

Package ‘regr0’

November 8, 2011

Type Package

Title Building regression models

Version 1.0-2

Author Werner Stahel

Maintainer Werner Stahel <stahel@stat.math.ethz.ch>

Depends MASS, survival

Suggests nnet

Description This package helps to perform efficient model building for regression models. It also contains some useful utility functions.

License GPL

LazyLoad yes

R topics documented:

regr-package	2
asinperc	4
c.months	4
condquant	5
d.blast	6
d.fossiles	6
d.rehab	7
d.surveyenvir	8
doc	9
drop1.mlm	10
drop1.regr	11
dropdata	12
fitcomp	13
fitted.polr	14
getmeth	15
last	15
legendr	16
leverage	17
logst	17

mframe	18
modelTable	19
nainf.exclude	21
plcoord	22
plfitpairs	23
plmatrix	23
plot.regr	25
plot.xdistResScale	29
plres2x	29
plresx	30
plTA.polr	33
polrregr	34
predict.regr	35
print.regr	36
regr	37
residuals.polr	39
residuals.survreg	41
robrange	42
showd	43
simresiduals	44
stamp	45
sumna	46
terms2order	46
Tobit	47
warn	48
xdistResdiff	49

Index **51**

regr-package	<i>Tools for Building Regression Models</i>
--------------	---

Description

This package helps to perform efficient model building for regression models. It also contains some useful utility functions.

Details

Package:	regr
Type:	Package
Version:	1.0
Date:	2008-08-20
License:	GPL
LazyLoad:	yes

The function `regr` is a wrapper function for fitting regression models of many types by calling well-known R functions:

- Ordinary linear regression,

- logistic, Poisson, other generalized linear models, also for
- ordinal or nominal response variable.

The output of the function is somewhat different from the output of the base R functions `lm`, `glm`, `polr` etc. – it should be more informative for the task of model building.

The next most important function is the `plot` method for results of `regr`, which gives more complete regression diagnostic plots than the specific plot methods of the mentioned incorporated R functions.

An important part of residual analysis which is often neglected is plotting residuals versus explanatory variables, often called term plots. This is part of `plot.regr`, but can also be called independently by calling `plresx`.

If a `regr` result is printed, the output is more informative than the classical tables.

The package contains the methods for the usual extractor functions, like `residuals`, `predict`, ..., as well as for `drop1`, `add1`, ...

The package contains a number of additional functions that should be useful for efficient data analysis. They include

- `plresx` to plot residuals against explanatory variables,
- `plres2x` to produce a plot showing interactions between two continuous regressors,
- `plmatrix`, a version of `pairs` that displays any rectangular scatterplot matrix and can also be used to produce the full usual quadratic scatterplot matrix for many variables in appropriate junks with panels that are still large enough to be useful.
- `lassoselect` to do model selection.

The package also includes some additional utility functions, like

- `logst` for log transformation of variables with zeros,
- `showd` for printing a "representative" part of a dataset.

For documenting the steps of the data analysis, there are several documenting functions:

`tit` and `doc` add attributes to data and results, which should allow for tracing the history of these objects (not yet thoroughly implemented).

`stamp` adds tracking information to plots, consisting of a project title, a step label, and the date when the plot is produced.

Some default options are added to `options` to reflect user preferences for output, see the help entries for `stamp`.

Author(s)

Werner A. Stahel, ETH Zurich, <stahel@stat.math.ethz.ch>

References

There will be a website on <http://stat.ethz.ch/~stahel/regression>

See Also

`library(car)`

Examples

```
r.savings <- regr(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)
plot(r.savings)
```

`asinperc`*arc sine Transformation*

Description

Calculates the sqrt arc sine of $x/100$, rescaled to be in the unit interval.
This transformation is useful for analyzing percentages or proportions of any kind.

Usage`asinperc(x)`**Arguments**

`x` vector of data values

Value

vector of transformed values

Note

This very simple function is provided in order to simplify formulas

Author(s)

Werner A. Stahel, ETH Zurich

Examples`asinperc(seq(0,100,10))`

`c.months`*Names of months and weekdays*

Description

Vector(s) containing the months' and weekdays' names. `c.mon` contains the 3 letter abbreviations of the months

Usage

`c.months`
`c.mon`
`c.weekdays`

Value

vector of names

Author(s)

Werner A. Stahel, ETH Zurich

Examples

```
c.months
c.mon[4:6]
c.weekdays
```

 condquant

Quantiles of a Conditional Distribution

Description

Calculates quantiles of a conditional distribution, as well as corresponding random numbers. The condition is simply to restrict the distribution (given by `dist`) to a range (given by `x`)

Usage

```
condquant(x, dist = "normal", sig = 1)
```

Arguments

<code>x</code>	matrix with 2 columns or vector of length 2 giving the limits for the conditional distribution
<code>dist</code>	(unconditional) distribution. Currently, only normal and logistic are implemented.
<code>sig</code>	standard deviation for normal distribution

Value

Matrix consisting of a row for each row of `x` and the following columns:

<code>median</code>	Median
<code>lowq, uppq</code>	lower and upper quartiles
<code>random</code>	random number according to the conditional distribution (one for each row)
<code>prob</code>	probability of the condition being true

Note

The function is needed for `residuals.polr` and `residuals.survreg`

Author(s)

Werner A. Stahel, Seminar for Statistics, ETH Zurich

d.blast	<i>tremor caused by blasting for excavation of a tunnel, with explanatory information</i>
---------	---

Description

When excavating a tunnel, blasting may cause damage in houses in the neighborhood. It is important to keep the tremor caused by the blasting under control. To this end, tremor is measured. There were 4 devices to measure tremor. They were moved to different locations while the excavation advanced.

Usage

```
data(d.blast)
```

Format

A data frame with 388 observations on the following 7 variables.

no number of blasting
 datetime date and time of the blasting
 device measurement device
 charge charge (loading) in kg
 distance between place of blasting and location
 tremor measured in mm/s
 location house in which tremor was measured

Source

Basler & Hofmann, Zurich

Examples

```
data(d.blast)
doc(d.blast)
coplot(log10(tremor)~distance|charge*location, data=d.blast)
```

d.fossiles	<i>Shapes of Shells and Environmental Variables</i>
------------	---

Description

This dataset relates shapes of Gephyrocapsa shells with environmental variables

Usage

```
data(d.fossiles)
```

Format

The format is: chr "d.fossiles"

Details

NA

Source

Joerg Bollmann, Jorijntje Henderiks and Bernhard Brabec, Geological Institute, ETH Zurich,

References

Joerg Bollmann, Jorijntje Henderiks and Bernhard Brabec, Geological Institute, ETH Zurich, Marine Micropaleontology 29, 319-350 (1997)

Examples

```
data(d.fossiles)
showd(d.fossiles)
regr(cbind(sAngle,lLength,rWidth)~SST.Mean+Salinity+lChlorophyll+region+N,
      data=d.fossiles)
```

d.rehab *~~ data name/kind ... ~~*

Description

~~ A concise (1-5 lines) description of the dataset. ~~

Usage

```
data(d.rehab)
```

Format

The format is: chr "d.rehab"

Details

If necessary,

Source

...

Examples

```
data(d.rehab)
```

d.surveyenvir

Survey on Environment

Description

A survey was conducted on attitudes towards environmental problems

Usage

```
data(d.surveyenvir)
```

Format

A data frame with 2038 observations on the following 10 variables.

age numeric. age in years

sex gender, either m for men or f for women

education level of education, with levels no.training, apprenticeship, no.degree, college, uni

location type of living environment, with levels Zentrum, Stadt.dicht, Stadt.locker Rand.dicht, Rand.locker, Land, sonstig

townsize size of town, with categories <2000, 2000-4999, 5000-19999, 20000-49999 50000-99999, 100000-499999, >500000 inhabitants

party political party, CDU, SPD, FDP, NPD, DKP, Gruene, sonstige, keine

disturbance level of disturbance, nicht etwas, ziemlich, sehr

gov Is the government active enough? Levels does.enough, not.enough

responsibility Who has the main responsibility? Answers individuals, government, both

weight weight of the observation according to sampling design

Source

"Umweltschutz im Privatbereich.", Erhebung des EMNID, Zentralarchiv fuer empirische Sozialforschung der Universitaet Koeln

Examples

```
data(d.surveyenvir)
t.r <- regr(disturbance~age+education+location, data=d.surveyenvir)
```

`doc`*Define and obtain the doc or tit attribute*

Description

The attributes `doc` and `tit` describe an object, typically a data frame or a model. `tit` should be a short description (title), `doc` should contain all documentation useful to identify the origin and the changes made to the object.

The `doc` and `tit` functions set them and extract these attributes.

