Effect Displays in R for Generalised Linear Models

John Fox[*]

McMaster University

Hamilton, Ontario, Canada

Abstract

This paper describes the implementation in R of a method for tabular or graphical display of terms in a complex generalised linear model. By complex, I mean a model that contains terms related by marginality or hierarchy, such as polynomial terms, or main effects and interactions. I call these tables or graphs *effect displays*. Effect displays are constructed by identifying high-order terms in a generalised linear model. Fitted values under the model are computed for each such term. The lower-order 'relatives' of a high-order term (e.g., main effects marginal to an interaction) are absorbed into the term, allowing the predictors appearing in the high-order term to range over their values. The values of other predictors are fixed at typical values: for example, a covariate could be fixed at its mean or median, a factor at its proportional distribution in the data, or to equal proportions in its several levels. Variations of effect displays are also described, including representation of terms higher-order to any appearing in the model.

1   Background and Motivation

This paper describes the implementation in R of a method for tabular or graphical display of terms in a complex generalised linear model. By complex, I mean a model that contains terms related by marginality or hierarchy (in the sense of Nelder, 1977), such as polynomial terms, or main effects and interactions. I call these tables or graphs *effect displays*.

I assume that readers are familiar with R, which is a free, open-source implementation of the S statistical programming language and computing environment. Extensive information on R is available at <http://www.r-project.org/>. The software described in the paper is distributed as the `effects` package for R, available on the Comprehensive R Archive Network (CRAN) at <http://cran.r-project.org/>.

The general approach implemented in the `effects` package is a modest extension of methods introduced by Fox (1987). Similar, if less general, ideas have a long history, dating at least to Fisher's (1936) "adjusted means" in the analysis of covariance. Goodnight and Harvey's (1978) "least-squares means" in analysis of variance and covariance, and Searle, Speed, and Milliken's (1980) "estimated population marginal means" are other examples, though all of these approaches are restricted to interactions among factors (i.e., categorical predictors) in a linear model. Recently King, Tomz, and Wittenberg (2000) and Tomz, Wittenberg, and King (2003) have presented similar ideas, but their approach, based on Monte-Carlo simulation of a model, is much more complicated than the one discussed here.

To motivate effect displays, consider the following data on police treatment of individuals arrested in Toronto for simple possession of small quantities of marijuana. (The data discussed here are part of a larger data set featured in a series of articles in the *Toronto Star* newspaper.) Under these circumstances police have the option of releasing an arrestee with a summons to appear in court — similar to a traffic ticket; alternatively, the individual may be brought to the police station for questioning and possible indictment. The principal question of interest is whether and how the probability of release is influenced by the subject's race, age, and other characteristics. The data are in the data frame `Arrests` in the `effects` package:

```
> library(effects)
> data(Arrests)
> dim(Arrests)
[1] 5226    8
> Arrests[sort(sample(5226, 10)),]  # sample 10 of 5226 observations
     released colour year age  sex employed citizen checks
275       Yes  White 1999  18 Male      Yes     Yes      3
596       Yes  White 2000  20 Male      Yes     Yes      3
676       Yes  White 1998  14 Male       No     Yes      0
2046      Yes  White 1999  23 Male      Yes     Yes      0
2236      Yes  White 2002  20 Male      Yes     Yes      2
3905      Yes  White 2001  19 Male      Yes     Yes      3
3910      Yes  White 1997  32 Male       No      No      2
4506      Yes  White 1997  32 Male       No      No      3
4730      Yes  White 1997  21 Male      Yes      No      0
5213       No  Black 2001  27 Male      Yes     Yes      3
>
```

Most of the variables in the data set are self explanatory, with the following exceptions:

- `colour` is either `Black` or `White`. The original data included the additional categories "Brown" and "Other," but their meaning is ambiguous and their use relatively infrequent.

- The observations span the years 1997 through (part of) 2002. A few arrests in 1996 were eliminated. In the analysis reported below, `year` is treated as a factor (i.e., as a categorical predictor).

- When suspects are stopped by the police, their names are checked in six data bases — of previous arrests, previous convictions, parole status, and so on. The variable `checks` records the number of data bases on which an individual's name appeared.

