CSSS 510: Lab 5

Ordered Probit

2017-11-17

0. Agenda

1. Deriving a likelihood function for the ordered probit model
2. Fitting an ordered probit model using optim() and glm()
3. Interpreting the results
4. Simulating predicted values and confidence intervals

1. Deriving a likelihood function for ordered probit

Recall from lecture the ordered probit model:

$$\Pr(y_i = j|\boldsymbol{x}_i) = \int_{\tau_{j-1}}^{\tau_j} \text{Normal}(\boldsymbol{x_i}\beta, 1)\text{d}\boldsymbol{x_i}\beta$$

We are saying that the probability that y_i is in category j is equal to the CDF of the standard normal distribution evaluated at $\boldsymbol{x_i}\beta$ between cutpoints τ_j and τ_{j-1}

How does this model differ from the others we've covered so far in the course?

1. Deriving a likelihood function for ordered probit

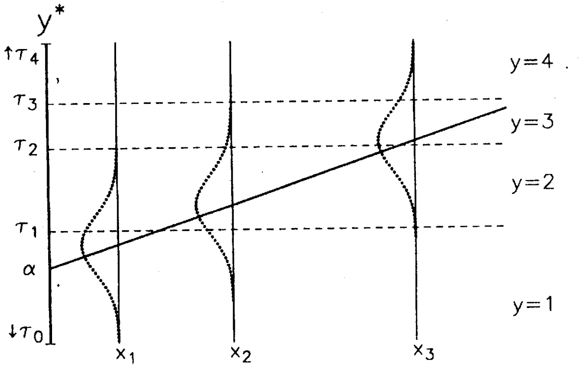


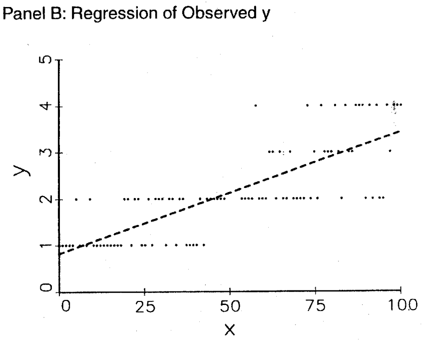
Figure 5.2. Distribution of y^* Given x for the Ordered Regression Model

1. Deriving a likelihood function for ordered probit

If we merely plot the observed values of y then we would obtain something like the following.

But, this runs into the same problems we found when using linear regression to fit a dichotomous outcome variable.

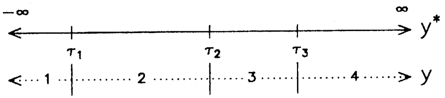
Furthermore, this includes the implicity assumption that the intervals between adjacent categories are equal.



1. Deriving a likelihood function for ordered probit

Instead, we refer back to the latent variable framework, which produces a similar relationship between our covariates and y^* as in logit but now we have more than two categories.

$$y_i = j \quad \text{if} \quad \tau_{j-1} \leq y_i^* < \tau_j \quad \text{for} \quad j = 1 \text{ to } M$$



τ are thresholds or cutpoints. y_i is the observed outcome variable. m is the specific outcome observed. y_i^* is the latent variable. M is the number of categories.

1. Deriving a likelihood function for ordered probit

$$y^* = \boldsymbol{x_i}\beta + \epsilon$$

$$y_i = j \quad \text{if} \quad \tau_{j-1} \leq y_i^* < \tau_j$$

$$\text{for} \quad j = 1 \text{ to } M$$

In a four category model, we have the following:

$$y_i = \begin{cases} 1 \Rightarrow \text{SD}, & \text{if } \tau_0 = -\infty \leq y_i^* < \tau_1 \\ 2 \Rightarrow \text{D}, & \text{if } \tau_1 \leq y_i^* < \tau_2 \\ 3 \Rightarrow \text{A}, & \text{if } \tau_2 \leq y_i^* < \tau_3 \\ 4 \Rightarrow \text{SA}, & \text{if } \tau_3 \leq y_i^* < \tau_4 = \infty \end{cases}$$

1. Deriving a likelihood function for ordered probit

Recall that in logit, we assume the errors of the latent variable follow a standard logistic distribution.

For probit and ordered probit, we assume the errors follow a standard normal distribution.

