Package 'fsdaR'

December 12, 2017

Date 2017-12-12

Title Robust Data Analysis Through Monitoring and Dynamic Visualization

Version 0.2-21

Author Marco Riani<fsda@unipr.it>, Andrea Cerioli <fsda@unipr.it>, Aldo Corbellini<fsda@unipr.it>,
Domenico Perrotta <fsda@unipr.it>, Francesca Torti <fsda@unipr.it>, Emmanuele Sordini<fsda@unipr.it>,
Valentin Todorov <valentin.todorov@chello.at>

Description Provides interface to the 'MATLAB' toolbox 'Flexible Statistical Data Analysis (FSDA)' which is a comprehensive and computationally efficient software package for robust statistics in regression, multivariate and categorical data analysis. The current R version implements tools for regression: (forward search, S- and MM-estimation, least trimmed squares (LTS) and least median of squares (LMS)). The distinctive feature of our regression package relies in the possibility of monitoring the statistics of interest as function of breakdown point, efficiency or subset size, depending on the estimator. This is accompanied by a rich set of graphical features, such as dynamic brushing, linking, particularly useful for exploratory data analysis.

Maintainer Valentin Todorov <valentin.todorov@chello.at>

Depends robustbase (>= 0.92-1)

Imports rJava, methods, stats4

Suggests MASS

SystemRequirements MATLAB

LazyLoad yes

LazyData yes

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Date/Publication 2017-12-12 14:30:13 UTC

R topics documented:

`fsdalms.object` *Description of* `fsdalms` *Objects*

Description

An object of class `fsdalms.object` holds information about the result of a call to `fsreg`.

Value

The object itself is basically a `list` with the following components:

`rew`	If `rew=TRUE` all subsequent output refers to reweighted else no reweighting is done.
`beta`	p-by-1 vector containing the estimated regression parameters.
`bs`	p x 1 vector containing the units forming subset associated with bLMS (bLTS).
`residuals`	residuals.
`scale`	scale estimate of the residuals.

weights	Vector like y containing weights. The elements of this vector are 0 or 1. These weights identify the h observations which are used to compute the final LTS (LMS) estimate. sum(weights)=h if there is not a perfect fit otherwise sum(weights) can be greater than h
h	The number of observations that have determined the LTS (LMS) estimator, i.e. the value of h.
outliers	vector containing the list of the units declared as outliers using confidence level specified in input scalar conflev.
conflev	confidence level which is used to declare outliers. Remark: conflev will be used to draw the horizontal lines (confidence bands) in the plots. Default value is 0.975
singsub	Number of subsets wihtout full rank. Notice that if this number is greater than 0.1*(number of subsamples) a warning is produced
X	the data matrix X
y	the response vector y

The object has class "fsdalms".

Examples

```
(out <- fsreg(Y~., data=hbk, method="LMS"))
class(out)
summary(out)
```

`fsdalts.object` *Description of* `fsdalts` *Objects*

Description

An object of class `fsdalts.object` holds information about the result of a call to `fsreg`.

Value

The object itself is basically a `list` with the following components:

rew	If rew=TRUE all subsequent output refers to reweighted else no reweighting is done.
beta	p-by-1 vector containing the estimated regression parameters.
bs	p x 1 vector containing the units forming subset associated with bLMS (bLTS).
residuals	residuals.
scale	scale estimate of the residuals.
weights	Vector like y containing weights. The elements of this vector are 0 or 1. These weights identify the h observations which are used to compute the final LTS (LMS) estimate. sum(weights)=h if there is not a perfect fit otherwise sum(weights) can be greater than h

h	The number of observations that have determined the LTS (LMS) estimator, i.e. the value of h.
outliers	vector containing the list of the units declared as outliers using confidence level specified in input scalar conflev.
conflev	confidence level which is used to declare outliers. Remark: conflev will be used to draw the horizontal lines (confidence bands) in the plots. Default value is 0.975
singsub	Number of subsets wihtout full rank. Notice that if this number is greater than 0.1*(number of subsamples) a warning is produced
X	the data matrix X
y	the response vector y

The object has class "fsdalts".

Examples

```
(out <- fsreg(Y~., data=hbk, method="LTS"))
class(out)
summary(out)
```

`fsr.object` *Description of* `fsr` *Objects*

Description

An object of class `fsr.object` holds information about the result of a call to `fsreg`.

Value

The object itself is basically a `list` with the following components:

beta	p-by-1 vector containing the estimated regression parameters (in step n-k).
scale	scalar containing the estimate of the scale (sigma).
residuals	residuals.
fittedvalues	fitted values.
outliers	kx1 vector containing the list of the k units declared as outliers or NULL if the sample is homogeneous.
mdr	(n-init) x 2 matrix 1st col = fwd search index, 2nd col = value of minimum deletion residual in each step of the fwd search
Un	(n-init) x 11 matrix which contains the unit(s) included in the subset at each step of the fwd search. REMARK: in every step the new subset is compared with the old subset. Un contains the unit(s) present in the new subset but not in the old one. Un(1,2) for example contains the unit included in step init+1. Un(end,2) contains the units included in the final step of the search.

nout	2 x 5 matrix containing the number of times mdr went out of particular quantiles. First row contains quantiles 1 99 99.9 99.99 99.999. Second row contains the frequency distribution.
constr	This output is produced only if the search found at a certain step is a non singular matrix X. In this case the search run in a constrained mode, that is including the units which produced a singular matrix in the last n-constr steps. out.constr is a vector which contains the list of units which produced a singular X matrix.
X	the data matrix X
y	the response vector y

The object has class "fsr".

Examples

```
(out <- fsreg(Y~., data=hbk, method="FS"))
class(out)
summary(out)
```

`fsreda.object` *Description of* `fsreda` *Objects*

Description

An object of class `fsreda.object` holds information about the result of a call to `fsreg`.

Value

The object itself is basically a `list` with the following components:

RES	n x (n-init+1) matrix containing the monitoring of scaled residuals: the first row is the residual for the first unit, ..., n-th row is the residual for the n-th unit.
LEV	(n+1) x (n-init+1) matrix containing the monitoring of leverage: the first row is the leverage for the first unit, ..., n-th row is the leverage for the n-th unit.
BB	n x (n-init+1) matrix containing the information about the units belonging to the subset at each step of the forward search: first col contains indexes of the units forming subset in the initial step; ...; last column contains units forming subset in the final step (all units).
mdr	n-init x 3 matrix which contains the monitoring of minimum deletion residual or (m+1)-ordered residual at each step of the forward search: first col is the fwd search index (from init to n-1); 2nd col = minimum deletion residual; 3rd col = (m+1)-ordered residual. Remark: these quantities are stored with sign, that is the min deletion residual is stored with negative sign if it corresponds to a negative residual.
msr	n-init+1 x 3 matrix which contains the monitoring of maximum studentized residual or m-th ordered residual: first col is the fwd search index (from init to n); 2nd col = maximum studentized residual; 3rd col = (m)-ordered studentized residual.

`nor` (n-init+1) x 4 matrix containing the monitoring of normality test in each step of the forward search: first col = fwd search index (from init to n); 2nd col = Asymmetry test; 3rd col = Kurtosis test; 4th col = Normality test.

`Bols` (n-init+1) x (p+1) matrix containing the monitoring of estimated beta coefficients in each step of the forward search.

`S2` (n-init+1) x 4 matrix containing the monitoring of S2 or R2 in each step of the forward search:

1. 1st col = fwd search index (from init to n);
2. 2nd col = monitoring of S2;
3. 3rd col = monitoring of R2;
4. 4th col = monitoring of rescaled S2.

In this case the estimated of s2 at step m is divided by the consistency factor (to make the estimate asymptotically unbiased).

`coo` (n-init+1) x 3 matrix containing the monitoring of Cook or modified Cook distance in each step of the forward search:

1. 1st col = fwd search index (from init to n);
2. 2nd col = monitoring of Cook distance;
3. 3rd col = monitoring of modified Cook distance.

`Tols` (n-init+1) x (p+1) matrix containing the monitoring of estimated t-statistics (as specified in option input 'tstat') in each step of the forward search

`Un` (n-init) x 11 matrix which contains the unit(s) included in the subset at each step of the fwd search.

REMARK: in every step the new subset is compared with the old subset. Un contains the unit(s) present in the new subset but not in the old one Un(1,2) for example contains the unit included in step init+1 Un(end,2) contains the units included in the final step of the search.

`betaINT` Confidence intervals for the elements of β. betaINT is a (n-init+1)-by-2*length(confint)-by-p 3D array. Each third dimension refers to an element of beta:

1. `betaINT[,,1]` is associated with first element of beta;
2. ...;
3. `betaINT[,,p]` is associated with last element of beta.

The first two columns contain the lower and upper confidence limits associated with conflev(1). Columns three and four contain the lower and upper confidence limits associated with conflev(2); ...; The last two columns contain the lower and upper confidence limits associated with conflev(end). For example `betaINT[,3:4,5]` contain the lower and upper confidence limits for the fifth element of beta using confidence level specified in the second element of input option `conflev`.

`sigma2INT` confidence interval for s2.

1. 1st col = fwd search index;
2. 2nd col = lower confidence limit based on conflev(1);
3. 3rd col = upper confidence limit based on conflev(1);
4. 4th col = lower confidence limit based on conflev(2);

5. 5th col = upper confidence limit based on conflev(2);
6. ...
7. penultimate col = lower confidence limit based on conflev(end);
8. last col = upper confidence limit based on conflev(end).

`X` the data matrix X

`y` the response vector y

The object has class `"fsreda"`.

Examples

```
(out <- fsreg(Y~., data=hbk, method="FS", monitoring=TRUE))
class(out)
summary(out)
```

`FSReda_control` *Creates an* `FSReda_control` *object*

Description

Creates an object of class `FSReda_control` to be used with the `fsreg()` function, containing various control parameters.

