The car Package

February 6, 2006

Version 1.1-0

Date 2006/2/6

Title Companion to Applied Regression

Author John Fox <jfox@mcmaster.ca>. I am grateful to Douglas Bates, David Firth, Michael Friendly, Gregor Gorjanc, Spencer Graves, Georges Monette, Henric Nilsson, Brian Ripley, Sanford Weisberg, and Achim Zeileis for various suggestions and contributions.

Maintainer John Fox <jfox@mcmaster.ca>

Depends R (>= 2.1.1)

Suggests MASS, nnet, leaps

LazyLoad yes

LazyData yes

Description This package accompanies J. Fox, An R and S-PLUS Companion to Applied Regression, Sage, 2002. The package contains mostly functions for applied regression, linear models, and generalized linear models, with an emphasis on regression diagnostics, particularly graphical diagnostic methods. There are also some utility functions. With some exceptions, I have tried not to duplicate capabilities in the basic distribution of R, nor in widely used packages. Where relevant, the functions in car are consistent with na.action = na.omit or na.exclude.

License GPL version 2 or newer


R topics documented:

Adler .................................................. 3
Angell ................................................. 4
Anova .................................................. 4
Anscome .............................................. 7
Ask ..................................................... 7
av.plots .............................................. 8
Baumann ............................................ 10
Bfox .................................................. 11
Blackmoor .......................................... 11
box.cox.powers ................................... 12
<table>
<thead>
<tr>
<th>R topics documented:</th>
</tr>
</thead>
<tbody>
<tr>
<td>box.cox</td>
</tr>
<tr>
<td>box.cox.var</td>
</tr>
<tr>
<td>box.tidwell</td>
</tr>
<tr>
<td>Burt</td>
</tr>
<tr>
<td>Can.pop</td>
</tr>
<tr>
<td>car-internal</td>
</tr>
<tr>
<td>car-package</td>
</tr>
<tr>
<td>ceres.plots</td>
</tr>
<tr>
<td>Chile</td>
</tr>
<tr>
<td>Chirot</td>
</tr>
<tr>
<td>Contrasts</td>
</tr>
<tr>
<td>Cook's Distances</td>
</tr>
<tr>
<td>Cowles</td>
</tr>
<tr>
<td>cr.plots</td>
</tr>
<tr>
<td>Davis</td>
</tr>
<tr>
<td>DavisThin</td>
</tr>
<tr>
<td>Duncan</td>
</tr>
<tr>
<td>durbin.watson</td>
</tr>
<tr>
<td>Ellipses</td>
</tr>
<tr>
<td>Ericksen</td>
</tr>
<tr>
<td>Florida</td>
</tr>
<tr>
<td>Freedman</td>
</tr>
<tr>
<td>Friendly</td>
</tr>
<tr>
<td>Ginzberg</td>
</tr>
<tr>
<td>Greene</td>
</tr>
<tr>
<td>Guyer</td>
</tr>
<tr>
<td>Hartnagel</td>
</tr>
<tr>
<td>hccm</td>
</tr>
<tr>
<td>influence.plot</td>
</tr>
<tr>
<td>Leinhardt</td>
</tr>
<tr>
<td>levene.test</td>
</tr>
<tr>
<td>leverage.plots</td>
</tr>
<tr>
<td>linear.hypothesis</td>
</tr>
<tr>
<td>logit</td>
</tr>
<tr>
<td>Mandel</td>
</tr>
<tr>
<td>Migration</td>
</tr>
<tr>
<td>Moore</td>
</tr>
<tr>
<td>Mroz</td>
</tr>
<tr>
<td>n.bins</td>
</tr>
<tr>
<td>ncv.test</td>
</tr>
<tr>
<td>Ornstein</td>
</tr>
<tr>
<td>outlier.test</td>
</tr>
<tr>
<td>panel.car</td>
</tr>
<tr>
<td>Prestige</td>
</tr>
<tr>
<td>qq.plot</td>
</tr>
<tr>
<td>Quartet</td>
</tr>
<tr>
<td>recode</td>
</tr>
<tr>
<td>reg.line</td>
</tr>
<tr>
<td>Robey</td>
</tr>
<tr>
<td>Sahlins</td>
</tr>
<tr>
<td>scatterplot.matrix</td>
</tr>
<tr>
<td>scatterplot</td>
</tr>
</tbody>
</table>
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


References

**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**


**References**


---

**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), and `polr` (in the `MASS` package). For linear models, F-tests are calculated; for generalized linear models, likelihood-ratio chisquare, Wald chisquare, or F-tests are calculated; for multinomial logit and proportional-odds logit models, likelihood-ratio tests are calculated.
Usage

Anova(mod, ...)

## S3 method for class 'lm':
Anova(mod, error, type=c("II", "III"), ...)

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

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

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

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

Arguments

mod lm, aov, glm, multinom, or polr 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" or "III".
test.statistic for a generalized linear model, whether to calculate "LR" (likelihood-ratio),
"Wald", or "F" tests.
error.estimate 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), which, e.g., is fixed to 1 for binomial and Poisson
models; or base the dispersion estimate on the residual deviance (deviance).
... arguments to be passed to linear.hypothesis; only use white.adjust
for a linear model.

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 for generalized linear models are actually differences of Wald statistics.
For tests for linear models, and Wald tests for generalized linear models, `Anova` finds the test statistics without refitting the model.

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

**Value**

An object of class `anova`, usually printed.

**Warning**

Be careful of type-III tests.

**Author(s)**

John Fox (jfox@mcmaster.ca)

**References**


**See Also**

`linear.hypothesis`, `anova`

**Examples**

```r
mod <- lm(conformity ~ fcategory*partner.status, data = Moore,
            contrasts = list(fcategory = contr.sum, partner.status = contr.sum))
Anova(mod)
## Anova Table (Type II tests)
##            Sum Sq Df  F value   Pr(>F)
## fcategory 11.61  2 0.2770 0.759564
## partner.status 212.21  1 10.1207 0.002874
## fcategory:partner.status 175.49  2 4.1846 0.022572
## Residuals     817.76 39

Anova(mod, type = "III")
## Anova Table (Type III tests)
##            Sum Sq Df   F value  Pr(>F)
## (Intercept) 5752.8  1 274.3592 < 2.2e-16
## fcategory   36.0  2  0.8589  0.431492
## partner.status 239.6  1 11.4250  0.001657
## fcategory:partner.status 175.5  2  4.1846  0.022572
## Residuals     817.8 39
```
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


References


Ask

Change Argument to a Function Interactively

Description

Ask allows you to change the argument to a function interactively. It is meant to be used, in lieu of a graphical control such as a slidebar, to adjust plotting parameters, which are most naturally passed as the argument to an anonymous function that sets up the plot.

Usage

`Ask(arg, fun, ...)`

Arguments

- `arg`: argument to `fun` to change. By specifying a vector of values, you can change several parameters via an argument to an anonymous function.
- `fun`: function to call; often an anonymous function that sets up a call to plotting functions.
- `...`: other arguments to `fun`; not necessary if `fun` is an anonymous function.
Details

Ask repeatedly prompts in the R Console for the value of `arg`. To exit, enter a blank line.

Value

Ask returns invisibly the value of the last call to `fun`; usually this will be `NULL`, and in any event is probably not of interest. If it is, use `print(Ask(arg, fun, ...))`.

Author(s)

John Fox (jfox@mcmaster.ca)

Examples

```r
## Not run:
attach(UN)

# enter the power-transformation parameter
# start with 1
Ask(p, function(p) qq.plot(box.cox(gdp, p),
                   ylab=paste("transformed gdp, power =",p)))

# enter an expression that evaluates to a 2-vector
# of powers; e.g., start with c(1,1); then interactively
# identify points in each plot
Ask(p, function(p) scatterplot(box.cox(gdp,p[1]),
                                      box.cox(infant.mortality, p[2]),
                                       xlab=paste("transformed GDP/capita, power =",p[1]),
                                       ylab=paste("transformed infant mortality, power =",p[2]),
                                       labels=rownames(UN)))
## End(Not run)
```

---

### av.plots

**Added-Variable Plots**

**Description**

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

**Usage**

```r
av.plots(model, variable, ask=missing(variable), one.page=!ask, ...)

avp(...)  

av.plot(model, ...)
```

```r
## S3 method for class 'lm':
av.plot(model, variable,
       labels=names(residuals(model)[!is.na(residuals(model))]),
       identify.points=TRUE, las=par("las"), col=palette()[2], pch=1, lwd=2,
```

---

av.plots
main="Added-Variable Plot", ...)

