Tutorial 13: Generalized linear models

Johannes Karreth

RPOS 517, Day 13

This tutorial shows you:

- how to fit logit, probit, and other generalized linear models in R
- how to create effect plots for these models
- how to calculate other quantities of interest for GLMs

Note on copying & pasting code from the PDF version of this tutorial: Please note that you may run into trouble if you copy & paste code from the PDF version of this tutorial into your R script. When the PDF is created, some characters (for instance, quotation marks or indentations) are converted into non-text characters that R won't recognize. To use code from this tutorial, please type it yourself into your R script or you may copy & paste code from the *source file* for this tutorial which is posted on my website.

Note on R functions discussed in this tutorial: I don't discuss many functions in detail here and therefore I encourage you to look up the help files for these functions or search the web for them before you use them. This will help you understand the functions better. Each of these functions is well-documented either in its help file (which you can access in R by typing `?ifelse`, for instance) or on the web. The *Companion to Applied Regression* (see our syllabus) also provides many detailed explanations.

As always, please note that this tutorial only accompanies the other materials for Day 13 and that you need to have worked through the reading for that day before tackling this tutorial. **More than on the other days of our seminar so far, these notes only scratch the surface of the theory behind GLMs.** I strongly encourage you to re-read in-depth chapters 14 and 15 in *AR* and other materials before you use GLMs in your work.

Noncontinuous outcomes and OLS

So far, we have only encountered continuous outcomes: variables with values across a continuous range, such as wages, life expectancy, feeling thermometers, etc. Many variables around social phenomena are not continuous: whether a citizen voted; which party she voted for; whether two countries go to war; whether a patient complies with a prescription; how a consumer rates a product; etc.

Binary outcomes

Among these outcomes, you could treat **binary** outcomes as continuous and fit a so-called *linear probability model* (see *AR* section 14.1.1), but will encounter the following issues (paraphrased from *AR*):

(a) the residuals will be dichotomous and therefore not normally distributed
(b) the error variance will not be constant
(c) the expected value of the errors (i.e., the linearity assumption) only holds for some values of a continuous x
(d) the model will yield predicted values ($\hat{y}$) not just between 0 and 1 (which you could interpret as the probability of an outcome of 1), but also below 0 and 1 - because it estimates a linear relationship between x and y. If you are fitting a model for predictive purposes, this could be inconvenient.

While (d) may not be a concern for you, (a), (b), and (c) present issues for OLS because the desirable qualities of the OLS estimator depend on assumptions about the error term. If these assumptions are violated, OLS may not be the best linear unbiased estimator (BLUE) anymore. To remedy this problem, one can assume a distribution for the latent probability of an outcome of 1.

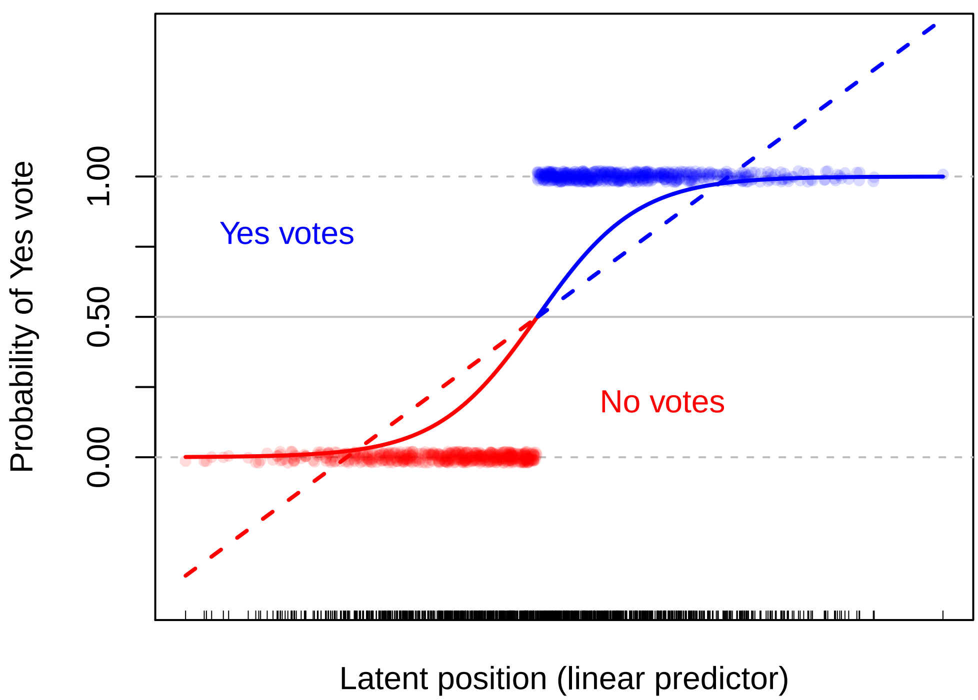
Other noncontinuous outcomes

Before we advance, think about other types of noncontinuous outcomes as well (see *AR* sections assigned for today and below for more details):

- ordered outcomes (e.g., rating a product as dissatisfactory, satisfactory, and excellent)
- discrete outcomes (e.g., vote choices in a multiparty system)
- count outcomes (with many values of 0 and 1 and decreasing frequency of higher values)

Binary outcomes: illustration

A typical binary outcome in political science is whether a representative votes for or against a particular bill. You can think of the actual vote as the realization of a legislator's latent position on an issue (compare this to section 14.1.3 in *AR*). If we place this latent position on a continuous scale, we may often observe the following relationship between latent position (x) and the probability of a "yes" vote: on positions at low values of the latent scale, legislators are highly unlikely to vote "yes". Only for legislators in the middle quintile (for example purposes) of the latent positions do we observe somewhat similar probabilities for a "yes" and "no" vote; and legislators close high values are all very likely to vote "yes". In the visualization below, you can see why a linear relationship between latent position and the probability to vote would not accurately reflect the data-generating process of what we observe as votes:



In this figure, you can see how the DGP (the solid line) creates observed Yes and No votes (the red and blue points): Yes votes when the probability of a Yes vote exceeds 0.5, and No votes below. Also note that these data do not include any random component: the middle value of the latent position clearly separates Yes and No votes.

You can also see how the above problems play out: the OLS estimate (the dashed line) doesn't fit the true relationship between the latent position and the probability of Yes votes (the solid line) all that well. In addition, the OLS estimates return probabilities below 0 and above 1.

