, the integral will be estimated numerically and the estimate rescaled. Presently this is implemented only in one dimension.
ev    The evaluation structure, $rbox()$ for tree structures; $lfgrid()$ for grids; $dat()$ for data points; $none()$ for none. A vector or matrix of evaluation points can also be provided, although in this case you may prefer to use the smooth.lf() interface to Locfit. Note that arguments flim, mg and cut are now given as arguments to the evaluation structure function, rather than to locfit.raw() directly (change effective 12/2001).
maxk Controls space assignment for evaluation structures. For the adaptive evaluation structures, it is impossible to be sure in advance how many vertices will be generated. If you get warnings about ‘Insufficient vertex space’, Locfit’s default assignment can be increased by increasing maxk. The default is maxk=100.
Integration type for density estimation. Available methods include "prod", "mult" and "mlin"; and "haz" for hazard rate estimation problems. The available integration methods depend on model specification (e.g. dimension, degree of fit). By default, the best available method is used.


If > 0; prints out some debugging information.

Don’t use!

Deprecated - see lp().

Value

An object with class "locfit". A standard set of methods for printing, plotting, etc. these objects is provided.

References

Consult the Web page http://www.locfit.info/.

locfit.robust

Robust Local Regression

Description

locfit.robust implements a robust local regression where outliers are iteratively identified and downweighted, similarly to the lowess method (Cleveland, 1979). The iterations and scale estimation are performed on a global basis.

The scale estimate is 6 times the median absolute residual, while the robust downweighting uses the bisquare function. These are performed in the S code so easily changed.

This can be interpreted as an extension of M estimation to local regression. An alternative extension (implemented in locfit via family="qrgauss") performs the iteration and scale estimation on a local basis.

Usage

locfit.robust(x, y, weights, ..., iter=3)

Arguments

x Either a locfit model formula or a numeric vector of the predictor variable.

y If x is numeric, y gives the response variable.

weights weights to use in the fitting.

... Other arguments to locfit.raw.

iter Number of iterations to perform
Value

"locfit" object.

References


See Also

locfit, locfit.raw

---

lp

Local Polynomial Model Term

Description

lp is a local polynomial model term for Locfit models. Usually, it will be the only term on the RHS of the model formula.

Smoothing parameters should be provided as arguments to lp(), rather than to locfit().

Usage

lp(..., nn, h, adpen, deg, acri, scale, style)

Arguments

... Predictor variables for the local regression model.

nn Nearest neighbor component of the smoothing parameter. Default value is 0.7, unless either h or adpen are provided, in which case the default is 0.

h The constant component of the smoothing parameter. Default: 0.

adpen Penalty parameter for adaptive fitting.

deg Degree of polynomial to use.

acri Criterion for adaptive bandwidth selection.

style Style for special terms (left, ang e.t.c.). Do not try to set this directly; call locfit instead.

scale A scale to apply to each variable. This is especially important for multivariate fitting, where variables may be measured in non-comparable units. It is also used to specify the frequency for ang terms. If scale=F (the default) no scaling is performed. If scale=T, marginal standard deviations are used. Alternatively, a numeric vector can provide scales for the individual variables.

See Also

locfit, locfit.raw
Examples

data(ethanol, package="locfit")
# fit with 50% nearest neighbor bandwidth.
fit <- locfit(NOx~lp(E, nn=0.5), data=ethanol)
# bivariate fit.
fit <- locfit(NOx~lp(E, C, scale=TRUE), data=ethanol)

# density estimation
data(geyser, package="locfit")
fit <- locfit.raw(lp(geyser, nn=0.1, h=0.8))

lscv

Least Squares Cross Validation Statistic.

Description

The calling sequence for lscv matches those for the locfit or locfit.raw functions. Note that this function is only designed for density estimation in one dimension. The returned object contains the least squares cross validation score for the fit.

The computation of $\int \hat{f}(x)^2 dx$ is performed numerically. For kernel density estimation, this is unlikely to agree exactly with other LSCV routines, which may perform the integration analytically.

Usage

lscv(x, ..., exact=FALSE)

Arguments

x model formula (or numeric vector, if exact=T)
... other arguments to locfit or lscv.exact
exact By default, the computation is approximate. If exact=TRUE, exact computation using lscv.exact is performed. This uses kernel density estimation with a constant bandwidth.

Value

A vector consisting of the LSCV statistic and fitted degrees of freedom.

See Also

locfit, locfit.raw, lscv.exact lscvplot

Examples

# approximate calculation for a kernel density estimate
data(geyser, package="locfit")
lscv(~lp(geyser, h=1, deg=0), ev=lfgrid(100, ll=1, ur=6), kern="gauss")
# same computation, exact
lscv(lp(geyser, h=1), exact=TRUE)
**lscv.exact**  
*Exact LSCV Calculation*

**Description**

This function performs the exact computation of the least squares cross validation statistic for one-dimensional kernel density estimation and a constant bandwidth.

At the time of writing, it is implemented only for the Gaussian kernel (with the standard deviation of 0.4; Locfit’s standard).

**Usage**

`lscv.exact(x, h)`

**Arguments**

- `x`  
  Numeric data vector.

- `h`  
  The bandwidth. If `x` is constructed with `lp()`, the bandwidth should be given there instead.

**Value**

A vector of the LSCV statistic and the fitted degrees of freedom.