Preliminary analysis of the data suggested a logit model including interactions between `colour` and `year` and between `colour` and `age`, and main effects of `employed`, `citizen`, and `checks`. The effects of `age` and `checks` appear to be reasonably linear on the logit scale and are modelled as such:

```
> Arrests$year <- as.factor(Arrests$year)
> arrests.mod <- glm(released ~ employed + citizen + checks
+       + colour*year + colour*age,
+      family=binomial, data=Arrests)
> summary(arrests.mod)

Call:
glm(formula = released ~ employed + citizen + checks + colour *
    year + colour * age, family = binomial, data = Arrests)

Deviance Residuals:
    Min      1Q   Median      3Q      Max
-2.4787   0.3241   0.4485   0.6262   1.7132

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        0.344433   0.310043   1.111 0.266602
employedYes        0.735064   0.084760   8.672  < 2e-16 ***
citizenYes         0.585984   0.113757   5.151 2.59e-07 ***
checks            -0.366642   0.026026 -14.088  < 2e-16 ***
colourWhite        1.212516   0.349734   3.467 0.000526 ***
```

```
year1998                -0.431179   0.260337  -1.656 0.097674 .
year1999                -0.094434   0.261522  -0.361 0.718029
year2000                -0.010897   0.259184  -0.042 0.966463
year2001                 0.243063   0.262991   0.924 0.355370
year2002                 0.212955   0.353243   0.603 0.546604
age                      0.028728   0.008618   3.333 0.000858 ***
colourWhite:year1998     0.651956   0.313448   2.080 0.037530 *
colourWhite:year1999     0.155950   0.307010   0.508 0.611478
colourWhite:year2000     0.295753   0.306159   0.966 0.334039
colourWhite:year2001    -0.380541   0.304022  -1.252 0.210683
colourWhite:year2002    -0.617318   0.419210  -1.473 0.140866
colourWhite:age         -0.037373   0.010199  -3.664 0.000248 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1


(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 4776.3  on 5225  degrees of freedom
Residual deviance: 4257.1  on 5209  degrees of freedom
AIC: 4291.1

Number of Fisher Scoring iterations: 4




> library(car)  # for the Anova function
> Anova(arrests.mod)
Anova Table (Type II tests)

Response: released
           LR Chisq Df Pr(>Chisq)
employed     72.673  1  < 2.2e-16 ***
citizen      25.783  1  3.820e-07 ***
checks      205.211  1  < 2.2e-16 ***
colour       19.572  1  9.687e-06 ***
year          6.087  5  0.2978477
age           0.459  1  0.4982736
colour:year  21.720  5  0.0005917 ***
colour:age   13.886  1  0.0001942 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

It is relatively difficult (but not impossible) to discern from inspection of the coefficients how `colour`, `year`, and `age` combine to influence the probability of release. (In contrast, the effects of `employed`, `citizen`, and `checks` are quite clear from the coefficients, although some mental arithmetic is required unless one is comfortable with interpreting effects on the logit scale.) The model employs the default 0/1 dummy coding for factors that R terms "treatment" contrasts (but the coding employed is immaterial to the effect displays described below).

2   Effect Displays

A general principle of interpretation for statistical models containing terms that are marginal to others is that high-order terms should be combined with their lower-order relatives — for example, an interaction between two factors should be combined with the main effects marginal to the interaction. In conformity with this principle, Fox (1987) suggests identifying the high-order terms in a generalised linear model. Fitted values under the model are computed for each such term. The lower-order 'relatives' of a high-order term (e.g., main effects marginal to an interaction) are absorbed into the term, allowing the predictors appearing in the high-order term to range over their values. The values of other predictors are fixed at typical values: for example, a covariate could be fixed at its mean or median, a factor at its proportional distribution in the data, or to equal proportions in its several levels. This procedure is illustrated below.

Consider the generalised linear model with linear predictor $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}$ and link function $g(\boldsymbol{\mu}) = \boldsymbol{\eta}$, where $\boldsymbol{\mu}$ is the expectation of the response vector $\mathbf{y}$. Suppose that we have an estimate $\mathbf{b}$ of $\boldsymbol{\beta}$, along with the estimated covariance matrix $\widehat{V}(\mathbf{b})$ of $\mathbf{b}$. Let the rows of $\mathbf{X}^*$ include all combinations of values of predictors appearing in a high-order term, along with typical values of the remaining predictors. The structure of $\mathbf{X}^*$ with respect, for example, to interactions, is the same as that of the model matrix $\mathbf{X}$. Then the fitted values $\widehat{\boldsymbol{\eta}}^* = \mathbf{X}^*\mathbf{b}$ represent the effect in question, and a table or graph of these values — or, alternatively, of the fitted values transformed to the scale of the response, $g^{-1}(\widehat{\boldsymbol{\eta}}^*)$ — is an effect display. The standard errors of $\widehat{\boldsymbol{\eta}}^*$, available as the square-root diagonal entries of $\mathbf{X}^*\widehat{V}(\mathbf{b})\mathbf{X}^{*\prime}$, may be used to compute pointwise confidence intervals for the effects, the end-points of which may then also be transformed to the scale of the response.

For example, `colour:age` is a high-order term of the logit model fit to the `Arrests` data in the preceding section. The lower-order relatives of this term are the constant, and the terms for `colour` and `age`. The arrestees in the data set range in age from 12 to 66, with the .025 and .975 quantiles at 15 and 45, respectively. As mentioned, there are two levels of `colour`, so if we set `age` to all integer values between 15 and 45, there are 62 combinations of values of these two predictors. Other predictors in the model (`employed`, `citizen`, `checks`, and `year`) will be set to typical values, as previously explained.

The matrix $\mathbf{X}^*$ then appears as follows (with columns in the same order as the coefficients in the R output above, and labelled by coefficient):

$$
\begin{bmatrix}
(b_1) & (b_2) & (b_3) & (b_4) & (b_5) & (b_6) & (b_7) & (b_8) & (b_9) & (b_{10}) & (b_{11}) & (b_{12}) & (b_{13}) & (b_{14}) & (b_{15}) & (b_{16}) & (b_{17}) \\
1 & 0.79 & 0.85 & 1.64 & 0 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 15 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0.79 & 0.85 & 1.64 & 1 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 15 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 15 \\
1 & 0.79 & 0.85 & 1.64 & 0 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 16 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0.79 & 0.85 & 1.64 & 1 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 16 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 16 \\
1 & 0.79 & 0.85 & 1.64 & 0 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 17 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0.79 & 0.85 & 1.64 & 1 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 17 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 17 \\
1 & 0.79 & 0.85 & 1.64 & 0 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 18 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 0.79 & 0.85 & 1.64 & 1 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 18 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 18 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
1 & 0.79 & 0.85 & 1.64 & 1 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 65 & 0.17 & 0.21 & 0.24 & 0.23 & 0.05 & 65
\end{bmatrix}
$$

In particular:

- Column 1 of $\mathbf{X}^*$ represents the constant.

- Column 2 reflects the 79 percent of arrestees who were at level `Yes` of `employed`, and hence had values of 1 on the treatment-coded contrast for this factor; 0.79 is therefore also the mean of the contrast. Note that this column, along with other constant columns in $\mathbf{X}^*$, is in effect absorbed in the constant term, and therefore influences only the average level of the computed effects (on the scale of the linear predictor).

- Column 3 reflects the 85 percent of arrestees who were in level `Yes` of `citizen`.

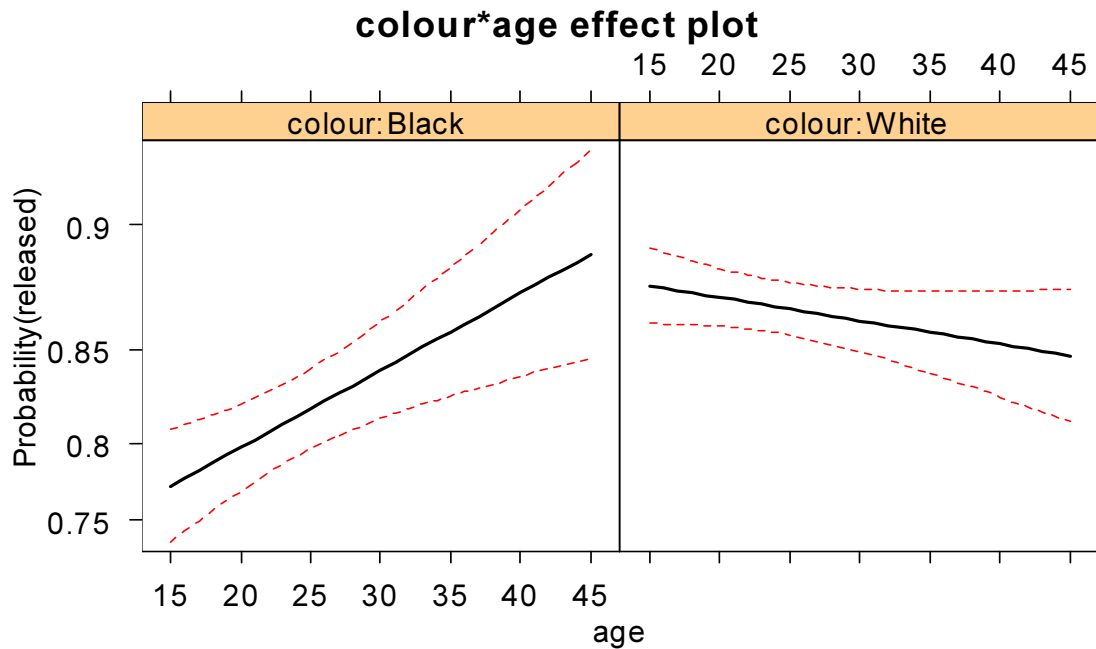- Column 4 reflects the average value of `checks`, 1.64.

Figure 1: Effect display for the interaction of `colour` and `age` in the logit model fit to the `Arrests` data. The vertical axis is labelled on the probability scale, and a 95-percent pointwise confidence interval is drawn around the estimated effect.

- Column 5 repeats the two values 0 and 1 for the contrast for `colour` (to be taken in combination with the values of `age` in column 11).

- Columns 6 through 10 represent the contrasts for `year`, and contain the proportions of arrestees in years 1998 through 2002; this reflects the use of the first level of year, 1997, as the baseline level.

- Column 11 contains the twice-repeated integer values of `age`, from 15 through 65.

- Columns 12 through 16 are for the interaction of `colour` with `year` (which is absorbed in the `colour` term).

- Column 17 is for the `colour` by `age` interaction.

Although it is not difficult to construct $\mathbf{X}^*$ directly in this manner, it is tedious to do so. A graphical effect display for the `colour` by `age` interaction, computed by the software described in Section 3, appears in Figure 1: Apparently age has quite a different relationship to the probability of release for blacks and whites: older blacks are *more* likely to be released than younger blacks, while older whites are *less* likely to be released than younger whites; the relationship between age and the probability of release is also steeper for blacks than for whites.

Notice from the unequal spacing of the tick marks on the vertical axis of Figure 1 that although the axis is labelled on the scale of the response (i.e., the probability scale), the effects are plotted on the scale of the linear predictor (the logit scale); consequently, the lines plotted on the display are straight. I return to this point below.

3   Effect-Display Software

The software in the `effects` package consists of several related functions, described in more specific detail in the help pages for the package given as an appendix to this paper:

- The function `effect` returns an object of class `effect`, containing information for constructing an effect display. The essential input to `effect` includes a linear (`lm`) or generalised-linear (`glm`) model object, and a term for which the effect is to be computed. This term will usually be a high-order term of the model, but it is possible to compute effects for lower-order terms (averaging over their higher-order relatives), and for terms that are not in the model but that have relatives that are in the model. For example, in the model fit to the `Arrests` data, the two way interactions `colour:year` and `colour:age` appear in the model, but not the three-way interaction `colour:year:age` (nor the two-way interaction `year:age`). Computing the `colour:year:age` effect combines the `colour:year` and `colour:age` interactions. In my experience, the result of combining lower-order interactions for overlapping predictors (here `colour` in `colour:year` and `colour:age`) is not always obvious, and such a display can therefore prove informative. For example, in Figure 2 the lines for each of blacks and whites are parallel across panels, reflecting the absence of a `year:age` interaction, but the relative heights of the lines vary, reflecting the `colour:year` interaction: In the earlier years, the line for blacks is mostly below that for whites, suggesting that (holding the other factors constant) at most ages whites were more likely than blacks to be released (particularly when we take into account the fact that most arrestees were young); in the last two years, however, the line for blacks is mostly above that for whites.

- Figure 2 is computed by the following command (note the warning message):