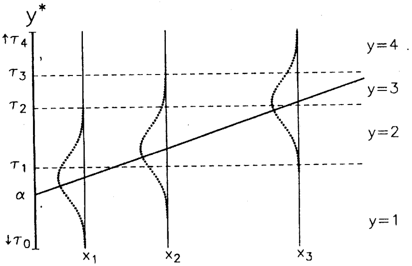


Figure 5.2. Distribution of y^* Given x for the Ordered Regression Model

$$\begin{aligned}
\Pr(y_i = 1|\boldsymbol{x_i}) &= \Pr(\tau_0 \leq y^* < \tau_1|\boldsymbol{x_i}) \\
&= \Pr(\tau_0 \leq \boldsymbol{x_i}\beta + \epsilon_i < \tau_1|\boldsymbol{x_i}) \\
&= \Pr(\tau_0 - \boldsymbol{x_i}\beta \leq \epsilon_i < \tau_1 - \boldsymbol{x_i}\beta|\boldsymbol{x_i}) \\
&= \Pr(\epsilon_i < \tau_1 - \boldsymbol{x_i}\beta|\boldsymbol{x_i}) - \Pr(\epsilon_i \leq \tau_0 - \boldsymbol{x_i}\beta|\boldsymbol{x_i}) \\
&= F(\tau_1 - \boldsymbol{x_i}\beta) - F(\tau_0 - \boldsymbol{x_i}\beta) \\
\Pr(y_i = j|\boldsymbol{x_i}) &= F(\tau_j - \boldsymbol{x_i}\beta) - F(\tau_{j-1} - \boldsymbol{x_i}\beta)
\end{aligned}$$

1. Deriving a likelihood function for ordered probit

For the ordered probit this becomes

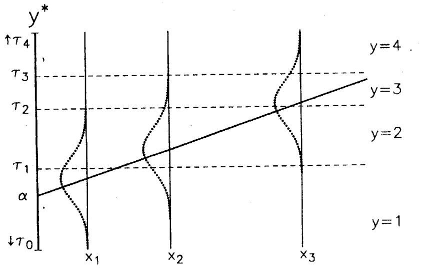


Figure 5.2. Distribution of y^* Given x for the Ordered Regression Model

$$\Pr(y_i = 1|\boldsymbol{x_i}) = \Phi(\tau_1 - \alpha - \boldsymbol{x_i}\beta)$$
$$\Pr(y_i = 2|\boldsymbol{x_i}) = \Phi(\tau_2 - \alpha - \boldsymbol{x_i}\beta) - \Phi(\tau_1 - \alpha - \boldsymbol{x_i}\beta)$$
$$\Pr(y_i = 3|\boldsymbol{x_i}) = \Phi(\tau_3 - \alpha - \boldsymbol{x_i}\beta) - \Phi(\tau_2 - \alpha - \boldsymbol{x_i}\beta)$$
$$\Pr(y_i = 4|\boldsymbol{x_i}) = 1 - \Phi(\tau_3 - \alpha - \boldsymbol{x_i}\beta)$$

1. Deriving a likelihood function for ordered probit

To identify the model, we commonly make one of two assumptions:

1. Assume that $\tau_1 = 0$. This is also the identifying assumption of logit and probit. optim() uses this.
2. Assume that $\alpha = 0$. polr() uses this.

The likelihood function for ordered probit finds the β and τ that make the observed data most likely.

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\tau}|\mathbf{y}, \mathbf{X}) = \prod_{i=1}^{n} \left\{ \prod_{j=1}^{m} [\Phi(\tau_j|\boldsymbol{x_i}\beta], 1) - \Phi(\tau_{j-1}|\boldsymbol{x_i}\beta, 1)]^{y_{ij}} \right\}$$

$$\mathcal{L}(\boldsymbol{\beta}, \boldsymbol{\tau}|\mathbf{y},\ \mathbf{X}) = \sum_{i=i}^{n} \sum_{j=1}^{m} y_{ij} \log[\Phi(\tau_j|\boldsymbol{x_i}\beta, 1) - \Phi(\tau_{j-1}|\boldsymbol{x_i}\beta, 1)]$$

We estimate β and τ that maximizes the likelihood that y_{ij} falls into category j. All other categories $\neq$ j are irrelevant.