**See Also**

- `lscv`, `lscvplot`

**Examples**

```r
data(geyser, package="locfit")
lscv.exact(lp(geyser, h=0.25))
# equivalent form using lscv
lscv(lp(geyser, h=0.25), exact=TRUE)
```

**lscvplot**  
*Compute the LSCV plot.*

**Description**

The `lscvplot` function loops through calls to the `lscv` function (and hence to `link(locfit)`), using a different smoothing parameter for each call. The returned structure contains the LSCV statistic for each density estimate, and can be used to produce an LSCV plot.
Usage

lscvplot(..., alpha)

Arguments

... arguments to the `lscv, locfit` functions.
alpha Matrix of smoothing parameters. The `lscvplot` function loops through calls to `lscv`, using each row of `alpha` as the smoothing parameter in turn. If `alpha` is provided as a vector, it will be converted to a one-column matrix, thus interpreting each component as a nearest neighbor smoothing parameter.

Value

An object with class "gcvplot", containing the smoothing parameters and LSCV scores. The actual plot is produced using `plot.gcvplot`.

See Also

`locfit, locfit.raw, gcv, lscv, plot.gcvplot`

---

`mcyc`  
_Acceleration of a Motorcycle Hitting a Wall_

Description

Measurements of the acceleration of a motorcycle as it hits a wall. Actually, rumored to be a concatenation of several such datasets.

Usage

data(mCYC)

Format

Data frame with time and accel variables.

Source

H"ardle (1990).

References

Fracture Counts in Coal Mines

Description

The number of fractures in the upper seam of coal mines, and four predictor variables. This dataset can be modeled using Poisson regression.

Usage

data(mine)

Format

A dataframe with the response frac, and predictor variables extrp, time, seamh and inb.

Source


References


Test dataset for minimax Local Regression

Description

50 observations, as used in Figure 13.1 of Loader (1999).

Usage

data(cltest)

Format

Data Frame with x and y variables.

References

morths  Henderson and Sheppard Mortality Dataset

Description

Observed mortality for 55 to 99.

Usage

data(morths)

Format

Data frame with age, n and number of deaths.

Source

Henderson and Sheppard (1919).

References


none  Locfit Evaluation Structure

Description

none() is an evaluation structure for locfit.raw(), specifying no evaluation points. Only the initial parametric fit is computed - this is the easiest and most efficient way to coerce Locfit into producing a parametric regression fit.

Usage

none()

Examples

data(ethanol, package="locfit")
# fit a fourth degree polynomial using locfit
fit <- locfit(NOx~E, data=ethanol, deg=4, ev=none())
plot(fit, get.data=TRUE)
penny

Penny Thickness Dataset

Description
For each year, 1945 to 1989, the thickness of two U.S. pennies was recorded.

Usage
data(penny)

Format
A dataframe.

Source

References

plot.eval

Plot evaluation points from a 2-d locfit object.

Description
This function is used to plot the evaluation structure generated by Locfit for a two dimensional fit. Vertices of the tree structure are displayed as \( O \); pseudo-vertices as \( * \).

Usage
plot.eval(x, add=FALSE, text=FALSE, ...)

Arguments
x "locfit" object.
add If TRUE, add to existing plot.
text If TRUE, numbers will be added indicating the order points were added.
... Arguments passed to and from other methods.

See Also
locfit.
plot.gcvplot

Examples

data(ethanol, package="locfit")
fit <- locfit(NOx ~ E + C, data=ethanol, scale=0)
plot.eval(fit)

plot.gcvplot

Produce a cross-validation plot.

Description

Plots the value of the GCV (or other statistic) in a gcvplot object against the degrees of freedom of the fit.

Usage

## S3 method for class 'gcvplot'
plot(x, xlab = "Fitted DF", ylab = x$cri, ...)

Arguments

x A gcvplot object, produced by gcvplot, aicplot etc.
xlab Text label for the x axis.
ylab Text label for the y axis.
... Other arguments to plot.

See Also

locfit, locfit.raw, gcv, aicplot, cpplot, gcvplot, lcvplot

Examples

data(ethanol)
plot(gcvplot(NOx~E, data=ethanol, alpha=seq(0.2,1.0,by=0.05)))
plot.lfeval

Plot a Locfit Evaluation Structure.

Description

Plots the evaluation points from a locfit or lfeval structure, for one- or two-dimensional fits.

Usage

## S3 method for class 'lfeval'
plot(x, add=FALSE, txt=FALSE, ...)

Arguments

x
A lfeval or locfit object

add
If TRUE, the points will be added to the existing plot. Otherwise, a new plot is created.

txt
If TRUE, the points are annotated with numbers in the order they were entered into the fit.

...
Additional graphical parameters.

Value

"lfeval" object.

See Also

lfeval, locfit.print.lfeval

plot.locfit

Plot an object of class locfit.

Description

The plot.locfit function generates grids of plotting points, followed by a call to preplot.locfit. The returned object is then passed to plot.locfit.1d, plot.locfit.2d or plot.locfit.3d as appropriate.

