Regression Methods Supported by the effects Package

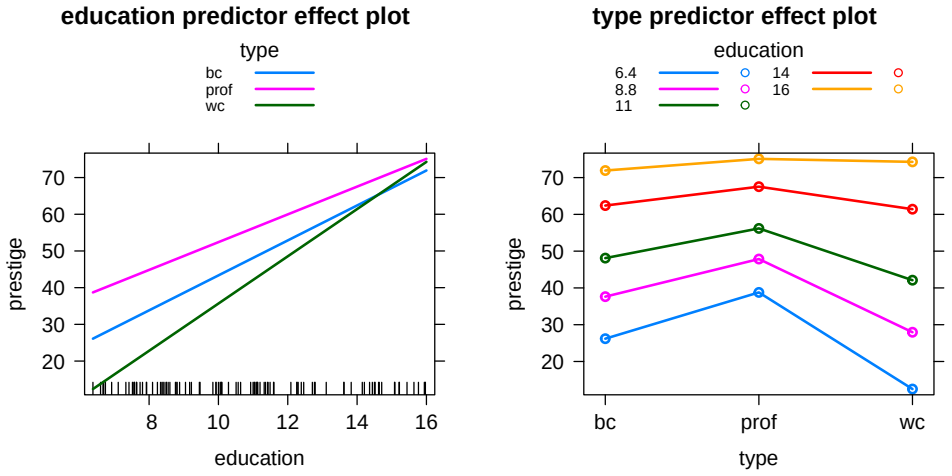
John Fox and Sanford Weisberg

2020-07-21

Effect plots allow visualizing the effect of a predictor on a response in models in which the dependence of the response depends on a linear combination of main-effects and interactions (Fox and Weisberg, 2019, Sec. 4.6.3). Table 1 provides a list of *some* of the regression modeling methods in R that can be used with effect plots.

The most basic type of model for which effects are appropriate is a standard linear model, for example

```
library(effects)
g1 <- lm(prestige ~ education + type + education:type, data = Prestige)
plot(predictorEffects(g1), lines=list(multiline=TRUE))
```



In this example the response `prestige` is modeled as a linear function of years of `education`, a factor `type` either blue collar, professional or white collar. Because of the interaction the estimated change in `prestige` as a function of `education` is different for each level of `type`, as is plainly evident in the graphs shown. The graph shown at the left varies `education` and fixes `type`, while the right graph varies `type` and fixes `education`. A more complicated model is a linear mixed model, for example

```
data(Orthodont, package="nlme")
g2 <- lme4::lmer(distance ~ age + Sex + (1 |Subject), data = Orthodont)
g2
```

```
Linear mixed model fit by REML ['lmerMod']
Formula: distance ~ age + Sex + (1 | Subject)
   Data: Orthodont
REML criterion at convergence: 437.5125
Random effects:
 Groups   Name        Std.Dev.
 Subject  (Intercept) 1.807
 Residual             1.432
```

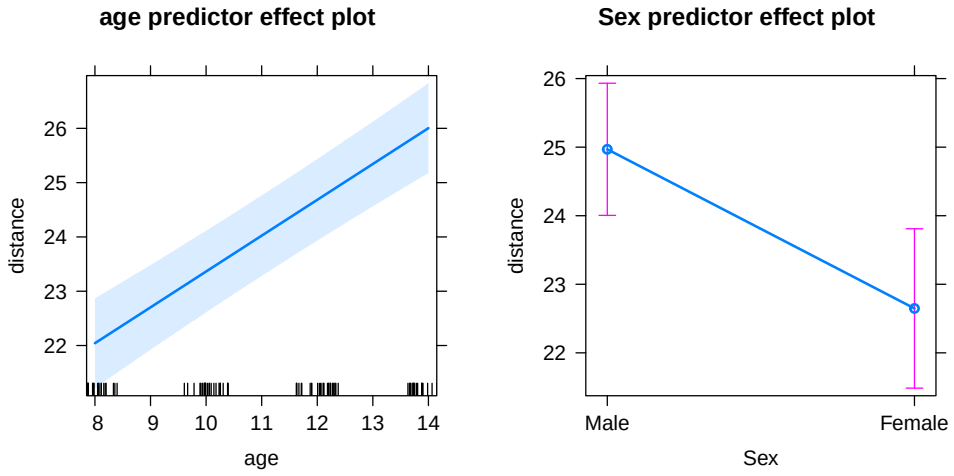
Table 1: R regression models known to be compatible with the **effects** package. The name before the double-colon is the name of the package that includes the function; for example `stats::lm()` means that `lm()` is in the **stats** package.

