Gov 2001: Section 7

March 12, 2013

Outline

Outline

The Data: Political Assassinations

Taken from Olken and Jones (2009), "Hit or Miss? The Effect of Assassinations on Institutions and War", *American Economic Journal: Macroeconomics*.

Dataframe is called as and contains information on assassination attempts, success or failure, and various covariates.

```
> as[as$country == ``United States'' & as$year == ``1975'',]
      country year leadername age tenure attempt
United States 1975       Ford  62    510    TRUE
      survived result dem_score civil_war war
             1     24        10         0   0
         pop  energy solo weapon
      215973 2208506    1    gun
```

Observations are country-year-leaders, so some country-years have multiple observations.

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The model:

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The model:

1. $Y_i \sim f_{\text{bern}}(y_i|\pi_i)$.

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The model:

1. $Y_i \sim f_{\text{bern}}(y_i|\pi_i)$.
2. $\pi_i = \Phi(X_i\beta)$ where Φ is the CDF

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The model:

1. $Y_i \sim f_{\text{bern}}(y_i|\pi_i)$.
2. $\pi_i = \Phi(X_i\beta)$ where Φ is the CDF of the standard normal distribution.

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The model:

1. $Y_i \sim f_{\text{bern}}(y_i|\pi_i)$.
2. $\pi_i = \Phi(X_i\beta)$ where Φ is the CDF of the standard normal distribution.
3. Y_i and Y_j are independent for all $i \neq j$.

The Probit Model

Let's try to predict assassination attempts with some of our covariates.

The model:

1. $Y_i \sim f_{\text{bern}}(y_i|\pi_i)$.
2. $\pi_i = \Phi(X_i\beta)$ where Φ is the CDF of the standard normal distribution.
3. Y_i and Y_j are independent for all $i \neq j$.

Like all CDF's, Φ has range 0 to 1, so it puts our π_i on the right scale.

$$\Phi(z) = \int_{-\infty}^{z} \frac{1}{\sqrt{2\pi}} \exp(\frac{z^2}{2})dz.$$

The Probit Model

We can then derive the log-likelihood for β:

$$\begin{aligned} L(\beta|\mathbf{y}) &\propto \prod_{i=1}^{n} f_{\text{bern}}(y_i|\pi_i) \\ &= \prod_{i=1}^{n} (\pi_i)^{y_i}(1-\pi_i)^{(1-y_i)} \end{aligned}$$

The Probit Model

We can then derive the log-likelihood for β:

$$\begin{aligned} L(\beta|\mathbf{y}) &\propto \prod_{i=1}^{n} f_{\text{bern}}(y_i|\pi_i) \\ &= \prod_{i=1}^{n} (\pi_i)^{y_i}(1-\pi_i)^{(1-y_i)} \end{aligned}$$

Therefore:

$$\begin{aligned} \ln L(\beta|\mathbf{y}) &\propto \sum_{i=1}^{n} y_i \ln(\pi_i) + (1-y_i)\ln(1-\pi_i) \\ &= \sum_{i=1}^{n} y_i \ln(\Phi(X_i\beta)) + (1-y_i)\ln(1-\Phi(X_i\beta)) \end{aligned}$$

Implementing the Likelihood

Here's one approach:

Implementing the Likelihood

Here's one approach:

```
ll.probit <- function(beta, y=y, X=X){
  if(sum(X[,1]) != nrow(X)) X <- cbind(1,X)
  phi <- pnorm(X%*%beta, log = TRUE)
  opp.phi <- pnorm(X%*%beta, log = TRUE, lower.tail = FALSE)
  logl <- sum(y*phi + (1-y)*opp.phi)
  return(logl)
}
```

Implementing the Likelihood

Here's one approach:

```
ll.probit <- function(beta, y=y, X=X){
  if(sum(X[,1]) != nrow(X)) X <- cbind(1,X)
  phi <- pnorm(X%*%beta, log = TRUE)
  opp.phi <- pnorm(X%*%beta, log = TRUE, lower.tail = FALSE)
  logl <- sum(y*phi + (1-y)*opp.phi)
  return(logl)
}
```

What's in our model?

We wish to predict assassination attempts for country-year-leaders.