Usage

```
FSReda_control(intercept = TRUE, init, nocheck = FALSE,
    tstat = c("trad", "scal"), conflev = c(0.95, 0.99))
```

Arguments

`intercept`	Indicator for constant term. Scalar. If `intercept=TRUE`, a model with constant term will be fitted (default), else, no constant term will be included.
`init`	Search initialization, scalar. It specifies the initial subset size to start monitoring exceedances of minimum deletion residual, if init is not specified it set equal to: `p+1`, if the sample size is smaller than 40 or `min(3*p+1,floor(0.5*(n+p+1)))`, otherwise. For example, if `init=100`, the procedure starts monitoring from step `m=100`.
`nocheck`	Check input arguments, scalar. If `nocheck=TRUE` no check is performed on matrix y and matrix `X`. Notice that y and `X` are left unchanged. In other words the additional column of ones for the intercept is not added. As default `nocheck=FALSE`.
`tstat`	The kind of t-statistics which have to be monitored. `tstat="trad"` implies monitoring of traditional t statistics (`out$Tols`). In this case the estimate of s2 at step m is based on s2m (notice that s2m«s2 when m/n is small) `tstat="scal"` (default) implies monitoring of rescaled t statistics. In this case the estimate of s2 at step m is based on s2m/vartruncnorm(m/n) where vartruncnorm(m/n) is the variance of the truncated normal distribution.

conflev — Confidence level which is used to declare units as outliers. Usually conflev=0.95, 0.975, 0.99 (individual alpha) or conflev=1-0.05/n, 1-0.025/n, 1-0.01/n (simultaneous alpha). Default value is 0.975.

Details

Creates an object of class FSReda_control to be used with the fsreg() function, containing various control parameters.

Value

An object of class "FSReda_control" which is basically a list with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also as FSR_control, MMreg_control and LXS_control

Examples

```
(out <- fsreg(Y~., data=hbk, method="FS", monitoring=TRUE,
    control=FSReda_control(tstat="scal")))
```

---

fsreg — *fsreg: an automatic outlier detection procedure in linear regression*

Description

An automatic outlier detection procedure in linear regression

Usage

```
fsreg(x, ...)

## S3 method for class 'formula'
fsreg(formula, data, subset, weights, na.action,
       model = TRUE, x.ret = FALSE, y.ret = FALSE,
       contrasts = NULL, offset, ...)

## Default S3 method:
fsreg(x, y, bsb, intercept = TRUE,
        family = c("homo", "hetero", "bayes"),
```

```
method = c("FS", "S", "MM", "LTS", "LMS"),
        monitoring = FALSE, control,
        ...)
```

Arguments

formula	a formula of the form `y ~ x1 + x2 + ...`.
data	data frame from which variables specified in `formula` are to be taken.
subset	an optional vector specifying a subset of observations to be used in the fitting process.
weights	an optional vector of weights to be used in the fitting process. **NOT USED YET**.
na.action	a function which indicates what should happen when the data contain NAs. The default is set by the `na.action` setting of `options`, and is `na.fail` if that is unset. The "factory-fresh" default is `na.omit`. Another possible value is NULL, no action. Value `na.exclude` can be useful.
model, x.ret, y.ret	logicals indicating if the model frame, the model matrix and the response are to be returned, respectively.
contrasts	an optional list. See the `contrasts.arg` of `model.matrix.default`.
offset	this can be used to specify an *a priori* known component to be included in the linear predictor during fitting. An `offset` term can be included in the formula instead or as well, and if both are specified their sum is used.
family	family of robust regression models, can be 'homo' for homoscedastic (same variance) regression model, 'hetero' for heteroskedastic regression model or 'bayes' Bayesian linear regression. The default is `family='homo'` for homoscedastic regression model.
method	robust regression estimation model, can be 'FS' for Forward search, 'S' for S regression, 'MM' for MM regression, 'LMS' or 'LTS'. The default is `method='FS'` for forward search estimation.
monitoring	wheather to perform monitoring for several quantities in each step of the forward search or for series of values of the breakdown point in case of S estimates or for series of values of the efficiency in case of MM estimates. Deafault is `monitoring=FALSE`. Enables to monitor several quantities in each step of the forward search
y	Response variable. Vector. Response variable, specified as a vector of length n, where n is the number of observations. Each entry in y is the response for the corresponding row of X. Missing values (NA's) and infinite values (Inf's) are allowed, since observations (rows) with missing or infinite values will automatically be excluded from the computations.
x	Predictor variables. Matrix. Matrix of explanatory variables (also called 'regressors') of dimension n x (p-1) where p denotes the number of explanatory variables including the intercept. Rows of X represent observations, and columns represent variables. By default, there is a constant term in the model, unless you explicitly remove it using input option `intercept=FALSE`, so do not include a

	column of 1s in X. Missing values (NA's) and infinite values (Inf's) are allowed, since observations (rows) with missing or infinite values will automatically be excluded from the computations.
`bsb`	Initial subset - vector of indices. If `bsb=0` (default) then the procedure starts with p units randomly chosen. If bsb is not 0 the search will start with `m0=length(bsb)`.
`intercept`	Indicator for constant term. Scalar. If `intercept=TRUE`, a model with constant term will be fitted (default), else, no constant term will be included.
`control`	A control object (SS) containing estimation options. If the control object is supplied, the parameters from it will be used. If parameters are passed also in the invocation statement, they will override the corresponding elements of the control object.
`...`	potential further arguments passed to lower level functions.

Value

Depending on the input parameters `family` and `method`, one of the following objects will be returned:

1. `fsr.object`
2. `sreg.object`
3. `mmreg.object`
4. `fsdalms.object`
5. `fsdalts.object`
6. `fsreda.object`
7. `sregeda.object`
8. `mmregeda.object`

Author(s)

FSDA team

References

Riani, M., Atkinson A.C., Cerioli A. (2009). Finding an unknown number of multivariate outliers. Journal of the Royal Statistical Society Series B, Vol. 71, pp. 201-221.

Examples

```
n <- 200
p <- 3

X <- matrix(data=rnorm(n*p), nrow=n, ncol=p)
y <- matrix(data=rnorm(n*1), nrow=n, ncol=1)
(out = fsreg(X, y))

## Now we use the formula interface:
```

```
(out1 = fsreg(y~X, control=FSR_control(plot=FALSE)))

## Or use the variables in a data frame
(out2 = fsreg(Y~., data=hbk, control=FSR_control(plot=FALSE)))

## let us compare to the LTS solution
(out3 = ltsReg(Y~., data=hbk))

## Now compute the model without intercept
(out4 = fsreg(Y~.-1, data=hbk, control=FSR_control(plot=FALSE)))

## And compare again with the LTS solution
(out5 = ltsReg(Y~.-1, data=hbk))

## using default (optional arguments)
(out6 = fsreg(Y~.-1, data=hbk, control=FSR_control(plot=FALSE, nsamp=1500, h=50)))
```

FSR_control *Creates an* FSR_control *object*

Description

Creates an object of class FSR_control to be used with the fsreg() function, containing various control parameters.

Usage

```
FSR_control(intercept = TRUE, h, nsamp = 1000, lms = 1, init, nocheck = FALSE,
    bonflev = "", msg = TRUE, bsbmfullrank = 1,
    plot = FALSE, bivarfit = "", multivarfit = "",
    labeladd = "", nameX, namey, ylim, xlim)
```

Arguments

intercept	Indicator for constant term. Scalar. If intercept=TRUE, a model with constant term will be fitted (default), else, no constant term will be included.
h	The number of observations that have determined the least trimmed squares estimator, scalar. h is an integer greater or equal than p but smaller then n. Generally if the purpose is outlier detection h=[0.5*(n+p+1)] (default value). h can be smaller than this threshold if the purpose is to find subgroups of homogeneous observations. In this function the LTS/LMS estimator is used just to initialize the search.
nsamp	Number of subsamples which will be extracted to find the robust estimator, scalar. If nsamp=0 all subsets will be extracted. They will be (n choose p). If the number of all possible subset is <1000 the default is to extract all subsets otherwise just 1000.

`lms` Criterion to use to find the initial subset to initialize the search (LMS, LTS with concentration steps, LTS without concentration steps or subset supplied directly by the user). The default value is 1 (Least Median of Squares is computed to initialize the search). On the other hand, if the user wants to initialze the search with LTS with all the default options for concentration steps then lms=2. If the user wants to use LTS without concentration steps, lms can be a scalar different from 1 or 2. If lms is a list it is possible to control a series of options for concentration steps (for more details see option `lms` inside `LXS_control`). If, on the other hand, the user wants to initialize the search with a prespecified set of units there are two possibilities:

1. lms can be a vector with length greater than 1 which contains the list of units forming the initial subset. For example, if the user wants to initialize the search with units 4, 6 and 10 then `lms=c(4, 6, 10)`;
2. lms is a struct which contains a field named bsb which contains the list of units to initialize the search. For example, in the case of simple regression through the origin with just one explanatory variable, if the user wants to initialize the search with unit 3 then `lms=list(bsb=3)`.

`init` Search initialization, scalar. It specifies the initial subset size to start monitoring exceedances of minimum deletion residual, if init is not specified it set equal to: `p+1`, if the sample size is smaller than 40 or `min(3*p+1,floor(0.5*(n+p+1)))`, otherwise. For example, if `init=100`, the procedure starts monitoring from step `m=100`.

`nocheck` Check input arguments, scalar. If `nocheck=TRUE` no check is performed on matrix `y` and matrix `X`. Notice that `y` and `X` are left unchanged. In other words the additional column of ones for the intercept is not added. As default `nocheck=FALSE`.

`bonflev` Option to be used if the distribution of the data is strongly non normal and, thus, the general signal detection rule based on consecutive exceedances cannot be used. In this case bonflev can be:

1. a scalar smaller than 1 which specifies the confidence level for a signal and a stopping rule based on the comparison of the minimum MD with a Bonferroni bound. For example if bonflev=0.99 the procedure stops when the trajectory exceeds for the first time the 99% bonferroni bound.
2. A scalar value greater than 1. In this case the procedure stops when the residual trajectory exceeds for the first time this value.

Default value is '', which means to rely on general rules based on consecutive exceedances.

`msg` Controls whether to display or not messages on the screen If `msg==1` (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.

`bsbmfullrank` How to behave in case subset at step m (say bsbm) produces a singular X. In other words, this options controls what to do when `rank(X[bsbm, ])` is smaller then number of explanatory variables. If `bsbmfullrank=1` (default) these units (whose number is say mnofullrank) are constrained to enter the search in the final n-mnofullrank steps else the search continues using as estimate of beta at step m the estimate of beta found in the previous step.