Usage

```
doc(x)
tit(x)
doc(x) <- value
tit(x) <- value
```

Arguments

<code>x</code>	object to which the <code>doc</code> or <code>tit</code> attribute should be attached or from which it is obtained
<code>value</code>	character vector (<code>doc</code>) or string (<code>tit</code>) to be stored

Details

Plotting and printing functions may search for the `tit` attribute or even for the `doc` attribute, depending on `c.env$docout`.

`doc(x) <- text` will append the existing `doc(x)` text to the new one unless the first element of `text` equals `"^"`, whereas `tit(x) <- string` replaces `tit(x)`.

Value

`doc` and `tit` return the respective attributes of object `x`

Author(s)

Werner A. Stahel, ETH Zurich

Examples

```
data(d.blast)
doc(d.blast)
doc(d.blast) <- "I will use this dataset in class soon."
doc(d.blast)
```

drop1.mlm	<i>Drop All Possible Single Terms from or Add Terms to a Multivariate Model</i>
-----------	---

Description

drop1.mlm / add1.mlm performs a test for the suitability of dropping / adding each term in scope from / to a multivariate regression or manova model

Usage

```
## S3 method for class 'mlm'
drop1(object, scope = NULL,
       test = c("Wilks", "Pillai", "Hotelling-Lawley", "Roy"), total = TRUE,
       add=FALSE, ...)
## S3 method for class 'mlm'
add1(object, scope = NULL,
      test = c("Wilks", "Pillai", "Hotelling-Lawley", "Roy"), ...)
```

Arguments

object	a multivariate regression or manova object, possibly with class regr
scope	as in general drop1 / add1
test	multivariate test to be used
total	should the test for null model also be performed?
add	if TRUE, the function performs add1 instead of dropping terms
...	not used

Value

matrix containing statistics for each term

Note

Note that regr performs such a (Wilks) test automatically. Bug: extract.AIC is not yet available

Author(s)

modification of summary.manova by W. Stahel

See Also

[drop1.regr](#), [drop1.lm](#)

Examples

```

data(d.fossiles)
r.lm <-
  lm(cbind(sAngle,lLength,rWidth)~SST.Mean+Salinity+lChlorophyll+region+N,
      data=d.fossiles)
drop1(r.lm) ## should be the same as summary(manova(...)) for its last row.
r.mregr <-
  regr(cbind(sAngle,lLength,rWidth)~SST.Mean+Salinity+lChlorophyll+region+N,
        data=d.fossiles)
r.mregr$drop1
drop1(r.mregr, test="Pillai")

```

drop1.regr	<i>Drop All Possible Single Terms from or Add Terms to a Model Fitted by regr</i>
------------	---

Description

drop1.regr / add1.regr performs a test for the suitability of dropping / adding each term in scope from / to a regression model fitted by `regr`

Usage

```

## S3 method for class 'regr'
drop1(object, scope = NULL, scale = 0, test = NULL, k = 2,
       sorted = FALSE, add=FALSE, ...)
## S3 method for class 'regr'
add1(object, scope = NULL, scale = 0, test = NULL, k = 2,
      sorted = FALSE, ...)

```

Arguments

object	a fitted model object
scope	a formula giving the terms to be considered for dropping or adding. Defaults to the terms in the 'object's formula for drop1 and to all first order interactions between these and squared terms in the continuous variables for add1 (see terms2order)
scale	an estimate of the error variance if applicable. Will be recalculated from the fit if ==0 or NULL
test	see drop1
k	the penalty constant in AIC / Cp
sorted	if TRUE, the result is sorted according to the AIC or, if absent, the p value
add	converts drop1.regr into an add1 method
...	further arguments passed to the model specific methods

Value

A data.frame of class "anova" summarizing the differences in fit between the models obtained by dropping or adding the terms and the original model given by object

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[drop1](#), [add1](#)

Examples

```
data(d.blast)
t.r <- regr(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
drop1(t.r)
add1(t.r)
```

dropdata

Drop Observations from a Data.frame

Description

Allows for dropping observations (rows) determined by row names or factor levels from a data.frame or matrix.

Usage

```
dropdata(data, rowid, incol = "row.names")
```

Arguments

data	a data.frame or matrix
rowid	vector of character strings identifying the rows to be dropped
incol	name or index of the column used to identify the observations (rows)

Value

The data.frame or matrix without the dropped observations. Attributes are kept.

Note

Ordinary subsetting by `[...]` drops attributes like `doc` or `tit`. Furthermore, row or column names cannot be used like indices, simply by a preceding minus sign to drop the respective data.

Author(s)

Werner A. Stahel, ETH Zurich

Examples

```
dd <- rbind(a=1:3,b=4:6,c=7:9)
dropdata(dd,"b")
```

fitcomp

*Component Effects for a Model Fit***Description**

Determines effects of varying each of the given variables while all others are held constant. This function is mainly used to produce plots of residuals versus explanatory variables, also showing component effects.

Usage

```
fitcomp(object, data = NULL, vars = NULL, se = FALSE,
        xm = NULL, xfromdata = FALSE, nxcomp = 51)
```

Arguments

object	a model fit, result of a fitting function
data	data frame in which the variables are found
vars	character vector of variable names which are used in addition to all variables in the formula of object
se	if TRUE, standard errors will be returned
xm	named vector of values of the fixed (central) point from which the individual variables are varied in turn. Defaults to the componentwise median of quantitative variables and the modes of factors.
xfromdata	if TRUE, the components effects will be evaluated for the data values in data, otherwise, the range of each variable is filled with nxcomp equidistant points. This is useful for residual plots with component effects
nxcomp	number of points used for each (quantitative) variable if xfromdata is FALSE

Details

The component effect is defined as the curve of fitted values obtained by varying the explanatory variable, keeping all the other variables at their "central value" (the mean of continuous variables and the mode of factors).

Value

A list consisting of

comp	component effects
x	the values of the x variables for which the effects have been calculated
xm	the values of the x variables that are held fixed while one of the variables is varied
se	standard errors of the component effects, if required by the argument se

Author(s)

Werner A. Stahel, ETH Zurich

See Also[predict](#)**Examples**

```
data(d.blast)
t.r <- regr(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
fitcomp(t.r,se=TRUE)
```

fitted.polr

*Fitted Values for polr objects***Description**

Calculates Fitted Values for ordered response regression. In addition to the types of the usual version, this function allows for extracting the link type of predictions.

Usage

```
## S3 method for class 'polr'
fitted(object, type = "link", na.action = object, ...)
```

Arguments

object	an object that inherits from class polr
type	type of fitted values (as with predict). link asks for the linear predictor, class returns the most likely (integer) Y values probs returns probabilities for each value of Y
na.action	object that possibly contains an na.action component. If it is of the class exclude, the fitted values will be expanded to fit the number of rows in the data.frame used for fitting the regression.
...	not used

Value

Vector of fitted values

Author(s)

Werner A. Stahel, Seminar for Statistics, ETH Zurich

Examples

```
house.plr <- polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
house.fit <- fitted(house.plr, type="link") # only possible with regr version
```

getmeth	<i>get S3 method of a generic function</i>
---------	--

Description

identical to `getS3method`

Usage

```
getmeth(fn, mt)
```

Arguments

fn	name of generic function, quoted or unquoted
mt	name of method, quoted or unquoted

Value

Source code of the method

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[getS3method](#)

Examples

```
getmeth(print, regr)
```

last	<i>Last Elements of a Vector</i>
------	----------------------------------

Description

Selects or drops the last element or the last `n` elements of a vector

Usage

```
last(data, n = 1)
```

Arguments

data	vector from which to select or drop
n	if >0 , last selects the last <code>n</code> elements from the result. if <0 , the last <code>abs(n)</code> elements are dropped, and the first <code>length(data)-abs(n)</code> elements from the result

Value

The selected elements of the vector

Note

This is a very simple function. It is defined mainly for selecting from the results of other functions without storing them.

Author(s)

Werner Stahel

Examples

```
x <- runif(rpois(1,10))
last(sort(x), 3)
last(sort(x), -5)
```

legendr

Add a Legend to a Plot

Description

Adds a legend to a plot as does [legend](#). This function just expresses the position relative to the range of the coordinates

Usage

```
legendr(x = 0.05, y = 0.95, legend, ...)
```

Arguments

x	position in horizontal direction, between 0 for left margin and 1 for right margin
y	position in vertical direction, between 0 for bottom margin and 1 for top margin
legend	text of the legend
...	arguments passed to legend

Value

See [legend](#)

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[legend](#)

Examples

```
ts.plot(ldeaths, mdeaths, fdeaths,xlab="year", ylab="deaths", lty=c(1:3))
legendr(0.7,0.95, c("total","female","male"), lty=1:3)
```

leverage	<i>Get leverage values</i>
----------	----------------------------

Description

Extracts the leverage component of a fit object using the `na.action` component if available

Usage

```
leverage(fit)
```

Arguments

<code>fit</code>	an object containing a component <code>fit\$leverage</code> and possibly a component <code>fit\$na.action</code>
------------------	--

Value

The vector `fit$leverage`, possibly expanded by missing values if `fit$na.action` has class `na.replace`

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[influence](#)

Examples

```
data(d.blast)
r.blast <-
  regr(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
t.lev <- leverage(r.blast)
length(t.lev)
dim(d.blast) ## number of observations is 388
length(r.blast$leverage) ## 362
## only observations without missings occur in $leverage
```

logst	<i>Started Logarithmic Transformation</i>
-------	---

Description

Transforms the data by a \log_{10} transformation, modifying small and zero observations such that the transformation yields finite values.

