Package 'effects'

January 2, 2012

Version 2.1-0

Date 2011/08/02

Title Effect Displays for Linear, Generalized Linear,Multinomial-Logit, Proportional-Odds Logit Models and
mixed-effects models

Author John Fox <jfox@mcmaster.ca>, Sanford Weisberg, and Jangman Hong. We are grateful to Robert Andersen, David Firth, and Michael Friendly, for various suggestions.

Maintainer John Fox <jfox@mcmaster.ca>

Depends R (>= 2.10), lattice, grid, nlme, MASS, nnet, colorspace

LazyLoad yes

LazyData yes

Description Graphical and tabular effect displays, e.g., of interactions, for linear generalized linear, multinomial-logit,and proportional-odds logit models.

License GPL (>= 2)

URL http://www.r-project.org, http://socserv.socsci.mcmaster.ca/jfox/

Repository CRAN

Repository/R-Forge/Project effects

Repository/R-Forge/Revision 61

Date/Publication 2011-08-04 11:57:13

R topics documented:

`effects-package` *Effect Displays for Linear, Generalized Linear, Multinomial-Logit, and Proportional-Odds Logit Models*

Description

Graphical and tabular effect displays, e.g., of interactions, for linear generalized linear, multinomial-logit, and proportional-odds logit models.

Details

Package:	effects
Version:	2.1-0
Date:	2011/08/02
Depends:	R (>= 2.4.0), lattice, grid, MASS, nnet, colorspace
LazyLoad:	yes
LazyData:	yes
License:	GPL (>= 2)
URL:	http://www.r-project.org, http://socserv.socsci.mcmaster.ca/jfox/

Index:

effect	Functions For Constructing Effect Plots
effects-deprecated	Deprecated Function in effects Package
summary.eff	Summarizing, Printing, and Plotting Effects

This package creates effect displays for various kinds of models, as explained in the references. Typical usage is `plot(allEffects(model))` or `plot(allEffects(model), ask=FALSE)`, where `model` is an appropriate fitted-model object. Additional arguments to `allEffects` and `plot` can be used to customize the resulting displays. The function `effect` can be employed to produce an effect display for a particular term in the model, or to which terms in the model are marginal. See `?effect` and `?plot.eff` for details.

Author(s)

John Fox <jfox@mcmaster.ca>, Sanford Weisberg, and Jangman Hong. We are grateful to Robert Andersen, David Firth, and Michael Friendly, for various suggestions.

Maintainer: John Fox <jfox@mcmaster.ca>

References

Fox, J. (1987) Effect displays for generalized linear models. *Sociological Methodology* **17**, 347–361.

Fox, J. (2003) Effect displays in R for generalised linear models. *Journal of Statistical Software* **8:15**, 1–27, <http://www.jstatsoft.org/counter.php?id=75&url=v08/i15/effect-displays-revised.pdf&ct=1>.

Fox, J. and R. Andersen (2006) Effect displays for multinomial and proportional-odds logit models. *Sociological Methodology* **36**, 225–255.

Arrests *Arrests for Marijuana Possession*

Description

Data on police treatment of individuals arrested in Toronto for simple possession of small quantities of marijuana. The data are part of a larger data set featured in a series of articles in the Toronto Star newspaper.

Usage

```
Arrests
```

Format

A data frame with 5226 observations on the following 8 variables.

released Whether or not the arrestee was released with a summons; a factor with levels: `No`; `Yes`.

colour The arrestee's race; a factor with levels: `Black`; `White`.

year 1997 through 2002; a numeric vector.

age in years; a numeric vector.

sex a factor with levels: `Female`; `Male`.

employed a factor with levels: `No`; `Yes`.

citizen a factor with levels: `No`; `Yes`.

checks Number of police data bases (of previous arrests, previous convictions, parole status, etc. – 6 in all) on which the arrestee's name appeared; a numeric vector

Source

Personal communication from Michael Friendly, York University.

Examples

```
summary(Arrests)
```

`BEPS` *British Election Panel Study*

Description

These data are drawn from the 1997-2001 British Election Panel Study (BEPS).

Usage

```
BEPS
```

Format

A data frame with 1525 observations on the following 10 variables.

`vote` Party choice: `Conservative`, `Labour`, or `Liberal Democrat`

`age` in years

`economic.cond.national` Assessment of current national economic conditions, 1 to 5.