`plot` Plot on the screen. Scalar. If `plots=TRUE` the plot of minimum deletion residual with envelopes based on n observations and the scatterplot matrix with the outliers highlighted is produced. If `plots=2` the user can also monitor the intermediate plots based on envelope superimposition. If `plots=FALSE` (default) no plot is produced.

`bivarfit` Wheather to superimpose bivariate least square lines on the plot (if `plot=TRUE`. This option adds one or more least squares lines, based on SIMPLE REGRESSION of y on Xi, to the plots of y|Xi. The default is `bivarfit=FALSE`: no line is fitted. If `bivarfit=1`, a single OLS line is fitted to all points of each bivariate plot in the scatter matrix y|X. If `bivarfit=2`, two OLS lines are fitted: one to all points and another to the group of the genuine observations. The group of the potential outliers is not fitted. If `bivarfit=0` one OLS line is fitted to each group. This is useful for the purpose of fitting mixtures of regression lines. If `bivarfit='i1'` or `bivarfit='i2'`, etc. an OLS line is fitted to a specific group, the one with index 'i' equal to 1, 2, 3 etc. Again, useful in case of mixtures.

`multivarfit` Wheather to superimpose multivariate least square lines. This option adds one or more least square lines, based on MULTIVARIATE REGRESSION of y on X, to the plots of y|Xi. The default is `multivarfit=FALSE`: no line is fitted. If `bivarfit=1`, a single OLS line is fitted to all points of each bivariate plot in the scatter matrix y|X. The line added to the scatter plot y|Xi is avconst + Ci*Xi, where Ci is the coefficient of Xi in the multivariate regression and avconst is the effect of all the other explanatory variables different from Xi evaluated at their centroid (that is overline(y)'C)). If `multivarfit=2`, same action as with `multivarfit=1` but this time we also add the line based on the group of unselected observations (i.e. the normal units).

`labeladd` Add outlier labels in plot. If `labeladd=TRUE`, we label the outliers with the unit row index in matrices X and y. The default value is `labeladd=FALSE`, i.e. no label is added.

`nameX` Add variable labels in plot. A vector of strings of length p containing the labels of the variables of the regression dataset. If it is empty (default) the sequence `X1, ..., Xp` will be created automatically

`namey` Add response label. A string containing the label of the response

`ylim` Control y scale in plot. Vector with two elements controlling minimum and maximum on the y axis. Default is to use automatic scale.

`xlim` Control x scale in plot. Vector with two elements controlling minimum and maximum on the x axis. Default is to use automatic scale.

Details

Creates an object of class `FSR_control` to be used with the `fsreg()` function, containing various control parameters.

Value

An object of class `"FSR_control"` which is basically a `list` with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also Sreg_control, MMreg_control, LXS_control, FSReda_control, Sregeda_control and MMregeda_control.

Examples

```
(out <- fsreg(Y~., data=hbk, method="FS", control=FSR_control(h=56, nsamp=500, lms=2)))
summary(out)
```

---

levfwdplot *Plots the trajectories of the monitored scaled (squared) residuals*

Description

Plots the trajectories of the monitored scaled (squared) residuals

Usage

```
levfwdplot(out,
    xlim, ylim, xlab, ylab, main, lwd, lty, col, cex.lab, cex.axis,
    xvalues,
    fg.thresh, fg.unit, fg.labstep, fg.lwd, fg.lty, fg.col, fg.mark, fg.cex,
    bg.thresh, bg.style,
   xground=c("lev", "res"), tag, datatooltip, label, nameX, namey, msg, databrush,
    standard, fground, bground, ...)
```

Arguments

out	An object containing monitoring of leverage, fsreda.object. The needed elements of out are 1. LEV: matrix containing the leverage monitored in each step of the forward search. Every row is associated with a unit. This matrix can be created using function fsreg() with method="FS", monitoring=TRUE. 2. Un: (for FSR only) - matrix containing the order of entry in the subset of each unit (required only when datatooltip is true or databrush is not empty). 3. y: a vector containing the response (required only when option databrush is requested). 4. X: a matrix containing the explanatory variables (required only when option databrush is requested). 5. Bols: (n-init+1) x (p+1) matrix containing the estimated beta coefficients monitored in each step of the robust procedure (required only when option databrush is requested and suboption multivarfit is requested).

ylim	Control y scale in plot. Vector with two elements controlling minimum and maximum on the y axis. Default is to use automatic scale.
xlim	Control x scale in plot. Vector with two elements controlling minimum and maximum on the x axis. Default is to use automatic scale.
xlab	a title for the x axis
ylab	a title for the y axis
main	an overall title for the plot
lwd	The line width, a positive number, defaulting to 1
lty	The line type. Line types can either be specified as an integer (1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". The latter two are not supported by Matlab.
col	colors to be used for the highlighted units
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex
xvalues	values for the x axis. Numeric vector of `ncol(RES)` controlling the x axis coordinates. The default value of xvalues is `(nrow(RES) - ncol(RES) + 1):nrow(RES)`
fg.thresh	(alternative to fg.unit) numeric vector of length 1 or 2 which specifies the highlighted trajectories. If `length(fthresh) == 1` the highlighted trajectories are those of units that throughtout the search had at leat once a residual greater (in absolute value) than thresh. The default value is `fg.thresh=2.5`. If `length(fthresh) == 2` the highlighted trajectories are those of units that throughtout the search had a residual at leat once bigger than `fg.thresh[2]` or smaller than `fg.thresh[1]`.
fg.unit	(alternative to fg.thresh), vector containing the list of the units to be highlighted. If `fg.unit` is supplied, `fg.thresh` is ignored.
fg.labstep	numeric vector which specifies the steps of the search where to put labels for the highlighted trajectories (units). The default is to put the labels at the initial and final steps of the search. `flabstep=''` means no label.
fg.lwd	The line width for the highlighted trajectories (units). Default is 1.
fg.lty	The line type for the highlighted trajectories (units). Line types can either be specified as an integer (1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". The latter two are not supported by Matlab.
fg.col	colors to be used for the highlighted units.
fg.mark	Controlls whether to plot highlighted trajectories as symbols. if `fg.mark==TRUE` each line is plotted using a different symbol else no marker is used (default).
fg.cex	controls the font size of the labels of the trajectories in foreground.
bg.thresh	numeric vector of length 1 or 2 which specifies how to define the unimmportant trajectories. Unimmportant trajectories will be plotted using a colormap, in greysh or will be hidden. If `length(thresh) == 1` the irrelevant units

are those which always had a residual smaller (in absolute value) than thresh. If `length(bthresh) == 2` the irrelevant units are those which always had a residual greater than bthresh(1) and smaller than bthresh(2). The default is: `bg.thresh=2.5` if `n > 100` and `bg.thresh=-Inf` if `n <= 100` i.e. to treat all trajectories as important if `n <= 100` and, if `n > 100`, to reduce emphasis only to trajectories having in all steps of the search a value of scaled residual smaller than 2.5.

`bg.style` specifies how to plot the unimportant trajectories as defined in option bthresh.

1. `bg.style="faint"`: unimportant trajectories are plotted using a colormap.
2. `bg.style="hide"`: unimportant trajectories are hidden.
3. `bg.style="greyish"`: unimportant trajectories are displayed in a faint grey.

When `n>100` the default option is `bg.style='faint'`. When `n <= 100` and `bg.thresh == -Inf` option bstyle is ignored. Remark: bground=" is equivalent to -Inf that is all trajectories are considered relevant.

`tag` Plot handle. String which identifies the handle of the plot which is about to be created. The default is to use tag 'pl_resfwd'. Notice that if the program finds a plot which has a tag equal to the one specified by the user, then the output of the new plot overwrites the existing one in the same window else a new window is created.

`xground` trajectories to highlight in connection with resfwdplot. If `xground="lev"` (default), the levfwdplot trajectories are put in foreground or in background depending on the leverage values. If `xground="res"`, the levfwdplot trajectories are put in foreground or in background depending on the residual values. See options `bg.thresh` and `fg.thresh`.

`datatooltip` Interactive clicking. It is inactive if this parameter is missing or empty. The default is `datatooltip=TRUE`, i.e. the user can select with the mouse an individual residual trajectory in order to have information about the corresponding unit. The information displayed depends on the estimator in use.

For example for class `fsreda.object` the information concerns the label and the step of the search in which the unit enters the subset. If datatooltip is a list it may contain the following fields:

1. DisplayStyle determines how the data cursor displays. Possible values are `'datatip'` and `'window'` (default). `'datatip'` displays data cursor information in a small yellow text box attached to a black square marker at a data point you interactively select. `'window'` displays data cursor information for the data point you interactively select in a floating window within the figure.
2. SnapToDataVertex: specifies whether the data cursor snaps to the nearest data value or is located at the actual pointer position. Possible values are `SnapToDataVertex='on'` (default) and `SnapToDataVertex='off'`.
3. LineColor: controls the color of the trajectory selected with the mouse. It can be an RGB triplet of values between 0 and 1, or character vector indicating a color name. Note that a RGB vector can be conveniently chosen with our MATLAB class FSColor, see documentation.

4. SubsetLinesColor: enables to control the color of the trajectories of the units that are in the subset at a given step of the search (if `levfwdplot()` is applied to an object of class `fsreda.object`) or have a weight greater than 0.9 (if `levfwdplot()` is applied to an object of class `sregeda.object` or `mmregeda.object`). This can be done (repeatedly) with a left mouse click in proximity of the step of interest. A right mouse click will terminate the selection by marking with a up-arrow the step corresponding to the highlighted lines. The highlighted lines by default are in red, but a different color can be specified as RGB triplet or character of color name. Note that a RGB vector can be conveniently chosen with our MATLAB class FSColor, see documentation. By default `SubsetLinesColor=""`, i.e. the modality is not active. Any initialization for `SubsetLinesColor` which cannot be interpreted as RGB vector will be converted to blue, i.e. `SubsetLinesColor` will be forced to be [0 0 1]. If `SubsetLinesColor` is not empty the previous option `LineColor` is ignored.