- `tenure`: number of days in office
- `age`: age of leader, in years
- `dem_score`: polity score, -10 to 10
- `civil_war`: is there currently a civil war?
- `war`: is country in an international conflict?
- `pop`: the country's population, in thousands
- `energy`: energy usage

Maximization

```
y <- as$attempt
X <- as[,c("tenure","age","dem_score","civil_war",
           "war","pop","energy")]
```

Maximization

```
y <- as$attempt
X <- as[,c("tenure","age","dem_score","civil_war",
           "war","pop","energy")]

yX <- na.omit(cbind(y,X))
y <- yX[,1]; X <- yX[,-1]
```

Maximization

```
y <- as$attempt
X <- as[,c("tenure","age","dem_score","civil_war",
           "war","pop","energy")]

yX <- na.omit(cbind(y,X))
y <- yX[,1]; X <- yX[,-1]

X$tenure <- (X$tenure-mean(X$tenure))/sd(X$tenure)
X$pop <- (X$pop-mean(X$pop))/sd(X$pop)
X$energy <- (X$energy-mean(X$energy))/sd(X$energy)
```

Maximization

```
y <- as$attempt
X <- as[,c("tenure","age","dem_score","civil_war",
           "war","pop","energy")]

yX <- na.omit(cbind(y,X))
y <- yX[,1]; X <- yX[,-1]

X$tenure <- (X$tenure-mean(X$tenure))/sd(X$tenure)
X$pop <- (X$pop-mean(X$pop))/sd(X$pop)
X$energy <- (X$energy-mean(X$energy))/sd(X$energy)

opt <- optim(par = rep(0,8), fn = ll.probit, y=y, X=X,
       method = "BFGS", control = list(fnscale = -1,
       maxit = 1000), hessian = TRUE)
```

The Fit Model

	Coef	SE	Z-stat
tenure	0.0291	0.0294	0.9922
age	-0.0058	0.0025	-2.2960
dem score	-0.0124	0.0045	-2.7921
civil war	0.0663	0.0890	0.7445
war	0.3175	0.1014	3.1298
pop	0.0409	0.0237	1.7264
energy	0.0269	0.0243	1.1034
intercept	-1.8362	0.1399	-13.1262

Expected Values

Expected values:

Expected Values

Expected values:

- Estimate: predicted probability of an assassination attempt

Expected Values

Expected values:

- Estimate: predicted probability of an assassination attempt
- at some level for the covariate values.

Expected Values

Expected values:

- Estimate: predicted probability of an assassination attempt
- at some level for the covariate values.

Let's identify some high risk situations ("X_{HR}")

Expected Values

Expected values:

- Estimate: predicted probability of an assassination attempt
- at some level for the covariate values.

Let's identify some high risk situations ("X_{HR}")

```
attempt.vec <- y == 1   # y is attempt without NAs
highrisk <- apply(X = X[attempt.vec,], MARGIN = 2, FUN = median)
```

Expected Values

Expected values:

- Estimate: predicted probability of an assassination attempt
- at some level for the covariate values.

Let's identify some high risk situations ("X_{HR}")

```
attempt.vec <- y == 1   # y is attempt without NAs
highrisk <- apply(X = X[attempt.vec,], MARGIN = 2, FUN = median)
```

Predictor:	tenure	age	dem score	civil war
Value:	1392	54	-3	0
	war	pop.	energy	
	0	12980000	3828	

Table : Median values for year-states with an assassination attempt.

Expected Values

What's the estimated probability of an assassination at X_{HR}?

Expected Values

What's the estimated probability of an assassination at X_{HR}?

```
library(MASS)
beta.draws <- mvrnorm(10000, mu = opt$par, Sigma
              = solve(-opt$hessian))
```

Expected Values

What's the estimated probability of an assassination at X_{HR}?

```
library(MASS)
beta.draws <- mvrnorm(10000, mu = opt$par, Sigma
              = solve(-opt$hessian))


p.ests <- c()
for(i in 1:10000){
  p.ass.att <- pnorm(highrisk%*%beta.draws[i,])
  outcomes <- rbinom(10000, 1, p.ass.att)
  p.ests[i] <- mean(outcomes)
}
```

Expected Values

What's the estimated probability of an assassination at X_{HR}?

```
library(MASS)
beta.draws <- mvrnorm(10000, mu = opt$par, Sigma
              = solve(-opt$hessian))


p.ests <- c()
for(i in 1:10000){
  p.ass.att <- pnorm(highrisk%*%beta.draws[i,])
  outcomes <- rbinom(10000, 1, p.ass.att)
  p.ests[i] <- mean(outcomes)
}


> mean(p.ests)
[1] 0.0166266
> quantile(p.ests, .025); quantile(p.ests, .975)
 2.5%    97.5%
0.0134  0.0201
```

Expected Values

What are the steps that I just took?