`economic.cond.household` Assessment of current household economic conditions, 1 to 5.

`Blair` Assessment of the Labour leader, 1 to 5.

`Hague` Assessment of the Conservative leader, 1 to 5.

`Kennedy` Assessment of the leader of the Liberal Democrats, 1 to 5.

`Europe` an 11-point scale that measures respondents' attitudes toward European integration. High scores represent 'Eurosceptic' sentiment.

`political.knowledge` Knowledge of parties' positions on European integration, 0 to 3.

`gender` `female` or `male`.

References

J. Fox and R. Andersen (2006) Effect displays for multinomial and proportional-odds logit models. *Sociological Methodology* **36**, 225–255.

Examples

```
summary(BEPS)

require(splines)  # for bs()
beps <- multinom(vote ~ age + gender + economic.cond.national + economic.cond.household
+ Blair + Hague + Kennedy + bs(Europe, 3)*political.knowledge, data=BEPS)
europe.knowledge <- effect("bs(Europe, 3)*political.knowledge", beps,
xlevels=list(Europe=seq(1, 11, length=50), political.knowledge=0:3),
   given.values=c(gendermale=0.5))

plot(europe.knowledge)

plot(europe.knowledge, style="stacked", colors=c("blue", "red", "orange"), rug=FALSE)
```

Cowles *Cowles and Davis's Data on Volunteering*

Description

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

Usage

```
Cowles
```

Format

This data frame contains the following columns:

neuroticism scale from Eysenck personality inventory.

extraversion scale from Eysenck personality inventory.

sex a factor with levels: female; male.

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

Source

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

Examples

```
summary(Cowles)
```

effect *Functions For Constructing Effect Plots*

Description

effect constructs an "eff" object for a term (usually a high-order term) in a linear (fit by lm or gls) or generalized linear model (fit by glm), or an "effpoly" object for a term in a multinomial or proportional-odds logit model (fit respectively by multinom or polr), absorbing the lower-order terms marginal to the term in question, and averaging over other terms in the model. The function can also be used with mixed-effects models fit with lmer from the lme4 package, or fit with lme from the nlme package. In mixed effects models the analysis is for the fixed effects only, not for random effects.

allEffects identifies all of the high-order terms in a model and returns a list of "eff" or "effpoly" objects (i.e., an object of type "efflist").