Usage

## S3 method for class 'locfit'
plot(x, xlim, pv, tv, m, mtv=6, band="none", tr=NULL,
     what = "coef", get.data=FALSE, f3d=(d == 2) && (length(tv) > 0), ...)
Arguments

x
locfit object.

xlim
Plotting limits. Eg. xlim=c(0, 0, 1, 1) plots over the unit square in two dimensions. Default is bounding box of the data.

pv
Panel variables, to be varied within each panel of a plot. May be specified as a character vector, or variable numbers. There must be one or two panel variables; default is all variables in one or two dimensions; Variable 1 in three or more dimensions. May by specified using either variable numbers or names.

tv
Trellis variables, to be varied from panel to panel of the plot.

m
Controls the plot resolution (within panels, for trellis displays). Default is 100 points in one dimension; 40 points (per dimension) in two or more dimensions.

mtv
Number of points for trellis variables; default 6.

band
Type of confidence bands to add to the plot. Default is "none". Other choices include "global" for bands using a global variance estimate; "local" for bands using a local variance estimate and "pred" for prediction bands (at present, using a global variance estimate). To obtain the global variance estimate for a fit, use rv. This can be changed with rv<-. Confidence bands, by default, are 95%, based on normal approximations and neglecting bias. To change the critical value or confidence level, or to obtain simultaneous instead of pointwise confidence, the critical value stored on the fit must be changed. See the kappa0 and crit functions.

tr
Transformation function to use for plotting. Default is the inverse link function, or the identity function if derivatives are requested.

what
What to plot. See predict.locfit.

get.data
If TRUE, original data is added to the plot. Default: FALSE.

f3d
Force the locfit.3d class on the prediction object, thereby generating a trellis style plot. Default: FALSE, unless a tv argument is' provided. Not available in R.

Other arguments to plot.locfit.1d, plot.locfit.2d or plot.locfit.3d as appropriate.

See Also

locfit, plot.locfit.1d, plot.locfit.2d, plot.locfit.3d, lines.locfit, predict.locfit, preplot.locfit

Examples

x <- rnorm(100)
y <- dnorm(x) + rnorm(100) / 5
plot(locfit(y~x, band="global")

x <- cbind(rnorm(100), rnorm(100))
plot(locfit(~x), type="persp")
plot.locfit.1d  

Plot a one dimensional preplot.locfit object.

Description

This function is not usually called directly. It will be called automatically when plotting a one-dimensional locfit or preplot.locfit object.

Usage

## S3 method for class 'locfit.1d'
plot(x, add=FALSE, main='', xlab="default", ylab=x$yname,
     type="l", ylim, lty=1, col=1, ...)

Arguments

x  
One dimensional preplot.locfit object.

add  
If TRUE, the plot will be added to the existing plot.

main, xlab, ylab, type, yylim, lty, col

Graphical parameters passed on to plot (only if add=FALSE).

...  
Additional graphical parameters to the plot function (only if add=FALSE).

See Also

locfit, plot.locfit.preplot.locfit

plot.locfit.2d  

Plot a two-dimensional "preplot.locfit" object.

Description

This function is not usually called directly. It will be called automatically when plotting one-dimensional locfit or preplot.locfit objects.

Usage

## S3 method for class 'locfit.2d'
plot(x, type="contour", main, xlab, ylab, zlab=x$yname, ...)

Arguments

x  
Two dimensional preplot.locfit object.

type  
one of "contour", "persp", or "image".

main  
title for the plot.

xlab, ylab  
text labels for the x- and y-axes.

zlab  
if type="persp", the label for the z-axis.

...  
Additional arguments to the contour, persp or image functions.
plot.locfit.3d  

See Also  

locfit, plot.locfit, preplot.locfit

plot.locfit.3d  

Plot a high-dimensional "preplot.locfit" object using trellis displays.

Description

This function plots cross-sections of a Locfit model (usually in three or more dimensions) using trellis displays. It is not usually called directly, but is invoked by plot.locfit.

The R libraries lattice and grid provide a partial (at time of writing) implementation of trellis. Currently, this works with one panel variable.

Usage

```
## S3 method for class 'locfit.3d'
plot(x, main='', pv, tv, type = "level", pred.lab = x$vnames,
     resp.lab=xpname, crit = 1.96, ...)
```

Arguments

- `x` "preplot.locfit" object.
- `main` title for the plot.
- `pv` Panel variables. These are the variables (either one or two) that are varied within each panel of the display.
- `tv` Trellis variables. These are varied from panel to panel of the display.
- `type` Type of display. When there are two panel variables, the choices are "contour", "level" and "persp".
- `pred.lab` label for the predictor variable.
- `resp.lab` label for the response variable.
- `crit` critical value for the confidence level.
- `...` graphical parameters passed to xyplot or contourplot.

See Also

plot.locfit, preplot.locfit
The `plot.locfit()` function is implemented, roughly, as a call to `preplot.locfit()`, followed by a call to `plot.locfitpred()`. For most users, there will be little need to call `plot.locfitpred()` directly.

### Usage

```r
## S3 method for class 'preplot.locfit'
plot(x, pv, tv, ...)
```

### Arguments

- `x`: A `preplot.locfit` object, produced by `preplot.locfit()`.
- `pv`, `tv`, `...`: Other arguments to `locfit.1d`, `locfit.2d` or `locfit.3d` as appropriate.

### See Also

`locfit`, `plot.locfit`, `preplot.locfit`, `plot.locfit.1d`, `plot.locfit.2d`, `plot.locfit.3d`.

---

Plot method for simultaneous confidence bands created by the `scb` function.

### Usage

```r
## S3 method for class 'scb'
plot(x, add=FALSE, ...)
```

### Arguments

- `x`: scb object created by `scb`.
- `add`: If TRUE, bands will be added to the existing plot.
- `...`: Arguments passed to and from other methods.