Expected Values

What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty

Expected Values

What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty
2. For each $\tilde{\beta}$:

Expected Values

What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty
2. For each $\tilde{\beta}$:
    a. calculate $X_{HR}\tilde{\beta}$

Expected Values

What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty
2. For each $\tilde{\beta}$:
    a. calculate $X_{HR}\tilde{\beta}$
    b. plug into $\Phi()$ to get probability of attempt for that $\tilde{\beta}$ draw.

Expected Values

What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty
2. For each $\tilde{\beta}$:
    a. calculate $X_{HR}\tilde{\beta}$
    b. plug into $\Phi()$ to get probability of attempt for that $\tilde{\beta}$ draw.
    c. draw a bunch of $Bernoulli(\Phi(X_{HR}\tilde{\beta}))$.

Expected Values

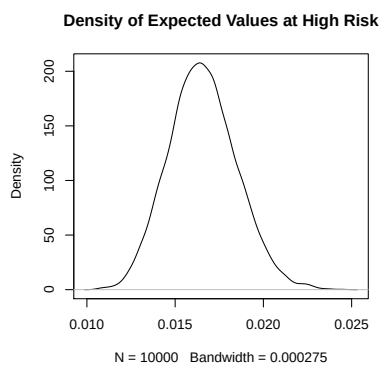
What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty
2. For each $\tilde{\beta}$:
    a. calculate $X_{HR}\tilde{\beta}$
    b. plug into $\Phi()$ to get probability of attempt for that $\tilde{\beta}$ draw.
    c. draw a bunch of $Bernoulli(\Phi(X_{HR}\tilde{\beta}))$.
    d. average over those draws $\Rightarrow$ one simulated $E[y|X_{HR}]$.

Expected Values

What are the steps that I just took?

1. simulate for $\hat{\beta} \Rightarrow$ estimation uncertainty
2. For each $\tilde{\beta}$:
    a. calculate $X_{HR}\tilde{\beta}$
    b. plug into $\Phi()$ to get probability of attempt for that $\tilde{\beta}$ draw.
    c. draw a bunch of $Bernoulli(\Phi(X_{HR}\tilde{\beta}))$.
    d. average over those draws $\Rightarrow$ one simulated $E[y|X_{HR}]$.
3. return to step a.



Expected Values: A Shortcut

What step could we skip here?

Expected Values: A Shortcut

What step could we skip here?

```
beta.draws <- mvrnorm(10000, mu = opt$par, Sigma
                = solve(-opt$hessian))
p.ests2 <- pnorm(highrisk%*%t(beta.draws))

> mean(p.ests2)
[1] 0.01659705
> quantile(p.ests2, .025); quantile(p.ests2, .975)
      2.5%         97.5%
0.01395935   0.01955867
```

Expected Values: A Shortcut

What step could we skip here?

```
beta.draws <- mvrnorm(10000, mu = opt$par, Sigma
                = solve(-opt$hessian))
p.ests2 <- pnorm(highrisk%*%t(beta.draws))

> mean(p.ests2)
[1] 0.01659705
> quantile(p.ests2, .025); quantile(p.ests2, .975)
      2.5%         97.5%
0.01395935   0.01955867
```

This works because $E[y|X_{HR}] = \pi_{HR}$; i.e. the parameter is the expected value of the outcome.

When wouldn't this work?

Rule of thumb: if $E[Y] = \theta$, you are safe taking the shortcut.

More Expected Values

What if I want a bunch of these to see how expected values change with some variable?