`label` Character vector containing the labels of the units (optional argument used when `datatooltip=TRUE`. If this field is not present labels row1, ..., rown will be automatically created and included in the pop up datatooltip window).

`nameX` Add variable labels in plot. A vector of strings of length p containing the labels of the variables of the regression dataset. If it is empty (default) the sequence `X1, ..., Xp` will be created automatically

`namey` Add response label. A string containing the label of the response

`msg` Controls whether to display or not messages on the screen If `msg==1` (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.

`databrush` interactive mouse brushing. If databrush is missing or empty (default), no brushing is done. The activation of this option (databrush is a scalar or a list) enables the user to select a set of trajectories in the current plot and to see them highlighted in the y|X plot, i.e. a matrix of scatter plots of y against each column of X, grouped according to the selection(s) done by brushing. If the plot y|X does not exist it is automatically created. In addition, brushed units are automatically highlighted in the minimum deletion residual plot if it is already open. The extension to the following plots will be available in future versions of the toolbox:

1. monitoring leverage plot;
2. maximum studentized residual;
3. s^2 and R^2;
4. Cook distance and modified Cook distance;
5. deletion t statistics.

Note that the window style of the other figures is set equal to that which contains the monitoring residual plot. In other words, if the monitoring residual plot is docked all the other figures will be docked too

If `databrush=TRUE` the default selection tool is a rectangular brush and it is possible to brush only once (that is persist='').

If `databrush=list(...)`, it is possible to use all optional arguments of function `selectdataFS()` and the following optional argument:

1. persist. Persist is an empty value or a character containing 'on' or 'off'. The default value is `persist=""`, that is brushing is allowed only once. If `persist="on"` or `persis="off"` brushing can be done as many time as the user requires. If `persist='on'` then the unit(s) currently brushed are added to those previously brushed. It is possible, every time a new brushing is done, to use a different color for the brushed units. If `persist='off'` every time a new brush is performed units previously brushed are removed.
2. bivarfit. This option adds one or more least square lines based on SIMPLE REGRESSION to the plots of y|X, depending on the selected groups.
3. bivarfitWheather to superimpose bivariate least square lines on the plot (if `plot=TRUE`. This option adds one or more least squares lines, based on SIMPLE REGRESSION of y on Xi, to the plots of y|Xi. The default is `bivarfit=FALSE`: no line is fitted. If `bivarfit=1`, a single OLS line is fitted to all points of each bivariate plot in the scatter matrix y|X. If `bivarfit=2`, two OLS lines are fitted: one to all points and another to the group of the genuine observations. The group of the potential outliers is not fitted. If `bivarfit=0` one OLS line is fitted to each group. This is useful for the purpose of fitting mixtures of regression lines. If `bivarfit='i1'` or `bivarfit='i2'`, etc. an OLS line is fitted to a specific group, the one with index 'i' equal to 1, 2, 3 etc. Again, useful in case of mixtures.
4. multivarfitWheather to superimpose multivariate least square lines. This option adds one or more least square lines, based on MULTIVARIATE REGRESSION of y on X, to the plots of y|Xi. The default is `multivarfit=FALSE`: no line is fitted. If `bivarfit=1`, a single OLS line is fitted to all points of each bivariate plot in the scatter matrix y|X. The line added to the scatter plot y|Xi is avconst + Ci*Xi, where Ci is the coefficient of Xi in the multivariate regression and avconst is the effect of all the other explanatory variables different from Xi evaluated at their centroid (that is overline(y)'C)). If `multivarfit=2`, same action as with `multivarfit=1` but this time we also add the line based on the group of unselected observations (i.e. the normal units).
5. labeladdAdd outlier labels in plot. If `labeladd=TRUE`, we label the outliers with the unit row index in matrices X and y. The default value is `labeladd=FALSE`, i.e. no label is added.

`standard` (MATLAB-style arguments) appearance of the plot in terms of xlim, ylim, axes labels and their font size style, color of the lines, etc.

`fground` MATLAB-style arguments for the fground trajectories in foregroud.

`bground` MATLAB-style arguments for the fground trajectories in backgroud.

`...` potential further arguments passed to lower level functions.

Details

No details

Value

No value returned

Author(s)

FSDA team

Examples

```
## Not run:

n <- 100
y <- rnorm(n)
X <- matrix(rnorm(n*4), nrow=n)

out <- fsreg(y~X, method="LTS")
out <- fsreg(y~X, method="FS", bsb=out$bs, monitoring=TRUE)
levfwdplot(out)


## End(Not run)
```

---

loyalty *Loyalty data*

Description

The loyalty data consist of 509 observations on the behaviour of customers with loyalty cards from a supermarket chain in Northern Italy. The response y is the amount in euros spent at the shop over six months and the explanatory variables are: X1, the number of visits to the supermarket in the six month period; X2, the age of the customer; X3, the number of members of the customers' family. To find out more about this data set please see Atkinson and Riani (2006), JCGS

Usage

```
data("loyalty")
```

Format

A data frame with 509 observations on the following 4 variables.

`visits` the number of visits to the supermarket in the six month period

`age` the age of the customer

`family` the number of members of the customers' family

`amount_spent` the amount in euros spent at the shop over six months

Details

To find out more about this data set please see Atkinson and Riani (2006), JCGS

Source

The data are themselves a random sample from a larger database. The sample of 509 observations is available at www.riani.it/trimmed.

References

Atkinson, A. and Riani, M (2006) Distribution Theory and Simulations for Tests of Outliers in Regression, *Journal of Computational and Graphical Statistics*, **15** 2, pp 460–476.

Examples

```
data(loyalty)
```

LXS_control *Creates an* LSX_control *object*

Description

Creates an object of class LXS_control to be used with the fsreg() function, containing various control parameters.

Usage

```
LXS_control(intercept = TRUE, lms, h, bdp, nsamp, rew = FALSE, conflev = 0,
    msg = TRUE, nocheck = FALSE, nomes = FALSE, plot = FALSE)
```

Arguments

intercept	Indicator for constant term. Scalar. If intercept=TRUE, a model with constant term will be fitted (default), else, no constant term will be included.
lms	Criterion to use to find the initial subset to initialize the search (LMS, LTS with concentration steps, LTS without concentration steps or subset supplied directly by the user). The default value is 1 (Least Median of Squares is computed to initialize the search). On the other hand, if the user wants to initialze the search with LTS with all the default options for concentration steps then lms=2. If the user wants to use LTS without concentration steps, lms can be a scalar different from 1 or 2. If lms is a list it is possible to control a series of options for concentration steps (for more details see option lms inside LXS_control). If, on the other hand, the user wants to initialize the search with a prespecified set of units there are two possibilities: 1. lms can be a vector with length greater than 1 which contains the list of units forming the initial subset. For example, if the user wants to initialize the search with units 4, 6 and 10 then lms=c(4, 6, 10); 2. lms is a struct which contains a field named bsb which contains the list of units to initialize the search. For example, in the case of simple regression through the origin with just one explanatory variable, if the user wants to initialize the search with unit 3 then lms=list(bsb=3).

h	The number of observations that have determined the least trimmed squares estimator, scalar. h is an integer greater or equal than p but smaller then n. Generally if the purpose is outlier detection h=[0.5*(n+p+1)] (default value). h can be smaller than this threshold if the purpose is to find subgroups of homogeneous observations. In this function the LTS/LMS estimator is used just to initialize the search.
bdp	Breakdown point. It measures the fraction of outliers the algorithm should resist. In this case any value greater than 0 but smaller or equal than 0.5 will do fine. If on the other hand the purpose is subgroups detection then bdp can be greater than 0.5. In any case however n*(1-bdp) must be greater than p. If this condition is not fulfilled an error will be given. Please specify h or bdp not both.
nsamp	Number of subsamples which will be extracted to find the robust estimator, scalar. If nsamp=0 all subsets will be extracted. They will be (n choose p). If the number of all possible subset is <1000 the default is to extract all subsets otherwise just 1000.
rew	LXS reweighted - if rew=1 the reweighted version of LTS (LMS) is used and the output quantities refer to the reweighted version else no reweighting is performed (default).
conflev	Confidence level which is used to declare units as outliers, usually conflev=0.95, 0.975, 0.99 (individual alpha) or 1-0.05/n, 1-0.025/n, 1-0.01/n (simultaneous alpha). Default value is 0.975.
msg	Controls whether to display or not messages on the screen If msg==1 (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.
nocheck	Check input arguments, scalar. If nocheck=TRUE no check is performed on matrix y and matrix X. Notice that y and X are left unchanged. In other words the additional column of ones for the intercept is not added. As default nocheck=FALSE.
nomes	It controls whether to display or not on the screen messages about estimated time to compute LMS (LTS). If nomes is equal to 1 no message about estimated time to compute LMS (LTS) is displayed, else if nomes is equal to 0 (default), a message about estimated time is displayed.
plot	Plot on the screen. Scalar. If plots=TRUE the plot of minimum deletion residual with envelopes based on n observations and the scatterplot matrix with the outliers highlighted is produced. If plots=2 the user can also monitor the intermediate plots based on envelope superimposition. If plots=FALSE (default) no plot is produced.

Details

Creates an object of class FSR_control to be used with the fsreg() function, containing various control parameters.

Value

An object of class "LXS_control" which is basically a list with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also as Sreg_control, MMreg_control and FSR_control

Examples

```
(out <- fsreg(Y~., data=hbk, method="LMS", control=LXS_control(h=56, nsamp=500, lms=2)))
```

---

mdrplot *Plots the trajectory of minimum deletion residual (mdr)*

Description

Plots the trajectory of minimum deletion residual (mdr).