## S3 method for class 'glm':
av.plot(model, variable,
  labels=names(residuals(model)[!is.na(residuals(model))]),
  identify.points=TRUE, las=par("las"), col=palette()[2], pch=1, lwd=2,
  main="Added-Variable Plot", type=c("Wang", "Weisberg"), ...)

Arguments

model  model object produced by lm or glm.
variable variable (if it exists in the search path) or name of variable. This argument usually is omitted for avp or av.plots.
ask    if TRUE, a menu is provided in the R Console for the user to select the term(s) to plot.
one.page if TRUE (and ask=FALSE), put all plots on one graph.
labels observation names.
identify.points if TRUE, then identify points interactively.
las    if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see par).
col    color for points and lines; the default is the second entry in the current color palette (see palette and par).
pch    plotting character for points; default is 1 (a circle, see par).
lwd    line width; default is 2 (see par).
main   title for plot.
type   if "Wang" use the method of Wang (1985); if "Weisberg" use the method in the Arc software associated with Cook and Weisberg (1999).
...    arguments to be passed down to av.plot.lm or av.plot.glm.

Details

The function intended for direct use is av.plots (for which avp is an abbreviation). By default, these functions are used interactively through a text menu.

The model can contain factors and interactions. An added-variable plot can be drawn for each column of the model matrix, including the constant.

Value

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

Author(s)

John Fox (jfox@mcmaster.ca)

References

Methods of Teaching Reading Comprehension

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

**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.

**Source**


**References**


---

**Blackmoor**

*Exercise Histories of Eating-Disordered and Control Subjects*

**Description**

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

**Usage**

*Blackmoor*
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 Blackmoor and Caroline Davis, York University.

---

**box.cox.powers**  
*Multivariate Unconditional Box-Cox Transformations*

Description

Estimates multivariate unconditional power transformations to multinormality by the method of maximum likelihood. The univariate case is obtained when only one variable is specified.

Usage

```r
box.cox.powers(X, start=NULL, hypotheses=NULL, ...
## S3 method for class 'box.cox.powers':
print(x, digits=4, ...
## S3 method for class 'box.cox.powers':
summary(object, digits=4, ...
```

Arguments

- **X** 
a numeric matrix of variables (or a vector for one variable) to be transformed.
- **start** 
start values for the power transformation parameters; if NULL (the default), univariate Box-Cox transformations will be computed and used as the start values.
- **hypotheses** 
if non-NULL, a list of hypotheses to be tested; each hypothesis should be a vector of values giving the power for each column of X. Note that the hypotheses that all powers are 1 and that all powers are 0 (log) are always tested.
- **...** 
optional arguments to be passed to the `optim` function.
- **digits** 
number of places to round result.
- **x, object** 
`box.cox.powers` object.

Details

Note that this is *unconditional* Box-Cox. That is, there is no regression model, and there are no predictors. The object is to make the distribution of the variable(s) as (multi)normal as possible. For Box-Cox regression, see the `boxcox` function in the MASS package.

The function estimates the Box-Cox powers, $x_j' = (x_j^{\lambda_j} - 1)/\lambda_j$ for $\lambda_j \neq 0$ and $x_j' = \log x_j$ for $\lambda_j = 0$. Subsequently using ordinary power transformations (i.e., $x^p$ for $p \neq 0$) achieves the same result.
Value

returns an object of class `box.cox.powers`, which may be printed or summarized. The `print` and `summary` methods are now identical; I’ve retained the latter for backwards compatibility.

Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

`boxcox`, `box.cox`, `box.cox.var`, `box.cox.axis`

Examples

```r
attach(Prestige)
box.cox.powers(cbind(income, education))
## Box-Cox Transformations to Multinormality
##
## Est.Power Std.Err. Wald(Power=0) Wald(Power=1)
## income 0.2617 0.1014 2.580 -7.280
## education 0.4242 0.4033 1.052 -1.428
##
## L.R. test, all powers = 0: 7.694 df = 2 p = 0.0213
## L.R. test, all powers = 1: 48.8727 df = 2 p = 0
plot(income, education)
plot(box.cox(income, .26), box.cox(education, .42))

box.cox.powers(income)
## Box-Cox Transformation to Normality
##
## Est.Power Std.Err. Wald(Power=0) Wald(Power=1)
## 0.1793 0.1108 1.618 -7.406
##
## L.R. test, power = 0: 2.7103 df = 1 p = 0.0997
## L.R. test, power = 1: 47.261 df = 1 p = 0
qq.plot(income)
qq.plot(income^.18)
```

---

**box.cox**

*Box-Cox Family of Transformations*

---

Description

Compute the Box-Cox power transformation of a variable.
Usage

box.cox(x, p, start=0)

bc(x, p, start=0)

Arguments

x numeric vector to transform.
p power (0 = log); if p is a vector then a matrix of transformed values with columns labelled by powers will be returned.
start constant to be added to each value of x prior to transformation.

Details

Computes $x' = (x^p - 1)/p$ for $p \neq 0$ and $x' = \log x$ for $p = 0$.
The values of x must all be positive; if not, a start should be added to each value to make all the values positive. The function will automatically compute the start and print a warning, if necessary.

bc is just an abbreviation for box.cox.

Value

a vector or matrix of transformed values.

Warning

These functions do not compute the maximum-likelihood estimate for a Box-Cox normalizing transformation. See box.cox.powers for estimating unconditional univariate and multivariate Box-Cox transformations, and boxcox in the MASS package for estimating the Box-Cox transformation of the response in a linear model.

Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

boxcox, box.cox.var, box.cox.powers, box.cox.axis
box.cox.var

Examples

box.cox(1:10, 2)
## [1] 0.0 1.5 4.0 7.5 12.0 17.5 24.0 31.5 40.0 49.5

box.cox(1:5, c(0, 2))
## [1] 0 2
## [2] 0.0000000 0.0
## [3] 0.6931472 1.5
## [4] 1.0986123 4.0

box.cox(-5:5, 2)
## [1] 0.0 1.5 4.0 7.5 12.0 17.5 24.0 31.5 40.0 49.5 60.0
## Warning message:
## start = 6 added to data prior to transformation in: box.cox(-5:5, 2)

options(digits=4)
box.cox(-5:5, 0, start=6)
## [1] 0.0000 0.6931 1.0986 1.3863 1.6094 1.7918 1.9459 2.0794 2.1972
## [10] 2.3026 2.3979

box.cox.var

Description

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

Usage

box.cox.var(y)

Arguments

y
response variable.

Details

The constructed variable is defined as \( y [\log(y/\bar{y}) - 1] \), where \( \bar{y} \) 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


See Also

boxcox, box.cox, box.cox.powers, box.cox.axis, av.plots

Examples

```r
mod <- lm(interlocks + 1 ~ assets, data=Ornstein)
mod.aux <- update(mod, . ~ . + box.cox.var(interlocks + 1))
summary(mod.aux)
## Call:
## lm(formula = interlocks + 1 ~ assets + box.cox.var(interlocks + 
## 1), data = Ornstein)
## Residuals:
##    Min     1Q Median     3Q    Max
## -23.189 -6.701  0.541  6.773 12.051
## Coefficients:
##            Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.461e+01  5.426e-01  26.920  <2e-16
## assets     -7.142e-05  5.119e-05  -1.395  0.164
## box.cox.var(interlocks + 1)  7.427e-01  4.136e-02  17.956  <2e-16
## Residual standard error: 7.247 on 245 degrees of freedom
## Multiple R-Squared: 0.7986, Adjusted R-squared: 0.797
## F-statistic: 485.7 on 2 and 245 degrees of freedom, p-value: 0

av.plots(mod.aux, "box.cox.var(interlocks + 1)")
```

---

**box.tidwell**  
**Box-Tidwell Transformations**

Description

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

Usage

```r
box.tidwell(y, ...)
```

## S3 method for class 'formula':
```r
box.tidwell(formula, other.x=NULL, data=NULL, subset,
na.action=options()$na.action, verbose=FALSE, tol=0.001,
max.iter=25, ...)
```
## Default S3 method:
box.tidwell(y, x1, x2=NULL, max.iter=25, tol=0.001,
verbose=FALSE, ...)

## S3 method for class 'box.tidwell':
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 *not* 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 `box.tidwell` 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.
- **tol**: if 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 *not* candidates for transformation.
- **...**: not for the user.
- **x**: `box.tidwell` 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 `box.tidwell`, which is normally just printed.

### Author(s)

John Fox (jfox@mcmaster.ca)

### References


Examples

```r
box.tidwell(prestige~income+education, ~ poly(women,2), data=Prestige)
```

## income education

<table>
<thead>
<tr>
<th>Initial Power</th>
<th>Score Statistic</th>
<th>p-value</th>
<th>MLE of Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.91030</td>
<td>-5.30129</td>
<td>0.00000</td>
<td>-0.03777</td>
</tr>
<tr>
<td>2.24354</td>
<td>2.40556</td>
<td>0.01615</td>
<td>2.19283</td>
</tr>
</tbody>
</table>

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.

