

# Package ‘car’

July 4, 2017

**Version** 2.1-5

**Date** 2017-06-25

**Title** Companion to Applied Regression

**Depends** R (>= 3.2.0)

**Imports** MASS, mgcv, nnet, pbkrtest (>= 0.4-4), quantreg, grDevices,  
utils, stats, graphics

**Suggests** alr4, boot, coxme, leaps, lme4, lmtest, Matrix, MatrixModels,  
nlme, rgl (>= 0.93.960), sandwich, SparseM, survival, survey,  
nloptr

**ByteCompile** yes

**LazyLoad** yes

**LazyData** yes

**Description** Functions and Datasets to Accompany J. Fox and S. Weisberg,  
An R Companion to Applied Regression, Second Edition, Sage, 2011.

**License** GPL (>= 2)

**URL** <https://r-forge.r-project.org/projects/car/>,  
<https://CRAN.R-project.org/package=car>,  
<http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/index.html>

**Author** John Fox [aut, cre],  
Sanford Weisberg [aut],  
Daniel Adler [ctb],  
Douglas Bates [ctb],  
Gabriel Baud-Bovy [ctb],  
Steve Ellison [ctb],  
David Firth [ctb],  
Michael Friendly [ctb],  
Gregor Gorjanc [ctb],  
Spencer Graves [ctb],  
Richard Heiberger [ctb],  
Rafael Laboissiere [ctb],  
Georges Monette [ctb],

Duncan Murdoch [ctb],  
 Henric Nilsson [ctb],  
 Derek Ogle [ctb],  
 Brian Ripley [ctb],  
 William Venables [ctb],  
 David Winsemius [ctb],  
 Achim Zeileis [ctb],  
 R-Core [ctb]

**Maintainer** John Fox <jfox@mcmaster.ca>

**Repository** CRAN

**Repository/R-Forge/Project** car

**Repository/R-Forge/Revision** 481

**Repository/R-Forge/DateTimeStamp** 2017-07-03 14:10:10

**Date/Publication** 2017-07-04 08:00:39 UTC

**NeedsCompilation** no

## R topics documented:

car-package	4
Adler	5
AMSSurvey	6
Angell	7
Anova	7
Anscombe	14
avPlots	15
Baumann	17
bcPower	18
Bfox	20
Blackmore	21
Boot	22
boxCox	24
boxCoxVariable	27
Boxplot	28
boxTidwell	29
Burt	31
CanPop	32
car-deprecated	32
car-internal.Rd	34
carWeb	34
ceresPlots	35
Chile	37
Chiot	38
compareCoefs	39
Contrasts	40
Cowles	42
crPlots	43

Davis	45
DavisThin	46
deltaMethod	47
densityPlot	50
Depredations	53
dfbetaPlots	54
Duncan	55
durbinWatsonTest	56
Ellipses	57
Ericksen	61
Florida	62
Freedman	63
Friendly	63
Ginzberg	64
Greene	65
Guyer	66
Hartnagel	66
hccm	67
Highway1	69
hist.boot	70
infIndexPlot	72
influencePlot	74
invResPlot	75
invTranPlot	76
KosteckiDillon	79
Leinhardt	80
levneTest	81
leveragePlots	82
linearHypothesis	84
LoBD	90
logit	92
Mandel	93
mcPlots	94
Migration	96
mmps	97
Moore	101
Mroz	102
ncvTest	103
OBrienKaiser	104
Ornstein	105
outlierTest	106
panel.car	107
plot.powerTransform	108
Pottery	110
powerTransform	110
Prestige	114
qqPlot	115
Quartet	118

recode . . . . .	118
regLine . . . . .	120
residualPlots . . . . .	121
Robey . . . . .	125
Sahlins . . . . .	125
Salaries . . . . .	126
scatter3d . . . . .	127
scatterplot . . . . .	131
scatterplotMatrix . . . . .	134
ScatterplotSmoothers . . . . .	137
showLabels . . . . .	139
sigmaHat . . . . .	141
SLID . . . . .	142
Soils . . . . .	143
some . . . . .	144
spreadLevelPlot . . . . .	145
States . . . . .	148
subsets . . . . .	149
symbol . . . . .	150
testTransform . . . . .	151
Transact . . . . .	153
TransformationAxes . . . . .	154
UN . . . . .	156
USPop . . . . .	157
vif . . . . .	157
Vocab . . . . .	159
wcrossprod . . . . .	159
WeightLoss . . . . .	160
which.names . . . . .	161
Womenlf . . . . .	162
Wong . . . . .	163
Wool . . . . .	164

<b>Index</b>	<b>165</b>
--------------	------------

---

car-package	<i>Companion to Applied Regression</i>
-------------	--

---

### Description

Functions and Datasets to Accompany J. Fox and S. Weisberg, *An R Companion to Applied Regression*, Second Edition, Sage, 2011.

**Details**

Package: car  
Version: 2.1-5  
Date: 2017-06-25  
Depends: R (>= 3.2.0)  
LinkingTo: Rcpp  
Imports: MASS, mgcv, nnet, pbkrtest (>= 0.4-4), quantreg, grDevices, utils, stats, graphics  
Suggests: alr4, boot, coxme, leaps, lme4, lmtest, Matrix, MatrixModels, nlme, rgl (>= 0.93.960), sandwich, SparseM, surv  
License: GPL (>= 2)  
URL: <https://CRAN.R-project.org/package=car>, <http://socserv.socsci.mcmaster.ca/jfox/Books/Compar>

**Author(s)**

John Fox <jfox@mcmaster.ca> and Sanford Weisberg. We are grateful to Douglas Bates, Gabriel Baud-Bovy, David Firth, Michael Friendly, Gregor Gorjanc, Spencer Graves, Richard Heiberger, Rafael Laboissiere, Georges Monette, Henric Nilsson, Derek Ogle, Brian Ripley, Achim Zeileis, and R Core for various suggestions and contributions.

Maintainer: John Fox <jfox@mcmaster.ca>

---

Adler

*Experimenter Expectations*

---

**Description**

The Adler data frame has 97 rows and 3 columns.

The “experimenters” were the actual subjects of the study. They collected ratings of the apparent successfulness of people in pictures who were pre-selected for their average appearance. The experimenters were told prior to collecting data that the pictures were either high or low in their appearance of success, and were instructed to get good data, scientific data, or were given no such instruction. Each experimenter collected ratings from 18 randomly assigned respondents; a few subjects were deleted at random to produce an unbalanced design.

**Usage**

Adler

**Format**

This data frame contains the following columns:

**instruction** a factor with levels: GOOD, good data; NONE, no stress; SCIENTIFIC, scientific data.

**expectation** a factor with levels: HIGH, expect high ratings; LOW, expect low ratings.

**rating** The average rating obtained.

**Source**

Adler, N. E. (1973) Impact of prior sets given experimenters and subjects on the experimenter expectancy effect. *Sociometry* **36**, 113–126.

**References**

Erickson, B. H., and Nosanchuk, T. A. (1977) *Understanding Data*. McGraw-Hill Ryerson.

---

AMSSurvey

*American Math Society Survey Data*

---

**Description**

Counts of new PhDs in the mathematical sciences for 2008-09 and 2011-12 categorized by type of institution, gender, and US citizenship status.

**Usage**

AMSSurvey

**Format**

A data frame with 24 observations on the following 5 variables.

**type** a factor with levels I(Pu) for group I public universities, I(Pr) for group I private universities, II and III for groups II and III, IV for statistics and biostatistics programs, and Va for applied mathematics programs.

**sex** a factor with levels Female, Male of the recipient

**citizen** a factor with levels Non-US, US giving citizenship status

**count** The number of individuals of each type in 2008-09

**count11** The number of individuals of each type in 2011-12

**Details**

These data are produced yearly by the American Math Society.

**Source**

<http://www.ams.org/employment/surveyreports.html> Supplementary Table 4 in the 2008-09 data. See <http://www.ams.org/profession/data/annual-survey/docsgtrtd> for more recent data.

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Phipps, Polly, Maxwell, James W. and Rose, Colleen (2009), *2009 Annual Survey of the Mathematical Sciences*, 57, 250–259, Supplementary Table 4, <http://www.ams.org/employment/2009Survey-First-Report-Supp.pdf>

---

 Angell

*Moral Integration of American Cities*


---

**Description**

The Angell data frame has 43 rows and 4 columns. The observations are 43 U. S. cities around 1950.

**Usage**

Angell

**Format**

This data frame contains the following columns:

**moral** Moral Integration: Composite of crime rate and welfare expenditures.

**hetero** Ethnic Heterogeneity: From percentages of nonwhite and foreign-born white residents.

**mobility** Geographic Mobility: From percentages of residents moving into and out of the city.

**region** A factor with levels: E Northeast; MW Midwest; S Southeast; W West.

**Source**

Angell, R. C. (1951) The moral integration of American Cities. *American Journal of Sociology* **57** (part 2), 1–140.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

 Anova

*Anova Tables for Various Statistical Models*


---

**Description**

Calculates type-II or type-III analysis-of-variance tables for model objects produced by `lm`, `glm`, `multinom` (in the **nnet** package), `polr` (in the **MASS** package), `coxph` (in the **survival** package), `coxme` (in the **coxme** package), `svyglm` (in the **survey** package), `r1m` (in the **MASS** package), `lmer` in the **lme4** package, `lme` in the **nlme** package, and (by the default method) for most models with a linear predictor and asymptotically normal coefficients (see details below). For linear models, F-tests are calculated; for generalized linear models, likelihood-ratio chi-square, Wald chi-square, or F-tests are calculated; for multinomial logit and proportional-odds logit models, likelihood-ratio tests are calculated. Various test statistics are provided for multivariate linear models produced by `lm` or `manova`. Partial-likelihood-ratio tests or Wald tests are provided for Cox models. Wald chi-square tests are provided for fixed effects in linear and generalized linear mixed-effects models. Wald chi-square or F tests are provided in the default case.

**Usage**

```

Anova(mod, ...)

Manova(mod, ...)

## S3 method for class 'lm'
Anova(mod, error, type=c("II","III", 2, 3),
white.adjust=c(FALSE, TRUE, "hc3", "hc0", "hc1", "hc2", "hc4"),
vcov=NULL, singular.ok, ...)

## S3 method for class 'aov'
Anova(mod, ...)

## S3 method for class 'glm'
Anova(mod, type=c("II","III", 2, 3),
      test.statistic=c("LR", "Wald", "F"),
      error, error.estimate=c("pearson", "dispersion", "deviance"),
      singular.ok, ...)

## S3 method for class 'multinom'
Anova(mod, type = c("II","III", 2, 3), ...)

## S3 method for class 'polr'
Anova(mod, type = c("II","III", 2, 3), ...)

## S3 method for class 'mlm'
Anova(mod, type=c("II","III", 2, 3), SSPE, error.df,
      idata, idesign, icontrasts=c("contr.sum", "contr.poly"), imatrix,
      test.statistic=c("Pillai", "Wilks", "Hotelling-Lawley", "Roy"),...)

## S3 method for class 'manova'
Anova(mod, ...)

## S3 method for class 'mlm'
Manova(mod, ...)

## S3 method for class 'Anova.mlm'
print(x, ...)

## S3 method for class 'Anova.mlm'
summary(object, test.statistic, univariate=object$repeated,
      multivariate=TRUE, p.adjust.method, ...)

## S3 method for class 'summary.Anova.mlm'
print(x, digits = getOption("digits"),
      SSP=TRUE, SSPE=SSP, ... )

## S3 method for class 'univaov'

```



```

print(x, digits = max(getOption("digits") - 2L, 3L),
      style=c("wide", "long"),
      by=c("response", "term"),
      ...)

## S3 method for class 'univaov'
as.data.frame(x, row.names, optional, by=c("response", "term"), ...)

## S3 method for class 'coxph'
Anova(mod, type=c("II", "III", 2, 3),
      test.statistic=c("LR", "Wald"), ...)

## S3 method for class 'coxme'
Anova(mod, type=c("II", "III", 2, 3),
      test.statistic=c("Wald", "LR"), ...)

## S3 method for class 'lme'
Anova(mod, type=c("II", "III", 2, 3),
      vcov.=vcov(mod), singular.ok, ...)

## S3 method for class 'mer'
Anova(mod, type=c("II", "III", 2, 3),
      test.statistic=c("Chisq", "F"), vcov.=vcov(mod), singular.ok, ...)

## S3 method for class 'merMod'
Anova(mod, type=c("II", "III", 2, 3),
      test.statistic=c("Chisq", "F"), vcov.=vcov(mod), singular.ok, ...)

## S3 method for class 'svyglm'
Anova(mod, ...)

## S3 method for class 'r1m'
Anova(mod, ...)

## Default S3 method:
Anova(mod, type=c("II", "III", 2, 3),
      test.statistic=c("Chisq", "F"), vcov.=vcov(mod),
      singular.ok, ...)

```

### Arguments

mod	lm, aov, glm, multinom, polr, mlm, coxph, coxme, lme, mer, merMod, svyglm, r1m, or other suitable model object.
error	for a linear model, an lm model object from which the error sum of squares and degrees of freedom are to be calculated. For F-tests for a generalized linear model, a glm object from which the dispersion is to be estimated. If not specified, mod is used.
type	type of test, "II", "III", 2, or 3.

<code>singular.ok</code>	defaults to TRUE for type-II tests, and FALSE for type-III tests (where the tests for models with aliased coefficients will not be straightforwardly interpretable); if FALSE, a model with aliased coefficients produces an error.
<code>test.statistic</code>	for a generalized linear model, whether to calculate "LR" (likelihood-ratio), "Wald", or "F" tests; for a Cox or Cox mixed-effects model, whether to calculate "LR" (partial-likelihood ratio) or "Wald" tests; in the default case or for linear mixed models fit by <code>lmer</code> , whether to calculate Wald "Chisq" or "F" tests. For a multivariate linear model, the multivariate test statistic to compute — one of "Pillai", "Wilks", "Hotelling-Lawley", or "Roy", with "Pillai" as the default. The summary method for <code>Anova.mlm</code> objects permits the specification of more than one multivariate test statistic, and the default is to report all four.
<code>error.estimate</code>	for F-tests for a generalized linear model, base the dispersion estimate on the Pearson residuals ("pearson", the default); use the dispersion estimate in the model object ("dispersion"); or base the dispersion estimate on the residual deviance ("deviance"). For binomial or Poisson GLMs, where the dispersion is fixed to 1, setting <code>error.estimate="dispersion"</code> is changed to "pearson", with a warning.
<code>white.adjust</code>	if not FALSE, the default, tests use a heteroscedasticity-corrected coefficient covariance matrix; the various values of the argument specify different corrections. See the documentation for <code>hccm</code> for details. If <code>white.adjust=TRUE</code> then the "hc3" correction is selected.
SSPE	For Anova for a multivariate linear model, the error sum-of-squares-and-products matrix; if missing, will be computed from the residuals of the model; for the print method for the summary of an Anova of a multivariate linear model, whether or not to print the error SSP matrix (defaults to TRUE).
SSP	if TRUE (the default), print the sum-of-squares and cross-products matrix for the hypothesis and the response-transformation matrix.
<code>error.df</code>	The degrees of freedom for error; if missing, will be taken from the model.
<code>idata</code>	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See <i>Details</i> for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
<code>idesign</code>	a one-sided model formula using the "data" in <code>idata</code> and specifying the intra-subject design.
<code>icontrasts</code>	names of contrast-generating functions to be applied by default to factors and ordered factors, respectively, in the within-subject "data"; the contrasts must produce an intra-subject model matrix in which different terms are orthogonal. The default is <code>c("contr.sum", "contr.poly")</code> .
<code>imatrix</code>	as an alternative to specifying <code>idata</code> , <code>idesign</code> , and (optionally) <code>icontrasts</code> , the model matrix for the within-subject design can be given directly in the form of list of named elements. Each element gives the columns of the within-subject model matrix for a term to be tested, and must have as many rows as there are responses; the columns of the within-subject model matrix for different terms must be mutually orthogonal.
<code>x, object</code>	object of class " <code>Anova.mlm</code> " to print or summarize.

<code>multivariate, univariate</code>	compute and print multivariate and univariate tests for a repeated-measures ANOVA or multivariate linear model; the default is TRUE for both for repeated measures and TRUE for multivariate for a multivariate linear model.
<code>p.adjust.method</code>	if given for a multivariate linear model when univariate tests are requested, the univariate tests are corrected for simultaneous inference by term; if specified, should be one of the methods recognized by <code>p.adjust</code> or TRUE, in which case the default (Holm) adjustment is used.
<code>digits</code>	minimum number of significant digits to print.
<code>style</code>	for printing univariate tests if requested for a multivariate linear model; one of "wide", the default, or "long".
<code>by</code>	if univariate tests are printed in "long" style, they can be ordered by "response", the default, or by "term".
<code>row.names, optional</code>	not used.
<code>vcov.</code>	in the default method, an optional coefficient-covariance matrix or function to compute a covariance matrix, computed by default by applying the generic <code>vcov</code> function to the model object. A similar argument may be supplied to the <code>lm</code> method, and the default (NULL) is to ignore the argument; if both <code>vcov.</code> and <code>white.adjust</code> are supplied to the <code>lm</code> method, the latter is used.
<code>...</code>	do not use.

## Details

The designations "type-II" and "type-III" are borrowed from SAS, but the definitions used here do not correspond precisely to those employed by SAS. Type-II tests are calculated according to the principle of marginality, testing each term after all others, except ignoring the term's higher-order relatives; so-called type-III tests violate marginality, testing each term in the model after all of the others. This definition of Type-II tests corresponds to the tests produced by SAS for analysis-of-variance models, where all of the predictors are factors, but not more generally (i.e., when there are quantitative predictors). Be very careful in formulating the model for type-III tests, or the hypotheses tested will not make sense.

As implemented here, type-II Wald tests are a generalization of the linear hypotheses used to generate these tests in linear models.

For tests for linear models, multivariate linear models, and Wald tests for generalized linear models, Cox models, mixed-effects models, generalized linear models fit to survey data, and in the default case, `Anova` finds the test statistics without refitting the model. The `svyglm` method simply calls the default method and therefore can take the same arguments.

The standard R `anova` function calculates sequential ("type-I") tests. These rarely test interesting hypotheses in unbalanced designs.

A MANOVA for a multivariate linear model (i.e., an object of class "mlm" or "manova") can optionally include an intra-subject repeated-measures design. If the intra-subject design is absent (the default), the multivariate tests concern all of the response variables. To specify a repeated-measures design, a data frame is provided defining the repeated-measures factor or factors via `idata`, with default contrasts given by the `icontrasts` argument. An intra-subject model-matrix is generated

from the formula specified by the `idesign` argument; columns of the model matrix corresponding to different terms in the intra-subject model must be orthogonal (as is insured by the default contrasts). Note that the contrasts given in `icontrasts` can be overridden by assigning specific contrasts to the factors in `idata`. As an alternative, the within-subjects model matrix can be specified directly via the `imatrix` argument. `Manova` is essentially a synonym for `Anova` for multivariate linear models.

If univariate tests are requested for the summary of a multivariate linear model, the object returned contains a `univaov` component of "univaov"; `print` and `as.data.frame` methods are provided for the "univaov" class.

For the default method to work, the model object must contain a standard `terms` element, and must respond to the `vcov`, `coef`, and `model.matrix` functions. If any of these requirements is missing, then it may be possible to supply it reasonably simply (e.g., by writing a missing `vcov` method for the class of the model object).

### Value

An object of class "anova", or "Anova.mlm", which usually is printed. For objects of class "Anova.mlm", there is also a `summary` method, which provides much more detail than the `print` method about the MANOVA, including traditional mixed-model univariate F-tests with Greenhouse-Geisser and Huynh-Feldt corrections.

### Warning

Be careful of type-III tests.

### Author(s)

John Fox <jfox@mcmaster.ca>; the code for the Mauchly test and Greenhouse-Geisser and Huynh-Feldt corrections for non-sphericity in repeated-measures ANOVA are adapted from the functions `stats::stats::mauchly.test.SSD` and `stats::sphericity` by R Core; `summary.Anova.mlm` and `print.summary.Anova.mlm` incorporates code contributed by Gabriel Baud-Bovy.

### References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Hand, D. J., and Taylor, C. C. (1987) *Multivariate Analysis of Variance and Repeated Measures: A Practical Approach for Behavioural Scientists*. Chapman and Hall.
- O'Brien, R. G., and Kaiser, M. K. (1985) MANOVA method for analyzing repeated measures designs: An extensive primer. *Psychological Bulletin* **97**, 316–333.

### See Also

[linearHypothesis](#), [anova.anova.lm](#), [anova.glm](#), [anova.mlm](#), [anova.coxph](#), [link\[survey\]{svyglm}](#).

**Examples**

```

## Two-Way Anova

mod <- lm(conformity ~ fcategory*partner.status, data=Moore,
  contrasts=list(fcategory=contr.sum, partner.status=contr.sum))
Anova(mod)

## One-Way MANOVA
## See ?Pottery for a description of the data set used in this example.

summary(Anova(lm(cbind(Al, Fe, Mg, Ca, Na) ~ Site, data=Pottery)))

## MANOVA for a randomized block design (example courtesy of Michael Friendly:
## See ?Soils for description of the data set)

soils.mod <- lm(cbind(pH,N,Dens,P,Ca,Mg,K,Na,Conduc) ~ Block + Contour*Depth,
  data=Soils)
Manova(soils.mod)
summary(Anova(soils.mod), univariate=TRUE, multivariate=FALSE,
  p.adjust.method=TRUE)

## a multivariate linear model for repeated-measures data
## See ?OBrienKaiser for a description of the data set used in this example.

phase <- factor(rep(c("pretest", "posttest", "followup"), c(5, 5, 5)),
  levels=c("pretest", "posttest", "followup"))
hour <- ordered(rep(1:5, 3))
idata <- data.frame(phase, hour)
idata

mod.ok <- lm(cbind(pre.1, pre.2, pre.3, pre.4, pre.5,
  post.1, post.2, post.3, post.4, post.5,
  fup.1, fup.2, fup.3, fup.4, fup.5) ~ treatment*gender,
  data=OBrienKaiser)
(av.ok <- Anova(mod.ok, idata=idata, idesign=~phase*hour))

summary(av.ok, multivariate=FALSE)

## A "doubly multivariate" design with two distinct repeated-measures variables
## (example courtesy of Michael Friendly)
## See ?WeightLoss for a description of the dataset.

imatrix <- matrix(c(
  1,0,-1, 1, 0, 0,
  1,0, 0,-2, 0, 0,
  1,0, 1, 1, 0, 0,
  0,1, 0, 0,-1, 1,
  0,1, 0, 0, 0,-2,
  0,1, 0, 0, 1, 1), 6, 6, byrow=TRUE)
colnames(imatrix) <- c("WL", "SE", "WL.L", "WL.Q", "SE.L", "SE.Q")
rownames(imatrix) <- colnames(WeightLoss)[-1]

```

```
(imatrix <- list(measure=imatrix[,1:2], month=imatrix[,3:6]))
contrasts(WeightLoss$group) <- matrix(c(-2,1,1, 0,-1,1), ncol=2)
(wl.mod<-lm(cbind(wl1, wl2, wl3, se1, se2, se3)~group, data=WeightLoss))
Anova(wl.mod, imatrix=imatrix, test="Roy")

## mixed-effects models examples:

## Not run:
library(nlme)
example(lme)
Anova(fm2)

## End(Not run)

## Not run:
library(lme4)
example(glmer)
Anova(gm1)

## End(Not run)
```

---

 Anscombe

*U. S. State Public-School Expenditures*


---

### Description

The Anscombe data frame has 51 rows and 4 columns. The observations are the U. S. states plus Washington, D. C. in 1970.

### Usage

Anscombe

### Format

This data frame contains the following columns:

**education** Per-capita education expenditures, dollars.

**income** Per-capita income, dollars.

**young** Proportion under 18, per 1000.

**urban** Proportion urban, per 1000.

### Source

Anscombe, F. J. (1981) *Computing in Statistical Science Through APL*. Springer-Verlag.

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

**Description**

These functions construct added-variable (also called partial-regression) plots for linear and generalized linear models.

**Usage**

```
avPlots(model, terms=~., intercept=FALSE, layout=NULL, ask, main, ...)
```

```
avp(...)
```

```
avPlot(model, ...)
```

```
## S3 method for class 'lm'
avPlot(model, variable,
id.method = list(abs(residuals(model, type="pearson")), "x"),
labels,
id.n = if(id.method[1]=="identify") Inf else 0,
id.cex=1, id.col=palette()[1], id.location="lr",
col = palette()[1], col.lines = palette()[2],
xlab, ylab, pch = 1, lwd = 2,
main=paste("Added-Variable Plot:", variable),
grid=TRUE,
ellipse=FALSE, ellipse.args=NULL,
  marginal.scale=FALSE, ...)
```

```
## S3 method for class 'glm'
avPlot(model, variable,
id.method = list(abs(residuals(model, type="pearson")), "x"),
labels,
id.n = if(id.method[1]=="identify") Inf else 0,
id.cex=1, id.col=palette()[1], id.location="lr",
col = palette()[1], col.lines = palette()[2],
xlab, ylab, pch = 1, lwd = 2, type=c("Wang", "Weisberg"),
main=paste("Added-Variable Plot:", variable), grid=TRUE,
ellipse=FALSE, ellipse.args=NULL, ...)
```

**Arguments**

model	model object produced by lm or glm.
terms	A one-sided formula that specifies a subset of the predictors. One added-variable plot is drawn for each term. For example, the specification <code>terms = ~.-X3</code> would plot against all terms except for X3. If this argument is a quoted name of one of the terms, the added-variable plot is drawn for that term only.

<code>intercept</code>	Include the intercept in the plots; default is FALSE.
<code>variable</code>	A quoted string giving the name of a regressor in the model matrix for the horizontal axis
<code>layout</code>	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
<code>main</code>	The title of the plot; if missing, one will be supplied.
<code>ask</code>	If TRUE, ask the user before drawing the next plot; if FALSE don't ask.
<code>...</code>	<code>avPlots</code> passes these arguments to <code>avPlot</code> . <code>avPlot</code> passes them to <code>plot</code> .
<code>id.method, labels, id.n, id.cex, id.col, id.location</code>	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
<code>col</code>	color for points; the default is the <i>second</i> entry in the current color palette (see <a href="#">palette</a> and <a href="#">par</a> ).
<code>col.lines</code>	color for the fitted line.
<code>pch</code>	plotting character for points; default is 1 (a circle, see <a href="#">par</a> ).
<code>lwd</code>	line width; default is 2 (see <a href="#">par</a> ).
<code>xlab</code>	x-axis label. If omitted a label will be constructed.
<code>ylab</code>	y-axis label. If omitted a label will be constructed.
<code>type</code>	if "Wang" use the method of Wang (1985); if "Weisberg" use the method in the Arc software associated with Cook and Weisberg (1999).
<code>grid</code>	If TRUE, the default, a light-gray background grid is put on the graph.
<code>ellipse</code>	If TRUE, plot a concentration ellipse; default is FALSE.
<code>ellipse.args</code>	Arguments to pass to the <code>link{dataEllipse}</code> function, in the form of a list with named elements; e.g., <code>ellipse.args=list(robust=TRUE)</code> will cause the ellipse to be plotted using a robust covariance-matrix.
<code>marginal.scale</code>	Consider an added-variable plot of Y versus X given Z. If this argument is FALSE then the limits on the horizontal axis are determined by the range of the residuals from the regression of X on Z and the limits on the vertical axis are determined by the range of the residuals from the regression of Y on Z. If the argument is TRUE, then the limits on the horizontal axis are determined by the range of X minus its mean, and on the vertical axis by the range of Y minus its means; adjustment is made if necessary to include outliers. This scaling allows visualization of the correlations between Y and Z and between X and Z. For example, if the X and Z are highly correlated, then the points will be concentrated on the middle of the plot.

## Details

The function intended for direct use is `avPlots` (for which `avp` is an abbreviation).



**Value**

These functions are used for their side effect of producing plots, but also invisibly return the coordinates of the plotted points.

**Author(s)**

John Fox <jfox@mcmaster.ca>, Sanford Weisberg <sandy@umn.edu>

**References**

- Cook, R. D. and Weisberg, S. (1999) *Applied Regression, Including Computing and Graphics*. Wiley.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Wang, P. C. (1985) Adding a variable in generalized linear models. *Technometrics* **27**, 273–276.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

**See Also**

[residualPlots](#), [crPlots](#), [ceresPlots](#), [link{dataEllipse}](#)

**Examples**

```
avPlots(lm(prestige~income+education+type, data=Duncan))

avPlots(glm(partic != "not.work" ~ hincome + children,
  data=Women1f, family=binomial))

m1 <- lm(partic ~ tfr + menwage + womwage + debt + parttime, Bfox)
par(mfrow=c(1,3))
plot(partic ~ womwage, Bfox) # marginal plot, ignoring other predictors
abline(lm(partic ~ womwage, Bfox), col="red", lwd=2)
grid()
avPlots(m1, ~ womwage) # av Plot, adjusting for others
avPlots(m1, ~ womwage, marginal.scale=TRUE) # av Plot, adjusting and scaling as in marginal plot
```

**Description**

The Baumann data frame has 66 rows and 6 columns. The data are from an experimental study conducted by Baumann and Jones, as reported by Moore and McCabe (1993) Students were randomly assigned to one of three experimental groups.

**Usage**

Baumann

**Format**

This data frame contains the following columns:

**group** Experimental group; a factor with levels: Basal, traditional method of teaching; DRTA, an innovative method; Strat, another innovative method.

**pretest.1** First pretest.

**pretest.2** Second pretest.

**post.test.1** First post-test.

**post.test.2** Second post-test.

**post.test.3** Third post-test.

**Source**

Moore, D. S. and McCabe, G. P. (1993) *Introduction to the Practice of Statistics, Second Edition*. Freeman, p. 794–795.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

bcPower

*Box-Cox, Box-Cox with Negatives Allowed, Yeo-Johnson and Basic Power Transformations*

---

**Description**

Transform the elements of a vector or columns of a matrix using, the Box-Cox, Box-Cox with negatives allowed, Yeo-Johnson, or simple power transformations.

**Usage**

```
bcPower(U, lambda, jacobian.adjusted=FALSE, gamma=NULL)
```

```
bcnPower(U, lambda, jacobian.adjusted = FALSE, gamma)
```

```
yjPower(U, lambda, jacobian.adjusted = FALSE)
```

```
basicPower(U,lambda, gamma=NULL)
```

**Arguments**

U	A vector, matrix or data.frame of values to be transformed
lambda	Power transformation parameter with one element for each column of U, usually in the range from -2 to 2, or if U
jacobian.adjusted	If TRUE, the transformation is normalized to have Jacobian equal to one. The default FALSE is almost always appropriate
gamma	For bcPower or basicPower, the transformation is of $U + \text{gamma}$ , where gamma is a positive number called a start that must be large enough so that $U + \text{gamma}$ is strictly positive. For the bcnPower, Box-cox power with negatives allowed, see the details below.

**Details**

The Box-Cox family of *scaled power transformations* equals  $((U + \gamma)^\lambda - 1)/\lambda$  for  $\lambda \neq 0$ , and  $\log(U)$  if  $\lambda = 0$ . If  $\gamma$  is not specified, it is set equal to zero.  $U + \text{gamma}$  must be strictly positive to use this family.

The Box-Cox family with negatives allowed was proposed by Hawkins and Weisberg (2017). It is the Box-Cox power transformation of  $z = .5 * (y + (y^2 + \gamma^2)^{1/2})$ , where  $\gamma$  is strictly positive if  $y$  includes negative values and non-negative otherwise. The value of  $z$  is always positive. The bcnPower transformations behave very similarly to the bcPower transformations, including much less bias than is introduced by setting the parameter  $\gamma$  to be non-zero in the Box-Cox family.

If family="yeo.johnson" then the Yeo-Johnson transformations are used. This is the Box-Cox transformation of  $U + 1$  for nonnegative values, and of  $|U| + 1$  with parameter  $2 - \lambda$  for  $U$  negative.

The basic power transformation returns  $U^\lambda$  if  $\lambda$  is not zero, and  $\log(\lambda)$  otherwise for  $U$  strictly positive.

If jacobian.adjusted is TRUE, then the scaled transformations are divided by the Jacobian, which is a function of the geometric mean of  $U$  for skewPower and yjPower and of  $U + \text{gamma}$  for bcPower. With this adjustment, the Jacobian of the transformation is always equal to 1.

Missing values are permitted, and return NA where ever U is equal to NA.

**Value**

Returns a vector or matrix of transformed values.

**Author(s)**

Sanford Weisberg, <sandy@umn.edu>

**References**

- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Hawkins, D. and Weisberg, S. (2017) Combining the Box-Cox Power and Generalized Log Transformations to Accomodate Negative Responses In Linear and Mixed-Effects Linear Models, submitted for publication.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley Wiley, Chapter 7.

Yeo, In-Kwon and Johnson, Richard (2000) A new family of power transformations to improve normality or symmetry. *Biometrika*, 87, 954-959.

### See Also

[powerTransform](#), [testTransform](#)

### Examples

```
U <- c(NA, (-3:3))
## Not run: bcPower(U, 0) # produces an error as U has negative values
bcPower(U, 0, gamma=4)
bcPower(U, .5, jacobian.adjusted=TRUE, gamma=4)
basicPower(U, lambda = 0, gamma=4)
yjPower(U, 0)
V <- matrix(1:10, ncol=2)
bcPower(V, c(0, 2))
basicPower(V, c(0,1))
```

---

Bfox

*Canadian Women's Labour-Force Participation*

---

### Description

The Bfox data frame has 30 rows and 7 columns. Time-series data on Canadian women's labor-force participation, 1946–1975.

### Usage

Bfox

### Format

This data frame contains the following columns:

**partic** Percent of adult women in the workforce.

**tfr** Total fertility rate: expected births to a cohort of 1000 women at current age-specific fertility rates.

**menwage** Men's average weekly wages, in constant 1935 dollars and adjusted for current tax rates.

**womwage** Women's average weekly wages.

**debt** Per-capita consumer debt, in constant dollars.

**parttime** Percent of the active workforce working 34 hours per week or less.

### Warning

The value of tfr for 1973 is misrecorded as 2931; it should be 1931.

**Source**

Fox, B. (1980) *Women's Domestic Labour and their Involvement in Wage Work*. Unpublished doctoral dissertation, p. 449.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

Blackmore	<i>Exercise Histories of Eating-Disordered and Control Subjects</i>
-----------	---

---

**Description**

The Blackmore data frame has 945 rows and 4 columns. Blackmore and Davis's data on exercise histories of 138 teenaged girls hospitalized for eating disorders and 98 control subjects.

**Usage**

Blackmore

**Format**

This data frame contains the following columns:

**subject** a factor with subject id codes.

**age** age in years.

**exercise** hours per week of exercise.

**group** a factor with levels: control, Control subjects; patient, Eating-disordered patients.

**Source**

Personal communication from Elizabeth Blackmore and Caroline Davis, York University.

**Description**

This function provides a simple front-end to the `boot` function in the package also called `boot`. Whereas `boot` is very general and therefore has many arguments, the `Boot` function has very few arguments, but should meet the needs of many users.