More Expected Values

What if I want a bunch of these to see how expected values change with some variable?

```
dem.rng <- -10:10
p.ests <- matrix(data = NA, ncol = length(dem.rng),
                 nrow=10000)
```

More Expected Values

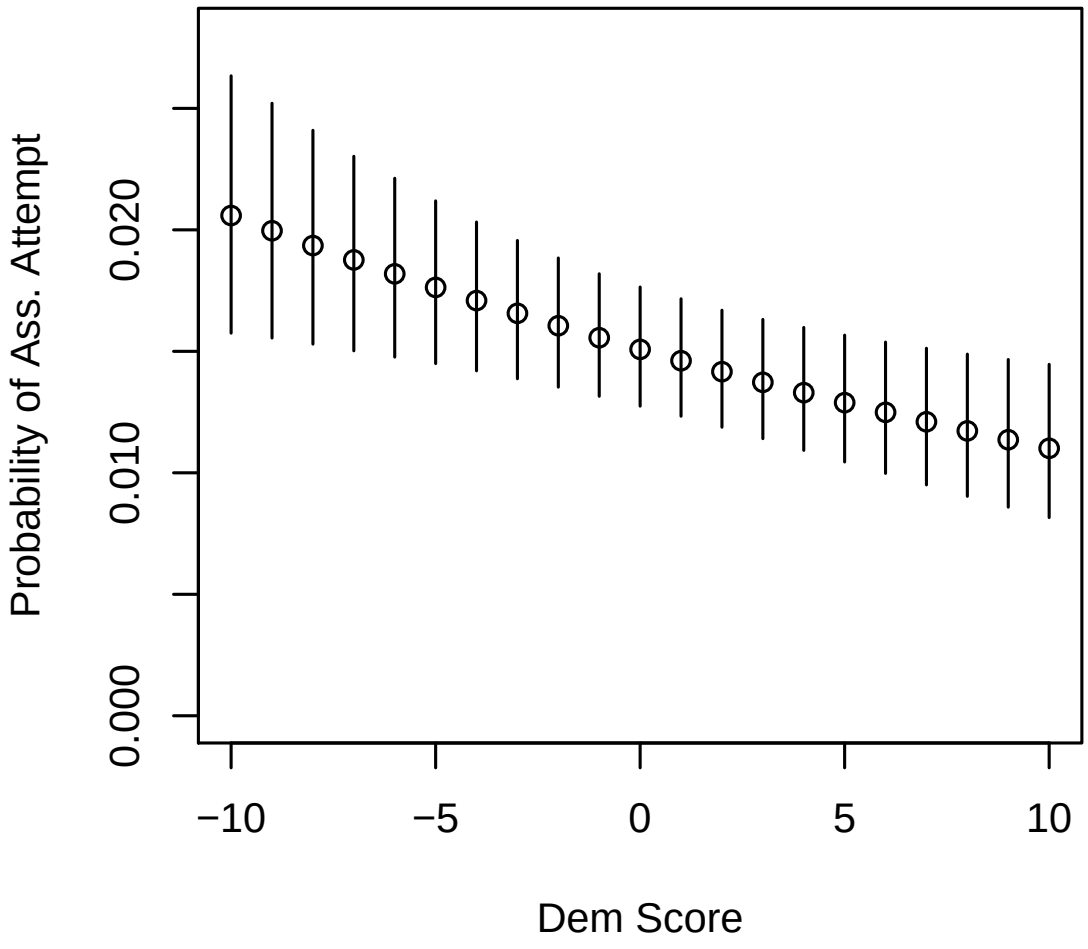
What if I want a bunch of these to see how expected values change with some variable?

```
dem.rng <- -10:10
p.ests <- matrix(data = NA, ncol = length(dem.rng),
                 nrow=10000)

for(j in 1:length(dem.rng)){
  highrisk.dem <- highrisk
  highrisk.dem["dem_score"] <- dem.rng[j]
  p.ests[,j] <- pnorm(highrisk.dem%*%t(beta.draws))
}
```

More Expected Values

What if I want a bunch of these to see how expected values change with some variable?

```
dem.rng <- -10:10
p.ests <- matrix(data = NA, ncol = length(dem.rng),
                 nrow=10000)

for(j in 1:length(dem.rng)){
  highrisk.dem <- highrisk
  highrisk.dem["dem_score"] <- dem.rng[j]
  p.ests[,j] <- pnorm(highrisk.dem%*%t(beta.draws))
}

plot(dem.rng, apply(p.ests,2,mean), ylim = c(0,.028))
segments(x0 = dem.rng, x1 = dem.rng,
         y0 = apply(p.ests, 2, quantile, .025),
         y1 = apply(p.ests, 2, quantile, .975))
```


Probability of Ass. Attempt
0.000
0.010
0.020
−10
−5
0
5
10
Dem Score

First Differences

- Compare expected values of the outcome

First Differences

- Compare expected values of the outcome
- for two different scenarios

First Differences

- Compare expected values of the outcome
- for two different scenarios
- hold all X's constant but one.