Usage

```
logst(data, calib=data, threshold=NULL, mult = 1)
```

Arguments

data	a vector or matrix of data, which is to be transformed
calib	a vector or matrix of data used to calibrate the transformation(s), i.e., to determine the constant c needed
threshold	constant c that determines the transformation
mult	a tuning constant affecting the transformation of small values, see Details

Details

Small values are determined by the threshold c. If not given by the argument threshold, then it is determined by the quartiles q_1 and q_3 of the non-zero data as those smaller than $c = q_1 / (q_3 / q_1)^{mult}$. The rationale is that for lognormal data, this constant identifies 2 percent of the data as small. Beyond this limit, the transformation continues linear with the derivative of the log curve at this point. See code for the formula.

The function chooses log10 rather than natural logs because they can be backtransformed relatively easily in the mind.

Value

the transformed data. The value c needed for the transformation is returned as `attr(, "threshold")`.

Note

The names of the function alludes to Tudey's idea of "started logs".

Author(s)

Werner A. Stahel, ETH Zurich

Examples

```
dd <- c(seq(0,1,0.1),5*10^rnorm(100,0,0.2))
dd <- sort(dd)
r.dl <- logst(dd)
plot(dd, r.dl, type="l")
abline(v=attr(r.dl,"threshold"),lty=2)
```

mframe

Multiple Frames for Plotting

Description

This is a short-cut to set some graphical parameters

Usage

```
mframe(mfrow = NULL, mfcol = NULL, mft = NULL, row = TRUE,
oma = c(0, 0, 2, 1), mar = getOption("mar"),
mgp = getOption("mgp"), ...)
```

Arguments

<code>mfrow, mfcol</code>	number of rows and columns of panels. The default is 1 for both, which will reset the subdivision of the plotting page.
<code>mft</code>	total number of panels, to be split into <code>mfrow</code> and <code>mfcol</code> by the function. The result depends on the current aspect ratio (ratio of height to width) of the plotting area.
<code>row</code>	if TRUE, the panels will be used by rows, otherwise, by columns
<code>oma, mar, mgp, ...</code>	further graphical parameters passed to <code>par(...)</code> .

Details

The function calls `par`. Its purpose is to simplify a call like `par(mfrow=c(3,4))` to `mframe(3,4)` and to set some defaults differently from `par`.

Value

a named `list` containing the old values of the parameters, as for `par`.

Author(s)

Werner A. Stahel, ETH Zurich

See Also

`par`; `mult.fig` from package `sfsmisc` with very similar goals.

Examples

```
mframe(2,3)
mframe(mft=15) ## will split the plotting area into >= 15 panels,
mframe() ## reset to 1 panel
```

modelTable

Collect results for different regression models

Description

`modelTable` produces a table that shows which variables are contained in various models. Typically, the different models are the results of various model selection functions and model development strategies.

`format.modelTable` and `print.modelTable` are used to turn the results into a nice form.

`compareTerms` returns a table indicating which terms appear in which models.

Usage

```

modelTable(models, data = NULL, seq = NULL)

## S3 method for class 'modelTable'
format(x, digits=3,
       stars = c("***", "** ", "*  ", ":  ", ".  "), sep="", ...)
## S3 method for class 'modelTable'
print(x, tex = FALSE, ...)

compareTerms(... , list=NULL, seq = NULL)

```

Arguments

models	a character vector containing the names of the models to be collected, or a list of model fits.
data	data on which all models are evaluated.
seq	The sequence in which the terms should appear in the table. If incomplete, the terms mentioned in seq will be the first rows in the result.
x	object of class modelTable
digits	to be used in formatting
stars	symbols used to indicate level of significance. They will be used for the intervals delimited by $c(-1, 0.001, 0.01, 0.05, 0.1, 1)$
sep	separator. "&" is used to prepare LaTeX format.
tex	if TRUE, the output is can be used in (La-)TeX.
...	other arguments, passed to print or format. In compareTerms, the models for which the terms should be compared can be listed as in c or list etc.
list	A list of models.

Value

modelTable produces a list of class "modelTable", with components

coef	the coefficients of the models, collected in a matrix. The columns of the matrix correspond to the models, the rows, to the terms. For terms with more than 1 degree of freedom, the matrix element is NA. If possible, standardized coefficients are given.
p	p values, in the same form
sd.terms	standard deviations for terms with 1 degree of freedom
sigma	estimated error standard deviations of the models

format.modelTable returns a character matrix and thereby prepares for print.modelTable. If needed, the output of format.modelTable can be reordered or subsetted.

Author(s)

Werner A. Stahel, ETH Zurich

See Also

on formatting and printing of P-values, the standard R functions [symnum](#), [format.pval](#).

Examples

```
data(d.blast)
r.blast <-
  regr(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
r.bl2 <- update(r.blast, ~.+location:log10(distance)+I(log10(charge)^2) )
r.mt <- modelTable(list(large=r.bl2,original=r.blast))
format(r.mt)
r.mt
compareTerms(r.blast, r.bl2, seq=c("log10(distance)", "log10(charge)"))
```

nainf.exclude	<i>Drop Rows Containing NA or Inf</i>
---------------	---------------------------------------

Description

Drops the rows of a data frame that contain an NA, an NaN, or an Inf value

Usage

```
nainf.exclude(object, ...)
```

Arguments

object	an R object, typically a data frame
...	further arguments special methods could require.

Details

This is a simple modification of [na.omit](#) and [na.exclude](#)

Value

The value is of the same type as the argument object, with possibly less elements.

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[na.omit](#)

Examples

```
t.d <- data.frame(V1=c(1,2,NA,4), V2=c(11,12,13,Inf))
nainf.exclude(t.d)
```

plcoord

Determines Values for Plotting with Limited "Inner" Plot Range

Description

For plots with an "inner plot range" (see Details) this function converts the data values to the coordinates in the plot

Usage

```
plcoord(x, range = NULL, limfac = 3, limext = 0.1)
```

Arguments

x	data to be represented
range	vector of 2 elements giving the inner plot range. Data beyond the given interval will be non-linearly transformed to fit within the (outer) plot margins. Defaults to robrange (x, fac=fac).
limfac	factor used to determine the default of range
limext	factor for extending the range to determine the outer plot range

Details

When plotting data that contain outliers, the non-outlying data is represented poorly. Rather than simply clipping outliers, one can split the plotting area into an inner region, where the (non-outlying) data is plotted as usual, and a plot area margin, in which outliers are represented on a highly non-linear scale that allows to display them all.

This function converts the data to the coordinates used in the graphical display, and also returns the inner range for plotting.

Value

vector of coordinates used for plotting, that is, unchanged x values for those within the range and transformed values for those outside.

Attributes:

attr(,"range")	the "inner" plot range, either the argument range or the values determined by default.
attr(,"nmod")	the number of modified observations

Author(s)

Werner Stahel

See Also

[robrange](#)

Examples

```
x <- c(rnorm(20),rnorm(3,5,20))
xmod <- plcoord(x)
plot(x,xmod)
plot(xmod)
abline(h=attr(xmod,"range"),lty=3, lwd=2)
```

plfitpairs

Plot Fitted Values for Multinomial Regression

Description

Displays a scatterplot matrix of fitted values obtained from a multinomial regression.

Usage

```
plfitpairs(object, ssize = 0.02, main = NULL)
```

Arguments

object	result of fitting multinom, possibly through regr .
ssize	symbol size
main	plot title

Value

none.

Author(s)

Werner A. Stahel, ETH Zurich

plmatrix

Scatterplot Matrix

Description

Plots a scatterplot matrix, for which the variables shown horizontally do not necessarily coincide with those shown vertically. If desired, the matrix is divided into several blocks such that it fills more than 1 plot page.