If you return to *AR*, you can identify the function that connects the probability of a Yes vote and the latent position as the cumulative density function (CDF) of the **link function** that we use to map the observed binary variable to an unobserved/latent linear predictor. In the case above, the CDF is the logistic distribution. Because the link function helps map a predictor to a non-continuous outcome (in this case, an observed 0 or 1), you can think of models using link functions as *generalized linear models* - the topic of this week.

Assessing the relationship between x and y in GLMs

The figure above also shows you at least one more important quality of GLMs: the "effect" of a predictor on an outcome is **not constant**. That is, you cannot make a statement of the form, "a one-unit increase in x is associated with a 0.2 increase in the probability of a Yes vote." Instead, the relationship between x and $\Pr(y)$ depends on the value of x. In the example above, the relationship is almost flat at low and high values of x, but quite steep in the middle. Also note that the relationship between x and $\Pr(y)$ is fairly linear around the center of x.

Obtaining estimates via maximum likelihood

Lastly, and as *AR* explains in more detail in section 14.1.5, you may already have concluded that GLMs cannot be fit by minimizing squared residuals, as you did to identify the parameter estimates in OLS. Instead, GLMs are fit using maximum likelihood estimation (MLE). A discussion of MLE goes beyond our course, but you should take a look at steps 1-3 in *AR* p. 354, which explains the logic behind MLE in general terms.

Example of a binary outcome: Female labor participation

In this section, I demonstrate the use of one GLM, the logit (or logistic regression) model for binary outcomes. Generally, GLMs are named after the link function they employ. Estimators using the logistic link function are logistic regression (or logit) models, for instance. We'll work with an example from John Fox's teaching datasets on female labor participation. The original source for the article is Mroz, T. A. (1987). "The sensitivity of an empirical model of married women's hours of work to economic and statistical assumptions." *Econometrica* 55, 765–799. The dataset is available at John Fox's website. This dataset is a sample of "753 married white women between the ages of 30 and 60 in 1975, with 428 working at some time during the year" (Mroz 1987: 769). It contains the following variables:

Variable	Description
`lfp`	Labor-force participation: no, yes.
`k5`	Number of children 5 years old or younger.
`k618`	Number of children 6 to 18 years old.
`age`	Age in years
`wc`	Wife's college attendance, no; yes.
`hc`	Husband's college attendance: no, yes.
`lwg`	Log expected wage rate; for women in the labor force
`inc`	Family income exclusive of wife's income

Because it is stored online in tab-separated format, I'll read it into R using the `read.table()` function.

```r
mroz.dat <- read.table("http://socserv.mcmaster.ca/jfox/Books/Applied-Regression-3E/datasets/Mroz.txt")
summary(mroz.dat)
```

```
##    lfp            k5              k618             age          wc
##  no :325   Min.   :0.0000   Min.   :0.000   Min.   :30.00   no :541
##  yes:428   1st Qu.:0.0000   1st Qu.:0.000   1st Qu.:36.00   yes:212
##            Median :0.0000   Median :1.000   Median :43.00
##            Mean   :0.2377   Mean   :1.353   Mean   :42.54
##            3rd Qu.:0.0000   3rd Qu.:2.000   3rd Qu.:49.00
##            Max.   :3.0000   Max.   :8.000   Max.   :60.00
##    hc           lwg               inc
##  no :458   Min.   :-2.0541   Min.   :-0.029
##  yes:295   1st Qu.: 0.8181   1st Qu.:13.025
##            Median : 1.0684   Median :17.700
##            Mean   : 1.0971   Mean   :20.129
##            3rd Qu.: 1.3997   3rd Qu.:24.466
##            Max.   : 3.2189   Max.   :96.000
```

To identify correlates of a married woman's propensity to be working, we fit a logistic regression model using the `glm()` function in R. This function works just like `lm()`, but you have to specify the link function (see above). In the case of a binary outcome, we'll use the logit link function by specifying the argument `family = binomial(link = "logit")`.

```r
mroz.logit <- glm(lfp ~ k5 + k618 + age + wc + hc + lwg + inc, data = mroz.dat,
                  family = binomial(link = "logit"))
summary(mroz.logit)
```

```
##
## Call:
## glm(formula = lfp ~ k5 + k618 + age + wc + hc + lwg + inc, family = binomial(link = "logit"),
##     data = mroz.dat)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1062  -1.0900   0.5978   0.9709   2.1893
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.182140   0.644375   4.938 7.88e-07 ***
## k5          -1.462913   0.197001  -7.426 1.12e-13 ***
## k618        -0.064571   0.068001  -0.950 0.342337
## age         -0.062871   0.012783  -4.918 8.73e-07 ***
## wcyes        0.807274   0.229980   3.510 0.000448 ***
## hcyes        0.111734   0.206040   0.542 0.587618
## lwg          0.604693   0.150818   4.009 6.09e-05 ***
## inc         -0.034446   0.008208  -4.196 2.71e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1029.75  on 752  degrees of freedom
```

```
## Residual deviance:  905.27  on 745  degrees of freedom
## AIC: 921.27
##
## Number of Fisher Scoring iterations: 4
```

This output looks familiar to you: coefficients, standard errors, z-values, and p-values. While you may often hear that coefficients from logit models (and other GLMs) are difficult to interpret, that is not necessarily true. The only difficulty with coefficients like those from a logit model is that the relationship between x and $\Pr(y)$ now depends on the value of x (see above). But keeping in mind the logit curve you saw above, and doing some calculations involving the logit cumulative density function, you can calculate that the effect of x on $\Pr(y)$ is about $0.25 \times \beta$ at the maximum steep point of the curve (at $\Pr(y) = 0.5$). That's good enough for a quick interpretation: a one-unit increase of x is associated with a maximum of a $0.25 \times \beta$ increase in the probability of y being 1.