**Usage**

```
Boot(object, f=coef, labels=names(f(object)), R=999, method=c("case", "residual"), ...)

## Default S3 method:
Boot(object, f=coef, labels=names(f(object)),
      R=999, method=c("case", "residual"), start = FALSE, ...)

## S3 method for class 'lm'
Boot(object, f=coef, labels=names(f(object)),
      R=999, method=c("case", "residual"), ...)

## S3 method for class 'glm'
Boot(object, f=coef, labels=names(f(object)),
      R=999, method=c("case", "residual"), ...)

## S3 method for class 'nls'
Boot(object, f=coef, labels=names(f(object)),
      R=999, method=c("case", "residual"), ...)
```

**Arguments**

<code>object</code>	A regression object of class <code>lm</code> , <code>glm</code> or <code>nls</code> . The function may work with other regression objects that support the update method and have a subset argument
<code>f</code>	A function whose one argument is the name of a regression object that will be applied to the updated regression object to compute the statistics of interest. The default is <code>coef</code> , to return to regression coefficient estimates. For example, <code>f = function(obj) coef(obj)[1]/coef(obj)[2]</code> will bootstrap the ratio of the first and second coefficient estimates.
<code>labels</code>	Provides labels for the statistics computed by <code>f</code> . If this argument is of the wrong length, then generic labels will be generated.
<code>R</code>	Number of bootstrap samples. The number of bootstrap samples actually computed may be smaller than this value if either the fitting method is iterative, or if the rank of a fitted <code>lm</code> or <code>glm</code> model is different in the bootstrap replication than in the original data.

method	The bootstrap method, either “case” for resampling cases or “residuals” for a residual bootstrap. See the details below. The residual bootstrap is available only for <code>lm</code> and <code>nls</code> objects and will return an error for <code>glm</code> objects.
...	Arguments passed to methods. The default method passes these on to the <code>boot</code> function.
start	Should the <code>f</code> parameters be passed as <code>start</code> values to the update in each bootstrap iteration? Alternatively, <code>start</code> can also be a numeric vector that is passed to the update.

### Details

Whereas the `boot` function is very general, `Boot` is very specific. It takes the information from a regression object and the choice of method, and creates a function that is passed as the `statistic` argument to `boot`. The argument `R` is also passed to `boot`. All other arguments to `boot` are kept at their default values.

The methods available for `lm` and `nls` objects are “case” and “residual”. The case bootstrap resamples from the joint distribution of the terms in the model and the response. The residual bootstrap fixes the fitted values from the original data, and creates bootstraps by adding a bootstrap sample of the residuals to the fitted values to get a bootstrap response. It is an implementation of Algorithm 6.3, page 271, of Davison and Hinkley (1997). For `nls` objects ordinary residuals are used in the resampling rather than the standardized residuals used in the `lm` method. The residual bootstrap for generalized linear models has several competing approaches, but none are without problems. If you want to do a residual bootstrap for a `glm`, you will need to write your own call to `boot`.

For the default object to work with other types of regression model, the model must have methods for the following generic functions: `residuals(object, type="pearson")` must return Pearson residuals; `fitted(object)` must return fitted values; `hatvalues(object)` should return the leverages, or perhaps the value 1.

An attempt to model fit to a bootstrap sample may fail. In a `lm` or `glm` fit, the bootstrap sample could have a different rank from the original fit. In an `nls` fit, convergence may not be obtained for some bootstraps. In either case, `NA` are returned for the value of the function `f`. The summary methods handle the `NA`s appropriately.

### Value

See `boot` for the returned value from this function. The `car` package includes additional generic functions `summary`, `vcov.boot`, `confint.boot` and `hist.boot` that work with `boot` objects.

### Author(s)

Sanford Weisberg, <sandy@umn.edu>.

### References

- Davison, A, and Hinkley, D. (1997) *Bootstrap Methods and their Applications*. Oxford: Oxford University Press.
- Fox, J. and Weisberg, S. (2011) *Companion to Applied Regression*, Second Edition. Thousand Oaks: Sage.

Fox, J. and Weisberg, S. (2012) *Bootstrapping*, <http://socserv.mcmaster.ca/jfox/Books/Companion/appendix/Appendix-Bootstrapping.pdf>.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley Wiley, Chapters 4 and 11.

### See Also

Functions that work with Boot objects from the boot package are `boot.array`, `boot.ci`, `plot.boot` and `empinf`. Additional functions in the car package are `summary.boot`, `confint.boot`, and `hist.boot`.

### Examples

```
m1 <- lm(Fertility ~ ., swiss)
betahat.boot <- Boot(m1, R=199) # 199 bootstrap samples--too small to be useful
summary(betahat.boot) # default summary
confint(betahat.boot)
hist(betahat.boot)
# Bootstrap for the estimated residual standard deviation:
sigmahat.boot <- Boot(m1, R=199, f=sigmaHat, labels="sigmaHat")
summary(sigmahat.boot)
confint(sigmahat.boot)
```

---

boxCox

*Graph the profile log-likelihood for Box-Cox transformations in 1D or in 2D with the bcPower family.*

---

### Description

Computes and optionally plots profile log-likelihoods for the parameter of the Box-Cox power family, the Yeo-Johnson power family, or for either of the parameters in a skew power family. This is a slight generalization of the `boxcox` function in the **MASS** package that allows for families of transformations other than the Box-Cox power family.

### Usage

```
boxCox(object, ...)
```

## Default S3 method:

```
boxCox(object,
        lambda = seq(-2, 2, 1/10), plotit = TRUE,
        interp = plotit, eps = 1/50,
        xlab=NULL, ylab=NULL,
        family="bcPower",
        param=c("lambda", "gamma"), gamma=NULL,
        grid=TRUE, ...)
```

## S3 method for class 'formula'

```
boxCox(object, lambda = seq(-2, 2, 1/10), plotit = TRUE, family = "bcPower",
```



```

    param = c("lambda", "gamma"), gamma = NULL, grid = TRUE,
    ...)

## S3 method for class 'lm'
boxCox(object, lambda = seq(-2, 2, 1/10), plotit = TRUE, ...)

boxCox2d(x, ksds = 4, levels = c(0.5, 0.95, 0.99, 0.999),
         main = "bcnPower Log-likelihood", grid=TRUE, ...)

```

## Arguments

object	a formula or fitted model object of class <code>lm</code> or <code>aov</code> .
lambda	vector of values of $\lambda$ , with default <code>(-2, 2)</code> in steps of 0.1, where the profile log-likelihood will be evaluated.
plotit	logical which controls whether the result should be plotted; default <code>TRUE</code> .
interp	logical which controls whether spline interpolation is used. Default to <code>TRUE</code> if plotting with <code>lambda</code> of length less than 100.
eps	Tolerance for <code>lambda = 0</code> ; defaults to 0.02.
xlab	defaults to <code>"lambda"</code> or <code>"gamma"</code> .
ylab	defaults to <code>"log-Likelihood"</code> or for <code>bcnPower</code> family to the appropriate label.
family	Defaults to <code>"bcPower"</code> for the Box-Cox power family of transformations. If set to <code>"yjPower"</code> the Yeo-Johnson family, which permits negative responses, is used. If set to <code>bcnPower</code> the function gives the profile log-likelihood for the parameter selected via <code>param</code> .
param	Relevant only to <code>family="bcnPower"</code> , produces a profile log-likelihood for the parameter selected, maximizing over the remaining parameter.
gamma	For use when the <code>family="bcnPower"</code> , <code>param="gamma"</code> . If this is a vector of positive values, then the profile log-likelihood for the location (or start) parameter in the <code>bcnPower</code> family is evaluated at these values of <code>gamma</code> . If <code>gamma</code> is <code>NULL</code> , then evaluation is done at 100 equally spaced points between <code>min(.01, gmax - 3*sd)</code> and <code>gmax + 3*sd</code> , where <code>gmax</code> is the maximum likelihood estimate of <code>gamma</code> , and <code>sd</code> is the sd of the response. See <a href="#">bcnPower</a> for the definition of <code>gamma</code> .
grid	If <code>TRUE</code> , the default, a light-gray background grid is put on the graph.
...	additional arguments passed to the <code>lm</code> method with <code>boxCox</code> . <code>formula</code> or passed to <code>contour</code> with <code>boxCox2d</code> .
x	An object created by a call to <a href="#">powerTransform</a> using <code>family="bcnPower"</code> .
ksds	Contour plotting of the log-likelihood surface will cover plus of minus <code>ksds</code> standard deviations on each axis.
levels	Contours will be drawn at the values of <code>levels</code> . For example, <code>levels=c(.5, .99)</code> would display two contours, at the 50% level and at the 99% level.
main	Title for the contour plot

## Details

The `boxCox` function is an elaboration of the `boxcox` function in the **MASS** package. The first 7 arguments are the same as in `boxcox`, and if the argument `family="bcPower"` is used, the result is essentially identical to the function in **MASS**. Two additional families are the `yjPower` and `bcnPower` families that allow a few values of the response to be non-positive. The `bcnPower` family has two parameters: a power  $\lambda$  and a start or location parameter  $\gamma$ , and the `boxCox` function can be used to obtain a profile log-likelihood for either parameter with  $\lambda$  as the default. Alternatively, the `boxCox2d` function can be used to get a contour plot of the profile log-likelihood.

## Value

Both functions are designed for their side effects of drawing a graph. The `boxCox` function returns a list of the lambda (or possibly, gamma) vector and the computed profile log-likelihood vector, invisibly if the result is plotted. If `plot.it=TRUE` plots log-likelihood vs lambda and indicates a 95 lambda. If `interp=TRUE`, spline interpolation is used to give a smoother plot.

## Author(s)

Sanford Weisberg, <sandy@umn.edu>

## References

- Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. *Journal of the Royal Statistical Society, Series B*. 26 211-46.
- Cook, R. D. and Weisberg, S. (1999) *Applied Regression Including Computing and Graphics*. Wiley.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Hawkins, D. and Weisberg, S. (2015) Combining the Box-Cox Power and Generalized Log Transformations to Accommodate Negative Responses, submitted for publication.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.
- Yeo, I. and Johnson, R. (2000) A new family of power transformations to improve normality or symmetry. *Biometrika*, 87, 954-959.

## See Also

[boxcox](#), [yjPower](#), [bcPower](#), [bcnPower](#), [powerTransform](#), [contour](#)

## Examples

```
with(trees, boxCox(Volume ~ log(Height) + log(Girth), data = trees,
  lambda = seq(-0.25, 0.25, length = 10)))

data("quine", package = "MASS")
with(quine, boxCox(Days ~ Eth*Sex*Age*Lrn, data = quine,
  lambda = seq(-0.05, 0.45, len = 20), family="yjPower"))
```

---

boxCoxVariable	<i>Constructed Variable for Box-Cox Transformation</i>
----------------	--

---

**Description**

Computes a constructed variable for the Box-Cox transformation of the response variable in a linear model.

**Usage**

```
boxCoxVariable(y)
```

**Arguments**

`y` response variable.

**Details**

The constructed variable is defined as  $y[\log(y/\tilde{y}) - 1]$ , where  $\tilde{y}$  is the geometric mean of  $y$ .

The constructed variable is meant to be added to the right-hand-side of the linear model. The t-test for the coefficient of the constructed variable is an approximate score test for whether a transformation is required.

If  $b$  is the coefficient of the constructed variable, then an estimate of the normalizing power transformation based on the score statistic is  $1 - b$ . An added-variable plot for the constructed variable shows leverage and influence on the decision to transform  $y$ .

**Value**

a numeric vector of the same length as  $y$ .

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

- Atkinson, A. C. (1985) *Plots, Transformations, and Regression*. Oxford.
- Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. *JRSS B* **26** 211–246.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[boxcox](#), [powerTransform](#), [bcPower](#)

**Examples**

```
mod <- lm(interlocks + 1 ~ assets, data=Ornstein)
mod.aux <- update(mod, . ~ . + boxCoxVariable(interlocks + 1))
summary(mod.aux)
# avPlots(mod.aux, "boxCoxVariable(interlocks + 1)")
```

---

 Boxplot

*Boxplots With Point Identification*


---

**Description**

Boxplot is a wrapper for the standard R `boxplot` function, providing point identification, axis labels, and a formula interface for boxplots without a grouping variable.

**Usage**

```
Boxplot(y, ...)

## Default S3 method:
Boxplot(y, g, labels, id.method = c("y", "identify", "none"),
        id.n=10, xlab, ylab, ...)

## S3 method for class 'formula'
Boxplot(formula, data = NULL, subset, na.action = NULL, labels.,
        id.method = c("y", "identify", "none"), xlab, ylab, ...)

## S3 method for class 'list'
Boxplot(y, xlab="", ylab="", ...)

## S3 method for class 'data.frame'
Boxplot(y, labels=rownames(y), ...)

## S3 method for class 'matrix'
Boxplot(y, ...)
```

**Arguments**

**y** a numeric variable for which the boxplot is to be constructed; a list of numeric variables, each element of which will be treated as a group; a numeric data frame or a numeric matrix, each of whose columns will be treated as a group.

**g** a grouping variable, usually a factor, for constructing parallel boxplots.

**labels, labels.** point labels; if not specified, Boxplot will use the row names of the data argument, if one is given, or observation numbers, or row names if y is a data frame or matrix (that has row names).

id.method	if "y" (the default), all outlying points are labeled; if "identify", points may be labeled interactive; if "none", no point identification is performed.
id.n	up to id.n high outliers and low outliers will be identified in each group, (default, 10).
xlab, ylab	text labels for the horizontal and vertical axes; if missing, Boxplot will use the variable names, or, in the case of a list, data frame, or matrix, empty labels.
formula	a 'model' formula, of the form $\sim y$ to produce a boxplot for the variable $y$ , or of the form $y \sim g$ , $y \sim g1*g2*...$ , or $y \sim g1 + g2 + ...$ to produce parallel boxplots for $y$ within levels of the grouping variable(s) $g$ , etc., usually factors.
data, subset, na.action	as for statistical modeling functions (see, e.g., <a href="#">lm</a> ).
...	further arguments, such as <code>at</code> , to be passed to <a href="#">boxplot</a> .

**Author(s)**

John Fox <jfox@mcmaster.ca>, with a contribution from Steve Ellison to handle `at` argument (see [boxplot](#)).

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[boxplot](#)

**Examples**

```
Boxplot(~income, data=Prestige, id.n=Inf) # identify all outliers
Boxplot(income ~ type, data=Prestige)
Boxplot(income ~ type, data=Prestige, at=c(1, 3, 2))
Boxplot(k5 + k618 ~ lfp*wc, data=Mroz)
with(Prestige, Boxplot(income, labels=rownames(Prestige)))
with(Prestige, Boxplot(income, type, labels=rownames(Prestige)))
Boxplot(scale(Prestige[, 1:4]))
```

---

boxTidwell

*Box-Tidwell Transformations*

---

**Description**

Computes the Box-Tidwell power transformations of the predictors in a linear model.

**Usage**

```

boxTidwell(y, ...)

## S3 method for class 'formula'
boxTidwell(formula, other.x=NULL, data=NULL, subset,
  na.action=getOption("na.action"), verbose=FALSE, tol=0.001,
  max.iter=25, ...)

## Default S3 method:
boxTidwell(y, x1, x2=NULL, max.iter=25, tol=0.001,
  verbose=FALSE, ...)

## S3 method for class 'boxTidwell'
print(x, digits, ...)

```

**Arguments**

formula	two-sided formula, the right-hand-side of which gives the predictors to be transformed.
other.x	one-sided formula giving the predictors that are <i>not</i> candidates for transformation, including (e.g.) factors.
data	an optional data frame containing the variables in the model. By default the variables are taken from the environment from which boxTidwell is called.
subset	an optional vector specifying a subset of observations to be used.
na.action	a function that indicates what should happen when the data contain NAs. The default is set by the na.action setting of options.
verbose	if TRUE a record of iterations is printed; default is FALSE.
tol	if the maximum relative change in coefficients is less than tol then convergence is declared.
max.iter	maximum number of iterations.
y	response variable.
x1	matrix of predictors to transform.
x2	matrix of predictors that are <i>not</i> candidates for transformation.
...	not for the user.
x	boxTidwell object.
digits	number of digits for rounding.

**Details**

The maximum-likelihood estimates of the transformation parameters are computed by Box and Tidwell's (1962) method, which is usually more efficient than using a general nonlinear least-squares routine for this problem. Score tests for the transformations are also reported.

**Value**

an object of class `boxTidwell`, which is normally just printed.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

- Box, G. E. P. and Tidwell, P. W. (1962) Transformation of the independent variables. *Technometrics* **4**, 531-550.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Examples**

```
boxTidwell(prestige ~ income + education, ~ type + poly(women, 2), data=Prestige)
```

---

Burt

*Fraudulent Data on IQs of Twins Raised Apart*

---

**Description**

The `Burt` data frame has 27 rows and 4 columns. The “data” were simply (and notoriously) manufactured. The same data are in the dataset “twins” in the `alr3` package, but with different labels.

**Usage**

```
Burt
```

**Format**

This data frame contains the following columns:

**IQbio** IQ of twin raised by biological parents

**IQfoster** IQ of twin raised by foster parents

**class** A factor with levels (note: out of order): high; low; medium.

**Source**

Burt, C. (1966) The genetic determination of differences in intelligence: A study of monozygotic twins reared together and apart. *British Journal of Psychology* **57**, 137–153.

---

CanPop

*Canadian Population Data*

---

### Description

The CanPop data frame has 16 rows and 2 columns. Decennial time-series of Canadian population, 1851–2001.

### Usage

CanPop

### Format

This data frame contains the following columns:

**year** census year.

**population** Population, in millions

### Source

Urquhart, M. C. and Buckley, K. A. H. (Eds.) (1965) *Historical Statistics of Canada*. Macmillan, p. 1369.

Canada (1994) *Canada Year Book*. Statistics Canada, Table 3.2.

Statistics Canada: <http://www12.statcan.ca/english/census01/products/standard/popdwell/Table-PR.cfm>.

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

car-deprecated

*Deprecated Functions in car Package*

---

### Description

These functions are provided for compatibility with older versions of the **car** package only, and may be removed eventually. Commands that worked in versions of the **car** package prior to version 2.0-0 will not necessarily work in version 2.0-0 and beyond, or may not work in the same manner.



## Usage

```
av.plot(...)  
av.plots(...)  
box.cox(...)  
bc(...)  
box.cox.powers(...)  
box.cox.var(...)  
box.tidwell(...)  
cookd(...)  
confidence.ellipse(...)  
ceres.plot(...)  
ceres.plots(...)  
cr.plot(...)  
cr.plots(...)  
data.ellipse(...)  
durbin.watson(...)  
levene.test(...)  
leverage.plot(...)  
leverage.plots(...)  
linear.hypothesis(...)  
ncv.test(...)  
outlier.test(...)  
qq.plot(...)  
scatterplot.matrix(...)  
spread.level.plot(...)
```

## Arguments

... pass arguments down.

## Details

`av.plot` and `av.plots` are now synonyms for the `avPlot` and `avPlots` functions.

`box.cox` and `bc` are now synonyms for `bcPower`.

`box.cox.powers` is now a synonym for `powerTransform`.

`box.cox.var` is now a synonym for `boxCoxVariable`.

`box.tidwell` is now a synonym for `boxTidwell`.

`cookd` is now a synonym for `cooks.distance` in the `stats` package.

`confidence.ellipse` is now a synonym for `confidenceEllipse`.

`ceres.plot` and `ceres.plots` are now synonyms for the `ceresPlot` and `ceresPlots` functions.

`cr.plot` and `cr.plots` are now synonyms for the `crPlot` and `crPlots` functions.

`data.ellipse` is now a synonym for `dataEllipse`.

`durbin.watson` is now a synonym for `durbinWatsonTest`.

`levene.test` is now a synonym for `leveneTest` function.

leverage.plot and leverage.plots are now synonyms for the [leveragePlot](#) and [leveragePlots](#) functions.

linear.hypothesis is now a synonym for the [linearHypothesis](#) function.

ncv.test is now a synonym for [ncvTest](#).

outlier.test is now a synonym for [outlierTest](#).

qq.plot is now a synonym for [qqPlot](#).

scatterplot.matrix is now a synonym for [scatterplotMatrix](#).

spread.level.plot is now a synonym for [spreadLevelPlot](#).

car-internal.Rd

*Internal Objects for the **car** package*

### Description

These objects (currently only the .carEnv environment) are exported for technical reasons and are not for direct use.

### Author(s)

John Fox <jfox@mcmaster.ca>

carWeb

*Access to the R Companion to Applied Regression website*

### Description

This function will access the website for *An R Companion to Applied Regression*.

### Usage

```
carWeb(page = c("webpage", "errata", "taskviews"), script, data)
```

### Arguments

- |        |   |
|--------|---|
| page   | A character string indicating what page to open. The default "webpage" will open the main web page, "errata" displays the errata sheet for the book, and "taskviews" fetches and displays a list of available task views from CRAN.       |
| script | The quoted name of a chapter in <i>An R Companion to Applied Regression</i> , like "chap-1", "chap-2", up to "chap-8". All the R commands used in that chapter will be displayed in your browser, where you can save them as a text file. |
| data   | The quoted name of a data file in <i>An R Companion to Applied Regression</i> , like "Duncan.txt" or "Prestige.txt". The file will be opened in your web browser. You do not need to specify the extension .txt                           |

**Value**

Either a web page or a PDF document is displayed. Only one of the three arguments `page`, `rfile`, or `data`, should be used.

**Author(s)**

Sanford Weisberg, based on the function `UsingR` in the **UsingR** package by John Verzani

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Examples**

```
## Not run: carWeb()
```

---

ceresPlots

*Ceres Plots*

---

**Description**

These functions draw Ceres plots for linear and generalized linear models.

**Usage**

```
ceresPlots(model, terms = ~., layout = NULL, ask, main,
  ...)
```

```
ceresPlot(model, ...)
```

```
## S3 method for class 'lm'
```

```
ceresPlot(model, variable,
  id.method = list(abs(residuals(model, type="pearson")), "x"),
  labels,
  id.n = if(id.method[1]=="identify") Inf else 0,
  id.cex=1, id.col=palette()[1], id.location="lr",
  line=TRUE, smoother=loessLine, smoother.args=list(),
  smooth, span,
  col=palette()[1], col.lines=palette()[-1],
  xlab, ylab, pch=1, lwd=2,
  grid=TRUE, ...)
```

```
## S3 method for class 'glm'
```

```
ceresPlot(model, ...)
```

**Arguments**

model	model object produced by <code>lm</code> or <code>glm</code> .
terms	A one-sided formula that specifies a subset of the predictors. One component-plus-residual plot is drawn for each term. The default <code>~.</code> is to plot against all numeric predictors. For example, the specification <code>terms = ~. - X3</code> would plot against all predictors except for <code>X3</code> . Factors and nonstandard predictors such as B-splines are skipped. If this argument is a quoted name of one of the predictors, the component-plus-residual plot is drawn for that predictor only.
layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
ask	If TRUE, ask the user before drawing the next plot; if FALSE, the default, don't ask. This is relevant only if not all the graphs can be drawn in one window.
main	Overall title for any array of cerers plots; if missing a default is provided.
...	<code>ceresPlots</code> passes these arguments to <code>ceresPlot</code> . <code>ceresPlot</code> passes them to <code>plot</code> .
variable	A quoted string giving the name of a variable for the horizontal axis
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
line	TRUE to plot least-squares line.
smoother	Function to add a nonparametric smooth.
smoother.args	see <a href="#">ScatterplotSmoother</a> s for available smoothers and arguments.
smooth, span	these arguments are included for backwards compatibility: if <code>smooth=TRUE</code> then <code>smoother</code> is set to <code>loessLine</code> , and if <code>span</code> is specified, it is added to <code>smoother.args</code> .
col	color for points; the default is the first entry in the current color palette (see <a href="#">palette</a> and <a href="#">par</a> ).
col.lines	a list of at least two colors. The first color is used for the ls line and the second color is used for the fitted lowess line. To use the same color for both, use, for example, <code>col.lines=c("red", "red")</code>
xlab, ylab	labels for the x and y axes, respectively. If not set appropriate labels are created by the function.
pch	plotting character for points; default is 1 (a circle, see <a href="#">par</a> ).
lwd	line width; default is 2 (see <a href="#">par</a> ).
grid	If TRUE, the default, a light-gray background grid is put on the graph

**Details**

Ceres plots are a generalization of component+residual (partial residual) plots that are less prone to leakage of nonlinearity among the predictors.

The function intended for direct use is `ceresPlots`.

The model cannot contain interactions, but can contain factors. Factors may be present in the model, but Ceres plots cannot be drawn for them.

### Value

NULL. These functions are used for their side effect: producing plots.

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

Cook, R. D. and Weisberg, S. (1999) *Applied Regression, Including Computing and Graphics*. Wiley.

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

### See Also

[crPlots](#), [avPlots](#), [showLabels](#)

### Examples

```
ceresPlots(lm(prestige~income+education+type, data=Prestige), terms= ~ . - type)
```

---

Chile

*Voting Intentions in the 1988 Chilean Plebiscite*

---

### Description

The Chile data frame has 2700 rows and 8 columns. The data are from a national survey conducted in April and May of 1988 by FLACSO/Chile. There are some missing data.

### Usage

Chile

**Format**

This data frame contains the following columns:

**region** A factor with levels: C, Central; M, Metropolitan Santiago area; N, North; S, South; SA, city of Santiago.

**population** Population size of respondent's community.

**sex** A factor with levels: F, female; M, male.

**age** in years.

**education** A factor with levels (note: out of order): P, Primary; PS, Post-secondary; S, Secondary.

**income** Monthly income, in Pesos.

**statusquo** Scale of support for the status-quo.

**vote** a factor with levels: A, will abstain; N, will vote no (against Pinochet); U, undecided; Y, will vote yes (for Pinochet).

**Source**

Personal communication from FLACSO/Chile.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

Chirot

*The 1907 Romanian Peasant Rebellion*

---

**Description**

The Chirot data frame has 32 rows and 5 columns. The observations are counties in Romania.

**Usage**

Chirot

**Format**

This data frame contains the following columns:

**intensity** Intensity of the rebellion

**commerce** Commercialization of agriculture

**tradition** Traditionalism

**midpeasant** Strength of middle peasantry

**inequality** Inequality of land tenure

**Source**

Chiro, D. and C. Ragin (1975) The market, tradition and peasant rebellion: The case of Romania. *American Sociological Review* **40**, 428–444 [Table 1].

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

compareCoefs	<i>Print estimated coefficients and their standard errors in a table for several regression models.</i>
--------------	---

---

**Description**

This simple function extracts estimates of regression parameters and their standard errors from one or more models and prints them in a table.

**Usage**

```
compareCoefs(..., se = TRUE, print=TRUE, digits = 3)
```

**Arguments**

...	One or more regression-model objects. These may be of class <code>lm</code> , <code>glm</code> , <code>nlm</code> , or any other regression method for which the functions <code>coef</code> and <code>vcov</code> return appropriate values, or if the object inherits from the <code>mer</code> class created by the <code>lme4</code> package or <code>lme</code> in the <code>nlme</code> package.
se	If TRUE, the default, show standard errors as well as estimates, if FALSE, show only estimates.
print	If TRUE, the default, the results are printed in a nice format using <code>printCoefmat</code> . If FALSE, the results are returned as a matrix
digits	Passed to the <code>printCoefmat</code> function for printing the result.

**Value**

This function is used for its side-effect of printing the result. It returns a matrix of estimates and standard errors.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Examples**

```

mod1 <- lm(prestige ~ income + education, data=Duncan)
mod2 <- update(mod1, subset=-c(6,16))
mod3 <- update(mod1, . ~ . + type)
compareCoefs(mod1)
compareCoefs(mod1, mod2)
compareCoefs(mod1, mod2, mod3)
compareCoefs(mod1, mod2, se=FALSE)

```

---

**Contrasts**
*Functions to Construct Contrasts*


---

**Description**

These are substitutes for similarly named functions in the **stats** package (note the uppercase letter starting the second word in each function name). The only difference is that the contrast functions from the **car** package produce easier-to-read names for the contrasts when they are used in statistical models.

The functions and this documentation are adapted from the **stats** package.

**Usage**

```
contr.Treatment(n, base = 1, contrasts = TRUE)
```

```
contr.Sum(n, contrasts = TRUE)
```

```
contr.Helmert(n, contrasts = TRUE)
```

**Arguments**

<code>n</code>	a vector of levels for a factor, or the number of levels.
<code>base</code>	an integer specifying which level is considered the baseline level. Ignored if <code>contrasts</code> is <code>FALSE</code> .
<code>contrasts</code>	a logical indicating whether contrasts should be computed.

**Details**

These functions are used for creating contrast matrices for use in fitting analysis of variance and regression models. The columns of the resulting matrices contain contrasts which can be used for coding a factor with `n` levels. The returned value contains the computed contrasts. If the argument `contrasts` is `FALSE` then a square matrix is returned.

Several aspects of these contrast functions are controlled by options set via the `options` command:

`decorate.contrasts` This option should be set to a 2-element character vector containing the prefix and suffix characters to surround contrast names. If the option is not set, then `c("[", "]")` is used. For example, setting `options(decorate.contrasts=c(".", ""))` produces contrast names that are separated from factor names by a period. Setting `options(decorate.contrasts=c("", ""))` reproduces the behaviour of the R base contrast functions.



`decorate.contr.Treatment` A character string to be appended to contrast names to signify treatment contrasts; if the option is unset, then "T." is used.

`decorate.contr.Sum` Similar to the above, with default "S.".

`decorate.contr.Helmert` Similar to the above, with default "H.".

`contr.Sum.show.levels` Logical value: if TRUE (the default if unset), then level names are used for contrasts; if FALSE, then numbers are used, as in `contr.sum` in the base package.

Note that there is no replacement for `contr.poly` in the base package (which produces orthogonal-polynomial contrasts) since this function already constructs easy-to-read contrast names.

### Value

A matrix with  $n$  rows and  $k$  columns, with  $k = n - 1$  if `contrasts` is TRUE and  $k = n$  if `contrasts` is FALSE.

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

### See Also

[contr.treatment](#), [contr.sum](#), [contr.helmert](#), [contr.poly](#)

### Examples

```
# contr.Treatment vs. contr.treatment in the base package:

lm(prestige ~ (income + education)*type, data=Prestige,
   contrasts=list(type="contr.Treatment"))

## Call:
## lm(formula = prestige ~ (income + education) * type, data = Prestige,
##     contrasts = list(type = "contr.Treatment"))
##
## Coefficients:
##      (Intercept)          income          education
##      2.275753          0.003522          1.713275
##      type[T.prof]          type[T.wc]  income:type[T.prof]
##      15.351896          -33.536652          -0.002903
##      income:type[T.wc]  education:type[T.prof]  education:type[T.wc]
##      -0.002072          1.387809          4.290875

lm(prestige ~ (income + education)*type, data=Prestige,
   contrasts=list(type="contr.treatment"))

## Call:
## lm(formula = prestige ~ (income + education) * type, data = Prestige,
```

```
## contrasts = list(type = "contr.treatment")
##
## Coefficients:
## (Intercept)          income          education
## 2.275753          0.003522          1.713275
## typeprof           typewc          income:typeprof
## 15.351896         -33.536652         -0.002903
## income:typewc      education:typeprof  education:typewc
## -0.002072          1.387809           4.290875
```

---

Cowles

*Cowles and Davis's Data on Volunteering*

---

## Description

The Cowles data frame has 1421 rows and 4 columns. These data come from a study of the personality determinants of volunteering for psychological research.

## Usage

Cowles

## Format

This data frame contains the following columns:

**neuroticism** scale from Eysenck personality inventory

**extraversion** scale from Eysenck personality inventory

**sex** a factor with levels: female; male

**volunteer** volunteering, a factor with levels: no; yes

## Source

Cowles, M. and C. Davis (1987) The subject matter of psychology: Volunteers. *British Journal of Social Psychology* **26**, 97–102.

---

 crPlots *Component+Residual (Partial Residual) Plots*


---

**Description**

These functions construct component+residual plots (also called partial-residual plots) for linear and generalized linear models.

**Usage**

```
crPlots(model, terms = ~., layout = NULL, ask, main,
        ...)

crp(...)

crPlot(model, ...)

## S3 method for class 'lm'
crPlot(model, variable,
       id.method = list(abs(residuals(model, type="pearson")), "x"),
       labels,
       id.n = if(id.method[1]=="identify") Inf else 0,
       id.cex=1, id.col=palette()[1], id.location="lr",
       order=1, line=TRUE, smoother=loessLine,
       smoother.args=list(), smooth, span,
       col=palette()[1], col.lines=palette()[-1],
       xlab, ylab, pch=1, lwd=2, grid=TRUE, ...)
```

**Arguments**

model	model object produced by lm or glm.
terms	A one-sided formula that specifies a subset of the predictors. One component-plus-residual plot is drawn for each term. The default <code>~.</code> is to plot against all numeric predictors. For example, the specification <code>terms = ~ . - X3</code> would plot against all predictors except for X3. If this argument is a quoted name of one of the predictors, the component-plus-residual plot is drawn for that predictor only.
layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
ask	If TRUE, ask the user before drawing the next plot; if FALSE, the default, don't ask. This is relevant only if not all the graphs can be drawn in one window.
main	The title of the plot; if missing, one will be supplied.

...	crPlots passes these arguments to crPlot. crPlot passes them to plot.
variable	A quoted string giving the name of a variable for the horizontal axis
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is id.n=0 for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
order	order of polynomial regression performed for predictor to be plotted; default 1.
line	TRUE to plot least-squares line.
smoother	Function to add a nonparametric smooth.
smoother.args	see <a href="#">ScatterplotSmoothers</a> for available smoothers and arguments.
smooth, span	these arguments are included for backwards compatibility: if smooth=TRUE then smoother is set to loessLine, and if span is specified, it is added to smoother.args.
col	color for points; the default is the first entry in the current color palette (see <a href="#">palette</a> and <a href="#">par</a> ).
col.lines	a list of at least two colors. The first color is used for the ls line and the second color is used for the fitted lowess line. To use the same color for both, use, for example, col.lines=c("red", "red")
xlab, ylab	labels for the x and y axes, respectively. If not set appropriate labels are created by the function.
pch	plotting character for points; default is 1 (a circle, see <a href="#">par</a> ).
lwd	line width; default is 2 (see <a href="#">par</a> ).
grid	If TRUE, the default, a light-gray background grid is put on the graph

### Details

The function intended for direct use is crPlots, for which crp is an abbreviation.

The model cannot contain interactions, but can contain factors. Parallel boxplots of the partial residuals are drawn for the levels of a factor.

### Value

NULL. These functions are used for their side effect of producing plots.

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

- Cook, R. D. and Weisberg, S. (1999) *Applied Regression, Including Computing and Graphics*. Wiley.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[ceresPlots](#), [avPlots](#)

**Examples**

```
crPlots(m<-lm(prestige~income+education, data=Prestige))
# get only one plot
crPlots(m, terms=~ . - education)

crPlots(lm(prestige ~ log2(income) + education + poly(women,2), data=Prestige))

crPlots(glm(partic != "not.work" ~ hincome + children,
  data=Women1f, family=binomial))
```

---

 Davis

*Self-Reports of Height and Weight*


---

**Description**

The Davis data frame has 200 rows and 5 columns. The subjects were men and women engaged in regular exercise. There are some missing data.

**Usage**

Davis

**Format**

This data frame contains the following columns:

**sex** A factor with levels: F, female; M, male.

**weight** Measured weight in kg.

**height** Measured height in cm.

**repwt** Reported weight in kg.

**reph** Reported height in cm.

**Source**

Personal communication from C. Davis, Departments of Physical Education and Psychology, York University.

**References**

Davis, C. (1990) Body image and weight preoccupation: A comparison between exercising and non-exercising women. *Appetite*, **15**, 13–21.

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

DavisThin

*Davis's Data on Drive for Thinness*

---

### Description

The DavisThin data frame has 191 rows and 7 columns. This is part of a larger dataset for a study of eating disorders. The seven variables in the data frame comprise a "drive for thinness" scale, to be formed by summing the items.

