Profesor: Héctor Bahamonde, PhD.

e:hector.bahamonde@uoh.cl

w:www.hectorbahamonde.com

Curso: MLE.

TA: Gonzalo Barría.

I. Outcomes Ordenados: Ordered Logit/Probit

Los *outcomes ordenados* son aquellos donde la variable dependiente es categórica, pero representa cierto orden. Uno de los ejemplos más típicos, es el de la *escala de Likert*. La escala de Likert se utiliza para caracterizar, por ejemplo, niveles de aprobación/desaprobación de candidatos, políticas públicas, etc. La escala de Likert tiene en general cinco niveles de respuesta: *muy de acuerdo*, *acuerdo*, *neutral*, *desacuerdo*, *muy en desacuerdo*.

Lamentablemente, analistas insisten en analizar estas variables dependientes intervalares usando métodos lineales OLS (Long 1997, p. 115). Esto genera sesgos en los análisis porque asume que los intervalos numéricos entre cada categoría son constantes. Es decir, la distancia (numérica) que existe entre *muy de acuerdo* y *acuerdo* es la misma que *desacuerdo* y *muy en desacuerdo*. Y esto no es cierto. Es por esto que debemos considerar este *data generating process* distinto, y utilizar métodos de MLE. En otras palabras, no podemos asumir que el proceso ordinal sea necesariamente intervalar. El otro punto que muestra la figura (en el panel inferior) es que los errores son heteroesquedásticos. Y modelar estos errores asumiendo residuos homoesquedasticos también es un error.

Modelo Latente Una manera de motivar este modelo es vía modelos latentes. En esta motivación, tú verías una variable dependiente y_i sólo con 1's, 2's, 3's, 4's y 5's (continuando con el ejemplo de la escala de Likert). Sin embargo, el *data generating process* de y_i es un proceso *latente* $y_i^\star$ (que no ves), que de manera análoga a la motivación logit, es gatillado por umbrales (o "thresholds") τ. Formalmente,

$$y_i = \begin{Bmatrix} 1_{\text{muy de acuerdo}} & \text{si } \tau_0 = -\infty \leq y_i^{\star} < \tau_1 \\ 2_{\text{acuerdo}} & \text{si } \tau_1 \leq y_i^{\star} < \tau_2 \\ 3_{\text{neutral}} & \text{si } \tau_2 \leq y_i^{\star} < \tau_3 \\ 4_{\text{desacuerdo}} & \text{si } \tau_3 \leq y_i^{\star} < \tau_4 \\ 5_{\text{muy en desacuerdo}} & \text{si } \tau_4 \leq y_i^{\star} < \tau_5 = \infty \end{Bmatrix} \tag{1}$$

Como lo podrás notar, esta motivación es muy similar al modelo latente del modelo logit,

$$y_i = \begin{Bmatrix} 1 & \text{si } y_i^{\star} > \tau \\ 0 & \text{si } y_i^{\star} \leq \tau \end{Bmatrix} \tag{2}$$

Y de hecho, el modelo estructural ordered probit/logit te tendría que ser muy familiar,

$$y_i^{\star} = \boldsymbol{x_i}\boldsymbol{\beta} + \epsilon_i \tag{3}$$

Una manera gráfica de ver esta motivación vía modelos latentes, es a través de la siguiente figura,

