Package: robust (via r-universe)

October 28, 2026

Version 0.7-5

VersionNote Released 0.7-4 on 2024-02-04 on CRAN

Title Port of the S+ ``Robust Library''

Description Methods for robust statistics, a state of the art in the early 2000s, notably for robust regression and robust multivariate analysis.

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

Depends fit.models

Imports graphics, stats, utils, lattice, MASS, robustbase, rrcov

License GPL (>= 3)

URL https://github.com/valentint/robust

BugReports https://github.com/valentint/robust/issues

NeedsCompilation yes

Author Jiahui Wang [aut], Ruben Zamar [aut], Alfio Marazzi [aut], Victor Yohai [aut], Matias Salibian-Barrera [aut], Ricardo Maronna [aut], Eric Zivot [aut], David Rocke [aut], Doug Martin [aut], Martin Maechler [aut], Kjell Konis [aut], Valentin Todorov [aut, cre]

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

Date/Publication 2024-08-18 02:51:03 UTC

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

RemoteRef HEAD

RemoteSha b7b61c6eddbc0e5c234ba561c70435626105e88b

Contents

anova.glmRob 3
anova.lmRob 4
breslow.dat 5
covClassic 6

covRob . . . 7
covRob.control . . . 9
ddPlot.covfm . . . 10
distancePlot.covfm . . . 11
drop1.lmRob . . . 12
ellipsesPlot.covfm . . . 13
fitdstn . . . 14
fitdstnRob . . . 15
gammaRob . . . 16
gammaRob.control . . . 17
gen.data . . . 17
glmRob . . . 18
glmRob.control . . . 19
glmRob.cubif . . . 20
glmRob.cubif.control . . . 21
glmRob.mallows . . . 22
glmRob.mallows.control . . . 23
glmRob.misclass . . . 23
glmRob.misclass.control . . . 24
glmRob.object . . . 25
leuk.dat . . . 26
lmRob . . . 27
lmRob.control . . . 29
lmRob.fit . . . 30
lmRob.fit.compute . . . 31
lmRob.object . . . 32
lmRob.RFPE . . . 33
lognormRob . . . 34
lognormRob.control . . . 35
lsRobTest . . . 35
mallows.dat . . . 36
overlaidDenPlot.fdfm . . . 36
plot.covfm . . . 37
plot.fdfm . . . 39
plot.glmRob . . . 39
plot.lmRob . . . 40
predict.glmRob . . . 41
predict.lmRob . . . 43
qqPlot.fdfm . . . 44
rb.lmRob . . . 44
residuals.glmRob . . . 45
screePlot.covfm . . . 45
stack.dat . . . 46
step.lmRob . . . 47
summary.covfm . . . 48
summary.glmRob . . . 50
summary.lmRob . . . 50
test.lmRob . . . 51

update.lmRob 52
weibullRob 53
weibullRob.control 54
weight.funs 55
woodmod.dat 56

Index **57**

`anova.glmRob` *ANOVA for Robust Generalized Linear Model Fits*

Description

Compute an analysis of variance table for one or more robust generalized linear model fits.

Usage

```
## S3 method for class 'glmRob'
anova(object, ..., test = c("none", "Chisq", "F", "Cp"))
## S3 method for class 'glmRoblist'
anova(object, ..., test = c("none", "Chisq", "F", "Cp"))
```

Arguments

`object`	a glmRob object.
`...`	additional glmRob objects.
`test`	a character string specifying the test statistic to be used. Can be one of "F", "Chisq", "Cp" or "none" for no test.

Value

an anova object.

See Also

`glmRob`, `anova`, `anova.glmRoblist`.

Examples

```
data(breslow.dat)

bres.int <- glmRob(sumY ~ Age10 + Base4*Trt, family = poisson(), data = breslow.dat)
anova(bres.int)

bres.main <- glmRob(sumY ~ Age10 + Base4 + Trt, family = poisson(), data = breslow.dat)
anova(bres.main, bres.int)
```

anova.lmRob *ANOVA for Robust Linear Model Fits*

Description

Compute an analysis of variance table for one or more robust linear model fits.

Usage

```
## S3 method for class 'lmRob'
anova(object, ..., test = c("RF", "RWald"))
## S3 method for class 'lmRoblist'
anova(object, const, ipsi, yc, test = c("RWald", "RF"), ...)
```

Arguments

object	an lmRob object.
...	additional arguments required by the generic anova function. If ... contains additional robustly fitted linear models then the function anova.lmRoblist is dispatched.
const	a numeric value containing the tuning constant.
ipsi	an integer value specifying the psi-function.
yc	a numeric value containing the tuning constant.
test	a single character value specifying which test should be computed in the Anova table. The possible choices are "RWald" and "RF".

Details

The default test used by anova is the "RWald" test, which is the Wald test based on robust estimates of the coefficients and covariance matrix. If test is "RF", the robustified F-test is used instead.

Value

an anova object.

References

Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., and Stahel, W. A. (1986). *Robust statistics: the approach based on influence functions*. John Wiley & Sons.

See Also

lmRob, anova.

Examples

```
data(stack.dat)
stack.small <- lmRob(Loss ~ Water.Temp + Acid.Conc., data = stack.dat)
stack.full <- lmRob(Loss ~ ., data = stack.dat)
anova(stack.full)
anova(stack.full, stack.small)
```

`breslow.dat` *Breslow Data*

Description

Patients suffering from simple or complex partial seizures were randomized to receive either the antiepileptic drug progabide or a placebo. At each of four successive postrandomization clinic visits, the number of seizures occuring over the previous two weeks was reported.

Usage

```
data(breslow.dat)
```

Format

A data frame with 59 observations on the following 12 variables.

`ID` an integer value specifying the patient identification number.

`Y1` an integer value, the number of seizures during the first two week period.

`Y2` an integer value, the number of seizures during the second two week period.

`Y3` an integer value, the number of seizures during the third two week period.

`Y4` an integer value, the number of seizures during the fourth two week period.

`Base` an integer value giving the eight-week baseline seizure count.

`Age` an integer value giving the age of the parient in years.

`Trt` the treatment: a factor with levels `placebo` and `progabide`.

`Ysum` an integer value, the sum of `Y1`, `Y2`, `Y3` and `Y4`.

`sumY` an integer value, the sum of `Y1`, `Y2`, `Y3` and `Y4`.

`Age10` a numeric value, `Age` divided by 10.

`Base4` a numeric value, `Base` divided by 4.

References

Breslow, N. E., and Clayton, D. G. (1993), "Approximate Inference in Generalized Linear Mixed Models," Journal of the American Statistical Association, Vol. 88, No. 421, pp. 9-25.

Thrall, P. F., and Vail, S. C. (1990), "Some Covariance Models for Longitudinal Count Data With Overdispersion," Biometrics, Vol. 46, pp. 657-671.

Examples

```
data(breslow.dat)
```

`covClassic` *Classical Covariance Estimation*

Description

Compute an estimate of the covariance/correlation matrix and location vector using classical methods.

Its main intention is to return an object compatible to that produced by `covRob`, but fit using classical methods.

Usage

```
covClassic(data, corr = FALSE, center = TRUE, distance = TRUE,
           na.action = na.fail, unbiased = TRUE, ...)
```

Arguments

`data`	a numeric matrix or data frame containing the data.
`corr`	a logical flag. If `corr` = `TRUE` then the estimated correlation matrix is computed.
`center`	a logical flag or a numeric vector of length `p` (where `p` is the number of columns of `x`) specifying the center. If `center` = `TRUE` then the center is estimated. Otherwise the center is taken to be 0.
`distance`	a logical flag. If `distance` = `TRUE` the Mahalanobis distances are computed.
`na.action`	a function to filter missing data. The default `na.fail` produces an error if missing values are present. An alternative is `na.omit` which deletes observations that contain one or more missing values.
`unbiased`	logical indicating if an unbiased estimate of the covariance matrix is should becomputed. If false, the maximum likelihood estimate is computed.
`...`	additional .

Value

a list with class "covClassic" containing the following elements:

`call`	an image of the call that produced the object with all the arguments named.
`cov`	a numeric matrix containing the estimate of the covariance/correlation matrix.
`center`	a numeric vector containing the estimate of the location vector.
`dist`	a numeric vector containing the squared Mahalanobis distances. Only present if `distance` = `TRUE` in the `call`.
`corr`	a logical flag. If `corr` = `TRUE` then `cov` contains an estimate of the correlation matrix of `x`.

Note

Originally, and in S-PLUS, this function was called `cov`; it has been renamed, as that did mask the function in the standard package **stats**.

See Also

`covRob`, `var`, `cov.wt`.

Examples

```
data(stack.dat)
covClassic(stack.dat)
```

---

covRob *Robust Covariance/Correlation Matrix Estimation*

Description

Compute robust estimates of multivariate location and scatter.

Usage

```
covRob(data, corr = FALSE, distance = TRUE, na.action = na.fail,
       estim = "auto", control = covRob.control(estim, ...), ...)
```

Arguments

`data`	a numeric matrix or data frame containing the data.
`corr`	a logical flag. If `corr = TRUE` then the estimated correlation matrix is computed.
`distance`	a logical flag. If `distance = TRUE` the squared Mahalanobis distances are computed.
`na.action`	a function to filter missing data. The default `na.fail` produces an error if missing values are present. An alternative is `na.omit` which deletes observations that contain one or more missing values.
`estim`	a character string specifying the robust estimator to be used. The choices are: "mcd" for the Fast MCD algorithm of Rousseeuw and Van Driessen, "weighted" for the Reweighted MCD, "donostah" for the Donoho-Stahel projection based estimator, "M" for the constrained M estimator provided by Rocke, "pairwiseQC" for the orthogonalized quadrant correlation pairwise estimator, and "pairwiseGK" for the Orthogonalized Gnanadesikan-Kettenring pairwise estimator. The default "auto" selects from "donostah", "mcd", and "pairwiseQC" with the goal of producing a good estimate in a reasonable amount of time.
`control`	a list of control parameters to be used in the numerical algorithms. See `covRob.control` for the possible control parameters and their default settings. This argument is ignored when `estim = "auto"`.
`...`	control parameters may be passed directly when `estim != "auto"`.