### Usage

```
DavisThin
```

### Format

This data frame contains the following columns:

**DT1** a numeric vector

**DT2** a numeric vector

**DT3** a numeric vector

**DT4** a numeric vector

**DT5** a numeric vector

**DT6** a numeric vector

**DT7** a numeric vector

### Source

Davis, C., G. Claridge, and D. Cerullo (1997) Personality factors predisposing to weight preoccupation: A continuum approach to the association between eating disorders and personality disorders. *Journal of Psychiatric Research* **31**, 467–480. [personal communication from the authors.]

### References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

deltaMethod	<i>Estimate and Standard Error of a Nonlinear Function of Estimated Regression Coefficients</i>
-------------	---

---

## Description

deltaMethod is a generic function that uses the delta method to get a first-order approximate standard error for a nonlinear function of a vector of random variables with known or estimated covariance matrix.

## Usage

```
deltaMethod(object, ...)

## Default S3 method:
deltaMethod(object, g, vcov., func=g, constants, level=0.95, ...)
## S3 method for class 'lm'
  deltaMethod(object, g, vcov.=vcov,
              parameterNames=names(coef(object)), ...)
## S3 method for class 'nls'
deltaMethod(object, g, vcov.=vcov, ...)
## S3 method for class 'multinom'
  deltaMethod(object, g, vcov. = vcov,
              parameterNames = if (is.matrix(coef(object)))
                colnames(coef(object)) else names(coef(object)), ...)
## S3 method for class 'polr'
  deltaMethod(object, g, vcov.=vcov, ...)
## S3 method for class 'survreg'
  deltaMethod(object, g, vcov. = vcov,
              parameterNames = names(coef(object)), ...)
## S3 method for class 'coxph'
  deltaMethod(object, g, vcov. = vcov,
              parameterNames = names(coef(object)), ...)
## S3 method for class 'mer'
  deltaMethod(object, g, vcov. = vcov,
              parameterNames = names(fixef(object)), ...)
## S3 method for class 'merMod'
  deltaMethod(object, g, vcov. = vcov,
              parameterNames = names(fixef(object)), ...)
## S3 method for class 'lme'
  deltaMethod(object, g, vcov. = vcov,
              parameterNames = names(fixef(object)), ...)
## S3 method for class 'lmList'
  deltaMethod(object, g, ...)
```

**Arguments**

object	For the default method, object is either (1) a vector of $p$ named elements, so <code>names(object)</code> returns a list of $p$ character strings that are the names of the elements of object; or (2) a model object for which there are <code>coef</code> and <code>vcov</code> methods, and for which the named coefficient vector returned by <code>coef</code> is asymptotically normally distributed with asymptotic covariance matrix returned by <code>vcov</code> . For the other methods, object is a regression object for which <code>coef(object)</code> or <code>fixef(object)</code> returns a vector of parameter estimates.
g	A quoted string that is the function of the parameter estimates to be evaluated; see the details below.
vcov.	The (estimated) covariance matrix of the coefficient estimates. For the default method, this argument is required. For all other methods, this argument must either provide the estimated covariance matrix or a function that when applied to object returns a covariance matrix. The default is to use the function <code>vcov</code> .
func	A quoted string used to annotate output. The default of <code>func = g</code> is usually appropriate.
parameterNames	A character vector of length $p$ that gives the names of the parameters in the same order as they appear in the vector of estimates. This argument will be useful if some of the names in the vector of estimates include special characters, like $I(x^2)$ , or $x1 : x2$ that will confuse the numerical differentiation function. See details below.
constants	This argument is a named vector whose elements are constants that are used in the <code>f</code> argument. This is needed only when the function is called from within another function to comply to R scoping rules. See example below.
level	level for confidence interval, default 0.95.
...	Used to pass arguments to the generic method.

**Details**

Suppose  $x$  is a random vector of length  $p$  that is at least approximately normally distributed with mean  $\beta$  and estimated covariance matrix  $C$ . Then any function  $g(\beta)$  of  $\beta$ , is estimated by  $g(x)$ , which is in large samples normally distributed with mean  $g(\beta)$  and estimated variance  $h'Ch$ , where  $h$  is the first derivative of  $g(\beta)$  with respect to  $\beta$  evaluated at  $x$ . This function returns both  $g(x)$  and its standard error, the square root of the estimated variance.

The default method requires that you provide  $x$  in the argument `object`,  $C$  in the argument `vcov.`, and a text expression in argument `g` that when evaluated gives the function  $g$ . The call `names(object)` must return the names of the elements of  $x$  that are used in the expression  $g$ .

Since the delta method is often applied to functions of regression parameter estimates, the argument `object` may be the name of a regression object from which the the estimates and their estimated variance matrix can be extracted. In most regression models, estimates are returned by the `coef(object)` and the variance matrix from `vcov(object)`. You can provide an alternative function for computing the sample variance matrix, for example to use a sandwich estimator.

For mixed models using `lme4` or `nlme`, the coefficient estimates are returned by the `fixef` function, while for `multinom`, `lmlist` and `nlsList` coefficient estimates are returned by `coef` as a matrix. Methods for these models are provided to get the correct estimates and variance matrix.



The argument `g` must be a quoted character string that gives the function of interest. For example, if you set `m2 <- lm(Y ~ X1 + X2 + X1:X2)`, then `deltaMethod(m2, "X1/X2")` applies the delta method to the ratio of the coefficient estimates for `X1` and `X2`. The argument `g` can consist of constants and names associated with the elements of the vector of coefficient estimates.

In some cases the names may include characters including such as the colon `:` used in interactions, or mathematical symbols like `+` or `-` signs that would confuse the function that computes numerical derivatives, and for this case you can replace the names of the estimates with the `parameterNames` argument. For example, the ratio of the `X2` main effect to the interaction term could be computed using `deltaMethod(m2, "b1/b3", parameterNames=c("b0", "b1", "b2", "b3"))`. The name "(Intercept)" used for the intercept in linear and generalized linear models is an exception, and it will be correctly interpreted by `deltaMethod`.

For `multinom` objects, the `coef` function returns a matrix of coefficients, with each row giving the estimates for comparisons of one category to the baseline. The `deltaMethod` function applies the delta method to each row of this matrix. Similarly, for `lmList` and `nlsList` objects, the delta method is computed for each element of the list of models fit.

For nonlinear regression objects of type `nls`, the call `coef(object)` returns the estimated coefficient vectors with names corresponding to parameter names. For example, `m2 <- nls(y ~ theta/(1 + gamma * x), start = li` will have parameters named `c("theta", "gamma")`. In many other familiar regression methods, such as `lm` and `glm`, the names of the coefficient estimates are the corresponding variable names, not parameter names.

For mixed-effects models fit with `lmer` and `nlmer` from the `lme4` package or `lme` and `nlme` from the `nlme` package, only fixed-effect coefficients are considered.

For regression models for which methods are not provided, you can extract the named vector of coefficient estimates and an estimate of its covariance matrix and then apply the default `deltaMethod` function.

Earlier versions of `deltaMethod` included an argument `parameterPrefix` that implemented the same functionality as the `parameterNames` argument, but it caused several unintended bugs that were not easily fixed without the change in syntax.

## Value

A `data.frame` with two components named `Estimate` for the estimate, `SE` for its standard error. The value of `g` is given as a row label.

## Author(s)

Sanford Weisberg, <sandy@umn.edu>, and John Fox <jfox@mcmaster.ca>

## References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.  
 Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.  
 Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley, Section 6.1.2.

## See Also

First derivatives of `g` are computed using symbolic differentiation by the function `D`.

## Examples

```

m1 <- lm(time ~ t1 + t2, data = Transact)
deltaMethod(m1, "b1/b2", parameterNames= paste("b", 0:2, sep=""))
deltaMethod(m1, "t1/t2") # use names of preds. rather than coefs.
deltaMethod(m1, "t1/t2", vcov=hccm) # use hccm function to est. vars.
# to get the SE of 1/intercept, rename coefficients
deltaMethod(m1, "1/b0", parameterNames= paste("b", 0:2, sep=""))
# The next example calls the default method by extracting the
# vector of estimates and covariance matrix explicitly
deltaMethod(coef(m1), "t1/t2", vcov=vcov(m1))
# the following works:
a <- 5
deltaMethod(m1, "(t1 + a)/t2")
# ..but embedded in a function this will fail
f1 <- function(mod, ...) {
  z <- 3
  deltaMethod(m1, "(t1+z)/t2", ...)
}
## Not run: f1(m1)
# if z is defined globally f1 could even return the wrong answer.
# the following function works
f2 <- function(mod, ...) {
  deltaMethod(m1, "(t1+z)/t2", ...)
}
f2(m1, constants=c(z=3))
# as does
f3 <- function(mod) {
  a <- 3
  deltaMethod(m1, "(t1+z)/t2", constants=c(z=a))
}
f3(m1)

```

---

densityPlot

*Nonparametric Density Estimates*

---

## Description

densityPlot constructs and graphs nonparametric density estimates, possibly conditioned on a factor. By default it uses the standard R [density](#) function or optionally adaptiveKernel.

## Usage

```

densityPlot(x, ...)

## Default S3 method:
densityPlot(x, g, method=c("kernel", "adaptive"),
  bw=if (method == "adaptive") bw.nrd0 else "SJ", adjust=1,
  kernel, xlim, ylim, normalize=FALSE,
  xlab=deparse(substitute(x)), ylab="Density",

```

```

col=palette(), lty=seq_along(col), lwd=2, grid=TRUE,
legend.location="topright", legend.title=deparse(substitute(g)), show.bw=FALSE,
rug=TRUE, ...)

## S3 method for class 'formula'
densityPlot(formula, data = NULL, subset,
  na.action = NULL, xlab, ylab, ...)

adaptiveKernel(x, kernel=dnorm, bw=bw.nrd0, adjust=1.0, n=500,
  from, to, cut=3, na.rm=TRUE)

```

### Arguments

x	a numeric variable, the density of which is estimated.
g	an optional factor to divide the data.
formula	an R model formula, of the form <code>~ variable</code> to estimate the unconditional density of variable, or <code>variable ~ factor</code> to estimate the density of variable within each level of factor.
data	an optional data frame containing the data.
subset	an optional vector defining a subset of the data.
na.action	a function to handle missing values; defaults to the value of the R <code>na.action</code> option, initially set to <code>na.omit</code> .
method	either "adaptive" (the default) for an adaptive-kernel estimate or "kernel" for a fixed-bandwidth kernel estimate.
bw	the geometric mean bandwidth for the adaptive-kernel or bandwidth of the kernel density estimate(s). Must be a numerical value or a function to compute the bandwidth (default <code>bw.nrd0</code> ) for the adaptive kernel estimate; for the kernel estimate, may either the quoted name of a rule to compute the bandwidth, or a numeric value. If plotting by groups, <code>bw</code> may be a vector of values, one for each group. See <a href="#">density</a> and <code>bw.SJ</code> for details of the kernel estimator.
adjust	a multiplicative adjustment factor for the bandwidth; the default, 1, indicates no adjustment; if plotting by groups, <code>adjust</code> may be a vector of adjustment factors, one for each group. The default bandwidth-selection rule tends to give a value that's too large if the distribution is asymmetric or has multiple modes; try setting <code>adjust &lt; 1</code> , particularly for the adaptive-kernel estimator.
kernel	for <code>densityPlot</code> this is the name of the kernel function for the kernel estimator (the default is "gaussian", see <a href="#">density</a> ); or a kernel function for the adaptive-kernel estimator (the default is <code>dnorm</code> , producing the Gaussian kernel). For <code>adaptivekernel</code> this is a kernel function, defaulting to <code>dnorm</code> , which is the Gaussian kernel (standard-normal density).
xlim, ylim	axis limits; if missing, determined from the range of x-values at which the densities are estimated and the estimated densities.
normalize	if TRUE (the default is FALSE), the estimated densities are rescaled to integrate approximately to 1; particularly useful if the density is estimated over a restricted domain, as when <code>from</code> or <code>to</code> are specified.

xlab	label for the horizontal-axis; defaults to the name of the variable x.
ylab	label for the vertical axis; defaults to "Density".
col	vector of colors for the density estimate(s); defaults to the color <a href="#">palette</a> .
lty	vector of line types for the density estimate(s); defaults to the successive integers, starting at 1.
lwd	line width for the density estimate(s); defaults to 2.
grid	if TRUE (the default), grid lines are drawn on the plot.
legend.location	location for the legend when densities are plotted for several groups; defaults to "upperright"; see <a href="#">legend</a> .
legend.title	label for the legend, which is drawn if densities are plotted by groups; the default is the name of the factor g.
n	number of equally spaced points at which the adaptive-kernel estimator is evaluated; the default is 500.
from, to, cut	the range over which the density estimate is computed; the default, if missing, is $\min(x) - \text{cut} \cdot \text{bw}$ , $\max(x) + \text{cut} \cdot \text{bw}$ .
na.rm	remove missing values from x in computing the adaptive-kernel estimate? The default is TRUE.
show.bw	if TRUE, show the bandwidth(s) in the horizontal-axis label or (for multiple groups) the legend; the default is FALSE.
rug	if TRUE (the default), draw a rug plot (one-dimensional scatterplot) at the bottom of the density estimate.
...	arguments to be passed down.

### Value

densityPlot invisibly returns the "density" object computed (or list of "density" objects) and draws a graph. adaptiveKernel returns an object of class "density" (see [density](#)).

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

- W. N. Venables and B. D. Ripley (2002) *Modern Applied Statistics with S*. New York: Springer.
- B.W. Silverman (1986) *Density Estimation for Statistics and Data Analysis*. London: Chapman and Hall.

### See Also

[density](#), [bw.SJ](#), [plot.density](#)

**Examples**

```

densityPlot(~ income, show.bw=TRUE, data=Prestige)
densityPlot(~ income, method="adaptive", show.bw=TRUE, data=Prestige)
densityPlot(~ income, method="adaptive", from=0, normalize=TRUE, show.bw=TRUE, data=Prestige)

densityPlot(income ~ type, method="adaptive", data=Prestige)

plot(adaptiveKernel(UN$infant.mortality, from=0, adjust=0.75), col="magenta")
lines(density(na.omit(UN$infant.mortality), from=0, adjust=0.75), col="blue")
rug(UN$infant.mortality, col="cyan")
legend("topright", col=c("magenta", "blue"), lty=1,
      legend=c("adaptive kernel", "kernel"), inset=0.02)

```

---

 Depredations

*Minnesota Wolf Depredation Data*


---

**Description**

Wolf depredations of livestock on Minnesota farms, 1976-1998.

**Usage**

Depredations

**Format**

A data frame with 434 observations on the following 5 variables.

longitude longitude of the farm

latitude latitude of the farm

number number of depredations 1976-1998

early number of depredations 1991 or before

late number of depredations 1992 or later

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Harper, Elizabeth K. and Paul, William J. and Mech, L. David and Weisberg, Sanford (2008), Effectiveness of Lethal, Directed Wolf-Depredation Control in Minnesota, *Journal of Wildlife Management*, 72, 3, 778-784. <http://dx.doi.org/10.2193/2007-273>

dfbetaPlots

*dfbeta and dfbetas Index Plots***Description**

These functions display index plots of dfbeta (effect on coefficients of deleting each observation in turn) and dfbetas (effect on coefficients of deleting each observation in turn, standardized by a deleted estimate of the coefficient standard error). In the plot of dfbeta, horizontal lines are drawn at 0 and +/- one standard error; in the plot of dfbetas, horizontal lines are drawn at 0 and +/- 1.

**Usage**

```
dfbetaPlots(model, ...)

dfbetasPlots(model, ...)

## S3 method for class 'lm'
dfbetaPlots(model, terms=~ ., intercept=FALSE, layout=NULL, ask,
  main, xlab, ylab, labels=rownames(dfbeta),
  id.method="y",
  id.n=if(id.method[1]=="identify") Inf else 0, id.cex=1,
  id.col=palette()[1], id.location="lr", col=palette()[1], grid=TRUE, ...)

## S3 method for class 'lm'
dfbetasPlots(model, terms=~., intercept=FALSE, layout=NULL, ask,
  main, xlab, ylab,
  labels=rownames(dfbeta), id.method="y",
  id.n=if(id.method[1]=="identify") Inf else 0, id.cex=1,
  id.col=palette()[1], id.location="lr", col=palette()[1], grid=TRUE, ...)
```

**Arguments**

model	model object produced by lm or glm.
terms	A one-sided formula that specifies a subset of the terms in the model. One dfbeta or dfbetas plot is drawn for each regressor. The default <code>~.</code> is to plot against all terms in the model with the exception of an intercept. For example, the specification <code>terms = ~.-X3</code> would plot against all terms except for X3. If this argument is a quoted name of one of the terms, the index plot is drawn for that term only.
intercept	Include the intercept in the plots; default is FALSE.
layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.

<code>main</code>	The title of the graph; if missing, one will be supplied.
<code>xlab</code>	Horizontal axis label; defaults to "Index".
<code>ylab</code>	Vertical axis label; defaults to coefficient name.
<code>ask</code>	If TRUE, ask the user before drawing the next plot; if FALSE, the default, don't ask.
<code>...</code>	optional additional arguments to be passed to <a href="#">plot</a> , <a href="#">points</a> , and <a href="#">showLabels</a> .
<code>id.method, labels, id.n, id.cex, id.col, id.location</code>	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
<code>col</code>	color for points; defaults to the first entry in the color <a href="#">palette</a> .
<code>grid</code>	If TRUE, the default, a light-gray background grid is put on the graph

**Value**

NULL. These functions are used for their side effect: producing plots.

**Author(s)**

John Fox <jfox@mcmaster.ca>, Sanford Weisberg <sandy@umn.edu>

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.  
 Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[dfbeta](#), [dfbetas](#)

**Examples**

```
dfbetaPlots(lm(prestige ~ income + education + type, data=Duncan))

dfbetasPlots(glm(partic != "not.work" ~ hincome + children,
  data=Women1f, family=binomial))
```

---

Duncan

*Duncan's Occupational Prestige Data*

---

**Description**

The Duncan data frame has 45 rows and 4 columns. Data on the prestige and other characteristics of 45 U. S. occupations in 1950.

**Usage**

Duncan

**Format**

This data frame contains the following columns:

**type** Type of occupation. A factor with the following levels: prof, professional and managerial; wc, white-collar; bc, blue-collar.

**income** Percent of males in occupation earning \$3500 or more in 1950.

**education** Percent of males in occupation in 1950 who were high-school graduates.

**prestige** Percent of raters in NORC study rating occupation as excellent or good in prestige.

**Source**

Duncan, O. D. (1961) A socioeconomic index for all occupations. In Reiss, A. J., Jr. (Ed.) *Occupations and Social Status*. Free Press [Table VI-1].

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

durbinWatsonTest	<i>Durbin-Watson Test for Autocorrelated Errors</i>
------------------	---

---

**Description**

Computes residual autocorrelations and generalized Durbin-Watson statistics and their bootstrapped p-values. dwt is an abbreviation for durbinWatsonTest.

**Usage**

```
durbinWatsonTest(model, ...)

dwt(...)

## S3 method for class 'lm'
durbinWatsonTest(model, max.lag=1, simulate=TRUE, reps=1000,
  method=c("resample", "normal"),
  alternative=c("two.sided", "positive", "negative"), ...)

## Default S3 method:
durbinWatsonTest(model, max.lag=1, ...)

## S3 method for class 'durbinWatsonTest'
print(x, ...)
```



**Arguments**

<code>model</code>	a linear-model object, or a vector of residuals from a linear model.
<code>max.lag</code>	maximum lag to which to compute residual autocorrelations and Durbin-Watson statistics.
<code>simulate</code>	if TRUE p-values will be estimated by bootstrapping.
<code>reps</code>	number of bootstrap replications.
<code>method</code>	bootstrap method: "resample" to resample from the observed residuals; "normal" to sample normally distributed errors with 0 mean and standard deviation equal to the standard error of the regression.
<code>alternative</code>	sign of autocorrelation in alternative hypothesis; specify only if <code>max.lag = 1</code> ; if <code>max.lag &gt; 1</code> , then <code>alternative</code> is taken to be "two.sided".
<code>...</code>	arguments to be passed down.
<code>x</code>	<code>durbinWatsonTest</code> object.

**Value**

Returns an object of type "durbinWatsonTest".

**Note**

p-values are available only from the `lm` method.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

**Examples**

```
durbinWatsonTest(lm(fconvict ~ tfr + partic + degrees + mconvict, data=Hartnagel))
```

**Description**

These functions draw ellipses, including data ellipses, and confidence ellipses for linear, generalized linear, and possibly other models.

**Usage**

```

ellipse(center, shape, radius, log="", center.pch=19, center.cex=1.5,
  segments=51, draw=TRUE, add=draw, xlab="", ylab="",
  col=palette()[2], lwd=2, fill=FALSE, fill.alpha=0.3, grid=TRUE, ...)

dataEllipse(x, y, groups, group.labels = group.levels, ellipse.label,
  weights, log = "", levels = c(0.5, 0.95), center.pch = 19,
  center.cex = 1.5, draw = TRUE, plot.points = draw, add = !plot.points,
  segments = 51, robust = FALSE, xlab = deparse(substitute(x)),
  ylab = deparse(substitute(y)),
  col = if (missing(groups)) palette()[1:2] else palette()[1:length(group.levels)],
  pch = if (missing(groups)) 1 else seq(group.levels), lwd = 2,
  fill = FALSE, fill.alpha = 0.3, grid = TRUE, labels, id.method = "mahal",
  id.n = if (id.method[1] == "identify") Inf else 0, id.cex = 1,
  id.col = if (missing(groups)) palette()[1] else palette()(1:length(groups)),
  id.location="lr", ...)

confidenceEllipse(model, ...)

## S3 method for class 'lm'
confidenceEllipse(model, which.coef, L, levels=0.95, Scheffe=FALSE, dfn,
  center.pch=19, center.cex=1.5, segments=51, xlab, ylab,
  col=palette()[2], lwd=2, fill=FALSE, fill.alpha=0.3, draw=TRUE, add=!draw, ...)

## S3 method for class 'glm'
confidenceEllipse(model, chisq, ...)

## Default S3 method:
confidenceEllipse(model, which.coef, L, levels=0.95, Scheffe=FALSE, dfn,
  center.pch=19, center.cex=1.5, segments=51, xlab, ylab,
  col=palette()[2], lwd=2, fill=FALSE, fill.alpha=0.3, draw=TRUE, add=!draw, ...)

```

**Arguments**

<code>center</code>	2-element vector with coordinates of center of ellipse.
<code>shape</code>	$2 \times 2$ shape (or covariance) matrix.
<code>radius</code>	radius of circle generating the ellipse.
<code>log</code>	when an ellipse is to be added to an existing plot, indicates whether computations were on logged values and to be plotted on logged axes; "x" if the x-axis is logged, "y" if the y-axis is logged, and "xy" or "yx" if both axes are logged. The default is "", indicating that neither axis is logged.
<code>center.pch</code>	character for plotting ellipse center; if FALSE or NULL the center point isn't plotted.
<code>center.cex</code>	relative size of character for plotting ellipse center.
<code>segments</code>	number of line-segments used to draw ellipse.
<code>draw</code>	if TRUE produce graphical output; if FALSE, only invisibly return coordinates of ellipse(s).

<code>add</code>	if TRUE add ellipse(s) to current plot.
<code>xlab</code>	label for horizontal axis.
<code>ylab</code>	label for vertical axis.
<code>x</code>	a numeric vector, or (if <code>y</code> is missing) a 2-column numeric matrix.
<code>y</code>	a numeric vector, of the same length as <code>x</code> .
<code>groups</code>	optional: a factor to divide the data into groups; a separate ellipse will be plotted for each group (level of the factor).
<code>group.labels</code>	labels to be plotted for the groups; by default, the levels of the groups factor.
<code>ellipse.label</code>	a label for the ellipse(s) or a vector of labels; if several ellipses are drawn and just one label is given, then that label will be repeated. The default is not to label the ellipses.
<code>weights</code>	a numeric vector of weights, of the same length as <code>x</code> and <code>y</code> to be used by <code>cov.wt</code> or <code>cov.trob</code> in computing a weighted covariance matrix; if absent, weights of 1 are used.
<code>plot.points</code>	if FALSE data ellipses are drawn, but points are not plotted.
<code>levels</code>	draw elliptical contours at these (normal) probability or confidence levels.
<code>robust</code>	if TRUE use the <code>cov.trob</code> function in the <b>MASS</b> package to calculate the center and covariance matrix for the data ellipse.
<code>model</code>	a model object produced by <code>lm</code> or <code>glm</code> .
<code>which.coef</code>	2-element vector giving indices of coefficients to plot; if missing, the first two coefficients (disregarding the regression constant) will be selected.
<code>L</code>	As an alternative to selecting coefficients to plot, a transformation matrix can be specified to compute two linear combinations of the coefficients; if the <code>L</code> matrix is given, it takes precedence over the <code>which.coef</code> argument. <code>L</code> should have two rows and as many columns as there are coefficients. It can be given directly as a numeric matrix, or specified by a pair of character-valued expressions, in the same manner as for the <code>link{linearHypothesis}</code> function, but with no right-hand side.
<code>Scheffe</code>	if TRUE scale the ellipse so that its projections onto the axes give Scheffe confidence intervals for the coefficients.
<code>dfn</code>	“numerator” degrees of freedom (or just degrees of freedom for a GLM) for drawing the confidence ellipse. Defaults to the number of coefficients in the model (disregarding the constant) if <code>Scheffe</code> is TRUE, or to 2 otherwise; selecting <code>dfn = 1</code> will draw the “confidence-interval generating” ellipse, with projections on the axes corresponding to individual confidence intervals with the stated level of coverage.
<code>chisq</code>	if TRUE, the confidence ellipse for the coefficients in a generalized linear model are based on the chisquare statistic, if FALSE on the $F$ -statistic. This corresponds to using the default and linear-model methods respectively.
<code>col</code>	color for lines and ellipse center; the default is the <i>second</i> entry in the current color palette (see <code>palette</code> and <code>par</code> ). For <code>dataEllipse</code> , two colors can be given, in which case the first is for plotted points and the second for lines and the ellipse center; if ellipses are plotted for groups, then this is a vector of colors for the groups.

<code>pch</code>	for <code>dataEllipse</code> this is the plotting character (default, symbol 1, a hollow circle) to use for the points; if ellipses are plotted by groups, then this a vector of plotting characters, with consecutive symbols starting with 1 as the default.
<code>lwd</code>	line width; default is 2 (see <a href="#">par</a> ).
<code>fill</code>	fill the ellipse with translucent color <code>col</code> (default, FALSE)?
<code>fill.alpha</code>	transparency of fill (default = 0.3).
<code>...</code>	other plotting parameters to be passed to <code>plot</code> and <code>line</code> .
<code>labels, id.method, id.n, id.cex, id.col, id.location</code>	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
<code>grid</code>	If TRUE, the default, a light-gray background grid is put on the graph

### Details

The ellipse is computed by suitably transforming a unit circle.

`dataEllipse` superimposes the normal-probability contours over a scatterplot of the data.

### Value

These functions are mainly used for their side effect of producing plots. For greater flexibility (e.g., adding plot annotations), however, `ellipse` returns invisibly the (x, y) coordinates of the calculated ellipse. `dataEllipse` and `confidenceEllipse` return invisibly the coordinates of one or more ellipses, in the latter instance a list named by levels.

### Author(s)

Georges Monette, John Fox <jfox@mcmaster.ca>, and Michael Friendly.

### References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Monette, G. (1990) Geometry of multiple regression and 3D graphics. In Fox, J. and Long, J. S. (Eds.) *Modern Methods of Data Analysis*. Sage.

### See Also

[cov.trob](#), [cov.wt](#), [linearHypothesis](#).

### Examples

```
dataEllipse(Duncan$income, Duncan$education, levels=0.1*1:9,
            ellipse.label=0.1*1:9, lty=2, fill=TRUE, fill.alpha=0.1)
confidenceEllipse(lm(prestige~income+education, data=Duncan), Scheffe=TRUE)
confidenceEllipse(lm(prestige~income+education, data=Duncan),
                 L=c("income + education", "income - education"))

wts <- rep(1, nrow(Duncan))
```

```

wts[c(6, 16)] <- 0 # delete Minister, Conductor
with(Duncan, {
  dataEllipse(income, prestige, levels=0.68)
  dataEllipse(income, prestige, levels=0.68, robust=TRUE, plot.points=FALSE, col="green3")
  dataEllipse(income, prestige, weights=wts, levels=0.68, plot.points=FALSE, col="brown")
  dataEllipse(income, prestige, weights=wts, robust=TRUE, levels=0.68,
  plot.points=FALSE, col="blue")
})

with(Prestige, dataEllipse(income, education, type, id.n=2, pch=15:17,
  labels=rownames(Prestige), xlim=c(0, 25000), center.pch="+",
  group.labels=c("Blue Collar", "Professional", "White Collar"),
  ylim=c(5, 20), level=.95, fill=TRUE, fill.alpha=0.1))

```

Ericksen

*The 1980 U.S. Census Undercount***Description**

The Ericksen data frame has 66 rows and 9 columns. The observations are 16 large cities, the remaining parts of the states in which these cities are located, and the other U. S. states.

**Usage**

```
Ericksen
```

**Format**

This data frame contains the following columns:

**minority** Percentage black or Hispanic.

**crime** Rate of serious crimes per 1000 population.

**poverty** Percentage poor.

**language** Percentage having difficulty speaking or writing English.

**highschool** Percentage age 25 or older who had not finished highschool.

**housing** Percentage of housing in small, multiunit buildings.

**city** A factor with levels: city, major city; state, state or state-remainder.

**conventional** Percentage of households counted by conventional personal enumeration.

**undercount** Preliminary estimate of percentage undercount.

**Source**

Ericksen, E. P., Kadane, J. B. and Tukey, J. W. (1989) Adjusting the 1980 Census of Population and Housing. *Journal of the American Statistical Association* **84**, 927–944 [Tables 7 and 8].

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

Florida

*Florida County Voting*

---

### **Description**

The Florida data frame has 67 rows and 11 columns. Vote by county in Florida for President in the 2000 election.

### **Usage**

Florida

### **Format**

This data frame contains the following columns:

**GORE** Number of votes for Gore

**BUSH** Number of votes for Bush.

**BUCHANAN** Number of votes for Buchanan.

**NADER** Number of votes for Nader.

**BROWNE** Number of votes for Browne (whoever that is).

**HAGELIN** Number of votes for Hagelin (whoever that is).

**HARRIS** Number of votes for Harris (whoever that is).

**MCREYNOLDS** Number of votes for McReynolds (whoever that is).

**MOOREHEAD** Number of votes for Moorehead (whoever that is).

**PHILLIPS** Number of votes for Phillips (whoever that is).

**Total** Total number of votes.

### **Source**

Adams, G. D. and Fastnow, C. F. (2000) A note on the voting irregularities in Palm Beach, FL. Formerly at '<http://madison.hss.cmu.edu/>', but no longer available there.

---

Freedman

*Crowding and Crime in U. S. Metropolitan Areas*

---

**Description**

The Freedman data frame has 110 rows and 4 columns. The observations are U. S. metropolitan areas with 1968 populations of 250,000 or more. There are some missing data.

**Usage**

Freedman

**Format**

This data frame contains the following columns:

**population** Total 1968 population, 1000s.

**nonwhite** Percent nonwhite population, 1960.

**density** Population per square mile, 1968.

**crime** Crime rate per 100,000, 1969.

**Source**

United States (1970) *Statistical Abstract of the United States*. Bureau of the Census.

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Freedman, J. (1975) *Crowding and Behavior*. Viking.

---

Friendly

*Format Effects on Recall*

---

**Description**

The Friendly data frame has 30 rows and 2 columns. The data are from an experiment on subjects' ability to remember words based on the presentation format.

**Usage**

Friendly

**Format**

This data frame contains the following columns:

**condition** A factor with levels: Before, Recalled words presented before others; Meshed, Recalled words meshed with others; SFR, Standard free recall.

**correct** Number of words correctly recalled, out of 40 on final trial of the experiment.

**Source**

Friendly, M. and Franklin, P. (1980) Interactive presentation in multitrial free recall. *Memory and Cognition* **8** 265–270 [Personal communication from M. Friendly].

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

Ginzberg

*Data on Depression*

---

**Description**

The Ginzberg data frame has 82 rows and 6 columns. The data are for psychiatric patients hospitalized for depression.

**Usage**

Ginzberg

**Format**

This data frame contains the following columns:

**simplicity** Measures subject's need to see the world in black and white.

**fatalism** Fatalism scale.

**depression** Beck self-report depression scale.

**adjsimp** Adjusted Simplicity: Simplicity adjusted (by regression) for other variables thought to influence depression.

**adjfatal** Adjusted Fatalism.

**adjdep** Adjusted Depression.

**Source**

Personal communication from Georges Monette, Department of Mathematics and Statistics, York University, with the permission of the original investigator.



## References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

Greene

*Refugee Appeals*

---

## Description

The Greene data frame has 384 rows and 7 columns. These are cases filed in 1990, in which refugee claimants rejected by the Canadian Immigration and Refugee Board asked the Federal Court of Appeal for leave to appeal the negative ruling of the Board.

## Usage

Greene

## Format

This data frame contains the following columns:

**judge** Name of judge hearing case. A factor with levels: Desjardins, Heald, Hugessen, Iacobucci, MacGuigan, Mahoney, Marceau, Pratte, Stone, Urie.

**nation** Nation of origin of claimant. A factor with levels: Argentina, Bulgaria, China, Czechoslovakia, El. Salvador, Fiji, Ghana, Guatemala, India, Iran, Lebanon, Nicaragua, Nigeria, Pakistan, Poland, Somalia, Sri.Lanka.

**rater** Judgment of independent rater. A factor with levels: no, case has no merit; yes, case has some merit (leave to appeal should be granted).

**decision** Judge's decision. A factor with levels: no, leave to appeal not granted; yes, leave to appeal granted.

**language** Language of case. A factor with levels: English, French.

**location** Location of original refugee claim. A factor with levels: Montreal, other, Toronto.

**success** Logit of success rate, for all cases from the applicant's nation.

## Source

Personal communication from Ian Greene, Department of Political Science, York University.

## References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

Guyer

*Anonymity and Cooperation*

---

### Description

The Guyer data frame has 20 rows and 3 columns. The data are from an experiment in which four-person groups played a prisoner's dilemma game for 30 trials, each person making either a cooperative or competitive choice on each trial. Choices were made either anonymously or in public; groups were composed either of females or of males. The observations are 20 groups.

### Usage

Guyer

### Format

This data frame contains the following columns:

**cooperation** Number of cooperative choices (out of 120 in all).

**condition** A factor with levels: A, Anonymous; P, Public-Choice.

**sex** Sex. A factor with levels: F, Female; M, Male.

### Source

Fox, J. and Guyer, M. (1978) Public choice and cooperation in n-person prisoner's dilemma. *Journal of Conflict Resolution* **22**, 469–481.

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

Hartnagel

*Canadian Crime-Rates Time Series*

---

### Description

The Hartnagel data frame has 38 rows and 7 columns. The data are an annual time-series from 1931 to 1968. There are some missing data.

### Usage

Hartnagel

**Format**

This data frame contains the following columns:

**year** 1931–1968.

**tfr** Total fertility rate per 1000 women.

**partic** Women’s labor-force participation rate per 1000.

**degrees** Women’s post-secondary degree rate per 10,000.

**fconvict** Female indictable-offense conviction rate per 100,000.

**fttheft** Female theft conviction rate per 100,000.

**mconvict** Male indictable-offense conviction rate per 100,000.

**mttheft** Male theft conviction rate per 100,000.

**Details**

The post-1948 crime rates have been adjusted to account for a difference in method of recording. Some of your results will differ in the last decimal place from those in Table 14.1 of Fox (1997) due to rounding of the data. Missing values for 1950 were interpolated.