2. Fitting an ordered probit model using optim() and glm()

```r
rm(list=ls())

## Likelihood for 4 category ordered probit
llk.oprobit4 <- function(param, x, y) {
  # preliminaries
  os <- rep(1, nrow(x))
  x <- cbind(os, x)
  b <- param[1:ncol(x)]
  t2 <- param[(ncol(x)+1)]
  t3 <- param[(ncol(x)+2)]

  # probabilities and penalty function
  xb <- x%*%b
  p1 <- log(pnorm(-xb))
  if (t2<=0)  p2 <- -(abs(t2)*10000)    # penalty function to keep t2>0
  else p2 <- log(pnorm(t2-xb)-pnorm(-xb))
  if (t3<=t2) p3 <- -((t2-t3)*10000)    # penalty to keep t3>t2
  else p3 <- log(pnorm(t3-xb)-pnorm(t2-xb))
  p4 <- log(1-pnorm(t3-xb))

  # -1 * log likelihood (optim is a minimizer)
  -sum(cbind(y==1,y==2,y==3,y==4) * cbind(p1,p2,p3,p4))
}
```

2. Fitting an ordered probit model using optim() and glm()

```
## Load libraries
library(MASS)
library(simcf)
library(tile)
```

```
## Loading required package: grid
```

```
library(RColorBrewer)

## Nice colors
brewer <- brewer.pal(9, "Set1")
red <- brewer[1]
blue <- brewer[2]
green <- brewer[3]
purple <- brewer[4]
orange <- brewer[5]
nicegray <- "gray45"

## Load data
workmom <- read.csv("ordwarm2.csv", header=TRUE, sep=",")
workmom77 <- workmom[workmom$yr89==0, ]
workmom89 <- workmom[workmom$yr89==1, ]
```

2. Fitting an ordered probit model using optim() and glm()

```
## Data from 1977, 1989 GSS:  Attitudes towards working mothers
y <- workmom77$warm    # Mother can have warm feelings towards child?

x <- cbind(workmom77$male, workmom77$white, workmom77$age,
           workmom77$ed, workmom77$prst)
## male respondent; white resp; age of resp;
## years of education of respondent;
## prestige of respondent's occupation (% considering prestigious)

# Model specification (for polr, simcf)
model <- warm ~ male + white + age + ed + prst

# Use optim directly to get MLE
ls.result <- lm(model, data=workmom77)   # use ls estimates as starting values
stval <- c(coef(ls.result),1,2)          # initial guesses
oprobit.res77 <- optim(stval, llk.oprobit4, method="BFGS", x=x, y=y, hessian=T)
pe77 <- oprobit.res77$par                # point estimates
vc77 <- solve(oprobit.res77$hessian)     # var-cov matrix
se77 <- sqrt(diag(vc77))                 # standard errors
ll77 <- -oprobit.res77$value             # likelihood at maximum
```

2. Fitting an ordered probit model using optim() and glm()

```
pe77
```

```
##  (Intercept)         male        white          age           ed
##  1.321689587 -0.397228916 -0.238335818 -0.011703265  0.044989779
##         prst
##  0.002116029  1.016421525  2.077970893
```

```
se77
```

```
## (Intercept)        male       white         age          ed        prst
## 0.179052337 0.058429471 0.091086552 0.001915842 0.012076026 0.002586886
##
## 0.041094751 0.054430991
```

```
ll77
```

```
## [1] -1758.824
```

4. Simulating predicted values and confidence intervals

```r
# Use MASS::polr to do ordered probit
workmom77$warmf <- factor(workmom77$warm, labels=c("Strongly Disagree",
                                                   "Disagree",
                                                   "Agree",
                                                   "Strongly Agree"))
glm.res77 <- polr(warmf ~ male + white + age + ed + prst, data=workmom77,
                  method="probit", na.action=na.omit)

# Simulate parameters from predictive distributions
sims <- 10000
simbetas <- mvrnorm(sims, pe77, vc77)       # draw parameters, using MASS::mvrnorm

# Create example counterfactuals
xhyp <- cfMake(model, workmom77, nscen=10)

xhyp <- cfName(xhyp,"Male", scen=1)
xhyp <- cfChange(xhyp, "male", x=1, xpre=0, scen=1)

xhyp <- cfName(xhyp, "Female", scen=2)
xhyp <- cfChange(xhyp, "male", x=0, xpre=1, scen=2)

xhyp <- cfName(xhyp, "Nonwhite", scen=3)
xhyp <- cfChange(xhyp, "white", x=0, xpre=1, scen=3)

xhyp <- cfName(xhyp, "White", scen=4)
xhyp <- cfChange(xhyp, "white", x=1, xpre=0, scen=4)
```