First Differences

- Compare expected values of the outcome
- for two different scenarios
- hold all X's constant but one.

Let's find:

$$E[y|X_{War}] - E[y|X_{Nowar}].$$

First Differences

```
highrisk.war <- highrisk
highrisk.war["war"] <- 1
highrisk.nowar <- highrisk
highrisk.nowar["war"] <- 0
```

First Differences

```
highrisk.war <- highrisk
highrisk.war["war"] <- 1
highrisk.nowar <- highrisk
highrisk.nowar["war"] <- 0

fd.ests <- pnorm(highrisk.war%*%t(beta.draws)) -
           pnorm(highrisk.nowar%*%t(beta.draws))
```

First Differences

```
highrisk.war <- highrisk
highrisk.war["war"] <- 1
highrisk.nowar <- highrisk
highrisk.nowar["war"] <- 0

fd.ests <- pnorm(highrisk.war%*%t(beta.draws)) -
           pnorm(highrisk.nowar%*%t(beta.draws))

> mean(fd.ests)
[1] 0.01891
> quantile(fd.ests, .025); quantile(fd.ests, .975)
   2.5%      97.5%
0.00578    0.03609
```

Ordered probit

Suppose our dependent variable is an ordered scale. For example:

- Customers tell you how much they like your product on a 5-point scale from "a lot" to "very little."
- Voters identify their ideology on a 7-point scale: "very liberal," "moderately liberal," "somewhat liberal," "neutral," "somewhat conservative," "moderately conservative," and "very conservative."
- Foreign businesses rate their host country from "not corrupt" to "very corrupt".

What are the problems with using a linear model to study these processes?

A working example: Economic sanctions

Lisa Martin (1992) asks what determines cooperation on sanctions?

Her dependent variable `Coop` measures cooperation on a four-point scale.

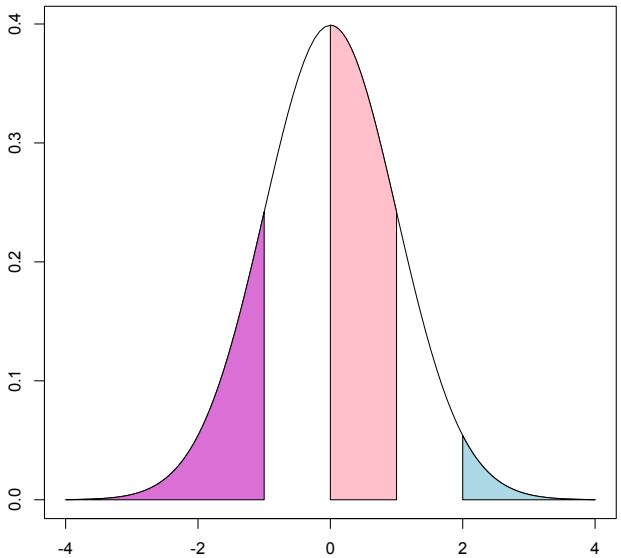
Ordered probit: The intuition

How can we derive the ordered probit?

- Suppose there is a latent (unobserved) data distribution, $Y^* \sim f_{stn}(y^*|\mu_i)$.
- This latent distribution has a systematic component, $\mu_i = x_i\beta$.
- Any realizations, y_i^*, are completely unobserved.
- What you *do* observe is whether y_i^* is between some threshold parameters.

Ordered probit: The intuition

- Threshold parameters τ_j for $j = 1, ..., m$
- Although y_i^* is unobserved, we do observe which of the categories it falls into.



Ordered probit: deriving the likelihood

In equation form,

$$y_{ij} = \begin{cases} 1 & \text{if } \tau_{j-i} < y_i^* \leq \tau_j \\ 0 & \text{otherwise} \end{cases}$$

Our stochastic component is still Bernoulli:

$$Pr(Y_{ij}|\pi) = \pi_{i1}^{y_{i1}} \pi_{i2}^{y_{i2}} \pi_{i3}^{y_{i3}} \ldots$$

where $\sum_{j=1}^{M} \pi_{ij} = 1$

Ordered probit: Deriving the likelihood

Like the regular probit and logit, the key here is deriving π_{ij}.