**Source**

Personal communication from T. Hartnagel, Department of Sociology, University of Alberta.

**References**

Fox, J., and Hartnagel, T. F (1979) Changing social roles and female crime in Canada: A time series analysis. *Canadian Review of Sociology and Anthropology*, **16**, 96–104.

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

hccm

*Heteroscedasticity-Corrected Covariance Matrices*

---

**Description**

Calculates heteroscedasticity-corrected covariance matrices linear models fit by least squares or weighted least squares. These are also called “White-corrected” or “White-Huber” covariance matrices.

**Usage**

```
hccm(model, ...)

## S3 method for class 'lm'
hccm(model, type=c("hc3", "hc0", "hc1", "hc2", "hc4"),
singular.ok=TRUE, ...)
```

## Default S3 method:

```
hccm(model, ...)
```

**Arguments**

model	a unweighted or weighted linear model, produced by <code>lm</code> .
type	one of "hc0", "hc1", "hc2", "hc3", or "hc4"; the first of these gives the classic White correction. The "hc1", "hc2", and "hc3" corrections are described in Long and Ervin (2000); "hc4" is described in Cribari-Neto (2004).
singular.ok	if FALSE (the default is TRUE), a model with aliased coefficients produces an error; otherwise, the aliased coefficients are ignored in the coefficient covariance matrix that's returned.
...	arguments to pass to <code>hccm.lm</code> .

**Details**

The classical White-corrected coefficient covariance matrix ("hc0") (for an unweighted model) is

$$V(b) = (X'X)^{-1}X'diag(e_i^2)X(X'X)^{-1}$$

where  $e_i^2$  are the squared residuals, and  $X$  is the model matrix. The other methods represent adjustments to this formula. If there are weights, these are incorporated in the corrected covariance matrix.

The function `hccm.default` simply catches non-`lm` objects.

**Value**

The heteroscedasticity-corrected covariance matrix for the model.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Cribari-Neto, F. (2004) Asymptotic inference under heteroskedasticity of unknown form. *Computational Statistics and Data Analysis* **45**, 215–233.
- Long, J. S. and Ervin, L. H. (2000) Using heteroscedasity consistent standard errors in the linear regression model. *The American Statistician* **54**, 217–224.
- White, H. (1980) A heteroskedastic consistent covariance matrix estimator and a direct test of heteroskedasticity. *Econometrica* **48**, 817–838.

**Examples**

```
options(digits=4)
mod<-lm(interlocks~assets+nation, data=Ornstein)
vcov(mod)
##           (Intercept)      assets nation0TH  nationUK  nationUS
## (Intercept)  1.079e+00 -1.588e-05 -1.037e+00 -1.057e+00 -1.032e+00
```

```

## assets      -1.588e-05  1.642e-09  1.155e-05  1.362e-05  1.109e-05
## nation0TH   -1.037e+00  1.155e-05  7.019e+00  1.021e+00  1.003e+00
## nationUK    -1.057e+00  1.362e-05  1.021e+00  7.405e+00  1.017e+00
## nationUS    -1.032e+00  1.109e-05  1.003e+00  1.017e+00  2.128e+00
hccm(mod)
##            (Intercept)      assets  nation0TH  nationUK  nationUS
## (Intercept)  1.664e+00 -3.957e-05 -1.569e+00 -1.611e+00 -1.572e+00
## assets      -3.957e-05  6.752e-09  2.275e-05  3.051e-05  2.231e-05
## nation0TH   -1.569e+00  2.275e-05  8.209e+00  1.539e+00  1.520e+00
## nationUK    -1.611e+00  3.051e-05  1.539e+00  4.476e+00  1.543e+00
## nationUS    -1.572e+00  2.231e-05  1.520e+00  1.543e+00  1.946e+00

```

---

Highway1

*Highway Accidents*


---

### Description

The data comes from a unpublished master's paper by Carl Hoffstedt. They relate the automobile accident rate, in accidents per million vehicle miles to several potential terms. The data include 39 sections of large highways in the state of Minnesota in 1973. The goal of this analysis was to understand the impact of design variables, Acpts, Slim, Sig, and Shld that are under the control of the highway department, on accidents.

### Usage

Highway1

### Format

This data frame contains the following columns:

**rate** 1973 accident rate per million vehicle miles

**len** length of the Highway1 segment in miles

**adt** average daily traffic count in thousands

**trks** truck volume as a percent of the total volume

**sigs1** (number of signalized interchanges per mile times len + 1)/len, the number of signals per mile of roadway, adjusted to have no zero values.

**slim** speed limit in 1973

**shld** width in feet of outer shoulder on the roadway

**lane** total number of lanes of traffic

**acpt** number of access points per mile

**itg** number of freeway-type interchanges per mile

**lwid** lane width, in feet

**htype** An indicator of the type of roadway or the source of funding for the road, either MC, FAI, PA, or MA

## Source

Carl Hoffstedt. This differs from the dataset Highway in the alr4 package only by addition of transformation of some of the columns.

## References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley, Section 7.2.

---

 hist.boot

---

*Methods Functions to Support boot Objects*


---

## Description

The Boot function in the **car** package uses the [boot](#) function from the **boot** package to do a straight-forward case or residual bootstrap for a least-squares regression object. These are method functions for standard generics to summarize the results of the bootstrap.

## Usage

```
## S3 method for class 'boot'
hist(x, parm, layout = NULL, ask, main = "", freq = FALSE,
     estPoint = TRUE, point.col = "black", point.lty = 2, point.lwd = 2,
     estDensity = !freq, den.col = "blue", den.lty = 1, den.lwd = 2,
     estNormal = !freq, nor.col = "red", nor.lty = 2, nor.lwd = 2,
     ci = c("bca", "none", "perc"), level = 0.95, legend = c("top",
     "none", "separate"), box = TRUE, ...)

## S3 method for class 'boot'
summary(object, parm, high.moments = FALSE, extremes = FALSE, ...)

## S3 method for class 'boot'
confint(object, parm, level = 0.95, type = c("bca", "norm",
     "basic", "perc", "all"), ...)

## S3 method for class 'boot'
vcov(object, ...)
```

## Arguments

x, object	An object created by a call to boot of class "boot".
parm	A vector of numbers or coefficient names giving the coefficients for which a histogram or confidence interval is desired. If numbers are used, 1 corresponds to the intercept, if any. The default is all coefficients.

layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
ask	If TRUE, ask the user before drawing the next plot; if FALSE, don't ask.
main	Main title for the graphs. The default is <code>main=""</code> for no title.
freq	The usual default for <code>hist</code> is <code>freq=TRUE</code> to give a frequency histogram. The default here is <code>freq=FALSE</code> to give a density histogram. A density estimate and/or a fitted normal density can be added to the graph if <code>freq=FALSE</code> but not if <code>freq=TRUE</code> .
estPoint, point.col, point.lty, point.lwd	If <code>estPoint=TRUE</code> , the default, a vertical line is drawn on the histogram at the value of the point estimate computed from the complete data. The remaining three optional arguments set the color, line type and line width of the line that is drawn.
estDensity, den.col, den.lty, den.lwd	If <code>estDensity=TRUE</code> and <code>freq=FALSE</code> , the default, a kernel density estimate is drawn on the plot with a call to the density function with no additional arguments. The remaining three optional arguments set the color, line type and line width of the lines that are drawn.
estNormal, nor.col, nor.lty, nor.lwd	If <code>estNormal=TRUE</code> and <code>freq=FALSE</code> , the default, a normal density with mean and sd computed from the data is drawn on the plot. The remaining three optional arguments set the color, line type and line width of the lines that are drawn.
ci	A confidence interval based on the bootstrap will be added to the histogram using the BCa method if <code>ci="bca"</code> or using the percentile method if <code>ci="perc"</code> . No interval is drawn if <code>ci="none"</code> . The default is <code>"bca"</code> . The interval is indicated by a thick horizontal line at $y=0$ . For some bootstraps the BCa method is unavailable, in which case a warning is issued and <code>ci="perc"</code> is substituted.
legend	A legend can be added to the (array of) histograms. The value <code>"top"</code> puts at the top-left of the plots. The value <code>"separate"</code> puts the legend in its own graph following all the histograms. The value <code>"none"</code> suppresses the legend.
box	Add a box around each histogram.
...	Additional arguments passed to <code>hist</code> ; for other methods this is included for compatibility with the generic method. For example, the argument <code>border=par()\$bg</code> in <code>hist</code> will draw the histogram transparently, leaving only the density estimates.
high.moments	Should the skewness and kurtosis be included in the summary? Default is FALSE.
extremes	Should the minimum, maximum and range be included in the summary? Default is FALSE.
level	Confidence level, a number between 0 and 1. In <code>confint</code> , <code>level</code> can be a vector; for example <code>level=c(.68, .90, .95)</code> will return the estimated quantiles at <code>c(.025, .05, .16, .84, .95, .975)</code> .

**type** Selects the confidence interval type. The types implemented are the "percentile" method, which uses the function `quantile` to return the appropriate quantiles for the confidence limit specified, the default `bca` which uses the bias-corrected and accelerated method presented by Efron and Tibshirani (1993, Chapter 14). For the other types, see the documentation for `boot`.

### Value

`hist` is used for the side-effect of drawing an array of histograms of each column of the first argument. `summary` returns a matrix of summary statistics for each of the columns in the bootstrap object. The `confint` method returns confidence intervals. The `vcov` returns the sample covariance of the bootstrap sample estimates.

### Author(s)

Sanford Weisberg, <sandy@umn.edu>

### References

Efron, B. and Tibshirani, R. (1993) *An Introduction to the Bootstrap*. New York: Chapman and Hall.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2012) *Bootstrapping*, <http://socserv.mcmaster.ca/jfox/Books/Companion/appendix/Appendix-Bootstrapping.pdf>.

Weisberg, S. (2013) *Applied Linear Regression*, Fourth Edition, Wiley

### See Also

See Also `Boot`, `hist`, `density`

### Examples

```
m1 <- lm(Fertility ~ ., swiss)
betahat.boot <- Boot(m1, R=99) # 99 bootstrap samples--too small to be useful
summary(betahat.boot) # default summary
confint(betahat.boot)
hist(betahat.boot)
```

---

infIndexPlot

*Influence Index Plot*

---

### Description

Provides index plots of Cook's distances, leverages, Studentized residuals, and outlier significance levels for a regression object.



**Usage**

```

infIndexPlot(model, ...)

influenceIndexPlot(model, ...)

## S3 method for class 'lm'
infIndexPlot(model,
  vars=c("Cook", "Studentized", "Bonf", "hat"),
  main="Diagnostic Plots",
  labels, id.method = "y",
  id.n = if(id.method[1]=="identify") Inf else 0,
  id.cex=1, id.col=palette()[1], id.location="lr", grid=TRUE, ...)

```

**Arguments**

model	A regression object of class lm or glm.
vars	All the quantities listed in this argument are plotted. Use "Cook" for Cook's distances, "Studentized" for Studentized residuals, "Bonf" for Bonferroni p-values for an outlier test, and "hat" for hat-values (or leverages). Capitalization is optional. All may be abbreviated by the first one or more letters.
main	main title for graph
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is id.n=0 for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
grid	If TRUE, the default, a light-gray background grid is put on the graph
...	Arguments passed to plot

**Value**

Used for its side effect of producing a graph. Produces four index plots of Cook's distance, Studentized Residuals, the corresponding Bonferroni p-values for outlier tests, and leverages.

**Author(s)**

Sanford Weisberg, <sandy@umn.edu>

**References**

- Cook, R. D. and Weisberg, S. (1999) *Applied Regression, Including Computing and Graphics*. Wiley.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

**See Also**

[cooks.distance](#), [rstudent](#), [outlierTest](#), [hatvalues](#)

**Examples**

```
m1 <- lm(prestige ~ income + education + type, Duncan)
influenceIndexPlot(m1)
```

---

influencePlot

*Regression Influence Plot*

---

**Description**

This function creates a “bubble” plot of Studentized residuals by hat values, with the areas of the circles representing the observations proportional to Cook’s distances. Vertical reference lines are drawn at twice and three times the average hat value, horizontal reference lines at -2, 0, and 2 on the Studentized-residual scale.

**Usage**

```
influencePlot(model, ...)

## S3 method for class 'lm'
influencePlot(model, scale=10,
  xlab="Hat-Values", ylab="Studentized Residuals",
  labels, id.method = "noteworthy",
  id.n = if(id.method[1]=="identify") Inf else 0,
  id.cex=1, id.col=palette()[1], id.location="lr", ...)
```

**Arguments**

**model** a linear or generalized-linear model.

**scale** a factor to adjust the size of the circles.

**xlab, ylab** axis labels.

**labels, id.method, id.n, id.cex, id.col, id.location** settings for labelling points; see `link{showLabels}` for details. To omit point labelling, set `id.n=0`, the default. The default `id.method="noteworthy"` is used only in this function and indicates setting labels for points with large Studentized residuals, hat-values or Cook’s distances. Set `id.method="identify"` for interactive point identification.

**...** arguments to pass to the plot and points functions.

**Value**

If points are identified, returns a data frame with the hat values, Studentized residuals and Cook’s distance of the identified points. If no points are identified, nothing is returned. This function is primarily used for its side-effect of drawing a plot.

**Author(s)**

John Fox <jfox@mcmaster.ca>, minor changes by S. Weisberg <sandy@umn.edu>

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[cooks.distance](#), [rstudent](#), [hatvalues](#), [showLabels](#)

**Examples**

```
influencePlot(lm(prestige ~ income + education, data=Duncan))
```

---

 invResPlot

---

*Inverse Response Plots to Transform the Response*


---

**Description**

For a `lm` model, draws an inverse.response plot with the response  $Y$  on the vertical axis and the fitted values  $\hat{Y}$  on the horizontal axis. Uses `nls` to estimate  $\lambda$  in the function  $\hat{Y} = b_0 + b_1 Y^\lambda$ . Adds the fitted curve to the plot. `invResPlot` is an alias for `inverseResponsePlot`.

**Usage**

```
inverseResponsePlot(model, lambda=c(-1,0,1), robust=FALSE, xlab=NULL, ...)
```

```
## S3 method for class 'lm'
```

```
inverseResponsePlot(model, lambda=c(-1,0,1), robust=FALSE,  
  xlab=NULL, labels=names(residuals(model)), ...)
```

```
invResPlot(model, ...)
```

**Arguments**

<code>model</code>	A <code>lm</code> regression object
<code>lambda</code>	A vector of values for <code>lambda</code> . A plot will be produced with curves corresponding to these <code>lambda</code> s and to the least squares estimate of <code>lambda</code>
<code>xlab</code>	The horizontal axis label. If <code>NULL</code> , it is constructed by the function.
<code>labels</code>	Case labels if labeling is turned on; see <a href="#">invTranPlot</a> and <a href="#">showLabels</a> for arguments.
<code>robust</code>	If <code>TRUE</code> , then estimation uses Huber M-estimates with the median absolute deviation to estimate scale and <code>k=1.345</code> . The default is <code>FALSE</code> .
<code>...</code>	Other arguments passed to <code>invTranPlot</code> and then to <code>plot</code> .

**Value**

As a side effect, a plot is produced with the response on the horizontal axis and fitted values on the vertical axis. Several lines are added to be plot as the ols estimates of the regression of  $\hat{Y}$  on  $Y^\lambda$ , interpreting  $\lambda = 0$  to be natural logarithms.

Numeric output is a list with elements

lambda	Estimate of transformation parameter for the response
RSS	The residual sum of squares at the minimum if robust=FALSE. If robust = TRUE, the value of Huber objective function is returned.

**Author(s)**

Sanford Weisberg, sandy@umn.edu

**References**

- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Pendergast, L, and Sheather, S. (in press). On sensitivity of response plot estimation of a robust estimation approach. *Scandinavian Journal of Statistics*.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley, Chapter 7.

**See Also**

[invTranPlot](#), [powerTransform](#), [showLabels](#)

**Examples**

```
m2 <- lm(rate ~ log(len) + log(adl) + slim + shld + log(sigs1), Highway1)
invResPlot(m2)
```

---

invTranPlot

*Choose a Predictor Transformation Visually or Numerically*

---

**Description**

invTranPlot draws a two-dimensional scatterplot of  $Y$  versus  $X$ , along with the OLS fit from the regression of  $Y$  on  $(X^\lambda - 1)/\lambda$ . invTranEstimate finds the nonlinear least squares estimate of  $\lambda$  and its standard error.

**Usage**

```

invTranPlot(x, ...)

## S3 method for class 'formula'
invTranPlot(x, data, subset, na.action, ...)

## Default S3 method:
invTranPlot(x, y, lambda=c(-1, 0, 1), robust=FALSE,
            lty.lines=rep(c("solid", "dashed", "dotdash", "longdash", "twodash"),
                          length=1 + length(lambda)), lwd.lines=2,
            col=palette()[1], col.lines=palette(),
            xlab=deparse(substitute(x)), ylab=deparse(substitute(y)),
            family="bcPower", optimal=TRUE, key="auto",
            id.method = "x",
            labels,
            id.n = if(id.method[1]=="identify") Inf else 0,
            id.cex=1, id.col=palette()[1], id.location="lr", grid=TRUE, ...)

invTranEstimate(x, y, family="bcPower", confidence=0.95, robust=FALSE)

```

**Arguments**

x	The predictor variable, or a formula with a single response and a single predictor
y	The response variable
data	An optional data frame to get the data for the formula
subset	Optional, as in <code>lm</code> , select a subset of the cases
na.action	Optional, as in <code>lm</code> , the action for missing data
lambda	The powers used in the plot. The optimal power that minimizes the residual sum of squares is always added unless <code>optimal</code> is <code>FALSE</code> .
robust	If <code>TRUE</code> , then the estimated transformation is computed using Huber M-estimation with the MAD used to estimate scale and $k=1.345$ . The default is <code>FALSE</code> .
family	The transformation family to use, "bcPower", "yjPower", or a user-defined family.
confidence	returns a profile likelihood confidence interval for the optimal transformation with this confidence level. If <code>FALSE</code> , or if <code>robust=TRUE</code> , no interval is returned.
optimal	Include the optimal value of <code>lambda</code> ?
lty.lines	line types corresponding to the powers
lwd.lines	the width of the plotted lines, defaults to 2 times the standard
col	color(s) of the points in the plot. If you wish to distinguish points according to the levels of a factor, we recommend using symbols, specified with the <code>pch</code> argument, rather than colors.
col.lines	color of the fitted lines corresponding to the powers. The default is to use the colors returned by <code>palette</code>

key	The default is "auto", in which case a legend is added to the plot, either above the top margin or in the bottom right or top right corner. Set to NULL to suppress the legend.
xlab	Label for the horizontal axis.
ylab	Label for the vertical axis.
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is id.n=0 for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
...	Additional arguments passed to the plot method, such as pch.
grid	If TRUE, the default, a light-gray background grid is put on the graph

### Value

invTranPlot plots a graph and returns a data frame with  $\lambda$  in the first column, and the residual sum of squares from the regression for that  $\lambda$  in the second column.

invTranEstimate returns a list with elements lambda for the estimate, se for its standard error, and RSS, the minimum value of the residual sum of squares.

### Author(s)

Sanford Weisberg, <sandy@umn.edu>

### References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Pendergast, L, and Sheather, S. (in press). On sensitivity of response plot estimation of a robust estimation approach. *Scandinavian Journal of Statistics*.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley Wiley.

### See Also

[inverseResponsePlot](#), [optimize](#)

### Examples

```
with(UN, invTranPlot(gdp, infant.mortality))
with(UN, invTranEstimate(gdp, infant.mortality))
```

---

KosteckiDillon

*Treatment of Migraine Headaches*

---

### Description

Subset of data on migraine treatments collected by Tammy Kostecki-Dillon.

### Usage

KosteckiDillon

### Format

A data frame with 4152 observations on 133 subjects for the following 9 variables.

`id` Patient id.

`time` time in days relative to the onset of treatment, which occurs at time 0.

`dos` time in days from the start of the study, January 1 of the first year of the study.

`hatype` a factor with levels Aura Mixed No Aura, the type of migraine experienced by a subject.

`age` at onset of treatment, in years.

`airq` a measure of air quality.

`medication` a factor with levels none reduced continuing, representing subjects who discontinued their medication, who continued but at a reduced dose, or who continued at the previous dose.

`headache` a factor with levels no yes.

`sex` a factor with levels female male.

### Details

The data consist of headache logs kept by 133 patients in a treatment program in which bio-feedback was used to attempt to reduce migraine frequency and severity. Patients entered the program at different times over a period of about 3 years. Patients were encouraged to begin their logs four weeks before the onset of treatment and to continue for one month afterwards, but only 55 patients have data preceding the onset of treatment.

### Source

Personal communication from Georges Monette (and adapted from his description of the data).

### References

Kostecki-Dillon, T., Monette, G., and Wong, P. (1999). Pine trees, comas, and migraines. *York University Institute for Social Research Newsletter*, 14:2.

### Examples

```
summary(KosteckiDillon)
```

---

Leinhardt

*Data on Infant-Mortality*

---

### Description

The Leinhardt data frame has 105 rows and 4 columns. The observations are nations of the world around 1970.

### Usage

Leinhardt

### Format

This data frame contains the following columns:

**income** Per-capita income in U. S. dollars.

**infant** Infant-mortality rate per 1000 live births.

**region** A factor with levels: Africa; Americas; Asia, Asia and Oceania; Europe.

**oil** Oil-exporting country. A factor with levels: no, yes.

### Details

The infant-mortality rate for Jamaica is misprinted in Leinhardt and Wasserman; the correct value is given here. Some of the values given in Leinhardt and Wasserman do not appear in the original New York Times table and are of dubious validity.

### Source

Leinhardt, S. and Wasserman, S. S. (1979) Exploratory data analysis: An introduction to selected methods. In Schuessler, K. (Ed.) *Sociological Methodology 1979* Jossey-Bass.

*The New York Times*, 28 September 1975, p. E-3, Table 3.

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.



---

leveneTest	<i>Levene's Test</i>
------------	----------------------

---

**Description**

Computes Levene's test for homogeneity of variance across groups.

**Usage**

```
leveneTest(y, ...)
## S3 method for class 'formula'
leveneTest(y, data, ...)
## S3 method for class 'lm'
leveneTest(y, ...)
## Default S3 method:
leveneTest(y, group, center=median, ...)
```

**Arguments**

y	response variable for the default method, or a lm or formula object. If y is a linear-model object or a formula, the variables on the right-hand-side of the model must all be factors and must be completely crossed.
group	factor defining groups.
center	The name of a function to compute the center of each group; mean gives the original Levene's test; the default, median, provides a more robust test.
data	a data frame for evaluating the formula.
...	arguments to be passed down, e.g., data for the formula and lm methods; can also be used to pass arguments to the function given by center (e.g., center=mean and trim=0.1 specify the 10% trimmed mean).

**Value**

returns an object meant to be printed showing the results of the test.

**Note**

adapted from a response posted by Brian Ripley to the r-help email list.

**Author(s)**

John Fox <jfox@mcmaster.ca>; original generic version contributed by Derek Ogle

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.  
 Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Examples**

```

with(Moore, leveneTest(conformity, fcategory))
with(Moore, leveneTest(conformity, interaction(fcategory, partner.status)))
leveneTest(conformity ~ fcategory*partner.status, data=Moore)
leveneTest(lm(conformity ~ fcategory*partner.status, data=Moore))
leveneTest(conformity ~ fcategory*partner.status, data=Moore, center=mean)
leveneTest(conformity ~ fcategory*partner.status, data=Moore, center=mean, trim=0.1)

```

---

leveragePlots

*Regression Leverage Plots*


---

**Description**

These functions display a generalization, due to Sall (1990) and Cook and Weisberg (1991), of added-variable plots to multiple-df terms in a linear model. When a term has just 1 df, the leverage plot is a rescaled version of the usual added-variable (partial-regression) plot.

**Usage**

```

leveragePlots(model, terms = ~., layout = NULL, ask,
              main, ...)

leveragePlot(model, ...)

## S3 method for class 'lm'
leveragePlot(model, term.name,
             id.method = list(abs(residuals(model, type="pearson")), "x"),
             labels,
             id.n = if(id.method[1]=="identify") Inf else 0,
             id.cex=1, id.col=palette()[1], id.location="lr",
             col=palette()[1], col.lines=palette()[2], lwd=2,
             xlab, ylab, main="Leverage Plot", grid=TRUE, ...)

## S3 method for class 'glm'
leveragePlot(model, ...)

```

**Arguments**

model	model object produced by lm
terms	A one-sided formula that specifies a subset of the predictors. One added-variable plot is drawn for each term. The default <code>~.</code> is to plot against all numeric predictors. For example, the specification <code>terms = ~ . - X3</code> would plot against all predictors except for X3. If this argument is a quoted name of one of the predictors, the added-variable plot is drawn for that predictor only.

layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
ask	if TRUE, a menu is provided in the R Console for the user to select the term(s) to plot.
xlab, ylab	axis labels; if missing, labels will be supplied.
main	title for plot; if missing, a title will be supplied.
...	arguments passed down to method functions.
term.name	Quoted name of term in the model to be plotted; this argument is omitted for <code>leveragePlots</code> .
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
col	color(s) of points
col.lines	color of the fitted line
lwd	line width; default is 2 (see <a href="#">par</a> ).
grid	If TRUE, the default, a light-gray background grid is put on the graph

### Details

The function intended for direct use is `leveragePlots`.

The model can contain factors and interactions. A leverage plot can be drawn for each term in the model, including the constant.

`leveragePlot.glm` is a dummy function, which generates an error message.

### Value

NULL. These functions are used for their side effect: producing plots.

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

- Cook, R. D. and Weisberg, S. (1991). Added Variable Plots in Linear Regression. In Stahel, W. and Weisberg, S. (eds.), *Directions in Robust Statistics and Diagnostics*. Springer, 47-60.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Sall, J. (1990) Leverage plots for general linear hypotheses. *American Statistician* **44**, 308–315.

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

```
leveragePlots(lm(prestige~(income+education)*type, data=Duncan))
```

---

linearHypothesis	<i>Test Linear Hypothesis</i>
------------------	-------------------------------

---

**Description**

Generic function for testing a linear hypothesis, and methods for linear models, generalized linear models, multivariate linear models, linear and generalized linear mixed-effects models, generalized linear models fit with `svyglm` in the **survey** package, robust linear models fit with `r1m` in the **MASS** package, and other models that have methods for `coef` and `vcov`. For mixed-effects models, the tests are Wald chi-square tests for the fixed effects.

**Usage**

```
linearHypothesis(model, ...)

lht(model, ...)

## Default S3 method:
linearHypothesis(model, hypothesis.matrix, rhs=NULL,
  test=c("Chisq", "F"), vcov.=NULL, singular.ok=FALSE, verbose=FALSE,
  coef. = coef(model), ...)

## S3 method for class 'lm'
linearHypothesis(model, hypothesis.matrix, rhs=NULL,
  test=c("F", "Chisq"), vcov.=NULL,
  white.adjust=c(FALSE, TRUE, "hc3", "hc0", "hc1", "hc2", "hc4"),
  singular.ok=FALSE, ...)

## S3 method for class 'glm'
linearHypothesis(model, ...)

## S3 method for class 'nlsList'
linearHypothesis(model, ..., vcov., coef.)

## S3 method for class 'mlm'
linearHypothesis(model, hypothesis.matrix, rhs=NULL, SSPE, V,
  test, idata, icontrasts=c("contr.sum", "contr.poly"), idesign, itterms,
  check.imatrix=TRUE, P=NULL, title="", singular.ok=FALSE, verbose=FALSE, ...)

## S3 method for class 'polr'
```

```

linearHypothesis(model, hypothesis.matrix, rhs=NULL, vcov.,
verbose=FALSE, ...)

## S3 method for class 'linearHypothesis.mlm'
print(x, SSP=TRUE, SSPE=SSP,
      digits=getOption("digits"), ...)

## S3 method for class 'lme'
linearHypothesis(model, hypothesis.matrix, rhs=NULL,
vcov.=NULL, singular.ok=FALSE, verbose=FALSE, ...)

## S3 method for class 'mer'
linearHypothesis(model, hypothesis.matrix, rhs=NULL,
vcov.=NULL, test=c("Chisq", "F"), singular.ok=FALSE, verbose=FALSE, ...)

## S3 method for class 'merMod'
linearHypothesis(model, hypothesis.matrix, rhs=NULL,
      vcov.=NULL, test=c("Chisq", "F"), singular.ok=FALSE, verbose=FALSE, ...)

## S3 method for class 'svyglm'
linearHypothesis(model, ...)

## S3 method for class 'rlm'
linearHypothesis(model, ...)

matchCoefs(model, pattern, ...)

## Default S3 method:
matchCoefs(model, pattern, coef.=coef, ...)

## S3 method for class 'lme'
matchCoefs(model, pattern, ...)

## S3 method for class 'mer'
matchCoefs(model, pattern, ...)

## S3 method for class 'merMod'
matchCoefs(model, pattern, ...)

## S3 method for class 'mlm'
matchCoefs(model, pattern, ...)

```

## Arguments

`model` fitted model object. The default method of `linearHypothesis` works for models for which the estimated parameters can be retrieved by `coef` and the corresponding estimated covariance matrix by `vcov`. See the *Details* for more infor-

	mation.
<code>hypothesis.matrix</code>	matrix (or vector) giving linear combinations of coefficients by rows, or a character vector giving the hypothesis in symbolic form (see <i>Details</i> ).
<code>rhs</code>	right-hand-side vector for hypothesis, with as many entries as rows in the hypothesis matrix; can be omitted, in which case it defaults to a vector of zeroes. For a multivariate linear model, <code>rhs</code> is a matrix, defaulting to 0.
<code>singular.ok</code>	if FALSE (the default), a model with aliased coefficients produces an error; if TRUE, the aliased coefficients are ignored, and the hypothesis matrix should not have columns for them. For a multivariate linear model: will return the hypothesis and error SSP matrices even if the latter is singular; useful for computing univariate repeated-measures ANOVAs where there are fewer subjects than <code>df</code> for within-subject effects.
<code>idata</code>	an optional data frame giving a factor or factors defining the intra-subject model for multivariate repeated-measures data. See <i>Details</i> for an explanation of the intra-subject design and for further explanation of the other arguments relating to intra-subject factors.
<code>icontrasts</code>	names of contrast-generating functions to be applied by default to factors and ordered factors, respectively, in the within-subject “data”; the contrasts must produce an intra-subject model matrix in which different terms are orthogonal.
<code>idesign</code>	a one-sided model formula using the “data” in <code>idata</code> and specifying the intra-subject design.
<code>iterms</code>	the quoted name of a term, or a vector of quoted names of terms, in the intra-subject design to be tested.
<code>check.imatrix</code>	check that columns of the intra-subject model matrix for different terms are mutually orthogonal (default, TRUE). Set to FALSE only if you have <i>already</i> checked that the intra-subject model matrix is block-orthogonal.
<code>P</code>	transformation matrix to be applied to the repeated measures in multivariate repeated-measures data; if NULL <i>and</i> no intra-subject model is specified, no response-transformation is applied; if an intra-subject model is specified via the <code>idata</code> , <code>idesign</code> , and (optionally) <code>icontrasts</code> arguments, then <code>P</code> is generated automatically from the <code>iterms</code> argument.
<code>SSPE</code>	in <code>linearHypothesis</code> method for <code>mlm</code> objects: optional error sum-of-squares-and-products matrix; if missing, it is computed from the model. In <code>print</code> method for <code>linearHypothesis.mlm</code> objects: if TRUE, print the sum-of-squares and cross-products matrix for error.
<code>test</code>	character string, “F” or “Chisq”, specifying whether to compute the finite-sample F statistic (with approximate F distribution) or the large-sample Chi-squared statistic (with asymptotic Chi-squared distribution). For a multivariate linear model, the multivariate test statistic to report — one or more of “Pillai”, “Wilks”, “Hotelling-Lawley”, or “Roy”, with “Pillai” as the default.
<code>title</code>	an optional character string to label the output.
<code>V</code>	inverse of sum of squares and products of the model matrix; if missing it is computed from the model.

<code>vcov.</code>	a function for estimating the covariance matrix of the regression coefficients, e.g., <code>hccm</code> , or an estimated covariance matrix for <code>model</code> . See also <code>white.adjust</code> .
<code>coef.</code>	a vector of coefficient estimates. The default is to get the coefficient estimates from the <code>model</code> argument, but the user can input any vector of the correct length.
<code>white.adjust</code>	logical or character. Convenience interface to <code>hccm</code> (instead of using the argument <code>vcov.</code> ). Can be set either to a character value specifying the type argument of <code>hccm</code> or <code>TRUE</code> , in which case "hc3" is used implicitly. The default is <code>FALSE</code> .
<code>verbose</code>	If <code>TRUE</code> , the hypothesis matrix, right-hand-side vector (or matrix), and estimated value of the hypothesis are printed to standard output; if <code>FALSE</code> (the default), the hypothesis is only printed in symbolic form and the value of the hypothesis is not printed.
<code>x</code>	an object produced by <code>linearHypothesis.lm</code> .
<code>SSP</code>	if <code>TRUE</code> (the default), print the sum-of-squares and cross-products matrix for the hypothesis and the response-transformation matrix.
<code>digits</code>	minimum number of significant digits to print.
<code>pattern</code>	a <a href="#">regular expression</a> to be matched against coefficient names.
<code>...</code>	arguments to pass down.

## Details

`linearHypothesis` computes either a finite-sample F statistic or asymptotic Chi-squared statistic for carrying out a Wald-test-based comparison between a model and a linearly restricted model. The default method will work with any model object for which the coefficient vector can be retrieved by `coef` and the coefficient-covariance matrix by `vcov` (otherwise the argument `vcov.` has to be set explicitly). For computing the F statistic (but not the Chi-squared statistic) a `df.residual` method needs to be available. If a `formula` method exists, it is used for pretty printing.

The method for "lm" objects calls the default method, but it changes the default test to "F", supports the convenience argument `white.adjust` (for backwards compatibility), and enhances the output by the residual sums of squares. For "glm" objects just the default method is called (bypassing the "lm" method). The `svyglm` method also calls the default method.

The function `lht` also dispatches to `linearHypothesis`.

The hypothesis matrix can be supplied as a numeric matrix (or vector), the rows of which specify linear combinations of the model coefficients, which are tested equal to the corresponding entries in the right-hand-side vector, which defaults to a vector of zeroes.

Alternatively, the hypothesis can be specified symbolically as a character vector with one or more elements, each of which gives either a linear combination of coefficients, or a linear equation in the coefficients (i.e., with both a left and right side separated by an equals sign). Components of a linear expression or linear equation can consist of numeric constants, or numeric constants multiplying coefficient names (in which case the number precedes the coefficient, and may be separated from it by spaces or an asterisk); constants of 1 or -1 may be omitted. Spaces are always optional. Components are separated by plus or minus signs. Newlines or tabs in hypotheses will be treated as spaces. See the examples below.

If the user sets the arguments `coef.` and `vcov.`, then the computations are done without reference to the `model` argument. This is like assuming that `coef.` is normally distributed with estimated

variance `vcov.` and the `linearHypothesis` will compute tests on the mean vector for `coef.`, without actually using the `model` argument.