4. Simulating predicted values and confidence intervals

```
xhyp <- cfName(xhyp, "Age + 1sd = 61", scen=5)
xhyp <- cfChange(xhyp, "age",
                 x=mean(na.omit(workmom77$age))+sd(na.omit(workmom77$age)),
                 xpre=mean(na.omit(workmom77$age)),
                 scen=5)

xhyp <- cfName(xhyp, "Age - 1sd = 28", scen=6)
xhyp <- cfChange(xhyp, "age",
                 x=mean(na.omit(workmom77$age))-sd(na.omit(workmom77$age)),
                 xpre=mean(na.omit(workmom77$age)),
                 scen=6)

xhyp <- cfName(xhyp, "High School Grad", scen=7)
xhyp <- cfChange(xhyp, "ed", x=12, xpre=mean(na.omit(workmom77$ed)), scen=7)

xhyp <- cfName(xhyp,"College Grad", scen=8)
xhyp <- cfChange(xhyp, "ed", x=16, xpre=mean(na.omit(workmom77$ed)), scen=8)

xhyp <- cfName(xhyp,"High Prestige Job (+1 sd)", scen=9)
xhyp <- cfChange(xhyp, "prst",
                 x=mean(na.omit(workmom77$prst))+sd(na.omit(workmom77$prst)),
                 xpre=mean(na.omit(workmom77$prst)),
                 scen=9)

xhyp <- cfName(xhyp,"Low Prestige Job (-1 sd)", scen=10)
xhyp <- cfChange(xhyp, "prst",
                 x=mean(na.omit(workmom77$prst))-sd(na.omit(workmom77$prst)),
                 xpre=mean(na.omit(workmom77$prst)),
                 scen=10)
```

4. Simulating predicted values and confidence intervals

```
# Simulate expected probabilities (all four categories)
oprobit.ev77 <- oprobitsimev(xhyp, simbetas, cat=4)

# Simulate first differences (all four categories)
oprobit.fd77 <- oprobitsimfd(xhyp, simbetas, cat=4)

# Simulate relative risks (all four categories)
oprobit.rr77 <- oprobitsimrr(xhyp, simbetas, cat=4)

# Plot predicted probabilities for all four categories, sorted by size
sorted <- order(oprobit.ev77$pe[,1])
scenNames <- row.names(xhyp$x)
```

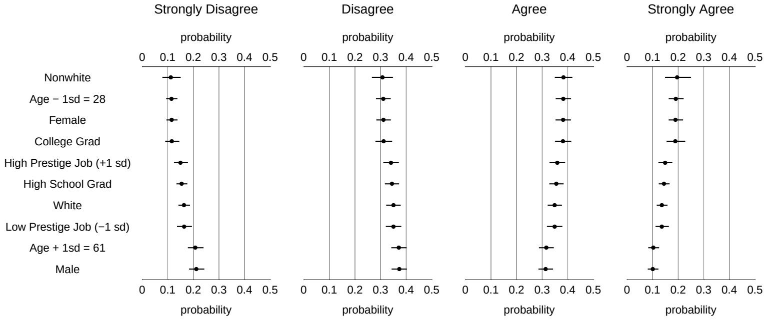
4. Simulating predicted values and confidence intervals

```
trace1 <- ropeladder(x = oprobit.ev77$pe[sorted,1],
                     lower = oprobit.ev77$lower[sorted,1],
                     upper = oprobit.ev77$upper[sorted,1],
                     labels = scenNames[sorted],
                     size=0.5,
                     lex=1.5,
                     lineend="square",
                     plot=1
)

trace2 <- ropeladder(x = oprobit.ev77$pe[sorted,2],
                     lower = oprobit.ev77$lower[sorted,2],
                     upper = oprobit.ev77$upper[sorted,2],
                     size=0.5,
                     lex=1.5,
                     lineend="square",
                     plot=2
)

trace3 <- ropeladder(x = oprobit.ev77$pe[sorted,3],
                     lower = oprobit.ev77$lower[sorted,3],
                     upper = oprobit.ev77$upper[sorted,3],
                     size=0.5,
                     lex=1.5,
                     lineend="square",
                     plot=3
)
```