```
> plot(effect("colour:year:age", arrests.mod, xlevels=list(age=15:45)),
+     multiline=TRUE, ylab="Probability(released)", rug=FALSE)
Warning message:
colour:year:age does not appear in the model in:
  effect("colour:year:age", arrests.mod, xlevels = list(age = 15:45))
```

  - In this command, I have supplied an optional argument to the `effect` function: `xlevels=list(age=15:45))` specifies values for `age`, which otherwise would default to 10 equally spaced values across the full range of the predictor. As mentioned, the ages 15 through 45 encompass 95 percent of the observations, excluding therefore a small number of very young and very old individuals.

  - There are also optional arguments supplied to `plot`:

    - `multiline=TRUE` specifies that separate graphs should not be drawn for each level of `colour`. By default, confidence envelopes are suppressed in a multi-line effect display.

    - `ylab="Probability(released)"` gives a non-default label for the vertical axis (the default would simply be the name of the response, `released`).

    - `rug=FALSE` suppresses a "rug plot" (one-dimensional scatterplot) for `age`, which would otherwise appear on the horizontal axis of the graph. In some instances — but not here, where the data set is quite large — displaying the distribution of the variable on the horizontal axis in this manner can give the viewer a rough sense of where the data are located.

  - The function `all.effects` takes a linear or generalised linear model object as its required argument, finds all high-order terms in the model, and returns a list of effects corresponding to these terms.

  - There are `print`, `summary`, and `plot` methods both for `effect` objects and for `effect.list` objects (returned by `effect` and `all.effects`, respectively). Effect plots are created using Trellis graphics (via the `lattice` package in R). The `plot` method for `effect.list` objects presents a text menu from which the user can select effects to graph. For example, to produce Figure 1:
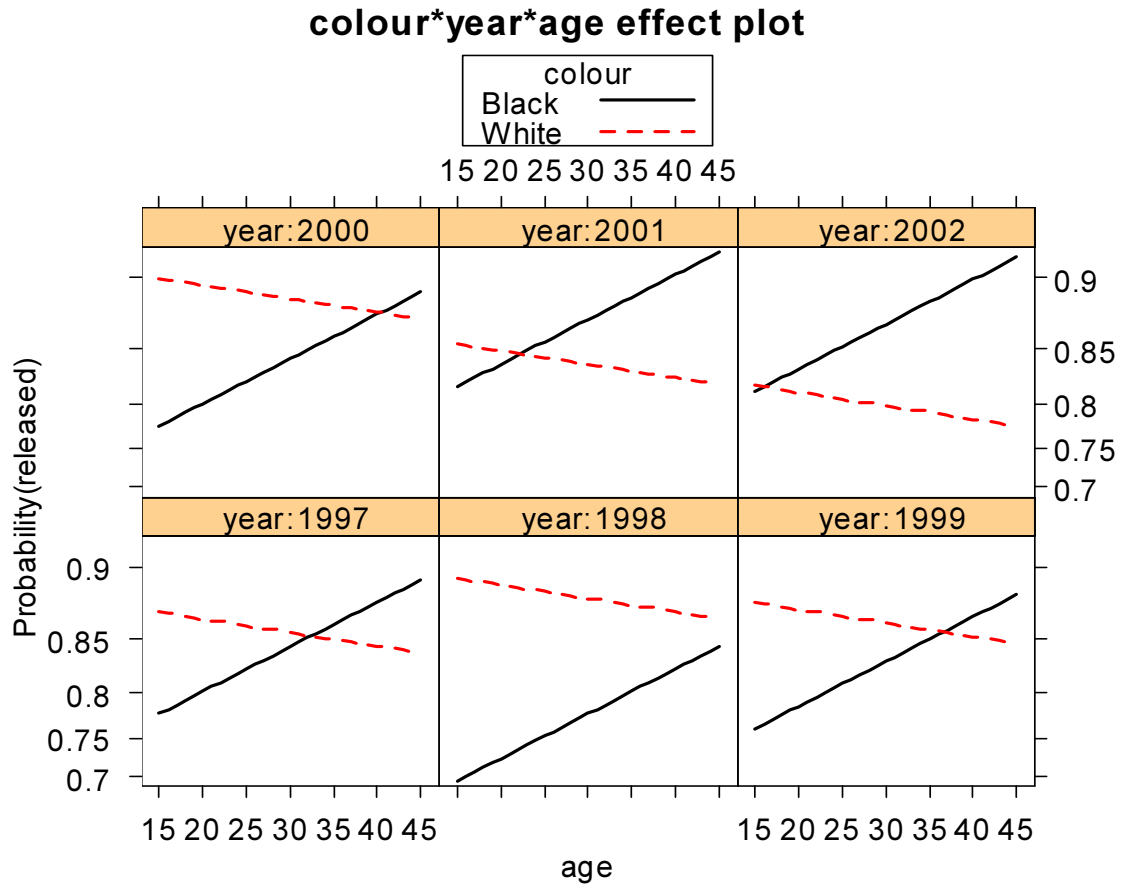
Figure 2: A graph of the `colour` by `year` by `age` effect, which corresponds to a term not in the model. The model includes `colour:year` and `colour:age` interactions, but not the `colour:year:age` interaction.