**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**


Canadian Population Data

**Description**


**Usage**

Can.pop

**Format**

This data frame contains the following columns:

- year  census year.
- population  Population, in millions
Source


Canada (1994) *Canada Year Book*. Statistics Canada [Table 3.2].

References


car-internal  

*Internal car functions*

Description

Internal functions for package car.

Usage

df.terms(model, term, ...)
## Default S3 method:
df.terms(model, term, ...)
## S3 method for class 'multinom':
df.terms(model, term, ...)
## S3 method for class 'polr':
df.terms(model, term, ...)
has.intercept(model, ...)
## Default S3 method:
has.intercept(model, ...)
inv(x)
is.aliased(model)
mfrow(n, max.plots=0)
nice(x, direction=c("round", "down", "up"))
predictor.names(model, ...)
## Default S3 method:
predictor.names(model, ...)
relatives(term, names, factors)
responseName(model, ...)
## S3 method for class 'chisq.test':
print(x, ...)
## Default S3 method:
responseName(model, ...)
response(model, ...)
## Default S3 method:
response(model, ...)
term.names(model, ...)
## Default S3 method:
term.names(model, ...)
Arguments

model
term
...
x
n
max.plots
direction
names
factors

Details

These functions are not intended to be called by the user.

Author(s)

John Fox (jfox@mcmaster.ca)

car-package  Companion to Applied Regression

Description

This package accompanies J. Fox, An R and S-PLUS Companion to Applied Regression, Sage, 2002. The package contains mostly functions for applied regression, linear models, and generalized linear models, with an emphasis on regression diagnostics, particularly graphical diagnostic methods. There are also some utility functions. With some exceptions, I have tried not to duplicate capabilities in the basic distribution of R, nor in widely used packages. Where relevant, the functions in car are consistent with na.action = na.omit or na.exclude.

Details

Package: car
Version: 1.1-0
Date: 2006/2/6
Depends: R (>= 2.1.1)
Suggests: MASS, nnet, leaps
License: GPL version 2 or newer
URL: http://www.r-project.org, http://socserv.socsci.mcmaster.ca/jfox/

Index:

Adler  Experimenter Expectations
Angell  Moral Integration of American Cities
Anova  Anova Tables for Various Statistical Models
Anscombe  U. S. State Public-School Expenditures
Ask
Baumann
Bfox
Blackmoor
Burt
Can.pop
Chile
Chirot
Contrasts
Cowles
Davis
DavisThin
Duncan
Ericksen
Florida
Freedman
Friendly
Ginzberg
Greene
Guyer
Hartnagel
Leinhardt
Mandel
Migration
Moore
Mroz
Ornstein
Prestige
Quartet
Robey
SLID
Sahlins
States
UN
US.pop
Var
Vocab
Womenlf
av.plots
box.cox
box.cox.powers
box.cox.var
box.tidwell
ceres.plots
cookd
cr.plots
Change Argument to a Function Interactively
Methods of Teaching Reading Comprehension
Canadian Women's Labour-Force Participation
Exercise Histories of Eating-Disordered and Control Subjects
Fraudulent Data on IQs of Twins Raised Apart
Canadian Population Data
Voting Intentions in the 1988 Chilean Plebiscite
The 1907 Romanian Peasant Rebellion
Functions to Construct Contrasts
Cowles and Davis's Data on Volunteering
Self-Reports of Height and Weight
Davis's Data on Drive for Thinness
Duncan's Occupational Prestige Data
The 1980 U.S. Census Undercount
Florida County Voting
Crowding and Crime in U. S. Metropolitan Areas
Format Effects on Recall
Data on Depression
Refugee Appeals
Anonymity and Cooperation
Canadian Crime-Rates Time Series
Data on Infant-Mortality
Contrived Collinear Data
Canadian Interprovincial Migration Data
Status, Authoritarianism, and Conformity
U.S. Women's Labor-Force Participation
Interlocking Directorates Among Major Canadian Firms
Prestige of Canadian Occupations
Four Regression Datasets
Fertility and Contraception
Survey of Labour and Income Dynamics
Agricultural Production in Mazulu Village
Education and Related Statistics for the U.S. States
GDP and Infant Mortality
Population of the United States
Variance-Covariance Matrices (deprecated)
Vocabulary and Education
Canadian Women's Labour-Force Participation
Added-Variable Plots
Box-Cox Family of Transformations
Multivariate Unconditional Box-Cox Transformations
Constructed Variable for Box-Cox Transformation
Box-Tidwell Transformations
Ceres Plots
Cook's Distances for Linear and Generalized Linear Models
Component+Residual (Partial Residual) Plots
**ceres.plot()**

These functions calculate Ceres plots for linear and generalized linear model.

**Usage**

```r
ceres.plot(model, ...)  
ceres.plot(model, ...)  
```

## S3 method for class 'im':

```r  
```
### Arguments

- **model**
  - model object produced by `lm` or `glm`.

- **variable**
  - variable (if it exists in the search path) or name of variable. This argument usually is omitted for `ceres.plots`.

- **ask**
  - if `TRUE`, a menu is provided in the R Console for the user to select the variable(s) to plot, and to modify the span for the smoother used to draw a nonparametric-regression line on the plot.

- **one.page**
  - if `TRUE` (and `ask=FALSE`), put all plots on one graph.

- **span**
  - span for lowess smoother.

- **iter**
  - number of robustness iterations for nonparametric-regression smooth; defaults to 3 for a linear model and to 0 for a non-Gaussian glm.

- **line**
  - `TRUE` to plot least-squares line.

- **smooth**
  - `TRUE` to plot nonparametric-regression (lowess) line.

- **las**
  - if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see `par`).

- **col**
  - color for points and lines; the default is the second entry in the current color palette (see `palette` and `par`).

- **pch**
  - plotting character for points; default is 1 (a circle, see `par`).

- **lwd**
  - line width; default is 2 (see `par`).

- **main**
  - title for plot.

- **...**
  - pass arguments down.

### 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 `ceres.plots`. By default, this function is used interactively through a text menu.

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)
Chile

References


See Also
cr.plots, av.plots

Examples

## Not run:
ceres.plots(lm(prestige~income+education+type, data=Prestige))
## End(Not run)

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

Chirot

**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**


**References**


---

**Contrasts**

**Functions to Construct Contrasts**

**Description**

These are substitutes for similarly named functions in the base 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 base package.

**Usage**

- `contr.Treatment(n, base = 1, contrasts = TRUE)`
- `contr.Sum(n, contrasts = TRUE)`
- `contr.Helmert(n, contrasts = TRUE)`
Arguments

n   a vector of levels for a factor, or the number of levels.
base an integer specifying which level is considered the baseline level. Ignored if contrasts is FALSE.
contrasts 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)

See Also

contr.treatment, contr.sum, contr.helmert, contr.poly

Examples

```r
# 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"))
```
Cook’s Distances

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

```r
## Cook’s Distances
Cook’s Distances for Linear and Generalized Linear Models

Description

This function now simply calls `cooks.distance` in the base package.

Usage

```r
cookd(model, ...)
```

Arguments

- `model` lm or glm model object.
- `...` other arguments to be passed to `cooks.distance`.

Details

Cook’s distances for generalized linear models are approximations, as described in Williams (1987) (except that the Cook’s distances are scaled as $F$ rather than as chi-square values).

This function is retained primarily for consistency with An R and S-PLUS Companion to Applied Regression. Other deletion diagnostics formerly in the car package have been rewritten and moved to the base package; these include `influence`, `rstudent`, `hatvalues`, `dfbeta`, and `dfbetas`.

Value

`cookd` returns a vector with one entry for each observation.
Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

cooks.distance, influence

Examples

```r
plot(cookd(lm(prestige ~ income + education, data=Duncan)))
```

---

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

Component+Residual (Partial Residual) Plots

Description

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

Usage