Usage

```
mdrplot(out, quant = c(0.01, 0.5, 0.99), exact = 0, sign = TRUE,
    mplus1 = FALSE, envm,
    xlim, ylim, xlab, ylab, main,
    lwdenv, lwd, cex.lab, cex.axis,
    tag, datatooltip, label, nameX, namey, databrush,
    ...)
```

Arguments

out	An object returned by FSReda() (see FSReda_control). The needed elements of out are 1. mdr: Minimum deletion residual. A matrix containing the monitoring of minimum deletion residual in each step of the forward search. The first column of mdr must contain the fwd search index. 2. Un: (for FSR only) - matrix containing the order of entry in the subset of each unit (required only when datatooltip is true or databrush is not empty). 3. y: a vector containing the response (required only when option databrush is requested). 4. X: a matrix containing the explanatory variables (required only when option databrush is requested). 5. Bols: (n-init+1) x (p+1) matrix containing the estimated beta coefficients monitored in each step of the robust procedure (required only when option databrush is requested and suboption multivarfit is requested).

quant	Quantiles for which envelopes have to be computed. The default is to produce 1%, 50% and 99% envelopes. In other words the default is quant=c(0.01, 0.5, 0.99).
exact	Exact of approximate cdf for envelope calculation. If it exact=TRUE the calculation of the quantiles of the T and F distribution is based on functions finv() and tinv() from the Matlab statistics toolbox, otherwise the calculations of the former quantiles is based on functions invcdff() and invcdft(). The solution has a tolerance of 1e-8. Remark: the use of functions tinv and finv is more precise but requires more time. The default value is exact=0 (approximate solution).
sign	Wheather to use MDR with sign: if sign=TRUE (default) we distinguish steps for which minimum deletion residual was associated with positive or negative value of the residual. Steps associated with positive values of mdr are plotted in black, while other steps are plotted in red.
mplus1	Wheather to plot the (m+1)-th order statistic. Specifies if it is necessary to plot the curve associated with (m+1)-th order statistic.
envm	Sample size for drawing enevlopes. Specifies the size of the sample which is used to superimpose the envelope. The default is to add an envelope based on all the observations (size n envelope).
ylim	Control y scale in plot. Vector with two elements controlling minimum and maximum on the y axis. Default is to use automatic scale.
xlim	Control x scale in plot. Vector with two elements controlling minimum and maximum on the x axis. Default is to use automatic scale.
xlab	a title for the x axis
ylab	a title for the y axis
main	an overall title for the plot
lwdenv	Controls the width of the lines associated with the envelopes, default is lvdenv=1.
lwd	Controls the linewidth of the curve which contains the monitoring of minimum deletion residual.
cex.lab	The magnification to be used for x and y labels relative to the current setting of cex
cex.axis	The magnification to be used for axis annotation relative to the current setting of cex
tag	Plot handle. String which identifies the handle of the plot which is about to be created. The default is to use tag 'pl_mdr'. Notice that if the program finds a plot which has a tag equal to the one specified by the user, then the output of the new plot overwrites the existing one in the same window else a new window is created.
datatooltip	
label	Character vector containing the labels of the units (optional argument used when datatooltip=TRUE. If this field is not present labels row1, ..., rown will be automatically created and included in the pop up datatooltip window).
nameX	Add variable labels in plot. A vector of strings of length p containing the labels of the variables of the regression dataset. If it is empty (default) the sequence X1, ..., Xp will be created automatically

namey	Add response label. A string containing the label of the response
databrush	
...	potential further arguments passed to lower level functions.

Details

No details

Value

No value returned

Author(s)

FSDA team

Examples

```
## Not run:

n <- 100
y <- rnorm(n)
X <- matrix(rnorm(n*4), nrow=n)

out <- fsreg(y~X, method="LTS")
out <- fsreg(y~X, method="FS", bsb=out$bs, monitoring=TRUE)
mdrplot(out)

## End(Not run)
```

---

mmreg.object *Description of mmreg Objects*

Description

An object of class mmreg.object holds information about the result of a call to fsreg with method="MM".

Value

The object itself is basically a list with the following components:

beta	p-by-1 vector containing the MM estimate of regression coefficients.
auxscale	scalar, S estimate of the scale (or supplied external estimate of scale, if option InitialEst is not empty).
residuals	residuals.
fittedvalues	fitted values.
weights	n x 1 vector. Weights assigned to each observation.

Sbeta	p x 1 vector containing S estimate of regression coefficients (or supplied initial external estimate of regression coefficients, if option InitialEst is not empty)
Ssingsub	Number of subsets without full rank in the S preliminary part. Notice that out.singsub > 0.1*(number of subsamples) produces a warning.
outliers	kx1 vector containing the list of the k units declared as outliers or NULL if the sample is homogeneous.
conflev	Confidence level which is used to declare units as outliers. Usually `conflev=0.95, 0.975, 0.99` (individual alpha) or `conflev=1-0.05/n, 1-0.025/n, 1-0.01/n` (simultaneous alpha). Default value is 0.975
rhofunc	Specifies the rho function which has been used to weight the residuals. If a different rho function is specified for S and MM loop then insted of `rhofunc` we will have `rhofuncS` and `rhofuncMM`.
rhofuncparam	Vector which contains the additional parameters for the specified rho function which has been used. For hyperbolic rho function the value of k =sup CVC. For Hampel rho function the parameters a, b and c. If a different rho function is specified for S and MM loop then insted of `rhofuncparam` we will have `rhofuncparamS` and `rhofuncparamMM`.
X	the data matrix X
y	the response vector y

The object has class `"mmreg"`.

Examples

```
(out <- fsreg(Y~., data=hbk, method="MM"))
class(out)
summary(out)
```

`mmregeda.object` *Description of* `mmregeda` *Objects*

Description

An object of class `mmregeda.object` holds information about the result of a call to `fsreg` when `method="MM"` and `monitoring=TRUE`.

Value

The object itself is basically a `list` with the following components:

auxscale	scalar, S estimate of the scale (or supplied external estimate of scale, if option InitialEst is not empty).
Beta	p x length(eff) matrix containing MM estimate of regression coefficients for each value of eff.

RES	n x length(eff) matrix containing the monitoring of scaled residuals for each value of eff.
Weights	n x length(eff) matrix containing the estimates of the weights for each value of eff
Outliers	Boolean matrix containing the list of the units declared as outliers for each value of eff using confidence level specified in input scalar conflev.
conflev	Confidence level which is used to declare units as outliers. Remark: conflev will be used to draw the horizontal line (confidence band) in the plot.
Ssingsub	Number of subsets wihtout full rank. Notice that Notice that singsub > 0.1*(number of subsamples) produces a warning
rhofunc	string identifying the rho function which has been used.
rhofuncparam	vector which contains the additional parameters for the specified rho function which have been used. For hyperbolic rho function the value of k =sup CVC. For Hampel rho function the parameters a, b and c.
eff	vector containing the value of eff which have been used.
X	the data matrix X
y	the response vector y

The object has class "mmregeda".

Examples

```
(out <- fsreg(Y~., data=hbk, method="MM", monitoring=TRUE))
class(out)
summary(out)
```

MMregeda_control *Creates an* MMregeda_control *object*

Description

Creates an object of class MMregeda_control to be used with the fsreg() function, containing various control parameters.

Usage

```
MMregeda_control(intercept = TRUE, InitialEst, Soptions, eff, effshape,
refsteps = 3, tol = 1e-07, conflev, nocheck = FALSE, plot = FALSE)
```

Arguments

intercept	Indicator for constant term. Scalar. If intercept=TRUE, a model with constant term will be fitted (default), else, no constant term will be included.
InitialEst	Starting values of the MM-estimator, a list with the fiollowing elements: loc, a $p x 1$ vector, location vector estimate and scale, a scaler, estimate of the scale. If empty (default) the program will use S estimators. In this last case it is possible to specify the options given in function Sreg.
Soptions	Options to pass to Sreg, an Sreg_control object. The options are: Srhofunc, Snsamp, Srefsteps, Sreftol, Srefstepsbestr, Sreftolbestr, Sminsctol, Sbestr. See function Sreg_control for more details on these options. It is necessary to add to the S options the letter S at the beginning. For example, if you want to use the optimal rho function the supplied option is 'Srhofunc','optimal'. For example, if you want to use 3000 subsets, the supplied option is 'Snsamp',3000
eff	Vector defining nominal efficiency (i.e. a series of numbers between 0.5 and 0.99). The default value is the sequence seq(0.5, 0.99, 0.01)
effshape	Location or scale efficiency. If effshape=1 efficiency refers to shape efficiency else (default) efficiency refers to location efficiency.
refsteps	Number of refining iterations in each subsample (default is refsteps=3). refsteps = 0 means "raw-subsampling" without iterations.
tol	Scalar controlling tolerance in the MM loop. The default value is tol=1e-6.
conflev	Confidence level which is used to declare units as outliers. Usually conflev=0.95, 0.975, 0.99 (individual alpha) or conflev=1-0.05/n, 1-0.025/n, 1-0.01/n (simultaneous alpha). Default value is 0.975
nocheck	Check input arguments, scalar. If nocheck=TRUE no check is performed on matrix y and matrix X. Notice that y and X are left unchanged. In other words the additional column of ones for the intercept is not added. As default nocheck=FALSE.
plot	Plot on the screen. Scalar. If plots=TRUE the plot of minimum deletion residual with envelopes based on n observations and the scatterplot matrix with the outliers highlighted is produced. If plots=2 the user can also monitor the intermediate plots based on envelope superimposition. If plots=FALSE (default) no plot is produced.

Details

Creates an object of class MMregeda_control to be used with the fsreg() function, containing various control parameters.

Value

An object of class "MMregeda_control" which is basically a list with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also as FSR_control, Sreg_control, MMreg_control and LXS_control

Examples

```
(out <- fsreg(Y~., data=hbk, method="MM", monitoring=TRUE,
    control=MMregeda_control(eff=seq(0.75, 0.99, 0.01))))
```

MMreg_control *Creates an* MMreg_control *object*

Description

Creates an object of class MMreg_control to be used with the fsreg() function, containing various control parameters for calling the MATLAB function MMreg().