Details

The covRob function selects a robust covariance estimator that is likely to provide a *good* estimate in a reasonable amount of time. Presently this selection is based on the problem size. The Donoho-Stahel estimator is used if there are less than 1000 observations and less than 10 variables or less than 5000 observations and less than 5 variables. If there are less than 50000 observations and less than 20 variables then the MCD is used. For larger problems, the Orthogonalized Quadrant Correlation estimator is used.

The MCD and Reweighted-MCD estimates (estim = "mcd" and estim = "weighted" respectively) are computed using the covMcd function in the robustbase package. By default, covMcd returns the reweighted estimate; the actual MCD estimate is contained in the components of the output list prefixed with raw.

The M estimate (estim = "M") is computed using the CovMest function in the rrcov package. For historical reasons the Robust Library uses the MCD to compute the initial estimate.

The Donoho-Stahel (estim = "donostah") estimator is computed using the CovSde function provided in the rrcov package.

The pairwise estimators (estim = "pairwisegk" and estim = "pairwiseqc") are computed using the CovOgk function in the rrcov package.

Value

an object of class "covRob" with components:

call	an image of the call that produced the object with all the arguments named.
cov	a numeric matrix containing the final robust estimate of the covariance/correlation matrix.
center	a numeric vector containing the final robust estimate of the location vector.
dist	a numeric vector containing the squared Mahalanobis distances computed using robust estimates of covariance and location contained in cov and center. If distance = FALSE this element will me missing.
raw.cov	a numeric matrix containing the initial robust estimate of the covariance/correlation matrix. If there is no initial robust estimate then this element is set to NA.
raw.center	a numeric vector containing the initial robust estimate of the location vector. If there is no initial robust estimate then this element is set to NA.
raw.dist	a numeric vector containing the squared Mahalanobis distances computed using the initial robust estimates of covariance and location contained in raw.cov and raw.center. If distance = FALSE or if there is no initial robust estimate then this element is set to NA.
corr	a logical flag. If corr = TRUE then cov and raw.cov contain robust estimates of the correlation matrix of data.
estim	a character string containing the name of the robust estimator.
control	a list containing the control parameters used by the robust estimator.

Note

Version 0.3-8 of the Robust Library: all of the functions origianlly contributed by the S-Plus Robust Library have been replaced by dependencies on the robustbase and rrcov packages. Computed results may differ from earlier versions of the Robust Library. In particular, the MCD estimators are now adjusted by a small sample size correction factor. Additionally, a bug was fixed where the final MCD covariance estimate produced with `estim = "mcd"` was not rescaled for consistency.

References

R. A. Maronna and V. J. Yohai (1995) The Behavior of the Stahel-Donoho Robust Multivariate Estimator. *Journal of the American Statistical Association* **90** (429), 330–341.

P. J. Rousseeuw and K. van Driessen (1999) A fast algorithm for the minimum covariance determinant estimator. *Technometrics* **41**, 212–223.

D. L. Woodruff and D. M. Rocke (1994) Computable robust estimation of multivariate location and shape on high dimension using compound estimators. *Journal of the American Statistical Association*, **89**, 888–896.

R. A. Maronna and R. H. Zamar (2002) Robust estimates of location and dispersion of high-dimensional datasets. *Technometrics* **44** (4), 307–317.

See Also

`CovSde`, `covMcd`, `CovOgk`, `CovMest`, `covRob.control`, `covClassic`.

Examples

```
data(stackloss)
covRob(stackloss)
```

---

covRob.control *Control Parameters for Robust Covariance Estimation*

Description

This function is used to create a list of control parameters for the underlying robust estimator used in the covRob function.

Usage

```
covRob.control(estim, ...)
```

Arguments

`estim`	a character vector of length one giving the name of the estimator to generate the control parameters for.
`...`	control parameters appropriate for the robust estimator specified in `estim` in the form `name = value` and separated by commas. Omitted parameters receive their default values.

Details

The control parameters are estimator specific. Information on the control parameters (and their default values) can be found in the help files of each of the robust covariance estimators.

Value

a list of control parameters appropriate for the robust estimator given in `estim`. The value of `estim` occupies the first element of the list.

See Also

This function is a utility function for covRob.

The underlying robust estimators are: CovSde, covMcd and CovOgk. Power-users should consider calling these functions directly.

Examples

```
mcd.control <- covRob.control("mcd", quan = 0.75, ntrial = 1000)

ds.control <- covRob.control("donostah", prob = 0.95)

qc.control <- covRob.control("pairwiseqc")
```

---

ddPlot.covfm *Distance - Distance Plot*

Description

For a `covfm` object containing 2 models, this function plots the Mahalanobis distance from the first model on the y-axis and the Mahalanobis distance from the second model on the x-axis.

Usage

```
ddPlot.covfm(x, level = 0.95, strip = "", id.n = 3, ...)
```

Arguments

`x`	a `"covfm"` object.
`level`	a single numeric value between 0 and 1 giving the chi-squared percent point used to compute the outlyingness threshold.
`strip`	a character string printed in the "strip" at the top of the plot.
`id.n`	a single nonnegative integer specifying the number of extreme points to label in the plot.
`...`	additional arguments are passed to `xyplot`.

Value

if the models can be compared then the plotted `trellis` object is invisibly returned. Otherwise `x` is returned invisibly.

Examples

```
data(woodmod.dat)
woodm.fm <- fit.models(list(Robust = "covRob", Classical = "covClassic"),
                       data = woodmod.dat)
ddPlot.covfm(woodm.fm, main = "Plot Title", xlab = "x-axis label",
             ylab = "y-axis label", pch = 4, col = "purple")
```

`distancePlot.covfm` *Side-by-Side Mahalanobis Distance Plot*

Description

Produces side-by-side plots of Mahalanobis distance computed using the location and covariance matrix estimates contained in each element of a `covfm` object.

Usage

```
distancePlot.covfm(x, level = 0.95, id.n = 3, ...)
```

Arguments

`x`	a `"covfm"` object.
`level`	a single numeric value between 0 and 1 giving the chi-squared percent point used to compute the outlyingness threshold.
`id.n`	a single nonnegative integer specifying the number of extreme points to label in the plot.
`...`	additional arguments are passed to `xyplot`.

Value

the `trellis` object is invisibly returned.

Examples

```
data(woodmod.dat)
woodm.fm <- fit.models(list(Robust = "covRob", Classical = "covClassic"),
                       data = woodmod.dat)
distancePlot.covfm(woodm.fm, main = "Plot Title", xlab = "x-axis label",
                   ylab = "y-axis label", pch = 4, col = "purple")
```

drop1.lmRob *Compute an Anova Object by Dropping Terms*

Description

drop1.lmRob is used to investigate a robust Linear Model object by recomputing it, successively omitting each of a number of specified terms.

Usage

```
## S3 method for class 'lmRob'
drop1(object, scope, scale, keep, fast = FALSE, ...)
```

Arguments

object	an lmRob object.
scope	an optional formula object describing the terms to be dropped. Typically this argument is omitted, in which case all possible terms are dropped (without breaking hierarchy rules). The scope can also be a character vector of term labels. If the argument is supplied as a formula, any . is interpreted relative to the formula implied by the object argument.
scale	a single numeric value containing a residual scale estimate. If missing, the scale estimate in object is used.
keep	a character vector of names of components that should be saved for each subset model. Only names from the set "coefficients", "fitted" and "residuals" are allowed. If keep == TRUE, the complete set is saved. The default behavior is not to keep anything.
fast	a logical value. If TRUE the robust initial estimate (used when fitting each of the reduced models) is replaced by a weighted least squares estimate using the robust weights in object.
...	additional arguments required by the generic drop1 function.

Details

This function is a method for the generic function drop1 for class "lmRob".

Value

An anova object is constructed, consisting of the term labels, the degrees of freedom, and Robust Final Prediction Errors (RFPE) for each subset model. If keep is missing, the anova object is returned. If keep is present, a list with components "anova" and "keep" is returned. In this case, the "keep" component is a matrix of mode "list", with a column for each subset model, and a row for each component kept.

See Also

anova, drop1, lmRob.object.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)
drop1(stack.rob)
```

`ellipsesPlot.covfm` *Ellipses Plot - Visual Correlation Matrix Comparison*

Description

When there are 3 or more variables in the data, this function produces a matrix with ellipses drawn in the upper triangle. The ellipse in cell i, j of the plot is drawn to be a contour of a standard bivariate normal with correlation ρ_{ij}. One ellipse is drawn in each cell for each model in the `covfm` object. When there are 2 variables in the data, this function produces a scatter plot of the data with an overlaid 95% confidence ellipse for each model in the `covfm` object.

Usage

```
ellipsesPlot.covfm(x, ...)
```

Arguments

`x`	a `"covfm"` object.
`...`	additional arguments are ignored.

Value

x is invisibly returned.

Examples

```
data(woodmod.dat)
woodm.fm <- fit.models(list(Robust = "covRob", Classical = "covClassic"),
                       data = woodmod.dat)
ellipsesPlot.covfm(woodm.fm)
```

`fitdstn` *Maximum-likelihood Fitting of Univariate Distributions*

Description

Maximum-likelihood fitting of univariate distributions.

Usage

```
fitdstn(x, densfun, ...)
```

Arguments

`x`	a numeric vector containing the sample.
`densfun`	a character string naming the distribution. Distributions 'gamma', 'lognormal', and 'weibull' are supported.
`...`	additional arguments are ignored.

Details

This function relies on the `fitdistr` function for the computations. The returned object is modified to support plotting and comparison.

Value

a list with class "fitdstn" containing the following elements:

`estimate`	a named numeric vector containing the parameter estimates.
`sd`	a named numeric vector containing the standard deviations of the parameter estimates.
`vcov`	a numeric matrix containing the variance-covariance matrix of the estimated parameter vector.
`n`	a single numeric value indicating the number of sample points in `x`.
`loglik`	a single numeric value giving the maxized the log-likelihood.
`call`	the matched call.
`densfun`	the character string `densfun` provided in the arguments.
`x`	the data provided in `x`.

Note

The `print` method displays the estimated parameters and their standard errors (in parentheses).

See Also

An important goal here is the comparison with *robust* fits to the same distributions, see `fitdstnRob`.

`fitdistr` which provides many more choices for `densfun`.

`fitdstnRob` *Robust Fitting of Univariate Distributions*

Description

Robust Fitting of Univariate Distributions.

Usage