### See Also

`scb`
Examples

```r
# correct confidence bands for a linear logistic model
data(insect)
fit <- scb(deaths ~ lconc, type=4, w=nins, data=insect,
          deg=1, family="binomial", kern="parm")
plot(fit)
```

---

**plotbyfactor**

*x-y scatterplot, colored by levels of a factor.*

Description

Produces a scatter plot of x-y data, with different classes given by a factor f. The different classes are identified by different colours and/or symbols.

Usage

```r
plotbyfactor(x, y, f, data, col = 1:10, pch = "O", add = FALSE, lg,
             xlab = deparse(substitute(x)), ylab = deparse(substitute(y)),
             log = "", ...)```

Arguments

- `x`: Variable for x axis.
- `y`: Variable for y axis.
- `f`: Factor (or variable for which as.factor() works).
- `data`: Data frame for variables x, y, f. Default: sys.parent().
- `col`: Color numbers to use in plot. Will be replicated if shorter than the number of levels of the factor f. Default: 1:10.
- `pch`: Vector of plot characters. Replicated if necessary. Default: "O".
- `add`: If TRUE, add to existing plot. Otherwise, create new plot.
- `lg`: Coordinates to place a legend. Default: Missing (no legend).
- `xlab`, `ylab`: Axes labels.
- `log`: Should the axes be in log scale? Use "x", "y", or "xy" to specify which axis to be in log scale.
- `...`: Other graphical parameters, labels, titles e.t.c.

Examples

```r
data(iris)
plotbyfactor(petal.wid, petal.len, species, data=iris)
```
points.locfit
Add 'locfit' points to existing plot

Description

This function shows the points at which the local fit was computed directly, rather than being interpolated. This can be useful if one is unsure of the validity of interpolation.

Usage

```r
## S3 method for class 'locfit'
points(x, tr, ...)
```

Arguments

- `x` "locfit" object. Should be a model with one predictor.
- `tr` Back transformation.
- `...` Other arguments to the default `points` function.

See Also

`locfit`, `plot.locfit`, `points`

predict.locfit
Prediction from a Locfit object.

Description

The `locfit` function computes a local fit at a selected set of points (as defined by the `ev` argument). The `predict.locfit` function is used to interpolate from these points to any other points. The method is based on cubic hermite polynomial interpolation, using the estimates and local slopes at each fit point.

The motivation for this two-step procedure is computational speed. Depending on the sample size, dimension and fitting procedure, the local fitting method can be expensive, and it is desirable to keep the number of points at which the direct fit is computed to a minimum. The interpolation method used by `predict.locfit()` is usually much faster, and can be computed at larger numbers of points.

Usage

```r
## S3 method for class 'locfit'
predict(object, newdata=NULL, where = "fitp",
         se.fit=FALSE, band="none", what="coef", ...)
```
Arguments

- **object**: Fitted object from `locfit()`.
- **newdata**: Points to predict at. Can be given in several forms: vector/matrix; list, data frame.
- **se.fit**: If TRUE, standard errors are computed along with the fitted values.
- **where, what, band**: Arguments passed on to `preplot.locfit`.
- **...**: Additional arguments to `preplot.locfit`.

Value

- If `se.fit`=!T, a numeric vector of predictors. If `se.fit`=T, a list with components `fit`, `se.fit` and `residual.scale`.

Examples

```r
data(ethanol, package="locfit")
fit <- locfit(NOx ~ E, data=ethanol)
predict(fit, c(0.6,0.8,1.0))
```

---

**preplot.locfit**  
*Prediction from a Locfit object.*

Description

`preplot.locfit` can be called directly, although it is more usual to call `plot.locfit` or `predict.locfit`. The advantage of `preplot.locfit` is in S-Plus 5, where arithmetic and transformations can be performed on the "preplot.locfit" object.

`plot(preplot(fit))` is essentially synonymous with `plot(fit)`.

Usage

```r
## S3 method for class 'locfit'
preplot(object, newdata=NULL, where, tr=NULL, what="coef", band="none", get.data=FALSE, f3d=FALSE, ...)
```

Arguments

- **object**: Fitted object from `locfit()`.
- **newdata**: Points to predict at. Can be given in several forms: vector/matrix; list, data frame.
- **where**: An alternative to `newdata`. Choices include "grid" for the grid `lfmarg(object); "data" for the original data points and "fitp" for the direct fitting points (ie. no interpolation).
- **tr**: Transformation for likelihood models. Default is the inverse of the link function.
What to compute predicted values of. The default, `what = "coef"`, works with the fitted curve itself. Other choices include "nlx" for the length of the weight diagram; "inf1" for the influence function; "band" for the bandwidth; "degr" for the local polynomial degree; "lik" for the maximized local likelihood; "rdf" for the local residual degrees of freedom and "vari" for the variance function. The interpolation algorithm for some of these quantities is questionable.

Compute standard errors for the fit and include confidence bands on the returned object. Default is "none". Other choices include "global" for bands using a global variance estimate; "local" for bands using a local variance estimate and "pred" for prediction bands (at present, using a global variance estimate). To obtain the global variance estimate for a fit, use `rv`. This can be changed with `rv<-.` Confidence bands, by default, are 95%, based on normal approximations and neglecting bias. To change the critical value or confidence level, or to obtain simultaneous instead of pointwise confidence, the critical value stored on the fit must be changed. See the `kappa` and `crit` functions.

If TRUE, the original data is attached to the returned object, and added to the plot.

If TRUE, sets a flag that forces plotting using the trellis style. Not available in R.

Arguments passed to and from other methods.

Value

An object with class "preplot.locfit", containing the predicted values and additional information used to construct the plot.

See Also

`locfit`, `predict.locfit`, `plot.locfit`.

`preplot.locfit.raw`  
*Prediction from a Locfit object.*

Description

`preplot.locfit.raw` is an internal function used by `predict.locfit` and `preplot.locfit`. It should not normally be called directly.

Usage