A linear hypothesis for a multivariate linear model (i.e., an object of class `"mlm"`) can optionally include an intra-subject transformation matrix for a repeated-measures design. If the intra-subject transformation is absent (the default), the multivariate test concerns all of the corresponding coefficients for the response variables. There are two ways to specify the transformation matrix for the repeated measures:

1. The transformation matrix can be specified directly via the `P` argument.
2. A data frame can be provided defining the repeated-measures factor or factors via `idata`, with default contrasts given by the `icontrasts` argument. An intra-subject model-matrix is generated from the one-sided formula specified by the `idesign` argument; columns of the model matrix corresponding to different terms in the intra-subject model must be orthogonal (as is insured by the default contrasts). Note that the contrasts given in `icontrasts` can be overridden by assigning specific contrasts to the factors in `idata`. The repeated-measures transformation matrix consists of the columns of the intra-subject model matrix corresponding to the term or terms in `iterms`. In most instances, this will be the simpler approach, and indeed, most tests of interests can be generated automatically via the `Anova` function.

`matchCoefs` is a convenience function that can sometimes help in formulating hypotheses; for example `matchCoefs(mod, ":")` will return the names of all interaction coefficients in the model `mod`.

## Value

For a univariate model, an object of class `"anova"` which contains the residual degrees of freedom in the model, the difference in degrees of freedom, Wald statistic (either `"F"` or `"Chisq"`), and corresponding p value. The value of the linear hypothesis and its covariance matrix are returned respectively as `"value"` and `"vcov"` attributes of the object (but not printed).

For a multivariate linear model, an object of class `"linearHypothesis.mlm"`, which contains sums-of-squares-and-product matrices for the hypothesis and for error, degrees of freedom for the hypothesis and error, and some other information.

The returned object normally would be printed.

## Author(s)

Achim Zeileis and John Fox <jfox@mcmaster.ca>

## References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Hand, D. J., and Taylor, C. C. (1987) *Multivariate Analysis of Variance and Repeated Measures: A Practical Approach for Behavioural Scientists*. Chapman and Hall.
- O'Brien, R. G., and Kaiser, M. K. (1985) MANOVA method for analyzing repeated measures designs: An extensive primer. *Psychological Bulletin* **97**, 316–333.



**See Also**

[anova](#), [Anova](#), [waldtest](#), [hccm](#), [vcovHC](#), [vcovHAC](#), [coef](#), [vcov](#)

**Examples**

```

mod.davis <- lm(weight ~ repwt, data=Davis)

## the following are equivalent:
linearHypothesis(mod.davis, diag(2), c(0,1))
linearHypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"))
linearHypothesis(mod.davis, c("(Intercept)", "repwt"), c(0,1))
linearHypothesis(mod.davis, c("(Intercept)", "repwt = 1"))

## use asymptotic Chi-squared statistic
linearHypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"), test = "Chisq")

## the following are equivalent:
## use HC3 standard errors via white.adjust option
linearHypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"),
  white.adjust = TRUE)
## covariance matrix *function*
linearHypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"), vcov = hccm)
## covariance matrix *estimate*
linearHypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"),
  vcov = hccm(mod.davis, type = "hc3"))

mod.duncan <- lm(prestige ~ income + education, data=Duncan)

## the following are all equivalent:
linearHypothesis(mod.duncan, "1*income - 1*education = 0")
linearHypothesis(mod.duncan, "income = education")
linearHypothesis(mod.duncan, "income - education")
linearHypothesis(mod.duncan, "1income - 1education = 0")
linearHypothesis(mod.duncan, "0 = 1*income - 1*education")
linearHypothesis(mod.duncan, "income-education=0")
linearHypothesis(mod.duncan, "1*income - 1*education + 1 = 1")
linearHypothesis(mod.duncan, "2income = 2*education")

mod.duncan.2 <- lm(prestige ~ type*(income + education), data=Duncan)
coefs <- names(coef(mod.duncan.2))

## test against the null model (i.e., only the intercept is not set to 0)
linearHypothesis(mod.duncan.2, coefs[-1])

## test all interaction coefficients equal to 0
linearHypothesis(mod.duncan.2, coefs[grep(":", coefs)], verbose=TRUE)
linearHypothesis(mod.duncan.2, matchCoefs(mod.duncan.2, ":"), verbose=TRUE) # equivalent
lh <- linearHypothesis(mod.duncan.2, coefs[grep(":", coefs)])
attr(lh, "value") # value of linear function
attr(lh, "vcov") # covariance matrix of linear function

```

```

## a multivariate linear model for repeated-measures data
## see ?OBrienKaiser for a description of the data set used in this example.

mod.ok <- lm(cbind(pre.1, pre.2, pre.3, pre.4, pre.5,
                  post.1, post.2, post.3, post.4, post.5,
                  fup.1, fup.2, fup.3, fup.4, fup.5) ~ treatment*gender,
            data=OBrienKaiser)
coef(mod.ok)

## specify the model for the repeated measures:
phase <- factor(rep(c("pretest", "posttest", "followup"), c(5, 5, 5)),
               levels=c("pretest", "posttest", "followup"))
hour <- ordered(rep(1:5, 3))
idata <- data.frame(phase, hour)
idata

## test the four-way interaction among the between-subject factors
## treatment and gender, and the intra-subject factors
## phase and hour

linearHypothesis(mod.ok, c("treatment1:gender1", "treatment2:gender1"),
                title="treatment:gender:phase:hour", idata=idata, idesign=~phase*hour,
                iterm="phase:hour")

## mixed-effects models examples:

## Not run:
library(nlme)
example(lme)
linearHypothesis(fm2, "age = 0")

## End(Not run)

## Not run:
library(lme4)
example(glmer)
linearHypothesis(gm1, matchCoefs(gm1, "period"))

## End(Not run)

```

---

LoBD

*Cancer drug data use to provide an example of the use of the skew power distributions.*

---

## Description

A portion of an experiment to determine the limit of blank/limit of detection in a biochemical assay.

**Usage**

LoBD

**Format**

A data frame with 84 observations on the following 9 variables.

pool a factor with levels 1 2 3 4 5 6 7 8 9 10 11 12 denoting the 12 pools used in the experiment; each pool had a different level of drug.

I1L1 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I1L2 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I2L1 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I2L2 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I3L1 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I3L2 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I4L1 a numeric vector giving the measured concentration in pmol/L of drug in the assay

I4L2 a numeric vector giving the measured concentration in pmol/L of drug in the assay

**Details**

Important characteristics of a clinical chemistry assay are its limit of blank (LoB), and its limit of detection (LoD). The LoB, conceptually the highest reading likely to be obtained from a zero-concentration sample, is defined operationally by the upper 95% point of readings obtained from samples that do not contain the analyte. The LoD, conceptually the lowest level of analyte that can be reliably determined not to be blank, is defined operationally as true value at which there is a 95% chance of the reading being above the LoB.

These data are from a portion of a LoB/D study of an assay for a drug used to treat certain cancers. Twelve pools were used, four of them blanks of different types, and eight with successively increasing drug levels. The 8 columns of the data set refer to measurements made using different instruments I and reagent lots L.

**Source**

Used as an illustrative example for Box-Cox type transformations with negative readings in Hawkins and Weisberg (2015). For examples of its use, see [bcnPower](#).

**References**

Hawkins, D. and Weisberg, S. (2015) Combining the Box-Cox Power and Generalized Log Transformations to Accommodate Negative Responses, submitted for publication.

**Examples**

LoBD

---

logit *Logit Transformation*

---

### Description

Compute the logit transformation of proportions or percentages.

### Usage

```
logit(p, percents=range.p[2] > 1, adjust)
```

### Arguments

p	numeric vector or array of proportions or percentages.
percents	TRUE for percentages.
adjust	adjustment factor to avoid proportions of 0 or 1; defaults to 0 if there are no such proportions in the data, and to .025 if there are.

### Details

Computes the logit transformation  $\text{logit} = \log[p/(1 - p)]$  for the proportion  $p$ .

If  $p = 0$  or  $1$ , then the logit is undefined. `logit` can remap the proportions to the interval  $(\text{adjust}, 1 - \text{adjust})$  prior to the transformation. If it adjusts the data automatically, `logit` will print a warning message.

### Value

a numeric vector or array of the same shape and size as  $p$ .

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

### See Also

[probabilityAxis](#)

## Examples

```
options(digits=4)
logit(.1*0:10)
## [1] -3.6636 -1.9924 -1.2950 -0.8001 -0.3847  0.0000  0.3847
## [8]  0.8001  1.2950  1.9924  3.6636
## Warning message:
## proportions remapped to (0.025, 0.975) in: logit(0.1 * 0:10)

logit(.1*0:10, adjust=0)
## [1]   -Inf -2.1972 -1.3863 -0.8473 -0.4055  0.0000  0.4055
## [8]  0.8473  1.3863  2.1972    Inf
```

---

Mandel

*Contrived Collinear Data*

---

## Description

The Mandel data frame has 8 rows and 3 columns.

## Usage

```
Mandel
```

## Format

This data frame contains the following columns:

**x1** first predictor.

**x2** second predictor.

**y** response.

## Source

Mandel, J. (1982) Use of the singular value decomposition in regression analysis. *The American Statistician* **36**, 15–24.

## References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

 mcPlots

*Draw Linear Model Marginal and Conditional Plots in Parallel or Overlaid*


---

### Description

the `mcPlot` function draws two plots, or overlay the two graphs on one plot. For a response  $Y$  and a regressor  $X$ , the first plot is the *marginal* plot of  $Y$  versus  $X$  with both variables centered, visualizing the conditional distribution of  $Y$  given  $X$  ignoring all other regressors. The second plot is an added-variable for  $X$  after all other regressors, visualizing the conditional distribution of  $Y$  given  $X$  after adjusting for all other predictors. The added variable plot by default is drawn using the same `xlim` and `ylim` as the centered marginal plot to emphasize that conditioning removes variation in both the regressor and the response. The plot is primarily intended as a pedagogical tool for understanding coefficients in first-order models.

### Usage

```
mcPlots(model, terms=~., layout=NULL, ask, overlaid=TRUE, ...)
```

```
mcPlot(model, ...)
```

```
## S3 method for class 'lm'
```

```
mcPlot(model, variable,
        id.method = list(abs(residuals(model, type="pearson")), "x"),
        labels,
        id.n = if(id.method[1]=="identify") Inf else 0,
        id.cex=1, id.col=palette()[1], id.location="lr",
        col.marginal="blue", col.conditional="red", col.arrows="gray",
        pch = c(16, 1), lwd = 2, grid=TRUE,   ###removed arg main
        ellipse=FALSE, ellipse.args=list(levels=0.5),
        overlaid=TRUE, new=TRUE, ...)
```

### Arguments

<code>model</code>	model object produced by <code>lm</code> .
<code>terms</code>	A one-sided formula that specifies a subset of the predictors. One added-variable plot is drawn for each numeric predictor and for each basis function used to define a factor. For example, the specification <code>terms = ~ . - X3</code> would plot against all terms except for <code>X3</code> . If this argument is a quoted name of one of the terms, the added-variable plot is drawn for that term only.
<code>variable</code>	A quoted string giving the name of a numeric predictor in the model matrix for the horizontal axis. To plot against a factor, you need to specify the full name of one of the indicator variables that define the factor. For example, for a factor called <code>type</code> with levels <code>A</code> , <code>B</code> and <code>C</code> , using the usual drop-first level parameterization of the factor, the regressors for <code>type</code> would be <code>typeB</code> or <code>typeC</code> .

layout	If set to a value like <code>c(1, 2)</code> or <code>c(6, 2)</code> , the layout of the graph will have this many rows and columns. If not set, behavior depends on the value of the <code>overlaid</code> argument; see the details
ask	If TRUE, ask the user before drawing the next plot; if FALSE don't ask.
...	mcPlots passes these arguments to mcPlot. mcPlot passes arguments to plot.
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
overlaid	If TRUE, the default, overlay the marginal and conditional plots on the same graph; otherwise plot them side-by-side. See the details below
col.marginal, col.conditional	colors for points, lines, ellipses in the marginal and conditional plots, respectively
col.arrows	color for the arrows with <code>overlaid=TRUE</code>
pch	Plotting character for marginal and conditional plots, respectively.
lwd	line width; default is 2 (see <a href="#">par</a> ).
grid	If TRUE, the default, a light-gray background grid is put on the graph.
ellipse	If TRUE, plot a concentration ellipse; default is FALSE.
ellipse.args	Arguments to pass to the <a href="#">dataEllipse</a> function, in the form of a list with named elements; e.g., <code>ellipse.args=list(robust=TRUE)</code> will cause the ellipse to be plotted using a robust covariance-matrix.
new	if TRUE, the default, the plot window is reset when <code>overlaid=FALSE</code> using <code>par{mfrow=c(1, 2)}</code> . If FALSE, the layout of the plot window is not reset. Users will ordinarily ignore this argument.

## Details

With an `lm` object, suppose the response is  $Y$ ,  $X$  is a focal numeric predictor of interest, and  $Z$  is all the remaining predictors, possibly including interactions and factors. This function produces two graphs. The first graph is the marginal plot of  $Y$  versus  $X$ , with each variable centered around its mean. The second conditional plot is the added-variable plot of  $e(Y|Z)$  versus  $e(X|Z)$  where  $e(a|b)$  means the Pearson residuals from the regression of  $a$  on  $b$ . If `overlaid=TRUE`, these two plots are overlaid in one graph, with the points in different colors. In addition, each point in the marginal plot is joined to its value in the conditional plot by an arrow. Least squares regression lines fit to the marginal and conditional graphs are also shown; data ellipsoids can also be added. If `overlaid=FALSE`, then the two graphs are shown in side-by-side plots as long as the second argument to `layout` is equal to 2, or `layout` is set by the function. The arrows are omitted if the graphs are not overlaid.

These graphs are primarily for teaching, as the marginal plot shows the relationship between  $Y$  and  $X$  ignoring  $Z$ , while the conditional is the relationship between  $Y$  and  $X$  given  $Z$ . By keeping the scales the same in both graphs the effect of conditioning on both  $X$  and  $Y$  can be visualized.

This function is intended for first-order models with numeric predictors only. If the focal predictor is a factor, then one (pair) of mcPlots will be produced for each of the basis variables that define the factor, and the resulting plots are not generally meaningful because they depend on parameterization. If the mean function includes interactions, then mcPlots for main effects may violate

the hierarchy principle, and may also be of little interest. `mcPlots` for interactions of numerical predictors, however, can be useful.

These graphs are closely related to the ARES plots proposed by Cook and Weisberg (1989). This plot would benefit from animation.

### Value

These functions are used for their side effect of producing plots.

### Author(s)

John Fox <jfox@mcmaster.ca>, Sanford Weisberg <sandy@umn.edu>

### References

Cook, R. D. and Weisberg, S. (1989) *Regression diagnostics with dynamic graphics*, Technometrics, 31, 277.

Fox, J. (2016) *Applied Regression Analysis and Generalized Linear Models*, Third Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

### See Also

[avPlots](#), [residualPlots](#), [crPlots](#), [ceresPlots](#), [dataEllipse](#)

### Examples

```
m1 <- lm(partic ~ tfr + menwage + womwage + debt + parttime, data = Bfox)
mcPlot(m1, "womwage")
mcPlot(m1, "womwage", overlaid=FALSE, ellipse=TRUE)
```

---

Migration

*Canadian Interprovincial Migration Data*

---

### Description

The Migration data frame has 90 rows and 8 columns.

### Usage

Migration



## Format

This data frame contains the following columns:

- source** Province of origin (source). A factor with levels: ALTA, Alberta; BC, British Columbia; MAN, Manitoba; NB, New Brunswick; NFLD, New Foundland; NS, Nova Scotia; ONT, Ontario; PEI, Prince Edward Island; QUE, Quebec; SASK, Saskatchewan.
- destination** Province of destination (1971 residence). A factor with levels: ALTA, Alberta; BC, British Columbia; MAN, Manitoba; NB, New Brunswick; NFLD, New Foundland; NS, Nova Scotia; ONT, Ontario; PEI, Prince Edward Island; QUE, Quebec; SASK, Saskatchewan.
- migrants** Number of migrants (from source to destination) in the period 1966–1971.
- distance** Distance (between principal cities of provinces): NFLD, St. John; PEI, Charlottetown; NS, Halifax; NB, Fredricton; QUE, Montreal; ONT, Toronto; MAN, Winnipeg; SASK, Regina; ALTA, Edmonton; BC, Vancouver.
- pops66** 1966 population of source province.
- pops71** 1971 population of source province.
- popd66** 1966 population of destination province.
- popd71** 1971 population of destination province.

## Details

There is one record in the data file for each migration stream. You can average the 1966 and 1971 population figures for each of the source and destination provinces.

## Source

- Canada (1962) *Map*. Department of Mines and Technical Surveys.
- Canada (1971) *Census of Canada*. Statistics Canada, Vol. 1, Part 2 [Table 32].
- Canada (1972) *Canada Year Book*. Statistics Canada [p. 1369].

## References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

mmps

*Marginal Model Plotting*

---

## Description

For a regression object, draw a plot of the response on the vertical axis versus a linear combination  $u$  of regressors in the mean function on the horizontal axis. Added to the plot are a smooth for the graph, along with a smooth from the plot of the fitted values on  $u$ . `mmps` is an alias for `marginalModelPlots`, and `mmp` is an alias for `marginalModelPlot`.

**Usage**

```

marginalModelPlots(...)

mmmps(model, terms= ~ ., fitted=TRUE, layout=NULL, ask,
       main, groups, key=TRUE, ...)

marginalModelPlot(...)

## S3 method for class 'lm'
mmp(model, variable, sd = FALSE,
     xlab = deparse(substitute(variable)),
     smoother = loessLine, smoother.args=list(span=2/3),
     key=TRUE, pch, groups=NULL, ...)

## Default S3 method:
mmp(model, variable, sd = FALSE,
     xlab = deparse(substitute(variable)), smoother=loessLine,
     smoother.args, key=TRUE, pch, groups=NULL,
     col.line = palette()[c(4, 2)], col=palette()[1],
     labels, id.method="y",
     id.n=if(id.method[1]=="identify") Inf else 0,
     id.cex=1, id.col=palette()[1], id.location="lr", grid=TRUE, ...)

## S3 method for class 'glm'
mmp(model, variable, sd = FALSE,
     xlab = deparse(substitute(variable)), smoother=gamLine,
     smoother.args=list(k=3), key=TRUE, pch, groups=NULL,
     col.line = palette()[c(4, 2)], col=palette()[1],
     labels, id.method="y",
     id.n=if(id.method[1]=="identify") Inf else 0,
     id.cex=1, id.col=palette()[1], id.location="lr", grid=TRUE, ...)

```

**Arguments**

model	A regression object, usually of class either <code>lm</code> or <code>glm</code> , for which there is a predict method defined.
terms	A one-sided formula. A marginal model plot will be drawn for each term on the right-side of this formula that is not a factor. The default is <code>~ .</code> , which specifies that all the terms in <code>formula(object)</code> will be used. If a conditioning argument is given, eg <code>terms = ~.   a</code> , then separate colors and smoothers are used for each unique non-missing value of <code>a</code> . See examples below.
fitted	If the default <code>TRUE</code> , then a marginal model plot in the direction of the fitted values or linear predictor of a generalized linear model will be drawn.
layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does

	not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
<code>ask</code>	If TRUE, ask before clearing the graph window to draw more plots.
<code>main</code>	Main title for the array of plots. Use <code>main=""</code> to suppress the title; if missing, a title will be supplied.
<code>...</code>	Additional arguments passed from <code>mmps</code> to <code>mmp</code> and then to <code>plot</code> . Users should generally use <code>mmps</code> , or equivalently <code>marginalModelPlots</code> .
<code>variable</code>	The quantity to be plotted on the horizontal axis. The default is the predicted values <code>predict(object)</code> . It can be any other vector of length equal to the number of observations in the object. Thus the <code>mmp</code> function can be used to get a marginal model plot versus any regressor or predictor while the <code>mmps</code> function can be used only to get marginal model plots for the first-order regressors in the formula. In particular, terms defined by a spline basis are skipped by <code>mmps</code> , but you can use <code>mmp</code> to get the plot for the variable used to define the splines.
<code>sd</code>	If TRUE, display sd smooths. For a binomial regression with all sample sizes equal to one, this argument is ignored as the SD bounds don't make any sense.
<code>xlab</code>	label for horizontal axis
<code>smoother</code>	the name of the smoother to use, selected from the choices described at <a href="#">ScatterplotSmoothers</a> . For linear models and the default method, the default smoother is the function <code>loessLine</code> . For generalized linear models the default is <code>gamLine</code> , using the <code>gam</code> package and using splines.
<code>smoother.args</code>	arguments passed to the smoother. For linear models the defaults match the smoother used before September 2012, and may be changed later. See <a href="#">ScatterplotSmoothers</a> .
<code>groups</code>	The name of a vector that specifies a grouping variable for separate colors/smoothers. This can also be specified as a conditioning argument on the <code>terms</code> argument.
<code>key</code>	If TRUE, include a key at the top of the plot, if FALSE omit the key. If grouping is present, the key is only printed for the upper-left plot.
<code>id.method, labels, id.n, id.cex, id.col, id.location</code>	Arguments for labeling points. The default <code>id.n=0</code> suppresses labeling, and setting this argument greater than zero will include labeling. See <a href="#">showLabels</a> for these arguments.
<code>pch</code>	plotting character to use if no grouping is present.
<code>col.line</code>	colors for data and model smooth, respectively. Using the default palette, these are blue and red.
<code>col</code>	color(s) for the plotted points.
<code>grid</code>	If TRUE, the default, a light-gray background grid is put on the graph

## Details

`mmp` and `marginalModelPlot` draw one marginal model plot against whatever is specified as the horizontal axis. `mmps` and `marginalModelPlots` draws marginal model plots versus each of the terms in the `terms` argument and versus fitted values. `mmps` skips factors and interactions if they are specified in the `terms` argument. Terms based on polynomials or on splines (or potentially any term that is represented by a matrix of regressors) will be used to form a marginal model plot by returning

a linear combination of the terms. For example, if you specify `terms = ~ X1 + poly(X2, 3)` and `poly(X2, 3)` was part of the original model formula, the horizontal axis of the marginal model plot for X2 will be the value of `predict(model, type="terms")[, "poly(X2, 3)"]`. If the predict method for the model you are using doesn't support `type="terms"`, then the polynomial/spline term is skipped. Adding a conditioning variable, e.g., `terms = ~ a + b | c`, will produce marginal model plots for a and b with different colors and smoothers for each unique non-missing value of c.

The smoothers used were changed in September 2012. For linear models, the default smoother is still loess with the same smoothing parameters as were used in the past, but these can be changed with the argument `smoother.args`. For generalized linear models, the default smoother uses `gamLine`, fitting a generalized additive model with the same family, link and weights as the fit of the model. SD smooths are not computed for generalized linear models.

For generalized linear models the default number of elements in the spline basis is `k=3`; this is done to allow fitting for predictors with just a few support points. If you have many support points you may wish to set `k` to a higher number, or `k=-1` for the default used by `gam`.

### Value

Used for its side effect of producing plots.

### Author(s)

Sanford Weisberg, <sandy@umn.edu>

### References

Cook, R. D., & Weisberg, S. (1997). Graphics for assessing the adequacy of regression models. *Journal of the American Statistical Association*, 92(438), 490-499.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition. Sage.

Weisberg, S. (2005) *Applied Linear Regression*, Third Edition, Wiley, Section 8.4.

### See Also

[ScatterplotSmoothers](#), [plot](#)

### Examples

```
## Not run:
c1 <- lm(infant.mortality ~ gdp, UN)
mmmps(c1)
c2 <- update(c1, ~ poly(gdp, 4), data=na.omit(UN))
# plot against predict(c2, type="terms")[, "poly(gdp, 4)"] and
# and against gdp
mmmps(c2, ~ poly(gdp,4) + gdp)
# include SD lines
p1 <- lm(prestige ~ income + education, Prestige)
mmmps(p1, sd=TRUE)
# condition on type:
mmmps(p1, ~. | type)
# logistic regression example
```

```
# smoothers return warning messages.  
# fit a separate smoother and color for each type of occupation.  
  
m1 <- glm(lfp ~ ., family=binomial, data=Mroz)  
mmps(m1)  
  
## End(Not run)
```

---

Moore

*Status, Authoritarianism, and Conformity*

---

## Description

The Moore data frame has 45 rows and 4 columns. The data are for subjects in a social-psychological experiment, who were faced with manipulated disagreement from a partner of either of low or high status. The subjects could either conform to the partner's judgment or stick with their own judgment.

## Usage

Moore

## Format

This data frame contains the following columns:

**partner.status** Partner's status. A factor with levels: high, low.

**conformity** Number of conforming responses in 40 critical trials.

**fcategory** F-Scale Categorized. A factor with levels (note levels out of order): high, low, medium.

**fscore** Authoritarianism: F-Scale score.

## Source

Moore, J. C., Jr. and Krupat, E. (1971) Relationship between source status, authoritarianism and conformity in a social setting. *Sociometry* **34**, 122–134.

Personal communication from J. Moore, Department of Sociology, York University.

## References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

Mroz

*U.S. Women's Labor-Force Participation*

---

### Description

The Mroz data frame has 753 rows and 8 columns. The observations, from the Panel Study of Income Dynamics (PSID), are married women.

### Usage

Mroz

### Format

This data frame contains the following columns:

**lfp** labor-force participation; a factor with levels: no; yes.

**k5** number of children 5 years old or younger.

**k618** number of children 6 to 18 years old.

**age** in years.

**wc** wife's college attendance; a factor with levels: no; yes.

**hc** husband's college attendance; a factor with levels: no; yes.

**lwg** log expected wage rate; for women in the labor force, the actual wage rate; for women not in the labor force, an imputed value based on the regression of lwg on the other variables.

**inc** family income exclusive of wife's income.

### Source

Mroz, T. A. (1987) The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions. *Econometrica* **55**, 765–799.

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. (2000) *Multiple and Generalized Nonparametric Regression*. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Long, J. S. (1997) *Regression Models for Categorical and Limited Dependent Variables*. Sage.

---

`ncvTest`*Score Test for Non-Constant Error Variance*

---

**Description**

Computes a score test of the hypothesis of constant error variance against the alternative that the error variance changes with the level of the response (fitted values), or with a linear combination of predictors.

**Usage**

```
ncvTest(model, ...)  
  
## S3 method for class 'lm'  
ncvTest(model, var.formula, ...)  
  
## S3 method for class 'glm'  
ncvTest(model, ...) # to report an error
```

**Arguments**

<code>model</code>	a weighted or unweighted linear model, produced by <code>lm</code> .
<code>var.formula</code>	a one-sided formula for the error variance; if omitted, the error variance depends on the fitted values.
<code>...</code>	arguments passed down to methods functions; not currently used.

**Details**

This test is often called the Breusch-Pagan test; it was independently suggested with some extension by Cook and Weisberg (1983).

`ncvTest.glm` is a dummy function to generate an error when a `glm` model is used.

**Value**

The function returns a `chisqTest` object, which is usually just printed.

**Author(s)**

John Fox <jfox@mcmaster.ca>, Sandy Weisberg <sandy@umn.edu>

**References**

Breusch, T. S. and Pagan, A. R. (1979) A simple test for heteroscedasticity and random coefficient variation. *Econometrica* **47**, 1287–1294.

Cook, R. D. and Weisberg, S. (1983) Diagnostics for heteroscedasticity in regression. *Biometrika* **70**, 1–10.

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

### See Also

[hccm](#), [spreadLevelPlot](#)

### Examples

```
ncvTest(lm(interlocks ~ assets + sector + nation, data=Ornstein))
```

```
ncvTest(lm(interlocks ~ assets + sector + nation, data=Ornstein),
        ~ assets + sector + nation, data=Ornstein)
```

---

OBrienKaiser

*O'Brien and Kaiser's Repeated-Measures Data*

---

### Description

These contrived repeated-measures data are taken from O'Brien and Kaiser (1985). The data are from an imaginary study in which 16 female and male subjects, who are divided into three treatments, are measured at a pretest, posttest, and a follow-up session; during each session, they are measured at five occasions at intervals of one hour. The design, therefore, has two between-subject and two within-subject factors.

The contrasts for the treatment factor are set to  $-2, 1, 1$  and  $0, -1, 1$ . The contrasts for the gender factor are set to `contr.sum`.

### Usage

```
OBrienKaiser
```

### Format

A data frame with 16 observations on the following 17 variables.

`treatment` a factor with levels control A B

`gender` a factor with levels F M

`pre.1` pretest, hour 1

`pre.2` pretest, hour 2

`pre.3` pretest, hour 3

`pre.4` pretest, hour 4

`pre.5` pretest, hour 5

`post.1` posttest, hour 1

`post.2` posttest, hour 2



post.3 posttest, hour 3  
 post.4 posttest, hour 4  
 post.5 posttest, hour 5  
 fup.1 follow-up, hour 1  
 fup.2 follow-up, hour 2  
 fup.3 follow-up, hour 3  
 fup.4 follow-up, hour 4  
 fup.5 follow-up, hour 5

### Source

O'Brien, R. G., and Kaiser, M. K. (1985) MANOVA method for analyzing repeated measures designs: An extensive primer. *Psychological Bulletin* **97**, 316–333, Table 7.

### Examples

```
OBrienKaiser
contrasts(OBrienKaiser$treatment)
contrasts(OBrienKaiser$gender)
```

---

Ornstein

*Interlocking Directorates Among Major Canadian Firms*

---

### Description

The Ornstein data frame has 248 rows and 4 columns. The observations are the 248 largest Canadian firms with publicly available information in the mid-1970s. The names of the firms were not available.

### Usage

```
Ornstein
```

### Format

This data frame contains the following columns:

**assets** Assets in millions of dollars.

**sector** Industrial sector. A factor with levels: AGR, agriculture, food, light industry; BNK, banking; CON, construction; FIN, other financial; HLD, holding companies; MAN, heavy manufacturing; MER, merchandizing; MIN, mining, metals, etc.; TRN, transport; WOD, wood and paper.

**nation** Nation of control. A factor with levels: CAN, Canada; OTH, other foreign; UK, Britain; US, United States.

**interlocks** Number of interlocking director and executive positions shared with other major firms.

**Source**

Ornstein, M. (1976) The boards and executives of the largest Canadian corporations. *Canadian Journal of Sociology* **1**, 411–437.

Personal communication from M. Ornstein, Department of Sociology, York University.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

outlierTest	<i>Bonferroni Outlier Test</i>
-------------	--------------------------------

---

**Description**

Reports the Bonferroni p-values for Studentized residuals in linear and generalized linear models, based on a t-test for linear models and normal-distribution test for generalized linear models.

**Usage**

```
outlierTest(model, ...)

## S3 method for class 'lm'
outlierTest(model, cutoff=0.05, n.max=10, order=TRUE,
labels=names(rstudent), ...)

## S3 method for class 'outlierTest'
print(x, digits=5, ...)
```

**Arguments**

model	an lm or glm model object.
cutoff	observations with Bonferonni p-values exceeding cutoff are not reported, unless no observations are nominated, in which case the one with the largest Studentized residual is reported.
n.max	maximum number of observations to report (default, 10).
order	report Studentized residuals in descending order of magnitude? (default, TRUE).
labels	an optional vector of observation names.
...	arguments passed down to methods functions.
x	outlierTest object.
digits	number of digits for reported p-values.

**Details**

For a linear model, p-values reported use the t distribution with degrees of freedom one less than the residual df for the model. For a generalized linear model, p-values are based on the standard-normal distribution. The Bonferroni adjustment multiplies the usual two-sided p-value by the number of observations. The `lm` method works for `glm` objects. To show all of the observations set `cutoff=Inf` and `n.max=Inf`.

**Value**

an object of class `outlierTest`, which is normally just printed.

**Author(s)**

John Fox <jfox@mcmaster.ca> and Sanford Weisberg

**References**

- Cook, R. D. and Weisberg, S. (1982) *Residuals and Influence in Regression*. Chapman and Hall.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.
- Williams, D. A. (1987) Generalized linear model diagnostics using the deviance and single case deletions. *Applied Statistics* **36**, 181–191.

**Examples**

```
outlierTest(lm(prestige ~ income + education, data=Duncan))
```

---

panel.car

*Panel Function for Coplots*

---

**Description**

a panel function for use with `coplot` that plots points, a lowess line, and a regression line.

**Usage**

```
panel.car(x, y, col, pch, cex=1, span=0.5, lwd=2,  
reg.line=lm, lowess.line=TRUE, ...)
```

**Arguments**

x	vector giving horizontal coordinates.
y	vector giving vertical coordinates.
col	point color.
pch	plotting character for points.
cex	character expansion factor for points.
span	span for lowess smoother.
lwd	line width, default is 2.
reg.line	function to compute coefficients of regression line, or FALSE for no line.
lowess.line	if TRUE plot lowess smooth.
...	other arguments to pass to functions lines and regLine.

**Value**

NULL. This function is used for its side effect: producing a panel in a coplot.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**See Also**

[coplot](#), [regLine](#)

**Examples**

```
coplot
```

---

plot.powerTransform    *plot Method for powerTransform Objects*

---

**Description**

This function provides a simple function for plotting data using power transformations.

**Usage**

```
## S3 method for class 'powerTransform'
plot(x, z = NULL, round = TRUE, plot = pairs, ...)
```

**Arguments**

x	name of the power transformation object
z	Additional variables of the same length as those used to get the transformation to be plotted, default is NULL.
round	If TRUE, the default, use rounded transforms, if FALSE use the MLEs.
plot	Plotting method. Default is pairs. Another possible choice is scatterplot.matrix from the car package.
...	Optional arguments passed to the plotting method

**Details**

The data used to estimate transformations using powerTransform are plotted in the transformed scale.

**Value**

None. Produces a graph as a side-effect.

**Author(s)**

Sanford Weisberg, <sandy@umn.edu>

**References**

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Linear Regression*, Second Edition, Sage.

**See Also**

[powerTransform](#)

**Examples**

```
summary(a3 <- powerTransform(cbind(len, adt, trks, shld, sigs1) ~ 1, Highway1))  
with(Highway1, plot(a3, z=rate, col=as.numeric(htype)))
```

Pottery

*Chemical Composition of Pottery*

---

**Description**

The data give the chemical composition of ancient pottery found at four sites in Great Britain. They appear in Hand, et al. (1994), and are used to illustrate MANOVA in the SAS Manual. (Suggested by Michael Friendly.)

**Usage**

Pottery

**Format**

A data frame with 26 observations on the following 6 variables.

Site a factor with levels AshleyRails Caldicot IsleThorns Llanedyrn

Al Aluminum

Fe Iron

Mg Magnesium

Ca Calcium

Na Sodium

**Source**

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and E., O. (1994) *A Handbook of Small Data Sets*. Chapman and Hall.

**Examples**Pottery

---

powerTransform

*Finding Univariate or Multivariate Power Transformations*

---

**Description**

powerTransform uses the maximum likelihood-like approach of Box and Cox (1964) to select a transformation of a univariate or multivariate response for normality, linearity and/or constant variance. Available families of transformations are the default Box-Cox power family and two additional families that are modifications of the Box-Cox family that allow for (a few) negative responses. The summary method automatically computes two or three likelihood ratio type tests concerning the transformation powers.

**Usage**

```
powerTransform(object, ...)

## Default S3 method:
powerTransform(object, family="bcPower", ...)

## S3 method for class 'lm'
powerTransform(object, family="bcPower", ...)

## S3 method for class 'formula'
powerTransform(object, data, subset, weights, na.action,
  family="bcPower", ...)

## S3 method for class 'lmerMod'
powerTransform(object, family="bcPower", ...)
```

**Arguments**

object	This can either be an object of class <code>lm</code> or <code>lmerMod</code> , a formula, or a matrix or vector; see below.
family	The quoted name of a family of transformations. The available options are "bcPower" for the default for the Box-Cox power family; "bcnPower" for a two-parameter modification of the Box-Cox family that allows negative responses (Hawkins and Weisberg (2017)), and the "yjPower" family (Yeo and Johnson(2000)), another modification of the Box-Cox family that allows a few negative values. All three families are documented at <a href="#">bcPower</a> .
data	A data frame or environment, as in 'lm'.
subset	Case indices to be used, as in 'lm'.
weights	Weights as in 'lm'.
na.action	Missing value action, as in 'lm'.
...	Additional arguments that used in the interative algorithm; defaults are generally adequate.