```
> arrests.effects <- all.effects(arrests.mod, xlevels=list(age=15:45))
> plot(arrests.effects, ylab="Probability(released)", rug=FALSE)

1:employed
2:citizen
3:checks
4:colour*year
5:colour*age
Selection: 5

1:employed
2:citizen
3:checks
4:colour*year
5:colour*age
Selection: 0
>
```

The `plot` method for `effect` objects allows one to graph effects on the response scale (rather than the scale of the linear predictor) or to transform the vertical axis of the plot in an arbitrary manner. Plotting effects on the response scale makes them more complex, however: Except in the case of the identity link, effects that are linear on the scale of the linear predictor are not linear on the scale of the response. Consequently, for example, profiles of effects are not parallel on the response scale even in the absence of interactions. As well, plotting on the scale of the linear predictor makes the choice of "typical" values for excluded effects less critical, in that these values only affect the labeling of the vertical axis (i.e., the level of the effects) and not the configuration of the display. Applied to the preceding example, plotting on the response (probability) scale produces Figure 3:

```
> plot(effect("colour:year:age", arrests.mod, xlevels=list(age=15:45)),
+     rescale.axis=FALSE, multiline=TRUE, ylab="Probability(released)",
+     rug=FALSE)
Warning message:
colour:year:age does not appear in the model in:
  effect("colour:year:age", arrests.mod, xlevels = list(age = 15:45))
```

3.1   Some Details

Many aspects of effect displays can be controlled through optional arguments to the `effect` function and to the `plot`, `print`, and `summary` methods for `effect` objects. Defaults are set in a manner intended to produce pleasing displays. For example, the predictor with the largest number of values (or levels) is plotted by default on the horizontal axis; similarly, in a multi-line display, the predictor with the smallest number of values is used to define the lines, different colours and line types are selected to represent the values of the predictor, and a legend is drawn at the top of the plot (as in Figures 2 and 3). All this can be modified via optional arguments.

Some thought should be given to the scale on which effects are displayed. The default behaviour is to compute and graph effects on the scale of the linear predictor but to label the vertical axis on the scale of the response. Arbitrary transformations are supported, however. For example, if a linear model is fit to the log of the response (e.g., log dollar income), then the exponential function could be used to transform ticks mark labels to the original scale (dollars). See the `transformation` argument to `effect`, and the `type` and `rescale.axis` arguments to the `plot` method for `effect` objects.

The `effect` objects contain a variety of information that can be used to construct custom displays, and `print` and `summary` methods for `effect` objects present tables of the results (note that an interaction may be specified either with asterisks or colons — they are treated as equivalent since lower-order terms are in any event absorbed):
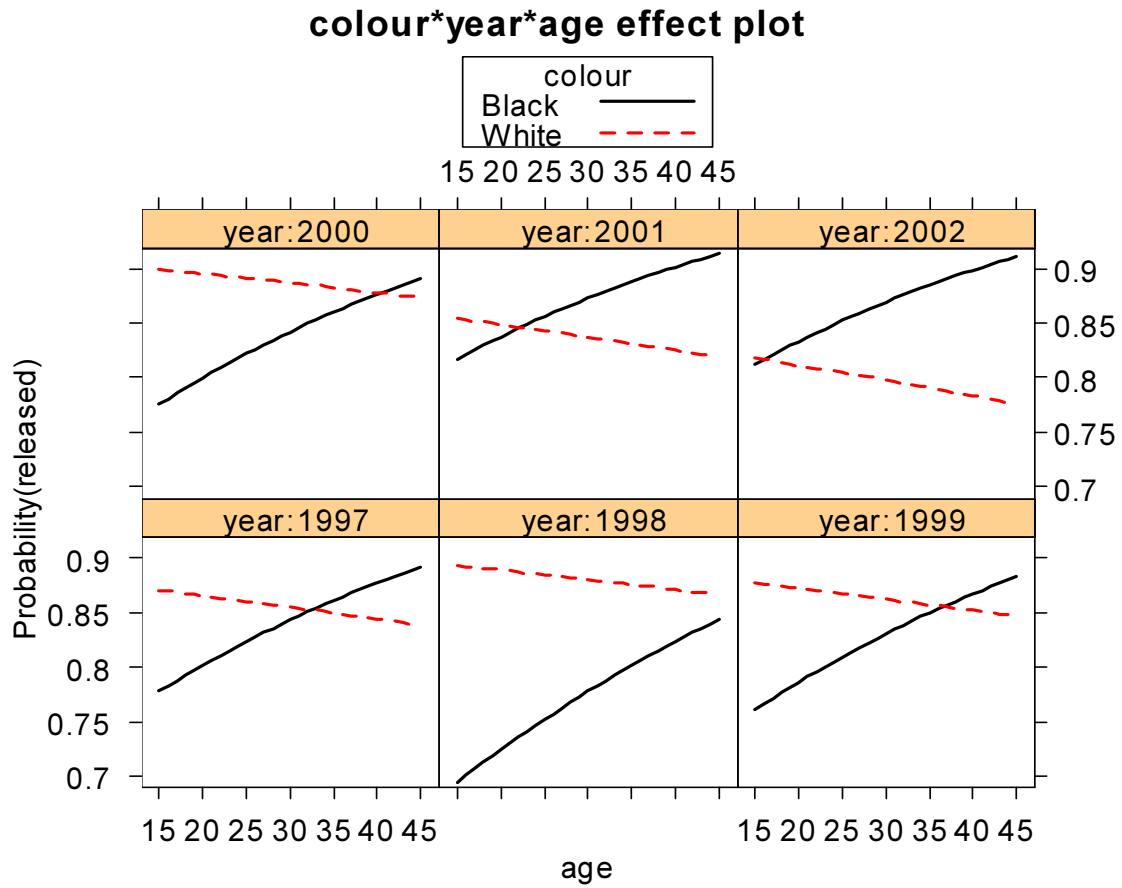
Figure 3: Plotting on the scale of the response (the probability scale) — compare to Figure 2. Note that the tick marks on the vertical axis are equally spaced; that the lines in the plot are not quite straight; and that corresponding lines are not quite parallel across panels.

```
> colour.year <- effect("colour*year", arrests.mod)
> colour.year

 colour*year effect
      year
colour       1997        1998        1999        2000        2001        2002
  Black 0.8186570 0.7457543 0.8042136 0.8170335 0.8519954 0.8481584
  White 0.8615922 0.8858824 0.8687663 0.8922020 0.8443687 0.8059984

> summary(colour.year)

 colour*year effect
      year
colour       1997        1998        1999        2000        2001        2002
  Black 0.8186570 0.7457543 0.8042136 0.8170335 0.8519954 0.8481584
  White 0.8615922 0.8858824 0.8687663 0.8922020 0.8443687 0.8059984

 Lower 95 Percent Confidence Limits
      year
colour       1997        1998        1999        2000        2001        2002
  Black 0.7453235 0.6868669 0.7527411 0.7689683 0.8104012 0.7629002
  White 0.8228821 0.8581418 0.8434657 0.8703516 0.8180900 0.7409131