Usage

```
MMreg_control(intercept = TRUE, InitialEst, Soptions, eff, effshape,
    rhofunc = c("bisquare", "optimal", "hyperbolic", "hampel"), rhofuncparam,
    refsteps = 3, tol = 1e-07, conflev,
    msg = TRUE, nocheck = FALSE, plot = FALSE)
```

Arguments

intercept	Indicator for constant term. Scalar. If intercept=TRUE, a model with constant term will be fitted (default), else, no constant term will be included.
InitialEst	Starting values of the MM-estimator, a list with the fiollowing elements: loc, a $p x 1$ vector, location vector estimate and scale, a scaler, estimate of the scale. If empty (default) the program will use S estimators. In this last case it is possible to specify the options given in function Sreg.
Soptions	Options to pass to Sreg, an Sreg_control object. The options are: Srhofunc, Snsamp, Srefsteps, Sreftol, Srefstepsbestr, Sreftolbestr, Sminsctol, Sbestr. See function Sreg_control for more details on these options. It is necessary to add to the S options the letter S at the beginning. For example, if you want to use the optimal rho function the supplied option is 'Srhofunc','optimal'. For example, if you want to use 3000 subsets, the supplied option is 'Snsamp',3000

`eff` Scalar defining nominal efficiency (i.e. a number between 0.5 and 0.99). The default value is 0.95.

`effshape` Location or scale efficiency. If `effshape=1` efficiency refers to shape efficiency else (default) efficiency refers to location efficiency.

`rhofunc` Specifies the rho function which must be used to weight the residuals. Possible values are 'bisquare' 'optimal' 'hyperbolic' 'hampel'.

1. 'bisquare' uses Tukey's rho and psi functions. See TBrho and TBpsi.
2. 'optimal' uses optimal rho and psi functions. See OPTrho and OPTpsi.
3. 'hyperbolic' uses hyperbolic rho and psi functions. See HYPrho and HYPpsi.
4. 'hampel' uses Hampel rho and psi functions. See HArho and HApsi.

The default is 'bisquare'.

`rhofuncparam` Additional parameters for the specified rho function. For hyperbolic rho function it is possible to set up the value of k = sup CVC (the default value of k is 4.5).

For Hampel rho function it is possible to define parameters a, b and c (the default values are a=2, b=4, c=8)

`refsteps` Number of refining iterations in each subsample (default is `refsteps=3`). `refsteps = 0` means "raw-subsampling" without iterations.

`tol` Scalar controlling tolerance in the MM loop. The default value is `tol=1e-6`

`conflev` Confidence level which is used to declare units as outliers. Usually conflev=0.95, 0.975, 0.99 (individual alpha) or conflev=1-0.05/n, 1-0.025/n, 1-0.01/n (simultaneous alpha). Default value is 0.975

`msg` Controls whether to display or not messages on the screen If `msg==1` (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.

`nocheck` Check input arguments, scalar. If `nocheck=TRUE` no check is performed on matrix `y` and matrix `X`. Notice that `y` and `X` are left unchanged. In other words the additional column of ones for the intercept is not added. As default `nocheck=FALSE`.

`plot` Plot on the screen. Scalar. If `plots=TRUE` the plot of minimum deletion residual with envelopes based on n observations and the scatterplot matrix with the outliers highlighted is produced. If `plots=2` the user can also monitor the intermediate plots based on envelope superimposition. If `plots=FALSE` (default) no plot is produced.

Details

Creates an object of class `MMreg_control` to be used with the `fsreg()` function, containing various control parameters.

Value

An object of class `"MMreg_control"` which is basically a `list` with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also as FSR_control, MMreg_control and LXS_control

Examples

```
(out <- fsreg(Y~., data=hbk, method="MM", control=MMreg_control(eff=0.99, rhofunc="optimal")))
```

---

resfwdplot *Plots the trajectories of the monitored scaled (squared) residuals*

Description

Plots the trajectories of the monitored scaled (squared) residuals

Usage

```
resfwdplot(out,
    xlim, ylim, xlab, ylab, main, lwd, lty, col, cex.lab, cex.axis,
    xvalues,
    fg.thresh, fg.unit, fg.labstep, fg.lwd, fg.lty, fg.col, fg.mark, fg.cex,
    bg.thresh, bg.style,
    tag, datatooltip, label, nameX, namey, msg, databrush,
    standard, fground, bground, ...)
```

Arguments

out	An object returned by one of the monitoring functions (see FSReda_control, Sregeda_control and MMregeda_control). The object is one of fsreda.object, sregeda.object or mmregeda.object. The needed elements of out are 1. RES: matrix containing the residuals monitored in each step of the forward search or any other robust procedure. Every row is associated with a residual (unit). This matrix can be created using function FSReda, Sregeda, MMregeda. 2. Un: (for FSR only) - matrix containing the order of entry in the subset of each unit (required only when datatooltip is true or databrush is not empty). 3. bdp: (for Sreg only) - vector containing a sequence of breakdown point values to monitor on. 4. eff: (for MMreg only) - vector containing a sequence of efficiency values to monitor on.

5. y: a vector containing the response (required only when option databrush is requested).
6. X: a matrix containing the explanatory variables (required only when option databrush is requested).
7. Bols: (n-init+1) x (p+1) matrix containing the estimated beta coefficients monitored in each step of the robust procedure (required only when option databrush is requested and suboption multivarfit is requested).

ylim — Control y scale in plot. Vector with two elements controlling minimum and maximum on the y axis. Default is to use automatic scale.

xlim — Control x scale in plot. Vector with two elements controlling minimum and maximum on the x axis. Default is to use automatic scale.

xlab — a title for the x axis

ylab — a title for the y axis

main — an overall title for the plot

lwd — The line width, a positive number, defaulting to 1

lty — The line type. Line types can either be specified as an integer (1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". The latter two are not supported by Matlab.

col — colors to be used for the highlighted units

cex.lab — The magnification to be used for x and y labels relative to the current setting of cex

cex.axis — The magnification to be used for axis annotation relative to the current setting of cex

xvalues — values for the x axis. Numeric vector of ncol(RES) controlling the x axis coordinates. The default value of xvalues is (nrow(RES) - ncol(RES) + 1):nrow(RES)

fg.thresh — (alternative to fg.unit) numeric vector of length 1 or 2 which specifies the highlighted trajectories. If length(fthresh) == 1 the highlighted trajectories are those of units that throughtout the search had at leat once a residual greater (in absolute value) than thresh. The default value is fg.thresh=2.5. If length(fthresh) == 2 the highlighted trajectories are those of units that throughtout the search had a residual at leat once bigger than fg.thresh[2] or smaller than fg.thresh[1].

fg.unit — (alternative to fg.thresh), vector containing the list of the units to be highlighted. If fg.unit is supplied, fg.thresh is ignored.

fg.labstep — numeric vector which specifies the steps of the search where to put labels for the highlighted trajectories (units). The default is to put the labels at the initial and final steps of the search. flabstep='' means no label.

fg.lwd — The line width for the highlighted trajectories (units). Default is 1.

fg.lty — The line type for the highlighted trajectories (units). Line types can either be specified as an integer (1=solid (default), 2=dashed, 3=dotted, 4=dotdash, 5=longdash, 6=twodash) or as one of the character strings "solid", "dashed", "dotted", "dotdash", "longdash", or "twodash". The latter two are not supported by Matlab.

fg.col — colors to be used for the highlighted units.

fg.mark — Controlls whether to plot highlighted trajectories as symbols. if fg.mark==TRUE each line is plotted using a different symbol else no marker is used (default).

fg.cex — controls the font size of the labels of the trajectories in foreground.

bg.thresh — numeric vector of length 1 or 2 which specifies how to define the unimmportant trajectories. Unimmportant trajectories will be plotted using a colormap, in greysh or will be hidden. If length(thresh) == 1 the irrelevant units are those which always had a residual smaller (in absolute value) than thresh. If length(bthresh) == 2 the irrelevant units are those which always had a residual greater than bthresh(1) and smaller than bthresh(2). The default is: bg.thresh=2.5 if n > 100 and bg.thresh=-Inf if n <= 100 i.e. to treat all trajectories as important if n <= 100 and, if n > 100, to reduce emphasis only to trajectories having in all steps of the search a value of scaled residual smaller than 2.5.

bg.style — specifies how to plot the unimportant trajectories as defined in option bthresh.

1. bg.style="faint": unimportant trajectories are plotted using a colormap.
2. bg.style="hide": unimportant trajectories are hidden.
3. bg.style="greyish": unimportant trajectories are displayed in a faint grey.

When n>100 the default option is bg.style='faint'. When n <= 100 and bg.thresh == -Inf option bstyle is ignored. Remark: bground=" is equivalent to -Inf that is all trajectories are considered relevant.

tag — Plot handle. String which identifies the handle of the plot which is about to be created. The default is to use tag 'pl_resfwd'. Notice that if the program finds a plot which has a tag equal to the one specified by the user, then the output of the new plot overwrites the existing one in the same window else a new window is created.

datatooltip — Interactive clicking. It is inactive if this parameter is missing or empty. The default is datatooltip=TRUE, i.e. the user can select with the mouse an individual residual trajectory in order to have information about the corresponding unit. The information displayed depends on the estimator in use.

For example for class fsreda.object the information concerns the label and the step of the search in which the unit enters the subset. If datatooltip is a list it may contain the following fields:

1. DisplayStyle determines how the data cursor displays. Possible values are 'datatip' and 'window' (default). 'datatip' displays data cursor information in a small yellow text box attached to a black square marker at a data point you interactively select. 'window' displays data cursor information for the data point you interactively select in a floating window within the figure.
2. SnapToDataVertex: specifies whether the data cursor snaps to the nearest data value or is located at the actual pointer position. Possible values are SnapToDataVertex='on' (default) and SnapToDataVertex='off'.
3. LineColor: controls the color of the trajectory selected with the mouse. It can be an RGB triplet of values between 0 and 1, or character vector indi-

cating a color name. Note that a RGB vector can be conveniently chosen with our MATLAB class FSColor, see documentation.

4. SubsetLinesColor: enables to control the color of the trajectories of the units that are in the subset at a given step of the search (if `resfwdplot()` is applied to an object of class `fsreda.object`) or have a weight greater than 0.9 (if `resfwdplot()` is applied to an object of class `sregeda.object` or `mmregeda.object`). This can be done (repeatedly) with a left mouse click in proximity of the step of interest. A right mouse click will terminate the selection by marking with a up-arrow the step corresponding to the highlighted lines. The highlighted lines by default are in red, but a different color can be specified as RGB triplet or character of color name. Note that a RGB vector can be conveniently chosen with our MATLAB class FSColor, see documentation. By default `SubsetLinesColor=""`, i.e. the modality is not active. Any initialization for `SubsetLinesColor` which cannot be interpreted as RGB vector will be converted to blue, i.e. `SubsetLinesColor` will be forced to be [0 0 1]. If `SubsetLinesColor` is not empty the previous option `LineColor` is ignored.