Usage

```
plmatrix(x, y = NULL, data = NULL, panel = 1.panel,
  nrows = 0, ncols = nrows, save = TRUE,
  robrange. = FALSE, range. = NULL, pch = NULL, col = 1,
  reference = 0, ltyref = 3, log = "", xaxs = "r", yaxs = "r", vnames = NULL,
  main = "", cex.points = NA, cex.lab = 0.9, cex.text = 1.3, cex.title = 1,
  bty = "o", oma = NULL, ...)
```

Arguments

x	data for columns (x axis), or formula defining column variables. If it is a formula containing a left hand side, the left side variables will be used last.
y	data or formula for rows (y axis). Defaults to x
data	data.frame containing the variables in case x or y is a formula
panel	a function that generates the marks of the individual panels, see Details. Defaults essentially to points or text depending on the argument pch
nrows	number of rows of panels on a page
ncols	number of columns of panels on a page
save	if y is not provided and save==TRUE, the first row and the last column are suppressed.
robrange.	if TRUE, robust plot ranges will be used
range.	plot ranges, given as a matrix with 2 rows (min, max) and colnames identifying the variables.
pch	plotting character. A vector of integers, characters or strings can also be given for the default panel function
col	color(s) to be used for plotting the observations
reference	coordinates for reference lines to be shown in the panels. A named vector can be used to define a value for each or any variable.
ltyref	line type for reference lines
log	specifies logarithmic scale of axes. "x" asks for log scale on horizontal axis, "y", on vertical axis, "xy", on both axes.
xaxs, yaxs	styles for x and y axis, see par
vnames	labels for the variables
main	main title for the plot (to be repeated on each plot page)
cex.points	character expansion for showing the observations
cex.lab, cex.text	character expansion for variable labels in the margin and in the "diagonal", respectively, relative to cex
cex.title	character expansion for the main title
bty	box type for each panel, see par
oma	width of outer margins, ee par
...	further arguments passed to the panel function

Details

If x or y is a data.frame, it is converted to a numerical matrix.

The panel function can be user written. It needs ≥ 6 arguments, which are given:

- the values of the horizontal variable,
- the values of the vertical variable,
- the index of the variable shown horizontally, among the y variables,
- the index of the variable shown vertically, among the x variables,
- argument pch, and

- argument col

Since large scatterplot matrices lead to tiny panels, `plmatrix` splits the matrix into blocks of at most `nrows` rows and `ncols` columns. If these numbers are missing, they default to `nrows=5` and `ncols=6` for landscape pages, and to `nrows=8` and `ncols=5` for portrait pages.

Value

none

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[pairs](#)

Examples

```
plmatrix(iris[,1:4], main="Iris", pch=as.numeric(iris[, "Species"]))
```

plot.regr

Diagnostic Plots for Regr Objects

Description

Diagnostic plots for fitted regression models: Residuals versus fit (Tukey-Anscombe plot) and/or target variable versus fit; Absolute residuals versus fit to assess equality of error variances; Normal Q-Q plot (for ordinary regression models); Residuals versus leverages to identify influential observations; Residuals versus sequence; and Residuals versus explanatory variables. These plots are adjusted to the type of regression model.

Usage

```
## S3 method for class 'regr'
plot(x, data = NULL, markprop = NULL, lab = NULL, cex.lab = 0.7,
     mf = NULL, mfcol=FALSE, mar=c(3,3,2,1), mgp=c(2,0.7,0),
     oma = 2*(prod(mf)>1), cex=par("cex"), ask = NULL,
     multnrows = 0, multncols = 0,
     lty = c(1, 2, 5, 3, 4, 6, 1, 1), lwd = c(1, 1, 2, 1, 1.5, 1, 1, 1),
     colors = getOption("colors.ra"), pch = NULL, col = NULL,
     main = NULL, cex.title = NULL,
     wsymbols = NULL, symbol.size = NULL,
     smooth = TRUE, smooth.par = NA, smooth.iter = NA,
     smooth.sim = 19, nxsmooth = 51,
     plotsselect = NULL,
     weights = NULL, leverage.cooklim = 1:2,
     reslim = TRUE, reslimfac=3.0, reslimext=0.1,
     yaxp=NULL, resaxp=NULL, stresaxp=NULL,
     ylim = TRUE, ylimfac=3.0, ylimext=0.1, jitterbinary=TRUE,
     glm.restype = "deviance", condprobrange=c(0.05,0.8),
     sequence = NA, xplot = TRUE, x.se = FALSE, x.smooth = smooth,
     addcomp=FALSE, ...)
```

Arguments

<code>x</code>	regr object (result of a call to <code>regr</code>). This is the only argument that is needed. All others have useful defaults.
<code>data</code>	data set where explanatory variables are found
<code>markprop</code>	proportion of extreme residuals to be labeled
<code>lab</code>	labels of observations used to identify them
<code>cex.lab</code>	character size for lab
<code>mf</code>	vector of 2 elements, indicating the number of rows and columns of panels on each plot page. Defaults to <code>c(2, 2)</code> , except for multivariate models, where it adjusts to the number of target variables
<code>mfcol</code>	if TRUE, the panel will be filled columnwise
<code>mar, mgp, oma, cex, ask</code>	see <code>?par</code>
<code>multnrows, multncols</code>	number of rows and columns of panels on one page, for residuals of multivariate regression only
<code>lty, lwd, colors</code>	three vectors of length 6 each, defining the line types, line widths, and colors to be used for ... [1] observations; [2] reference lines; [3] smooth; [4] simulated smooths; [5] component effects in <code>plresx</code> ; [6] for confidence bands of component effects. [7] (random) observations in the case of <code>glm.restype="cond.quant"</code> ; [8] bars showing conditional quantiles for <code>glm.restype="cond.quant"</code> ;
<code>pch</code>	plotting character to use for unlabeled points
<code>col</code>	color(s) to use for points
<code>main</code>	main title to be used on each page, defaults to the formula of object. If <code>main</code> starts by ":", this colon is dropped and the formula is appended to the remainder. If <code>getOption("doc")</code> is <code>\$>1\$</code> , then the <code>tit</code> attribute of <code>x</code> (see <code>tit</code>) is printed on a separate line.
<code>cex.title</code>	character size for title
<code>wsymbols</code>	if TRUE, points are displayed by a bubble whose size indicates the weight as given in <code>object\$weights</code> . If it is a vector of appropriate size, it will be used to determine the sizes of such bubbles.
<code>symbol.size</code>	determines the size of weight symbols
<code>smooth</code>	if TRUE, smooths are added to the plots where appropriate. If it is a function, it is assumed to define the smooths.
<code>smooth.par, smooth.iter</code>	arguments to the smooth function
<code>smooth.sim</code>	number of simulated smooths added to each plot
<code>nxsmooth</code>	number of values on the horizontal axes where smooths are evaluated
<code>plotselect</code>	which plots should be shown? See Details

weights	if TRUE, residuals will be plotted versus object\$weights. Alternatively, a vector of weights can be specified
leverage.cooklim	levels of Cook distance for which contours are plotted in the leverage plot
reslim	limits for plotting residuals directly. Beyond these limits, points are shown in the margins with squeezes scale
reslimfac, reslimext	arguments passed to robrange if selected
resaxp, stresaxp	axis parameters used for residuals and standardized residuals, respectively, passed to argument yaxp of plot, see ?par.
ylim, ylimfac, ylimext, yaxp	same, for plotting y values
jitterbinary	logical: should variables with only 2 distinct values be treated as factors, i.e., jittered?
glm.restype	type of residuals to be used for glm model. In addition to those allowed in residuals() for glm objects, type cond.quant is possible for (ungrouped) binary regression. See ?residuals.polr for an explanation.
condprobrange	if conditional probabilities are shown by bars: show only those that are contained in condprobrange (vector of length 2)
sequence	logical: should residuals be plotted versus sequence?
xplot	if TRUE, residuals will be plotted versus all (raw) explanatory variables in the model (by calling plresx)
x.se	logical: Should confidence bands be drawn in plot of residuals versus explanatory variables?
x.smooth	like smooth, for the latter plots
addcomp	logical: should component effects be added to residuals for residuals versus input variables plots?
...	further arguments to be passed to plot

Details

Argument `plotselect` is used to determine which plots will be shown. It should be a named vector of numbers indicating

- 0** do not show
- 1** show without smooth
- 2** show with smooth

The default is `c(yfit=0, ta=3, tascale = NA, weights = NA, qq = NA, leverage = 2, resmatrix = 1, qqmult = 3)`. Modify this vector to change the selection and the sequence in which the plots appear. Alternatively, provide a named vector defining all plot that should be shown, like `plotselect = c(ta = 2, leverage = 1)`.

The names of `plotselect` refer to:

- yfit** response versus fitted values
- ta** residuals versus fitted values (Tukey-Anscombe plot)
- tascale** residuals versus fitted values, defaults to TRUE for ordinary regression, FALSE for glm and others

weights residuals versus weights

qq normal Q-Q plot, defaults to TRUE for ordinary regression, FALSE for glm and others

leverage residuals versus leverage (hat diagonal)

resmatrix scatterplot matrix of residuals for multivariate regression

qqmult qq plot for Mahalanobis lengths versus sqrt of chisquare quantiles.

In the Tukey-Anscombe plot, the reference line indicates a "contour" line with constant values of the response variable, $Y = \hat{y} + r = \text{constant}$. It has slope -1. It is useful to judge whether any curvature shown by the smooth might disappear after a nonlinear, monotone transformation of the response.