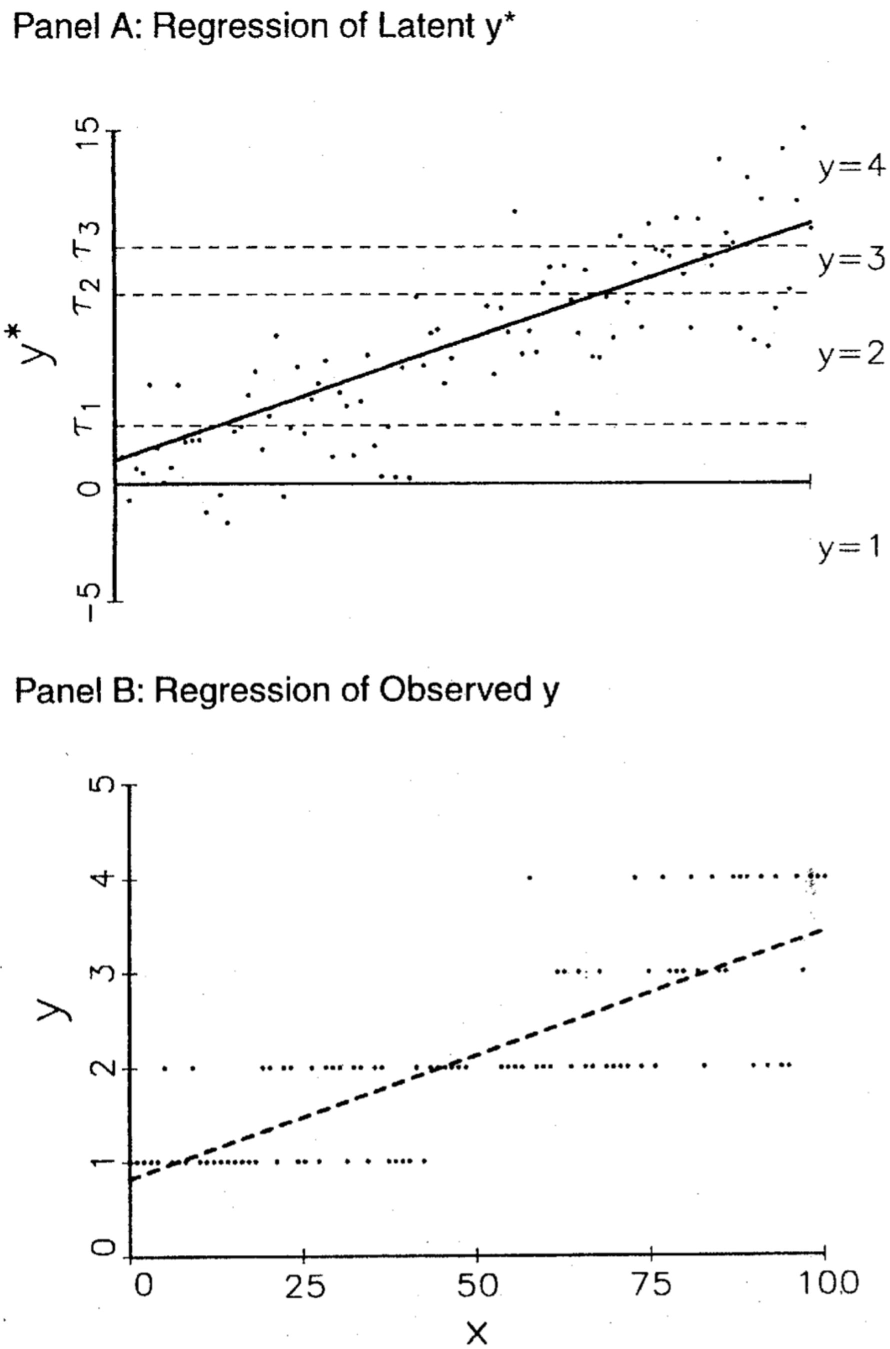


Figure 5.1. Regression of a Latent Variable y^* Compared to the Regression of the Corresponding Observed Variable y

Como sabemos, la estimación vía modelos latentes no es posible: no podemos estimar una regresión entre $y_i^\star$ y $\boldsymbol{x}$ (Long 1997, p. 117). El otro punto que muestra la figura (en el panel inferior) es que los errores son heteroesquedásticos.

Supuestos Distribucionales Debido a que esta es una extensión directa del modelo logit/probit, tenemos dos opciones de distribuciones, logit y probit. Como ya sabemos, **estas son las distribuciones de los errores**. El PDF del modelo ordered probit es formalmente,

$$\phi(e) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{\epsilon^2}{2}) \tag{4}$$

donde $\phi(e) \sim (0, 1)$.

El PDF del modelo ordered logit es formalmente definido como sigue,

$$\lambda(\epsilon) = \frac{\exp(\epsilon)}{[1 + \exp(\epsilon)]^2} \tag{5}$$

donde $\lambda(\epsilon) \sim (0, \frac{\pi^2}{3})$.