**Details**

This function implements the Box and Cox (1964) method of selecting a power transformation of a variable toward normality, and its generalization by Velilla (1993) to a multivariate response. Cook and Weisberg (1999) and Weisberg (2014) suggest the usefulness of transforming a set of predictors  $z_1, z_2, z_3$  for multivariate normality. It also includes two additional families that allow for negative values.

If the 'object' argument is of class 'lm' or 'lmerMod', the Box-Cox procedure is applied to the conditional distribution of the response given the predictors. For 'lm' objects, the response may be multivariate, and each column will have its own transformation. With 'lmerMod' the response must be univariate.

The 'object' argument may also be a formula. For example,  $z \sim x_1 + x_2 + x_3$  will estimate a transformation for the response  $z$  from a family after fitting a linear model with the given

formula. `cbind(y1, y2, y3) ~ 1` specifies transformations to multivariate normality with no predictors. A vector value for ‘object’, for example `powerTransform(ais$LBM)`, is equivalent to `powerTransform(LBM ~ 1, ais)`. Similarly, `powerTransform(cbind(ais$LBM, ais$SSF))`, where the first argument is a matrix rather than a formula is equivalent to specification of a multivariate linear model `powerTransform(cbind(LBM, SSF) ~ 1, ais)`.

Three families of power transformations are available. The default Box-Cox power family (`family="bcPower"`) of power transformations effectively replaces a vector by that vector raised to a power, generally in the range from -3 to 3. For powers close to zero, the log-transformation is suggested. In practical situations, after estimating a power using the `powerTransform` function, a variable would be replaced by a simple power transformation of it, for example, if  $\lambda \approx 0.5$ , then the corresponding variable would be replaced by its square root; if  $\lambda$  is close enough to zero, the variable would be replaced by its natural logarithm. The Box-Cox family requires the responses to be strictly positive.

The `family="bcnPower"`, or Box-Cox with negatives, family proposed by Hawkins and Weisberg (2017) allows for (a few) non-positive values, while allowing for the transformed data to be interpreted similarly to the interpretation of Box-Cox transformed values. This family is the Box-Cox transformation of  $z = .5 * (y + (y^2 + \gamma^2)^{1/2})$  that depends on a location parameter  $\gamma$ . The quantity  $z$  is positive for all values of  $y$ . If  $\gamma = 0$  and  $y$  is strictly positive, then the Box-Cox and the `bcnPower` transformations are identical. When fitting the Box-Cox with negatives family, `lambda` is restricted to the range [-3, 3], and `gamma` is restricted to the range from .01 to the largest positive value of the variable, since values outside these ranges are unreasonable in practice.

The final family `family="yjPower"` uses the Yeo-Johnson transformation, which is the Box-Cox transformation of  $U + 1$  for nonnegative values, and of  $|U| + 1$  with parameter  $2 - \lambda$  for  $U$  negative and thus it provides a family for fitting when (a few) observations are negative. Because of the unusual constraints on the powers for positive and negative data, this transformation is not used very often, as results are difficult to interpret. In practical problems, a variable would be replaced by its Yeo-Johnson transformation computed using the `yjPower` function.

The function `testTransform` is used to obtain likelihood ratio tests for any specified value for the transformation parameter(s).

Computations maximize the likelihood-like functions described by Box and Cox (1964) and by Velilla (2000). For univariate responses, the computations are very stable and problems are unlikely, although for ‘lmer’ models computations may be very slow because the model is refit many times. For multivariate responses with the `bcnPower` family, the computing algorithm may fail. In this case we recommend adding the argument `itmax = 1` to the call to `powerTransform`. This will return the starting value estimates of the transformation parameters, fitting a d-dimensional response as if all the d responses were independent.

## Value

An object of class `powerTransform` or class `bcnPowerTransform` if `family="bcnPower"` that inherits from `powerTransform` is returned, including the components listed below.

A summary method presents estimated values for the transformation power ‘`lambda`’ and for the ‘`bcnPower`’ family the location parameter ‘`gamma`’ as well. Standard errors and Wald 95% confidence intervals based on the standard errors are computed from the inverse of the sample Hessian matrix evaluated at the estimates. The interval estimates for the ‘`gamma`’ parameters will generally be very wide, reflecting little information available about the location parameter. Likelihood ratio type tests are also provided. For the ‘`bcnPower`’ family these are based on the profile loglikelihood for ‘`lambda`’ alone; that is, we treat ‘`gamma`’ as a nuisance parameter and average over it.



The components of the returned object includes

lambda	Estimated transformation parameter
roundlam	Convenient rounded values for the estimates. These rounded values will usually be the desired transformations.
gamma	Estimated location parameters for bcnPower, NULL otherwise
invHess	Estimated covariance matrix of the estimated parameters
llik	Value of the log-likelihood at the estimates

The summary method for powerTransform returns an array with columns labeled "Est Power" for the value of lambda that maximizes the likelihood; "Rounded Pwr" for roundlam, and columns "Wald Lwr Bnd" and "Wald Ur Bnd" for a 95 percent Wald normal theory confidence interval for lambda computed as the estimate plus or minus 1.96 times the standard error.

### Author(s)

Sanford Weisberg, <sandy@umn.edu>

### References

- Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. *Journal of the Royal Statistical Society, Series B*. 26 211-46.
- Cook, R. D. and Weisberg, S. (1999) *Applied Regression Including Computing and Graphics*. Wiley.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.
- Hawkins, D. and Weisberg, S. (2017) Combining the Box-Cox Power and Generalized Log Transformations to Accomodate Negative Responses In Linear and Mixed-Effects Linear Models, submitted for publication.
- Velilla, S. (1993) A note on the multivariate Box-Cox transformation to normality. *Statistics and Probability Letters*, 17, 259-263.
- Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.
- Yeo, I. and Johnson, R. (2000) A new family of power transformations to improve normality or symmetry. *Biometrika*, 87, 954-959.

### See Also

[testTransform](#), [bcPower](#), [bcnPower](#), [transform](#), [optim](#), [boxCox](#).

### Examples

```
# Box Cox Method, univariate
summary(p1 <- powerTransform(cycles ~ len + amp + load, Wool))
# fit linear model with transformed response:
coef(p1, round=TRUE)
summary(m1 <- lm(bcPower(cycles, p1$roundlam) ~ len + amp + load, Wool))

# Multivariate Box Cox uses Highway1 data
summary(powerTransform(cbind(len, adt, trks, sigs1) ~ 1, Highway1))
```

```

# Multivariate transformation to normality within levels of 'htype'
summary(a3 <- powerTransform(cbind(len, adt, trks, sigs1) ~ htype, Highway1))

# test lambda = (0 0 0 -1)
testTransform(a3, c(0, 0, 0, -1))

# save the rounded transformed values, plot them with a separate
# color for each highway type
transformedY <- bcPower(with(Highway1, cbind(len, adt, trks, sigs1)),
  coef(a3, round=TRUE))
## Not run: scatterplotMatrix( ~ transformedY|htype, Highway1)

# With negative responses, use the bcnPower family
m2 <- lm(I1L1 ~ pool, LoBD)
summary(p2 <- powerTransform(m2, family="bcnPower"))
testTransform(p2, .5)
summary(powerTransform(update(m2, cbind(LoBD$I1L2, LoBD$I1L1) ~ .), family="bcnPower"))
# multivariate bcnPower, with 8 responses
summary(powerTransform(update(m2, as.matrix(LoBD[, -1]) ~ .), family="bcnPower"))
# multivariate bcnPower, fit with one iteration using starting values as estimates
summary(powerTransform(update(m2, as.matrix(LoBD[, -1]) ~ .), family="bcnPower", itmax=1))

# mixed effects model
## Not run:
data <- reshape(LoBD[1:20, ], varying=names(LoBD)[-1], direction="long", v.names="y")
names(data) <- c("pool", "assay", "y", "id")
data$assay <- factor(data$assay)
require(lme4)
m2 <- lmer(y ~ pool + (1|assay), data)
summary(l2 <- powerTransform(m2, family="bcnPower", verbose=TRUE))

## End(Not run)

```

---

Prestige

*Prestige of Canadian Occupations*


---

## Description

The Prestige data frame has 102 rows and 6 columns. The observations are occupations.

## Usage

```
Prestige
```

## Format

This data frame contains the following columns:

**education** Average education of occupational incumbents, years, in 1971.

- income** Average income of incumbents, dollars, in 1971.
- women** Percentage of incumbents who are women.
- prestige** Pineo-Porter prestige score for occupation, from a social survey conducted in the mid-1960s.
- census** Canadian Census occupational code.
- type** Type of occupation. A factor with levels (note: out of order): bc, Blue Collar; prof, Professional, Managerial, and Technical; wc, White Collar.

### Source

- Canada (1971) *Census of Canada*. Vol. 3, Part 6. Statistics Canada [pp. 19-1–19-21].
- Personal communication from B. Blishen, W. Carroll, and C. Moore, Departments of Sociology, York University and University of Victoria.

### References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

 qqPlot

*Quantile-Comparison Plots*


---

### Description

Plots empirical quantiles of a variable, or of studentized residuals from a linear model, against theoretical quantiles of a comparison distribution.

### Usage

```
qqPlot(x, ...)

qqp(...)

## Default S3 method:
qqPlot(x, distribution="norm", ...,
       ylab=deparse(substitute(x)), xlab=paste(distribution, "quantiles"),
       main=NULL, las=par("las"),
       envelope=.95,
       col=palette()[1], col.lines=palette()[2], lwd=2, pch=1, cex=par("cex"),
       line=c("quartiles", "robust", "none"),
       labels = if(!is.null(names(x))) names(x) else seq(along=x),
       id.method = "y",
       id.n =if(id.method[1]=="identify") Inf else 0,
       id.cex=1, id.col=palette()[1], id.location="lr", grid=TRUE)
```

```
## S3 method for class 'lm'
qqPlot(x, xlab=paste(distribution, "Quantiles"),
ylab=paste("Studentized Residuals(", deparse(substitute(x)), ")"),
sep=""), main=NULL,
distribution=c("t", "norm"), line=c("robust", "quartiles", "none"),
las=par("las"), simulate=TRUE, envelope=.95,
reps=100, col=palette()[1], col.lines=palette()[2], lwd=2,
pch=1, cex=par("cex"),
labels, id.method = "y",
id.n = if(id.method[1]=="identify") Inf else 0,
id.cex=1, id.col=palette()[1], id.location="lr", grid=TRUE, ...)
```

### Arguments

<code>x</code>	vector of numeric values or <code>lm</code> object.
<code>distribution</code>	root name of comparison distribution – e.g., "norm" for the normal distribution; <code>t</code> for the t-distribution.
<code>ylab</code>	label for vertical (empirical quantiles) axis.
<code>xlab</code>	label for horizontal (comparison quantiles) axis.
<code>main</code>	label for plot.
<code>envelope</code>	confidence level for point-wise confidence envelope, or FALSE for no envelope.
<code>las</code>	if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see <a href="#">par</a> ).
<code>col</code>	color for points; the default is the <i>first</i> entry in the current color palette (see <a href="#">palette</a> and <a href="#">par</a> ).
<code>col.lines</code>	color for lines; the default is the <i>second</i> entry in the current color palette.
<code>pch</code>	plotting character for points; default is 1 (a circle, see <a href="#">par</a> ).
<code>cex</code>	factor for expanding the size of plotted symbols; the default is 1.
<code>labels</code>	vector of text strings to be used to identify points, defaults to <code>names(x)</code> or observation numbers if <code>names(x)</code> is NULL.
<code>id.method</code>	point identification method. The default <code>id.method="y"</code> will identify the <code>id.n</code> points with the largest value of <code>abs(y-mean(y))</code> . See <a href="#">showLabels</a> for other options.
<code>id.n</code>	number of points labeled. If <code>id.n=0</code> , the default, no point identification.
<code>id.cex</code>	set size of the text for point labels; the default is <code>cex</code> (which is typically 1).
<code>id.col</code>	color for the point labels.
<code>id.location</code>	The default "lr" identifies to the left or right of the point; the alternative "ab" identifies above or below the point.
<code>lwd</code>	line width; default is 2 (see <a href="#">par</a> ).
<code>line</code>	"quartiles" to pass a line through the quartile-pairs, or "robust" for a robust-regression line; the latter uses the <code>r1m</code> function in the MASS package. Specifying <code>line = "none"</code> suppresses the line.

simulate	if TRUE calculate confidence envelope by parametric bootstrap; for lm object only. The method is due to Atkinson (1985).
reps	integer; number of bootstrap replications for confidence envelope.
...	arguments such as df to be passed to the appropriate quantile function.
grid	If TRUE, the default, a light-gray background grid is put on the graph

### Details

Draws theoretical quantile-comparison plots for variables and for studentized residuals from a linear model. A comparison line is drawn on the plot either through the quartiles of the two distributions, or by robust regression.

Any distribution for which quantile and density functions exist in R (with prefixes q and d, respectively) may be used. When plotting a vector, the confidence envelope is based on the SEs of the order statistics of an independent random sample from the comparison distribution (see Fox, 2008). Studentized residuals from linear models are plotted against the appropriate t-distribution with a point-wise confidence envelope computed by default by a parametric bootstrap, as described by Atkinson (1985). The function qqp is an abbreviation for qqPlot.

### Value

These functions return the labels of identified points.

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.  
Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.  
Atkinson, A. C. (1985) *Plots, Transformations, and Regression*. Oxford.

### See Also

[qqplot](#), [qqnorm](#), [qqline](#), [showLabels](#)

### Examples

```
x<-rchisq(100, df=2)
qqPlot(x)
qqPlot(x, dist="chisq", df=2)

qqPlot(lm(prestige ~ income + education + type, data=Duncan),
envelope=.99)
```

---

Quartet

*Four Regression Datasets*

---

### Description

The Quartet data frame has 11 rows and 5 columns. These are contrived data.

### Usage

Quartet

### Format

This data frame contains the following columns:

**x** X-values for datasets 1–3.

**y1** Y-values for dataset 1.

**y2** Y-values for dataset 2.

**y3** Y-values for dataset 3.

**x4** X-values for dataset 4.

**y4** Y-values for dataset 4.

### Source

Anscombe, F. J. (1973) Graphs in statistical analysis. *American Statistician* **27**, 17–21.

---

recode

*Recode a Variable*

---

### Description

Recodes a numeric vector, character vector, or factor according to simple recode specifications. Recode is an alias for recode that avoids name clashes with packages, such as **Hmisc**, that have a recode function.

### Usage

```
recode(var, recodes, as.factor.result, as.numeric.result=TRUE, levels)
```

```
Recode(...)
```

**Arguments**

<code>var</code>	numeric vector, character vector, or factor.
<code>recodes</code>	character string of recode specifications: see below.
<code>as.factor.result</code>	return a factor; default is TRUE if <code>var</code> is a factor, FALSE otherwise.
<code>as.numeric.result</code>	if TRUE (the default), and <code>as.factor.result</code> is FALSE, then the result will be coerced to numeric if all values in the result are numerals—i.e., represent numbers.
<code>levels</code>	an optional argument specifying the order of the levels in the returned factor; the default is to use the sort order of the level names.
<code>...</code>	arguments to be passed to <code>recode</code> .

**Details**

Recode specifications appear in a character string, separated by semicolons (see the examples below), of the form `input=output`. If an input value satisfies more than one specification, then the first (from left to right) applies. If no specification is satisfied, then the input value is carried over to the result. NA is allowed on input and output. Several recode specifications are supported:

**single value** For example, `0=NA`.

**vector of values** For example, `c(7, 8, 9)='high'`.

**range of values** For example, `7:9='C'`. The special values `lo` and `hi` may appear in a range. For example, `lo:10=1`. *Note:* `:` is *not* the R sequence operator. In addition you may not use `:` with the collect operator, so for example `c(1, 3, 5:7)` will cause an error.

**else** everything that does not fit a previous specification. For example, `else=NA`. Note that `else` matches *all* otherwise unspecified values on input, including NA.

If all of the output values are numeric, and if `as.factor.result` is FALSE, then a numeric result is returned; if `var` is a factor, then by default so is the result.

**Value**

a recoded vector of the same length as `var`.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[cut](#), [factor](#)

**Examples**

```
x<-rep(1:3,3)
x
## [1] 1 2 3 1 2 3 1 2 3
recode(x, "c(1,2)='A';
else='B'")
## [1] "A" "A" "B" "A" "A" "B" "A" "A" "B"
Recode(x, "1:2='A'; 3='B'")
## [1] "A" "A" "B" "A" "A" "B" "A" "A" "B"
```

regLine

*Plot Regression Line***Description**

Plots a regression line on a scatterplot; the line is plotted between the minimum and maximum x-values.

**Usage**

```
regLine(mod, col=palette()[2], lwd=2, lty=1,...)
```

**Arguments**

mod	a model, such as produced by <code>lm</code> , that responds to the <code>coef</code> function by returning a 2-element vector, whose elements are interpreted respectively as the intercept and slope of a regression line.
col	color for points and lines; the default is the <i>second</i> entry in the current color palette (see <a href="#">palette</a> and <a href="#">par</a> ).
lwd	line width; default is 2 (see <a href="#">par</a> ).
lty	line type; default is 1, a solid line (see <a href="#">par</a> ).
...	optional arguments to be passed to the lines plotting function.

**Details**

In contrast to `abline`, this function plots only over the range of the observed x-values. The x-values are extracted from `mod` as the second column of the model matrix.

**Value**

NULL. This function is used for its side effect: adding a line to the plot.

**Author(s)**

John Fox <jfox@mcmaster.ca>



**See Also**[abline](#), [lines](#)**Examples**

```
plot(repwt ~ weight, pch=c(1,2)[sex], data=Davis)
regLine(lm(repwt~weight, subset=sex=="M", data=Davis))
regLine(lm(repwt~weight, subset=sex=="F", data=Davis), lty=2)
```

residualPlots

*Residual Plots and Curvature Tests for Linear Model Fits***Description**

Plots the residuals versus each term in a mean function and versus fitted values. Also computes a curvature test for each of the plots by adding a quadratic term and testing the quadratic to be zero. This is Tukey's test for nonadditivity when plotting against fitted values.

**Usage**

```
### This is a generic function with only one required argument:
```

```
residualPlots (model, ...)
```

```
## Default S3 method:
```

```
residualPlots(model, terms = ~., layout = NULL, ask,
              main = "", fitted = TRUE, AsIs=TRUE, plot = TRUE,
              tests = TRUE, groups, ...)
```

```
## S3 method for class 'lm'
```

```
residualPlots(model, ...)
```

```
## S3 method for class 'glm'
```

```
residualPlots(model, ...)
```

```
### residualPlots calls residualPlot, so these arguments can be
```

```
### used with either function
```

```
residualPlot(model, ...)
```

```
## Default S3 method:
```

```
residualPlot(model, variable = "fitted", type = "pearson",
             groups,
             plot = TRUE,
             linear = TRUE,
             quadratic = if(missing(groups)) TRUE else FALSE,
             smoother=NULL, smoother.args=list(),
```

```

col.smooth=palette()[3],
labels,
id.method = "r",
id.n = if(id.method[1]=="identify") Inf else 0,
id.cex=1, id.col=palette()[1], id.location="lr",
col = palette()[1], col.quad = palette()[2],
pch=1,
xlab, ylab, lwd = 1, lty = 1,
grid=TRUE, key=!missing(groups), ...)

## S3 method for class 'lm'
residualPlot(model, ...)

## S3 method for class 'glm'
residualPlot(model, variable = "fitted", type = "pearson",
plot = TRUE, quadratic = FALSE,
smoother = loessLine, smoother.args=list(k=3), ...)

```

## Arguments

model	A regression object.
terms	A one-sided formula that specifies a subset of the predictors. One residual plot is drawn for each specified. The default <code>~ .</code> is to plot against all predictors. For example, the specification <code>terms = ~ . - X3</code> would plot against all predictors except for X3. To get a plot against fitted values only, use the arguments <code>terms = ~ 1</code> , Interactions are skipped. For polynomial terms, the plot is against the first-order variable (which may be centered and scaled depending on how the poly function is used). Plots against factors are boxplots. Plots against other matrix terms, like splines, use the result of <code>predict(model), type="terms"[, variable]</code> as the horizontal axis; if the <code>predict</code> method doesn't permit this type, then matrix terms are skipped.  A grouping variable can also be specified in the terms, so, for example <code>terms= ~ .   type</code> would use the factor type to set a different color and symbol for each level of type. Any fits in the plots will also be done separately for each level of group.
layout	If set to a value like <code>c(1, 1)</code> or <code>c(4, 3)</code> , the layout of the graph will have this many rows and columns. If not set, the program will select an appropriate layout. If the number of graphs exceed nine, you must select the layout yourself, or you will get a maximum of nine per page. If <code>layout=NA</code> , the function does not set the layout and the user can use the <code>par</code> function to control the layout, for example to have plots from two models in the same graphics window.
ask	If TRUE, ask the user before drawing the next plot; if FALSE, don't ask.
main	Main title for the graphs. The default is <code>main=""</code> for no title.
fitted	If TRUE, the default, include the plot against fitted values.
AsIs	If FALSE, terms that use the "as-is" function I are skipped; if TRUE, the default, they are included.
plot	If TRUE, draw the plot(s).

tests	If TRUE, display the curvature tests. With glm's, the argument start is ignored in computing the curvature tests.
...	Additional arguments passed to residualPlot and then to plot.
variable	Quoted variable name for the horizontal axis, or "fitted" to plot versus fitted values.
type	Type of residuals to be used. Pearson residuals are appropriate for lm objects since these are equivalent to ordinary residuals with ols and correctly weighted residuals with wls. Any quoted string that is an appropriate value of the type argument to <code>residuals.lm</code> or "rstudent" or "rstandard" for Studentized or standardized residuals.
groups	A list of group indicators. Points in different groups will be plotted with different colors and symbols. If missing, no grouping. In residualPlots, the grouping variable can also be set in the terms argument, as described above. The default is no grouping.
linear	If TRUE, adds a horizontal line at zero if no groups. With groups, display the within level of groups ols regression of the residuals as response and the horizontal axis as the regressor.
quadratic	if TRUE, fits the quadratic regression of the vertical axis on the horizontal axis and displays a lack of fit test. Default is TRUE for lm and FALSE for glm or if groups not missing.
smoother	the name of the smoother to use, selected from the choices described at <a href="#">ScatterplotSmoother</a> s. For lm objects the default is NULL. For glm object the default is loessLine.
smoother.args	arguments passed to the smoother. See <a href="#">ScatterplotSmoother</a> s. For generalized linear models the number of elements in the spline basis is set to k=3; this is done to allow fitting for predictors with just a few support points. If you have many support points you may wish to set k to a higher number, or k=-1 for the default used by <code>gam</code> .
col.smooth	color for the smoother if groups missing, and ignored if groups is set.
id.method, labels, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is id.n=0 for labeling no points. See <a href="#">showLabels</a> for details of these arguments.
col	default color for points. If groups is set, col can be a list at least as long as the number of levels for groups giving the colors for each groups.
col.quad	default color for quadratic fit if groups is missing. Ignored if groups are used.
pch	plotting character. The default is pch=1. If groups are used, pch can be set to a vector at least as long as the number of groups.
xlab	X-axis label. If not specified, a useful label is constructed by the function.
ylab	Y-axis label. If not specified, a useful label is constructed by the function.
lwd	line width for lines.
lty	line type for quadratic.
grid	If TRUE, the default, a light-gray background grid is put on the graph
key	Should a key be added to the plot? Default is !is.null(groups).

## Details

`residualPlots` draws one or more residuals plots depending on the value of the `terms` and `fitted` arguments. If `terms = ~ .`, the default, then a plot is produced of residuals versus each first-order term in the formula used to create the model. Interaction terms, spline terms, and polynomial terms of more than one predictor are skipped. In addition terms that use the “as-is” function, e.g.,  $I(X^2)$ , will also be skipped unless you set the argument `AsIs=TRUE`. A plot of residuals versus fitted values is also included unless `fitted=FALSE`.

In addition to plots, a table of curvature tests is displayed. For plots against a term in the model formula, say  $X_1$ , the test displayed is the t-test for  $I(X^2)$  in the fit of `update(model, ~. + I(X^2))`. Econometricians call this a specification test. For factors, the displayed plot is a boxplot, no curvature test is computed, and grouping is ignored. For fitted values, the test is Tukey’s one-degree-of-freedom test for nonadditivity. You can suppress the tests with the argument `tests=FALSE`. If grouping is used curvature tests are not displayed.

`residualPlot`, which is called by `residualPlots`, should be viewed as an internal function, and is included here to display its arguments, which can be used with `residualPlots` as well. The `residualPlot` function returns the curvature test as an invisible result.

`residCurvTest` computes the curvature test only. For any factors a boxplot will be drawn. For any polynomials, plots are against the linear term. Other non-standard predictors like B-splines are skipped.

## Value

For `lm` objects, returns a data.frame with one row for each plot drawn, one column for the curvature test statistic, and a second column for the corresponding p-value. This function is used primarily for its side effect of drawing residual plots.

## Author(s)

Sanford Weisberg, <sandy@umn.edu>

## References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition. Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley, Chapter 8

## See Also

See Also [lm](#), [identify](#), [showLabels](#)

## Examples

```
m1 <- lm(prestige ~ income, data=Prestige)
residualPlots(m1)
residualPlots(m1, terms= ~ 1 | type) # plot vs. yhat grouping by type
```

---

Robey

*Fertility and Contraception*

---

**Description**

The Robey data frame has 50 rows and 3 columns. The observations are developing nations around 1990.

**Usage**

Robey

**Format**

This data frame contains the following columns:

**region** A factor with levels: Africa; Asia, Asia and Pacific; Latin.Amer, Latin America and Caribbean; Near .East, Near East and North Africa.

**tfr** Total fertility rate (children per woman).

**contraceptors** Percent of contraceptors among married women of childbearing age.

**Source**

Robey, B., Shea, M. A., Rutstein, O. and Morris, L. (1992) The reproductive revolution: New survey findings. *Population Reports*. Technical Report M-11.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

Sahlins

*Agricultural Production in Mazulu Village*

---

**Description**

The Sahlins data frame has 20 rows and 2 columns. The observations are households in a Central African village.

**Usage**

Sahlins

**Format**

This data frame contains the following columns:

**consumers** Consumers/Gardener, ratio of consumers to productive individuals.

**acres** Acres/Gardener, amount of land cultivated per gardener.

**Source**

Sahlins, M. (1972) *Stone Age Economics*. Aldine [Table 3.1].

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

Salaries

*Salaries for Professors*

---

**Description**

The 2008-09 nine-month academic salary for Assistant Professors, Associate Professors and Professors in a college in the U.S. The data were collected as part of the on-going effort of the college's administration to monitor salary differences between male and female faculty members.

**Usage**

Salaries

**Format**

A data frame with 397 observations on the following 6 variables.

rank a factor with levels AssocProf AsstProf Prof

discipline a factor with levels A ("theoretical" departments) or B ("applied" departments).

yrs.since.phd years since PhD.

yrs.service years of service.

sex a factor with levels Female Male

salary nine-month salary, in dollars.

**References**

Fox J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition Sage.

**Description**

The `scatter3d` function uses the `rgl` package to draw 3D scatterplots with various regression surfaces. The function `Identify3d` allows you to label points interactively with the mouse: Press the right mouse button (on a two-button mouse) or the centre button (on a three-button mouse), drag a rectangle around the points to be identified, and release the button. Repeat this procedure for each point or set of “nearby” points to be identified. To exit from point-identification mode, click the right (or centre) button in an empty region of the plot.

**Usage**

```
scatter3d(x, ...)

## S3 method for class 'formula'
scatter3d(formula, data, subset, radius, xlab, ylab, zlab, labels, ...)

## Default S3 method:
scatter3d(x, y, z,
  xlab=deparse(substitute(x)), ylab=deparse(substitute(y)),
  zlab=deparse(substitute(z)), axis.scales=TRUE, axis.ticks=FALSE,
  revolutions=0, bg.col=c("white", "black"),
  axis.col=if (bg.col == "white") c("darkmagenta", "black", "darkcyan")
  else c("darkmagenta", "white", "darkcyan"),
  surface.col=c("blue", "green", "orange", "magenta", "cyan", "red",
    "yellow", "gray"), surface.alpha=0.5,
  neg.res.col="red", pos.res.col="green",
  square.col=if (bg.col == "white") "black" else "gray", point.col="yellow",
  text.col=axis.col, grid.col=if (bg.col == "white") "black" else "gray",
  fogtype=c("exp2", "linear", "exp", "none"),
  residuals=(length(fit) == 1), surface=TRUE, fill=TRUE, grid=TRUE,
  grid.lines=26, df.smooth=NULL, df.additive=NULL,
  sphere.size=1, radius=1, threshold=0.01, speed=1, fov=60,
  fit="linear", groups=NULL, parallel=TRUE,
  ellipsoid=FALSE, level=0.5, ellipsoid.alpha=0.1,
  id.method=c("mahal", "xz", "y", "xyz", "identify", "none"),
  id.n=if (id.method == "identify") Inf else 0,
  labels=as.character(seq(along=x)), offset = ((100/length(x))^(1/3)) * 0.02,
  model.summary=FALSE, ...)

Identify3d(x, y, z, axis.scales=TRUE, groups = NULL, labels = 1:length(x),
  col = c("blue", "green", "orange", "magenta", "cyan", "red", "yellow", "gray"),
  offset = ((100/length(x))^(1/3)) * 0.02)
```

**Arguments**

formula	“model” formula, of the form $y \sim x + z$ or (to plot by groups) $y \sim x + z \mid g$ , where $g$ evaluates to a factor or other variable dividing the data into groups.
data	data frame within which to evaluate the formula.
subset	expression defining a subset of observations.
x	variable for horizontal axis.
y	variable for vertical axis (response).
z	variable for out-of-screen axis.
xlab, ylab, zlab	axis labels.
axis.scales	if TRUE, label the values of the ends of the axes. <i>Note:</i> For <code>Identify3d</code> to work properly, the value of this argument must be the same as in <code>scatter3d</code> .
axis.ticks	if TRUE, print interior axis-“tick” labels; the default is FALSE. (The code for this option was provided by David Winsemius.)
revolutions	number of full revolutions of the display.
bg.col	background colour; one of “white”, “black”.
axis.col	colours for axes; if <code>axis.scales</code> is FALSE, then the second colour is used for all three axes.
surface.col	vector of colours for regression planes, used in the order specified by <code>fit</code> ; for multi-group plots, the colours are used for the regression surfaces and points in the several groups.
surface.alpha	transparency of regression surfaces, from 0.0 (fully transparent) to 1.0 (opaque); default is 0.5.
neg.res.col, pos.res.col	colours for lines representing negative and positive residuals.
square.col	colour to use to plot squared residuals.
point.col	colour of points.
text.col	colour of axis labels.
grid.col	colour of grid lines on the regression surface(s).
fogtype	type of fog effect; one of “exp2”, “linear”, “exp”, “none”.
residuals	plot residuals if TRUE; if <code>residuals=“squares”</code> , then the squared residuals are shown as squares (using code adapted from Richard Heiberger). Residuals are available only when there is one surface plotted.
surface	plot surface(s) (TRUE or FALSE).
fill	fill the plotted surface(s) with colour (TRUE or FALSE).
grid	plot grid lines on the regression surface(s) (TRUE or FALSE).
grid.lines	number of lines (default, 26) forming the grid, in each of the $x$ and $z$ directions.
df.smooth	degrees of freedom for the two-dimensional smooth regression surface; if NULL (the default), the <code>gam</code> function will select the degrees of freedom for a smoothing spline by generalized cross-validation; if a positive number, a fixed regression spline will be fit with the specified degrees of freedom.



<code>df.additive</code>	degrees of freedom for each explanatory variable in an additive regression; if NULL (the default), the gam function will select degrees of freedom for the smoothing splines by generalized cross-validation; if a positive number or a vector of two positive numbers, fixed regression splines will be fit with the specified degrees of freedom for each term.
<code>sphere.size</code>	general size of spheres representing points; the actual size is dependent on the number of observations.
<code>radius</code>	relative radii of the spheres representing the points. This is normally a vector of the same length as the variables giving the coordinates of the points, and for the formula method, that must be the case or the argument may be omitted, in which case spheres are the same size; for the default method, the default for the argument, 1, produces spheres all of the same size. The radii are scaled so that their median is 1.
<code>threshold</code>	if the actual size of the spheres is less than the threshold, points are plotted instead.
<code>speed</code>	relative speed of revolution of the plot.
<code>fov</code>	field of view (in degrees); controls degree of perspective.
<code>fit</code>	one or more of "linear", "quadratic", "smooth", "additive"; to display fitted surface(s); partial matching is supported – e.g., <code>c("lin", "quad")</code> .
<code>groups</code>	if NULL (the default), no groups are defined; if a factor, a different surface or set of surfaces is plotted for each level of the factor; in this event, the colours in <code>surface.col</code> are used successively for the points, surfaces, and residuals corresponding to each level of the factor.
<code>parallel</code>	when plotting surfaces by groups, should the surfaces be constrained to be parallel? A logical value, with default TRUE.
<code>ellipsoid</code>	plot concentration ellipsoid(s) (TRUE or FALSE).
<code>level</code>	expected proportion of bivariate-normal observations included in the concentration ellipsoid(s); default is 0.5.
<code>ellipsoid.alpha</code>	transparency of ellipsoids, from 0.0 (fully transparent) to 1.0 (opaque); default is 0.1.
<code>id.method</code>	if "mahal" (the default), relatively extreme points are identified automatically according to their Mahalanobis distances from the centroid (point of means); if "identify", points are identified interactively by right-clicking and dragging a box around them; right-click in an empty area to exit from interactive-point-identification mode; if "xz", identify extreme points in the predictor plane; if "y", identify unusual values of the response; if "xyz" identify unusual values of an variable; if "none", no point identification. See <a href="#">showLabels</a> for more information.
<code>id.n</code>	Number of relatively extreme points to identify automatically (default, 0 unless <code>id.method="identify"</code> ).
<code>model.summary</code>	print summary or summaries of the model(s) fit (TRUE or FALSE). <code>scatter3d</code> rescales the three variables internally to fit in the unit cube; this rescaling will affect regression coefficients.

labels	text labels for the points, one for each point; in the default method defaults to the observation indices, in the formula method to the row names of the data.
col	colours for the point labels, given by group. There must be at least as many colours as groups; if there are no groups, the first colour is used. Normally, the colours would correspond to the <code>surface.col</code> argument to <code>scatter3d</code> .
offset	vertical displacement for point labels (to avoid overplotting the points).
...	arguments to be passed down.

**Value**

`scatter3d` does not return a useful value; it is used for its side-effect of creating a 3D scatterplot. `Identify3d` returns the labels of the identified points.