Predicted probabilities

But to take advantage of the nonlinear relationship estimated by the logit model, you should investigate the predicted probability of a "1" outcome across the range of each predictor. To do this, use the "effects" package written by John Fox. It works in two steps: first, calculate the `effect` of a predictor; second, `plot` the predictor. The package provides a shorthand to plot *all* effects. We'll look at this first:

```r
# install.packages("effects")
library(effects)
all.effects <- allEffects(mod = mroz.logit)
summary(all.effects)
```

```
##  model: lfp ~ k5 + k618 + age + wc + hc + lwg + inc
##
##  k5 effect
## k5
##          0        0.5          1        1.5          2        2.5
## 0.65959332 0.48251362 0.30971959 0.17757166 0.09411937 0.04761597
##          3
## 0.02349352
##
##  Lower 95 Percent Confidence Limits
## k5
##           0          0.5           1         1.5           2         2.5
## 0.617154812 0.436140401 0.243444533 0.114715652 0.049217887 0.020201735
##           3
## 0.008133988
##
##  Upper 95 Percent Confidence Limits
## k5
##          0        0.5          1        1.5          2        2.5
## 0.69961721 0.52919000 0.38485901 0.26457539 0.17255031 0.10812654
##          3
## 0.06592884
##
##  k618 effect
## k618
##         0         2         4         6         8
```

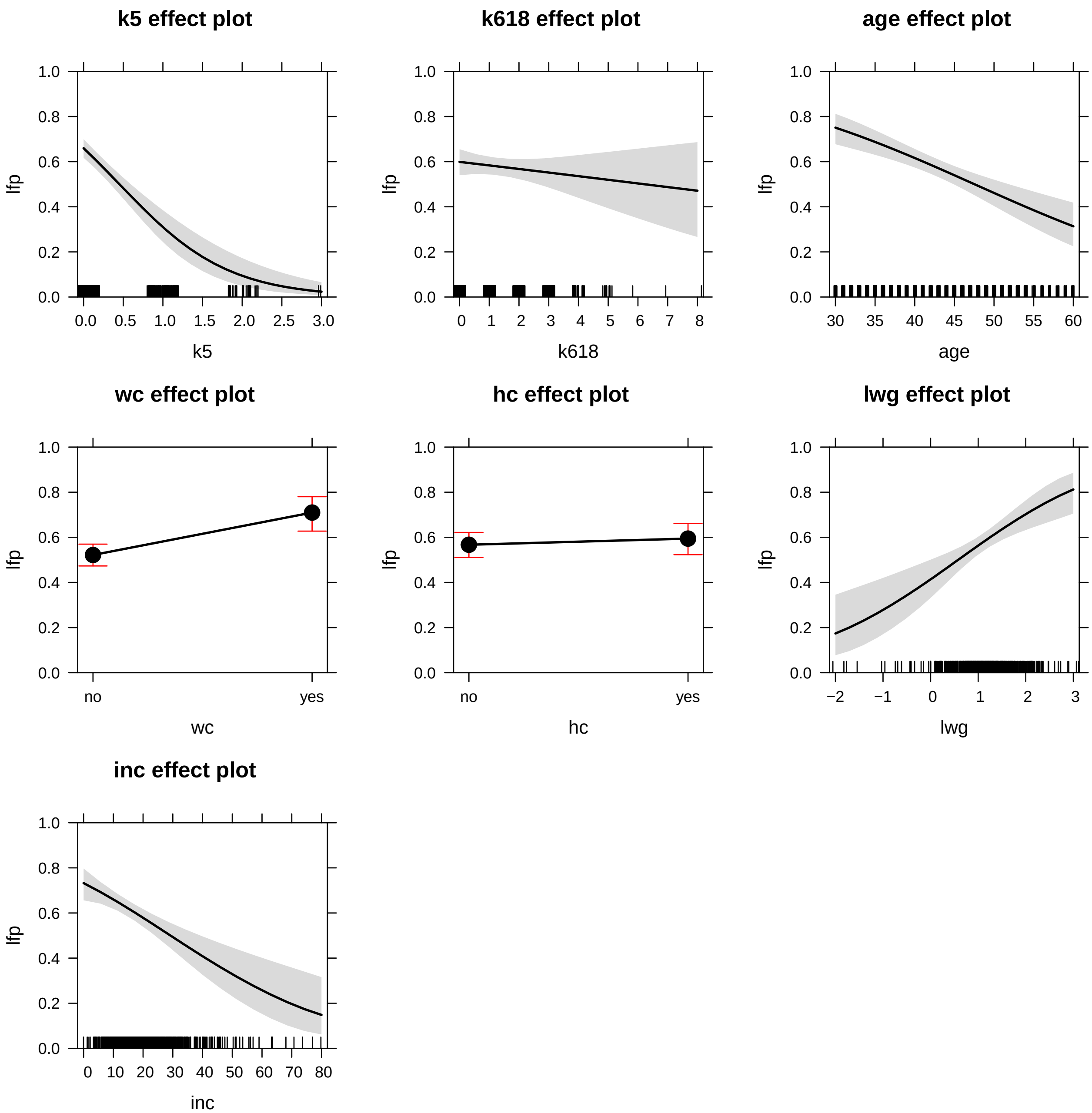
```
## 0.5989532 0.5675750 0.5356451 0.5034203 0.4711670
##
##  Lower 95 Percent Confidence Limits
## k618
##         0         2         4         6         8
## 0.5401157 0.5229579 0.4394343 0.3486106 0.2660092
##
##  Upper 95 Percent Confidence Limits
## k618
##         0         2         4         6         8
## 0.6550704 0.6111222 0.6292740 0.6575769 0.6865513
##
##  age effect
## age
##        30        35        40        45        50        55        60
## 0.7506321 0.6873230 0.6161600 0.5396486 0.4612219 0.3846689 0.3134304
##
##  Lower 95 Percent Confidence Limits
## age
##        30        35        40        45        50        55        60
## 0.6770939 0.6301447 0.5740968 0.4982728 0.4032849 0.3079293 0.2246363
##
##  Upper 95 Percent Confidence Limits
## age
##        30        35        40        45        50        55        60
## 0.8120711 0.7393185 0.6565542 0.5804851 0.5202258 0.4676112 0.4183839
##
##  wc effect
## wc
##        no       yes
## 0.5215977 0.7096569
##
##  Lower 95 Percent Confidence Limits
## wc
##        no       yes
## 0.4730588 0.6274663
##
##  Upper 95 Percent Confidence Limits
## wc
##        no       yes
## 0.5697321 0.7800700
##
##  hc effect
## hc
##        no       yes
## 0.5670810 0.5942794
##
##  Lower 95 Percent Confidence Limits
## hc
##        no       yes
## 0.5109247 0.5230778
##
##  Upper 95 Percent Confidence Limits
## hc
```

```
##        no       yes
## 0.6215653 0.6617255
##
##  lwg effect
## lwg
##        -2        -1         0         1         2         3
## 0.1737788 0.2780036 0.4134569 0.5634068 0.7025966 0.8122027
##
##  Lower 95 Percent Confidence Limits
## lwg
##         -2          -1           0           1           2           3
## 0.07725509 0.16989422 0.33107929 0.52386555 0.63240897 0.70524162
##
##  Upper 95 Percent Confidence Limits
## lwg
##        -2        -1         0         1         2         3
## 0.3457173 0.4200922 0.5009805 0.6021582 0.7643758 0.8865915
##
##  inc effect
## inc
##         0        20        40        60        80
## 0.7324514 0.5788775 0.4083571 0.2573687 0.1482215
##
##  Lower 95 Percent Confidence Limits
## inc
##          0          20          40          60          80
## 0.65607467 0.53987096 0.32591408 0.15192987 0.06157788
##
##  Upper 95 Percent Confidence Limits
## inc
##         0        20        40        60        80
## 0.7971121 0.6169238 0.4963004 0.4013519 0.3157571
```

Summarizing this object returns point estimates and 95% confidence intervals for predicted probabilities of being in the labor force across the range of each predictor.