```
fitdstnRob(x, densfun, ...)
```

Arguments

`x`	A numeric vector containing the sample.
`densfun`	a character string naming the distribution. Distributions 'gamma', 'lognormal', and 'weibull' are recognized.
`...`	additional arguments are passed to the fitting functions.

Value

a list with class "fitdstn" containing the following elements:

`estimate`	a named numeric vector containing the parameter estimates.
`sd`	a named numeric vector containing the standard deviations of the parameter estimates.
`vcov`	a numeric matrix containing the variance-covariance matrix of the estimated parameter vector.
`mu`	a single numeric value containing an estimate of the mean.
`V.mu`	a single numeric value containing the variance of the estimated mean.
`control`	a list containing the control parameters used by the estimator.
`call`	the matched call.
`densfun`	the character string `densfun` provided in the arguments.
`x`	the data provided in `x`.

The `print` method displays the estimated parameters and their standard errors (in parentheses).

See Also

`gammaRob`, `lognormRob`, `weibullRob`.

The classical counterparts, see `fitdstn`.

gammaRob *Robust Estimation of Gamma Distribution Parameters*

Description

Robust estimation of gamma distribution parameters

Usage

```
gammaRob(x, estim = c("M", "tdmean"), control = gammaRob.control(estim, ...), ...)
```

Arguments

x	a numeric vector containing the sample.
estim	a character string specifying which estimator to use.
control	a list of control parameters appropriate for the estimator in estim.
...	control parameters may also be given here.

Value

a list with class "fitdstn" containing the following elements:

estimate	a named numeric vector containing the parameter estimates.
sd	a named numeric vector containing the standard deviations of the parameter estimates.
vcov	a numeric matrix containing the variance-covariance matrix of the estimated parameter vector.
mu	a single numeric value containing an estimate of the mean.
V.mu	a single numeric value containing the variance of the estimated mean.
control	a list containing the control parameters used by the estimator.

The print method displays the estimated parameters and their standard errors (in parentheses).

See Also

gammaRob.control, fitdstnRob.

gammaRob.control *Control Parameters for gammaRob*

Description

Create a list of control parameters for the gammaRob function.

Usage

```
gammaRob.control(estim, ...)
```

Arguments

estim	a character string specifying the estimator.
...	control parameters appropriate for the estimator given in estim.

Value

a list of control parameters appropriate for the specified estimator.

gen.data *Generate Data With Contamination*

Description

Generates a random dataset with some amount of contaimination.

Usage

```
gen.data(coeff, n = 100, eps = 0.1, sig = 3, snr = 1/20, seed = 837)
```

Arguments

coeff	a numeric vector of length 3 containing the true coefficients.
n	a positive integer giving the number of observations in the data set.
eps	a numeric value between 0 and 0.5 specifying the fraction of contamination.
sig	a positive numeric value giving the standard deviation of the uncontaminated data.
snr	a positive numeic value giving the signal to noise ratio, well not really.
seed	an integer value giving the seed for the random number generator.

Value

a data frame with n rows and 4 columns. The regressors are generated as: rnorm(n,1), rnorm(n,1)^3, exp(rnorm(n,1)). It also generates an unused vector x4.

glmRob *Fit a Robust Generalized Linear Model*

Description

Produces an object of class glmRob which is a Robust Generalized Linear Model fit.

Usage

```
glmRob(formula, family = binomial(), data, weights, subset,
       na.action, method = "cubif",
       model = TRUE, x = FALSE, y = TRUE,
       control = glmRob.control, contrasts = NULL, ...)
```

Arguments

formula	a formula expression as for other regression models, of the form response ~ predictors. See the documentation of lm and formula for details.
family	a family object - only binomial and poisson are implemented. See the documentation of glm for details.
data	an optional data frame in which to interpret the variables occuring in the formula.
weights	an optional vector of weights to be used in the fitting process. Should be NULL or a numeric vector.
subset	an expression specifying the subset of the data to which the model is fit. This can be a logical vector (which is replicated to have length equal to the number of observations), a numeric vector indicating which observations are included, or a character vector of the row names to be included. By default all observations are used.
na.action	a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (na.fail) is to create an error if any missing values are found. A possible alternative is na.omit which omits the rows that contain one or more missing values.
method	a character vector indicating the fitting method. The choices are method = "cubif" for the conditionally unbiased bounded influence estimator, method = "mallows" for Mallow's leverage downweighting estimator, and method = "misclass" for a consistent estimate based on the misclassification model. The Mallows' and misclassification estimators are only defined for logistic regression models with Bernoulli response.
model	a logical flag. If TRUE then the model frame is returned.
x	a logical flag. If TRUE then the model matrix is returned.
y	a logical flag. If TRUE then the response variable is returned.

contrasts	a list of contrasts to be used for some or all of the factors appearing as variables in the model formula. The names of the list should be the names of the corresponding variables, and the elements should either be contrast-type matrices (matrices with as many rows as levels of the factor and with columns linearly independent of each other and of a column of one's), or else they should be functions that compute such contrast matrices.
control	a list of iteration and algorithmic constants to control the conditionally unbiased bounded influence robust fit. See glmRob.cubif.control for their names and default values. These can also be set as arguments of glmRob itself.
...	control arguments maybe specified directly.

Value

a list with class glmRob containing the robust generalized linear model fit. See glmRob.object for details.

References

Copas, J. B. (1988). Binary Regression Models for Contaminated Data. JRSS 50, 225-265.

Kunsch, L., Stefanski L. and Carroll, R. (1989). Conditionally Unbiased Bounded-Influence Estimation in General Regression Models, with Applications to Generalized Linear Models. JASA 50, 460-466.

Carroll, R. J. and Pederson, S. (1993). On Robustness in the Logistic Regression Model. JRSS 55, 693-706.

Marazzi, A. (1993). Algorithms, routines and S functions for robust statistics. Wadsworth & Brooks/Cole, Pacific Grove, CA.

See Also

glmRob.control, glmRob.object, glmRob.cubif.control, glmRob.mallows.control, glmRob.misclass.control, glm.

Examples

```
data(breslow.dat)

glmRob(sumY ~ Age10 + Base4*Trt, family = poisson(),
       data = breslow.dat, method = "cubif")
```

---

glmRob.control *glmRob Control Parameters*

Description

Generates a list of control parameters for glmRob. The main purpose of this function is to implement the default behaviour for glmRob. Use the functions listed in the See Also section to generate control lists for the different robust estimators.

Usage

```
glmRob.control(method, ...)
```

Arguments

method	a character vector specifying which extimator the control parameters should be generated for. The choices are "cubif", "mallows", and "misclass".
...	additional arguments are included in the control (if appropriate for the estimator specified by method).

Value

a list of control parameters appropriate for the fitting method specified by the method argument.

See Also

glmRob.cubif.control, glmRob.mallows.control, glmRob.misclass.control.

---

glmRob.cubif *Robust GLM CUBIF Fitter*

Description

Robustly fit a generalized linear model using a **c**onditionally **u**nbiased **b**ounded **inf**luence ("cubif") estimator. This function is called by the high-level function glmRob when method = "cubif" (the default) is specified.

Usage

```
glmRob.cubif(x, y, intercept = FALSE, offset = 0,
        family = binomial(), null.dev = TRUE, control)
```

Arguments

x	a numeric model matrix.
y	either a numeric vector containing the response or, in the case of the binomial family, a two-column numeric matrix containing the number of successes and failures.
intercept	a logical value. If TRUE a column of ones is added to the design matrix.
offset	a numeric vector containing the offset.
family	a family object.
null.dev	a logical value. If TRUE the null deviance is computed.
control	a list of control parameters. See glmRob.cubif.control.

Value

See glmRob.object.

References

Kunsch, L., Stefanski L. and Carroll, R. (1989). Conditionally Unbiased Bounded-Influence Estimation in General Regression Models, with Applications to Generalized Linear Models. JASA **84**, 460–466.

Marazzi, A. (1993). Algorithms, routines and S functions for robust statistics. Wadsworth & Brooks/Cole, Pacific Grove, CA.

See Also

glmRob, glmRob.cubif.control.

---

glmRob.cubif.control *Control Parameters for the Bounded Influence Robust GLM Estimator*

Description

Allows users to set parameters for glmRob.

Usage

```
glmRob.cubif.control(epsilon = 0.001, maxit = 50, bpar = 2, cpar = 1.5,
  trc = FALSE, ...)
```

Arguments

epsilon	a positive numeric values specifying the convergence threshold for the parameters.
maxit	a positive integer giving the maximum number of iterations.
bpar	bpar
cpar	a single positive numeric value specifying the tuning constant for the initial estimate. This is the truncation value for the likelihood equation for the initial estimate. It determines the starting point of the iterative algorithm to calculate the final estimate.
trc	a logical value. If TRUE the number of the current iteration is printed on the screen.
...	additional arguments are ignored.

Value

a list is returned containing the values specified in the Arguments section.

See Also

glmRob.

`glmRob.mallows` *Mallows Type Estimator*

Description

Computes the Mallows Type Estimator provided by glmRob.

Usage

```
glmRob.mallows(x, y, control, offset, null.dev, family, Terms)
```

Arguments

`x`	model matrix
`y`	a numeric vector of Bernoulli responses.
`control`	control parameters.
`offset`	offset
`null.dev`	a logical value. If `TRUE` the null deviance is computed and stored.
`family`	a binomial family object.
`Terms`	the `Terms` object created in glmRob.

Value

a list similar to `glmRob.object`.

See Also

`link{glmRob}`

Examples

```
data(mallows.dat)

glmRob(y ~ a + b + c, data = mallows.dat, family = binomial(), method = 'mallows')
```

glmRob.mallows.control *Control for Mallows-type Robust GLM Estimator*

Description

Allows users to set parameters for glmRob.

Usage