```r
cr.plots(model, variable, ask=missing(variable), one.page=!ask, span=0.5, ...)
crp(...)  
cr.plot(model, ...)  
## S3 method for class 'lm':
cr.plot(model, variable, order=1, line=TRUE, smooth=TRUE, iter, span=0.5, las=par('las'), col=palette()[2], pch=1, lwd=2, main="Component+Residual Plot", ...)
## S3 method for class 'glm':
cr.plot(model, ...)  
```

Arguments

- **model**: model object produced by `lm` or `glm`.
- **variable**: variable (if it exists in the search path) or name of variable. This argument usually is omitted for `crp` or `cr.plots`.
- **ask**: if TRUE, a menu is provided in the R Console for the user to select the variable(s) to plot, and to modify the span for the smoother used to draw a nonparametric-regression line on the plot.
- **one.page**: if TRUE (and ask=FALSE), put all plots on one graph.
- **order**: order of polynomial regression performed for predictor to be plotted.
- **line**: TRUE to plot least-squares line.
- **smooth**: TRUE to plot nonparametric-regression (lowess) line.
- **iter**: number of robustness iterations for nonparametric-regression smooth; defaults to 3 for a linear model and to 0 for a non-Gaussian glm.
- **span**: span for lowess smoother.
- **las**: if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see `par`).
- **col**: color for points and lines; the default is the second entry in the current color palette (see `palette` and `par`).
- **pch**: plotting character for points; default is 1 (a circle, see `par`).
- **lwd**: line width; default is 2 (see `par`).
- **main**: title for plot.
- **...**: pass arguments down.
The function intended for direct use is `cr.plots` (for which `crp` is an abbreviation). By default, these functions are used interactively through a text menu.

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: producing plots.

**Author(s)**

John Fox (jfox@mcmaster.ca)

**References**


**See Also**

ceres.plots, av.plots

**Examples**

```r
cr.plots(lm(prestige~income+education, data=Prestige),
  variable="income")
## Not run:
cr.plots(glm(partic ~= "not.work" ~ hincome + children,
  data=Womenlf, family=binomial))
## End(Not run)
```

---

**Details**

The function intended for direct use is `cr.plots` (for which `crp` is an abbreviation). By default, these functions are used interactively through a text menu.

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: producing plots.

**Author(s)**

John Fox (jfox@mcmaster.ca)

**References**


**See Also**

ceres.plots, av.plots

**Examples**

```r
cr.plots(lm(prestige~income+education, data=Prestige),
  variable="income")
## Not run:
cr.plots(glm(partic ~= "not.work" ~ hincome + children,
  data=Womenlf, family=binomial))
## End(Not run)
```

---

**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.
- **repht**: Reported height in cm.

Source

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

References


---

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

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


References


durbin.watson

Durbin-Watson Test for Autocorrelated Errors

Description

Computes residual autocorrelations and generalized Durbin-Watson statistics and their bootstrapped p-values.

Usage

durbin.watson(model, ...)

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

## Default S3 method:
Ellipses

```r
durbin.watson(model, max.lag=1, ...)
```

```r
## S3 method for class 'durbin.watson':
print(x, ...)
```

**Arguments**

- `model` a linear-model object, or a vector of residuals from a linear model.
- `max.lag` maximum lag to which to compute residual autocorrelations and Durbin-Watson statistics.
- `simulate` if TRUE p-values will be estimated by bootstrapping.
- `reps` number of bootstrap replications.
- `method` 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.
- `alternative` sign of autocorrelation in alternative hypothesis; specify only if `max.lag = 1`; if `max.lag > 1`, then `alternative` is taken to be "two.sided".
- `...` arguments to be passed down to method functions.
- `x` `durbin.watson` object.

**Value**

Returns an object of type "durbin.watson".

**Author(s)**

John Fox (jfox@mcmaster.ca)

**References**


**Examples**

```r
durbin.watson(lm(fconvict ~ tfr + partic + degrees + mconvict, data=Hartnagel))
```

```r
## lag Autocorrelation D-W Statistic p-value
## 1 0.688345 0.6168636 0
## Alternative hypothesis: rho != 0
```

---

**Ellipses, Data Ellipses, and Confidence Ellipses**

**Description**

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

ellipse(center, shape, radius, center.pch=19, center.cex=1.5,
   segments=51, add=TRUE, xlab="", ylab=",
   las=par('las'), col=palette()[2], lwd=2, lty=1, ...)

data.ellipse(x, y, levels=c(0.5, 0.9), center.pch=19, center.cex=1.5,
   plot.points=TRUE, add=!plot.points, segments=51, robust=FALSE,
   xlab=deparse(substitute(x)),
   ylab=deparse(substitute(y)),
   las=par('las'), col=palette()[2], pch=1, lwd=2, lty=1, ...)

confidence.ellipse(model, ...)

## S3 method for class 'lm':
confidence.ellipse(model, which.coef, levels=0.95, Scheffe=FALSE,
   center.pch=19, center.cex=1.5, segments=51, xlab, ylab,
   las=par('las'), col=palette()[2], lwd=2, lty=1, ...)

## S3 method for class 'glm':
confidence.ellipse(model, which.coef, levels=0.95, Scheffe=FALSE,
   center.pch=19, center.cex=1.5, segments=51, xlab, ylab,
   las=par('las'), col=palette()[2], lwd=2, lty=1, ...)

Arguments

center 2-element vector with coordinates of center of ellipse.
shape 2 x 2 shape (or covariance) matrix.
radius radius of circle generating the ellipse.
center.pch character for plotting ellipse center.
center.cex relative size of character for plotting ellipse center.
segments number of line-segments used to draw ellipse.
add if TRUE add ellipse to current plot.
xlab label for horizontal axis.
ylab label for vertical axis.
x a numeric vector, or (if y is missing) a 2-column numeric matrix.
y a numeric vector, of the same length as x.
plot.points if FALSE data ellipses are added to the current scatterplot, but points are not plotted.
levels draw elliptical contours at these (normal) probability or confidence levels.
robust if TRUE use the cov.trob function in the MASS package to calculate the center and covariance matrix for the data ellipse.
model a model object produced by lm or glm.
which.coef 2-element vector giving indices of coefficients to plot; if missing, the first two coefficients (disregarding the regression constant) will be selected.
Scheffe if TRUE scale the ellipse so that its projections onto the axes give Scheffe confidence intervals for the coefficients.
las

if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see par).

col

color for points and lines; the default is the second entry in the current color palette (see palette and par).

pch

plotting character for points; default is 1 (a circle, see par).

lwd

line width; default is 2 (see par).

lty

line type; default is 1, a solid line (see par).

... other plotting parameters to be passed to plot and line.

Details

The ellipse is computed by suitably transforming a unit circle.

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

Value

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

Author(s)

Georges Monette (Georges.Monette@mathstat.YorkU.CA) and John Fox (jfox@mcmaster.ca)

References


See Also

cov.trob.

Examples

data.ellipse(Prestige$income, Prestige$education, levels=0.1*1:9, lty=2)

confidence.ellipse(lm(prestige~income+education, data=Prestige), Scheffe=TRUE)

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
Florida

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


References


---

**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.
### Freedman

#### Source


---

#### 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


#### References


---

### Friendly

#### 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


Personal communication from M. Friendly, Department of Psychology, York University.

References


---

**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

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


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

Source


References


---

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.
- **ftheft**: Female theft conviction rate per 100,000.
- **mconvict**: Male indictable-offense conviction rate per 100,000.
- **mtheft**: 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.
**hccm**

**Source**

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

**References**


---

**hccm**  
**Heteroscedasticity-Corrected Covariance Matrices**

**Description**

Calculates heteroscedasticity-corrected covariance matrices for unweighted linear models. These are also called “White-corrected” covariance matrices.

**Usage**

```r
hccm(model, ...)  
## S3 method for class 'lm':  
hccm(model, type=c("hc3", "hc0", "hc1", "hc2", "hc4"), ...)  
## Default S3 method:  
hccm(model, ...)```

**Arguments**

- `model`  
an unweighted linear model, produced by `lm`.
- `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).
- `...`  
arguments to pass to `hccm.lm`.

**Details**

The classical White-corrected coefficient covariance matrix ("hc0") 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.

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


See Also

ncv.test, spread.level.plot

Examples

```r
options(digits=4)
mod<-lm(interlocks~assets+nation, data=Ornstein)
Var(mod)
## (Intercept) assets nationOTH 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
## nationOTH -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 nationOTH 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
## nationOTH -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
```

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

```r
influence.plot(model, ...)
```

## S3 method for class 'lm':

```r
influence.plot(model, scale = 10, col = c(1, 2), labels = names(rstud),
identify.cex=par("cex"), identify.col=par("col"), ...)
```
Arguments
  model a linear or generalized-linear model.
  scale a factor to adjust the size of the circles.
  col colors for plotting points that do not and do have noteworthy Cook’s distances.
  labels if FALSE do not identify points interactively with the mouse; otherwise a vector
         of observation labels.
  identify.cex, identify.col arguments to be passed to identify.
  ... arguments to pass to the plot function.

Value
  Returns the indices of identified points.

Author(s)
  John Fox (jfox@mcmaster.ca)

References

See Also
  cookd, rstudent, hatvalues

Examples
  ## Not run:
  influence.plot(lm(prestige ~ income + education, data=Duncan))

  ## End(Not run)

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.