Estimacion: Probabilidades y Likelihood Continuando con el Equation 3, $\boldsymbol{x_i}\boldsymbol{\beta} + \epsilon_i$ es posible de ser calculado en términos de probabilidades de la siguiente manera,

$$\Pr(y_i = 1|\boldsymbol{x_i}) = \Pr(\tau_0 \leq \boldsymbol{x_i}\boldsymbol{\beta} + \epsilon_i < \tau_1|x_i) \tag{6}$$

donde es posible de generalizar esta notación con el resto de los valores de y_i. Y asumiendo que las observaciones son independientes entre sí, el likelihood está dado por,

$$L(\boldsymbol{\beta}, \boldsymbol{\tau}|y, \boldsymbol{X}) = \prod_{i=1}^{N} \Pr(y_i) \tag{7}$$

II. Programación

Carguemos los datos

```
p_load(foreign)
dat = read.dta("https://github.com/hbahamonde/MLE/raw/master/Datasets/nes92_ordered.dta")
```

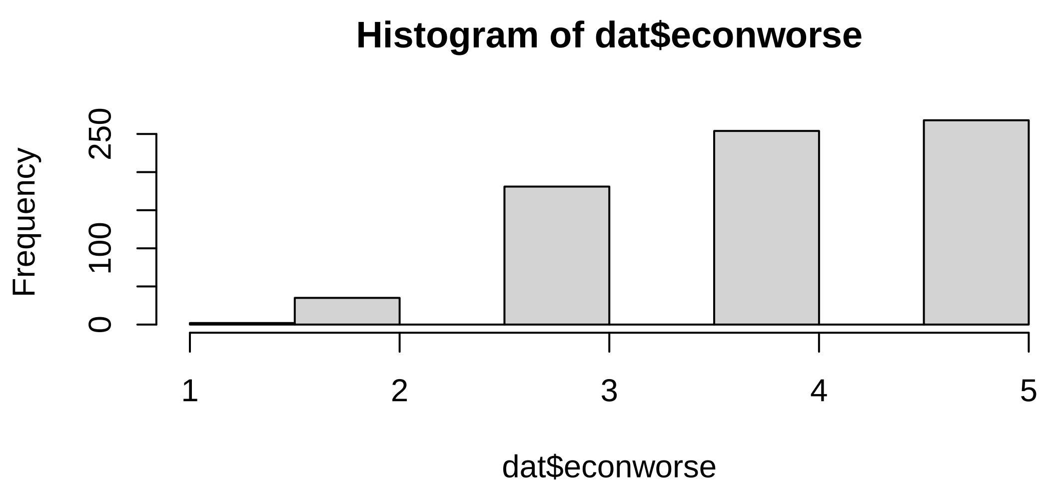
Hagamos un resumen,

```
summary(dat)

##     bushapp        ideology        bushideo      clintideo      distbushideo
##  Min.   :0.00   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :0.000
##  1st Qu.:0.00   1st Qu.:2.000   1st Qu.:4.00   1st Qu.:2.000   1st Qu.:1.000
##  Median :1.00   Median :5.000   Median :6.00   Median :3.000   Median :2.000
##  Mean   :1.25   Mean   :4.245   Mean   :5.15   Mean   :3.096   Mean   :2.106
##  3rd Qu.:2.00   3rd Qu.:6.000   3rd Qu.:6.00   3rd Qu.:4.000   3rd Qu.:3.000
##  Max.   :3.00   Max.   :7.000   Max.   :7.00   Max.   :7.000   Max.   :6.000
##  NA's   :23     NA's   :151     NA's   :72     NA's   :84      NA's   :175
##  distclintideo     econworse        oppforce     gulfwarworthit
##  Min.   :0.000   Min.   :1.000   Min.   :1.000   Min.   :0.0000
##  1st Qu.:1.000   1st Qu.:3.000   1st Qu.:3.000   1st Qu.:0.0000
##  Median :2.000   Median :4.000   Median :3.000   Median :1.0000
##  Mean   :2.068   Mean   :4.015   Mean   :2.964   Mean   :0.5835
##  3rd Qu.:3.000   3rd Qu.:5.000   3rd Qu.:3.000   3rd Qu.:1.0000
##  Max.   :6.000   Max.   :5.000   Max.   :5.000   Max.   :1.0000
##  NA's   :190     NA's   :10      NA's   :8       NA's   :37
##       pid             educyears        govtemp          union
##  Min.   :-3.0000   Min.   : 2.00   Min.   :0.000   Min.   :0.0000
##  1st Qu.:-2.0000   1st Qu.:12.00   1st Qu.:0.000   1st Qu.:0.0000
##  Median : 0.0000   Median :13.00   Median :0.000   Median :0.0000
##  Mean   :-0.1092   Mean   :13.57   Mean   :0.136   Mean   :0.1653
##  3rd Qu.: 2.0000   3rd Qu.:16.00   3rd Qu.:0.000   3rd Qu.:0.0000
##  Max.   : 3.0000   Max.   :17.00   Max.   :1.000   Max.   :1.0000
##  NA's   :8         NA's   :5
##      faminc          minority         _est_m2          _est_m1
##  Min.   :  1.50   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 21.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
##  Median : 37.50   Median :0.0000   Median :1.0000   Median :1.0000
```

```
##  Mean   : 41.93   Mean   :0.1347   Mean   :0.6787   Mean   :0.6787
##  3rd Qu.: 55.00   3rd Qu.:0.0000   3rd Qu.:1.0000   3rd Qu.:1.0000
##  Max.   :140.00   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
##  NA's   :55
```