Value

none

Note

This is a function under development. Future versions may behave differently and may not be compatible with this version.

Author(s)

Werner A. Stahel, ETH Zurich

See Also

plot.lm

Examples

```
r.savings <- regr(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)
plot(r.savings)

data(d.blast)
r.blast <-
  regr(log10(tremor)~location+log10(distance)+log10(charge),
        data=d.blast)
plot(r.blast, seq=FALSE, xplot=~.+no)

data(d.fossiles)
r.foss <-
  regr(cbind(sAngle,lLength,rWidth)~SST.Mean+Salinity+lChlorophyll+region+N,
        data=d.fossiles)
plot(r.foss, plotselect=c(ta=3, resmatrix=1, qqmult=1),xplot=FALSE)
```

plot.xdistResScale *Illustrate Completeness Test for Regression Model*

Description

Plots mean absolute differences of residuals against mean x-distance of pairs of observations, classified on the basis of their x-distance. The distance is proportional to the Mahalanobis distance in the design space.

Usage

```
## S3 method for class 'xdistResScale'
plot(x, xlab = "distance in x space",
     ylab = "average sq residual difference /2", col.aux = "grey30")
```

Arguments

x	an object of class xdistResScale
xlab, ylab	label for axes
col.aux	color to be used for auxiliary lines none. See package vignette. Werner A. Stahel, ETH Zurich xdistResScale

Examples

```
data(d.blast)
rr <- regr(tremor~distance+charge, data=d.blast)
## an inadequate model!
xdrd <- xdistResdiff(rr)
plot(xdrd)
```

plres2x *Plot Residuals vs. Two Explanatory Variables*

Description

Plot 2 variables, showing a third one with line symbols. Most suitable for showing residuals of a model as this third variable.

Usage

```
plres2x(formula = NULL, reg = NULL, data = reg, restricted = NULL,
        size = 0, slwd = 1, scol = 2, xlab = NULL, ylab = NULL,
        xlim = NULL, ylim = NULL, main = NULL, cex.title = NULL, ...)
```

Arguments

formula	a formula of the form $z \sim x + y$, where x , y are the 2 variables shown by the coordinates of points, and z is shown by line symbols: their orientation corresponds to the sign of z , and their length, to the absolute value.
reg	the result of the model fit
data	the data.frame where the variables are found
restricted	absolute value which truncates the size. Truncation is shown by stars at the end of the line symbols.
size	the symbols are scaled so that size is the length of the largest symbol, as a percentage of the length of the horizontal axis. Defaults to $5/\log_{10}(n)$, where n is the number of observations
slwd	line width of the line symbols
scol	color of the symbols
xlab, ylab	labels for horizontal and vertical axes. Default to the variable names
xlim, ylim	plot ranges for horizontal and vertical axes. They are expanded to accomodate the symbols
main	main title of plot. Defaults to formula
cex.title	character expansion for the main title
...	further arguments, passed to plot

Value

none.

Author(s)

Werner A. Stahel and Andreas Ruckstuhl

Examples

```
data(d.blast)
t.r <- regr(log10(tremor)~location+log10(distance)+log10(charge),
            data=d.blast)
plres2x(~distance+charge, t.r)
```

plresx

Plot Residuals Against Explanatory Variables

Description

Diagnostic plots for fitted regression models: Residuals are plotted against explanatory variables, including smooth for actual residuals and simulated ones, as well as a reference reference line for constant values of the response variable.

Usage

```
plresx(x, data = NULL, resid = NULL, partial.resid = TRUE,
       glm.restype = "deviance", weights = NULL, lab = NULL, cex.lab = 1,
       vars = NULL, sequence = FALSE, se = FALSE,
       addcomp = FALSE, rug = FALSE, jitter = NULL,
       smooth = TRUE, smooth.par = NA, smooth.iter = NA,
       smooth.sim = 19, nxsmooth = 51, reflines = NULL,
       lty = c(1,2,5,3,6,4,1,1), lwd = c(1,1,2,1,1.5,1,1,1),
       colors = getOption("colors.ra"), pch = NULL, col = NULL,
       xlabs = NULL, ylabs = NULL, main = NULL, cex.title = NULL,
       ylim = TRUE, ylimfac = 3, ylimext = 0.1, yaxp = NULL, jitterbinary=TRUE,
       cex = par("cex"), wsymbols = NULL, symbol.size = NULL,
       condprobrange=c(0.05,0.8),
       ask = NULL, multnrows = 0, multncols = 0, ...)
```

Arguments

x	fitted regression object, result of calling link{regr}
data	data set. defaults to the data set used when calling regr
resid	residuals. If not NULL, these will be used instead of residuals(x)
partial.resid	if TRUE (default), residuals for observations will be displayed
glm.restype	type of residuals to be plotted in case of a glm model. In addition to those allowed in residuals() for glm objects, type cond.quant is possible for (un-grouped) binary regression. See ?residuals.polr for an explanation.
weights	weights to be used for smoothing as well as plotting of observations without label
lab	character string labels to be used for displaying observed residuals
cex.lab	character expansion for labels
vars	variables against which the residuals shall be plotted. Defaults to the variables contained in the formula of object. vars can be a formula (of which all variables on the right hand side will be used) or a character vector of variable names. All variables must be contained in data
sequence	if TRUE, the residuals are also plotted against their sequence number
se	if TRUE, standard errors of the fit are plotted
addcomp	if TRUE, the component effect is added to the residuals as is common for a "partial residual plot"
rug	if TRUE, the "rug" is displayed. It consists of tick marks for the observed values.
jitter	amount of jitter to be added for displaying factors
smooth	if TRUE, a smooth line will be fitted to each plot. By default, loess is used. Alternatively, another smoothing function can be given as argument smooth
smooth.par	parameter driving the smooth: argument span if loess is used, which defaults to $3 * \text{nobs}^{\log_{10}(1/2)} * (1 + \text{glm})$, where $\text{glm} = 1$ for glm models
smooth.iter	iterations used in the smoothing algorithm
smooth.sim	number of simulated smooth lines to be drawn
nxsmooth	number of x values for which the smooth is calculated
reflines	logical: Should reference lines be added to the plots of residuals on x? Default is TRUE if available.

lty, lwd, colors	vector of line types and colors to be used for
	[1] observations (only colors[1] matters);
	[2] horizontal reference line;
	[3] smooth;
	[4] simulated smooths;
	[5] reference lines related to component effect;
	[6] for confidence bands of component effects;
	[7] (rnorm) observations in the case of <code>glm.restype="cond.quant"</code> ;
	[8] bars showing conditional quantiles for <code>glm.restype="cond.quant"</code> ;
pch	plotting character to use for unlabeled points
col	color(s) to use for points
xlabs	labels of x variables to be used instead of those given by vars
ylabs	label for residuals
main	main title
cex.title	character expansion for main title
ylim	if TRUE, the range of the y axis will be restricted by calling <code>robrange</code>
ylimfac	argument passed to <code>robrange</code> if selected
ylimext	argument passed to <code>plcoord</code>
yaxp	axis parameters used for residuals, passed to <code>plot</code> , see <code>?par</code> .
jitterbinary	logical: should variables with only 2 distinct values be treated as factors, i.e., jittered?
cex	character expansion
wsymbols	if TRUE, weights are used as symbol size
symbol.size	maximal symbol size, relative to <code>0.02*par("pin")[1]</code>
condprobrange	if conditional probabilities are shown by bars: show only those that are contained in <code>condprobrange</code> (vector of length 2)
ask	as in <code>par</code>
multnrows, multncols	number of rows and columns for multiple frames in case of multivariate residuals
...	further arguments passed to <code>plot</code>

Details

The plots have several unusual features:

- Showing outliers in the plot margins and marking them with their labels (if `markprop>0`).
- A reference line is shown. It follows a "contour" of an approximately constant value of the response variable – more precisely, it shows the "component effect" with opposite sign. It helps decide if a curvature shown by the smooth can be removed by transforming the explanatory variable forming the horizontal axis. The component effect is defined as the curve of fitted values obtained by varying the explanatory variable, keeping all the other variables at their "central value" (the mean of continuous variables and the mode of factors).

Value

The names of the variables against which the residuals are plotted are returned invisibly

Note

This function is still under construction

Author(s)

Werner A. Stahel, ETH Zurich

References

There will be a website

See Also

[plot.regr](#), [termpplot](#)

Examples

```
r.savings <- regr(sr ~ pop15 + pop75 + dpi + ddpi, data = LifeCycleSavings)
plot(r.savings)

data(d.blast)
r.blast <-
  regr(log10(tremor)~location+log10(distance)+log10(charge),
        data=d.blast)
plresx(r.blast, xplot=~.+no)
```

plTA.polr

Plot Residuals against Fitted Values for polr objects

Description

Displays the "Tukey-Anscombe" plot for ordered regression models.