To plot these effects, simply use the `plot()` function and add the option `rescale.axis = FALSE`:

```
plot(all.effects, rescale.axis = FALSE, ylim = c(0, 1))
```



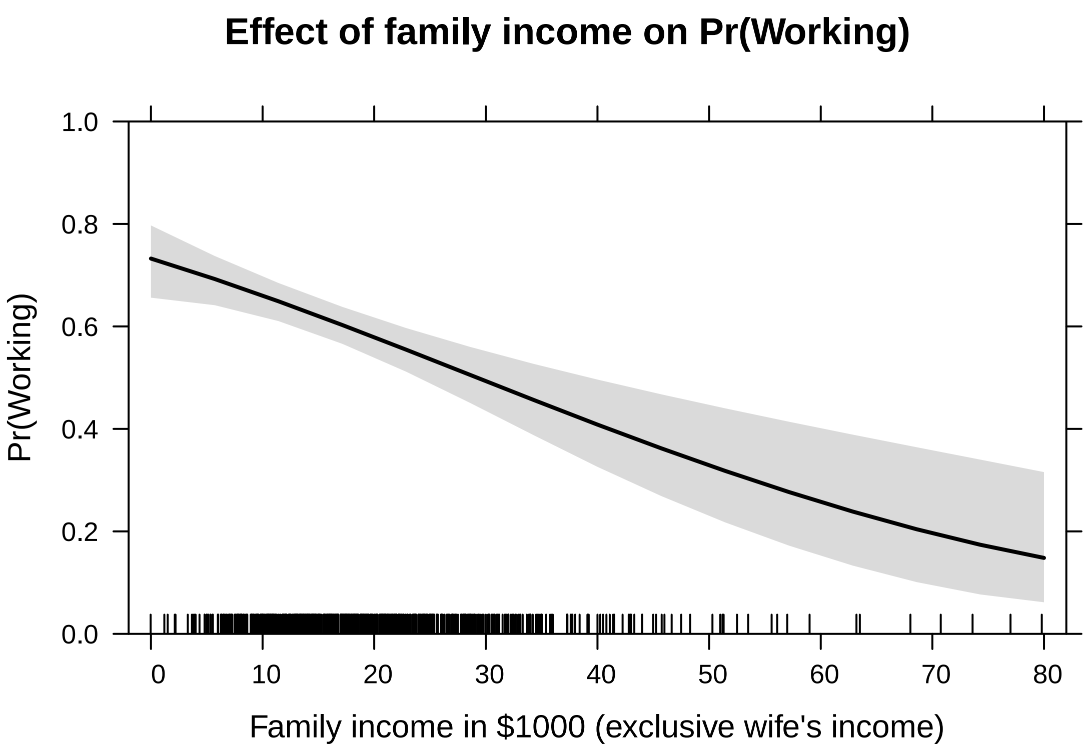
To have more control over these plots, you can plot individual effects as follows:

```
inc.eff <- effect(mod = mroz.logit, term = "inc")
summary(inc.eff)
```

```
##
##  inc effect
## inc
##         0        20        40        60        80
## 0.7324514 0.5788775 0.4083571 0.2573687 0.1482215
##
##  Lower 95 Percent Confidence Limits
## inc
```

```
##          0          20          40          60          80
## 0.65607467 0.53987096 0.32591408 0.15192987 0.06157788
##
##  Upper 95 Percent Confidence Limits
## inc
##         0         20         40         60         80
## 0.7971121 0.6169238 0.4963004 0.4013519 0.3157571
```

```
plot(inc.eff, rescale.axis = FALSE, ylim = c(0, 1),
     main = "Effect of family income on Pr(Working)",
     xlab = "Family income in $1000 (exclusive wife's income)",
     ylab = "Pr(Working)")
```



Lastly, note that GLMs do not render R^2 as a meaningful model fit statistic. There are plenty of alternative measures for model fit. One of them is the proportional reduction of error: how much does the model better classify outcomes than does the modal prediction (i.e., predicting the modal outcome for each observation)? To calculate PRE, you can use the `pre()` function from Dave Armstrong's "DAMisc" package.

```
# install.packages("DAmisc")
library(DAMisc)
pre(mroz.logit)
```

```
## mod1:  lfp ~ k5 + k618 + age + wc + hc + lwg + inc
## mod2:  lfp ~ 1
##
## Analytical Results
##  PMC =  0.568
##  PCP =  0.693
##  PRE =  0.289
```

```
## ePMC =  0.509
## ePCP =  0.585
## ePRE =  0.154
```

This discussion barely scratches the surface of GLMs and maximum likelihood estimation. Please see* AR *and the syllabus for further reading recommendations. Please also know that many of the terms we've discussed for OLS (e.g., robust standard errors, fixed effects, interaction terms) work differently for GLMs.