Usage

```
effect(term, mod, ...)

## S3 method for class 'lm'
effect(term, mod, xlevels=list(), default.levels=10, given.values,
se=TRUE, confidence.level=.95,
    transformation=list(link=family(mod)$linkfun, inverse=family(mod)$linkinv),
    typical=mean, ...)

## S3 method for class 'gls'
effect(term, mod, xlevels=list(), default.levels=10, given.values,
se=TRUE, confidence.level=.95, transformation=NULL, typical=mean, ...)

## S3 method for class 'multinom'
effect(term, mod, confidence.level=.95, xlevels=list(), default.levels=10,
given.values, se=TRUE, typical=mean, ...)

## S3 method for class 'polr'
effect(term, mod, confidence.level=.95, xlevels=list(), default.levels=10,
given.values, se=TRUE, typical=mean, latent=FALSE, ...)

## S3 method for class 'mer'
effect(term, mod, ...)

## S3 method for class 'lme'
effect(term, mod, ...)

allEffects(mod, ...)

## S3 method for class 'mer'
```

```
allEffects(mod, ...)

## S3 method for class 'lme'
allEffects(mod, ...)

## S3 method for class 'eff'
as.data.frame(x, row.names=NULL, optional=TRUE, ...)

## S3 method for class 'effpoly'
as.data.frame(x, row.names=NULL, optional=TRUE, ...)

## S3 method for class 'efflatent'
as.data.frame(x, row.names=NULL, optional=TRUE, ...)

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

Arguments

`term`	the quoted name of a term, usually, but not necessarily, a high-order term in the model. The term must be given exactly as it appears in the printed model, although either colons (`:`) or asterisks (`*`) may be used for interactions.
`mod`	an object of class `"lm"`, `"gls"`, `"glm"`, `"multinom"`, `"polr"` `"mer"` or `"lme"`.
`xlevels`	an optional list of values at which to set covariates, with components of the form `covariate.name = vector.of.values`.
`default.levels`	number of values for covariates that are not specified explicitly via `xlevels`; covariate values set by default are evenly spaced between the minimum and maximum values in the data.
`given.values`	a numeric vector of named elements, setting particular columns of the model matrix to specific values for terms *not* appearing in an effect; if specified, this argument takes precedence over the application of the function given in the `typical` argument (below). Care must be taken in specifying these values — e.g., for a factor, the values of all contrasts should be given and these should be consistent with each other.
`se`	if `TRUE`, the default, calculate standard errors and confidence limits for the effects.
`confidence.level`	level at which to compute confidence limits based on the standard-normal distribution; the default is `0.95`.
`transformation`	a two-element list with elements `link` and `inverse`. For a generalized linear model, these are by default the link function and inverse-link (mean) function. For a linear model, these default to `NULL`. If `NULL`, the identify function, `I`, is used; this effect can also be achieved by setting the argument to `NULL`. The inverse-link may be used to transform effects when they are printed or plotted; the link may be used in positioning axis labels (see below). If the link is not given, an attempt will be made to approximate it from the inverse-link.

typical	a function to be applied to the columns of the model matrix over which the effect is "averaged"; the default is mean.
latent	if TRUE, effects in a proportional-odds logit model are computed on the scale of the latent response; if FALSE (the default) effects are computed as individual-level probabilities and logits.
x	an object of class "eff" or "effpoly".
row.names, optional	not used.
object	an object of class "eff" for which the covariance matrix of the effects is desired.
...	arguments to be passed down.

Details

Normally, the functions to be used directly are allEffects, to return a list of high-order effects, and the generic plot function to plot the effects. (see plot.efflist, plot.eff, and plot.effpoly). Plots are drawn using the xyplot (or in some cases, the densityplot) function in the lattice package. Effects may also be printed (implicitly or explicitly via print) or summarized (using summary) (see print.efflist, summary.efflist, print.eff, summary.eff, print.effpoly, and summary.effpoly).

If asked, the effect function will compute effects for terms that have higher-order relatives in the model, averaging over those terms (which rarely makes sense), or for terms that do not appear in the model but are higher-order relatives of terms that do. For example, for the model Y ~ A*B + A*C + B*C, one could compute the effect corresponding to the absent term A:B:C, which absorbs the constant, the A, B, and C main effects, and the three two-way interactions. In either of these cases, a warning is printed.

In calculating effects, the strategy for 'safe' prediction described in Hastie (1992: Sec. 7.3.3) is employed.

Value

For lm, glm, mer and lme, effect returns an "eff" object, and for multinom and polr, an "effpoly" object, with the following components:

term	the term to which the effect pertains.
formula	the complete model formula.
response	a character string giving the name of the response variable.
y.levels	(for "effpoly" objects) levels of the polytomous response variable.
variables	a list with information about each predictor, including its name, whether it is a factor, and its levels or values.
fit	(for "eff" objects) a one-column matrix of fitted values, representing the effect on the scale of the linear predictor; this is a ravelled table, representing all combinations of predictor values.
prob	(for "effpoly" objects) a matrix giving fitted probabilities for the effect for the various levels of the the response (columns) and combinations of the focal predictors (rows).

`logit` (for `"effpoly"` objects) a matrix giving fitted logits for the effect for the various levels of the the response (columns) and combinations of the focal predictors (rows).

`x` a data frame, the columns of which are the predictors in the effect, and the rows of which give all combinations of values of these predictors.

`model.matrix` the model matrix from which the effect was calculated.

`data` a data frame with the data on which the fitted model was based.

`discrepancy` the percentage discrepancy for the 'safe' predictions of the original fit; should be very close to 0.

`model` (for `"effpoly"` objects) `"multinom"` or `"polr"`, as appropriate.

`vcov` (for `"eff"` objects) a covariance matrix for the effect, on the scale of the linear predictor.

`se` (for `"eff"` objects) a vector of standard errors for the effect, on the scale of the linear predictor.

`se.prob, se.logit` (for `"effpoly"` objects) matrices of standard errors for the effect, on the probability and logit scales.

`lower, upper` (for `"eff"` objects) one-column matrices of confidence limits, on the scale of the linear predictor.

`lower.prob, upper.prob, lower.logit, upper.logit` (for `"effpoly"` objects) matrices of confidence limits for the fitted logits and probabilities; the latter are computed by transforming the former.

`confidence.level` for the confidence limits.

`transformation` (for `"eff"` objects) a two-element list, with element `link` giving the link function, and element `inverse` giving the inverse-link (mean) function.

`effectList` returns a list of `"eff"` or `"effpoly"` objects corresponding to the high-order terms of the model.

Warnings and Limitations

The `effect` function handles factors and covariates differently, and becomes confused if one is changed to the other in a model formula. Consequently, formulas that include calls to `as.factor`, `factor`, or `numeric` (as, e.g., in `as.factor(income)`) will cause errors. Instead, create the modified variables outside of the model formula (e.g., `fincome <- as.factor(income)`) and use these in the model formula.

Factors cannot have colons in level names (e.g., `"level:A"`); the `effect` function will confuse the colons with interactions; rename levels to remove or replace the colons (e.g., `"level.A"`).

Binomial generalized linear models cannot have a matrix of successes and failures on the left-hand side of the model formula; instead specify the *proportion* of successes (i.e., successes/(successes + failures)) as the response, and give the number of binomial trials (i.e., successes + failures) in the weights argument to `glm`.

Author(s)

John Fox <jfox@mcmaster.ca> and Jangman Hong. Extension to mer and lme objects by Sanford Weisberg <sandy@umn.edu>.

References

Fox, J. (1987) Effect displays for generalized linear models. *Sociological Methodology* **17**, 347–361.

Fox, J. (2003) Effect displays in R for generalised linear models. *Journal of Statistical Software* **8:15**, 1–27, <http://www.jstatsoft.org/v08/i15/>.

Fox, J. and R. Andersen (2006) Effect displays for multinomial and proportional-odds logit models. *Sociological Methodology* **36**, 225–255.

Fox, J. and J. Hong (2009). Effect displays in R for multinomial and proportional-odds logit models: Extensions to the effects package. *Journal of Statistical Software* **32:1**, 1–24.", <http://www.jstatsoft.org/v32/i01/>.

Hastie, T. J. (1992) Generalized additive models. In Chambers, J. M., and Hastie, T. J. (eds.) *Statistical Models in S*, Wadsworth.

See Also

print.eff, summary.eff, plot.eff, print.summary.eff, print.effpoly, summary.effpoly, plot.effpoly, print.efflist, summary.efflist, plot.efflist, xyplot, densityplot

Examples