Usage

```
plTA.polr(object, colbars = grey(0.7), colref = grey(0.7), ploty = FALSE)
```

Arguments

object	result of a call to polr
colbars	color to be used for plotting the bars representing the residuals
colref	color for reference line
ploty	if TRUE, the latent response will be plotted instead of the residuals

Details

The plot intends to show the residuals for the latent variable Z that determines the response values Y . Since the precise residuals of the latent variable cannot be known, the plot shows their conditional distribution, given the observed response values. This conditional distribution is characterized by its median (by $-$) and the two other quartiles (by vertical bars).

Value

none.

Author(s)

Werner A. Stahel, ETH Zurich

References

See <http://stat.ethz.ch/~stahel/regression>

See Also

[plot.regr](#), [residuals.polr](#)

Examples

```
house.plr <- polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
plTA.polr(house.plr)
```

polrregr

Fit a polr Regression and Return a Corresponding regr object

Description

Fit a regression for an ordered target variable by polr and return a corresponding regr object.

Usage

```
polrregr(formula, data, weights, start, ...,
subset, na.action, contrasts = NULL, Hess = FALSE, model = TRUE,
method = c("logistic", "probit", "cloglog", "cauchit"))
```

Arguments

formula, data, weights, start, subset, na.action, contrasts, Hess, model, method, ...
see ?polr of package MASS

Value

see ?polr of package MASS and ?regr

Author(s)

Werner A. Stahel, ETH Zurich

See Also[polr](#), [regr](#)

`predict.regr`*Prediction Methods for 'regr' Objects*

Description

Calculates predicted values for regr objects. The possible results depend on the class of the fitted model.

Usage

```
## S3 method for class 'regr'
predict(object, newdata = NULL, se.fit = FALSE,
        scale = object$sigma,
        interval = c("none", "confidence", "prediction"), level = 0.95,
        type = NULL, terms = NULL, na.action = na.pass, ...)

## S3 method for class 'polr'
predict(object, newdata=NULL,
        type = c("class", "probs", "link"), ...)
```

Arguments

<code>object</code>	Object of class <code>regr</code>
<code>newdata</code>	An optional data frame in which to look for variables with which to predict. If omitted, the fitted values are used.
<code>se.fit</code>	if TRUE, standard errors will be calculated if possible
<code>scale</code>	Scale parameter for std.err. calculation
<code>interval</code>	Type of interval requested
<code>level</code>	Confidence level
<code>type</code>	Type of prediction: response or model term For glm or ordered regression, type <code>link</code> gives estimated values of the linear predictor.
<code>terms</code>	If <code>type="terms"</code> , which terms (default is all terms)
<code>na.action</code>	function determining what should be done with missing values in <code>newdata</code> . The default is to insert NA.
<code>...</code>	further arguments passed to specific methods

Details

`regr` is a "super class" which includes many specific model classes such as `lm`, `glm`, `polr`, ...
. `predict.regr` is a wrapper function that calls the specific methods corresponding to the specific model class.

Value

vector of predictions, or matrix with columns `fit`, `lwr`, and `upr` if `interval` is set.

If `se.fit` is `TRUE`, a list with the following components is returned:

<code>fit</code>	vector or matrix as above
<code>se.fit</code>	standard error of predicted means
<code>residual.scale</code>	residual standard deviations
<code>df</code>	degrees of freedom for residual~Describe the value returned

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[predict.lm](#), [residuals.polr](#)

Examples

```
data(d.blast)
r.blast <- regr(log10(tremor)~location+log10(distance)+log10(charge),
               data=d.blast)
t.pr <- predict(r.blast)
showd(t.pr)

data(d.fossiles)
r.mregr <-
  regr(cbind(sAngle,lLength,rWidth)~SST.Mean+Salinity+lChlorophyll+region+N,
       data=d.fossiles)
t.pr <- predict(r.mregr)
showd(t.pr)
```

`print.regr`

Print Method for 'regr' objects

Description

Prints 'regr' objects.

Usage

```
print.regr(x, correlation = FALSE, dummycoef = NULL,
           digits = max(3, getOption("digits") - 2), symbolic.cor = p > 4,
           signif.stars = getOption("show.signif.stars"),
           residuals = FALSE, niterations = FALSE, ...)
```

Arguments

x	'regr' object
correlation	logical: should correlation among estimated coefficients be shown?
dummycoef	logical: should the dummy coefficients for the factors be shown?
digits	number of significant digits to be shown
symbolic.cor	logical: should correlations be shown by symbols?
signif.stars	logical: should significance asterisks be shown?
residuals	logical: should summary(residuals) be shown?
niterations	???
...	further arguments, passed to the print methods used.

Details

The `tit` attribute of `x` will be printed if available and `getOption("doc") > 0`, and any `doc` attribute, if `getOption("doc") >= 2` (see [tit](#)).

Value

Print functions return the object to be printed invisibly.

Author(s)

Werner A. Stahel, ETH Zurich

Examples

```
data(d.blast)
r.blast <-
  regr(log10(tremor)~location+log10(distance)+log10(charge),
       data=d.blast)
print(r.blast, correlation=TRUE, symbolic.cor=TRUE, dummycoef=FALSE)
```

regr

Fitting Regression Models

Description

'regr' fits regression models of various types: ordinary, robust, generalized linear, multinomial, ordered response, multivariate, ...

It is a wrapper function which calls the respective S fitting functions and yields an extensive result, to be used by the respective print and plot methods.

Usage

```
regr(formula, data, tit = NULL, family = NULL,
     calcdisp = NULL, suffmean = 3, nonlinear = FALSE, start=NULL,
     robust = FALSE, method=NULL, init.reg = "f.ltsreg",
     subset = NULL, weights = NULL, na.action = nainf.exclude,
     model = TRUE, x = FALSE, termtable = TRUE, vif = TRUE, ...)
```

Arguments

formula	a symbolic description of the model to be fit.
data	data frame containing the variables in the model.
tit	title (becomes tit attribute of result)
family	character string describing the type of model and the fitting procedure. By default, regr will fit a model that is suitable for the response variable, if possible.\ Can be either "gaussian" for ordinary regression, including multivariate response, "binomial" for logistic regression, "poisson" for Poisson regression, "multinomial" for multinomial regression, "polr" for ordered response (cumulative logits) regression, or a suitable argument for glm, i.e.: A description of the error distribution and link function to be used in the model. This can be a character string naming a family function, a family function or the result of a call to a family function. (See 'family' for details of family functions.)
calcdisp	a logical value indicating whether, for family=binomial and family=poisson, dispersion should be calculated (TRUE) or set to 1 (FALSE)
suffmean	!!!
nonlinear	if TRUE, a non-linear model is expected, to be fitted by nlm
start	if nonlin is TRUE, starting values for nlm
robust	if TRUE, robust fitting is used if a function is available.
method	if suitable, a method argument that is passed to the fitting procedure. ...
init.reg	function used to initialize robust methods
subset	an optional vector specifying a subset of observations to be used in the fitting process. Names of variables in data can be used to generate the vector.
weights	an optional vector of weights to be used for fitting.
na.action	a function which indicates what should happen when the data contain 'NA's. 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	logical values indicating whether the model frame or the model.matrix, respectively, should be included as a component of the returned value.
termtable	if FALSE, the regr type term table will not be generated
vif	a logical value indicating whether the collinearity measure R2.j should be calculated
...	other argument to be passed to the fitting function

Value

regr returns a list object of class regr and secondary class as produced by the fitting function. The components include generally those of the results of the fitting function and its summary. Additional components are described here in a draft version.

stres	Standardized residuals
sigma	estimated standard deviation for normal errors,\ sqrt(dispersion) for glm
testcoef	Table for testing terms both for single and multiple degree terms (continuous or binary explanatory variables and factors)
allcoef	All coefficients. Estimated effects of factors are given for all their levels.

h	leverage values
vartype	types of the input variables (c for continuous, b for binary, f for factor, o for ordered factor)
binfac	levels of binary factors
fitfun	R function that has been called for fitting the model

Author(s)

Werner A. Stahel, Seminar for Statistics, ETH Zurich

See Also

[lm](#), [glm](#), [rlm](#), [multinom](#), [polr](#),...