Example of a binary outcome: Voting on NAFTA

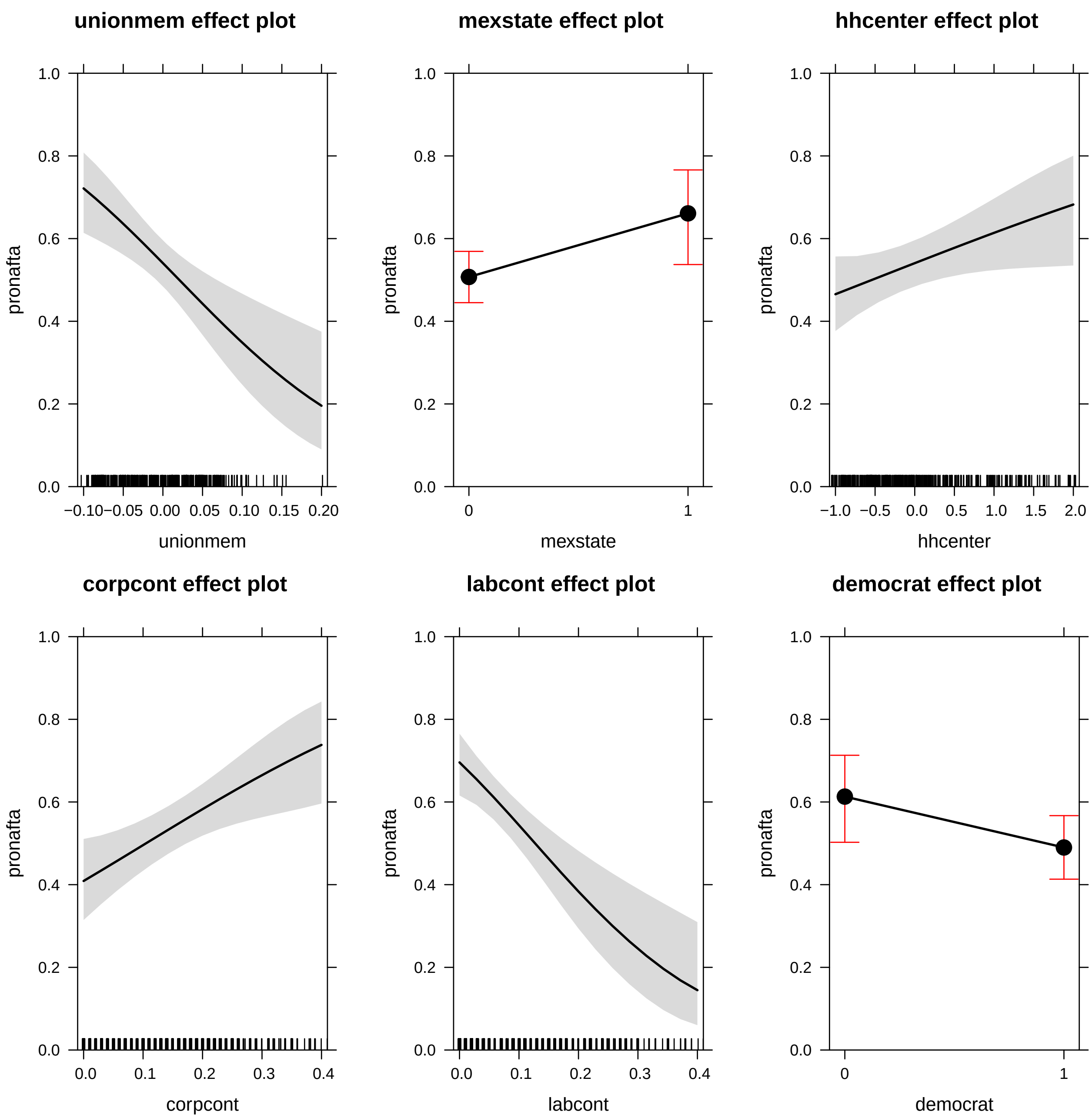
Our second, in-class, example addresses the correlates of U.S. Representatives' vote choice on the North American Free Trade Agreement in 1993. The data come from the article "The Strategic Timing of Position Taking in Congress: A Study of the North American Free Trade Agreement" by Janet Box-Steffensmeier, Laura Arnold, and Chris Zorn in the *APSR* in 1993. Replication data for the article are available at http://hdl.handle.net/1902.1/15557. I renamed some of the variables and uploaded the modified version of the data to my website. See the article for a description of the research question and variables.

```r
nafta.dat <- read.csv("http://www.jkarreth.net/files/naftavote.csv")
summary(nafta.dat)
```

```
##     pronafta         unionmem             perotvot
##  Min.   :0.0000   Min.   :-1.030e-01   Min.   :-1.528e-01
##  1st Qu.:0.0000   1st Qu.:-4.750e-02   1st Qu.:-4.343e-02
##  Median :1.0000   Median : 1.222e-04   Median : 1.100e-02
##  Mean   :0.5392   Mean   : 8.379e-05   Mean   : 2.000e-08
##  3rd Qu.:1.0000   3rd Qu.: 4.488e-02   3rd Qu.: 4.572e-02
##  Max.   :1.0000   Max.   : 2.013e-01   Max.   : 1.469e-01
##  NA's   :1
##     mexstate         hhcenter             corpcont         labcont
##  Min.   :0.0000   Min.   :-1.6200880   Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:-0.5729380   1st Qu.:0.0700   1st Qu.:0.0000
##  Median :0.0000   Median :-0.1810879   Median :0.1400   Median :0.0600
##  Mean   :0.2092   Mean   : 0.0000003   Mean   :0.1511   Mean   :0.1018
##  3rd Qu.:0.0000   3rd Qu.: 0.4400119   3rd Qu.:0.2100   3rd Qu.:0.1750
##  Max.   :1.0000   Max.   : 2.6503120   Max.   :0.4200   Max.   :0.4800
##
##     democrat         district           name
##  Min.   :0.0000   Min.   : 0.000   Johnson:  5
##  1st Qu.:0.0000   1st Qu.: 3.000   Smith  :  5
##  Median :1.0000   Median : 6.000   Andrews:  3
##  Mean   :0.5931   Mean   : 9.963   Brown  :  3
##  3rd Qu.:1.0000   3rd Qu.:13.000   Collins:  3
##  Max.   :1.0000   Max.   :52.000   Lewis  :  3
##                                    (Other):413
```

```r
nafta.dat$pronafta <- factor(nafta.dat$pronafta)
nafta.dat$mexstate <- factor(nafta.dat$mexstate)
nafta.dat$democrat <- factor(nafta.dat$democrat)

logit.mod <- glm(pronafta ~ unionmem + mexstate + hhcenter +
```