```
glmRob.mallows.control(wt.fn = wt.carroll, wt.tuning = 8, ...)
```

Arguments

wt.fn	a weight function that might depend on a tuning constant. This function will be evaluated at the square root of the robust Mahalanobis distances of the covariates divided by their dimension.
wt.tuning	a tuning constant for wt.fn.
...	additional arguments are ignored.

Value

a list is returned, consisting of these parameters packaged to be used by glmRob(). The values for glmRob.mallows.control() can be supplied directly in a call to glmRob(). These values are filtered through glmRob.mallows.control() inside glmRob().

See Also

glmRob.

glmRob.misclass *Consistent Misclassification Estimator*

Description

Computes the consistent misclassification estimate provided in glmRob.

Usage

```
glmRob.misclass(x, y, control, offset, null.dev, family, Terms)
```

Arguments

x	model matrix.
y	response.
control	control parameters.
offset	offset.
null.dev	a logical value.
family	a binomial family object.
Terms	the Terms object computed in glmRob.

Value

a list similar to glmRob.object.

See Also

glmRob

Examples

```
data(leuk.dat)

glmRob(y ~ ag + wbc, data = leuk.dat, family = binomial(), method = 'misclass')
```

---

glmRob.misclass.control

Control for Misclassification Robust GLM Estimator

Description

Allows users to set parameters for glmRob.

Usage

```
glmRob.misclass.control(mc.gamma = 0.01, mc.maxit = 30, mc.trc = FALSE,
  mc.tol = 0.001, mc.initial = NULL, ...)
```

Arguments

mc.gamma	a real number between 0 and 1 that represents the probability of misclassification of a response variable.
mc.maxit	maximum number of iterations.
mc.trc	a logical value indicating whether a trace of the current parameter values is printed to the screen while the algorithm iterates.
mc.tol	convergence threshold.
mc.initial	a vector of initial values to start the iterations. If ommited, the coeficients resulting from a non-robust glm fit are used.
...	additional arguments are ignored.

Value

a list containing the parameters packaged to be used by glmRob. The values for glmRob.misclass.control can be supplied directly in a call to glmRob. These values are filtered through glmRob.misclass.control inside glmRob.

See Also

glmRob

---

glmRob.object *Robust Generalized Linear Model Fit*

Description

These are objects of class glmRob which represent the robust fit of a generalized linear regression model, as estimated by glmRob().

Value

coefficients	the coefficients of the linear.predictors, which multiply the columns of the model matrix. The names of the coefficients are the names of the single-degree-of-freedom effects (the columns of the model matrix). If the model is over-determined there will be missing values in the coefficients corresponding to inestimable coefficients.
linear.predictors	the linear fit, given by the product of the model matrix and the coefficients.
fitted.values	the fitted mean values, obtained by transforming linear.predictors using the inverse link function.
residuals	the residuals from the final fit; also known as working residuals, they are typically not interpretable.
deviance	up to a constant, minus twice the log-likelihood evaluated at the final coefficients. Similar to the residual sum of squares.
null.deviance	the deviance corresponding to the model with no predictors.
family	a 3 element character vector giving the name of the family, the link and the variance function.
rank	the number of linearly independent columns in the model matrix.
df.residuals	the number of degrees of freedom of the residuals.
call	a copy of the call that produced the object.
assign	the same as the assign component of an "lm" object.
contrasts	the same as the contrasts component of an "lm" object.
terms	the same as the terms component of an "lm" object.
ni	vector of the number of repetitions on the dependent variable. If the model is poisson then ni is a vector of 1s.

weights	weights from the final fit.
iter	number of iterations used to compute the estimates.
y	the dependent variable.
contrasts	the same as the contrasts term of an "lm" object. The object will also contain other components related to the numerical fit that are not relevant for the associated methods.

Methods

anova, coefficients, deviance, fitted.values, family, formula, plot, print, residuals, summary.

Structure

The following components must be included in a legitimate "glmRob" object. Residuals, fitted values, and coefficients should be extracted by the generic functions of the same name, rather than by the "\$" operator. The family function returns the entire family object used in the fitting, and deviance can be used to extract the deviance of the fit.

See Also

glmRob.

---

leuk.dat *Leuk Data*

Description

An exmaple data set for the misclassification fitter in glmRob.

Usage

```
data(leuk.dat)
```

Format

A data frame with 33 observations on the following 3 variables.

wbc a numeric vector.

ag a numeric vector.

y a numeric vector.

Source

Don't know - if you know please email the package maintainer.

Examples

```
data(leuk.dat)
```

lmRob *High Breakdown and High Efficiency Robust Linear Regression*

Description

Performs a robust linear regression with high breakdown point and high efficiency regression.

Usage

```
lmRob(formula, data, weights, subset, na.action,
      model = TRUE, x = FALSE, y = FALSE, contrasts = NULL,
      nrep = NULL, control = lmRob.control(...), ...)
```

Arguments

formula	a formula object, with the response on the left of a ~ operator, and the terms, separated by + operators, on the right.
data	a data.frame in which to interpret the variables named in the formula, or in the subset and the weights argument. If this is missing, then the variables in the formula should be on the search list. This may also be a single number to handle some special cases - see below for details.
weights	vector of observation weights; if supplied, the algorithm fits to minimize the sum of a function of the square root of the weights multiplied into the residuals. The length of weights must be the same as the number of observations. The weights must be nonnegative and it is strongly recommended that they be strictly positive, since zero weights are ambiguous, compared to use of the subset argument.
subset	expression saying which subset of the rows of the data should be used in the fit. This can be a logical vector (which is replicated to have length equal to the number of observations), or a numeric vector indicating which observation numbers are to be included, or a character vector of the row names to be included. All observations are included by default.
na.action	a function to filter missing data. This is applied to the model.frame after any subset argument has been used. The default (with na.fail) is to create an error if any missing values are found. A possible alternative is na.exclude, which deletes observations that contain one or more missing values.
model	a logical flag: if TRUE, the model frame is returned in component model.
x	a logical flag: if TRUE, the model matrix is returned in component x.
y	a logical flag: if TRUE, the response is returned in component y.
contrasts	a list giving contrasts for some or all of the factors appearing in the model formula. The elements of the list should have the same name as the variable and should be either a contrast matrix (specifically, any full-rank matrix with as many rows as there are levels in the factor), or else a function to compute such a matrix given the number of levels.

nrep	the number of random subsamples to be drawn. If "Exhaustive" resampling is being used, the value of nrep is ignored.
control	a list of control parameters to be used in the numerical algorithms. See lmRob.control for the possible control parameters and their default settings.
...	additional arguments are passed to the ccontrol functions.

Details

By default, the lmRob function automatically chooses an appropriate algorithm to compute a final robust estimate with high breakdown point and high efficiency. The final robust estimate is computed based on an initial estimate with high breakdown point. For the initial estimation, the alternate M-S estimate is used if there are any factor variables in the predictor matrix, and an S-estimate is used otherwise. To compute the S-estimate, a random resampling or a fast procedure is used unless the data set is small, in which case exhaustive resampling is employed. See lmRob.control for how to choose between the different algorithms.

Value

a list describing the regression. Note that the solution returned here is an approximation to the true solution based upon a random algorithm (except when "Exhaustive" resampling is chosen). Hence you will get (slightly) different answers each time if you make the same call with a different seed. See lmRob.control for how to set the seed, and see lmRob.object for a complete description of the object returned.

References

Gervini, D., and Yohai, V. J. (1999). A class of robust and fully efficient regression estimates; mimeo, Universidad de Buenos Aires.

Marazzi, A. (1993). *Algorithms, routines, and S functions for robust statistics*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Maronna, R. A., and Yohai, V. J. (2000). Robust regression with both continuous and categorical predictors. *Journal of Statistical Planning and Inference* **89**, 197–214.

Pena, D., and Yohai, V. (1999). A Fast Procedure for Outlier Diagnostics in Large Regression Problems. *Journal of the American Statistical Association* **94**, 434–445.

Yohai, V. (1988). High breakdown-point and high efficiency estimates for regression. *Annals of Statistics* **15**, 642–665.

Yohai, V., Stahel, W. A., and Zamar, R. H. (1991). A procedure for robust estimation and inference in linear regression; in Stahel, W. A. and Weisberg, S. W., Eds., *Directions in robust statistics and diagnostics, Part II*. Springer-Verlag.

See Also

lmRob.control, lmRob.object.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)
```

`lmRob.control` *Control Parameters for Robust Linear Regression*

Description

Allows the users to set values affecting the estimation procedure for robust regression in `lmRob`.

Usage

```
lmRob.control(tlo = 1e-4, tua = 1.5e-06, mxr = 50, mxf = 50, mxs = 50,
     tl = 1e-06, estim = "Final", initial.alg = "Auto", final.alg = "MM",
     seed = 1313, level = 0.1, efficiency = 0.9,
     weight = c("Optimal", "Optimal"), trace = TRUE)
```

Arguments

`tlo`	the relative tolerance in the iterative algorithms.
`tua`	the tolerance used for the determination of pseudo-rank.
`mxr`	the maximum number of iterations in the refinement step.
`mxf`	the maximum number of iterations for computing final coefficient estimates.
`mxs`	the maximum number of iterations for computing scale estimate.
`tl`	the tolerance for scale denominators. If a scale estimate becomes less than `tl`, the scale estimate is set equal to `tl`.
`estim`	parameter that determines the type of estimator to be computed. If `estim="Initial"`, only the initial estimates are computed; if `estim="Final"`, then final estimates are returned.
`initial.alg`	parameter that determines the algorithm for initial estimates. Valid choices are `"Auto"` for data-dependent algorithm, `"Random"` for random resampling, `"Exhaustive"` for exhaustive resampling, `"Fast"` for fast procedure, and `"Genetic"` for genetic algorithm. By default, `lmRob` uses `"Auto"`.
`final.alg`	parameter that determines the type of the final estimates. Valid choices are `"Adaptive"` for the robust efficient weighted least squares as proposed in Gervini and Yohai (1999), and `"MM"` for MM-estimate as proposed in Yohai, Stahel and Zamar (1991). By default, `lmRob` uses `"MM"`.
`seed`	seed parameter used in the random sampling and genetic algorithm for the computation of initial estimates.
`weight`	a character vector that determines the type of loss functions to be used. The first determines the loss function used for the initial estimates, and the second determines the loss function used for the final M-estimates. Valid choices are `"Optimal"` and `"Bisquare"`.
`level`	the level of significance of the test for bias of the final MM-estimates, if desired later on.
`efficiency`	the asymptotic efficiency of the final estimate.
`trace`	a logical flag: if `TRUE`, the remaining computing time will be printed.