Examples

```
data(d.blast)
( r.blast <-
  regr(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast) )
## Anorexia
data(anorexia, package="MASS")
r.anorexia <- regr(Postwt ~ Prewt + Treat + offset(Prewt),
  data = anorexia)
## Annette Dobson (1990) "An Introduction to Generalized Linear Models".
## Page 9: Plant Weight Data.
ctl <- c(4.17,5.58,5.18,6.11,4.50,4.61,5.17,4.53,5.33,5.14)
trt <- c(4.81,4.17,4.41,3.59,5.87,3.83,6.03,4.89,4.32,4.69)
d.dob <- data.frame(group = gl(2,10,20, labels=c("Ctl","Trt")),
  weight = c(ctl, trt))
(r.dob <- regr(weight ~ group, data=d.dob))

## multinomial regression
data(d.surveyenvir)
t.r <- regr(disturbance~age+education+location, data=d.surveyenvir)

## ordered regression
t.r <- regr(Sat ~ Infl + Type + Cont, weights = housing$Freq,
  data = housing)
plot(t.r)

## multivariate regression
data(d.fossiles)
r.mregr <-
  regr(cbind(sAngle,lLength,rWidth)~SST.Mean+Salinity+lChlorophyll+region+N,
  data=d.fossiles)
```

residuals.polr

Residuals of a Binary or Ordered Regression

Description

Calculates quartiles and random numbers according to the conditional distribution of residuals for the latent variable of a binary or ordinal regression, given the observed response value. See Details for an explanation.

Usage

```
## S3 method for class 'polr'
residuals(object, na.action = object, ...)
```

Arguments

object	the result of polr or of glm(, family=binomial) with binary data.
na.action	treatment of NAs
...	unused

Details

For binary and ordinal regression, the regression models can be described by introducing a latent response variable Z of which the observed response Y is a classified version, and for which a linear regression applies. The errors of this "latent regression" have a logistic distribution. Given the linearly predicted value $\eta[i]$, which is the fitted value for the latent variable, the residual for $Z[i]$ can therefore be assumed to have a logistic distribution.

This function calculates quantiles and random numbers according to the conditional distribution of residuals for $Z[i]$, given the observed $y[i]$.

Value

	a data.frame with the variables
median	medians of the conditional distributions
lowq	lower quartiles
uppq	upper quartiles
random	random numbers, drawn according to the conditional distributions
fit	linear predictor values
y	observed response values

Author(s)

Werner A. Stahel, ETH Zurich

References

See <http://stat.ethz.ch/~stahel/regression>

See Also

[condquant](#), [plot.regr](#)

Examples

```
house.plr <- polr(Sat ~ Infl + Type + Cont, weights = Freq, data = housing)
house.resid <- residuals(house.plr)
```

residuals.survreg	<i>Residuals of a regression for survival data</i>
-------------------	--

Description

Calculates residuals for uncensored observations and quartiles and random numbers according to the conditional distribution of residuals for censored observations. See Details for an explanation.

Usage

```
## S3 method for class 'survreg'  
residuals(object, type = "response", ...)
```

Arguments

object	an object inheriting from classes Survreg or coxph
type	type of residuals required
...	unused

Details

For censored observations, the conditional distribution of the survival time, given the censoring, is used to calculate quantiles ...

Value

matrix with class condquant

Note

BUG: residuals.coxph is not yet implemented in this way.

Author(s)

Werner A. Stahel, ETH Zurich

References

See <http://stat.ethz.ch/~stahel/regression>

See Also

[condquant](#), [plot.regr](#)

robrange	<i>Robust Range of Data</i>
----------	-----------------------------

Description

Determines a robust range of the data on the basis of the trimmed mean and variance

Usage

```
robrange(data, trim = 0.2, fac = 3)
```

Arguments

data	a vector of data. Missing values are dropped
trim	trimming proportion
fac	factor used for expanding the range, see Details

Details

The function determines the trimmed mean m and then the "upper trimmed mean" s of absolute deviations from m , multiplied by fac . The robust minimum is then defined as $m - fac * s$ or $\min(data)$, whichever is larger, and similarly for the maximum.

Value

The robust range.

Author(s)

Werner Stahel, ETH Zurich

See Also

[plcoord](#)

Examples

```
x <- c(rnorm(20), rnorm(3, 5, 20))
robrange(x)
```

`showd`*Show a Part of a Data.frame*

Description

Shows a part of the `data.frame` which allows for grasping the nature of the data. The function is typically used to make sure that the data is what was desired and to grasp the nature of the variables in the phase of getting acquainted with the data.

Usage

```
showd(data, first = 3, nrow. = 4, ncol. = NULL)
```

Arguments

<code>data</code>	a <code>data.frame</code> (or a matrix)
<code>first</code>	the first <code>first</code> rows will be shown and ...
<code>nrow.</code>	a selection of <code>nrow.</code> rows will be shown in addition. They will be selected with equal row number differences. The last row is always included.
<code>ncol.</code>	number of columns (variables) to be shown. The first and last columns will also be included. If <code>ncol.</code> has more than one element, it is used to identify the columns directly.

Details

The `tit` attribute of `data` will be printed if available and `getOption("doc") > 0`, and any `doc` attribute, if `getOption("doc") >= 2` (see `tit`).

Value

returns invisibly the character vector containing the formatted data

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[head](#) and [tail](#).

Examples

```
showd(iris)
data(d.rehab)
options(doc=1)
showd(d.rehab, ncol=7)
```

simresiduals	<i>Simulate Residuals</i>
--------------	---------------------------

Description

Simulates residuals for a given normal regression model

Usage

```
simresiduals(object, nrep, resgen=NULL, glm.restype="deviance")
```

Arguments

object	result of fitting a regression
nrep	number of replicates
resgen	if a function, it is used to generate random residuals, which will be multiplied by <code>object\$sigma</code> . If TRUE, <code>rnorm</code> will be used. If NULL (default) the standardized residuals of <code>object</code> will be randomly permuted
glm.restype	type of residuals to be generated (for glm)

Details

The simulated residuals are obtained by replacing the response variable by permuted standardized residuals of the fitted regression, multiplied by the scale `object$sigma`. If standardized residuals are not available, ordinary residuals are used.

Value

A list consisting of

simres	a matrix of which each column contains an set of simulated residuals
simstres	corresponding standardized residuals

Author(s)

Werner A. Stahel, ETH Zurich

Examples

"will follow"

stamp

Add an Identification Line to a Plot

Description

A line is added to the current plot in the lower right corner that contains project information and date.

Usage

```
stamp(sure = TRUE, outer.margin = NULL,  
      project = getOption("project"), step=getOption("step"),  
      stamp=getOption("stamp"), ...)
```

Arguments

sure	if FALSE, the stamp will only be added if <code>getOption("stamp")>0</code>
outer.margin	if TRUE, the stamp is put to the outer margin of the plot. This is the default if the plot is currently split into panels.
project, step	character string describing the project and the step of analysis.
stamp	controls default action, see details
...	arguments passed to <code>mtext</code>

Details

The function is used to document plots produced during a data analysis. It is called by all plotting functions of this package. For getting final presentation versions of the plots, the stamp can be suppressed by changing the default by calling `options(stamp=0)`.

In more detail: If `stamp==0` (or `getOption("stamp")==0`) the function will only do its thing if `sure==TRUE`.

If `stamp==2`, it will certainly do it.

If `stamp==1` and `sure==FALSE`, the stamp is added when a plot page is complete.

Value

invisibly returns the string that is added to the plot

Author(s)

Werner A. Stahel, ETH Zurich

Examples

```
options(project="Example A", step="regression analysis")  
plot(1:10)  
stamp()
```

sumna

Count NAs

Description

Count the missing or non-finite values for each column of a matrix or data.frame

Usage

```
sumna(object, inf = TRUE)
```

Arguments

`object` a vector, matrix, or data.frame
`inf` if TRUE, Inf and NaN values are counted along with NAs

Value

numerical vector containing the missing value counts for each column

Note

This is a simple shortcut for `apply(is.na(object), 2, sum)` or `apply(!is.finite(object), 2, sum)`

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[is.na](#), [is.finite](#)

Examples

```
t.d <- data.frame(V1=c(1,2,NA,4), V2=c(11,12,13,Inf), V3=c(21,NA,23,Inf))
sumna(t.d)
```

terms2order

Second Order Terms of a Formula

Description

Create all square and interaction terms of a given formula

Usage

```
terms2order(object, squared = TRUE)
```

Arguments

object	An object for which <code>formula(object)</code> returns a formula. Usually a regression fit
squared	logical: Should square terms be added to the interactions?

Details

The 'squares' are the squares of the terms of `dataClass` 'numeric' appearing in the formula (cf. `attr(object$terms, "dataClass")`), except for terms that are already squares (contain `^`) or interactions (contain `:`).

Value

'term.labels' of the generated terms

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[add1.regr](#)

Examples

```
data(d.blast)
t.r <- regr(log10(tremor)~location+log10(distance)+log10(charge), data=d.blast)
terms2order(t.r)
```

Tobit

Prepare a Response for a Tobit Model

Description

Returns a `Surv` object that allows for setting up a Tobit regression model by calling `survreg`

Usage

```
Tobit(data, limit = 0, log = FALSE)
```

Arguments

data	the variable to be used as the response in the Tobit regression
limit	Lower limit which censors the observations. If <code>log</code> is <code>TRUE</code> , then the default is the minimal value of <code>logst(data)</code> , and if <code>limit>0</code> , it refers to the untransformed data.
log	logical. If <code>TRUE</code> , data will be log transformed by calling <code>logst</code> .