En esta aplicación pensaremos en la variable `econworse`: *Cree usted que la economia ha empeorado?* [`Muy de acuerdo`, `de acuerdo`, `neutral`, `desacuerdo`, `muy en desacuerdo`]. Veámos cómo se ve esta variable.

```
hist(dat$econworse)
```



El paquete de `R` que usaremos se llama `polr`—éste especifica que la variable dependiente debe ser `factor`.

```
dat$econworse.f = as.factor(dat$econworse) # transforma a factor
head(dat$econworse.f) # ve como queda

## [1] 4 5 5 5 4 5
## Levels: 1 2 3 4 5
```

Ahora estimemos un `ologit` y un `oprobit`.

	Model 1	Model 2
ideology	-0.27^{***}	-0.17^{***}
	(0.05)	(0.03)
educyears	-0.09^{*}	-0.06^{*}
	(0.04)	(0.02)
faminc	-0.00	-0.00
	(0.00)	(0.00)
1|2	-8.97^{***}	-4.63^{***}
	(1.16)	(0.48)
2|3	-5.57^{***}	-3.27^{***}
	(0.62)	(0.35)
3|4	-3.44^{***}	-2.11^{***}
	(0.58)	(0.33)
4|5	-1.91^{***}	-1.17^{***}
	(0.57)	(0.33)
AIC	1353.41	1350.17
BIC	1383.68	1380.44
Log Likelihood	-669.70	-668.09
Deviance	1339.41	1336.17
Num. obs.	558	558

$^{***}p < 0.001$; $^{**}p < 0.01$; $^{*}p < 0.05$

Table 1: *Statistical models*

```
p_load(MASS)
o.logit = polr(econworse.f ~ ideology + educyears + faminc, data = dat,  method = "logistic") # o-lo
o.probit = polr(econworse.f ~ ideology + educyears + faminc, data = dat,  method = "probit") # o-pro
```

Desde ahora en adelante, prestaremos más atención a la presentación de resultados. Hagamos una tabla.

```
p_load(texreg)
texreg(list(o.logit, o.probit)) # usa "screenreg" no "texreg".
```

Ya que los resultados son (casi) siempre similares, durante el resto de la clase solo veremos el `o.logit`.

Fíjate que vemos mas interceptos, uno por cada τ. Debido a que y_i tiene cinco valores, hay cuatro τ. Esto se puede interpretar así,