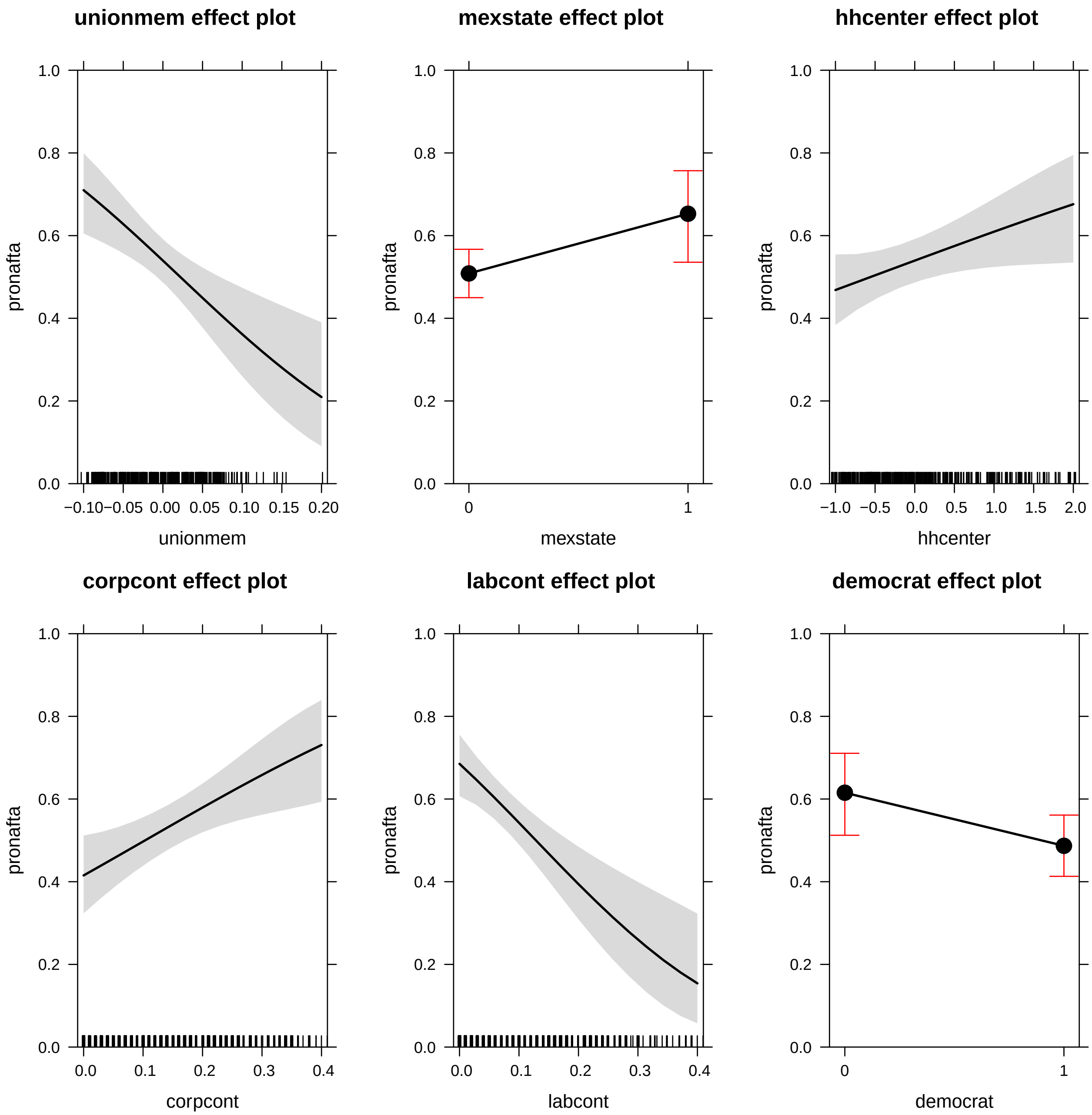
```r
                     corpcont + labcont + democrat,
                   data = nafta.dat,
                   family = binomial(link = "logit"))

library(effects)
plot(allEffects(logit.mod), rescale.axis = FALSE, ylim = c(0, 1))
```



Note that using the probit instead of logit link function returns identical predicted probabilities:

```r
probit.mod <- glm(pronafta ~ unionmem + mexstate + hhcenter +
                     corpcont + labcont + democrat,
                   data = nafta.dat,
                   family = binomial(link = "probit"))
```

```r
plot(allEffects(probit.mod), rescale.axis = FALSE, ylim = c(0, 1))
```



Example of an ordered outcomes: Beer ratings

This example estimates an ordered logit model to identify correlates of how an individual rated 35 different beers. We will discuss more details in class. The outcome variable, `quality` takes three values, 1, 2, and 3, for how the rater evaluated each beer's quality (low, medium, and high).

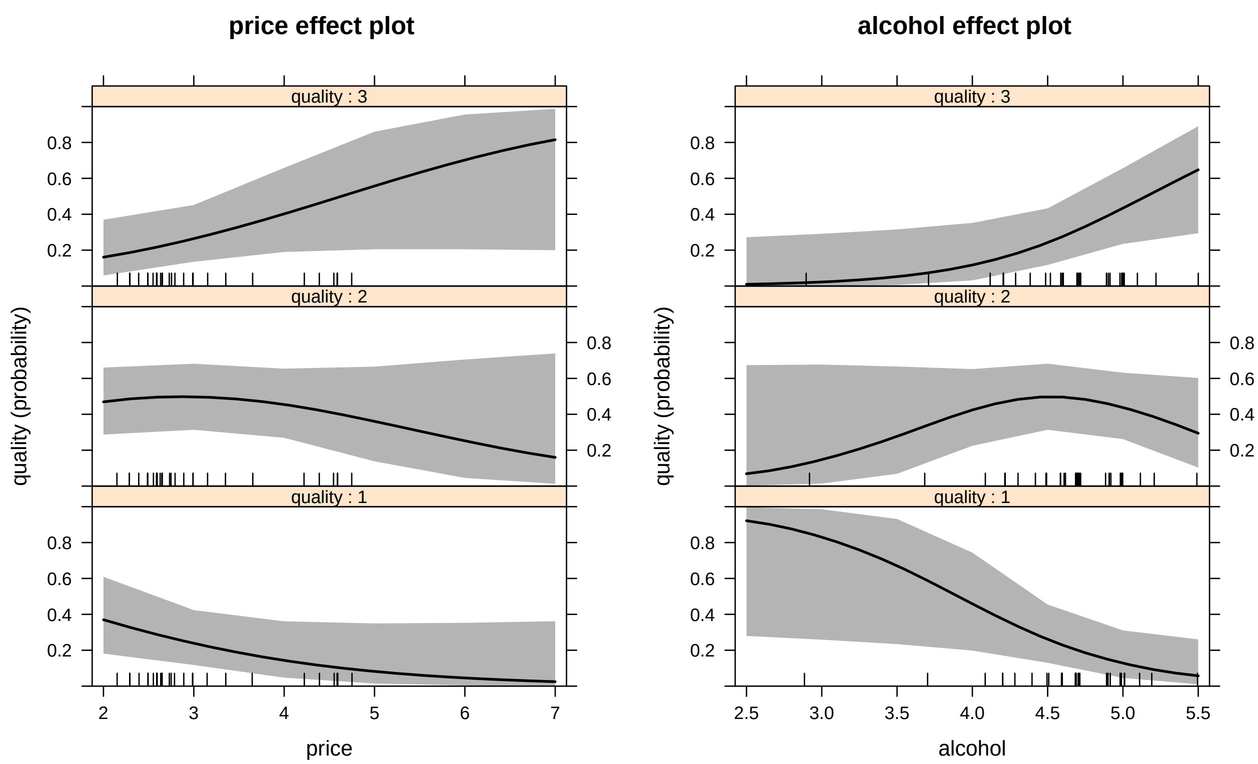
```r
beer.dat <- read.csv("http://www.jkarreth.net/files/beer.csv")
summary(beer.dat)
```

```
##       price          alcohol         quality     
##  Min.   :1.590   Min.   :2.300   Min.   :1.000  
##  1st Qu.:2.490   1st Qu.:4.500   1st Qu.:1.000  
##  Median :2.650   Median :4.700   Median :2.000  
##  Mean   :3.027   Mean   :4.577   Mean   :2.029  
##  3rd Qu.:3.250   3rd Qu.:4.900   3rd Qu.:3.000  
##  Max.   :7.190   Max.   :5.500   Max.   :3.000  
```