Source


References


---

**levene.test**

**Levene's Test**

**Description**

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

**Usage**

`levene.test(y, group)`

**Arguments**

- `y` response variable.
- `group` factor defining groups.

**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)

**References**

leverage.plots 

Regression Leverage Plots

Description

These functions display a generalization, due to Sall (1990), 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

leverage.plots(model, term.name, ask=missing(term.name), ...)

leverage.plot(model, ...)

## S3 method for class 'lm':
leverage.plot(model, term.name,
  labels=names(residuals(model)[!is.na(residuals(model))]),
  identify.points=TRUE, las=par('las'), col=palette()[2], pch=1, lwd=2,
  main="Leverage Plot", ...)

## S3 method for class 'glm':
leverage.plot(model, ...)

Arguments

model model object produced by lm
term.name name of term in the model to be plotted; this argument is usually omitted for leverage.plots.
ask if TRUE, a menu is provided in the R Console for the user to select the term(s) to plot.
labels observation names.
identify.points if TRUE, then identify points interactively.
las if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see par).
col color for points and lines; the default is the second entry in the current color palette (see palette and par).
pch plotting character for points; default is 1 (a circle, see par).
lwd line width; default is 2 (see par).
main title for plot.
... arguments passed down to method functions.

Details

The function intended for direct use is leverage.plots. By default, this function is used interactively through a text menu.

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

leverage.plot.glm is a dummy function, which generates an error message.
linear.hypothesis

Value

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

Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

av.plots

Examples

## Not run:
leverage.plots(lm(prestige~(income+education)*type, data=Duncan))
## End(Not run)

linear.hypothesis  Test Linear Hypothesis

Description

Generic function for testing a linear hypothesis, and methods for linear models, generalized linear models, and other models that have methods for coef and vcov.

Usage

linear.hypothesis(model, ...)
lht(model, ...)

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

## S3 method for class 'lm':
linear.hypothesis(model, hypothesis.matrix, rhs=NULL, 
test=c("F", "Chisq"), vcov.=NULL, white.adjust=FALSE, ...)

## S3 method for class 'glm':
linear.hypothesis(model, ...)

## S3 method for class 'glm':
linear.hypothesis(model, ...)

Arguments

model f fitted model object. The default method 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 information.

hypothesis.matrix matrix (or vector) giving linear combinations of coefficients by rows, or a character vector giving the hypothesis in symbolic form (see Details).

rhs 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.

test character 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).

vcov. a function for estimating the covariance matrix of the regression coefficients, e.g., `hccm`, or an estimated covariance matrix for `model`. See also `white.adjust`.

white.adjust logical or character. Convenience interface to `hccm` (instead of using the argument `vcov`). Can be set either to a character specifying the type argument of `hccm` or `TRUE`, in which case "hc3" is used implicitly. For backwards compatibility.

verbose If `TRUE`, the hypothesis matrix and right-hand-side vector are printed to standard output; if `FALSE` (the default), the hypothesis is only printed in symbolic form.

... arguments to pass down.

Details

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 residual sums of squares. For "glm" objects just the default method is called (bypassing the "lm" method).

The function `lht` also dispatches to `linear.hypothesis`.

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 righ-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 positive or negative signs. See the examples below.
Value

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.

Author(s)

Achim Zeleis and John Fox ⟨jfox@mcmaster.ca⟩

References


See Also

anova, Anova,waldtest, hccm, vcovHC, vcovHAC, coef, vcov

Examples

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

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

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

## the following are equivalent:
## use HC3 standard errors via white.adjust option
linear.hypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"),
                  white.adjust = TRUE)
## covariance matrix *function*
linear.hypothesis(mod.davis, c("(Intercept) = 0", "repwt = 1"), vcov = hccm)
## covariance matrix *estimate*
linear.hypothesis(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:
linear.hypothesis(mod.duncan, "1*income - 1*education = 0")
linear.hypothesis(mod.duncan, "(income = education")
linear.hypothesis(mod.duncan, "income - education")
linear.hypothesis(mod.duncan, "1income - leducation = 0")
linear.hypothesis(mod.duncan, "0 - 1*income - 1*education")
linear.hypothesis(mod.duncan, "income-education=0")
linear.hypothesis(mod.duncan, "1*income - 1*education + 1 = 1")
linear.hypothesis(mod.duncan, "2income = 2education")

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)
logit

Description
Compute the logit transformation of proportions or percentages.

Usage
logit(p, percent = max(p, na.rm = TRUE) > 1, adjust)

Arguments
- p: numeric vector or array of proportions or percentages.
- percent: 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 \( \logit(p) = \log\left(\frac{p}{1-p}\right) \) 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)

See Also
prob.axis

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**


**References**


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.
### Moore

**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 (1972) *Canada Year Book*. Statistics Canada [p. 1369].

### References


### 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


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

References


---

**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


References


n.bins

Number of Bins for Histogram

Description

Several rules for calculating the number of bins to use for a histogram.

Usage

n.bins(x, rule=c("freedman.diaconis", "sturges", "scott", "simple"))

Arguments

x numeric vector, variable for histogram
rule see below.

Details

"freedman.diaconis": \( n^{1/3} \times \text{range}/2 \times \text{IQR} \).
"sturges": \( \text{ceiling}(\log_2 n + 1) \).
"scott": \( \text{ceiling}(n^{1/3} \times \text{range}/3.5s) \).
"simple": \( \text{floor}(10 \log_{10} n) \) for \( n > 100 \), or \( \text{floor}(2\sqrt{n}) \) for \( n \leq 100 \).

where \( n \) is the number of observations, \( \text{range} \) is the range of \( x \), \( \text{IQR} \) is the inter-quartile range of \( x \), and \( s \) is the standard deviation of \( x \).

Value

the number of bins.

Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

hist
Examples

attach(UN)
n.bins(gdp)
## [1] 19
n.bins(gdp, "sturges")
## [1] 9
n.bins(gdp, "scott")
## [1] 8
n.bins(gdp, "simple")
## [1] 22
hist(gdp)
hist(gdp, nclass=n.bins(gdp))

ncv.test  

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

ncv.test(model, ...)

## S3 method for class 'lm':
ncv.test(model, var.formula, data=NULL, subset, na.action, ...)

## S3 method for class 'glm':
ncv.test(model, ...)

Arguments

model  
a weighted or unweighted linear model, produced by lm.

var.formula  
a one-sided formula for the error variance; if omitted, the error variance depends on the fitted values.

data  
an optional data frame containing the variables in the model. By default the variables are taken from the environment from which ncv.test 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.

...  
arguments passed down to methods functions.

Details

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

ncv.test.glm is a dummy function to generate an error when a glm model is used.
Value

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

Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

`hccm`, `spread.level.plot`

Examples

```r
ncv.test(lm(interlocks~assets+sector+nation, data=Ornstein))
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 46.98537 Df = 1 p = 7.151835e-12
ncv.test(lm(interlocks~assets+sector+nation, data=Ornstein),
         ~ assets+sector+nation, data=Ornstein)
## Non-constant Variance Score Test
## Variance formula: ~ assets + sector + nation
## Chisquare = 74.73535 Df = 13 p = 1.066320e-10
```

### 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


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

References


---

**outlier.test**  
**Bonferroni Outlier Test**

**Description**

Reports the Bonferroni p-value for the most extreme observation. At present, there are methods for studentized residuals in linear and generalized linear models.

**Usage**

```r
outlier.test(model, ...)  
## S3 method for class 'lm':  
outlier.test(model, labels=names(rstud), ...)  
## S3 method for class 'glm':  
outlier.test(model, labels=names(rstud), ...)  
## S3 method for class 'outlier.test':  
print(x, digits=options("digits")[[1]], ...)  
```

**Arguments**

- `model`  
as a suitable model object.

- `labels`  
an optional vector of observation names.

- `...`  
arguments passed down to methods functions.

- `x`  
`outlier.test` object.

- `digits`  
number of digits for printed output.

**Details**

For a linear model, the p-value reported is for the largest absolute studentized residual, using the $t$ distribution with degrees of freedom one less than the residual df for the model. For a generalized linear model, the largest absolute studentized residual is also used, but with the standard-normal distribution. The Bonferroni adjustment multiplies the usual two-sided p-value by the number of observations.
Value

an object of class outlier.test, which is normally just printed.

Author(s)

John Fox (jfox@mcmaster.ca)

References


Examples

outlier.test(lm(prestige~income+education, data=Duncan))
## max|rstudent| df unadjusted p Bonferroni p
## 3.134519 41 0.003177202 0.1429741
##
## Observation: minister

panel.car Panel Function 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, regression.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.

regression.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 reg.line.
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, reg.line

Examples

coplot(prestige~income|education, panel=panel.car,
col="red", data=Prestige)

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


Personal communication from B. Blishen, W. Carroll, and C. Moore, Departments of Sociology, York University and University of Victoria.