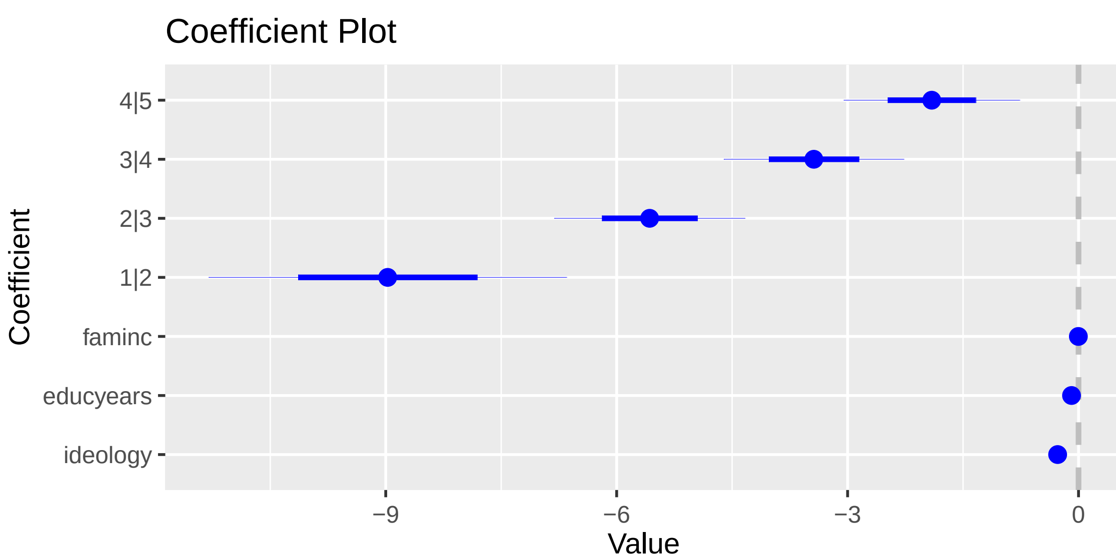
$$\begin{aligned}
\text{logit}(Pr(y_i \leq 1)) &= -8.97 - 0.27 \times \texttt{ideology}_i - 0.09 \times \texttt{educyears}_i - 0 \times \texttt{faminc}_i \\
\text{logit}(Pr(y_i \leq 2)) &= -5.57 - 0.27 \times \texttt{ideology}_i - 0.09 \times \texttt{educyears}_i - 0 \times \texttt{faminc}_i \\
\text{logit}(Pr(y_i \leq 3)) &= -3.44 - 0.27 \times \texttt{ideology}_i - 0.09 \times \texttt{educyears}_i - 0 \times \texttt{faminc}_i \\
\text{logit}(Pr(y_i \leq 4)) &= -1.91 - 0.27 \times \texttt{ideology}_i - 0.09 \times \texttt{educyears}_i - 0 \times \texttt{faminc}_i
\end{aligned} \tag{8}$$

III. Interpretación

Ahora interpretaremos el modelo.

Intervalos de Confianza Inspeccionemos los intervalos de confianza,

```
p_load(coefplot)
coefplot(o.logit)
```



El eje x del gráfico está en escala de logit, o *log-odds*. Es decir, si subo una unidad en `ideology`, esperamos que `econworse.f` suba -0.27 **en la escala logit, o log-odds** manteniendo las otras variables constantes en sus medias.

Odds Ratios Calculemos ahora los *odds ratios*.

```
exp(coef(o.logit))

##  ideology educyears    faminc
## 0.7622486 0.9131977 0.9977306
```

Esto quiere decir que cuando subo una unidad en `ideology` (i.e. me vuelvo mas derechista) es 0.76 más *posible* que encuentre la economía peor (`econworse`), manteniendo el resto de las variables constantes en sus medias. El supuesto que permite esta comparacion, i.e. de que los odds ratios se aplican a cualquier nivel de la y_i, se llama **parallel regression assumption** (Long 1997, p. 140). Por esto es que estos *odds ratios* son *proporcionales* (aplican en cualquier intervalo de `ideology`). Este supuesto es testable vía el *Brant test*.

```
p_load(brant)
brant(o.logit)

## --------------------------------------------
## Test for X2 df probability
## --------------------------------------------
## Omnibus 7.22 9 0.61
## ideology 4.74 3 0.19
## educyears 2.43 3 0.49
## faminc 1.25 3 0.74
## --------------------------------------------
##
## H0: Parallel Regression Assumption holds
```

La H_0 es que se cumple el supuesto de la regresión paralela. Si la probabilidad de la H_1 (que aparece en la tabla) es “alta”, el supuesto—probablemente—no se cumple.