 Upper 95 Percent Confidence Limits
      year
colour       1997        1998        1999        2000        2001        2002
  Black 0.8744318 0.796843 0.8471467 0.8569591 0.8857510 0.9065155
  White 0.8929427 0.908775 0.8905083 0.9107475 0.8674663 0.8578700
```

By default, the `print` and `summary` methods express effects on the scale of the response, but this behaviour may be modified via the argument `type`. In general, I find tabular displays of effects less satisfactory than graphical displays, but when the predictors are factors with relatively few levels, tabular displays may suffice.

To calculate fitted values for effect displays, the `effect` function implements the strategy for "safe prediction" described by Hastie (1992: Sec. 7.3.3; see also Venables and Ripley, 2002: Sec. 6.4). This is an issue for terms in a linear or generalised linear model with bases that depend upon the data, such as orthogonal polynomial regressors or regression splines. In such cases, naive generation of a new model matrix to be used along with the coefficients for the original fit produces incorrect results.

A simple example of effect plots incorporating regression splines, orthogonal-polynomial regressors, and a transformed predictor appears in Figure 4. The data for this example pertain to the rated prestige of 102 Canadian occupations. The `prestige` of the occupations is regressed on three predictors, all derived from the 1971 Census of Canada: the average `income` of occupational incumbents, in dollars (represented in the model as the log of `income`); the average `education` of occupational incumbents, in years (represented by a B-spline with three degrees of freedom); and the percentage of occupational incumbents who were `women` (represented by an orthogonal polynomial of degree two):
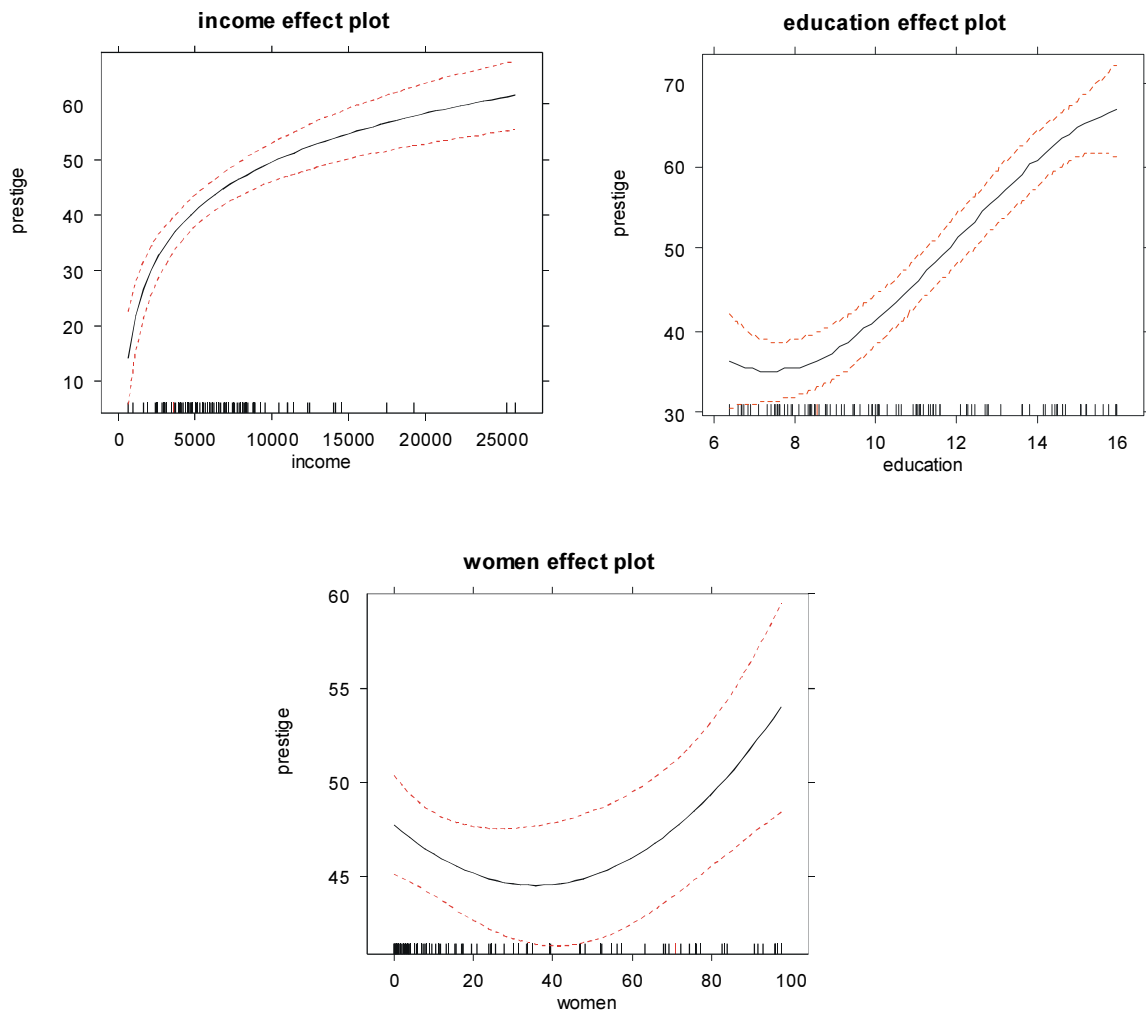
Figure 4: Effect plots for the predictors of `prestige` in the Canadian occupational prestige data. The model includes the log of `income`, a B-spline in `education`, and a quadratic in `women`.

```
> data(Prestige)
> library(splines) # for bs
> prestige.mod <- lm(prestige ~ log(income) + bs(education, df=3) + poly(women, 2),
+     data=Prestige)
> summary(prestige.mod)

Call:
lm(formula = prestige ~ log(income) + bs(education, df = 3) +
    poly(women, 2), data = Prestige)

Residuals:
     Min       1Q    Median       3Q      Max
-13.9603   -4.9834   0.3720    3.7870   17.0921

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)              -72.916     15.494  -4.706 8.58e-06 ***
log(income)               12.672      1.836   6.901 5.74e-10 ***
bs(education, df = 3)1     -8.197      7.822  -1.048  0.29735
bs(education, df = 3)2     25.660      5.497   4.668 9.97e-06 ***
bs(education, df = 3)3     30.418      4.585   6.634 2.00e-09 ***
poly(women, 2)1           11.978      9.384   1.276  0.20489
poly(women, 2)2           18.465      6.828   2.704  0.00811 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.721 on 95 degrees of freedom
Multiple R-Squared: 0.8564,     Adjusted R-squared: 0.8474
F-statistic: 94.46 on 6 and 95 DF,  p-value: < 2.2e-16

> Anova(prestige.mod)
Anova Table (Type II tests)

Response: prestige
                     Sum Sq Df F value     Pr(>F)
log(income)          2151.5  1 47.6249 5.744e-10 ***
bs(education, df = 3) 6067.0  3 44.7653 < 2.2e-16 ***
poly(women, 2)        400.2  2  4.4295   0.01448 *
Residuals            4291.8 95
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

> plot(all.effects(prestige.mod, default.levels=50))

1:log(income)
2:bs(education, df = 3)
3:poly(women, 2)
Selection: 1

1:log(income)
2:bs(education, df = 3)
3:poly(women, 2)
Selection: 2
```

```
1:log(income)
2:bs(education, df = 3)
3:poly(women, 2)
Selection: 3

1:log(income)
2:bs(education, df = 3)
3:poly(women, 2)
Selection: 0
>
```

When a model includes interactions between or among covariates, consideration should be given to the values at which the covariates are set in effect displays and to the form of the display. Consider the following example, taken from research by Cowles and Davis (1987) on volunteering for psychological experiments (discussed in greater detail in Fox, 1987). The response variable in the study is dichotomous: whether or not each of 1421 subjects volunteered to participate in an experiment. The authors modelled the data in a logistic regression of volunteering on the factor sex, the personality dimensions neuroticism and extraversion, and the product of neuroticism and extraversion. The two covariates each can take on integer values between 0 and 24.