Value

a list containing the values used for each of the control parameters.

See Also

lmRob.

Examples

```
data(stack.dat)
my.control <- lmRob.control(weight=c("Bisquare","Optimal"))
stack.bo <- lmRob(Loss ~ ., data = stack.dat, control = my.control)
```

---

lmRob.fit *Robust Fitter Functions for Linear Models*

Description

These are the basic computing engines called by lmRob used to robustly fit linear models. These functions are not intended to be used directly.

Usage

```
lmRob.fit(x, y, x1.idx = NULL, nrep = NULL, robust.control = NULL, ...)

lmRob.wfit(x, y, w, x1.idx = NULL, nrep = NULL, robust.control = NULL, ...)
```

Arguments

x	a numeric matrix containing the design matrix.
y	a numeric vector containing the linear model response.
w	a numeric vector containing the weights.
x1.idx	a numeric vector containing the indices of columns of the design matrix arising from the coding of factor variables.
nrep	the number of random subsamples to be drawn. If "Exhaustive" resampling is being used, the value of nrep is ignored.
robust.control	a list of control parameters to be used in the numerical algorithms. See lmRob.control for the possible control parameters and their default settings.
...	additional arguments.

`lmRob.fit.compute` *Fit a Robust Linear Model*

Description

Fits a robust linear model with high breakdown point and high efficiency estimates. This is used by `lmRob`, but not supposed to be called by the users directly.

Usage

```
lmRob.fit.compute(x, y, x1.idx = NULL, nrep = NULL, robust.control = NULL, ...)
```

Arguments

`x`	a numeric matrix containing the design matrix.
`y`	a numeric vector containing the linear model response.
`x1.idx`	a numeric vector containing the indices of columns of the design matrix arising from the coding of factor variables.
`nrep`	the number of random subsamples to be drawn. If `"Exhaustive"` resampling is being used, the value of `nrep` is ignored.
`robust.control`	a list of control parameters to be used in the numerical algorithms. See `lmRob.control` for the possible control parameters and their default settings.
`...`	additional arguments.

Value

an object of class `"lmRob"`. See `lmRob.object` for a complete description of the object returned.

References

Gervini, D., and Yohai, V. J. (1999). A class of robust and fully efficient regression estimates, mimeo, Universidad de Buenos Aires.

Marazzi, A. (1993). *Algorithms, routines, and S functions for robust statistics*. Wadsworth & Brooks/Cole, Pacific Grove, CA.

Maronna, R. A., and Yohai, V. J. (1999). Robust regression with both continuous and categorical predictors, mimeo, Universidad de Buenos Aires.

Yohai, V. (1988). High breakdown-point and high efficiency estimates for regression, *Annals of Statistics*, 15, 642-665.

Yohai, V., Stahel, W. A., and Zamar, R. H. (1991). A procedure for robust estimation and inference in linear regression, in Stahel, W. A. and Weisberg, S. W., Eds., *Directions in robust statistics and diagnostics, Part II*. Springer-Verlag.

See Also

`lmRob`, `lmRob.control`.

`lmRob.object` *Robust Linear Model Objects*

Description

These are objects of class `lmRob` which represent the robust fit of a linear regression model, as estimated by `lmRob` function.

Value

`coefficients`	vector of coefficients for the robust regression. If `est="final"`, these are final estimates; if `est="initial"`, these are initial estimates.
`T.coefficients`	the vector of coefficients for the initial estimate, if `est="final"`.
`scale`	the scale estimate computed using the initial estimates.
`residuals`	the residual vector corresponding to the estimates returned in `coefficients`.
`T.residuals`	the residual vector corresponding to the estimates returned in `T.coefficients`.
`fitted.values`	the fitted values corresponding to the estimates returned in `coefficients`.
`T.fitted.values`	the fitted values corresponding to the estimates returned in `T.coefficients`.
`cov`	the estimated covariance matrix of the estimates in `coefficients`.
`T.cov`	the estimated covariance matrix of the estimates in `T.coefficients`.
`rank`	the rank of the design matrix `x`.
`iter.refinement`	the number of iterations required to refine the initial estimates.
`df.residuals`	the degrees of freedom in the residuals (the number of rows in `x` minus the rank of `x`).
`est`	a character string that specifies the type of estimates returned. If `est="initial"`, the initial estimates are returned; if `est="final"`, the final estimates are returned.
`control`	a list of control parameters, passed to the function `lmRob` as the `robust.control` argument that produced the `lmRob` object.
`genetic.control`	a list of control parameters, passed to the function `lmRob` as the `genetic.control` argument that produced the `lmRob` object, if present.
`dev`	the robust deviance if final MM-estimates are returned.
`T.dev`	the robust deviance corresponding to initial S-estimates if applies.
`r.squared`	the fraction of variation in `y` explained by the robust regression on `x` corresponding to the final MM-estimates in `coefficients`, if applies.
`T.r.squared`	the fraction of variation in `y` explained by the robust regression on `x` corresponding to the initial S-estimates in `T.coefficients`, if applies.
`M.weights`	the robust estimate weights corresponding to the final MM-estimates in `coefficients`, if applies.

T.M.weights	the robust estimate weights corresponding to the initial S-estimates in T.coefficients, if applies.
iter.final.coef	the number of iterations required to compute the final MM-estimates of the coefficients, if applies.
call	an image of the call that produced the object, but with the arguments all named and with the actual formula included as the formula argument.
assign	the same as the assign component of an "lm" object.
contrasts	the same as the contrasts component of an "lm" object.
terms	the same as the terms component of an "lm" object.

Generation

This class of objects is returned from the lmRob function.

Methods

add1, anova, coef, deviance, drop1, fitted, formula, labels, plot, print, residuals, summary, update.

Structure

The following components must be included in a legitimate "lmRob" object:

See Also

lmRob.

---

lmRob.RFPE *Robust Final Prediction Errors*

Description

Computes the robust Final Prediction Errors (FPE) for a robust regression fit using M-estimates.

Usage

```
lmRob.RFPE(object, scale = NULL)
```

Arguments

object	an lmRob object.
scale	a numeric value specifying the scale estimate used to compute the robust FPE. Usually this should be the scale estimate from an encompassing model. If NULL, the scale estimate in object is used.

Value

a single numeric value giving the robust final prediction error.

See Also

lmRob, step.lmRob, drop1.lmRob.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)
lmRob.RFPE(stack.rob)
```

---

lognormRob *Robust Estimation of Lognormal Distribution Parameters*

Description

Robust estimation of lognormal distribution parameters.

Usage

```
lognormRob(x, estim = c("tdmean"), control = lognormRob.control(estim, ...), ...)
```

Arguments

x	a numeric vector containing the sample.
estim	a character string specifying which estimator to use.
control	a list of control parameters appropriate for the estimator in estim.
...	control parameters may also be given here.

Value

a list with class "fitdstn" containing the following elements:

estimate	a named numeric vector containing the parameter estimates.
sd	a named numeric vector containing the standard deviations of the parameter estimates. Missing in current implementation.
vcov	a numeric matrix containing the variance-covariance matrix of the estimated parameter vector. Missing in current implementation.
mu	a single numeric value containing an estimate of the mean.
V.mu	a single numeric value containing the variance of the estimated mean.
control	a list containing the control parameters used by the estimator.

Note

The `print` method displays the estimated parameters and their standard errors (in parentheses).

See Also

`lognormRob.control`, `fitdstnRob`.

`lognormRob.control` *Control Parameters for lognormRob*

Description

Create a list of control parameters for the `lognormRob` function.

Usage

```
lognormRob.control(estim, ...)
```

Arguments

`estim`	a character string specifying the estimator.
`...`	control parameters appropriate for the estimator given in `estim`.

Value

a list of control parameters appropriate for the specified estimator.

`lsRobTest` *Bias Test for Least-Squares Regression Estimates*

Description

Test for bias between least-squares and robust MM linear regression estimates.

Usage

```
lsRobTest(object, test = c("T2", "T1"), ...)
```

Arguments

`object`	an `lmRob` object (the output of the `lmRob` function).
`test`	either `"T1"` or `"T2"`. Selects the null hypothesis. T2 (the default): the error distribution not bias inducing. T1: the residual error distribution is normal.
`...`	additional arguments are ignored.

Examples

```
rob.fit <- lmRob(stack.loss ~ ., data = stackloss)
lsRobTest(rob.fit)
lsRobTest(rob.fit, test = "T1")
```

---

mallows.dat *Mallows Data*

Description

An exmaple data set for the mallows fitter in glmRob.

Usage

```
data(mallows.dat)
```

Format

A data frame with 70 observations on the following 4 variables.

`y` a numeric vector.

`a` a numeric vector.

`b` a numeric vector.

`c` a numeric vector.

Source

Don't know - if you know please email the package maintainer.

Examples

```
data(mallows.dat)
```

---

overlaidDenPlot.fdfm *Overlaid Density Plot*

Description

Plot the estimated densities over a histogram of the data.

Usage

```
overlaidDenPlot.fdfm(x, trunc = 1.0 - 1e-3, ...)
```

Arguments

x	an fdfm object.
trunc	if non NULL, the maximum x-value of the plot is the largest truncate quantile among the estimated distributions.
...	additional arguments are passed to the plotting functions.

Value

x is invisibly returned.

Examples

```
 data(los, package="robustbase")

## Not run:

 los.fm <- fit.models(c(Robust = "fitdstnRob", MLE = "fitdstn"),
                      x = los, densfun = "gamma")


 los.fm <- fit.models(c(Robust = "fitdstnRob", MLE = "fitdstn"),
                      x = los, densfun = "weibull")

 overlaidDenPlot.fdfm(los.fm, xlab = "x-axis label", ylab = "y-axis label",
                     main = "Plot Title")