Details

Tobit regression is a special case of regression with left censored response data. The function `survreg` is suitable for fitting. In `regr`, this is done automatically.

Value

A Surv object.

Author(s)

Werner A. Stahel

See Also

Surv

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==> Define data, use random,
##--or do help(data=index) for the standard data sets.

## The function is currently defined as
function(data, limit=NULL, log=FALSE)
{
  ## Purpose:  create a Surv object for tobit regression
  ## -----
  ## Arguments:
  ## -----
  ## Author: Werner Stahel, Date:  1 Jan 2010, 21:49
  if (is.null(limit)) limit <- min(data)
  if (log) {
    if (limit<=0) {
      data <- logst(data)
      limit <- min(data)
    } else {
      data <- log10(data)
      limit <- log10(limit)
    }
  }
  data <- pmax(data,limit)
  if (sum(data==limit)<=1)
    warning("Tobit: <= 1 observation equal to 'limit'")
  rr <- Surv(data, data!=limit, type='left')
  attr(rr,'distribution') <- 'gaussian'
  class(rr) <- c(class(rr), "matrix")
  rr
}
```

warn

List Warnings

Description

Gives a List of Warnings

Usage

warn()

Details

This function simplifies the output of `warnings` if there are several identical warnings, by counting their occurrence

Value

the table of warnings

Author(s)

Werner A. Stahel, ETH Zurich

See Also

[warnings](#)

Examples

```
## not to be run as an example()
## for (i in 3:6)
##   m <- matrix(1:7, 3,i)
##   warn()
```

xdistResdiff

Residual Differences for Near Replicates: Tabulate and Test

Description

A test for the completeness of a linear regression model can be performed based on comparing the differences of residuals for pairs of observations that are close to each other to the estimated standard deviation of the

Usage

```
xdistResdiff(object, perc = c(3, 10, 80), trim = 0.1,
             nmax = 100, nsim = 100, out = "aggregate")
xdistResdiff(x, perc = c(3, 10, 90), trim = 1/6)
```

Arguments

<code>object</code>	an object containing the result of fitting a linear model by <code>regr</code>
<code>perc</code>	Percentage points to define distance classes
<code>trim</code>	Trimming proportion for calculating means of absolute residual differences
<code>nmax</code>	maximal number of observations to form pairs
<code>nsim</code>	number of simulations for determining the null distribution of test statistics
<code>out</code>	determines the value of <code>xdistResdiff</code> : if <code>=="aggregate"</code> (the default), the value will be produced by calling <code>xdistResdiff</code> , otherwise, all <code>x</code> distances and respective residual differences will be returned.

Details

See package vignette.

Value

For `xdistResdiff` with `out="aggregate"` and `xdistResScale`, a matrix is returned with a row for each class of `x` distances and the columns

<code>xdist</code>	mean <code>x</code> distance
<code>rdiff.mean</code>	absolute differences of residuals for pairs of observations in the distance class, averaged over the class
<code>rdiff.simmean</code>	mean of (trimmed) means for simulated data
<code>rdiff.se</code>	standard error of (trimmed) means as obtained from simulation

The matrix carries along the following attributes:

<code>perc</code>	given argument <code>perc</code>
<code>xd.classlim</code>	the actual class limits corresponding to <code>perc</code>
<code>trim</code>	given argument <code>trim</code>
<code>rdiff.grandmean</code>	overall mean of absolute residual differences
<code>p-values</code>	<code>p</code> values for the classes as obtained from simulation, and <code>p</code> -value for the sum of squares statistic
<code>class</code>	The value has S3 class <code>xdistResScale</code> and <code>matrix</code> . If <code>xdistResdiff</code> with <code>out</code> different from "aggregate", then a <code>data.frame</code> is returned containing a row for each pair of observations and the columns <code>\itemid1</code> , <code>id2</code> the labels of the two observations <code>\itemxdist</code> the <code>x</code> distance between the two observations <code>\itemresdiff</code> the difference of residuals for the two observations The value has S3 class <code>xdistResdiff</code> and <code>data.frame</code> .

See package vignette.

Werner A. Stahel, ETH Zurich

##

```
data(d.blast) rr <- regr(tremor~distance+charge, data=d.blast) ## an inadequate model! xdistResdiff(rr)
xdrd <- xdistResdiff(rr, out="all") showd(xdrd[,1:10]) xdistResScale(xdrd) ## same as first call of xdiffResdiff
```

regression test

Index

- *Topic **NA**
 - nainf.exclude, 21
 - sumna, 46
- *Topic **\textasciitildekwd1**
 - plot.xdistResScale, 29
 - Tobit, 47
- *Topic **\textasciitildekwd2**
 - plot.xdistResScale, 29
 - Tobit, 47
- *Topic **aplot**
 - legendr, 16
 - plcoord, 22
- *Topic **attribute**
 - doc, 9
- *Topic **datasets**
 - d.blast, 6
 - d.fossiles, 6
 - d.rehab, 7
 - d.surveyenvir, 8
- *Topic **debugging**
 - getmeth, 15
- *Topic **distribution**
 - condquant, 5
- *Topic **hplot**
 - plfitpairs, 23
 - plmatrix, 23
 - plot.regr, 25
 - plres2x, 29
 - plresx, 30
 - plTA.polr, 33
- *Topic **manip**
 - asinperc, 4
 - dropdata, 12
 - last, 15
 - logst, 17
 - nainf.exclude, 21
 - plcoord, 22
- *Topic **miscellaneous**
 - c.months, 4
- *Topic **misc**
 - stamp, 45
 - terms2order, 46
- *Topic **package**
 - regr-package, 2
- *Topic **print**
 - print.regr, 36
 - showd, 43
- *Topic **regression**
 - drop1.regr, 11
 - fitcomp, 13
 - fitted.polr, 14
 - leverage, 17
 - modelTable, 19
 - plot.regr, 25
 - plres2x, 29
 - plresx, 30
 - plTA.polr, 33
 - polrregr, 34
 - predict.regr, 35
 - regr, 37
 - residuals.polr, 39
 - residuals.survreg, 41
 - simresiduals, 44
- *Topic **univar**
 - robrange, 42
- *Topic **utilities**
 - getmeth, 15
 - mframe, 18
 - showd, 43
 - stamp, 45
 - warn, 48
- [.modelTable (modelTable), 19
- add1, 12
- add1.mlm (drop1.mlm), 10
- add1.regr, 47
- add1.regr (drop1.regr), 11
- asinperc, 4
- c.mon (c.months), 4
- c.months, 4
- c.weekdays (c.months), 4
- compareTerms (modelTable), 19
- condquant, 5, 40, 41
- d.blast, 6
- d.fossiles, 6

- d.rehab, 7
- d.surveyenvir, 8
- doc, 9, 12
- doc<- (doc), 9
- drop1, 11, 12
- drop1.lm, 10
- drop1.mlml, 10
- drop1.multinom (drop1.regr), 11
- drop1.regr, 10, 11
- dropdata, 12
- fitcomp, 13
- fitted.polr, 14
- format.modelTable (modelTable), 19
- format.pval, 20
- getmeth, 15
- getOption, 45
- getS3method, 15
- glm, 39
- head, 43
- influence, 17
- is.finite, 46
- is.na, 46
- last, 15
- legend, 16
- legendr, 16
- leverage, 17
- list, 19
- lm, 39
- loess, 31
- logst, 17, 47
- mframe, 18
- modelTable, 19
- mult.fig, 19
- multinom, 39
- na.exclude, 21
- na.omit, 21
- nainf.exclude, 21
- nainf.omit (nainf.exclude), 21
- pairs, 25
- par, 19, 24, 32
- plcoord, 22, 42
- plfitpairs, 23
- plmatrix, 23
- plot.regr, 25, 33, 34, 40, 41
- plot.xdistResScale, 29
- plres2x, 29
- plresx, 30
- plTA.polr, 33
- points, 24
- polr, 35, 39
- polrregr, 34
- predict, 14
- predict.lm, 36
- predict.nlm (predict.regr), 35
- predict.polr (predict.regr), 35
- predict.regr, 35
- print.modelTable (modelTable), 19
- print.regr, 36
- regr, 11, 23, 35, 37
- regr-package, 2
- residuals.coxph (residuals.survreg), 41
- residuals.polr, 27, 31, 34, 36, 39
- residuals.survreg, 41
- r1m, 39
- robrange, 22, 32, 42
- showd, 43
- simresiduals, 44
- stamp, 3, 45
- sumna, 46
- symnum, 20
- tail, 43
- termplot, 33
- terms2order, 11, 46
- text, 24
- tit, 12, 26, 37, 43
- tit (doc), 9
- tit<- (doc), 9
- Tobit, 47
- warn, 48
- warnings, 49
- xdistResdiff, 49
- xdistResScale (xdistResdiff), 49