```
mod.cowles <- glm(volunteer ~ sex + neuroticism*extraversion,
    data=Cowles, family=binomial)
eff.cowles <- allEffects(mod.cowles, xlevels=list(neuroticism=0:24,
    extraversion=seq(0, 24, 6)), given.values=c(sexmale=0.5))
eff.cowles

plot(eff.cowles, 'sex', ylab="Prob(Volunteer)")

plot(eff.cowles, 'neuroticism:extraversion', ylab="Prob(Volunteer)",
    ticks=list(at=c(.1,.25,.5,.75,.9)))

plot(eff.cowles, 'neuroticism:extraversion', multiline=TRUE,
    ylab="Prob(Volunteer)")

plot(effect('sex:neuroticism:extraversion', mod.cowles,
    xlevels=list(neuroticism=0:24, extraversion=seq(0, 24, 6))), multiline=TRUE)

mod.beps <- multinom(vote ~ age + gender + economic.cond.national +
economic.cond.household + Blair + Hague + Kennedy +
Europe*political.knowledge, data=BEPS)
plot(effect("Europe*political.knowledge", mod.beps,
xlevels=list(Europe=1:11, political.knowledge=0:3)))

plot(effect("Europe*political.knowledge", mod.beps,
```

```
xlevels=list(Europe=1:11, political.knowledge=0:3),
given.values=c(gendermale=0.5)),
style="stacked", colors=c("blue", "red", "orange"), rug=FALSE)

mod.wvs <- polr(poverty ~ gender + religion + degree + country*poly(age,3),
data=WVS)
plot(effect("country*poly(age, 3)", mod.wvs))

plot(effect("country*poly(age, 3)", mod.wvs), style="stacked")

plot(effect("country*poly(age, 3)", latent=TRUE, mod.wvs))

mod.pres <- lm(prestige ~ log(income, 10) + poly(education, 3) + poly(women, 2),
    data=Prestige)
eff.pres <- allEffects(mod.pres, default.levels=50)
plot(eff.pres, ask=FALSE)

mod.hart <- gls(fconvict ~ mconvict + tfr + partic + degrees, data=Hartnagel,
    correlation=corARMA(p=2, q=0), method="ML")
plot(allEffects(mod.hart), ask=FALSE)