References

**qq.plot**  

*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**

```r
qq.plot(x, ...)  
qqp(...)  
```

```r
## Default S3 method:  
qq.plot(x, distribution="norm", ylab=deparse(substitute(x)),  
xlab=paste(distribution, "quantiles"), main=NULL, las=par("las"),  
envelope=.95, labels=FALSE, col=palette()[2], lwd=2, pch=1, cex=1,  
line=c("quartiles", "robust", "none"), ...)  
```

```r
## S3 method for class 'lm':  
qq.plot(x, main=NULL, xlab=paste(distribution, "Quantiles"),  
ylab=paste("Studentized Residuals("), deparse(substitute(x)), ",")",  
sep = ""),  
distribution=c("t", "norm"), line=c("quartiles", "robust", "none"),  
las=par('las'), simulate=FALSE, envelope=0.95, labels=names(rstudent),  
reps=100, col=palette()[2], lwd=2, pch=1, cex=1, ...)  
```

**Arguments**

- `x` vector of numeric values or `lm` object.
- `distribution` root name of comparison distribution – e.g., `norm` for the normal distribution; `t` for the t-distribution.
- `ylab` label for vertical (empirical quantiles) axis.
- `xlab` label for horizontal (comparison quantiles) axis.
- `main` label for plot.
- `envelope` confidence level for point-wise confidence envelope, or `FALSE` for no envelope.
- `labels` vector of point labels for interactive point identification, or `FALSE` for no labels.
- `las` if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see `par`).
- `col` color for points and lines; the default is the second entry in the current color palette (see `palette` and `par`).
- `pch` plotting character for points; default is 1 (a circle, see `par`).
- `cex` factor for expanding the size of plotted symbols; the default is 1.
- `lwd` line width; default is 2 (see `par`). Confidence envelopes are drawn at half this line width.
line

"quartiles" to pass a line through the quartile-pairs, or "robust" for a robust-regression line; the latter uses the rlm function in the MASS package. Specifying line = "none" 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.

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. Studentized residuals are plotted against the appropriate t-distribution.

The function qqp is an abbreviation for qq.plot.

Value

NULL. These functions are used only for their side effect (to make a graph).

Author(s)

John Fox (jfox@mcmaster.ca)

References


See Also

qqplot, qqnorm, qqline

Examples

x<-rchisq(100, df=2)
qq.plot(x)
qq.plot(x, dist="chisq", df=2)

qq.plot(lm(interlocks~assets+sector+nation, data=Ornstein), sim=TRUE)
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

recode
Recode a Variable

Description
Recodes a numeric vector, character vector, or factor according to simple recode specifications.

Usage
recode(var, recodes, as.factor.result)

Arguments
- var numeric vector, character vector, or factor.
- recodes character string of recode specifications: see below.
- as.factor.result return a factor; default is TRUE if var is a factor, FALSE otherwise.
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.

**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.

Value

a recoded vector of the same length as var; if var is a factor, then so is the result.

Author(s)

John Fox (jfox@mcmaster.ca)

See Also

cut, factor

Examples

```r
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"
```

---

**reg.line**

Plot Regression Line

Description

Plots a regression line on a scatterplot; the line is plotted between the minimum and maximum x-values.

Usage

```r
reg.line(mod, col=palette()[2], lwd=2, lty=1,...)
```
Arguments

mod  a model, such as produced by lm, that responds to the coefficients 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 second entry in the current color palette (see palette and par).
lwd  line width; default is 2 (see par).
lty  line type; default is 1, a solid line (see par).
...  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)
reg.line(lm(repwt~weight, subset=sex=="M", data=Davis))
reg.line(lm(repwt~weight, subset=sex=="F", data=Davis), lty=2)

---

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


References


---

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


---

scatterplot.matrix

*Scatterplot Matrices*

Description

Scatterplot matrices with univariate displays down the diagonal; spm is an abbreviation for scatterplot.matrix. This function just sets up a call to `pairs`.
scatplot.matrix

Usage

scatplot.matrix(x, ...)

## S3 method for class 'formula':
somatplot.matrix(formula, data=NULL, subset, ...)

## Default S3 method:
somatplot.matrix(x, labels=colnames(x),
    diagonal=c("density", "boxplot", "histogram", "qqplot", "none"),
    adjust=1, nclass, plot.points=TRUE, smooth=TRUE, span=0.5, reg.line=lm,
    transform=FALSE, ellipse=FALSE, levels=c(.5, .9), robust=FALSE,
    groups=FALSE, by.groups=FALSE, col=palette(),
    pch=1:n.groups, lwd=1,
    cex=par("cex"), cex.axis=par("cex.axis"), cex.labels=NULL,
    cex.main=par("cex.main"),
    legend.plot=length(levels(groups)) > 1, ...)

spm(x, ...)

Arguments

x
    a data matrix, numeric data frame, or formula.

formula
    a one-side "model" formula, of the form ~ x1 + x2 + ... + xk or ~
    x1 + x2 + ... + xk | z where z evaluates to a factor or other vari-
    able to divide the data into groups.

data
    for scatplot.matrix.formula,a data frame within which to evalu-
    ate the formula.

subset
    expression defining a subset of observations.

labels
    variable labels (for the diagonal of the plot).

diagonal
    contents of the diagonal panels of the plot.

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.

smooth
    if TRUE a lowess smooth is plotted in each off-diagonal panel.

span
    span for lowess smoother.

reg.line
    if not FALSE a line is plotted using the function given by this argument; e.g.,
    using rlm in package MASS plots a robust-regression line.

transform
    if TRUE, multivariate normalizing Box-Cox transformations are computed and
    plotted; if a vector of powers, one for each variable, these are applied as Box-
    Cox power transformations prior to plotting.

ellipse
    if TRUE data-concentration ellipses are plotted in the off-diagonal panels.

levels
    levels or levels at which concentration ellipses are plotted; the default is c(.5,
    .9).

robust
    if TRUE use the cov.trob function in the MASS package to calculate the cen-
    ter and covariance matrix for the data ellipse.

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.
pch  plotting characters for points; default is the plotting characters in order (see \texttt{par}).
col  colors for points and lines; the default is the in the current color palette, starting at the second entry (see \texttt{palette} and \texttt{par}).
lwd  width for lines.
cex, cex.axis, cex.labels, cex.main  set sizes of various graphical elements; (see \texttt{par}).
legend.plot  if TRUE then a legend for the groups is plotted in the bottom-right cell.
...  arguments to pass down.

\textbf{Value}

\texttt{NULL}. This function is used for its side effect: producing a plot.

\textbf{Author(s)}

John Fox (jfox@mcmaster.ca)

\textbf{See Also}

\texttt{pairs}, \texttt{scatterplot}, \texttt{data.ellipse}, \texttt{box.cox.powers}, \texttt{box.cox}, \texttt{cov.trob}.

\textbf{Examples}

\begin{verbatim}
scatterplot.matrix(~income + education + prestige | type, data=Duncan)
scatterplot.matrix(~income + education + prestige,
    transform=TRUE, data=Duncan)
\end{verbatim}

\begin{verbatim}
scatterplot
\end{verbatim}

\textit{Scatterplots with Boxplots}

\textbf{Description}

Makes fancy scatterplots, with boxplots in the margins, a lowess smooth, and a regression line; \texttt{sp} is an abbreviation for \texttt{scatterplot}.

\textbf{Usage}

\begin{verbatim}
scatterplot(x, ...)
\end{verbatim}

\begin{verbatim}
## S3 method for class 'formula':
scatterplot(formula, data, xlab, ylab, subset, labels=FALSE, ...)
\end{verbatim}

\begin{verbatim}
## Default S3 method:
scatterplot(x, y, smooth=TRUE, span=0.5, reg.line=lm,
    boxplots="xy", xlab=deparse(substitute(x)), ylab=deparse(substitute(y)),
    las=par('las'), lwd=1, labels=FALSE, 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"),
\end{verbatim}
scatterplot

```
sp(...)```

**Arguments**

- `formula` "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.
- `data` data frame within which to evaluate the formula.
- `subset` expression defining a subset of observations.
- `x` vector of horizontal coordinates.
- `y` vector of vertical coordinates.
- `smooth` if TRUE a lowess nonparametric regression line is drawn on the plot.
- `span` span for the lowess smooth.
- `reg.line` function to draw a regression line on the plot or FALSE not to plot a regression line.
- `boxplots` if "x" a boxplot for \( x \) is drawn above the plot; if "y" a boxplot for \( y \) is drawn to the right of the plot; if "xy" both boxplots are drawn.
- `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 `par`).
- `lwd` width of plotted lines.
- `labels` if not FALSE a vector of point labels, to be used interactively to identify points on the plot.
- `log` same as the log argument to `plot`, 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 `jitter` 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 NULL, determined from the data.
- `ylim` the y limits (min,max) of the plot; if NULL, 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.
- `ellipse` if TRUE data-concentration ellipses are plotted.
- `levels` level or levels at which concentration ellipses are plotted; the default is \( c(.5, .9) \).
- `robust` if TRUE use the `cov.trob` function in the MASS package to calculate the center and covariance matrix for the data ellipse.
- `col` colors for points and lines; the default is the current color palette, starting at the second entry (see `palette` and `par`).
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
some

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.

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

```r
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.
Details

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

spread.level.plot(x, ...)

slp(x, ...)