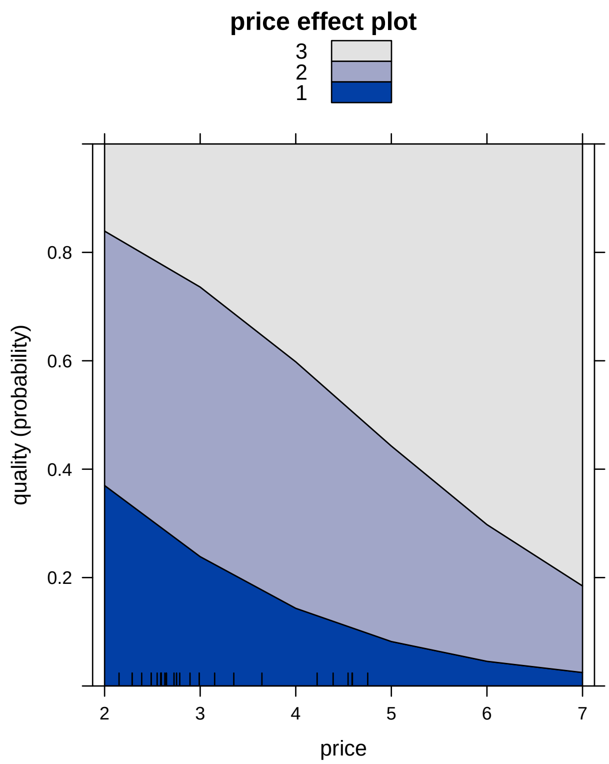
To estimate the ordered logit model, you can use the `polr()` function from the "MASS" package. To estimate the model, you have to convert the outcome to a factor variable.

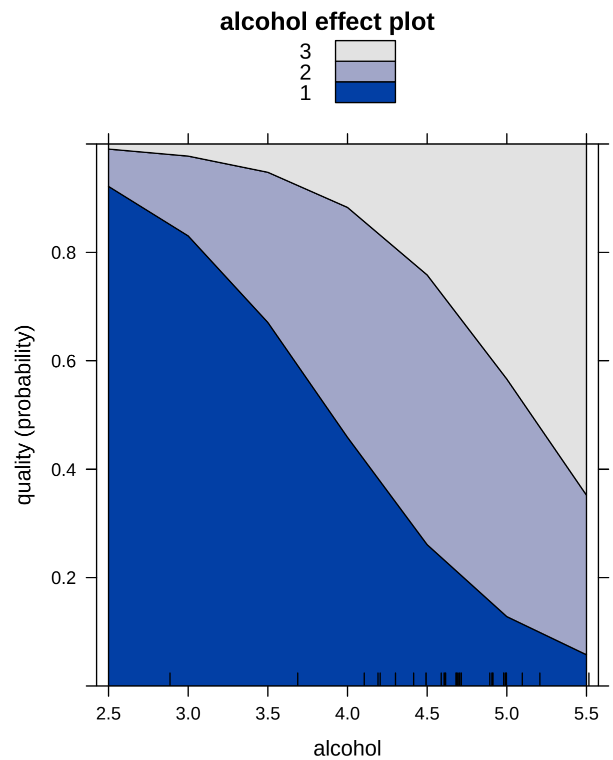
```
# install.packages("MASS")
library(MASS)
beer.dat$quality <- as.factor(beer.dat$quality)
beer.mod <- polr(quality ~ price + alcohol, data = beer.dat,
                 method = "logistic")
```

```
plot(allEffects(beer.mod), rescale.axis = FALSE, ylim = c(0, 1))
```




```
plot(allEffects(beer.mod), rescale.axis = FALSE, ylim = c(0, 1), style = "stacked")
```





Example of a categorical outcome: Vote choice

This example estimates a multinomial logit model to identify correlates of party vote choice in Austrian elections. We will discuss more details in class. The outcome variable is vote choice for one of Austria's four major parties: the Greens (on the left), the SPÖ (center-left), the ÖVP (center-right), and the FPÖ (far-right). Predictors include an assessment of the development of the economy under the incumbent party (the SPÖ), respondents left/right self-placement, and some demographic characteristics.

To fit the multinomial logit model, you can use the `multinom()` function from the "nnet" package.

```
votes.dat <- read.csv("http://www.jkarreth.net/files/austriavote.csv")
summary(votes.dat)
```

```
##     econwor          econbet             age            income
##  Min.   :0.0000   Min.   :0.0000   Min.   :19.00   Min.   : 1.000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:34.00   1st Qu.: 6.000
##  Median :0.0000   Median :0.0000   Median :44.00   Median : 9.000
##  Mean   :0.4634   Mean   :0.1761   Mean   :46.26   Mean   : 8.894
##  3rd Qu.:1.0000   3rd Qu.:0.0000   3rd Qu.:57.00   3rd Qu.:12.000
##  Max.   :1.0000   Max.   :1.0000   Max.   :95.00   Max.   :14.000
##      rural           gender        leftrt           vote
##  Min.   :1.000   Min.   :1.0   Min.   : 0.000   FPO  : 77
##  1st Qu.:3.000   1st Qu.:1.0   1st Qu.: 5.000   Green: 89
##  Median :5.000   Median :1.0   Median : 5.000   OVP  :593
##  Mean   :3.854   Mean   :1.5   Mean   : 5.452   SPO  :456
##  3rd Qu.:5.000   3rd Qu.:2.0   3rd Qu.: 7.000
##  Max.   :5.000   Max.   :2.0   Max.   :10.000
```