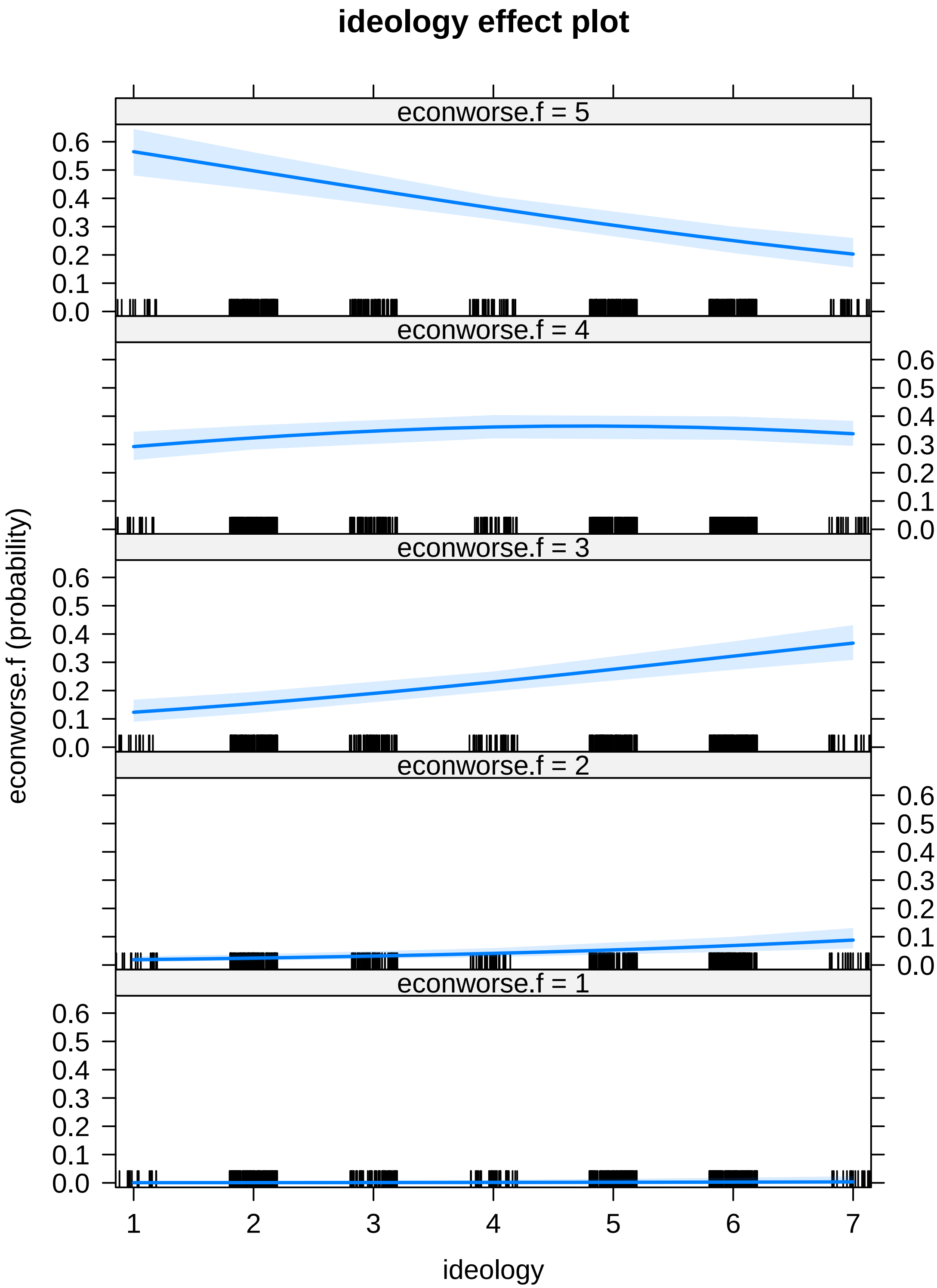
Cambios Marginales Calculemos ahora los cambios marginales. Pensemos en dos perfiles.

```
p_load(margins)
# 1
```