`label` Character vector containing the labels of the units (optional argument used when `datatooltip=TRUE`. If this field is not present labels row1, ..., rown will be automatically created and included in the pop up datatooltip window).

`nameX` Add variable labels in plot. A vector of strings of length p containing the labels of the variables of the regression dataset. If it is empty (default) the sequence `X1, ..., Xp` will be created automatically

`namey` Add response label. A string containing the label of the response

`msg` Controls whether to display or not messages on the screen If `msg==1` (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.

`databrush` interactive mouse brushing. If databrush is missing or empty (default), no brushing is done. The activation of this option (databrush is a scalar or a list) enables the user to select a set of trajectories in the current plot and to see them highlighted in the y|X plot, i.e. a matrix of scatter plots of y against each column of X, grouped according to the selection(s) done by brushing. If the plot y|X does not exist it is automatically created. In addition, brushed units are automatically highlighted in the minimum deletion residual plot if it is already open. The extension to the following plots will be available in future versions of the toolbox:

1. monitoring leverage plot;
2. maximum studentized residual;
3. s^2 and R^2;
4. Cook distance and modified Cook distance;
5. deletion t statistics.

Note that the window style of the other figures is set equal to that which contains the monitoring residual plot. In other words, if the monitoring residual plot is docked all the other figures will be docked too

If `databrush=TRUE` the default selection tool is a rectangular brush and it is possible to brush only once (that is persist=").

If `databrush=list(...)`, it is possible to use all optional arguments of function `selectdataFS()` and the following optional argument:

1. persist. Persist is an empty value or a character containing 'on' or 'off'. The default value is `persist=""`, that is brushing is allowed only once. If `persist="on"` or `persis="off"` brushing can be done as many time as the user requires. If `persist='on'` then the unit(s) currently brushed are added to those previously brushed. It is possible, every time a new brushing is done, to use a different color for the brushed units. If `persist='off'` every time a new brush is performed units previously brushed are removed.
2. bivarfit. This option adds one or more least square lines based on SIMPLE REGRESSION to the plots of y|X, depending on the selected groups.
3. bivarfitWheather to superimpose bivariate least square lines on the plot (if `plot=TRUE`. This option adds one or more least squares lines, based on SIMPLE REGRESSION of y on Xi, to the plots of y|Xi. The default is `bivarfit=FALSE`: no line is fitted. If `bivarfit=1`, a single OLS line is fitted to all points of each bivariate plot in the scatter matrix y|X. If `bivarfit=2`, two OLS lines are fitted: one to all points and another to the group of the genuine observations. The group of the potential outliers is not fitted. If `bivarfit=0` one OLS line is fitted to each group. This is useful for the purpose of fitting mixtures of regression lines. If `bivarfit='i1'` or `bivarfit='i2'`, etc. an OLS line is fitted to a specific group, the one with index 'i' equal to 1, 2, 3 etc. Again, useful in case of mixtures.
4. multivarfitWheather to superimpose multivariate least square lines. This option adds one or more least square lines, based on MULTIVARIATE REGRESSION of y on X, to the plots of y|Xi. The default is `multivarfit=FALSE`: no line is fitted. If `bivarfit=1`, a single OLS line is fitted to all points of each bivariate plot in the scatter matrix y|X. The line added to the scatter plot y|Xi is avconst + Ci*Xi, where Ci is the coefficient of Xi in the multivariate regression and avconst is the effect of all the other explanatory variables different from Xi evaluated at their centroid (that is overline(y)'C)). If `multivarfit=2`, same action as with `multivarfit=1` but this time we also add the line based on the group of unselected observations (i.e. the normal units).
5. labeladdAdd outlier labels in plot. If `labeladd=TRUE`, we label the outliers with the unit row index in matrices X and y. The default value is `labeladd=FALSE`, i.e. no label is added.

`standard` (MATLAB-style arguments) appearance of the plot in terms of xlim, ylim, axes labels and their font size style, color of the lines, etc.

`fground` MATLAB-style arguments for the fground trajectories in foregroud.

`bground` MATLAB-style arguments for the fground trajectories in backgroud.

`...` potential further arguments passed to lower level functions.

Details

No details

Value

No value returned

Author(s)

FSDA team

Examples

```
## Not run:

n <- 100
y <- rnorm(n)
X <- matrix(rnorm(n*4), nrow=n)

out <- fsreg(y~X, method="LTS")
out <- fsreg(y~X, method="FS", bsb=out$bs, monitoring=TRUE)
resfwdplot(out)


## End(Not run)
```

---

sreg.object *Description of sreg Objects*

Description

An object of class `sreg.object` holds information about the result of a call to `fsreg`.

Value

The object itself is basically a `list` with the following components:

`beta`	p-by-1 vector containing the estimated regression parameters (in step n-k).
`scale`	scalar containing the estimate of the scale (sigma).
`bs`	p x 1 vector containing the units forming best subset associated with S estimate of regression coefficient.
`residuals`	residuals.
`fittedvalues`	fitted values.
`outliers`	kx1 vector containing the list of the k units declared as outliers or NULL if the sample is homogeneous.
`conflev`	Confidence level which is used to declare units as outliers. Usually `conflev=0.95, 0.975, 0.99` (individual alpha) or `conflev=1-0.05/n, 1-0.025/n, 1-0.01/n` (simultaneous alpha). Default value is 0.975
`singsub`	Number of subsets wihtout full rank. Notice that `singsub > 0.1*(number of subsamples)` produces a warning

weights	n x 1 vector containing the estimates of the weights
rhofunc	Specifies the rho function which has been used to weight the residuals.
rhofuncparam	Vector which contains the additional parameters for the specified rho function which has been used. For hyperbolic rho function the value of k =sup CVC. For Hampel rho function the parameters a, b and c.
X	the data matrix X
y	the response vector y

The object has class "sreg".

Examples

```
(out <- fsreg(Y~., data=hbk, method="S"))
class(out)
summary(out)
```

---

sregeda.object *Description of* sregeda *Objects*

Description

An object of class sregeda.object holds information about the result of a call to fsreg when method="S" and monitoring=TRUE.

Value

The object itself is basically a list with the following components:

Beta	matrix containing the S estimator of regression coefficients for each value of bdp.
Scale	vector containing the estimate of the scale (sigma) for each value of bdp. This is the value of the objective function.
BS	p x 1 vector containing the units forming best subset associated with S estimate of regression coefficient.
RES	n x length(bdp) matrix containing the monitoring of scaled residuals for each value of bdp.
Weights	n x length(bdp) matrix containing the estimates of the weights for each value of bdp
Outliers	Boolean matrix containing the list of the units declared as outliers for each value of bdp using confidence level specified in input scalar conflev.
conflev	Confidence level which is used to declare units as outliers. Remark: conflev will be used to draw the horizontal line (confidence band) in the plot.

Singsub	Number of subsets wihtout full rank. Notice that singsub[bdp[jj]] > 0.1*(number of subsamples) produces a warning
rhofunc	Specifies the rho function which has been used to weight the residuals.
rhofuncparam	Vector which contains the additional parameters for the specified rho function which has been used. For hyperbolic rho function the value of k =sup CVC. For Hampel rho function the parameters a, b and c.
X	the data matrix X
y	the response vector y

The object has class "sregeda".

Examples

```
## Not run:
    (out <- fsreg(Y~., data=hbk, method="S", monitoring=TRUE))
    class(out)
    summary(out)

## End(Not run)
```

Sregeda_control *Creates an* Sregeda_control *object*

Description

Creates an object of class Sregeda_control to be used with the fsreg() function, containing various control parameters.

Usage

```
Sregeda_control(intercept = TRUE, bdp = seq(0.5, 0.01, -0.01),
    rhofunc = c("bisquare", "optimal", "hyperbolic", "hampel"), rhofuncparam,
   nsamp = 1000, refsteps = 3, reftol = 1e-06, refstepsbestr = 50, reftolbestr = 1e-08,
    minsctol = 1e-07, bestr = 5,
    conflev, msg = TRUE, nocheck = FALSE, plot = FALSE)
```

Arguments

intercept	Indicator for constant term. Scalar. If intercept=TRUE, a model with constant term will be fitted (default), else, no constant term will be included.
bdp	Breakdown point. It measures the fraction of outliers the algorithm should resist. In this case any value greater than 0 but smaller or equal than 0.5 will do fine. The default value of bdp is a sequence from 0.5 to 0.01 with step 0.01
rhofunc	Specifies the rho function which must be used to weight the residuals. Possible values are 'bisquare' 'optimal' 'hyperbolic' 'hampel'. 1. 'bisquare' uses Tukey's rho and psi functions. See TBrho and TBpsi.

2. 'optimal' uses optimal rho and psi functions. See OPTrho and OPTpsi.
3. 'hyperbolic' uses hyperbolic rho and psi functions. See HYPrho and HYPpsi.
4. 'hampel' uses Hampel rho and psi functions. See HArho and HApsi.

The default is 'bisquare'.

`rhofuncparam` Additional parameters for the specified rho function. For hyperbolic rho function it is possible to set up the value of k = sup CVC (the default value of k is 4.5).

For Hampel rho function it is possible to define parameters a, b and c (the default values are a=2, b=4, c=8)

`nsamp` Number of subsamples which will be extracted to find the robust estimator, scalar. If `nsamp=0` all subsets will be extracted. They will be `(n choose p)`. If the number of all possible subset is `<1000` the default is to extract all subsets otherwise just 1000.

`refsteps` Number of refining iterations in each subsample (default is `refsteps=3`). `refsteps = 0` means "raw-subsampling" without iterations.

`reftol` Tolerance for the refining steps. The default value is 1e-6

`refstepsbestr` Scalar defining number of refining iterations for each best subset (default = 50).

`reftolbestr` Tolerance for the refining steps for each of the best subsets. The default value is `reftolbestr=1e-8`.