4. Simulating predicted values and confidence intervals

```
trace4 <- ropeladder(x = oprobit.ev77$pe[sorted,4],
                     lower = oprobit.ev77$lower[sorted,4],
                     upper = oprobit.ev77$upper[sorted,4],
                     size=0.5,
                     lex=1.5,
                     lineend="square",
                     plot=4
)

file <- "mothers4catEV"
tile(trace1, trace2, trace3, trace4,
     limits = c(0,0.5),
     gridlines = list(type="xt"),
     topaxis=list(add=TRUE, at=c(0,0.1,0.2,0.3,0.4,0.5)),
     xaxistitle=list(labels="probability"),
     topaxistitle=list(labels="probability"),
     plottitle=list(labels=c("Strongly Disagree", "Disagree",
                             "Agree", "Strongly Agree")),
     width=list(spacer=3),
     height = list(plottitle=3,xaxistitle=3.5,topaxistitle=3.5),
     output=list(outfile=file, width=12)
)
```

4. Simulating predicted values and confidence intervals



4. Simulating predicted values and confidence intervals

```
## Re-simulate, now collapsing presentation to two categories
## ("SD/D" vs "SA/A")

## Simulate expected probabilities (all four categories)
oprobit.ev77c <- oprobitsimev(xhyp, simbetas, cat=4,
                              recode=list(c(1,2), c(3,4)) )

## Simulate first differences (all four categories)
oprobit.fd77c <- oprobitsimfd(xhyp, simbetas, cat=4,
                              recode=list(c(1,2), c(3,4)) )

## Simulate relative risks (all four categories)
oprobit.rr77c <- oprobitsimrr(xhyp, simbetas, cat=4,
                              recode=list(c(1,2), c(3,4)) )
```

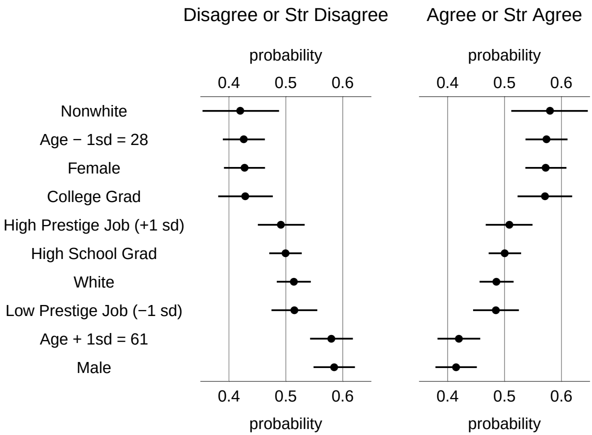
4. Simulating predicted values and confidence intervals

```
## Make a new rl plot, EV of Dd vs aA
trace1b <- ropeladder(x = oprobit.ev77c$pe[sorted,1],
                      lower = oprobit.ev77c$lower[sorted,1],
                      upper = oprobit.ev77c$upper[sorted,1],
                      labels = scenNames[sorted],
                      size=0.65,
                      lex=1.75,
                      lineend="square",
                      plot=1
)

trace2b <- ropeladder(x = oprobit.ev77c$pe[sorted,2],
                      lower = oprobit.ev77c$lower[sorted,2],
                      upper = oprobit.ev77c$upper[sorted,2],
                      size=0.65,
                      lex=1.75,
                      lineend="square",
                      plot=2
)
```