Function	Comments
glm-type models	
`stats::lm()`	Standard linear regression model. A multivariate response, thus fitting a multivariatel linear model, are permitted, and effect plots are drawn for each response separately.
`stats::glm()`	Generalized linear models
`nlme::lme()`	Linear mixed-effects models. Effects plots for predictors in the fixed-effects part of the model.
`nlme::gls()`	Linear model fit with generalized least squares
`lmer::lmer()`	Linear mixed-effect models. Effects plots are for predictors in the fixed-effects part of the model
`lmer::glmer()`	Generalized linear mixed-effect models. Effects plots for predictors in the fixed-effects part of the model
`survey::svyglm()`	Survey-weighted generalized linear models
`MASS:glmmPQL()`	Generalized linear mixed models via partial quadratic likelihood
`robustlmm::rlmer()`	Robust linear mixed-models
`betareg::betareg()`	Beta regression for rates and proportions
`AER::ivreg()`	Instrumental-variable regression
multinomial type models	
`nnet::multinom()`	Multinomial log-linear models
`poLCA::poLCA()`	Latent class analysis of polytomous outcomes, even though this is not strictly a regression model
ordinal type models	
`MASS:polr()`	Ordinal logistic and probit models
`ordinal::clm()`	Cumulative link proportional odds models, similar to `polr()`
`ordinal::clm2()`	Updated version of `ordinal::clm()`
`ordinal::clmm()`	Cumulative link proportional odds models with random effects

```
Number of obs: 108, groups:  Subject, 27
Fixed Effects:
(Intercept)          age    SexFemale
    17.7067       0.6602      -2.3210
```

This model has a fixed effect part with response `distance` and predictors `age` and `Sex`. The random effect varies by `Subject`. Effect plots are based only on the fixed-effects in the model,

```
plot(predictorEffects(g2))
```



1 Types of Effects Plots

There are three basic types of effects plots. Both of the models just fit are of the `glm-type` which visualizes the dependence of a response on a set of main effects and interactions among fixed effect predictors. As shown in Table 1 most of the models used with effects are of this type.

The `mutlinomial-type` arises when the response is a multinomial random variable, also modeled as a linear function of fixed-effect main effects and interactions. The `poLCA::poLCA` function is of the multinomial-type even though it is philsophocially different from multinomial regression, as it has a latent variable as its response rather than an observable multinomial response.

The `ordinal-type` is used to fit a multinomial response whose categories are ordered, initially applied to results from `MASS::polr`. The other functions shown in Table 1 do similar fitting, but allowing for some generalization in defining cutpoints between categories, and in allowing for random effects.

2 Regression Models of the Glm-type

Effect plots for models of the `glm-type` are drawn by collecting information from the regression model of interest and then using that information in the code for drawing generalized linear model effect plots. The required information is summarized in Table 2.

For a regression model of the `glm-type` for which the defaults don't work we provide a simple mechanism that may allow the model to be used with **effects**. We illustrate by a few examples that are included in the effects package.

2.1 glmmPQL

Objects of type `glmmPQL` do not respond to the generic `family` function, but the name of the family can be obtained from the call:

```
effSources.glmmPQL <- function(mod)
 {list(family = mod$family)}
```

Table 2: Values that must be supplied from a regression object to draw effects plots for a `glm-like` regression model. In the table, the regression model object is called `m1`.

Argument	Description
`call`	The call that created the regression model is generally returned by either `m1$call` or `m1@call`. The call is used to find the usual `data` and `subset` arguments that Effects needs to draw the plots. See the example in Section **??** for an example where the `call` must be modified.
formula	The formula for the linear predictor is required for Effect plots. By default Effects uses `insight::find_formula(m1)$conditional` (see `https://easystats.github.io/insight/`) that will return the fixed-effect formula for many regression models.
`family`	Many `glm-type` models include a family, including an error distribution and a link function. These are often returned by the default `stats::family(m1)`.
`coefficients`	The vector of fixed-effect parameter estimates is required for Effect plots. The default value is `effect::effCoef(m1)` that calls `insight::get_parameters(m1)` and then reformats the result from a two-column data frame with a names in the first column and values in the second column to a vector of named values, as is expected by the effects package.
`vcov`	The estimated covariance of fixed-effect estimates is required. Effects uses `stats::vcov(m1)` by default.

2.2 gls

The `weights` argument has a different meaning in `nlme::gls` and `glm`, the `call` must be modified to set `weights=NULL`

```
effSources.gls <- function(mod){
  cl <- mod$call
  cl$weights <- NULL
  list(call = cl)
}
```

2.3 betareg

The `betareg::betareg` function fits data similar to a binomial regression but with beta errors adapting these models for use with Effects is considerably more complex than the two previous examples.

```
effSources.gls <- function(mod){
  coef <- mod$coefficients$mean
  vco <- vcov(mod)[1:length(coef), 1:length(coef)]
# betareg uses beta errors with mean link given in mod$link$mean.
# Construct a family based on the binomial() family
  fam <- binomial(link=mod$link$mean)
# adjust the variance function to account for beta variance
  fam$variance <- function(mu)
    f0 <- function(mu, eta) (1-mu)*mu/(1+eta)
    do.call("f0", list(mu, mod$coefficient$precision))
# adjust initialize
  fam$initialize <- expression(mustart <- y)
# collect arguments
```

```
  args <- list(
    call = mod$call,
    formula = formula(mod),
    family=fam,
    coefficients = coef,
    vcov = vco)
  args
}
```

References

Fox, J. and S. Weisberg (2019). *An* R *Companion to Applied Regression* (3nd ed.). Thousand Oaks CA: Sage.