```
## S3 method for class 'locfit.raw'
preplot(object, newdata, where, what, band, ...)
```
Arguments

object      Fitted object from \texttt{locfit}().
newdata     New data points.
where       Type of data provided in \texttt{newdata}.
what        What to compute predicted values of.
band        Compute standard errors for the fit and include confidence bands on the returned
            object.
            ...  Arguments passed to and from other methods.

Value

A list containing raw output from the internal prediction routines.

See Also

\texttt{locfit, predict.locfit, preplot.locfit}.

---

\texttt{print.gcvplot}  \hspace{1cm} \textit{Print method for gcvplot objects}

Description

Print method for “gcvplot” objects. Actually, equivalent to \texttt{plot.gcvplot(). scb} function.

Usage

\begin{verbatim}
## S3 method for class 'gcvplot'
print(x, ...)
\end{verbatim}

Arguments

x            gcvplot object.
            ...  Arguments passed to and from other methods.

See Also

\texttt{gcvplot.plot.gcvplot summary.gcvplot}
print.locfit

\begin{verbatim}
print.locfit
\end{verbatim}

\textit{Description}

Prints a short summary of a "locfit" object.

\textbf{Usage}

\begin{verbatim}
## S3 method for class 'locfit'
print(x, ...)
\end{verbatim}

\textbf{Arguments}

\begin{itemize}
\item \texttt{x} \hspace{1cm} locfit object.
\item \texttt{...} \hspace{1cm} Arguments passed to and from other methods.
\end{itemize}

\textbf{See Also}

\begin{verbatim}
locfit
\end{verbatim}
print.preplot.locfit  Print method for preplot.locfit objects.

Description
Print method for objects created by the preplot.locfit function.

Usage
### S3 method for class 'preplot.locfit'
print(x, ...)

Arguments
- x: "preplot.locfit" object.
- ...: Arguments passed to and from other methods.

See Also
- preplot.locfit, predict.locfit

print.scb  Print method for simultaneous confidence bands

Description
Print method for simultaneous confidence bands created by the scb function.

Usage
### S3 method for class 'scb'
print(x, ...)

Arguments
- x: "scb" object created by scb.
- ...: Arguments passed to and from other methods.

See Also
- scb
print.summary.locfit  Print a Locfit summary object.

Description
Print method for "summary.locfit" objects.

Usage
## S3 method for class 'summary.locfit'
pindent(x, ...)

Arguments
x  Object from summary.locfit.
... Arguments passed to and from methods.

See Also
summary.locfit()

rbox  Local Regression, Likelihood and Density Estimation.

Description
rbox() is used to specify a rectangular box evaluation structure for locfit.raw(). The structure begins by generating a bounding box for the data, then recursively divides the box to a desired precision.

Usage
rbox(cut=0.8, type="tree", ll, ur)

Arguments

type  If type="tree", the cells are recursively divided according to the bandwidths at each corner of the cell; see Chapter 11 of Loader (1999). If type="kdtree", the K-D tree structure used in Loess (Cleveland and Grosse, 1991) is used.
cut  Precision of the tree; a smaller value of cut results in a larger tree with more nodes being generated.
ll  Lower left corner of the initial cell. Length should be the number of dimensions of the data provided to locfit.raw().
ur  Upper right corner of the initial cell. By default, ll and ur are generated as the bounding box for the data.
**regband**  

**References**  

**Examples**  
```r  
data(ethanol, package="locfit")  
plot.eval(locfit(NOx~E+C, data=ethanol, scale=0, ev=rbox(cut=0.8)))  
plot.eval(locfit(NOx~E+C, data=ethanol, scale=0, ev=rbox(cut=0.3)))  
```

---

**regband**  

*Bandwidth selectors for local regression.*

**Description**  

Function to compute local regression bandwidths for local linear regression, implemented as a front end to *locfit*().  
This function is included for comparative purposes only. Plug-in selectors are based on flawed logic, make unreasonable and restrictive assumptions and do not use the full power of the estimates available in Locfit. Any relation between the results produced by this function and desirable estimates are entirely coincidental.

**Usage**  

```r  
regband(formula, what = c("CP", "GCV", "GKK", "RSW"), deg=1, ...)  
```

**Arguments**  

- `formula`: Model Formula (one predictor).  
- `what`: Methods to use.  
- `deg`: Degree of fit.  
- `...`: Other Locfit options.

**Value**  
Vector of selected bandwidths.
residuals.locfit  

*Fitted values and residuals for a Locfit object.*

**Description**

`residuals.locfit` is implemented as a front-end to `fitted.locfit`, with the type argument set.

**Usage**