**Note**

You have to install the `rgl` package to produce 3D plots.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[rgl-package](#), [gam](#)

**Examples**

```

if(interactive() && require(rgl) && require(mgcv)){
  scatter3d(prestige ~ income + education, data=Duncan)
  Sys.sleep(5) # wait 5 seconds
  scatter3d(prestige ~ income + education | type, data=Duncan)
  Sys.sleep(5)
  scatter3d(prestige ~ income + education | type, surface=FALSE,
            ellipsoid=TRUE, revolutions=3, data=Duncan)
  scatter3d(prestige ~ income + education, fit=c("linear", "additive"),
            data=Prestige)
  Sys.sleep(5)
  scatter3d(prestige ~ income + education | type,
            radius=(1 + women)^(1/3), data=Prestige)
}
## Not run:
# drag right mouse button to identify points, click right button in open area to exit
scatter3d(prestige ~ income + education, data=Duncan, id.method="identify")
scatter3d(prestige ~ income + education | type, data=Duncan, id.method="identify")

## End(Not run)

```

---

scatterplot

*Scatterplots with Boxplots*


---

## Description

Makes enhanced scatterplots, with boxplots in the margins, a nonparametric regression smooth, smoothed conditional spread, outlier identification, and a regression line; `sp` is an abbreviation for `scatterplot`.

## Usage

```
scatterplot(x, ...)
```

```
## S3 method for class 'formula'
```

```
scatterplot(formula, data, subset, xlab, ylab, legend.title, legend.coords,
labels, ...)
```

```
## Default S3 method:
```

```
scatterplot(x, y,
  smoother=loessLine, smoother.args=list(), smooth, span,
  spread=!by.groups, reg.line=lm,
  boxplots=if (by.groups) "" else "xy",
  xlab=deparse(substitute(x)), ylab=deparse(substitute(y)), las=par("las"),
  lwd=1, lty=1,
  labels, id.method = "mahal",
  id.n = if(id.method[1]=="identify") length(x) else 0,
  id.cex = 1, id.col = palette()[1], id.location="lr",
  log="", jitter=list(), xlim=NULL, ylim=NULL,
  cex=par("cex"), cex.axis=par("cex.axis"), cex.lab=par("cex.lab"),
  cex.main=par("cex.main"), cex.sub=par("cex.sub"),
  groups, by.groups=!missing(groups),
  legend.title=deparse(substitute(groups)), legend.coords, legend.columns,
  ellipse=FALSE, levels=c(.5, .95), robust=TRUE,
  col=if (n.groups == 1) palette()[3:1] else rep(palette(), length=n.groups),
  pch=1:n.groups,
  legend.plot=!missing(groups), reset.par=TRUE, grid=TRUE, ...)
```

```
sp(x, ...)
```

## Arguments

<code>x</code>	vector of horizontal coordinates (or first argument of generic function).
<code>y</code>	vector of vertical coordinates.
<code>formula</code>	a “model” formula, of the form $y \sim x$ or (to plot by groups) $y \sim x \mid z$ , where $z$ evaluates to a factor or other variable dividing the data into groups. If $x$ is a factor, then parallel boxplots are produced using the <a href="#">Boxplot</a> function.

data	data frame within which to evaluate the formula.
subset	expression defining a subset of observations.
smoother	a function to draw a nonparametric-regression smooth; the default is <code>loessLine</code> , which does loess smoothing. The function <code>gamLine</code> fits a generalized additive model and allows including a link and error function. See <code>ScatterplotSmoothers</code> . Setting this argument to something other than a function, e.g., <code>FALSE</code> suppresses the smoother.
smoother.args	a list of named values to be passed to the smoother function; the specified elements of the list depend upon the smoother (see <code>ScatterplotSmoothers</code> ).
smooth, span	these arguments are included for backwards compatibility: if <code>smooth=TRUE</code> then <code>smoother</code> is set to <code>loessLine</code> , and if <code>span</code> is specified, it is added to <code>smoother.args</code> .
spread	if <code>TRUE</code> , estimate the (square root) of the variance function. For <code>loessLine</code> and for <code>gamLine</code> , this is done by separately smoothing the squares of the positive and negative residuals from the mean fit, and then adding the square root of the fitted values to the mean fit. For <code>quantregLine</code> , fit the .25 and .75 quantiles with a quantile regression additive model. The default is <code>TRUE</code> if <code>by.groups=FALSE</code> and <code>FALSE</code> is by <code>by.groups=TRUE</code> .
reg.line	function to draw a regression line on the plot or <code>FALSE</code> not to plot a regression line.
boxplots	if "x" a boxplot for x is drawn below the plot; if "y" a boxplot for y is drawn to the left of the plot; if "xy" both boxplots are drawn; set to "" or <code>FALSE</code> to suppress both boxplots.
xlab	label for horizontal axis.
ylab	label for vertical axis.
las	if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see <code>par</code> ).
lwd	width of linear-regression lines (default 1).
lty	type of linear-regression lines (default 1, solid line).
id.method, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is <code>id.n=0</code> for labeling no points. See <code>showLabels</code> for details of these arguments. If the plot uses different colors for groups, then the <code>id.col</code> argument is ignored and label colors are determined by the <code>col</code> argument.
labels	a vector of point labels; if absent, the function tries to determine reasonable labels, and, failing that, will use observation numbers.
log	same as the <code>log</code> argument to <code>plot</code> , to produce log axes.
jitter	a list with elements x or y or both, specifying jitter factors for the horizontal and vertical coordinates of the points in the scatterplot. The <code>jitter</code> function is used to randomly perturb the points; specifying a factor of 1 produces the default jitter. Fitted lines are unaffected by the jitter.
xlim	the x limits (min, max) of the plot; if <code>NULL</code> , determined from the data.
ylim	the y limits (min, max) of the plot; if <code>NULL</code> , determined from the data.

groups	a factor or other variable dividing the data into groups; groups are plotted with different colors and plotting characters.
by.groups	if TRUE, regression lines are fit by groups.
legend.title	title for legend box; defaults to the name of the groups variable.
legend.coords	coordinates for placing legend; can be a list with components x and y to specify the coordinates of the upper-left-hand corner of the legend; or a quoted keyword, such as "topleft", recognized by <a href="#">legend</a> .
legend.columns	number of columns for the legend; if absent will be supplied automatically to prefer horizontal legends when plotted above the graph.
ellipse	if TRUE data-concentration ellipses are plotted.
levels	level or levels at which concentration ellipses are plotted; the default is <code>c(.5, .95)</code> .
robust	if TRUE (the default) use the <code>cov.trob</code> function in the MASS package to calculate the center and covariance matrix for the data ellipses.
col	colors for lines and points; the default is taken from the color palette, with <code>palette()[3]</code> for linear regression lines, <code>palette()[2]</code> for nonparametric regression lines, and <code>palette()[1]</code> for points if there are no groups, and successive colors for the groups if there are groups.
pch	plotting characters for points; default is the plotting characters in order (see <a href="#">par</a> ).
cex, cex.axis, cex.lab, cex.main, cex.sub	set sizes of various graphical elements; (see <a href="#">par</a> ).
legend.plot	if TRUE then a legend for the groups is plotted in the upper margin.
reset.par	if TRUE then plotting parameters are reset to their previous values when <code>scatterplot</code> exits; if FALSE then the <code>mar</code> and <code>mfcol</code> parameters are altered for the current plotting device. Set to FALSE if you want to add graphical elements (such as lines) to the plot.
...	other arguments passed down and to <code>plot</code> .
grid	If TRUE, the default, a light-gray background grid is put on the graph

**Value**

If points are identified, their labels are returned; otherwise NULL is returned invisibly.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**See Also**

[boxplot](#), [jitter](#), [legend](#), [scatterplotMatrix](#), [dataEllipse](#), [Boxplot](#), [cov.trob](#), [showLabels](#), [ScatterplotSmothers](#).

**Examples**

```

scatterplot(prestige ~ income, data=Prestige, ellipse=TRUE)

if (interactive()){
  scatterplot(prestige ~ income, data=Prestige, smoother=quantregLine)
}

scatterplot(prestige ~ income|type, data=Prestige, smoother=loessLine,
  smoother.args=list(span=1))

scatterplot(prestige ~ income|type, data=Prestige, legend.coords="topleft")

scatterplot(vocabulary ~ education, jitter=list(x=1, y=1),
  data=Vocab, id.n=0, smoother=FALSE)

scatterplot(infant.mortality ~ gdp, log="xy", data=UN, id.n=5)

scatterplot(income ~ type, data=Prestige)

## Not run:
scatterplot(infant.mortality ~ gdp, id.method="identify", data=UN)

scatterplot(infant.mortality ~ gdp, id.method="identify", smoother=loessLine, data=UN)

## End(Not run)

```

---

scatterplotMatrix      *Scatterplot Matrices*

---

**Description**

Enhanced scatterplot matrices with univariate displays down the diagonal; *spm* is an abbreviation for `scatterplotMatrix`. This function just sets up a call to `pairs` with custom panel functions.

**Usage**

```

scatterplotMatrix(x, ...)

## S3 method for class 'formula'
scatterplotMatrix(formula, data=NULL, subset, labels, ...)

## Default S3 method:
scatterplotMatrix(x, var.labels=colnames(x),
  diagonal=c("density", "boxplot", "histogram", "oned", "qqplot", "none"),
  adjust=1, nclass,
  plot.points=TRUE, smoother=loessLine, smoother.args=list(), smooth, span,
  spread = !by.groups, reg.line=lm,

```

```

transform=FALSE, family=c("bcPower", "yjPower"),
ellipse=FALSE, levels=c(.5, .95), robust=TRUE,
groups=NULL, by.groups=FALSE,
use=c("complete.obs", "pairwise.complete.obs"),
labels, id.method="mahal", id.n=0, id.cex=1, id.col=palette()[1], id.location="lr",
col=if (n.groups == 1) palette()[3:1] else rep(palette(), length=n.groups),
pch=1:n.groups, lwd=1, lty=1,
cex=par("cex"), cex.axis=par("cex.axis"), cex.labels=NULL,
cex.main=par("cex.main"),
legend.plot=length(levels(groups)) > 1, legend.pos=NULL, rowlattop=TRUE, ...)

```

```
spm(x, ...)
```

### Arguments

x	a data matrix, numeric data frame.
formula	a one-sided “model” formula, of the form $\sim x_1 + x_2 + \dots + x_k$ or $\sim x_1 + x_2 + \dots + x_k \mid z$ where $z$ evaluates to a factor or other variable to divide the data into groups.
data	for scatterplotMatrix.formula, a data frame within which to evaluate the formula.
subset	expression defining a subset of observations.
labels, id.method, id.n, id.cex, id.col, id.location	Arguments for the labelling of points. The default is id.n=0 for labeling no points. See <a href="#">showLabels</a> for details of these arguments. If the plot uses different colors for groups, then the id.col argument is ignored and label colors are determined by the col argument.
var.labels	variable labels (for the diagonal of the plot).
diagonal	contents of the diagonal panels of the plot. If plotting by groups, a different univariate display (with the exception of “histogram”) will be drawn for each group.
adjust	relative bandwidth for density estimate, passed to density function.
nclass	number of bins for histogram, passed to hist function.
plot.points	if TRUE the points are plotted in each off-diagonal panel.
smoother	a function to draw a nonparametric-regression smooth; the default is <a href="#">gamLine</a> , which uses the <a href="#">gam</a> function in the <a href="#">mgcv</a> package. For this and other smoothers, see <a href="#">ScatterplotSmoothers</a> . Setting this argument to something other than a function, e.g., FALSE suppresses the smoother.
smoother.args	a list of named values to be passed to the smoother function; the specified elements of the list depend upon the smoother (see <a href="#">ScatterplotSmoothers</a> ).
smooth, span	these arguments are included for backwards compatibility: if smooth=TRUE then smoother is set to <a href="#">loessLine</a> , and if span is specified, it is added to smoother.args.
spread	if TRUE, estimate the (square root) of the variance function. For <a href="#">loessLine</a> and for <a href="#">gamLine</a> , this is done by separately smoothing the squares of the positive and negative residuals from the mean fit, and then adding the square root of the fitted

	values to the mean fit. For <code>quantregLine</code> , fit the .25 and .75 quantiles with a quantile regression additive model. The default is TRUE if <code>by.groups=FALSE</code> and FALSE is <code>by.groups=TRUE</code> .
<code>reg.line</code>	if not FALSE a line is plotted using the function given by this argument; e.g., using <code>r1m</code> in package MASS plots a robust-regression line.
<code>transform</code>	if TRUE, multivariate normalizing power transformations are computed with <code>powerTransform</code> , rounding the estimated powers to ‘nice’ values for plotting; if a vector of powers, one for each variable, these are applied prior to plotting. If there are groups and <code>by.groups</code> is TRUE, then the transformations are estimated <i>conditional</i> on the groups factor.
<code>family</code>	family of transformations to estimate: “bcPower” for the Box-Cox family or “yjPower” for the Yeo-Johnson family (see <code>powerTransform</code> ).
<code>ellipse</code>	if TRUE data-concentration ellipses are plotted in the off-diagonal panels.
<code>levels</code>	levels or levels at which concentration ellipses are plotted; the default is <code>c(.5, .9)</code> .
<code>robust</code>	if TRUE use the <code>cov.trob</code> function in the MASS package to calculate the center and covariance matrix for the data ellipses.
<code>groups</code>	a factor or other variable dividing the data into groups; groups are plotted with different colors and plotting characters.
<code>by.groups</code>	if TRUE, regression lines are fit by groups.
<code>use</code>	if “complete.obs” (the default), cases with missing data are omitted; if “pairwise.complete.obs”), a in each panel of the plot.
<code>pch</code>	plotting characters for points; default is the plotting characters in order (see <code>par</code> ).
<code>col</code>	colors for lines and points; the default is taken from the color palette, with <code>palette()[3]</code> for linear regression lines, <code>palette()[2]</code> for nonparametric regression lines, and <code>palette()[1]</code> for points if there are no groups, and successive colors for the groups if there are groups.
<code>lwd</code>	width of linear-regression lines (default 1).
<code>lty</code>	type of linear-regression lines (default 1, solid line).
<code>cex, cex.axis, cex.labels, cex.main</code>	set sizes of various graphical elements (see <code>par</code> ).
<code>legend.plot</code>	if TRUE then a legend for the groups is plotted in the first diagonal cell.
<code>legend.pos</code>	position for the legend, specified as one of the keywords accepted by <code>legend</code> . If NULL, the default, the position will vary by the diagonal argument — e.g., “topright” for <code>diagonal=“density”</code> .
<code>row1atop</code>	If TRUE (the default) the first row is at the top, as in a matrix, as opposed to at the bottom, as in graph (argument suggested by Richard Heiberger).
<code>...</code>	arguments to pass down.

**Value**

NULL. This function is used for its side effect: producing a plot.

**Author(s)**

John Fox <jfox@mcmaster.ca>



## References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

## See Also

[pairs](#), [scatterplot](#), [dataEllipse](#), [powerTransform](#), [bcPower](#), [yjPower](#), [cov.trob](#), [showLabels](#), [ScatterplotSmoothers](#).

## Examples

```
scatterplotMatrix(~ income + education + prestige | type, data=Duncan)
scatterplotMatrix(~ income + education + prestige,
  transform=TRUE, data=Duncan, smoother=loessLine)
scatterplotMatrix(~ income + education + prestige | type, smoother=FALSE,
  by.group=TRUE, transform=TRUE, data=Duncan)
```

---

ScatterplotSmoothers *Smoothers to Draw Lines on Scatterplots*

---

## Description

These smoothers are used to draw nonparametric-regression lines on scatterplots produced by the [scatterplot](#), [scatterplotMatrix](#) and several other car functions. The functions are not meant to be called directly by the user, although the user can supply options via the `smoother.args` argument, the contents of which vary by the smoother (see *Details* below). The `gamLine` smoother uses the `gam` function in the `mgcv` package, the `loessLine` smoother uses the `loess` function in the `stats` package, and the `quantregLine` smoother uses the `rqss` function in the `quantreg` package.

## Usage

```
gamLine(x, y, col, log.x, log.y, spread=FALSE, smoother.args, draw=TRUE, offset=0)
```

```
loessLine(x, y, col, log.x, log.y, spread=FALSE, smoother.args, draw=TRUE, offset=0)
```

```
quantregLine(x, y, col, log.x, log.y, spread=FALSE, smoother.args, draw=TRUE, offset=0)
```

## Arguments

<code>x</code>	<code>\$x\$</code> coordinates of points.
<code>y</code>	<code>\$y\$</code> coordinates of points.
<code>col</code>	line color.
<code>log.x</code>	TRUE if the <code>\$x\$</code> -axis is logged.
<code>log.y</code>	TRUE if the <code>\$y\$</code> -axis is logged.
<code>spread</code>	the default is to plot only an estimated mean or median function. If this argument is TRUE, then a measure of spread is also plotted.

<code>smoother.args</code>	additional options accepted by the smoother, in the form of a list of named values (see <i>Details</i> below).
<code>draw</code>	if TRUE, the default, draw the smoother on the currently active graph. If FALSE, return a list with coordinates <code>x</code> and <code>y</code> for the points that make up the smooth and if requested <code>x.pos</code> , <code>y.pos</code> , <code>x.neg</code> , <code>y.neg</code> for the spread smooths.
<code>offset</code>	For use when <code>spread=TRUE</code> , the vertical axis is $\sqrt{\text{offset}^2 + \text{variance smooth}}$ .

## Details

The function `loessLine` is a re-implementation of the loess smoother that was used in `car` prior to September 2012. The main enhancement is the ability to set more arguments through the `smoother.args` argument.

The function `gamLine` is more general than the loess fitting because it allows fitting a generalized additive model using splines. You can specify an error distribution and link function.

The function `quantregLine` fits an additive model using splines with estimation based on L1 regression for the median and quantile regression if you ask for the spread. It is likely to be more robust than the other smoothers.

The argument `smoother.args` is a list of named elements used to pass additional arguments to the smoother. As of November, 2016, the smoother is evaluated at an equally spaced grid of 50 points in the range of the horizontal variable. With any of the smoothers you can change to say 100 evaluation points by using the argument `smoother.args=list(evaluation=100)`.

For `loessLine` the default value is `smoother.args=list(lty=1, lwd=2, lty.spread=2, lwd.spread=1, span=2/3)` (probably `span=2/3`). The arguments `lty` and `lwd` are the type and width respectively of the mean or median smooth, `smooth.lty` and `smooth.lwd` are the type and color of the spread smooths if requested. The arguments `span`, `degree` and `family` are passed to the `loess` function, `iterations=4` robustness iterations.

For `gamLine` the default is `smoother.args=list(lty=1, lwd=2, lty.spread=2, lwd.spread=1, k=-1, bs="tp", family="gaussian")`. The first four arguments are as for `loessLine`. The next two arguments are passed to the `gam` function to control the smoothing: `k=-1` allows `gam` to choose the number of splines in the basis function; `bs="tp"` provides the type of spline basis to be used with "tp" for the default thin-plate splines. The last three arguments allow providing a family, link and weights as in generalized linear models. See examples below. The spread argument is ignored unless `family="gaussian"` and `link=NULL`.

For `quantregLine` the default is `smoother.args=list(lty=1, lwd=2, lty.spread=2, lwd.spread=1, lambda=IQR(x))`. The first four arguments are as for `loessLine`. The last argument is passed to the `qss` function in `quantreg`. It is a smoothing parameter, here a robust estimate of the scale of the horizontal axis variable. This is an arbitrary choice, and may not work well in all circumstances.

## Author(s)

John Fox <jfox@mcmaster.ca> and Sanford Weisberg <sandy@umn.edu>.

## See Also

[scatterplot](#), [scatterplotMatrix](#), [gam](#), [loess](#), and [rqss](#).

**Examples**

```

scatterplot(prestige ~ income, data=Prestige)
scatterplot(prestige ~ income, data=Prestige, smoother=gamLine)
scatterplot(prestige ~ income, data=Prestige, smoother=quantregLine)

scatterplot(prestige ~ income | type, data=Prestige)
scatterplot(prestige ~ income | type, data=Prestige, smoother=gamLine)
scatterplot(prestige ~ income | type, data=Prestige, smoother=quantregLine)
scatterplot(prestige ~ income | type, data=Prestige, smoother=NULL)

scatterplot(prestige ~ income | type, data=Prestige, spread=TRUE)
scatterplot(prestige ~ income | type, data=Prestige, smoother=gamLine, spread=TRUE)
scatterplot(prestige ~ income | type, data=Prestige, smoother=quantregLine, spread=TRUE)

scatterplot(weight ~ repwt | sex, spread=TRUE, data=Davis, smoother=loessLine)
scatterplot(weight ~ repwt | sex, spread=TRUE, data=Davis, smoother=gamLine) # messes up
scatterplot(weight ~ repwt | sex, spread=TRUE, data=Davis, smoother=quantregLine) # robust
set.seed(12345)
w <- 1 + rpois(100, 5)
x <- rnorm(100)
p <- 1/(1 + exp(-(x + 0.5*x^2)))
y <- rbinom(100, w, p)
scatterplot(y/w ~ x, smoother=gamLine,
            smoother.args=list(family="binomial", weights=w))
scatterplot(y/w ~ x, smoother=gamLine,
            smoother.args=list(family=binomial, link="probit", weights=w))
scatterplot(y/w ~ x, smoother=gamLine,
            smoother.args=list(family=binomial, link="probit", weights=w))
scatterplot(y/w ~ x, smoother=loessLine, reg=FALSE)

y <- rbinom(100, 1, p)
scatterplot(y ~ x, smoother=gamLine, smoother.args=list(family=binomial))

```

---

showLabels

*Utility Functions to Identify and Mark Extreme Points in a 2D Plot.*


---

**Description**

This function is called by several graphical functions in the `car` package to mark extreme points in a 2D plot. Although the user is unlikely to call this function directly, the documentation below applies to all these other functions.

**Usage**

```

showLabels(x, y, labels=NULL, id.method="identify",
           id.n = length(x), id.cex=1, id.col=palette()[1], id.location="lr", ...)

```

## Arguments

<code>x</code>	Plotted horizontal coordinates.
<code>y</code>	Plotted vertical coordinates.
<code>labels</code>	Plotting labels. If NULL, case numbers will be used. If labels are long, the <code>substr</code> or <code>abbreviate</code> function can be used to shorten them.
<code>id.method</code>	How points are to be identified. See Details below.
<code>id.n</code>	Number of points to be identified. If set to zero, no points are identified.
<code>id.cex</code>	Controls the size of the plotted labels. The default is 1.
<code>id.col</code>	Controls the color of the plotted labels.
<code>id.location</code>	Where should the label be drawn? The default is "lr" to draw the label to the left of the point for points in the right-half of the graph and to the right for points in the left-half. The other option is "ab" for above the point for points below the middle of the graph and above the point below the middle.
<code>...</code>	additional arguments passed to <code>identify</code> or to <code>text</code> .

## Details

The argument `id.method` determine how the points to be identified are selected. For the default value of `id.method="identify"`, the `identify` function is used to identify points interactively using the mouse. Up to `id.n` points can be identified, so if `id.n=0`, which is the default in many functions in the `car` package, then no point identification is done.

Automatic point identification can be done depending on the value of the argument `id.method`.

- `id.method = "x"` select points according to their value of  $\text{abs}(x - \text{mean}(x))$
- `id.method = "y"` select points according to their value of  $\text{abs}(y - \text{mean}(y))$
- `id.method = "r"` select points according to their value of  $\text{abs}(y)$ , as may be appropriate in residual plots, or others with a meaningful origin at 0
- `id.method = "mahal"` Treat  $(x, y)$  as if it were a bivariate sample, and select cases according to their Mahalanobis distance from  $(\text{mean}(x), \text{mean}(y))$
- `id.method` can be a vector of the same length as `x` consisting of values to determine the points to be labeled. For example, for a linear model `m`, setting `id.method=cooks.distance(m)`, `id.n=4` will label the points corresponding to the four largest values of Cook's distance, or `id.method = which(abs(residual) > 2)` would label all observations with Pearson residuals greater than 2 in absolute value. Warning: If missing data are present, points may be incorrectly labelled.
- `id.method` can be a vector of case numbers or case-labels, in which case those cases will be labeled. Warning: If missing data are present, a list of case numbers may identify the wrong points. A list of case labels, however, will work correctly with missing values.

With `showLabels`, the `id.method` argument can be a list, so, for example `id.method=list("x", "y")` would label according to the horizontal and vertical axes variables.

Finally, if the axes in the graph are logged, the function uses logged-variables where appropriate.

**Value**

A utility function primarily used for its side-effect of drawing labels on a plot. Returns invisibly the labels of the selected points, or NULL if no points are selected. Although intended for use with other functions in the car package, this function can be used directly.

**Author(s)**

John Fox <jfox@mcmaster.ca>, Sanford Weisberg <sandy@umn.edu>

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.  
Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

**See Also**

[avPlots](#), [residualPlots](#), [crPlots](#), [leveragePlots](#)

**Examples**

```
plot(income ~ education, Prestige)
with(Prestige, showLabels(education, income,
  labels = rownames(Prestige), id.method=list("x", "y"), id.n=3))
m <- lm(income ~ education, Prestige)
plot(income ~ education, Prestige)
abline(m)
with(Prestige, showLabels(education, income,
  labels=rownames(Prestige), id.method=abs(residuals(m)), id.n=4))
```

---

sigmaHat

*Return the scale estimate for a regression model*

---

**Description**

This function provides a consistent method to return the estimated scale from a linear, generalized linear, nonlinear, or other model.

**Usage**

```
sigmaHat(object)
```

**Arguments**

object            A regression object of type lm, glm or nls

**Details**

For an lm or nls object, the returned quantity is the square root of the estimate of  $\sigma$ . For a glm object, the returned quantity is the square root of the estimated dispersion parameter.

**Value**

A nonnegative number

**Author(s)**

Sanford Weisberg, <sandy@umn.edu>

**Examples**

```
m1 <- lm(prestige ~ income + education, data=Duncan)
sigmaHat(m1)
```

---

SLID

*Survey of Labour and Income Dynamics*

---

**Description**

The SLID data frame has 7425 rows and 5 columns. The data are from the 1994 wave of the Canadian Survey of Labour and Income Dynamics, for the province of Ontario. There are missing data, particularly for wages.

**Usage**

SLID

**Format**

This data frame contains the following columns:

**wages** Composite hourly wage rate from all jobs.

**education** Number of years of schooling.

**age** in years.

**sex** A factor with levels: Female, Male.

**language** A factor with levels: English, French, Other.

**Source**

The data are taken from the public-use dataset made available by Statistics Canada, and prepared by the Institute for Social Research, York University.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Description**

Soil characteristics were measured on samples from three types of contours (Top, Slope, and Depression) and at four depths (0-10cm, 10-30cm, 30-60cm, and 60-90cm). The area was divided into 4 blocks, in a randomized block design. (Suggested by Michael Friendly.)

**Usage**

Soils

**Format**

A data frame with 48 observations on the following 14 variables. There are 3 factors and 9 response variables.

Group a factor with 12 levels, corresponding to the combinations of Contour and Depth

Contour a factor with 3 levels: Depression Slope Top

Depth a factor with 4 levels: 0-10 10-30 30-60 60-90

Gp a factor with 12 levels, giving abbreviations for the groups: D0 D1 D3 D6 S0 S1 S3 S6 T0 T1 T3 T6

Block a factor with levels 1 2 3 4

pH soil pH

N total nitrogen in %

Dens bulk density in gm/cm<sup>3</sup>

P total phosphorous in ppm

Ca calcium in me/100 gm.

Mg magnesium in me/100 gm.

K phosphorous in me/100 gm.

Na sodium in me/100 gm.

Conduc conductivity

**Details**

These data provide good examples of MANOVA and canonical discriminant analysis in a somewhat complex multivariate setting. They may be treated as a one-way design (ignoring Block), by using either Group or Gp as the factor, or a two-way randomized block design using Block, Contour and Depth (quantitative, so orthogonal polynomial contrasts are useful).

**Source**

Horton, I. F., Russell, J. S., and Moore, A. W. (1968) Multivariate-covariance and canonical analysis: A method for selecting the most effective discriminators in a multivariate situation. *Biometrics* **24**, 845–858. Originally from 'http://www.stat.lsu.edu/faculty/moser/exst7037/soils.sas' but no longer available there.

**References**

Khattree, R., and Naik, D. N. (2000) *Multivariate Data Reduction and Discrimination with SAS Software*. SAS Institute.

Friendly, M. (2006) Data ellipses, HE plots and reduced-rank displays for multivariate linear models: SAS software and examples. *Journal of Statistical Software*, 17(6), <http://www.jstatsoft.org/v17/i06>.

---

some

*Sample a Few Elements of an Object*

---

**Description**

Randomly select a few elements of an object, typically a data frame, matrix, vector, or list. If the object is a data frame or a matrix, then rows are sampled.

**Usage**

```
some(x, ...)
```

```
## S3 method for class 'data.frame'
```

```
some(x, n=10, ...)
```

```
## S3 method for class 'matrix'
```

```
some(x, n=10, ...)
```

```
## Default S3 method:
```

```
some(x, n=10, ...)
```

**Arguments**

x	the object to be sampled.
n	number of elements to sample.
...	arguments passed down.

**Value**

Sampled elements or rows.



**Note**

These functions are adapted from head and tail in the utils package.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

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

**Examples**

```
some(Duncan)
```

---

spreadLevelPlot	<i>Spread-Level Plots</i>
-----------------	---------------------------

---

**Description**

Creates plots for examining the possible dependence of spread on level, or an extension of these plots to the studentized residuals from linear models.

**Usage**

```
spreadLevelPlot(x, ...)

slp(...)

## S3 method for class 'formula'
spreadLevelPlot(x, data=NULL, subset, na.action,
  main=paste("Spread-Level Plot for", varnames[response],
    "by", varnames[-response]), ...)

## Default S3 method:
spreadLevelPlot(x, by, robust.line=TRUE,
  start=0, xlab="Median", ylab="Hinge-Spread", point.labels=TRUE, las=par("las"),
  main=paste("Spread-Level Plot for", deparse(substitute(x)),
    "by", deparse(substitute(by))), col=palette()[1], col.lines=palette()[2],
  pch=1, lwd=2, grid=TRUE, ...)

## S3 method for class 'lm'
spreadLevelPlot(x, robust.line=TRUE,
```

```

    smoother=loessLine, smoother.args=list(),
    xlab="Fitted Values",
    ylab="Absolute Studentized Residuals", las=par("las"),
    main=paste("Spread-Level Plot for\n", deparse(substitute(x))),
    pch=1, col=palette()[1], col.lines=palette()[2], col.smoother=palette()[3],
    lwd=2, grid=TRUE, labels,
    id.method = "mahal",
    id.n = if(id.method[1]=="identify") Inf else 0,
    id.cex=1, id.col=palette()[1], id.location="lr", ...)

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

```

### Arguments

<code>x</code>	a formula of the form $y \sim x$ , where $y$ is a numeric vector and $x$ is a factor, or an <code>lm</code> object to be plotted; alternatively a numeric vector.
<code>data</code>	an optional data frame containing the variables to be plotted. By default the variables are taken from the environment from which <code>spreadLevelPlot</code> is called.
<code>subset</code>	an optional vector specifying a subset of observations to be used.
<code>na.action</code>	a function that indicates what should happen when the data contain NAs. The default is set by the <code>na.action</code> setting of options.
<code>by</code>	a factor, numeric vector, or character vector defining groups.
<code>robust.line</code>	if <code>TRUE</code> a robust line is fit using the <code>r1m</code> function in the <code>MASS</code> package; if <code>FALSE</code> a line is fit using <code>lm</code> .
<code>smoother</code>	a function to draw a nonparametric-regression smooth; the default is <code>loessLine</code> , which does loess smoothing. The function <code>gamLine</code> fits a generalized additive model and allows including a link and error function. See <a href="#">ScatterplotSmoothers</a> . Setting this argument to something other than a function, e.g., <code>FALSE</code> suppresses the smoother.
<code>smoother.args</code>	a list of named values to be passed to the smoother function; the specified elements of the list depend upon the smoother (see <a href="#">ScatterplotSmoothers</a> ).
<code>start</code>	add the constant <code>start</code> to each data value.
<code>main</code>	title for the plot.
<code>xlab</code>	label for horizontal axis.
<code>ylab</code>	label for vertical axis.
<code>point.labels</code>	if <code>TRUE</code> label the points in the plot with group names.
<code>las</code>	if <code>0</code> , ticks labels are drawn parallel to the axis; set to <code>1</code> for horizontal labels (see <a href="#">par</a> ).
<code>col</code>	color for points; the default is the first entry in the current color palette (see <a href="#">palette</a> and <a href="#">par</a> ).
<code>col.lines</code>	color for lines; default is the second entry in the current palette
<code>col.smoother</code>	color for smooth line; default is third entry in the current palette.
<code>pch</code>	plotting character for points; default is <code>1</code> (a circle, see <a href="#">par</a> ).

`lwd` line width; default is 2 (see [par](#)).  
`grid` If TRUE, the default, a light-gray background grid is put on the graph  
`id.method, labels, id.n, id.cex, id.col, id.location`  
Arguments for the labelling of points. The default is `id.n=0` for labeling no points with the `lm` method. See [showLabels](#) for details of these arguments.  
`...` arguments passed to plotting functions.

### Details

Except for linear models, computes the statistics for, and plots, a Tukey spread-level plot of  $\log(\text{hinge-spread})$  vs.  $\log(\text{median})$  for the groups; fits a line to the plot; and calculates a spread-stabilizing transformation from the slope of the line.

For linear models, plots  $\log(\text{abs}(\text{studentized residuals}))$  vs.  $\log(\text{fitted values})$ . Point labeling was added in November, 2016.

The function `slp` is an abbreviation for `spreadLevelPlot`.

### Value

An object of class `spreadLevelPlot` containing:

`Statistics` a matrix with the lower-hinge, median, upper-hinge, and hinge-spread for each group. (Not for an `lm` object.)  
`PowerTransformation`  
spread-stabilizing power transformation, calculated as  $1 - \text{slope}$  of the line fit to the plot.

### Author(s)

John Fox <[jfox@mcmaster.ca](mailto:jfox@mcmaster.ca)>

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.  
Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.  
Hoaglin, D. C., Mosteller, F. and Tukey, J. W. (Eds.) (1983) *Understanding Robust and Exploratory Data Analysis*. Wiley.

### See Also

[hccm](#), [ncvTest](#)

### Examples

```
spreadLevelPlot(interlocks + 1 ~ nation, data=Ornstein)
slp(lm(interlocks + 1 ~ assets + sector + nation, data=Ornstein))
```

---

States

*Education and Related Statistics for the U.S. States*

---

### Description

The States data frame has 51 rows and 8 columns. The observations are the U. S. states and Washington, D. C.

### Usage

States

### Format

This data frame contains the following columns:

**region** U. S. Census regions. A factor with levels: ENC, East North Central; ESC, East South Central; MA, Mid-Atlantic; MTN, Mountain; NE, New England; PAC, Pacific; SA, South Atlantic; WNC, West North Central; WSC, West South Central.

**pop** Population: in 1,000s.

**SATV** Average score of graduating high-school students in the state on the *verbal* component of the Scholastic Aptitude Test (a standard university admission exam).

**SATM** Average score of graduating high-school students in the state on the *math* component of the Scholastic Aptitude Test.

**percent** Percentage of graduating high-school students in the state who took the SAT exam.

**dollars** State spending on public education, in \ \$1000s per student.

**pay** Average teacher's salary in the state, in \$1000s.

### Source

United States (1992) *Statistical Abstract of the United States*. Bureau of the Census.

### References

Moore, D. (1995) *The Basic Practice of Statistics*. Freeman, Table 2.1.

subsets

*Plot Output from regsubsets Function in leaps package***Description**

The `regsubsets` function in the **leaps** package finds optimal subsets of predictors. This function plots a measure of fit (see the `statistic` argument below) against subset size.