```
> data(Cowles)
> cowles.mod <- glm(volunteer ~ sex + neuroticism*extraversion,
+      data=Cowles, family=binomial)
> summary(cowles.mod)

Call:
glm(formula = volunteer ~ sex + neuroticism * extraversion, family = binomial,
    data = Cowles)

Deviance Residuals:
    Min      1Q   Median      3Q      Max
-1.4749  -1.0602  -0.8934   1.2609   1.9978

Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
(Intercept)            -2.358200   0.501037  -4.707 2.52e-06 ***
sexmale                -0.247152   0.111613  -2.214  0.02680 *
neuroticism             0.110776   0.037633   2.944  0.00324 **
extraversion            0.166816   0.037700   4.425 9.65e-06 ***
neuroticism:extraversion -0.008552  0.002933  -2.916  0.00354 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1933.5  on 1420  degrees of freedom
Residual deviance: 1897.4  on 1416  degrees of freedom
AIC: 1907.4

Number of Fisher Scoring iterations: 3
```

```
> Anova(cowles.mod)
Anova Table (Type II tests)

Response: volunteer
                       LR Chisq Df Pr(>Chisq)
sex                      4.9184  1   0.026572 *
neuroticism              0.3139  1   0.575316
extraversion            22.1372  1  2.538e-06 ***
neuroticism:extraversion 8.6213  1   0.003323 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The default effect display for the neuroticism by extraversion interaction is shown in Figure 5:

```
> plot(effect("neuroticism*extraversion", cowles.mod))
```

The display is not bad, but the default of 10 values for the conditioning variable `extraversion` produces too many panels in the graph; this problem is compounded for higher-order interactions involving covariates. An alternative multi-line display, setting `extraversion` to six values, appears in Figure 6:

```
> plot(effect("neuroticism*extraversion", cowles.mod,
+     xlevels=list(neuroticism=0:24, extraversion=seq(0, 24, 6))),
+     multiline=TRUE, ylab="Probability(Volunteer)")
```