## Not run:
data(cake, package="lme4")
require(lme4)
fm1 <- lmer(angle ~ recipe * temperature + (1|recipe:replicate), cake,
                   REML = FALSE)
plot(effect("recipe:temperature", fm1), grid=TRUE)
# lme4 and nlme conflict, so detach lme4
detach(package:lme4)
library(nlme)
cake$rep <- with(cake, paste( as.character(recipe), as.character(replicate), sep=""))
fm2 <- lme(angle ~ recipe * temperature, data=cake,
                   random = ~ 1 | rep, method="ML")
plot(effect("recipe:temperature", fm2), grid=TRUE)

## End(Not run)
```

`effects-deprecated` *Deprecated Function in effects Package*

Description

The `all.effects` function is provided only for compatibility with older versions of the effects package and may be removed; use `allEffects` instead.

Usage

```
all.effects(...)
```

Arguments

...	arguments to be passed to `allEffects`.

Author(s)

John Fox <jfox@mcmaster.ca>.

See Also

`allEffects`

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.

Source

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

References

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

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

`Prestige` *Prestige of Canadian Occupations*

Description

The `Prestige` data frame has 102 rows and 6 columns. The observations are occupations.

Usage

```
Prestige
```

Format

This data frame contains the following columns:

education Average education (years) of occupational incumbents, in 1971.

income Average income (dollars) of incumbents, 1971.

women Percentage of incumbents who are women, 1971.

prestige Pineo-Porter prestige score for occupation, from a social survey conducted in the mid-1960s.

census Canadian Census occupational code.

type Type of occupation. A factor with levels (note: out of order): `bc`, Blue Collar; `prof`, Professional, Managerial, and Technical; `wc`, White Collar.

Source

Canada (1971) *Census of Canada*. Vol. 3, Part 6. Statistics Canada [pp. 19-1–19-21].

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

References

Fox, J. (1997) *Applied Regression, Linear Models, and Related Methods.* Sage.

summary.eff *Summarizing, Printing, and Plotting Effects*

Description

summary, print, plot, and [methods for eff, effpoly, and efflist objects.

Usage

```
## S3 method for class 'eff'
print(x, type=c("response", "link"), ...)
## S3 method for class 'effpoly'
print(x, type=c("probability", "logits"), ...)
## S3 method for class 'efflatent'
print(x, ...)
## S3 method for class 'efflist'
print(x, ...)
## S3 method for class 'summary.eff'
print(x, ...)
## S3 method for class 'eff'
summary(object, type=c("response", "link"), ...)
## S3 method for class 'effpoly'
summary(object, type=c("probability", "logits"), ...)
## S3 method for class 'efflatent'
summary(object, ...)
## S3 method for class 'efflist'
summary(object, ...)
## S3 method for class 'eff'
plot(x, x.var = which.max(levels), z.var = which.min(levels),
    multiline = is.null(x$se), rug = TRUE, xlab, ylab, main = paste(effect,
        "effect plot"), colors = palette(), symbols = 1:10, lines = 1:10,
    cex = 1.5, ylim, factor.names = TRUE, type = c("response",
        "link"), ticks = list(at = NULL, n = 5), alternating = TRUE,
    rotx = 0, roty = 0, grid = FALSE, layout, rescale.axis = TRUE,
    key.args = NULL, row = 1, col = 1, nrow = 1, ncol = 1, more = FALSE,
    ...)
## S3 method for class 'effpoly'
plot(x, type = c("probability", "logit"), x.var = which.max(levels),
    rug = TRUE, xlab, ylab = paste(x$response, " (", type, ")",
        sep = ""), main = paste(effect, "effect plot"), colors,
    symbols = 1:10, lines = 1:10, cex = 1.5, factor.names = TRUE,
    style = c("lines", "stacked"), confint = (style == "lines" &&
        !is.null(x$confidence.level)), ylim, rotx = 0, alternating = TRUE,
    roty = 0, grid = FALSE, layout, key.args = NULL, row = 1,
    col = 1, nrow = 1, ncol = 1, more = FALSE, ...)
## S3 method for class 'efflist'
plot(x, selection, rows, cols, ask=TRUE, graphics=TRUE, ...)
```