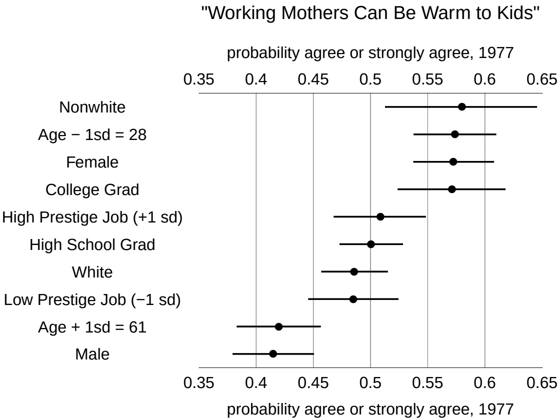
4. Simulating predicted values and confidence intervals

```r
file <- "mothers2catEV"
tile(trace1b, trace2b,
     limits = c(0.35,0.65),
     gridlines = list(type="xt"),
     xaxis=list(at=c(0.4, 0.5, 0.6)),
     topaxis=list(add=TRUE, at=c(0.4, 0.5, 0.6)),
     xaxistitle=list(labels="probability"),
     topaxistitle=list(labels="probability"),
     plottitle=list(labels=c("Disagree or Str Disagree",
                             "Agree or Str Agree")),
     width=list(spacer=3),
     height = list(plottitle=3,xaxistitle=3.5,topaxistitle=3.5),
     output=list(outfile=file, width=7)
)

## Revise traces and plot to show only "SA/A"
trace2b$entryheight <- 0.2
trace2b$plot <- 1
trace2b$labels <- trace1b$labels
file <- "mothers1catEV"
tile(trace2b,
     limits = c(0.35,0.65),
     gridlines = list(type="xt"),
     xaxis=list(at=seq(0.35, 0.65, 0.05)),
     topaxis=list(add=TRUE, at=seq(0.35, 0.65, 0.05)),
     xaxistitle=list(labels="probability agree or strongly agree, 1977"),
     topaxistitle=list(labels="probability agree or strongly agree, 1977"),
     plottitle=list(labels="\"Working Mothers Can Be Warm to Kids\""),
     width=list(plot=2.5),
     height = list(plottitle=3,xaxistitle=3.5,topaxistitle=3.5),
     output=list(outfile=file, width=7)
)
```

4. Simulating predicted values and confidence intervals



4. Simulating predicted values and confidence intervals



4. Simulating predicted values and confidence intervals

```
## Now estimate 1989 model
y <- workmom89$warm    # Mother can have warm feelings towards child?

x <- cbind(workmom89$male, workmom89$white, workmom89$age,
           workmom89$ed, workmom89$prst)

# Use optim directly to get MLE
ls.result <- lm(model, data=workmom89)   # use ls estimates as starting values
stval <- c(coef(ls.result),1,2)          # initial guesses
oprobit.res89 <- optim(stval, llk.oprobit4, method="BFGS", x=x, y=y, hessian=TRUE)
pe89 <- oprobit.res89$par                # point estimates
vc89 <- solve(oprobit.res89$hessian)     # var-cov matrix
se89 <- sqrt(diag(vc89))                 # standard errors
ll89 <- -oprobit.res89$value             # likelihood at maximum

simbetas89 <- mvrnorm(sims, pe89, vc89)       # draw parameters, using MASS::mvrnorm
```

4. Simulating predicted values and confidence intervals

```
# Create example counterfactuals -- for diffs
xhyp <- cfMake(model, workmom77, nscen=5)

xhyp <- cfName(xhyp, "Female (Male)", scen=1)
xhyp <- cfChange(xhyp, "male", x=0, xpre=1, scen=1)

xhyp <- cfName(xhyp, "Nonwhite (White)", scen=2)
xhyp <- cfChange(xhyp, "white", x=0, xpre=1, scen=2)

xhyp <- cfName(xhyp, "28 Year Olds (61)", scen=3)
xhyp <- cfChange(xhyp, "age",
                 x=mean(na.omit(workmom77$age))-sd(na.omit(workmom77$age)),
                 xpre=mean(na.omit(workmom77$age)),
                 scen=3)

xhyp <- cfName(xhyp,"College Grad (High School)", scen=4)
xhyp <- cfChange(xhyp, "ed", x=16, xpre=12, scen=4)

xhyp <- cfName(xhyp,"High Prestige Job (Low)", scen=5)
xhyp <- cfChange(xhyp, "prst",
                 x=mean(na.omit(workmom77$prst))+sd(na.omit(workmom77$prst)),
                 xpre=mean(na.omit(workmom77$prst)) - sd(na.omit(workmom77$prst)),
                 scen=5)
```