In instances such as this, better custom displays can often be produced by using the information contained in the `effect` object. An example, produced by the following code, is shown in Figure 7:

```
> ne.effect <- effect("neuroticism*extraversion", cowles.mod,
+     xlevels=list(neuroticism=seq(0, 24, 6), extraversion=0:24),
+     confidence.level=2/3)
> ne.sumry <- summary(ne.effect, type="link")  # for effects & confidence bounds
> fit   <- ne.sumry$effect  # fitted values for effect
> lower <- ne.sumry$lower   # lower confidence bounds
> upper <- ne.sumry$upper   # upper confidence bounds
> par(mar=c(5, 4, 4, 4)+.1) # leave some extra room
> plot(c(0, 30), range(c(lower, upper)), type="n", xaxt="n",
+     xlab="Extraversion", ylab="Logit of Volunteering")
> axis(1, at=seq(0, 24, 6))
> text(25, 1.8, "Neuroticism", adj=0)
> prob.axis() # right-side axis, from car package
> neuroticism <- seq(0, 24, 6)
> for (neuro in 1:5){  # plot effects
+     lines(0:24, fit[neuro,], lwd=2)
+     text(25, fit[neuro, 25], paste("N = ", neuroticism[neuro]), adj=0)
+     }
> extraversion <- 0:24
> extra <- seq(1, 25, by=6)
> for (neuro in c(1, 3, 5)){  # plot confidence bars
+         arrows(extraversion[extra], lower[neuro, extra],
+             extraversion[extra], upper[neuro, extra],
+             angle=90, code=3, lty=2, length=0.05, col="red")
+             }
>
```
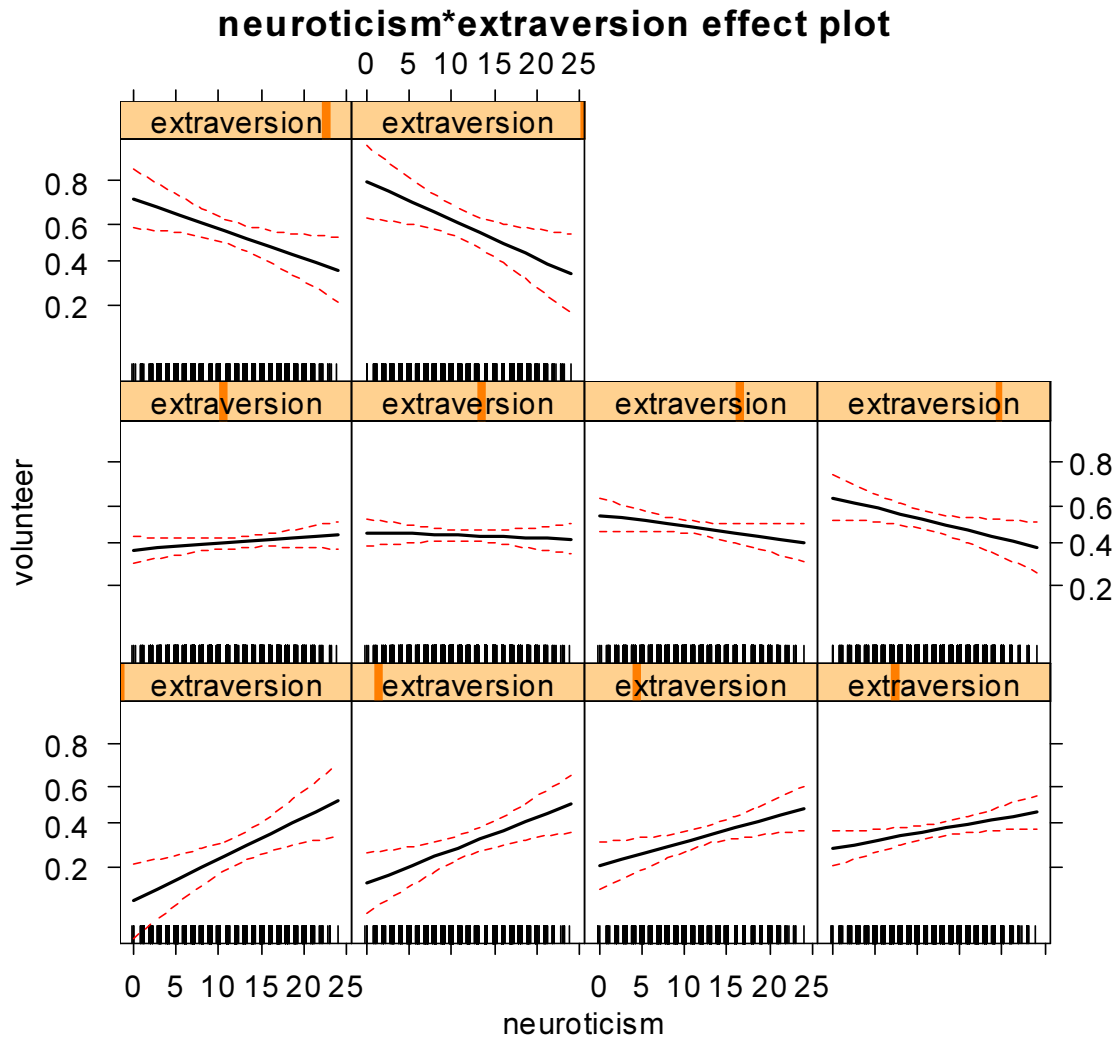
Figure 5: Default effect display for the interaction between two covariates — `neuroticism` and `extraversion` in Cowles and Davis's volunteering data.
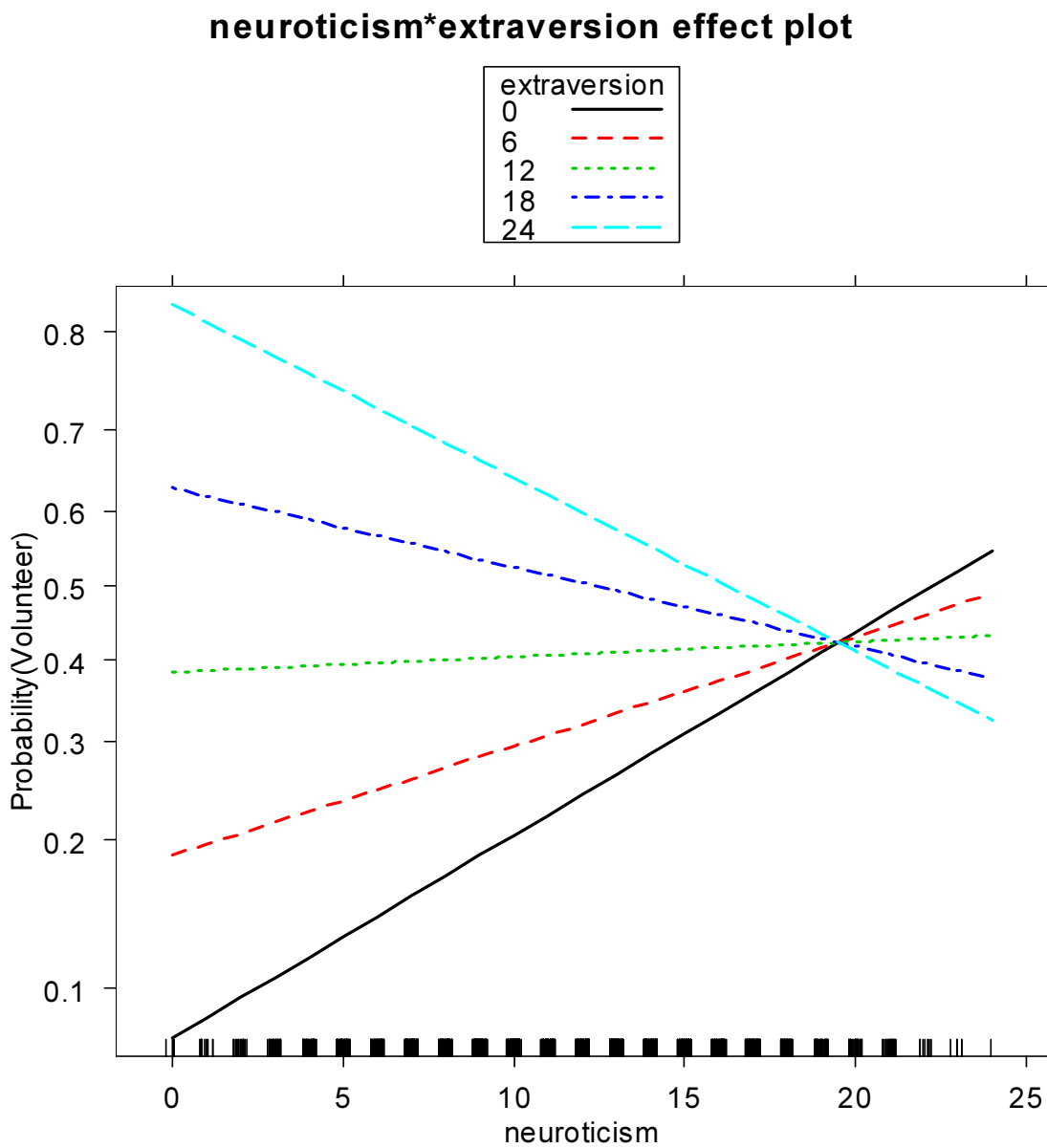
Figure 6: Multi-line effect display for the `neuroticism` by `extraversion` interaction, setting `extraversion` to six values.
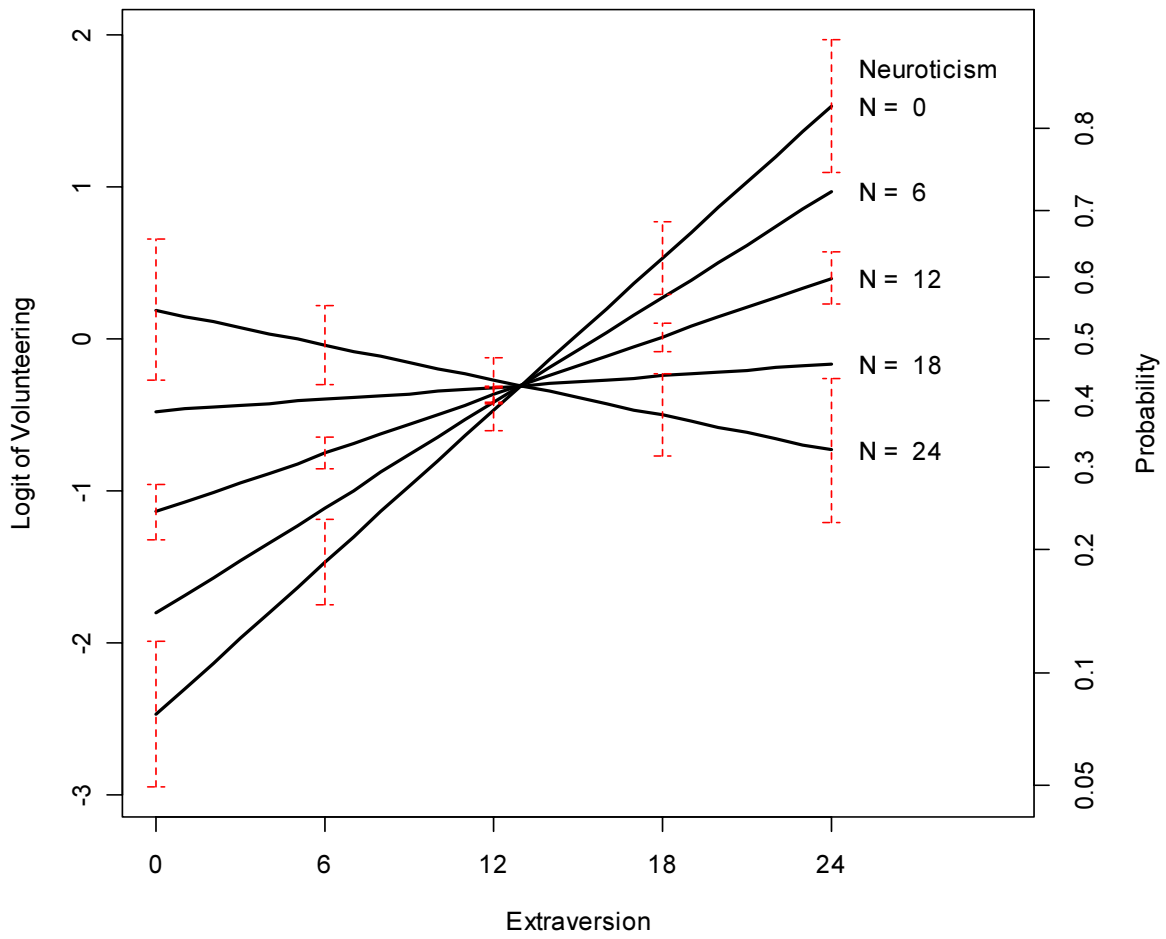
Figure 7: Custom effect display for the `neuroticism` by `extraversion` interaction.

In this display, `extraversion` is plotted on the horizontal axis, and several lines are produced for representative values of `neuroticism`, with the lines labelled at the right of the display rather than in a legend. To reduce clutter, confidence bars are drawn at intervals along the `extraversion` axis, and the confidence level is set to 2/3. Effects are plotted on the logit scale; the vertical axis at the left is labelled on the logit scale, but an additional probability axis appears at the right. The `summary` method for `effect` objects is used to extract matrices (more generally, arrays) of fitted values and confidence limits for the effects.

References

Cowles, M. and Davis, C. (1987). The subject matter of psychology: Volunteers. *British Journal of Social Psychology*, 29:97–102.

Fisher, R. A. (1936). *Statistical Methods for Research Workers, 6th Edition*. Oliver and Boyd, Edinburgh.