`minsctol` Value of tolerance for the iterative procedure for finding the minimum value of the scale for each subset and each of the best subsets (It is used by subroutine `minscale.m`). The default value is `minsctol=1e-7`.

`bestr` Defins the number of "best betas" to remember from the subsamples. These will be later iterated until convergence (default is `bestr=5`).

`conflev` Confidence level which is used to declare units as outliers. Usually conflev=0.95, 0.975, 0.99 (individual alpha) or conflev=1-0.05/n, 1-0.025/n, 1-0.01/n (simultaneous alpha). Default value is 0.975

`msg` Controls whether to display or not messages on the screen If `msg==1` (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.

`nocheck` Check input arguments, scalar. If `nocheck=TRUE` no check is performed on matrix `y` and matrix `X`. Notice that `y` and `X` are left unchanged. In other words the additional column of ones for the intercept is not added. As default `nocheck=FALSE`.

`plot` Plot on the screen. Scalar. If `plots=TRUE` the plot of minimum deletion residual with envelopes based on n observations and the scatterplot matrix with the outliers highlighted is produced. If `plots=2` the user can also monitor the intermediate plots based on envelope superimposition. If `plots=FALSE` (default) no plot is produced.

Details

Creates an object of class `Sregeda_control` to be used with the `fsreg()` function, containing various control parameters.

Value

An object of class "Sregeda_control" which is basically a list with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also as FSR_control, MMreg_control and LXS_control

Examples

```
## Not run:

(out <- fsreg(Y~., data=hbk, method="S", monitoring=TRUE,
    control=Sregeda_control(nsamp=500, rhofunc='hyperbolic')))

## End(Not run)
```

Sreg_control *Creates an* Sreg_control *object*

Description

Creates an object of class Sreg_control to be used with the fsreg() function, containing various control parameters for calling the MATLAB function Sreg().

Usage

```
Sreg_control(intercept = TRUE, bdp = 0.5,
    rhofunc = c("bisquare", "optimal", "hyperbolic", "hampel"), rhofuncparam,
   nsamp = 1000, refsteps = 3, reftol = 1e-06, refstepsbestr = 50, reftolbestr = 1e-08,
    minsctol = 1e-07, bestr = 5,
    conflev, msg = TRUE, nocheck = FALSE, plot = FALSE)
```

Arguments

intercept	Indicator for constant term. Scalar. If intercept=TRUE, a model with constant term will be fitted (default), else, no constant term will be included.
bdp	Breakdown point. It measures the fraction of outliers the algorithm should resist. In this case any value greater than 0 but smaller or equal than 0.5 will do fine. Note that given bdp nominal efficiency is automatically determined.
rhofunc	Specifies the rho function which must be used to weight the residuals. Possible values are 'bisquare' 'optimal' 'hyperbolic' 'hampel'.

1. 'bisquare' uses Tukey's rho and psi functions. See TBrho and TBpsi.
2. 'optimal' uses optimal rho and psi functions. See OPTrho and OPTpsi.
3. 'hyperbolic' uses hyperbolic rho and psi functions. See HYPrho and HYPpsi.
4. 'hampel' uses Hampel rho and psi functions. See HArho and HApsi.

The default is 'bisquare'.

`rhofuncparam` Additional parameters for the specified rho function. For hyperbolic rho function it is possible to set up the value of k = sup CVC (the default value of k is 4.5).

For Hampel rho function it is possible to define parameters a, b and c (the default values are a=2, b=4, c=8)

`nsamp` Number of subsamples which will be extracted to find the robust estimator, scalar. If `nsamp=0` all subsets will be extracted. They will be `(n choose p)`. If the number of all possible subset is `<1000` the default is to extract all subsets otherwise just 1000.

`refsteps` Number of refining iterations in each subsample (default is `refsteps=3`). `refsteps = 0` means "raw-subsampling" without iterations.

`reftol` Tolerance for the refining steps. The default value is 1e-6

`refstepsbestr` Scalar defining number of refining iterations for each best subset (default = 50).

`reftolbestr` Tolerance for the refining steps for each of the best subsets. The default value is `reftolbestr=1e-8`.

`minsctol` Value of tolerance for the iterative procedure for finding the minimum value of the scale for each subset and each of the best subsets (It is used by subroutine `minscale.m`). The default value is `minsctol=1e-7`.

`bestr` Defins the number of "best betas" to remember from the subsamples. These will be later iterated until convergence (default is `bestr=5`).

`conflev` Confidence level which is used to declare units as outliers. Usually conflev=0.95, 0.975, 0.99 (individual alpha) or conflev=1-0.05/n, 1-0.025/n, 1-0.01/n (simultaneous alpha). Default value is 0.975

`msg` Controls whether to display or not messages on the screen If `msg==1` (default) messages are displayed on the screen about step in which signal took place else no message is displayed on the screen.

`nocheck` Check input arguments, scalar. If `nocheck=TRUE` no check is performed on matrix y and matrix `X`. Notice that y and `X` are left unchanged. In other words the additional column of ones for the intercept is not added. As default `nocheck=FALSE`.

`plot` Plot on the screen. Scalar. If `plots=TRUE` the plot of minimum deletion residual with envelopes based on n observations and the scatterplot matrix with the outliers highlighted is produced. If `plots=2` the user can also monitor the intermediate plots based on envelope superimposition. If `plots=FALSE` (default) no plot is produced.

Details

Creates an object of class `Sreg_control` to be used with the `fsreg()` function, containing various control parameters.

Value

An object of class "Sreg_control" which is basically a list with components the input arguments of the function mapped accordingly to the corresponding Matlab function.

Author(s)

FSDA team

See Also

See Also as FSR_control, MMreg_control and LXS_control

Examples

```
(out <- fsreg(Y~., data=hbk, method="S", control=Sreg_control(bdp=0.25, nsamp=500)))
```

---

summary.fsdalms *Summary Method for* fsdalms *objects*

Description

summary method for class "fsdalms".

Usage

```
## S3 method for class 'fsdalms'
summary(object, correlation = FALSE, ...)
## S3 method for class 'summary.fsdalms'
print(x, digits = max(3, getOption("digits") - 3),
     signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

object, x	an object of class "fsdalms" (or "summary.fsdalms"); usually, a result of a call to fsreg.
correlation	logical; if TRUE, the correlation matrix of the estimated parameters is returned and printed.
digits	the number of significant digits to use when printing.
signif.stars	logical indicating if "significance stars" should be printer, see printCoefmat.
...	further arguments passed to or from other methods.

Details

summary.fsdalms(), the S3 method, simply returns an (S3) object of class "summary.fsdalms" for which there's a print method:

print.summary.fsdalms prints summary statistics for the forward search (FS) regression estimates. While the function print.fsdalms prints only the robust estimates of the coefficients, print.summary.fsdalms will print also the regression table.

Value

summary.fsdalms returns an summary.fsdalms object, whereas the print methods returns its first argument via invisible, as all print methods do.

See Also

fsreg, summary

Examples

```
data(Animals, package = "MASS")
brain <- Animals[c(1:24, 26:25, 27:28),]
lbrain <- log(brain)
(fs <- fsreg(brain~body, data=lbrain, method="LTS"))
summary(fs)

## compare to the result of ltsReg() from 'robustbase'
(lts <- ltsReg(brain~body, data=lbrain))
summary(lts)
```

---

summary.fsdalts *Summary Method for* fsdalts *objects*

Description

summary method for class "fsdalts".

Usage

```
## S3 method for class 'fsdalts'
summary(object, correlation = FALSE, ...)
## S3 method for class 'summary.fsdalts'
print(x, digits = max(3, getOption("digits") - 3),
     signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

object, x	an object of class "fsdalts" (or "summary.fsdalts"); usually, a result of a call to fsreg.
correlation	logical; if TRUE, the correlation matrix of the estimated parameters is returned and printed.
digits	the number of significant digits to use when printing.
signif.stars	logical indicating if "significance stars" should be printer, see printCoefmat.
...	further arguments passed to or from other methods.

Details

summary.fsdalts(), the S3 method, simply returns an (S3) object of class "summary.fsdalts" for which there's a print method:

print.summary.fsdalts prints summary statistics for the forward search (FS) regression estimates. While the function print.fsdalts prints only the robust estimates of the coefficients, print.summary.fsdalts will print also the regression table.

Value

summary.fsdalts returns an summary.fsdalts object, whereas the print methods returns its first argument via invisible, as all print methods do.

See Also

fsreg, summary

Examples

```
data(Animals, package = "MASS")
brain <- Animals[c(1:24, 26:25, 27:28),]
lbrain <- log(brain)
(fs <- fsreg(brain~body, data=lbrain, method="LTS"))
summary(fs)

## compare to the result of ltsReg() from 'robustbase'
(lts <- ltsReg(brain~body, data=lbrain))
summary(lts)
```

summary.fsr *Summary Method for FSR objects*

Description

summary method for class "fsr".

Usage

```
## S3 method for class 'fsr'
summary(object, correlation = FALSE, ...)
## S3 method for class 'summary.fsr'
print(x, digits = max(3, getOption("digits") - 3),
     signif.stars = getOption("show.signif.stars"), ...)
```

Arguments

object, x	an object of class "fsr" (or "summary.fsr"); usually, a result of a call to fsreg.
correlation	logical; if TRUE, the correlation matrix of the estimated parameters is returned and printed.
digits	the number of significant digits to use when printing.
signif.stars	logical indicating if "significance stars" should be printer, see printCoefmat.
...	further arguments passed to or from other methods.

Details

summary.fsr(), the S3 method, simply returns an (S3) object of class "summary.fsr" for which there's a print method:

print.summary.fsr prints summary statistics for the forward search (FS) regression estimates. While the function print.fsr prints only the robust estimates of the coefficients, print.summary.fsr will print also the regression table.

Value

summary.fsr returns an summary.fsr object, whereas the print methods returns its first argument via invisible, as all print methods do.

See Also

fsreg, summary

Examples

```
data(Animals, package = "MASS")
brain <- Animals[c(1:24, 26:25, 27:28),]
lbrain <- log(brain)
(fs <- fsreg(brain~body, data=lbrain, method="FS"))
summary(fs)
```

Index