```r
## S3 method for class 'locfit'
residuals(object, data=NULL, type="deviance", ...)
```

**Arguments**

- `object`  
  locfit object.
- `data`  
  The data frame for the original fit. Usually, shouldn’t be needed.
- `type`  
  Type of fit or residuals to compute. The default is "fit" for `fitted.locfit`, and "dev" for `residuals.locfit`. Other choices include "pear" for Pearson residuals; "raw" for raw residuals, "1dot" for likelihood derivative; "d2" for the deviance residual squared; 1ddot for the likelihood second derivative. Generally, type should only be used when what="coef".

*...*  
arguments passed to and from other methods.

**Value**

A numeric vector of the residuals.

---

right  

*One-sided right smooth for a Locfit model.*

**Description**

The `right()` function is used in a locfit model formula to specify a one-sided smooth: when fitting at a point `x`, only data points with `x_i ≤ x` should be used. This can be useful in estimating points of discontinuity, and in cross-validation for forecasting a time series. `right(x)` is equivalent to `lp(x,style="right")`.

When using this function, it will usually be necessary to specify an evaluation structure, since the fit is not smooth and locfit's interpolation methods are unreliable. Also, it is usually best to use `deg=0` or `deg=1`; otherwise the fits may be too variable. If nearest neighbor bandwidth specification is used, it does not recognize `right()`.

**Usage**

```r
right(x,...)
```
Arguments

x numeric variable.

... Other arguments to `lp()`.

See Also

`locfit`, `left`

Examples

```r
# compute left and right smooths
data(penny)
xev <- (1945:1988)+0.5
fitl <- locfit(thickness~left(year, h=10, deg=1), ev=xev, data=penny)
fitr <- locfit(thickness~right(year, h=10, deg=1), ev=xev, data=penny)
# plot the squared difference, to show the change points.
plot( xev, (predict(fitr, where="ev") - predict(fitl, where="ev"))^2 )
```

---

**rv**  
*Residual variance from a locfit object.*

Description

As part of the `locfit` fitting procedure, an estimate of the residual variance is computed; the `rv` function extracts the variance from the "locfit" object. The estimate used is the residual sum of squares (or residual deviance, for quasi-likelihood models), divided by the residual degrees of freedom.

For likelihood (not quasi-likelihood) models, the estimate is 1.0.

Usage

`rv(fit)`

Arguments

fit "locfit" object.

Value

Returns the residual variance estimate from the "locfit" object.

See Also

`locfit`, `rv<-`
Examples

```r
data(ethanol)
fit <- locfit(NOx~E, data=ethanol)
rv(fit)
```

---

**Description**

By default, Locfit uses the normalized residual sum of squares as the variance estimate when constructing confidence intervals. In some cases, the user may like to use alternative variance estimates; this function allows the default value to be changed.

**Usage**

```r
rva(fit) <- value
```

**Arguments**

- `fit` "locfit" object.
- `value` numeric replacement value.

**See Also**

`locfit()`, `rv()`, `plot.locfit()`

---

**Description**

`scb` is implemented as a front-end to `locfit`, to compute simultaneous confidence bands using the tube formula method and extensions, based on Sun and Loader (1994).

**Usage**

```r
scb(x, ..., ev = lfggrid(20), simul = TRUE, type = 1)
```
Arguments

x  A numeric vector or matrix of predictors (as in `locfit.raw`), or a model formula (as in `locfit`).

... Additional arguments to `locfit.raw`.

ev The evaluation structure to use. See `locfit.raw`.

simul Should the coverage be simultaneous or pointwise?

type Type of confidence bands. type=0 computes pointwise 95% bands. type=1 computes basic simultaneous bands with no corrections. type=2, 3, 4 are the centered and corrected bands for parametric regression models listed in Table 3 of Sun, Loader and McCormick (2000).

Value

A list containing the evaluation points, fit, standard deviations and upper and lower confidence bounds. The class is "scb"; methods for printing and plotting are provided.

References


See Also

`locfit`, `print.scb`, `plot.scb`.

Examples

```r
# corrected confidence bands for a linear logistic model
data(insect)
fit <- scb(deaths~lp(lconc,deg=1), type=4, w=nins,
          data=insect,family="binomial",kern="parm")
plot(fit)
```

sjpi  

*Sheather-Jones Plug-in bandwidth criterion.*

Description

Given a dataset and set of pilot bandwidths, this function computes a bandwidth via the plug-in method, and the assumed ‘pilot’ relationship of Sheather and Jones (1991). The S-J method chooses the bandwidth at which the two intersect.

The purpose of this function is to demonstrate the sensitivity of plug-in methods to pilot bandwidths and assumptions. This function does not provide a reliable method of bandwidth selection.
Usage

sjpi(x, a)

Arguments

x  data vector
a  vector of pilot bandwidths

Value

A matrix with four columns; the number of rows equals the length of a. The first column is the plug-in selected bandwidth. The second column is the pilot bandwidths a. The third column is the pilot bandwidth according to the assumed relationship of Sheather and Jones. The fourth column is an intermediate calculation.

References


See Also

locfit, locfit.raw, lcvplot

Examples

# Fig 10.2 (5-J parts) from Loader (1999).
data(geyser, package="locfit")
gf <- 2.5
a <- seq(0.05, 0.7, length=100)
z <- sjpi(geyser, a)

# the plug-in curve. Multiplying by gf=2.5 corresponds to Locfit's standard
# scaling for the Gaussian kernel.
plot(gf*z[, 2], gf*z[, 1], type = "l", xlab = "Pilot Bandwidth k", ylab = "Bandwidth h")

# Add the assumed curve.
lines(gf * z[, 3], gf * z[, 1], lty = 2)
legend(gf*0.05, gf*0.4, lty = 1:2, legend = c("Plug-in", "SJ assumed"))

smooth.lf  

Local Regression, Likelihood and Density Estimation.

Description

smooth.lf is a simple interface to the Locfit library. The input consists of a predictor vector (or matrix) and response. The output is a list with vectors of fitting points and fitted values. Most locfit.raw options are valid.
Usage