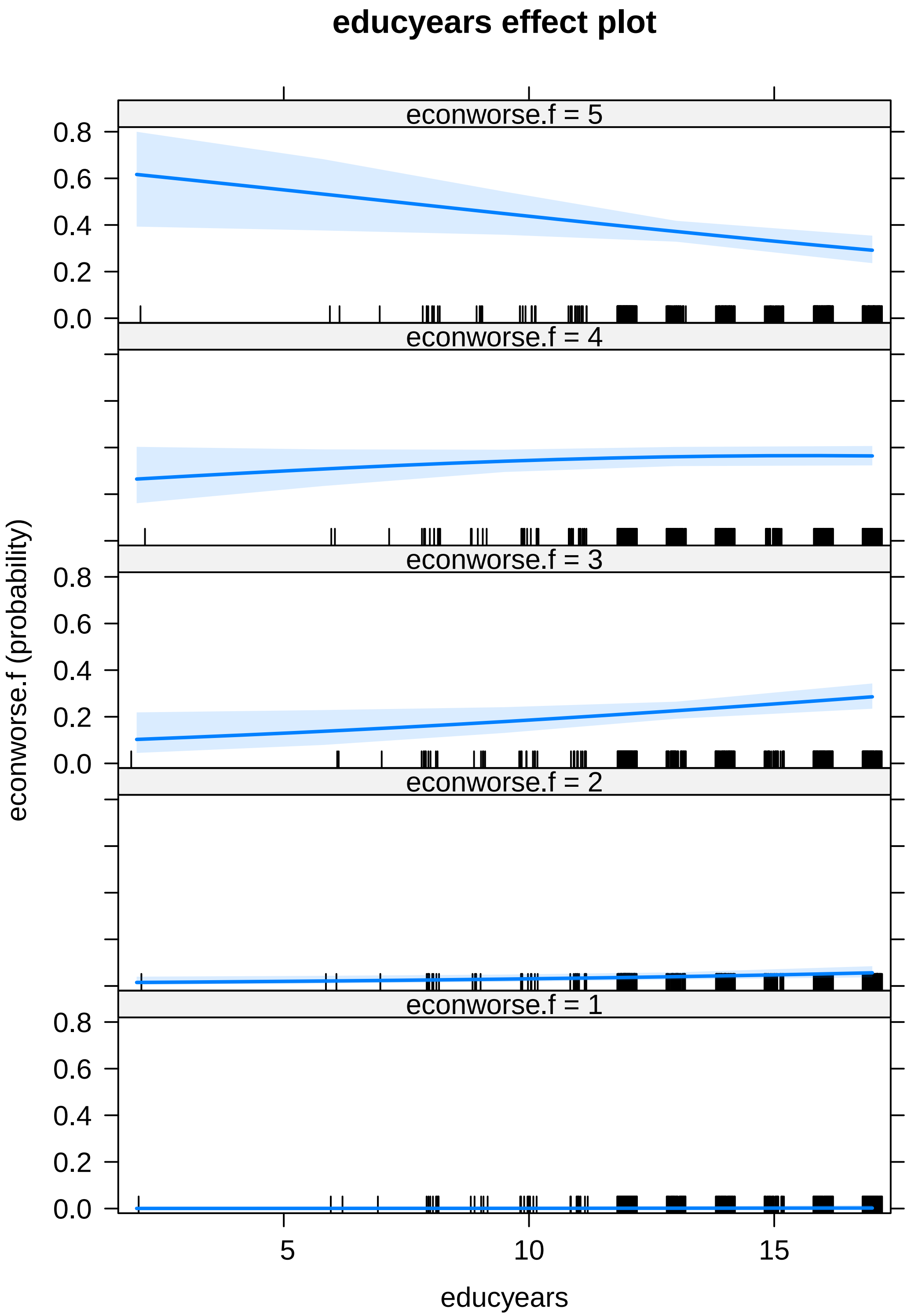
```
margins(o.logit, at = list(
  ideology = max(dat$ideology, na.rm = T), # derechista
  educyears = min(dat$educyears, na.rm = T)) # sin educ
  )

##  at(ideology) at(educyears)  ideology educyears      faminc
##             7             2 0.0003049  0.000102 0.000002552

# 2
margins(o.logit, at = list(
  ideology = min(dat$ideology, na.rm = T), # izquierdista
  educyears = min(dat$educyears, na.rm = T)) # sin educ
)

##  at(ideology) at(educyears)   ideology  educyears       faminc
##             1             2 0.00005992 0.00002004 0.0000005014
```

Predicted probabilities Calculemos ahora los *predicted probabilities*.