## End(Not run)
```

---

plot.covfm *Plot Method*

Description

Generic plot method for objects with classes "covfm", "covRob", and "covClassic".

Usage

```
## S3 method for class 'covfm'
plot(x, which.plots = c(4, 3, 5), ...)
## S3 method for class 'covRob'
plot(x, which.plots = c(4, 3, 5), ...)
## S3 method for class 'covClassic'
plot(x, which.plots = c(4, 3, 5), ...)
```

Arguments

x	an oject of class "covClassic", "covRob", or "covfm".
which.plots	either "ask", "all", or an integer vector specifying which plots to draw. If which.plots is an integer vector, use the plot numbers given here (or in the "ask" menu). The plot options are (2) Eigenvalues of Covariance Estimate, (3) Sqrt of Mahalanobis Distances, (4) Ellipses Matrix, and (5) Distance - Distance Plot.
...	additional arguments are passed to the plot subfunctions.

Details

The actual plot functions are only implemented for "fit.models" objects. When this method is dispatched on an object of class "cov" or "covRob" the object is cast as a "fit.models" object containing a single element and plotted with `plot.covfm`. The actual plotting is done by the subfunctions listed in the See Also section.

Value

x is invisibly returned.

Side Effects

The requested plots are drawn on a graphics device.

See Also

plot, covClassic, covRob, fit.models, ddPlot.covfm, ellipsesPlot.covfm, screePlot.covfm, distancePlot.covfm.

Examples

```
data(woodmod.dat)

woodm.cov <- covClassic(woodmod.dat)
woodm.covRob <- covRob(woodmod.dat)

plot(woodm.cov)
plot(woodm.covRob)

woodm.fm <- fit.models(list(Robust = "covRob", Classical = "covClassic"),
                       data = woodmod.dat)
plot(woodm.fm)
```

plot.fdfm *fdfm Plot Method*

Description

Comparison plots for fitted univariate distributions.

Usage

```
## S3 method for class 'fdfm'
plot(x, which.plots = 2:3, ...)
```

Arguments

x	an fdfm object.
which.plots	either "ask", "all", or an integer vector specifying which plots to draw. In the latter case, use the plot numbers given in the "ask" menu.
...	additional arguments are passed to the plotting functions.

Value

x is invisibly returned.

Examples

```
data(los, package = "robustbase")
los.fm <- fit.models(c(Robust = "fitdstnRob", MLE = "fitdstn"),
                     x = los, densfun = "gamma")
plot(los.fm)
```

plot.glmRob *Diagnostic Regression Plots*

Description

Creates a set plots useful for assessing a robustly fitted generalized linear model. The plot options are (2) Deviance Residuals vs. Predicted Values, (3) Response vs. Predicted Values, (4) Normal QQ Plot of Pearson Residuals, (5) QQ Plot of Deviance Residuals, (6) Standardized Deviance Residuals vs. Robust Distances, (7) Standardized Deviance Residuals vs. Index (Time), and (8) Sqrt of abs(Deviance Residuals) vs. Fitted Values.

Usage

```
## S3 method for class 'glmRob'
plot(x, which.plots = c(2, 5, 7, 6), ...)
```

Arguments

x	a glmRob object.
which.plots	either "ask", "all", or an integer vector specifying which plots to draw. If which.plots is an integer vector, use the plot numbers given in the description above (or in the "ask" menu).
...	additional arguments are pass to the ploting subfunctions which are listed in the see also section.

Details

This function casts the glmRob object as an glmfm object containing a single model. The actual ploting is then done by the function plot.glmfm.

Value

x is invisibly returned.

Side Effects

The selected plots are drawn on a graphics device.

References

Atkinson, A. C. (1985). Plots, Transformations and Regression. New York: Oxford University Press.

See Also

plot, glmRob, plot.glmfm.

---

plot.lmRob *Diagnostic Regression Plots*

Description

Creates a set plots useful for assessing a robustly fitted linear model. The plot options are (2) Normal QQ-Plot of Residuals, (3) Estimated Kernel Density of Residuals, (4) Robust Residuals vs Robust Distances, (5) Residuals vs Fitted Values, (6) Sqrt of abs(Residuals) vs Fitted Values, (7) Response vs Fitted Values, (8) Standardized Residuals vs Index (Time), (9) Overlaid Normal QQ-Plot of Residuals, and (10) Overlaid Estimated Density of Residuals. For simple linear regression models there is also the option to have a side-by-side plots of the the fit over a scatter plot of the data.

Usage

```
## S3 method for class 'lmRob'
plot(x, which.plots = c(5, 2, 6, 4), ...)
```

Arguments

x	an lmRob object.
which.plots	either "ask", "all", or an integer vector specifying which plots to draw. If which.plots is an integer vector, use the plot numbers given in the description above (or in the "ask" menu).
...	additional arguments are pass to the ploting subfunctions which are listed in the see also section.

Details

This function casts the lmRob object as an lmfm object containing a single model. The actual ploting is then done by the function plot.lmfm.

Value

x is invisibly returned.

Side Effects

The selected plots are drawn on a graphics device.

References

Atkinson, A. C. (1985). Plots, Transformations and Regression. New York: Oxford University Press.

See Also

plot, lmRob, plot.lmfm.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)
plot(stack.rob, which.plots = 6)
```

---

predict.glmRob *Predict Method for Robust Generalized Linear Model Fits*

Description

Obtains predictions and optionally estimates standard errors of those predictions from a fitted robust generalized linear model object.

Usage

```
## S3 method for class 'glmRob'
predict(object, newdata,
       type = c("link", "response", "terms"), se.fit = FALSE,
       terms = labels(object), dispersion = NULL, ...)
```

Arguments

object	a glmRob object.
newdata	optionally, a data frame in which to look for variables with which to predict. If omitted, the fitted linear predictors are used.
type	a character string specifying the type of prediction. The choices are "link" for predictions on the scale of the linear predictor, "response" for predctions on the scale of the response, and "terms" which returns a matrix giving the fitted values for each term in the model formula on the scale of the linear predictor.
se.fit	a logical value. If TRUE then standard errors for the predictions are computed.
terms	when type = "terms" all terms are returned. A character vector specifies which terms are to be returned.
dispersion	the dispersion of the generalized linear model fit to be assumed in computing the standard errors. If omitted, that returned by 'summary' applied to the object is used.
...	additional arguments required by the generic predict method.

Value

If se.fit = FALSE, a vector or matrix of predictions. Otherwise a list with components:

fit	Predictions
se.fit	Estimated standard errors

See Also

glmRob, predict.

Examples

```
data(breslow.dat)
bres.rob <- glmRob(sumY ~ Age10 + Base4 * Trt, family = poisson(), data = breslow.dat)
predict(bres.rob)
```

predict.lmRob *Use predict() on an lmRob Object*

Description

Extracts the fitted values from an lmRob object and returns a matrix of predictions.

Usage

```
## S3 method for class 'lmRob'
predict(object, newdata, type = "response", se.fit = FALSE, terms = labels(object), ...)
```

Arguments

object	an lmRob object.
newdata	a data frame containing the values at which predictions are required. This argument can be missing, in which case predictions are made at the same values used to compute the object. Only those predictors referred to in the right side of the formula in object need be present by name in newdata.
type	a single character value specifying the type of prediction. The only choice is "response". If "response" is selected, the predictions are on the scale of the response.
se.fit	a logical value. If TRUE, pointwise standard errors are computed along with the predictions.
terms	this argument is presently unused.
...	additional arguments required by the generic predict function.

Value

a vector of predictions, or a list consisting of the predictions and their standard errors if se.fit = TRUE.

Warning

predict can produce incorrect predictions when the newdata argument is used if the formula in object involves *data-dependent* transformations, such as poly(Age, 3) or sqrt(Age - min(Age)).

See Also

lmRob, predict.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)
predict(stack.rob)
predict(stack.rob, newdata = stack.dat[c(1,2,4,21), ], se.fit = TRUE)
```

qqPlot.fdfm *Comparison Quantile-Quantile Plot*

Description

Side-by-side quantile-quantile plots of the sample versus estimated quantiles.

Usage

```
qqPlot.fdfm(x, qqline = TRUE, ...)
```

Arguments

x	an fdfm object.
qqline	a logical value. If TRUE a qqline is included in each panel.
...	additional arguments are passed to xyplot.

Examples

```
data(los, package = "robustbase")
los.fm <- fit.models(c(Robust = "fitdstnRob", MLE = "fitdstn"),
                     x = los, densfun = "gamma")
qqPlot.fdfm(los.fm, xlab = "x-axis label", ylab = "y-axis label",
            main = "Plot Title", pch = 4, col = "purple")
```

rb.lmRob *Robust Bootstrap Standard Errors*

Description

Computes a robust bootstrap estimate of the standard error for each coefficient estimate in a robustly fitted linear model. This function is called by summary.lmRob and is not intended to be called directly by users.

Usage

```
rb.lmRob(lmRob.object, M = 1000, seed = 99, fixed = TRUE)
```

Arguments

lmRob.object	an lmRob object.
M	a positive integer giving the number of bootstrap subsamples.
seed	a positive integer specifying the seed for the random number generator.
fixed	a logical value. This should be set to TRUE.

Value

a numeric vector of robust bootstrap standard error estimates.

See Also

lmRob, summary.lmRob.

---

residuals.glmRob *Residuals Methods for glmRob Objects*

Description

Residuals methods for glmRob objects.

Usage

```
## S3 method for class 'glmRob'
residuals(object, type = c("deviance", "pearson", "working", "response"), ...)
```

Arguments

object	a glmRob object.
type	the type of residuals to be returned.
...	additional arguments are ignored.

Value

a numeric vector containing the residuals.

---

screePlot.covfm *Comparison Screeplot*

Description

Draws overlaid screeplots for the models in a covfm object.

Usage

```
screePlot.covfm(x, npcs, strip = "", ...)
```

Arguments

x	a "covfm" object.
npcs	a postive integer value specifying the number of components to be plotted.
strip	a character string printed in the "strip" at the top of the plot.
...	additional arguments are passed to xyplot.

Value

the `trellis` object is invisibly returned.

Examples