```
## S3 method for class 'efflist'
x[...]
```

Arguments

x	an object of class "eff", "effpoly", "efflist", or "summary.eff", as appropriate.
object	an object of class "eff", "effpoly", or "efflist", as appropriate.
type	for linear and generalized linear models, if "response" (the default), effects are printed or the vertical axis is labelled on the scale of the response variable; if "link", effects are printed or the vertical axis labelled on the scale of the linear predictor. For polytomous logit models, this argument takes either "probability" or "logit", with the former as the default.
x.var	the index (number) or quoted name of the covariate or factor to place on the horizontal axis of each panel of the effect plot. The default is the predictor with the largest number of levels or values.
z.var	for linear, generalized linear or mixed models, the index (number) or quoted name of the covariate or factor for which individual lines are to be drawn in each panel of the effect plot. The default is the predictor with the smallest number of levels or values. This argument is only used if multiline = TRUE.
multiline	for linear, generalized linear or mixed models, if TRUE, each panel of the display represents combinations of values of two predictors, with one predictor (corresponding to x.var) on the horzontal axis, and the other (corresponding to z.var) used to define lines in the graph; defaults to TRUE if there are no standard errors in the object being plotted, and FALSE otherwise.
confint	plot point-wise confidence bands around fitted effects (for multinomial and proportional-odds logit models); defaults to TRUE, in which case separate panels are used for different response levels.
rug	if TRUE, the default, a rug plot is shown giving the marginal distribution of the predictor on the horizontal axis, if this predictor is a covariate.
xlab	the label for the horizontal axis of the effect plot; if missing, the function will use the name of the predictor on the horizontal axis.
ylab	the label for the vertical axis of the effect plot; the default is constructed from the name of the response variable for the model from which the effect was computed.
main	the title for the plot, printed at the top; the default title is constructed from the name of the effect.
colors	colors[1] is used to plot effects, colors[2] to plot confidence bands. In a mulitline plot, the successive colors correspond to the levels of the z.var covariate or factor. In a stacked plot or a plot without confidence bands for a multinomial or proportional-odds logit model, the successive colors correspond to the levels of the response factor. In all but stacked plots, colors defaults to palette(); for stacked multinomial-logit plots, colors defaults to rainbow_hcl(levels), where levels is the number of levels of the response variable; for stacked proportional-odds model plots, colors defautls to

`sequential_hcl(levels)`. **Warning:** This argument *cannot* be abbreviated to `col`, which is used for a different purpose (see below).

`symbols, lines` corresponding to the levels of the `z.var` covariate or factor on a multiline plot, or to the successive levels of the response factor in a line plot for a polytomous logit model. These arguments are used only if `multiline = TRUE` or for polytomous logit models where the effects are plotted without confidence bands; in these cases a legend is drawn at the top of the display.

`cex` character expansion for plotted symbols; default is `1.5`.

`ylim` 2-element vector containing the lower and upper limits of the vertical axes; if `NULL`, the default, then the vertical axes are scaled from the data.

`factor.names` a logical value, default `TRUE`, that controls the inclusion of factor names in conditioning-variable labels.

`style` (for multinomial or proportional-odds logit models) `"lines"` (the default for a line plot, or `"stacked"` for a stacked-bar or stacked-area plot. In the latter case only fitted probabilities may be plotted and confidence envelopes cannot be shown.

`ticks` a two-item list controlling the placement of tick marks on the vertical axis, with elements `at` and `n`. If `at=NULL` (the default), the program attempts to find 'nice' locations for the ticks, and the value of `n` (default, `5`) gives the approximate number of tick marks desired; if `at` is non-`NULL`, then the value of `n` is ignored.

`alternating` if `TRUE` (the default), the tick labels alternate by panels in multi-panel displays from left to right and top to bottom; if `FALSE`, tick labels appear at the bottom and on the left.

`rotx, roty` rotation angles for the horizontal and vertical tick marks, respectively. Default is 0.

`grid` if `TRUE`, add grid lines to the plot. Default is FALSE.

`layout` the `layout` argument to the **lattice** function `xyplot` (or, in some cases `densityplot`), which is used to draw the effect display; if not specified, the plot will be formatted so that it appears on a single page.

`rescale.axis` if `TRUE` (the default), the tick marks on the vertical axis are labelled on the response scale (e.g., the probability scale for effects computed on the logit scale for a binomial GLM).

`key.args` additional arguments to be passed to the `key` trellis argument to `xyplot` or `densityplot`, e.g., to position the key (legend) in the plotting region.

`row, col, nrow, ncol, more`
These arguments are used to graph an effect as part of an array of plots; `row`, `col`, `nrow`, and `ncol` are used to compose the `split` argument and `more` the `more` argument to `print.trellis`. Normally these arguments are not set by the user, but by `plot.efflist`. **Warning:** Note that `col` is *not* used to specify colors; use `colors` instead (see above).

`selection` the optional index (number) or quoted name of the effect in an effect list to be plotted; if not supplied, a menu of high-order terms is presented or all effects are plotted.

rows, cols	Number of rows and columns in the "meta-array" of plots produced for an efflist object; if either argument is missing, then the meta-layout will be computed by the plot method.
ask	if selection is not supplied and ask is TRUE (the default), a menu of high-order terms is presented; if ask is FALSE, effects for all high-order terms are plotted in an array.
graphics	if TRUE (the default), then the menu of terms to plot is presented in a dialog box rather than as a text menu.
...	arguments to be passed down.

Details

In a generalized linear model, by default, the print and summary methods for eff objects print the computed effects on the scale of the response variable using the inverse of the link function. In a logit model, for example, this means that the effects are expressed on the probability scale.

By default, effects in a GLM are plotted on the scale of the linear predictor, but the vertical axis is labelled on the response scale. This preserves the linear structure of the model while permitting interpretation on what is usually a more familiar scale. This approach may also be used with linear models, for example to display effects on the scale of the response even if the data are analyzed on a transformed scale, such as log or square-root.

In a polytomous (multinomial or proportional-odds) logit model, by default effects are plotted on the probability scale; they may be alternatively plotted on the scale of the individual-level logits.

Value

The summary method for "eff" objects returns a "summary.eff" object with the following components (those pertaining to confidence limits need not be present):

header	a character string to label the effect.
effect	an array containing the estimated effect.
lower.header	a character string to label the lower confidence limits.
lower	an array containing the lower confidence limits.
upper.header	a character string to label the upper confidence limits.
upper	an array containing the upper confidence limits.

The [method for "efflist" objects is used to subset an "efflist" object and returns an object of the same class.

Author(s)

John Fox <jfox@mcmaster.ca> and Jangman Hong.

See Also

effect, allEffects, xyplot, densityplot, print.trellis rainbow_hcl, sequential_hcl

Examples