You use this to derive the likelihood that y_i^* will fall into category j:

$$\begin{aligned}
\pi_{ij} = Pr(Y_{ij} = 1) &= Pr(\tau_{j-1} < y_i^* < \tau_j) \\
&= \int_{\tau_{j-1}}^{\tau_j} f(y_i^* | \mu_i) dy_i^* \\
&= \int_{\tau_{j-1}}^{\tau_j} f(y_i^* | x_i\beta) dy_i^* \\
&= F(\tau_j | x_i\beta) - F(\tau_{j-1} | x_i\beta)
\end{aligned}$$

where F is the cumulative normal density with variance 1.

Ordered probit: Deriving the likelihood

But this is the likelihood of one observation falling in one of the categories. We want to generalize to all observations and all categories

$$L(\tau, \beta|y) = \prod_{i=1}^{n} \left\{ \prod_{j=1}^{m} [\pi_{ij}]^{y_{ij}} \right\}$$

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

where τ is a vector of threshold parameters that you'll have to estimate.

Then we take the log to get the log-likelihood

$$lnL(\tau, \beta|y) = \sum_{i=1}^{n} \sum_{j=1}^{m} y_{ji} ln[F(\tau_j|x_i\beta) - F(\tau_{j-1}|x_i\beta)]$$

Why does X not contain an intercept?

In the binary probit model, we have one cutoff point, say τ_1

$$\begin{aligned} Pr(Y = 1|X\beta) &= 1 - Pr(Y = 0|X\beta) \\ &= 1 - \Phi(\tau_1 - X\beta) \end{aligned}$$

Here, τ_1 is both the cutoff point and the intercept. By including an intercept in $X\beta$ we are setting τ_1 to zero.

Why does X not contain an intercept?

Now in the ordered probit model, we have more than one cutoff point:

$$\begin{aligned} P(y_{i1} = 1) &= Pr(X\beta \leq \tau_1) \\ P(y_{i2} = 1) &= Pr(\tau_1 \leq X\beta \leq \tau_2) \end{aligned}$$

If we included an intercept,

$$\begin{aligned} P(y_{i1} = 1) &= Pr(X\beta + A \leq \tau_1) \\ P(y_{i2} = 1) &= Pr(\tau_1 \leq X\beta + A \leq \tau_2) \end{aligned}$$

Or equivalently we could write this:

$$\begin{aligned} P(y_{i1} = 1) &= Pr(X\beta \leq \tau_1 - A) \\ P(y_{i2} = 1) &= Pr(\tau_1 - A \leq X\beta \leq \tau_2 - A) \end{aligned}$$

$\Rightarrow$ By estimating a cutoff point, we are estimating an intercept.

Ordered probit: Evaluating cooperation for sanctions

Load the data in R

```
library(Zelig)
data(sanction)
head(sanction)
```

	mil	coop	target	import	export	cost	num	ncost
1	1	4	3	1	1	4	15	major loss
2	0	2	3	0	1	3	4	modest loss
3	0	1	3	1	0	2	1	little effect
4	1	1	3	1	1	2	1	little effect
5	0	1	3	1	1	2	1	little effect
6	0	1	3	0	1	2	1	little effect

Ordered probit: Evaluating cooperation for sanctions

We're going to look at the covariates:

- `target` which is a measure of the economic health and political stability of the target country
- `cost` which is a measure of the cost of the cost of the sanctions

Ordered probit: Using Zelig

We estimate the model using Zelig the `oprobit` call:

```
z.out <- zelig(factor(coop) ~ target + cost + mil,
    model="oprobit", data=sanction)
```

Note that you could use `model = "ologit"` and get similar inferences.

Ordered Probit: Using Zelig

What does the output look like?

```
> z.out
Call:
zelig(formula = factor(coop) ~ 1 + target + cost + mil, model = "oprobit",
    data = sanction)

Coefficients:
             Value Std. Error    t value
target -0.1602725  0.1879563 -0.8527113
cost    0.7672444  0.1907797  4.0216242
mil     0.6389931  0.4165843  1.5338865

Intercepts:
    Value   Std. Error t value
1|2  1.1196  0.4435     2.5247
2|3  1.8709  0.4646     4.0270
3|4  2.9159  0.5249     5.5547

Residual Deviance: 162.945
AIC: 174.945
```

These are a little hard to interpret, so we turn to our bag of tricks...