## S3 method for class 'formula':
spread.level.plot(formula, data=NULL, subset, na.action,
        main=paste("Spread-Level Plot for", varnames[response],
               "by", varnames[-response]), ...)

## Default S3 method:
spread.level.plot(x, by,
        robust.line=any("MASS"==.packages(all=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()[-2], pch=1, lwd=2, ...)

## S3 method for class 'lm':
spread.level.plot(x, start=0,
        robust.line=any("MASS"==.packages(all=TRUE)),
        xlab="Fitted Values",
        ylab="Absolute Studentized Residuals", las=par("las")
        main=paste("Spread-Level Plot for", deparse(substitute(x))),
        pch=1, col=palette()[-2], lwd=2, ...)

## S3 method for class 'spread.level.plot':
print(x, ...)

Examples

some(data.frame(z=rnorm(100), u=runif(100)))
Arguments

x  a formula or an lm object to be plotted; alternatively a numeric vector.
formula a formula of the form y~x, where y is a numeric vector and x is a factor.
data an optional data frame containing the variables to be plotted. By default the variables are taken from the environment from which spread.level.plot 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.
by a factor, numeric or character vector defining groups.
robust.line if TRUE a robust line is fit using the rlm function in the MASS package; if FALSE a line is fit using lm.
start add the constant start to each data value.
main title for the plot.
xlab label for horizontal axis.
ylab label for vertical axis.
point.labels if TRUE label the points in the plot with group names.
las if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see par).
col color for points and lines; the default is the second entry in the current color palette (see palette and par).
pch plotting character for points; default is 1 (a circle, see par).
lwd line width; default is 2 (see par).
... arguments passed to plotting functions.

Details

Except for linear models, computes the statistics for, and plots, a Tukey spread-level plot of log(hinge-spread) vs. log(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(abs(studentized residuals) vs. log(fitted values).

The function slp is an abbreviation for spread.level.plot.

Value

A list 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 – slope of the line fit to the plot.

Author(s)

John Fox (jfox@mcmaster.ca)
References


See Also

hccm, ncv.test

Examples

spread.level.plot(interlocks+1~nation, data=Ornstein)
  ## Loading required package: MASS
  ## LowerHinge Median UpperHinge Hinge-Spread
  ## US     2  6.0   13   11
  ## UK     4  9.0   14   10
  ## CAN    6 13.0   30   24
  ## OTH    4 15.5   24   20
  ##
  ## Suggested power transformation: 0.1534487

slp(lm(interlocks ~ assets + sector + nation, data=Ornstein))
  ## Suggested power transformation: 0.3222165
  ## Warning message:
  ## Start = 3 added to fitted values to avoid 0 or negative values. in: spread.level.plot

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


References


---

### 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

```r
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,
    statistic=c("bic", "cp", "adjr2", "rsq", "rss"),
    las=par('las'), cex.subsets=1, ...
)
```

### Arguments

- **object**: a `regsubsets` object produced by the `regsubsets` function in the `leaps` package.
- **names**: 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 `object`.
- **abbrev**: minimum number of characters to use in abbreviating predictor names.
- **min.size**: minimum size subset to plot; default is 1.
- **max.size**: maximum size subset to plot; default is number of predictors.
- **legend**: TRUE to plot a legend of predictor names; defaults to TRUE if abbreviations are computed for predictor names. The legend is placed on the plot interactively with the mouse.
- **statistic**: statistic to plot for each predictor subset; one of: "bic", Bayes Information Criterion; "cp", Mallows C_p; "adjr2", R^2 adjusted for degrees of freedom; "rsq", unadjusted R^2; "rss", residual sum of squares.
- **las**: if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see `par`).
- **cex.subsets**: can be used to change the relative size of the characters used to plot the regression subsets; default is 1.
- **...**: arguments to be passed down to `subsets.regsubsets` and `plot`. 
Boxplots for transformations to symmetry

Description

`symbox` 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

```r
symbox(x, powers=c(-1, -.5, 0, .5, 1), start=0)
```

Arguments

- `x`: a numeric vector.
- `powers`: a vector of selected powers to which `x` is to be raised. A power of 0 is taken to mean \( \log(x) \). Negative powers are taken to mean \(-x^p\), to preserve the order of the data. For meaningful comparison of powers, 1 should be included in the vector of powers.
- `start`: a constant to be added to `x`; after adding the start, all data values must be positive.

Value

as returned by `boxplot`.

Author(s)

Gregor Gorjanc, John Fox (jfox@mcmaster.ca)
Transformation Axes

References


See Also

`boxplot, boxcox, box.cox`

Examples

```r
symbox(Prestige$income)
```

Transformation Axes

*Axes for Transformed Variables*

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

```r
power.axis(power, base=exp(1), side=c("right", "above", "left", "below"),
at, grid=FALSE, grid.col=gray(0.5), grid.lty=3,
axis.title="Untransformed Data", cex=1, las=par("las"))

box.cox.axis(power, side=c("right", "above", "left", "below"),
at, grid=FALSE, grid.col=gray(0.5), grid.lty=3,
axis.title="Untransformed Data", cex=1, las=par("las"))

prob.axis(at, side=c("right", "above", "left", "below"), grid=FALSE, grid.lty=3,
grid.col=gray(0.5), axis.title="Probability", interval=0.1, cex=1, las=par("la
```

Arguments

- `power` - power for Box-Cox or power transformation.
- `side` - side at which the axis is to be drawn; numeric codes are also permitted: `side = 1` for the bottom of the plot, `side=2` for the left side, `side = 3` for the top, `side = 4` for the right side.
- `at` - numeric vector giving location of tick marks on original scale; if missing, the function will try to pick nice locations for the ticks.
- `grid` - if TRUE grid lines for the axis will be drawn.
- `grid.col` - color of grid lines.
- `grid.lty` - line type for grid lines.
- `axis.title` - title for axis.
- `cex` - relative character expansion for axis label.
Transformation Axes

las if 0, ticks labels are drawn parallel to the axis; set to 1 for horizontal labels (see par).
base base of log transformation for power.axis when power = 0.
interval desired interval between tick marks on the probability scale.

Details

The transformations corresponding to the three functions are as follows:

power.axis: \( x' = x^p \) for \( p \neq 0 \) and \( x' = \log x \) for \( p = 0 \).
box.cox.axis: \( x' = (x^\lambda - 1)/\lambda \) for \( \lambda \neq 0 \) and \( x' = \log x \) for \( \lambda = 0 \).
prob.axis: \( \text{logit} = \log[p/(1 - p)] \).

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)

See Also

box.cox, logit

Examples

UN<-na.omit(UN)
attach(UN)
par(mar=c(5, 4, 4, 4)+.1)
plot(log(gdp, 10), log(infant.mortality, 10))
power.axis(0, base=10, side="above",
    at=c(50,200,500,2000,5000,20000),grid=TRUE, axis.title="GDP per capita")
power.axis(0, base=10, side="right",
    at=c(5,10,20,50,100), grid=TRUE, axis.title="infant mortality rate per 1000")
plot(box.cox(gdp, 0), box.cox(infant.mortality, 0))
box.cox.axis(0, side="above",
    grid=TRUE, axis.title="GDP per capita")
box.cox.axis(0, side="right",
    grid=TRUE, axis.title="infant mortality rate per 1000")

qq.plot(logit(infant.mortality/1000))
prob.axis()

qq.plot(logit(infant.mortality/1000))
prob.axis(c(.005, .01, .02, .04, .08, .16))
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 US dollars.

Source

US.pop Population of the United States

Description
The US.pop data frame has 21 rows and 1 columns. This is a decennial time-series, from 1790 to 1990.

Usage
US.pop

Format
This data frame contains the following columns:

- **year** census year.
- **population** Population in millions.