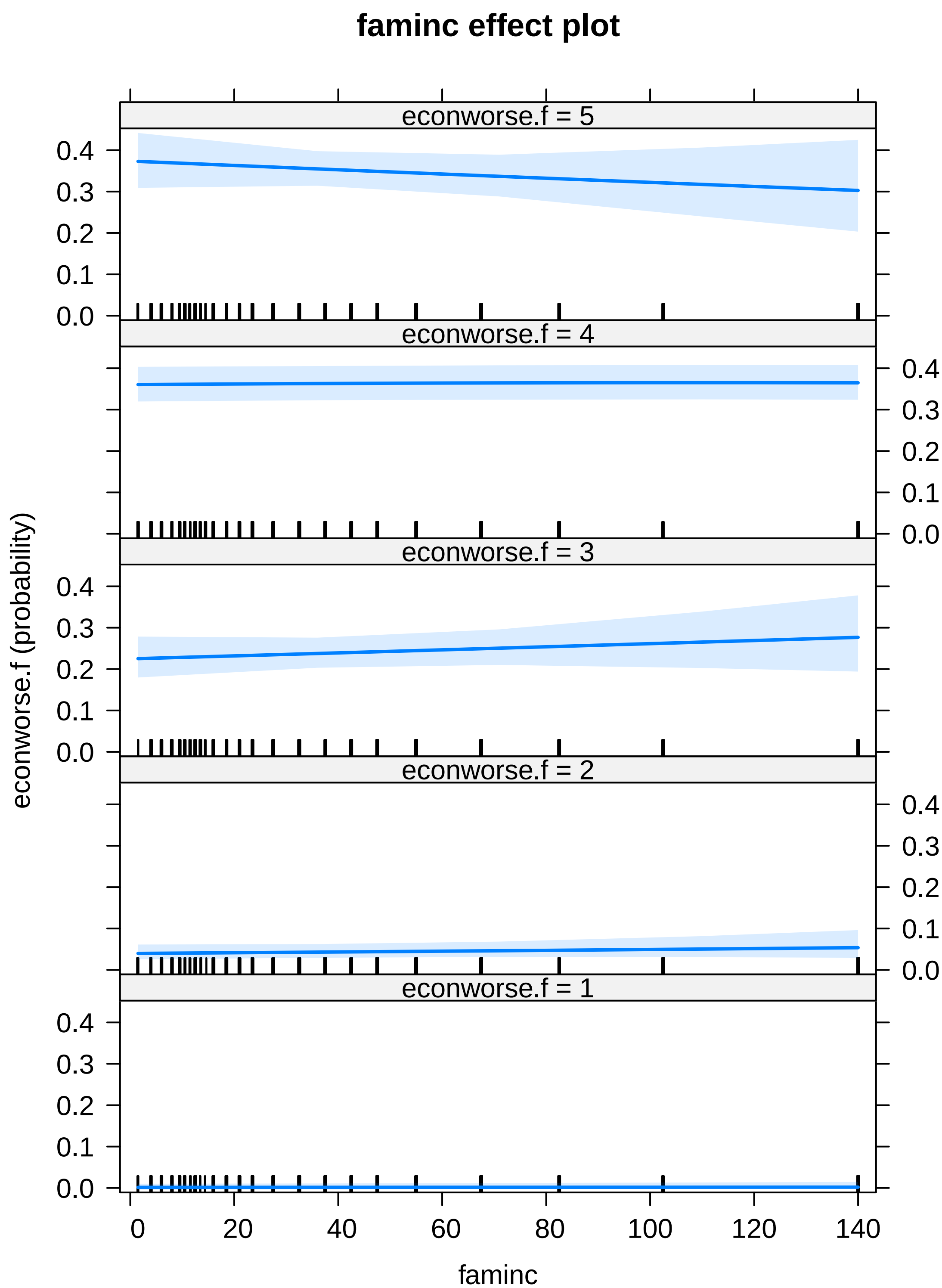
```
p_load(effects)
plot(effect("ideology", o.logit))
```


ideology effect plot
econworse.f = 5
econworse.f = 4
econworse.f = 3
econworse.f = 2
econworse.f = 1
0.6
0.5
0.4
0.3
0.2
0.1
0.0
econworse.f (probability)
1
2
3
4
5
6
7
ideology

```r
plot(effect("educyears", o.logit))
```


educyears effect plot
econworse.f = 5
econworse.f = 4
econworse.f = 3
econworse.f = 2
econworse.f = 1
0.0
0.2
0.4
0.6
0.8
econworse.f (probability)
5
10
15
educyears

```r
plot(effect("faminc", o.logit))
```


faminc effect plot
econworse.f = 5
econworse.f = 4
econworse.f = 3
econworse.f = 2
econworse.f = 1
econworse.f (probability)
0.0
0.1
0.2
0.3
0.4
0
20
40
60
80
100
120
140
faminc

```
knitr::purl('Ordered.Rnw')

## Error in parse_block(g[-1], g[1], params.src, markdown_mode):  Duplicate chunk label
'setup', which has been used for the chunk:
## if (!require("pacman")) install.packages("pacman"); library(pacman)
## p_load(knitr)
## set.seed(2020)
## options(scipen=9999999)

Stangle('Ordered.Rnw')

## Writing to file Ordered.R

## Error in match.arg(options$results, c("verbatim", "tex", "hide")):  'arg' should
be one of "verbatim", "tex", "hide"
```