4. Simulating predicted values and confidence intervals

```r
# Simulate expected probabilities (all four categories)
oprobit.ev77 <- oprobitsimev(xhyp, simbetas, cat=4)

# Simulate first differences (all four categories)
oprobit.fd77 <- oprobitsimfd(xhyp, simbetas, cat=4)

# Simulate relative risks (all four categories)
oprobit.rr77 <- oprobitsimrr(xhyp, simbetas, cat=4)


# Re-simulate, now collapsing presentation to two categories ("SD/D" vs "SA/A")

# Simulate expected probabilities (all four categories)
oprobit.ev77c <- oprobitsimev(xhyp, simbetas, cat=4,
                              recode=list(c(1,2), c(3,4)) )

# Simulate first differences (all four categories)
oprobit.fd77c <- oprobitsimfd(xhyp, simbetas, cat=4,
                              recode=list(c(1,2), c(3,4)) )

# Simulate relative risks (all four categories)
oprobit.rr77c <- oprobitsimrr(xhyp, simbetas, cat=4,
                              recode=list(c(1,2), c(3,4)) )

# Simulate first differences (all four categories)
oprobit.fd89c <- oprobitsimfd(xhyp, simbetas89, cat=4,
                              recode=list(c(1,2), c(3,4)) )

# Simulate relative risks (all four categories)
oprobit.rr89c <- oprobitsimrr(xhyp, simbetas89, cat=4,
                              recode=list(c(1,2), c(3,4)) )
```

4. Simulating predicted values and confidence intervals

```
# Make a new ropeladder plot, showing just change in probability of any agreement
sortedc <- rev(order(oprobit.fd77c$pe[,2]))
scenNames <- row.names(xhyp$x)

trace1c <- ropeladder(x = oprobit.fd77c$pe[sortedc,2],
                      lower = oprobit.fd77c$lower[sortedc,2],
                      upper = oprobit.fd77c$upper[sortedc,2],
                      labels = scenNames[sortedc],
                      sublabels="1977",
                      sublabelsyoffset=0.04,
                      col=orange,
                      size=0.65,
                      lex=1.75,
                      lineend="square",
                      plot=1
)

trace2c <- ropeladder(x = oprobit.fd89c$pe[sortedc,2],
                      lower = oprobit.fd89c$lower[sortedc,2],
                      upper = oprobit.fd89c$upper[sortedc,2],
                      labels = scenNames[sortedc],
                      sublabels = "1989",
                      sublabelsyoffset = -0.04,
                      col=blue,
                      size=0.65,
                      lex=1.75,
                      lineend="square",
                      entryheight=0.40,
                      subentryheight=.8,
                      plot=1
)
```

4. Simulating predicted values and confidence intervals

```
sigMark1 <- oprobit.fd77c$pe[sortedc,2]
is.na(sigMark1) <- (oprobit.fd77c$lower[sortedc,2]>0)
traceSig1 <- ropeladder(x=sigMark1,
                        col="white",
                        group=1,
                        plot=1)

sigMark2 <- oprobit.fd89c$pe[sortedc,2]
is.na(sigMark2) <- (oprobit.fd89c$lower[sortedc,2]>0)
traceSig2 <- ropeladder(x=sigMark2,
                        col="white",
                        group=2,
                        plot=1)

vertmark <- linesTile(x=c(0,0), y=c(0,1), plot=1)

file <- "mothersFD7789"
tile(trace1c, trace2c, vertmark, traceSig1, traceSig2,
     limits=c(-0.05,0.25),
     gridlines=list(type="xt"),
     topaxis=list(add=TRUE, at=seq(from=0, to=0.2, by=0.05),
                  labels=c("0%", "+5%", "+10%", "+15%", "+20%")),
     xaxis=list(at=seq(from=0, to=0.2, by=0.05), labels=c("0%", "+5%", "+10%", "+15%", "+20%")),
     xaxistitle=list(labels="difference in probability agree or strongly agree"),
     topaxistitle=list(labels="difference in probability agree or strongly agree"),
     plottitle=list(labels="\"Working Mothers Can Be Warm to Kids\""),
     width=list(plot=2),
     height=list(plottitle=3,xaxistitle=3.5,topaxistitle=3.5),
     output=list(outfile=file, width=6.75)
)
```

4. Simulating predicted values and confidence intervals