```
mod.cowles <- glm(volunteer ~ sex + neuroticism*extraversion,
    data=Cowles, family=binomial)
eff.cowles <- allEffects(mod.cowles, xlevels=list(neuroticism=0:24,
    extraversion=seq(0, 24, 6)))
eff.cowles

plot(eff.cowles, 'sex', ylab="Prob(Volunteer)", grid=TRUE, rotx=90)

plot(eff.cowles, 'neuroticism:extraversion', ylab="Prob(Volunteer)",
    ticks=list(at=c(.1,.25,.5,.75,.9)))

plot(eff.cowles, 'neuroticism:extraversion', multiline=TRUE,
    ylab="Prob(Volunteer)", key.args = list(x = 0.75, y = 0.75, corner = c(0, 0)))

plot(effect('sex:neuroticism:extraversion', mod.cowles,
    xlevels=list(neuroticism=0:24, extraversion=seq(0, 24, 6))), multiline=TRUE)

mod.beps <- multinom(vote ~ age + gender + economic.cond.national +
economic.cond.household + Blair + Hague + Kennedy +
Europe*political.knowledge, data=BEPS)
plot(effect("Europe*political.knowledge", mod.beps,
xlevels=list(Europe=1:11, political.knowledge=0:3)))

plot(effect("Europe*political.knowledge", mod.beps,
xlevels=list(Europe=1:11, political.knowledge=0:3),
given.values=c(gendermale=0.5)),
style="stacked", colors=c("blue", "red", "orange"), rug=FALSE)

mod.wvs <- polr(poverty ~ gender + religion + degree + country*poly(age,3),
data=WVS)
plot(effect("country*poly(age, 3)", mod.wvs))

plot(effect("country*poly(age, 3)", mod.wvs), style="stacked",
colors=c("gray75", "gray50", "gray25"))

plot(effect("country*poly(age, 3)", latent=TRUE, mod.wvs))

mod.pres <- lm(prestige ~ log(income, 10) + poly(education, 3) + poly(women, 2),
    data=Prestige)
eff.pres <- allEffects(mod.pres, default.levels=50)
plot(eff.pres, ask=FALSE)
plot(eff.pres[1:2], ask=FALSE)
```

Titanic *Survival of Passengers on the Titanic*

Description

Information on the survival status, sex, age, and passenger class of 1309 passengers in the Titanic disaster of 1912.

Usage