Fox, J. (1987). Effect displays for generalized linear models. In Clogg, C. C., editor, *Sociological Methodology 1987*, pages 347–361. American Sociological Association, Washington DC.

Goodnight, J. H. and Harvey, W. R. (1978). Least squares means in the fixed-effect general linear model. Technical Report R-103, SAS Institute, Cary NC.

Hastie, T. J. (1992). Generalized additive models. In Chambers, J. M. and Hastie, T. J., editors, *Statistical Models in S*, pages 249–307. Wadsworth, Pacific Grove CA.

King, G., Tomz, M., and Wittenberg, J. (2000). Making the most of statistical analyses: Improving interpretation and presentation. *American Journal of Political Science*, 44:347–361.

Nelder, J. A. (1977). A reformulation of linear models [with commentary]. *Journal of the Royal Statistical Society, Series A*, 140:48–76.

Searle, S. R., Speed, F. M., and Milliken, G. A. (1980). Population marginal means in the linear model: An alternative to least squares means. *The American Statistician*, 34:216–221.

Tomz, M., Wittenberg, J., and King, G. (2003). Clarify: Software for interpreting and presenting statistical results. *Journal of Statistical Software*, 8:1–29.

Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S, Fourth Edition*. Springer, New York.

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

    data(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.

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

    data(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 volunteeing, 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.

---

effect	*Functions For Constructing Effect Plots*

Description

`effect` constructs an `"effect"` object for a term (usually a high-order term) in a linear or generalized linear model, absorbing the lower-order terms marginal to the term in question, and averaging over other terms in the model.

`all.effects` identifies all of the high-order terms in a model and returns a list of `"effect"` objects (i.e., an object of type `"effect.list"`).

Usage

```
effect(term, mod, xlevels=list(), default.levels=10, se=TRUE,
    confidence.level=.95,
    transformation=list(link=family(mod)$linkfun, inverse=family(mod)$linkinv),
    typical=mean)

all.effects(mod, ...)

as.data.frame(x, row.names=NULL, optional=TRUE)
```

Arguments

`term`	the quoted name of a term, usually, but not necessarily, a high-order term in the model.
`mod`	an object of class `"lm"` or `"glm"`.
`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.

<dl>
<dt>se</dt>
<dd>if TRUE, the default, calculate standard errors and confidence limits for the effects.</dd>

<dt>confidence.level</dt>
<dd>level at which to compute confidence limits based on the standard-normal distribution; the default is 0.95.</dd>

<dt>transformation</dt>
<dd>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.</dd>

<dt>typical</dt>
<dd>a function to be applied to the columns of the model matrix over which the effect is "averaged"; the default is mean.</dd>

<dt>...</dt>
<dd>arguments to be passed down.</dd>

<dt>x</dt>
<dd>an object of type "effect".</dd>

<dt>row.names, optional</dt>
<dd>not used.</dd>
</dl>

Details

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

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

effect returns an "effect" object with the following components:

<dl>
<dt>term</dt>
<dd>the term to which the effect pertains.</dd>

<dt>formula</dt>
<dd>the complete model formula.</dd>

<dt>response</dt>
<dd>a character string giving the response variable.</dd>

<dt>variables</dt>
<dd>a list with information about each predictor, including its name, whether it is a factor, and its levels or values.</dd>
</dl>

3

`fit`	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.
`x`	a data frame, the columns of which are the predictors, and the rows of which give all combinations of values of the 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.
`se`	a vector of standard errors for the effect, on the scale of the linear predictor.
`lower, upper`	one-column matrices of confidence limits, on the scale of the linear predictor.
`confidence.level`	
	corresponding to the confidence limits.
`transformation`	
	a two-element list, with element `link` giving the link function, and element `inverse` giving the inverse-link (mean) function.

Author(s)

John Fox ⟨jfox@mcmaster.ca⟩.

References

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

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

See Also

`print.effect`, `summary.effect`, `plot.effect`, `print.summary.effect`, `print.effect.list`, `summary.effect.list`, `plot.effect.list`, `xyplot`

Examples

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

    model: volunteer ~ sex + neuroticism * extraversion

    sex effect
    sex
```

```
    female      male
    0.4409441 0.3811941

    neuroticism*extraversion effect
            extraversion
    neuroticism          0          6         12         18         24
            0  0.07801066 0.1871263 0.3851143 0.6301824 0.8225756
            1  0.08636001 0.1963396 0.3870453 0.6200668 0.8083638
            2  0.09551039 0.2058918 0.3889798 0.6098458 0.7932997
            3  0.10551835 0.2157839 0.3909179 0.5995275 0.7773775
            . . .
           23 0.51953129 0.4747277 0.4303273 0.3870199 0.3454282
           24 0.54709527 0.4895731 0.4323256 0.3768303 0.3243880

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

    Loading required package: lattice


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)

    Warning message:
    sex:neuroticism:extraversion does not appear in the model in:
      effect("sex:neuroticism:extraversion", mod.cowles,
      xlevels = list(neuroticism = 0:24,

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

Prestige   *Prestige of Canadian Occupations*

Description

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

Usage

```
data(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.effect	*Summarizing, Printing, and Plotting Effects*

Description

`summary`, `print`, and `plot` methods for `effect` and `effect.list` objects.

Usage

```
print(x, type=c("response", "link"), ...)
print(x, ...)
print{x, ...}
summary(object, type=c("response", "link"), ...)
summary(object, ...)
plot(x, x.var=which.max(levels),
    z.var=which.min(levels), multiline=is.null(x$se), rug=TRUE, xlab,
    ylab=x$response, 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, rescale.axis=TRUE, ...)
plot(x, selection, ...)
```

Arguments

x	an object of type `"effect"`, `"summary.effect"`, or `"effect.list"`, as appropriate.
object	an object of type `"effect"` or `"effect.list"`, as appropriate.
type	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.
x.var	the index (number) 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	the index (number) 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	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.
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 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.
symbols, lines	corresponding to the levels of the `z.var` covariate or factor on a multiline plot. These arguments are used only if `multiline = TRUE`; in this case a legend is drawn at the top of the display.
cex	character expansion for plotted symbols; default is `1`.
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.
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.
`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).
`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.
`...`	arguments to be passed down.

Details

In a generalized linear model, by default, the `print` and `summary` methods for `effect` 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.

Value

The `summary` method for `"effect"` objects returns a `"summary.effect"` 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.

Author(s)

John Fox ⟨jfox@mcmaster.ca⟩.

See Also

`effect`, `all.effects`, `xyplot`

Examples

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

```
    extraversion=seq(0, 24, 6)))
eff.cowles

    model: volunteer ~ sex + neuroticism * extraversion

    sex effect
    sex
    female      male
    0.4409441 0.3811941

    neuroticism*extraversion effect
            extraversion
    neuroticism          0         6        12        18        24
            0  0.07801066 0.1871263 0.3851143 0.6301824 0.8225756
            1  0.08636001 0.1963396 0.3870453 0.6200668 0.8083638
            2  0.09551039 0.2058918 0.3889798 0.6098458 0.7932997
            3  0.10551835 0.2157839 0.3909179 0.5995275 0.7773775
            . . .
            23 0.51953129 0.4747277 0.4303273 0.3870199 0.3454282
            24 0.54709527 0.4895731 0.4323256 0.3768303 0.3243880

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

    Loading required package: lattice


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)

    Warning message:
    sex:neuroticism:extraversion does not appear in the model in:
      effect("sex:neuroticism:extraversion", mod.cowles,
      xlevels = list(neuroticism = 0:24,

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