```
data(woodmod.dat)
woodm.fm <- fit.models(list(Robust = "covRob", Classical = "covClassic"),
                       data = woodmod.dat)
screePlot.covfm(woodm.fm, main = "Plot Title", xlab = "x-axis label",
                ylab = "y-axis label", pch = 4:5)
```

---

stack.dat *Brownlee's Stack-Loss Data*

Description

These data are from the operation of a plant for the oxidation of ammonia to nitric acid, measured on 21 consecutive days.

Usage

```
data(stack.dat)
```

Format

This data frame contains the following variables:

Loss the percentage of ammonia lost (times 10).

Air.Flow air flow into the plant

Water.Temp cooling water inlet temperature.

Acid.Conc. acid concentration as a percentage (coded by subtracting 50 and then multiplying by 10).

Source

Brownlee, K.A. (1965). *Statistical Theory and Methodology in Science and Engineering*. New York: John Wiley & Sons, Inc.

Examples

```
data(stack.dat)
stack.dat
```

step.lmRob *Build a Model in a Stepwise Fashion*

Description

Performs stepwise model selection on a robustly fitted linear model. Presently only the backward stepwise procedure is implemented.

Usage

```
step.lmRob(object, scope, scale,
           direction = c("both", "backward", "forward"),
           trace = TRUE, keep = NULL, steps = 1000, fast = FALSE, ...)
```

Arguments

object	an lmRob object.
scope	either a formula or a list with elements lower and upper each of which is a formula. The terms in the right-hand-side of lower are always included in the model and the additional terms in the right-hand-side of upper are the candidates for inclusion/exclusion from the model. If a single formula is given, it is taken to be upper and lower is set to the empty model. The . operator is interpreted in the context of the formula in object.
scale	a single numeric value containing a residual scale estimate. If missing, the scale estimate in object is used.
direction	a character value specifying the mode of stepwise search. The possibilities are "both", "backward", and "forward", with a default of "backward". Presently only "backward" stepwise searches are implemented.
trace	a logical value. If TRUE, information is printed during stepwise search.
keep	a filter function whose input is a fitted model object and the associated AIC statistic, and whose output is arbitrary. Typically keep will select a subset of the components of the object and return them. The default is not to keep anything.
steps	an integer value specifying the the maximum number of steps to be considered. The default is 1000 (essentially as many as required). It is typically used to stop the process early.
fast	a logical value. If TRUE the robust initial estimate (used when fitting each of the reduced models) is replaced by a weighted least squares estimate using the robust weights computed for the current fit. Note: the fast algorithm does not work in this version of the Robust Library.
...	additional arguments required by the generic step function.

Details

Presently only backward stepwise selection is supported. During each step the Robust Final Prediction Error (as computed by the function `lmRob.RFPE`) is calculated for the current model and for each sub-model achievable by deleting a single term. The function then either steps to the sub-model with the lowest Robust Final Prediction Error or, if the current model has the lowest Robust Final Prediction Error, terminates. The scale estimate from `object` is used to compute the Robust Final Prediction Error throughout the procedure unless the `scale` argument is provided in which case the user specified value is used.

Value

the model with the lowest Robust Final Prediction Error encountered during the stepwise procedure is returned. Additionally, an `anova` element corresponding to the steps taken in the search is appended to the returned object. If a `keep` function was provided then the kept values can be found in the `keep` element of the returned object.

See Also

`lmRob`, `lmRob.RFPE`, `drop1.lmRob`.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)

## The default behavior is to try dropping all terms ##
step.lmRob(stack.rob)

## Keep Water.Temp in the model ##
my.scope <- list(lower = . ~ Water.Temp, upper = . ~ .)
step.lmRob(stack.rob, scope = my.scope)
```

`summary.covfm` *Summary Method*

Description

The generic summary method for objects of class "covClassic", "covRob", and "covfm".

Usage

```
## S3 method for class 'covClassic'
summary(object, ...)
## S3 method for class 'covRob'
summary(object, ...)
## S3 method for class 'covfm'
summary(object, ...)
```

Arguments

object	an object of class "covClassic", "covRob", or "covfm".
...	additional arguments for the summary method.

Value

an object of class "summary.covClassic", "summary.covRob", or "summary.covfm" respectively. Objects of class "summary.cov" and "summary.covRob" have the following components. Objects of class "summary.covfm" are lists whose elements are "summary.cov" and "summary.covRob" objects.

call	an image of the call that produced the object with all the arguments named.
cov	a numeric matrix containing the estimate of the covariance/correlation matrix.
center	a numeric vector containing the estimate of the location vector.
evals	a numeric vector containing the eigenvalues of the covariance/correlation matrix.
dist	a numeric vector containing the Mahalanobis distances. Only present if distance = TRUE in the call.
corr	a logical flag. If corr = TRUE then cov contains an estimate of the correlation matrix of x.

See Also

summary, covClassic, covRob, fit.models.

Examples

```
  data(woodmod.dat)
  woodm.cov <- covClassic(woodmod.dat)
## IGNORE_RDIFF_BEGIN
  summary(woodm.cov)
## IGNORE_RDIFF_END

  woodm.covRob <- covRob(woodmod.dat)
  summary(woodm.covRob)

  woodm.fm <- fit.models(list(Robust = "covRob", Classical = "covClassic"),
                         data = woodmod.dat)
  summary(woodm.fm)
```

summary.glmRob *Summarizing Robust Generalized Linear Model Fits*

Description

Compute a summary of the robustly fitted generalized linear model.

Usage

```
## S3 method for class 'glmRob'
summary(object, correlation = TRUE, ...)
```

Arguments

object	a glmRob object.
correlation	a logical value. If TRUE then the correlation matrix of the coefficients is included in the summary.
...	additional arguments required by the generic summary function.

Value

The summary is returned in a list of class summary.glmRob and contains the following components:

comp1	Description of 'comp1'
comp2	Description of 'comp2'

...

Examples

```
data(breslow.dat)
bres.rob <- glmRob(sumY ~ Age10 + Base4*Trt, family = poisson(), data = breslow.dat)
bres.sum <- summary(bres.rob)
bres.sum
```

summary.lmRob *Summarizing Robust Linear Model Fits*

Description

Compute a summary of the robustly fitted linear model.

Usage

```
## S3 method for class 'lmRob'
summary(object, correlation = FALSE, bootstrap.se = FALSE, ...)
```

Arguments

object	an lmRob object.
correlation	a logical value. If TRUE then the correlation matrix of the coefficients is included in the summary.
bootstrap.se	a logical value. If TRUE then bootstrap standard error estimates are included in the summary.
...	additional arguments required by the generic summary function.

Value

The summary is returned in a list of class summary.lmRob and contains the following components:

sigma	a single numeric value containing the residual scale estimate.
df	a numeric vector of length 3 containing integer values: the rank of the model matrix, the residual degrees of freedom, and the number of coefficients in the model.
cov.unscaled	the unscaled covariance matrix; i.e, the matrix that, when multiplied by the estimate of the error variance, yields the estimated covariance matrix for the coefficients.
correlation	the correlation coefficient matrix for the coefficients in the model.
...	the remaining components are the same as the corresponding components in an lmRob object. Use the names function to obtain a list of the components.

Examples

```
data(stack.dat)
stack.rob <- lmRob(Loss ~ ., data = stack.dat)
stack.sum <- summary(stack.rob)
stack.sum
stack.bse <- summary(stack.rob, bootstrap.se = TRUE)
stack.bse
```

---

test.lmRob *Various Tests of Robust Regression Estimates*

Description

Conducts test for bias of robust MM-estimates and Least Squares (LS) estimates against S-estimates, or permutation test of the slope estimate in a straight line fit.

Usage

```
test.lmRob(object, type = "bias", level = NULL, n.permute = 99)
```

Arguments

object	an object of class "lmRob".
type	character string. Valid choices are "bias" for bias test, or "permutation" for permutation test.
level	the level of the test for bias of MM-estimate. By default, the level component of object$robust.control is used.
n.permute	a positive integer value specifying the number of permutations to use.

Value

the p-value of the permutation test, or an object of class "biasMM" representing the bias test, in which case the following components ARE included:

mm	a list describing the test of bias for final MM-estimates, with the following components: stat, the t-statistic; pchi, a chi-squared p-value; qchi, the quantile of the chi-squared distribution with degrees of freedom equal to object\$rank corresponding to the probability input in the level (object$robust.control$level).
ls	a list describing the test of bias for LS-estimates, with the following components: stat, the t-statistic; pchi, a chi-squared p-value.
level	the level of the test for bias of MM-estimate.

References

Yohai, V., Stahel, W. A., and Zamar, R. H. (1991). A procedure for robust estimation and inference in linear regression, in Stahel, W. A. and Weisberg, S. W., Eds., *Directions in robust statistics and diagnostics, Part II*. Springer-Verlag.

---

update.lmRob *Update an lmRob Model Object*

Description

A method for the generic update function for objects inheriting from class lmRob. See update for the general behavior of this function and for the interpretation of the arguments.

Usage

```
## S3 method for class 'lmRob'
update(object, formula., ..., evaluate = TRUE)
```

Arguments

object	an lmRob object.
formula.	a modeling formula, such as y ~ a + b. A single dot . on either side of the ~ gets replaced by the left or right side of the formula in object. The dot on the left can be omitted. By default, it refits object using the same formula as in object.
evaluate	a logical value. If TRUE the updated call is evaluated and returned. Otherwise the unevaluated call is returned.
...	additional arguments passed to the generic update function.

Details

If formula. is missing, update.lmRob alternates between the initial estimates and final estimates. Otherwise (when formula. is present), update.lmRob functions just like update.default.

Value

either a new updated object, or else an unevaluated expression for creating such an object.

See Also

lmRob. update.

---

weibullRob *Robust Estimation of Weibull Distribution Parameters*

Description

Robust estimation of Weibull distribution parameters.

Usage

```
weibullRob(x, estim = c("M", "tdmean"), control = weibullRob.control(estim, ...), ...)
```

Arguments

x	a numeric vector containing the sample.
estim	a character string specifying which estimator to use.
control	a list of control parameters appropriate for the estimator in estim.
...	control parameters may also be given here.