```
Titanic
```

Format

A data frame with 1309 observations on the following 4 variables.

`survived` `no` or `yes`.

`sex` `female` or `male`

`age` in years (and for some children, fractions of a year); age is missing for 263 of the passengers.

`passengerClass` `1st`, `2nd`, or `3rd` class.

Details

This is part of a larger data set compiled by Thomas Cason. Many additional details are given in the sources cited below.

Source

Data set `titanic3` from http://biostat.mc.vanderbilt.edu/twiki/bin/view/Main/DataSets.

References

http://www.encyclopedia-titanica.org/

F. E. Harrell, Jr. (2001) *Regression Modeling Strategies* New York: Springer.

Examples

```
summary(Titanic)

titanic <- glm(survived ~ (passengerClass + sex + age)^2, data=Titanic, family=binomial)

titanic.all <- allEffects(titanic, typical=median,
given.values=c(passengerClass2nd=1/3, passengerClass3rd=1/3, sexmale=0.5))

plot(titanic.all, ticks=list(at=c(.01, .05, seq(.1, .9, by=.2), .95, .99)), ask=FALSE)

plot(effect("passengerClass*sex*age", titanic, xlevels=list(age=0:65)),
ticks=list(at=c(.001, .005, .01, .05, seq(.1, .9, by=.2), .95, .99, .995)))
```

`Wells` *Well Switching in Bangladesh*

Description

Data on whether or not households in Bangladesh changed the wells that they were using.

Usage

```
Wells
```

Format

A data frame with 3020 observations on the following 5 variables.

`switch` whether or not the household switched to another well from an unsafe well: `no` or `yes`.

`arsenic` the level of arsenic contamination in the household's original well, in hundreds of micrograms per liter; all are above 0.5, which was the level identified as "safe".

`distance` in meters to the closest known safe well.

`education` in years of the head of the household.

`association` whether or not any members of the household participated in any community organizations: `no` or `yes`.

Details

The data are for an area of Arahazar upazila, Bangladesh. The researchers labelled each well with its level of arsenic and an indication of whether the well was "safe" or "unsafe." Those using unsafe wells were encouraged to switch. After several years, it was determined whether each household using an unsafe well had changed its well. These data are used by Gelman and Hill (2007) for a logistic-regression example.

Source

http://www.stat.columbia.edu/~gelman/arm/examples/arsenic/wells.dat.

References

A. Gelman and J. Hill (2007) *Data Analysis Using Regression and Multilevel/Hierarchical Models.* Cambridge: Cambridge University Press.

Examples

```
summary(Wells)
```

WVS *World Values Surveys*

Description

Data from the World Values Surveys 1995-1997 for Australia, Norway, Sweden, and the United States.

Usage

```
WVS
```

Format

A data frame with 5381 observations on the following 6 variables.

`poverty` "Do you think that what the government is doing for people in poverty in this country is about the right amount, too much, or too little?" (ordered): `Too Little`, `About Right`, `Too Much`.

`religion` Member of a religion: `no` or `yes`.

`degree` Held a university degree: `no` or `yes`.

`country` `Australia`, `Norway`, `Sweden`, or `USA`.

`age` in years.

`gender` `male` or `female`.

References

J. Fox and R. Andersen (2006) Effect displays for multinomial and proportional-odds logit models. *Sociological Methodology* **36**, 225–255.

Examples

```
summary(WVS)

require(splines)  # for bs()
wvs <- polr(poverty ~ gender + country*(religion + degree + bs(age, 4)), data=WVS)

plot(effect("country*bs(age,4)", wvs, xlevels=list(age=18:83),
  given.values=c(gendermale=0.5)), rug=FALSE)

plot(effect("country*bs(age,4)", wvs, xlevels=list(age=18:83),
  given.values=c(gendermale=0.5)), rug=FALSE, style="stacked")

plot(effect("country*bs(age,4)", wvs, xlevels=list(age=18:83),
  given.values=c(gendermale=0.5), latent=TRUE), rug=FALSE)
```

Index