```r
smooth.lf(x, y, xev=x, direct=FALSE, ...)
```

Arguments

- **x**: Vector (or matrix) of the independent variable(s).
- **y**: Response variable. If omitted, `x` is treated as the response and the predictor variable is `1:n`.
- **xev**: Fitting Points. Default is the data vector `x`.
- **direct**: Logical variable. If `T`, local regression is performed directly at each fitting point. If `F`, the standard Locfit method combining fitting and interpolation is used.
- **...**: Other arguments to `locfit.raw()`.

Value

A list with components `x` (fitting points) and `y` (fitted values). Also has a call component, so `update()` will work.

See Also

`locfit()`, `locfit.raw()`, `density.lf()`.

Examples

```r
# using smooth.lf() to fit a local likelihood model.
data(morths)
fit <- smooth.lf(morths$age, morths$deaths, weights=morths$n,
                 family="binomial")
plot(fit,type="l")

# update with the direct fit
fit1 <- update(fit, direct=TRUE)
lines(fit1,col=2)
print(max(abs(fit$y-fit1$y)))
```

Description

Spencer's 15 point rule is a weighted moving average operation for a sequence of observations equally spaced in time. The average at time `t` depends on the observations at times `t-7,...,t+7`.

Except for boundary effects, the function will reproduce polynomials up to degree 3.

Usage

```r
spence.15(y)
```
Arguments

y

Data vector of observations at equally spaced points.

Value

A vector with the same length as the input vector, representing the graduated (smoothed) values.

References


See Also

spence.21, spencer.

Examples

data(spencer)
yy <- spence.15(spencer$mortality)
plot(spencer$age, spencer$mortality)
lines(spencer$age, yy)

spence.21

Spencer's 21 point graduation rule.

Description

Spencer's 21 point rule is a weighted moving average operation for a sequence of observations equally spaced in time. The average at time t depends on the observations at times t-11,...,t+11. Except for boundary effects, the function will reproduce polynomials up to degree 3.

Usage

spence.21(y)

Arguments

y

Data vector of observations at equally spaced points.

Value

A vector with the same length as the input vector, representing the graduated (smoothed) values.

References

spencer

See Also

spence, spencer.

Examples

data(spencer)
  yy <- spence.21(spencer$mortality)
  plot(spencer$age, spencer$mortality)
  lines(spencer$age, yy)

spencer  

Spencer's Mortality Dataset

Description

Observed mortality rates for ages 20 to 45.

Usage

data(spencer)

Format

Data frame with age and mortality variables.

Source

Spencer (1904).

References


stamp

Stamp Thickness Dataset

Description

Thicknesses of 482 postage stamps of the 1872 Hidalgo issue of Mexico.

Usage

data(stamp)
Format

Data frame with thick (stamp thickness) and count (number of stamps) variables.

Source

Izenman and Sommer (1988).

References


store

Save S functions.

Description

I've gotta keep track of this mess somehow!

Usage

store(data=FALSE, grand=FALSE)

Arguments

data whether data objects are to be saved.
grand whether everything is to be saved.

summary.gcvplot

Summary method for a gcvplot structure.

Description

Computes a short summary for a generalized cross-validation plot structure.

Usage

## S3 method for class 'gcvplot'
summary(object, ...)

Arguments

object A gcvplot structure produced by a call to gcvplot, cpplot e.t.c.
... arguments to and from other methods.
Value

A matrix with two columns; one row for each fit computed in the `gcvplot` call. The first column is the fitted degrees of freedom; the second is the GCV or other criterion computed.

See Also

`locfit`, `gcv`, `gcvplot`

Examples

data(ethanol)
summary(gcvplot(NOx~E, data=ethanol, alpha=seq(0.2, 1.0, by=0.05)))

summary.locfit  Print method for a locfit object.

Description

Prints a short summary of a "locfit" object.

Usage

```r
## S3 method for class 'locfit'
summary(object, ...)
```

Arguments

- `object`  locfit object.
- `...`  arguments passed to and from methods.

Value

A `summary.locfit` object, containing a short summary of the `locfit` object.
**summary.preplot.locfit**

*Summary method for a preplot.locfit object.*

**Description**

Prints a short summary of a "preplot.locfit" object.

**Usage**