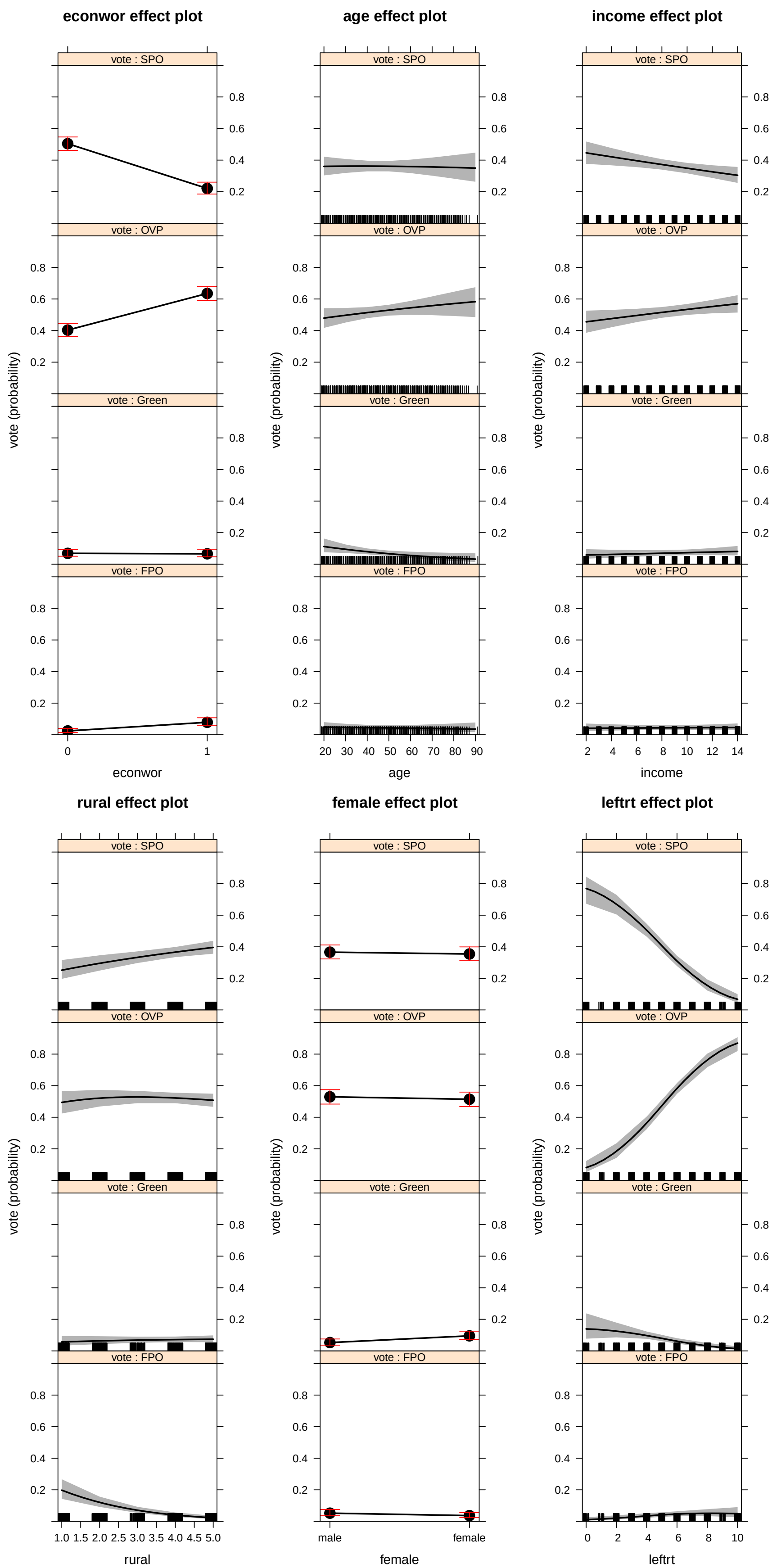
```r
# install.packages("nnet")
library(nnet)
votes.dat$econwor <- factor(votes.dat$econwor)
votes.dat$female <- factor(votes.dat$gender, levels = c(1, 2), labels = c("male", "female"))
votes.mod <- multinom(vote ~ econwor + age + income + rural + female + leftrt, data = votes.dat)
```

```
## # weights:  32 (21 variable)
## initial  value 1684.347649
## iter  10 value 1220.383764
## iter  20 value 1121.427609
## iter  30 value 1113.300047
## iter  30 value 1113.300045
## iter  30 value 1113.300045
## final  value 1113.300045
## converged
```

```r
summary(votes.mod)
```

```
## Call:
## multinom(formula = vote ~ econwor + age + income + rural + female +
##     leftrt, data = votes.dat)
##
## Coefficients:
##       (Intercept)   econwor1          age      income     rural
## Green   0.7711522 -1.2290573 -0.013456818  0.01776986 0.6073459
## OVP    -0.3461488 -0.7316678  0.007265787  0.00926312 0.5485146
## SPO     2.6916050 -2.0134602  0.004026299 -0.04141072 0.6549246
##       femalefemale     leftrt
## Green    0.9473325 -0.3789252
## OVP      0.3240601  0.0867785
## SPO      0.3212995 -0.3942745
##
## Std. Errors:
##       (Intercept)  econwor1         age     income      rural femalefemale
## Green   0.8847837 0.3515757 0.011424612 0.04623279 0.11349515    0.3328014
## OVP     0.6853877 0.2844044 0.008587375 0.03532260 0.08287117    0.2548040
## SPO     0.7096441 0.2954789 0.008989792 0.03740926 0.08926515    0.2690964
##           leftrt
## Green 0.09341987
## OVP   0.07049421
## SPO   0.07623634
##
## Residual Deviance: 2226.6
## AIC: 2268.6
```

```r
plot(allEffects(votes.mod), rescale.axis = FALSE, ylim = c(0, 1))
```


econwor effect plot
age effect plot
income effect plot
rural effect plot
female effect plot
leftrt effect plot
vote : SPO
vote : OVP
vote : Green
vote : FPO
vote (probability)
0.2
0.4
0.6
0.8
econwor
0
1
age
20 30 40 50 60 70 80 90
income
2 4 6 8 10 12 14
rural
1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
female
male
female
leftrt
0 2 4 6 8 10