Source

References
Var

Variance-Covariance Matrices (deprecated)

Description

Computes variance-covariance matrices or variances for model objects or data. The default method uses the function \texttt{var}.

These functions are now deprecated; instead, use the \texttt{vcov} function, now in the base package. Note that \texttt{vcov} has no \texttt{diagonal} argument and no default method.

Usage

\begin{verbatim}
Var(object, ...) \\
## Default S3 method: 
Var(object, diagonal=FALSE, ...) \\
## S3 method for class 'lm': 
Var(object, diagonal=FALSE, ...) \\
## S3 method for class 'glm': 
Var(object, diagonal=FALSE, ...) 
\end{verbatim}

Arguments

\begin{itemize}
\item \texttt{object} an object for which the covariance matrix is desired.
\item \texttt{...} arguments to be passed to \texttt{var} (e.g., \texttt{na.rm}).
\item \texttt{diagonal} if TRUE, return only the variances.
\end{itemize}

Value

A variance-covariance matrix or a vector of variances.

Author(s)

John Fox (jfox@mcmaster.ca)

See Also

\texttt{var}

Examples

\begin{verbatim}
data(Davis)
attach(Davis)
Var(cbind(weight, repwt), na.rm=TRUE) \\
## weight   repwt \\
## weight 233.8781 176.1014 \\
## repwt 176.1014 189.7966

Var(lm(weight~repwt)) \\
## (Intercept) repwt
\end{verbatim}
vif

## (Intercept)  9.2228211 -0.134640952
## repwt      -0.1346410  0.002051736

---

Description

 Calculates variance-inflation and generalized variance-inflation factors for linear and generalized linear models.

Usage

vif(mod)

## S3 method for class 'lm':
vif(mod)

Arguments

mod an object that inherits from class \texttt{lm}, such as an \texttt{lm} or \texttt{glm} object.

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 $GVIF^{1/(2 \times df)}$.

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, that inherit from class \texttt{lm}.

Value

A vector of vifs, or a matrix containing one row for each term in the model, and columns for the GVIF, df, and $GVIF^{1/(2 \times df)}$. 

Author(s)

Henric Nilsson and John Fox (jfox@mcmaster.ca)

References


Examples

vif(lm(prestige~income+education, data=Duncan))
## income education
## 2.104900 2.104900
vif(lm(prestige~income+education+type, data=Duncan))
## GVIF Df GVIF^(1/2Df)
## income 2.209178 1 1.486330
## education 5.297584 1 2.301648
## type 5.098592 2 1.502666

Vocab

Vocabulary and Education

Description

The Vocab data frame has 968 rows and 2 columns. The observations are respondents to the 1989 U. S. General Social Survey.

Usage

Vocab

Format

This data frame contains the following columns:

education Education, in years.
vocabulary Vocabulary test score: number correct on a 10-word test.

Source


References


which.names

Position of Row Names

Description

This function returns the indices of row names in a data frame or a vector of names.

Usage

which.names(names, object)
Arguments

names a name or character vector of names.
object a data frame or character vector of (row) names.

Value

Returns the index or indices of names within object.

Author(s)

John Fox (jfox@mcmaster.ca)

Examples

which.names(c('minister', 'conductor'), Duncan)
## [1] 6 16

Womenlf

Canadian Women’s Labour-Force Participation

Description

The Womenlf data frame has 263 rows and 4 columns. The data are from a 1977 survey of the Canadian population.

Usage

Womenlf

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

Index

*Topic `aplot`
  Ellipses, 32
  panel.car, 56
  reg.line, 61
  Transformation Axes, 74

*Topic `datasets`
  Adler, 2
  Angell, 2
  Anscombe, 5
  Baumann, 8
  Bfox, 9
  Blackmoor, 10
  Burt, 16
  Can.pop, 17
  Chile, 22
  Chirot, 23
  Cowles, 26
  Davis, 29
  DavisThin, 29
  Duncan, 30
  Ericksen, 34
  Florida, 34
  Freedman, 35
  Friendly, 36
  Ginzberg, 36
  Greene, 37
  Guyer, 38
  Hartnagel, 39
  Leinhardt, 42
  Mandel, 49
  Migration, 49
  Moore, 50
  Mroz, 51
  Ornstein, 54
  Prestige, 57
  Quartet, 60
  Robey, 62
  Sahlins, 63
  SLID, 67
  States, 71
  UN, 76
  US.pop, 76
  Vocab, 79

  *Womenlf, 80

*Topic `distribution`
  qq.plot, 58

*Topic `dpplot`
  n.bins, 52

*Topic `hplot`
  av.plots, 7
  ceres.plots, 21
  cr.plots, 27
  Ellipses, 32
  leverage.plots, 44
  scatterplot, 65
  scatterplot.matrix, 63
  spread.level.plot, 69
  subsets, 72
  symbox, 73

*Topic `htest`
  Anova, 3
  levene.test, 43
  linear.hypothesis, 45
  ncv.test, 53
  outlier.test, 55

*Topic `internal`
  car-internal, 17

*Topic `iplot`
  Ask, 6

*Topic `manip`
  box.cox, 12
  box.cox.var, 13
  logit, 48
  recode, 60

*Topic `misc`
  Ask, 6
  Var, 77

*Topic `models`
  Anova, 3
  box.cox.powers, 10
  Contrasts, 24
  Cook’s Distances, 25
  linear.hypothesis, 45

*Topic `multivariate`
  box.cox.powers, 10

*Topic `package`
INDEX

car-package, 18

*Topic regression
  
  Anova, 3
  av.plot, 7
  box.cox, 12
  box.cox.var, 13
  box.tidwell, 15
  ceres.plot, 21
  Contrasts, 24
  Cook’s Distances, 25
  cr.plot, 27
  durbin.watson, 31
  hccm, 40
  influence.plot, 41
  leverage.plot, 44
  linear.hypothesis, 45
  ncv.test, 53
  outlier.test, 55
  qq.plot, 58
  spread.level.plot, 69
  subsets, 72
  vif, 78

*Topic ts
  
  durbin.watson, 31

*Topic univar
  
  qq.plot, 58

*Topic utilities
  
  some, 68
    which.names, 79

abline, 62
Adler, 2
Angell, 2
Anova, 3, 47
anova, 5, 47
Anova.II.F.glm (Anova), 3
Anova.II.lm (Anova), 3
Anova.II.LR.glm (Anova), 3
Anova.II.multinom (Anova), 3
Anova.II.polr (Anova), 3
Anova.II.Wald.glm (Anova), 3
Anova.III.F.glm (Anova), 3
Anova.III.lm (Anova), 3
Anova.III.LR.glm (Anova), 3
Anova.III.multinom (Anova), 3
Anova.III.polr (Anova), 3
Anova.III.Wald.glm (Anova), 3
Anova.lm (Anova), 3
Anova.multinom (Anova), 3
Anova.polr (Anova), 3
Anscowbe, 5
Ask, 6
av.plot (av.plot), 7
av.plot (av.plot), 7
av.plot (av.plot), 7
av.plot (av.plot), 7
av.plot (av.plot), 7
Baumann, 8
bc (box.cox), 12
Bfox, 9
Blackmoor, 10
box.cox, 11, 12, 14, 65, 74, 75
box.cox.var, 11, 13, 14
box.cox.axis (Transformation Axes), 74
box.cox.powers, 10, 12–14, 65
box.cox.var, 11, 13, 14
box.tidwell, 15
boxcox, 11, 13, 14, 74
boxplot, 67, 74
Burt, 16
Can.pop, 17
car (car-package), 18
car-internal, 17
car-package, 18
ceres.plot (ceres.plot), 21
ceres.plot, 8, 21, 28
Chile, 22
Chirot, 23
coeff, 47
certainty.ellipse (Ellipses), 32
contrast.Helmert (Contrasts), 24
contrast.helmert, 25
contrast.poly, 25
contrast.Sum (Contrasts), 24
contrast.sum, 25
contrast.Treatment (Contrasts), 24
contrast.treatment, 25
Contrasts, 24
Cook’s Distances, 25
cookd, 42
cookd (Cook’s Distances), 25
cooks.distance, 26
cpplot, 57
cov.trob, 33, 65, 67
Cowles, 26
cr.plot (cr.plot), 27
cr.plot, 8, 22, 27
crp (cr.plot), 27
cut, 61
data.ellipse, 65, 67
data.ellipse (Ellipses), 32
Davis, 29
DavisThin, 29
df.terms (car-internal), 17
INDEX

tail, 69
term.names (car-internal), 17
Transformation Axes, 74

UN, 76
US.pop, 76

Var, 77
var, 77
vcov, 47
vcovHAC, 47
vcovHC, 47
vif, 78
Vocab, 79

waldtest, 47
which.names, 79
Womenlf, 80