```r
## S3 method for class 'preplot.locfit'
summary(object, ...)  
```

**Arguments**

- `object`: preplot.locfit object.
- `...`: arguments passed to and from methods.

**Value**

The fitted values from a preplot.locfit object.

---

**trimod**

*Generated sample from a bivariate trimodal normal mixture*

**Description**

This is a random sample from a mixture of three bivariate standard normal components; the sample was used for the examples in Loader (1996).

**Format**

Data frame with 225 observations and variables x0, x1.

**Source**

Randomly generated in S.

**References**

Description

xbar() is an evaluation structure for locfit.raw(), evaluating the fit at a single point, namely, the average of each predictor variable.

Usage

xbar()
Index

*Topic datasets
ais, 5
bad, 7
border, 7
chemdiab, 8
claw54, 8
cldem, 9
clttest, 9
cltrain, 10
c02, 10
diab, 15
ethanol, 16
gam.slist, 19
geyser, 20
deyser.round, 21
heart, 22
insect, 23
iris, 23
kangaroo, 24
livmet, 34
mcyc, 45
mine, 46
mmsamp, 46
morths, 47
penny, 48
spencer, 71
stamp, 71
trimod, 74

*Topic htest
aic, 4
aicplot, 4
cp, 11
cpplot, 12
gcv, 19
gcvplot, 20
deb, 26
dcv, 28
dcvplot, 28
dlcv, 43

*Topic math
expit, 16

*Topic methods
plot.gcvplot, 49
plot.locfit, 50
plot.locfit.1d, 52
plot.locfit.2d, 52
plot.locfit.3d, 53
plot.scb, 54
preplot.locfit.raw, 58
print.gcvplot, 59
print.locfit, 60
print.preplot.locfit, 61
print.scb, 61
print.summary.locfit, 62
summary.gcvplot, 72
summary.locfit, 73
summary.preplot.locfit, 74

*Topic models
ang, 6
cpar, 11
formula.locfit, 18
gam.1f, 18
left, 29
lf, 30
locfit.matrix, 37
lp, 42
right, 64

*Topic smooth
crit, 13
dat, 14
density.1f, 14
fitted.locfit, 17
hatmatrix, 22
kappa0, 25
INDEX

km.mrl, 27
lfeval, 31
lfgrid, 31
lfknots, 32
lflim, 32
lfmarg, 33
lines.locfit, 34
locfit, 35
locfit.censor, 36
locfit.quasi, 38
locfit.raw, 39
locfit.robust, 41
none, 47
plot.eval, 48
plot.lfeval, 50
plot.preplot.locfit, 54
plotbyfactor, 55
points.locfit, 56
predict.locfit, 56
preplot.locfit, 57
print.lfeval, 60
rbox, 62
residuals.locfit, 64
rv, 65
rva, 66
scb, 66
smooth.lf, 68
spence.15, 69
spence.21, 70
store, 72
xbar, 75

aic, 4, 4, 5, 12
aicplot, 4, 4, 49
ais, 5
ang, 6, 42

bad, 7
border, 7

chendiab, 8
claw54, 8
cldem, 9
cldtest, 9
clttrain, 10
co2, 10
cp, 11, 12
cpar, 11
cpplot, 11, 12, 49, 72
crit, 13, 13, 25, 51, 58
crit<- (crit), 13
dat, 14, 40
density.lf, 14, 69
diab, 15
eethanol, 16
expit, 16
fitted.locfit, 17, 37, 38, 64
formula.locfit, 18
gam.lf, 18, 30
gam.slist, 19
gcv, 5, 12, 19, 20, 29, 45, 49, 73
gcvplot, 19, 20, 49, 59, 72, 73
geyser, 20, 21
geyser.round, 21
hatmatrix, 22
heart, 22
insect, 23
iris, 23
kanga, 24
kappa0, 13, 25, 51, 58
kdeb, 26
km.mrl, 27, 37
lcv, 28, 28, 29
lcvplot, 28, 28, 49, 68
left, 29, 42, 65
lf, 18, 19, 30
lfeval, 31, 50, 60
lfgrid, 15, 31, 40
lfknots, 32
lflim, 32
lfmarg, 33, 57
lines, 34
lines.locfit, 22, 34, 51
livmet, 34
llines.locfit (lines.locfit), 34
locfit, 4–6, 11–13, 15, 18–20, 22, 25, 28–34,
35, 36–39, 41–43, 45, 48–54, 56–60,
63, 65–69, 73
locfit.censor, 27, 36
locfit.matrix, 37
locfit.quasi, 38
locfit.raw, 4, 5, 11, 12, 14, 15, 18-20, 22, 28-31, 35-38, 39, 41-43, 45, 47, 49, 62, 67-69, 75

locfit.robust, 41
lp, 6, 29, 40, 41, 42, 44, 65
lscv, 43, 44, 45
lscv.exact, 43, 44
lscvplot, 43, 44

mcyc, 45
mine, 46
mmsamp, 46
morths, 47

none, 40, 47

penny, 48
plot, 49, 52
plot.eval, 48
plot.gcvplot, 5, 12, 20, 29, 45, 49, 59
plot.lfeval, 31, 50, 60
plot.locfit, 13, 25, 33, 34, 50, 52-54, 56-58, 66
plot.locfit.1d, 22, 50, 51, 52, 54
plot.locfit.2d, 22, 50, 51, 52, 54
plot.locfit.3d, 22, 50, 51, 53, 54
plot.preplot.locfit, 54
plot.scb, 54, 67
plotbyfactor, 55
points, 56
points.locfit, 56
predict.locfit, 22, 51, 56, 57-59, 61
preplot.locfit, 50-54, 57, 58, 59, 61
preplot.locfit.raw, 58
print.gcvplot, 59
print.lfeval, 31, 50, 60
print.locfit, 60
print.preplot.locfit, 61
print.scb, 61, 67
print.summary.locfit, 62

rbox, 40, 62
regband, 63
residuals.locfit, 17, 38, 64, 64
right, 29, 64
rv, 40, 51, 58, 65, 66
rv<=, 65
rv<- (rva), 66
rva, 66