Value

a list of class "fitdstn" containing the following elements:

`estimate`	a named numeric vector containing the parameter estimates.
`sd`	a named numeric vector containing the standard deviations of the parameter estimates.
`vcov`	a numeric matrix containing the variance-covariance matrix of the estimated parameter vector.
`mu`	a single numeric value containing an estimate of the mean.
`V.mu`	a single numeric value containing the variance of the estimated mean.
`control`	a list containing the control parameters used by the estimator.

The `print` method displays the estimated parameters and their standard errors (in parentheses).

See Also

`weibullRob.control`, `fitdstnRob`.

`weibullRob.control` *Control Parameters for weibullRob*

Description

Create a list of control parameters for the `weibullRob` function.

Usage

```
weibullRob.control(estim, ...)
```

Arguments

`estim`	a character string specifying the estimator.
`...`	control parameters appropriate for the estimator given in `estim`.

Value

a list of control parameters appropriate for the specified estimator.

`weight.funs` *Weight Functions Psi, Rho, Chi*

Description

These functions compute the weights used by lmRob and its associated methods.

Usage

```
psi.weight(x, ips = 1, xk = 1.06)
rho.weight(x, ips = 1, xk = 1.06)
psp.weight(x, ips = 1, xk = 1.06)
chi.weight(x, ips = 1, xk = 1.06)
```

Arguments

`x`	a numeric vector.
`ips`	integer determining the weight function: `ips = 1` "optimal", `ips = 2` rescaled bisquare, `ips = 3` Huber, `ips = 4` smoothed Huber, which is currently only available for `psi.*()` and its derivative `psp.*()`.
`xk`	a numeric value specifying the tuning constant.

Details

See the section "Theoretical Details", p. 58-59, in chapter 2 of '`Robust.pdf`'.

Value

a numeric vector, say `r` of the same length as `x`, containing the function values $r_i = f(x_i)$.

Examples

```
x <- seq(-4,4, length=401)
f.x <- cbind(psi = psi.weight(x), psp = psp.weight(x),
             chi = chi.weight(x), rho = rho.weight(x))
es <- expression(psi(x), {psi*minute}(x), chi(x), rho(x))
leg <- as.expression(lapply(seq_along(es), function(i)
          substitute(C == E, list(C=colnames(f.x)[i], E=es[[i]]))))
matplot(x, f.x, type = "l", lwd = 1.5,
        main = "psi.weight(.) etc -- 'optimal'")
abline(h = 0, v = 0, lwd = 2, col = "#D3D3D380") # opaque gray
legend("bottom", leg, inset = .01,
       lty = 1:4, col = 1:4, lwd = 1.5, bg = "#FFFFFFC0")
```

woodmod.dat *Modified Wood Data*

Description

The explanatory variables from the Modified Data on Wood Specific Gravity analyzed in Rousseeuw and Leroy (1987).

Note that `data(wood, package="robustbase")` contains the same data, and additionally the y-variable.

Usage

```
data(woodmod.dat)
```

Format

This data frame contains the following variables:

V1 number of fibers per square milimeter in Springwood (coded by dividing by 1000).

V2 number of fibers per square milimeter in Summerwood (coded by dividing by 10000).

V3 fraction of Springwood.

V4 fraction of light absorption by Springwood.

V5 fraction of light absorption by Summerwood.

Source

Rousseeuw, P. J., and Leroy, A. M. (1987). *Robust Regression and Outlier Detection*. New York: Wiley.

Examples

```
data(woodmod.dat)
woodmod.dat

data(wood, package = "robustbase")
stopifnot(data.matrix(woodmod.dat) ==
          data.matrix(wood [,1:5]))
```

Index

∗ **datasets**
breslow.dat, 5
leuk.dat, 26
mallows.dat, 36
stack.dat, 46
woodmod.dat, 56
∗ **hplot**
ddPlot.covfm, 10
distancePlot.covfm, 11
ellipsesPlot.covfm, 13
overlaidDenPlot.fdfm, 36
plot.covfm, 37
plot.fdfm, 39
plot.glmRob, 39
plot.lmRob, 40
qqPlot.fdfm, 44
screePlot.covfm, 45
∗ **maximum-likelihood**
fitdstn, 14
∗ **methods**
anova.glmRob, 3
anova.lmRob, 4
drop1.lmRob, 12
lmRob.object, 32
plot.covfm, 37
plot.glmRob, 39
plot.lmRob, 40
predict.glmRob, 41
predict.lmRob, 43
step.lmRob, 47
summary.covfm, 48
summary.glmRob, 50
summary.lmRob, 50
update.lmRob, 52
∗ **models**
glmRob, 18
glmRob.object, 25
lmRob, 27
lmRob.fit.compute, 31
lmRob.object, 32
∗ **multivariate**
covClassic, 6
covRob, 7
∗ **regression**
anova.glmRob, 3
anova.lmRob, 4
drop1.lmRob, 12
glmRob, 18
glmRob.control, 19
glmRob.cubif, 20
glmRob.cubif.control, 21
glmRob.mallows, 22
glmRob.mallows.control, 23
glmRob.misclass, 23
glmRob.misclass.control, 24
glmRob.object, 25
lmRob, 27
lmRob.control, 29
lmRob.fit.compute, 31
lmRob.object, 32
lmRob.RFPE, 33
lsRobTest, 35
predict.glmRob, 41
predict.lmRob, 43
rb.lmRob, 44
step.lmRob, 47
summary.glmRob, 50
summary.lmRob, 50
test.lmRob, 51
update.lmRob, 52
∗ **robust**
anova.glmRob, 3
anova.lmRob, 4
covClassic, 6
covRob, 7
drop1.lmRob, 12
fitdstnRob, 15
gammaRob, 16

gammaRob.control, 17
glmRob, 18
glmRob.control, 19
glmRob.cubif, 20
glmRob.cubif.control, 21
glmRob.mallows, 22
glmRob.mallows.control, 23
glmRob.misclass, 23
glmRob.misclass.control, 24
glmRob.object, 25
lmRob, 27
lmRob.control, 29
lmRob.fit, 30
lmRob.fit.compute, 31
lmRob.object, 32
lmRob.RFPE, 33
lognormRob, 34
lognormRob.control, 35
lsRobTest, 35
predict.glmRob, 41
predict.lmRob, 43
rb.lmRob, 44
residuals.glmRob, 45
step.lmRob, 47
summary.glmRob, 50
summary.lmRob, 50
test.lmRob, 51
update.lmRob, 52
weibullRob, 53
weibullRob.control, 54
weight.funs, 55

∗ **utilities**

covRob.control, 9
gen.data, 17

add1, *33*
anova, *3*, *4*, *12*, *33*
anova.glmRob, 3
anova.glmRoblist, *3*
anova.glmRoblist *(*anova.glmRob*)*, 3
anova.lmRob, 4
anova.lmRoblist, *4*
anova.lmRoblist *(*anova.lmRob*)*, 4

breslow.dat, 5

chi.weight *(*weight.funs*)*, 55
coef, *33*
cov.wt, *7*
covClassic, 6, *9*, *38*, *49*
covMcd, *8–10*
CovMest, *8*, *9*
CovOgk, *8–10*
covRob, *6*, 7, *7*, *10*, *38*, *49*
covRob.control, 9, *9*
CovSde, *8–10*

ddPlot.covfm, 10, *38*
deviance, *33*
distancePlot.covfm, 11, *38*
drop1, *12*, *33*
drop1.lmRob, 12, *34*, *48*

ellipsesPlot.covfm, 13, *38*

fit.models, *38*, *49*
fitdistr, *14*
fitdstn, 14, *15*
fitdstnRob, *14*, 15, *16*, *35*, *54*
fitted, *33*
formula, *33*

gammaRob, *15*, 16
gammaRob.control, *16*, 17
gen.data, 17
glm, *19*
glmRob, *3*, 18, *20–26*, *40*, *42*
glmRob.control, 19, *19*
glmRob.cubif, 20
glmRob.cubif.control, *19, 20*, 21, *21*
glmRob.mallows, 22
glmRob.mallows.control, *19, 20*, 23
glmRob.misclass, 23
glmRob.misclass.control, *19, 20*, 24, *25*
glmRob.object, *19*, *21, 22*, *24*, 25

labels, *33*
leuk.dat, 26
lmRob, *4*, 27, *30–35*, *41*, *43*, *45*, *47, 48*, *53*
lmRob.control, *28*, 29, *31*
lmRob.fit, 30
lmRob.fit.compute, 31
lmRob.object, *12*, *28*, *31*, 32
lmRob.RFPE, 33, *48*
lmRob.wfit *(*lmRob.fit*)*, 30
lognormRob, *15*, 34
lognormRob.control, 35, *35*
lsRobTest, 35

mallows.dat, 36

names, *51*

overlaidDenPlot.fdfm, 36

plot, *33*, *38*, *40, 41*
plot.covClassic *(*plot.covfm*)*, 37
plot.covfm, 37
plot.covRob *(*plot.covfm*)*, 37
plot.fdfm, 39
plot.glmfm, *40*
plot.glmRob, 39
plot.lmfm, *41*
plot.lmRob, 40
predict, *42, 43*
predict.glmRob, 41
predict.lmRob, 43
print, *33*
psi.weight *(*weight.funs*)*, 55
psp.weight *(*weight.funs*)*, 55

qqPlot.fdfm, 44

rb.lmRob, 44
residuals, *33*
residuals.glmRob, 45
rho.weight *(*weight.funs*)*, 55

screePlot.covfm, *38*, 45
stack.dat, 46
step.lmRob, *34*, 47
summary, *33*, *49–51*
summary.covClassic *(*summary.covfm*)*, 48
summary.covfm, 48
summary.covRob *(*summary.covfm*)*, 48
summary.glmRob, 50
summary.lmRob, *44, 45*, 50

test.lmRob, 51

update, *33*, *52, 53*
update.lmRob, 52

var, *7*

weibullRob, *15*, 53
weibullRob.control, 54, *54*
weight.funs, 55
woodmod.dat, 56