**Usage**

```
subsets(object, ...)

## S3 method for class 'regsubsets'
subsets(object,
  names=abbreviate(object$xnames, minlength = abbrev),
  abbrev=1, min.size=1, max.size=length(names),
  legend="interactive",
  statistic=c("bic", "cp", "adjr2", "rsq", "rss"),
  las=par('las'), cex.subsets=1, ...)
```

**Arguments**

<code>object</code>	a <code>regsubsets</code> object produced by the <code>regsubsets</code> function in the <b>leaps</b> package.
<code>names</code>	a vector of (short) names for the predictors, excluding the regression intercept, if one is present; if missing, these are derived from the predictor names in <code>object</code> .
<code>abbrev</code>	minimum number of characters to use in abbreviating predictor names.
<code>min.size</code>	minimum size subset to plot; default is 1.
<code>max.size</code>	maximum size subset to plot; default is number of predictors.
<code>legend</code>	If not <code>FALSE</code> , in which case the legend is suppressed, the coordinates at which to place a legend of the abbreviated predictor names on the plot, in a form recognized by the <code>legend</code> function. If <code>"interactive"</code> , the legend is placed on the plot interactively with the mouse. By expanding the left or right plot margin, you can place the legend in the margin, if you wish (see <code>par</code> ).
<code>statistic</code>	statistic to plot for each predictor subset; one of: <code>"bic"</code> , Bayes Information Criterion; <code>"cp"</code> , Mallows's $C_p$ ; <code>"adjr2"</code> , $R^2$ adjusted for degrees of freedom; <code>"rsq"</code> , unadjusted $R^2$ ; <code>"rss"</code> , residual sum of squares.
<code>las</code>	if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see <code>par</code> ).
<code>cex.subsets</code>	can be used to change the relative size of the characters used to plot the regression subsets; default is 1.
<code>...</code>	arguments to be passed down to <code>subsets.regsubsets</code> and <code>plot</code> .

**Value**

NULL if the legend is TRUE; otherwise a data frame with the legend.

**Author(s)**

John Fox

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**See Also**

[regsubsets](#)

**Examples**

```
if (require(leaps)){
  subsets(regsubsets(undercount ~ ., data=Ericksen),
          legend=c(3.5, -37))
}
```

---

symbol

*Boxplots for transformations to symmetry*

---

**Description**

symbol first transforms  $x$  to each of a series of selected powers, with each transformation standardized to mean 0 and standard deviation 1. The results are then displayed side-by-side in boxplots, permitting a visual assessment of which power makes the distribution reasonably symmetric.

**Usage**

```
symbol(x, ...)
## S3 method for class 'formula'
symbol(formula, data=NULL, subset, na.action=NULL, ylab, ...)
## Default S3 method:
symbol(x, powers = c(-1, -0.5, 0, 0.5, 1), start=0,
       trans=bcPower, xlab="Powers", ylab, ...)
```

**Arguments**

x	a numeric vector.
formula	a one-sided formula specifying a single numeric variable.
data, subset, na.action	as for statistical modeling functions (see, e.g., <a href="#">lm</a> ).
xlab, ylab	axis labels; if ylab is missing, a label will be supplied.
powers	a vector of selected powers to which x is to be raised. For meaningful comparison of powers, 1 should be included in the vector of powers.
start	a constant to be added to x.
trans	a transformation function whose first argument is a numeric vector and whose second argument is a transformation parameter, given by the powers argument; the default is <a href="#">bcPower</a> , and another possibility is <a href="#">yjPower</a> .
...	arguments to be passed down.

**Value**

as returned by `boxplot`.

**Author(s)**

Gregor Gorjanc, John Fox <jfox@mcmaster.ca>, and Sanford Weisberg.

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition. Sage.

**See Also**

[boxplot](#), [boxcox](#), [bcPower](#), [yjPower](#)

**Examples**

```
symbol(~ income, data=Prestige)
```

---

testTransform	<i>Likelihood-Ratio Tests for Univariate or Multivariate Power Transformations to Normality</i>
---------------	---

---

**Description**

`testTransform` computes likelihood ratio tests for particular values of the power parameter based on `powerTransform` objects.

**Usage**

```
testTransform(object, lambda)

## S3 method for class 'powerTransform'
testTransform(object, lambda=rep(1, dim(object$y)[2]))

## S3 method for class 'lmerModpowerTransform'
testTransform(object, lambda=1)

## S3 method for class 'bcnPowerTransformlmer'
testTransform(object, lambda=1)
```

**Arguments**

**object**            An object created by a call to `powerTransform`.

**lambda**            A vector of powers of length equal to the number of variables transformed.

**Details**

The function `powerTransform` is used to estimate a power transformation for a univariate or multivariate sample or multiple linear regression problem, using the method of Box and Cox (1964). It is usual to round the estimates to nearby convenient values, and this function is used to compute a likelihood ratio test for values of the transformation parameter other than the ml-type estimate.

For one-parameter families of transformations, namely the Box-Cox power family `bcPower` and the Yeo-Johnson power family `yjPower`, this function computes a test based on twice the difference in the log-likelihood between the maximum likelihood-like estimate and the log-likelihood evaluated at the value of `lambda` specified.

For the `bcnPower` Box-Cox power with negatives allowed, the test is based on the profile loglikelihood maximizing over the location (or `gamma`) parameter(s). Thus, `gamma` is treated as a nuisance parameter.

**Value**

A data frame with one row giving the value of the test statistic, its degrees of freedom, and a p-value. The test is the likelihood ratio test, comparing the value of the log-likelihood at the hypothesized value to the value of the log-likelihood at the maximum likelihood estimate.

**Author(s)**

Sanford Weisberg, <sandy@umn.edu>

**References**

Box, G. E. P. and Cox, D. R. (1964) An analysis of transformations. *Journal of the Royal Statistical Society, Series B*. 26 211-46.

Cook, R. D. and Weisberg, S. (1999) *Applied Regression Including Computing and Graphics*. Wiley.



Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley.

### See Also

[powerTransform](#) and [bcnPower](#) for examples of the use of this function and other tests that might be of interest in some circumstances.

### Examples

```
summary(a3 <- powerTransform(cbind(len, adt, trks, sigs1) ~ htype, Highway1))
# test lambda = (0 0 0 -1)
testTransform(a3, c(0, 0, 0, -1))
summary(q1 <- powerTransform(lm(cbind(LoBD$I1L2, LoBD$I1L1) ~ pool, LoBD), family="bcnPower"))
testTransform(q1, c(.3, .8))
```

---

Transact

*Transaction data*

---

### Description

Data on transaction times in branch offices of a large Australian bank.

### Usage

Transact

### Format

This data frame contains the following columns:

**t1** number of type 1 transactions

**t2** number of type 2 transactions

**time** total transaction time, minutes

### Source

Cunningham, R. and Heathcote, C. (1989), Estimating a non-Gaussian regression model with multicollinearity. *Australian Journal of Statistics*, 31,12-17.

### References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley, Section 4.6.1.

### Description

These functions produce axes for the original scale of transformed variables. Typically these would appear as additional axes to the right or at the top of the plot, but if the plot is produced with `axes=FALSE`, then these functions could be used for axes below or to the left of the plot as well.

### Usage

```
basicPowerAxis(power, base=exp(1),
  side=c("right", "above", "left", "below"),
  at, start=0, lead.digits=1, n.ticks, grid=FALSE, grid.col=gray(0.50),
  grid.lty=2,
  axis.title="Untransformed Data", cex=1, las=par("las"))
```

```
bcPowerAxis(power, side=c("right", "above", "left", "below"),
  at, start=0, lead.digits=1, n.ticks, grid=FALSE, grid.col=gray(0.50),
  grid.lty=2,
  axis.title="Untransformed Data", cex=1, las=par("las"))
```

```
yjPowerAxis(power, side=c("right", "above", "left", "below"),
  at, lead.digits=1, n.ticks, grid=FALSE, grid.col=gray(0.50),
  grid.lty=2,
  axis.title="Untransformed Data", cex=1, las=par("las"))
```

```
probabilityAxis(scale=c("logit", "probit"),
  side=c("right", "above", "left", "below"),
  at, lead.digits=1, grid=FALSE, grid.lty=2, grid.col=gray(0.50),
  axis.title = "Probability", interval = 0.1, cex = 1, las=par("las"))
```

### Arguments

<code>power</code>	power for Box-Cox, Yeo-Johnson, or simple power transformation.
<code>scale</code>	transformation used for probabilities, "logit" (the default) or "probit".
<code>side</code>	side at which the axis is to be drawn; numeric codes are also permitted: <code>side = 1</code> for the bottom of the plot, <code>side=2</code> for the left side, <code>side = 3</code> for the top, <code>side = 4</code> for the right side.
<code>at</code>	numeric vector giving location of tick marks on original scale; if missing, the function will try to pick nice locations for the ticks.
<code>start</code>	if a <i>start</i> was added to a variable (e.g., to make all data values positive), it can now be subtracted from the tick labels.
<code>lead.digits</code>	number of leading digits for determining 'nice' numbers for tick labels (default is 1).

<code>n.ticks</code>	number of tick marks; if missing, same as corresponding transformed axis.
<code>grid</code>	if TRUE grid lines for the axis will be drawn.
<code>grid.col</code>	color of grid lines.
<code>grid.lty</code>	line type for grid lines.
<code>axis.title</code>	title for axis.
<code>cex</code>	relative character expansion for axis label.
<code>las</code>	if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see <a href="#">par</a> ).
<code>base</code>	base of log transformation for <code>power.axis</code> when <code>power = 0</code> .
<code>interval</code>	desired interval between tick marks on the probability scale.

### Details

The transformations corresponding to the three functions are as follows:

`basicPowerAxis`: Simple power transformation,  $x' = x^p$  for  $p \neq 0$  and  $x' = \log x$  for  $p = 0$ .

`bcPowerAxis`: Box-Cox power transformation,  $x' = (x^\lambda - 1)/\lambda$  for  $\lambda \neq 0$  and  $x' = \log x$  for  $\lambda = 0$ .

`yjPowerAxis`: Yeo-Johnson power transformation, for non-negative  $x$ , the Box-Cox transformation of  $x + 1$ ; for negative  $x$ , the Box-Cox transformation of  $|x| + 1$  with power  $2 - p$ .

`probabilityAxis`: logit or probit transformation,  $\text{logit} = \log[p/(1 - p)]$ , or  $\text{probit} = \Phi^{-1}(p)$ , where  $\Phi^{-1}$  is the standard-normal quantile function.

These functions will try to place tick marks at reasonable locations, but producing a good-looking graph sometimes requires some fiddling with the `at` argument.

### Value

These functions are used for their side effects: to draw axes.

### Author(s)

John Fox <jfox@mcmaster.ca>

### References

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

### See Also

[basicPower](#), [bcPower](#), [yjPower](#), [logit](#).

## Examples

```
UN <- na.omit(UN)
par(mar=c(5, 4, 4, 4) + 0.1) # leave space on right

with(UN, plot(log(gdp, 10), log(infant.mortality, 10)))
basicPowerAxis(0, base=10, side="above",
  at=c(50, 200, 500, 2000, 5000, 20000), grid=TRUE,
  axis.title="GDP per capita")
basicPowerAxis(0, base=10, side="right",
  at=c(5, 10, 20, 50, 100), grid=TRUE,
  axis.title="infant mortality rate per 1000")

with(UN, plot(bcPower(gdp, 0), bcPower(infant.mortality, 0)))
bcPowerAxis(0, side="above",
  grid=TRUE, axis.title="GDP per capita")
bcPowerAxis(0, side="right",
  grid=TRUE, axis.title="infant mortality rate per 1000")

with(UN, qqPlot(logit(infant.mortality/1000)))
probabilityAxis()

with(UN, qqPlot(qnorm(infant.mortality/1000)))
probabilityAxis(at=c(.005, .01, .02, .04, .08, .16), scale="probit")
```

---

 UN

*GDP and Infant Mortality*


---

## Description

The UN data frame has 207 rows and 2 columns. The data are for 1998 and are from the United Nations; the observations are nations of the world. There are some missing data.

## Usage

```
UN
```

## Format

This data frame contains the following columns:

**infant.mortality** Infant mortality rate, infant deaths per 1000 live births.

**gdp** GDP per capita, in U.S.~dollars.

## Source

United Nations (1998) Social indicators. Originally from ‘<http://www.un.org/Depts/unsd/social/main.htm>’ but no longer there.

**References**

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.  
Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

USPop

*Population of the United States*

---

**Description**

The USPop data frame has 22 rows and 1 columns. This is a decennial time-series, from 1790 to 2000.

**Usage**

USPop

**Format**

This data frame contains the following columns:

**year** census year.

**population** Population in millions.

**Source**

U.S.-Census Bureau: <http://www.census-charts.com/Population/pop-us-1790-2000.html>, downloaded 1 May 2008.

**References**

- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

---

vif

*Variance Inflation Factors*

---

**Description**

Calculates variance-inflation and generalized variance-inflation factors for linear and generalized linear models.

**Usage**

```
vif(mod, ...)
```

```
## Default S3 method:
```

```
vif(mod, ...)
```

**Arguments**

<code>mod</code>	an object that responds to <code>coef</code> , <code>vcov</code> , and <code>model.matrix</code> , such as an <code>lm</code> or <code>glm</code> object.
<code>...</code>	not used.

**Details**

If all terms in an unweighted linear model have 1 df, then the usual variance-inflation factors are calculated.

If any terms in an unweighted linear model have more than 1 df, then generalized variance-inflation factors (Fox and Monette, 1992) are calculated. These are interpretable as the inflation in size of the confidence ellipse or ellipsoid for the coefficients of the term in comparison with what would be obtained for orthogonal data.

The generalized vifs are invariant with respect to the coding of the terms in the model (as long as the subspace of the columns of the model matrix pertaining to each term is invariant). To adjust for the dimension of the confidence ellipsoid, the function also prints  $GVI\dot{F}^{1/(2 \times df)}$  where  $df$  is the degrees of freedom associated with the term.

Through a further generalization, the implementation here is applicable as well to other sorts of models, in particular weighted linear models and generalized linear models.

**Value**

A vector of vifs, or a matrix containing one row for each term in the model, and columns for the GVIF, df, and  $GVI\dot{F}^{1/(2 \times df)}$ .

**Author(s)**

Henric Nilsson and John Fox <jfox@mcmaster.ca>

**References**

- Fox, J. and Monette, G. (1992) Generalized collinearity diagnostics. *JASA*, **87**, 178–183.
- Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.
- Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Examples**

```
vif(lm

```
prestige ~ income + education, data=Duncan))
vif(lm

```
prestige ~ income + education + type, data=Duncan))
```


```


```

---

Vocab

*Vocabulary and Education*

---

### Description

The Vocab data frame has 21,638 rows and 5 columns. The observations are respondents to U.S. General Social Surveys, 1972-2004.

### Usage

Vocab

### Format

This data frame contains the following columns:

**year** Year of the survey.

**sex** Sex of the respondent, Female or Male.

**education** Education, in years.

**vocabulary** Vocabulary test score: number correct on a 10-word test.

### Source

National Opinion Research Center *General Social Survey*. GSS Cumulative Datafile 1972-2004, downloaded from <http://sda.berkeley.edu/archive.htm>.

### References

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

wcrossprod

*Weighted Matrix Crossproduct*

---

### Description

Given matrices  $x$  and  $y$  as arguments and an optional matrix or vector of weights,  $w$ , return a weighted matrix cross-product,  $t(x) w y$ . If no weights are supplied, or the weights are constant, the function uses `crossprod` for speed.

### Usage

wcrossprod(x, y, w)

**Arguments**

`x, y` `x, y` numeric matrices; `missing(y)` is taken to be the same matrix as `x`. Vectors are promoted to single-column or single-row matrices, depending on the context.

`w` A numeric vector or matrix of weights, conformable with `x` and `y`.

**Value**

A numeric matrix, with appropriate dimnames taken from `x` and `y`.

**Author(s)**

Michael Friendly, John Fox <jfox@mcmaster.ca>

**See Also**

[crossprod](#)

**Examples**

```
set.seed(12345)
n <- 24
drop <- 4
sex <- sample(c("M", "F"), n, replace=TRUE)
x1 <- 1:n
x2 <- sample(1:n)
extra <- c( rep(0, n - drop), floor(15 + 10 * rnorm(drop)) )
y1 <- x1 + 3*x2 + 6*(sex=="M") + floor(10 * rnorm(n)) + extra
y2 <- x1 - 2*x2 - 8*(sex=="M") + floor(10 * rnorm(n)) + extra
# assign non-zero weights to 'dropped' obs
wt <- c(rep(1, n-drop), rep(.2,drop))

X <- cbind(x1, x2)
Y <- cbind(y1, y2)
wcrossprod(X)
wcrossprod(X, w=wt)

wcrossprod(X, Y)
wcrossprod(X, Y, w=wt)

wcrossprod(x1, y1)
wcrossprod(x1, y1, w=wt)
```



**Description**

Contrived data on weight loss and self esteem over three months, for three groups of individuals: Control, Diet and Diet + Exercise. The data constitute a double-multivariate design.

**Usage**

```
WeightLoss
```

**Format**

A data frame with 34 observations on the following 7 variables.

group a factor with levels Control Diet DietEx.

w11 Weight loss at 1 month

w12 Weight loss at 2 months

w13 Weight loss at 3 months

se1 Self esteem at 1 month

se2 Self esteem at 2 months

se3 Self esteem at 3 months

**Details**

Helmert contrasts are assigned to group, comparing Control vs. (Diet DietEx) and Diet vs. DietEx.

**Source**

Originally taken from <http://www.csun.edu/~ata20315/psy524/main.htm>, but modified slightly. Courtesy of Michael Friendly.

---

which.names	<i>Position of Row Names</i>
-------------	------------------------------

---

**Description**

These functions return the indices of row names in a data frame or a vector of names. whichNames is just an alias for which.names.

**Usage**

```
which.names(names, object)
whichNames(...)
```

**Arguments**

names a name or character vector of names.  
 object a data frame or character vector of (row) names.  
 ... arguments to be passed to which.names.

**Value**

Returns the index or indices of names within object.

**Author(s)**

John Fox <jfox@mcmaster.ca>

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

**Examples**

```
which.names(c('minister', 'conductor'), Duncan)
## [1] 6 16
```

---

Women1f

*Canadian Women's Labour-Force Participation*

---

**Description**

The Women1f data frame has 263 rows and 4 columns. The data are from a 1977 survey of the Canadian population.

**Usage**

```
Women1f
```

**Format**

This data frame contains the following columns:

**partic** Labour-Force Participation. A factor with levels (note: out of order): fulltime, Working full-time; not.work, Not working outside the home; parttime, Working part-time.

**hincome** Husband's income, \$1000s.

**children** Presence of children in the household. A factor with levels: absent, present.

**region** A factor with levels: AtLantic, Atlantic Canada; BC, British Columbia; Ontario; Prairie, Prairie provinces; Quebec.

**Source**

*Social Change in Canada Project*. York Institute for Social Research.

**References**

Fox, J. (2008) *Applied Regression Analysis and Generalized Linear Models*, Second Edition. Sage.

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

---

Wong

*Post-Coma Recovery of IQ*

---

**Description**

The Wong data frame has 331 row and 7 columns. The observations are longitudinal data on recovery of IQ after comas of varying duration for 200 subjects.

**Usage**

Wong

**Format**

This data frame contains the following columns:

id patient ID number.

days number of days post coma at which IQs were measured.

duration duration of the coma in days.

sex a factor with levels Female and Male.

age in years at the time of injury.

piq performance (i.e., mathematical) IQ.

viq verbal IQ.

**Details**

The data are from Wong, Monette, and Weiner (2001) and are for 200 patients who sustained traumatic brain injuries resulting in comas of varying duration. After awakening from their comas, patients were periodically administered a standard IQ test, but the average number of measurements per patient is small ( $331/200 = 1.7$ ).

**Source**

Wong, P. P., Monette, G., and Weiner, N. I. (2001) Mathematical models of cognitive recovery. *Brain Injury*, **15**, 519–530.

**References**

Fox, J. (2016) *Applied Regression Analysis and Generalized Linear Models*, Third Edition. Sage.

**Examples**

```
summary(Wong)
```

---

Wool

*Wool data*

---

**Description**

This is a three-factor experiment with each factor at three levels, for a total of 27 runs. Samples of worsted yarn were with different levels of the three factors were given a cyclic load until the sample failed. The goal is to understand how cycles to failure depends on the factors.

**Usage**

```
Wool
```

**Format**

This data frame contains the following columns:

**len** length of specimen (250, 300, 350 mm)

**amp** amplitude of loading cycle (8, 9, 10 min)

**load** load (40, 45, 50g)

**cycles** number of cycles until failure

**Source**

Box, G. E. P. and Cox, D. R. (1964). An analysis of transformations (with discussion). *J. Royal Statist. Soc.*, B26, 211-46.

**References**

Fox, J. and Weisberg, S. (2011) *An R Companion to Applied Regression*, Second Edition, Sage.

Weisberg, S. (2014) *Applied Linear Regression*, Fourth Edition, Wiley, Section 6.3.

# Index

- \*Topic **algebra**
  - wcrossprod, 159
- \*Topic **aplot**
  - Ellipses, 57
  - panel.car, 107
  - regLine, 120
  - TransformationAxes, 154
- \*Topic **array**
  - wcrossprod, 159
- \*Topic **datasets**
  - Adler, 5
  - AMSSurvey, 6
  - Angell, 7
  - Anscombe, 14
  - Baumann, 17
  - Bfox, 20
  - Blackmore, 21
  - Burt, 31
  - CanPop, 32
  - Chile, 37
  - Chiot, 38
  - Cowles, 42
  - Davis, 45
  - DavisThin, 46
  - Depredations, 53
  - Duncan, 55
  - Ericksen, 61
  - Florida, 62
  - Freedman, 63
  - Friendly, 63
  - Ginzberg, 64
  - Greene, 65
  - Guyer, 66
  - Hartnagel, 66
  - Highway1, 69
  - KosteckiDillon, 79
  - Leinhardt, 80
  - LoBD, 90
  - Mandel, 93
  - Migration, 96
  - Moore, 101
  - Mroz, 102
  - OBrienKaiser, 104
  - Ornstein, 105
  - Pottery, 110
  - Prestige, 114
  - Quartet, 118
  - Robey, 125
  - Sahlins, 125
  - Salaries, 126
  - SLID, 142
  - Soils, 143
  - States, 148
  - Transact, 153
  - UN, 156
  - USPop, 157
  - Vocab, 159
  - WeightLoss, 160
  - Womenlf, 162
  - Wong, 163
  - Wool, 164
- \*Topic **distribution**
  - qqPlot, 115
- \*Topic **hplot**
  - avPlots, 15
  - Boxplot, 28
  - ceresPlots, 35
  - crPlots, 43
  - densityPlot, 50
  - dfbetaPlots, 54
  - Ellipses, 57
  - infIndexPlot, 72
  - invResPlot, 75
  - invTranPlot, 76
  - leveragePlots, 82
  - mcPlots, 94
  - mmps, 97
  - plot.powerTransform, 108

- residualPlots, 121
- scatter3d, 127
- scatterplot, 131
- scatterplotMatrix, 134
- ScatterplotSmoother, 137
- spreadLevelPlot, 145
- subsets, 149
- symbol, 150
- \*Topic **htest**
  - Anova, 7
  - leveneTest, 81
  - linearHypothesis, 84
  - ncvTest, 103
  - outlierTest, 106
- \*Topic **interface**
  - carWeb, 34
- \*Topic **manip**
  - boxCoxVariable, 27
  - logit, 92
  - recode, 118
- \*Topic **misc**
  - car-internal.Rd, 34
- \*Topic **models**
  - Anova, 7
  - Contrasts, 40
  - deltaMethod, 47
  - linearHypothesis, 84
- \*Topic **package**
  - car-package, 4
- \*Topic **print**
  - compareCoefs, 39
- \*Topic **regression**
  - Anova, 7
  - avPlots, 15
  - bcPower, 18
  - Boot, 22
  - boxCox, 24
  - boxCoxVariable, 27
  - boxTidwell, 29
  - ceresPlots, 35
  - Contrasts, 40
  - crPlots, 43
  - deltaMethod, 47
  - dfbetaPlots, 54
  - durbinWatsonTest, 56
  - hccm, 67
  - hist.boot, 70
  - infIndexPlot, 72
  - influencePlot, 74
  - invResPlot, 75
  - invTranPlot, 76
  - leveragePlots, 82
  - linearHypothesis, 84
  - mcPlots, 94
  - mmps, 97
  - ncvTest, 103
  - outlierTest, 106
  - plot.powerTransform, 108
  - powerTransform, 110
  - qqPlot, 115
  - residualPlots, 121
  - sigmaHat, 141
  - spreadLevelPlot, 145
  - subsets, 149
  - testTransform, 151
  - vif, 157
- \*Topic **ts**
  - durbinWatsonTest, 56
- \*Topic **univar**
  - qqPlot, 115
- \*Topic **utilities**
  - showLabels, 139
  - some, 144
  - which.names, 161
- .carEnv (car-internal.Rd), 34
- abbreviate, 140
- abline, 121
- adaptiveKernel (densityPlot), 50
- Adler, 5
- AMSSurvey, 6
- Angell, 7
- Anova, 7, 88, 89
- anova, 12, 89
- anova.coxph, 12
- anova.glm, 12
- anova.lm, 12
- anova.mlm, 12
- Anscombe, 14
- as.data.frame.univaov (Anova), 7
- av.plot (car-deprecated), 32
- av.plots (car-deprecated), 32
- avp (avPlots), 15
- avPlot, 33
- avPlot (avPlots), 15
- avPlots, 15, 33, 37, 45, 84, 96, 141

- basicPower, [155](#)
- basicPower (bcPower), [18](#)
- basicPowerAxis (TransformationAxes), [154](#)
- Baumann, [17](#)
- bc (car-deprecated), [32](#)
- bcnPower, [25](#), [26](#), [91](#), [113](#), [152](#), [153](#)
- bcnPower (bcPower), [18](#)
- bcPower, [18](#), [26](#), [27](#), [33](#), [111](#), [113](#), [137](#), [151](#), [152](#), [155](#)
- bcPowerAxis (TransformationAxes), [154](#)
- Bfox, [20](#)
- Blackmore, [21](#)
- Boot, [22](#), [72](#)
- boot, [23](#), [70](#), [72](#)
- boot.array, [24](#)
- boot.ci, [24](#)
- box.cox (car-deprecated), [32](#)
- box.tidwell (car-deprecated), [32](#)
- boxCox, [24](#), [113](#)
- boxcox, [26](#), [27](#), [151](#)
- boxCox2d (boxCox), [24](#)
- boxCoxVariable, [27](#), [33](#)
- Boxplot, [28](#), [131](#), [133](#)
- boxplot, [28](#), [29](#), [133](#), [151](#)
- boxTidwell, [29](#), [33](#)
- Burt, [31](#)
- bw.nrd0, [51](#)
- bw.SJ, [51](#), [52](#)
- CanPop, [32](#)
- car (car-package), [4](#)
- car-deprecated, [32](#)
- car-internal.Rd, [34](#)
- car-package, [4](#)
- carWeb, [34](#)
- ceres.plot (car-deprecated), [32](#)
- ceres.plots (car-deprecated), [32](#)
- ceresPlot, [33](#)
- ceresPlot (ceresPlots), [35](#)
- ceresPlots, [17](#), [33](#), [35](#), [45](#), [96](#)
- Chile, [37](#)
- Chirot, [38](#)
- coef, [48](#), [89](#), [158](#)
- compareCoefs, [39](#)
- confidence.ellipse (car-deprecated), [32](#)
- confidenceEllipse, [33](#)
- confidenceEllipse (Ellipses), [57](#)
- confint.boot, [23](#), [24](#)
- confint.boot (hist.boot), [70](#)
- contour, [26](#)
- contr.Helmert (Contrasts), [40](#)
- contr.helmert, [41](#)
- contr.poly, [41](#)
- contr.Sum (Contrasts), [40](#)
- contr.sum, [41](#)
- contr.Treatment (Contrasts), [40](#)
- contr.treatment, [41](#)
- Contrasts, [40](#)
- cookd (car-deprecated), [32](#)
- cooks.distance, [33](#), [74](#), [75](#)
- coplot, [108](#)
- cov.trob, [59](#), [60](#), [133](#), [137](#)
- cov.wt, [59](#), [60](#)
- Cowles, [42](#)
- cr.plot (car-deprecated), [32](#)
- cr.plots (car-deprecated), [32](#)
- crossprod, [159](#), [160](#)
- crp (crPlots), [43](#)
- crPlot, [33](#)
- crPlot (crPlots), [43](#)
- crPlots, [17](#), [33](#), [37](#), [43](#), [96](#), [141](#)
- cut, [119](#)
- D, [49](#)
- data.ellipse (car-deprecated), [32](#)
- dataEllipse, [33](#), [95](#), [96](#), [133](#), [137](#)
- dataEllipse (Ellipses), [57](#)
- Davis, [45](#)
- DavisThin, [46](#)
- deltaMethod, [47](#)
- density, [50–52](#), [72](#)
- densityPlot, [50](#)
- Depredations, [53](#)
- dfbeta, [55](#)
- dfbetaPlots, [54](#)
- dfbetas, [55](#)
- dfbetasPlots (dfbetaPlots), [54](#)
- Duncan, [55](#)
- durbin.watson (car-deprecated), [32](#)
- durbinWatsonTest, [33](#), [56](#)
- dwt (durbinWatsonTest), [56](#)
- ellipse (Ellipses), [57](#)
- Ellipses, [57](#)
- empinf, [24](#)
- Ericksen, [61](#)
- factor, [119](#)

- Florida, 62  
 Freedman, 63  
 Friendly, 63  
  
 gam, 100, 123, 128, 130, 135, 137, 138  
 gamLine, 132, 135, 146  
 gamLine (ScatterplotSmoothers), 137  
 Ginzberg, 64  
 Greene, 65  
 Guyer, 66  
  
 Hartnagel, 66  
 hatvalues, 74, 75  
 hccm, 10, 67, 87, 89, 104, 147  
 head, 145  
 Highway1, 69  
 hist, 72  
 hist.boot, 23, 24, 70  
  
 identify, 124, 140  
 Identify3d (scatter3d), 127  
 infIndexPlot, 72  
 influence.plot (influencePlot), 74  
 influenceIndexPlot (infIndexPlot), 72  
 influencePlot, 74  
 inverseResponsePlot, 78  
 inverseResponsePlot (invResPlot), 75  
 invResPlot, 75  
 invTranEstimate (invTranPlot), 76  
 invTranPlot, 75, 76, 76  
  
 jitter, 132, 133  
  
 KosteckiDillon, 79  
  
 legend, 52, 133, 136, 149  
 Leinhardt, 80  
 levene.test (car-deprecated), 32  
 leveneTest, 33, 81  
 leverage.plot (car-deprecated), 32  
 leverage.plots (car-deprecated), 32  
 leveragePlot, 34  
 leveragePlot (leveragePlots), 82  
 leveragePlots, 34, 82, 141  
 lht (linearHypothesis), 84  
 linear.hypothesis (car-deprecated), 32  
 linearHypothesis, 12, 34, 60, 84  
 lines, 121  
 lm, 29, 111, 124, 151  
 LoBD, 90  
  
 loess, 137, 138  
 loessLine, 132, 146  
 loessLine (ScatterplotSmoothers), 137  
 logit, 92, 155  
  
 makeHypothesis (linearHypothesis), 84  
 Mandel, 93  
 Manova (Anova), 7  
 marginalModelPlot (mmps), 97  
 marginalModelPlots (mmps), 97  
 matchCoefs (linearHypothesis), 84  
 mcPlot (mcPlots), 94  
 mcPlots, 94  
 Migration, 96  
 mmp (mmps), 97  
 mmps, 97  
 model.matrix, 158  
 Moore, 101  
 Mroz, 102  
  
 na.omit, 51  
 ncv.test (car-deprecated), 32  
 ncvTest, 34, 103, 147  
  
 OBrienKaiser, 104  
 optim, 113  
 optimize, 78  
 Ornstein, 105  
 outlier.test (car-deprecated), 32  
 outlierTest, 34, 74, 106  
  
 p.adjust, 11  
 pairs, 137  
 palette, 16, 36, 44, 52, 55, 59, 77, 116, 120, 146  
 panel.car, 107  
 par, 16, 36, 44, 59, 60, 83, 95, 116, 120, 132, 133, 136, 146, 147, 149, 155  
 plot, 55, 100, 132  
 plot.boot, 24  
 plot.density, 52  
 plot.powerTransform, 108  
 points, 55  
 Pottery, 110  
 powerTransform, 20, 25–27, 33, 76, 109, 110, 136, 137, 152, 153  
 Prestige, 114  
 print.Anova.mlm (Anova), 7  
 print.boxTidwell (boxTidwell), 29



- print.durbinWatsonTest
  - (durbinWatsonTest), 56
- print.linearHypothesis.mlm
  - (linearHypothesis), 84
- print.outlierTest(outlierTest), 106
- print.spreadLevelPlot
  - (spreadLevelPlot), 145
- print.summary.Anova.mlm (Anova), 7
- print.univaov (Anova), 7
- printCoefmat, 39
- printHypothesis (linearHypothesis), 84
- probabilityAxis, 92
- probabilityAxis (TransformationAxes), 154
  
- qq.plot (car-deprecated), 32
- qqline, 117
- qqnorm, 117
- qqp (qqPlot), 115
- qqPlot, 34, 115
- qqplot, 117
- qss, 138
- quantregLine (ScatterplotSmoothers), 137
- Quartet, 118
  
- Recode (recode), 118
- recode, 118
- regLine, 108, 120
- regsubsets, 149, 150
- regular expression, 87
- residCurvTest (residualPlots), 121
- residualPlot (residualPlots), 121
- residualPlots, 17, 96, 121, 141
- residuals.lm, 123
- Robey, 125
- rqss, 137, 138
- rstudent, 74, 75
  
- Sahlins, 125
- Salaries, 126
- scatter3d, 127
- scatterplot, 131, 137, 138
- scatterplot.matrix (car-deprecated), 32
- scatterplotMatrix, 34, 133, 134, 137, 138
- ScatterplotSmoothers, 36, 44, 99, 100, 123, 132, 133, 135, 137, 137, 146
- showLabels, 16, 36, 37, 44, 55, 60, 73, 75, 76, 78, 83, 95, 99, 116, 117, 123, 124, 129, 132, 133, 135, 137, 139, 147
  
- sigmaHat, 141
- SLID, 142
- slp (spreadLevelPlot), 145
- Soils, 143
- some, 144
- sp (scatterplot), 131
- spm (scatterplotMatrix), 134
- spread.level.plot (car-deprecated), 32
- spreadLevelPlot, 34, 104, 145
- States, 148
- subsets, 149
- substr, 140
- summary, 23
- summary.Anova.mlm (Anova), 7
- summary.boot, 24
- summary.boot (hist.boot), 70
- symbol, 150
  
- tail, 145
- testTransform, 20, 112, 113, 151
- Transact, 153
- transform, 113
- TransformationAxes, 154
- tukeyNonaddTest (residualPlots), 121
  
- UN, 156
- USPop, 157
  
- vcov, 48, 89, 158
- vcov.boot, 23
- vcov.boot (hist.boot), 70
- vcovHAC, 89
- vcovHC, 89
- vif, 157
- Vocab, 159
  
- waldtest, 89
- wcrossprod, 159
- WeightLoss, 160
- which.names, 161
- whichNames (which.names), 161
- Womenlf, 162
- Wong, 163
- Wool, 164
  
- yjPower, 26, 112, 137, 151, 152, 155
- yjPower (bcPower), 18
- yjPowerAxis (TransformationAxes), 154