Ordered Probit: Using Zelig

Suppose we want to compare cooperation when there is or is not military action addition to the sanction.

```
x.low <- setx(z.out, mil = 0)
x.high <- setx(z.out, mil = 1)
```

Ordered Probit: Using Zelig

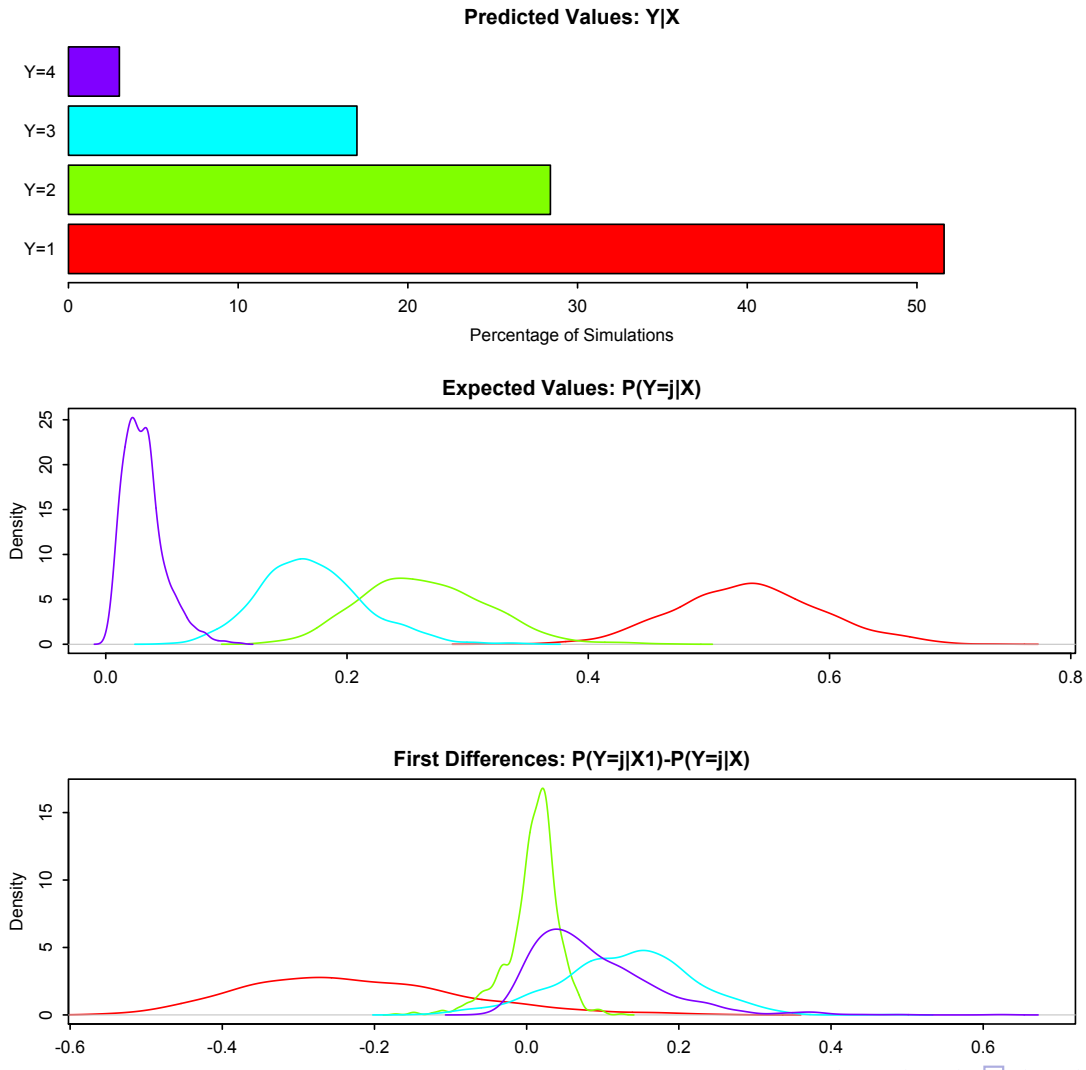
Now we can simulate values using these hypothetical military involvements:

```
s.out <- sim(z.out, x = x.low, x1 = x.high)
> summary(s.out)

  Model: oprobit
  Number of simulations: 1000

Values of X
  (Intercept)   target     cost mil
1           1 2.141026 1.807692   0

Values of X1
  (Intercept)   target     cost mil
1           1 2.141026 1.807692   1

Expected Values: P(Y=j|X)
        mean         sd        2.5%      97.5%
1 0.53319553 0.06030198 0.420702369 0.65622723
2 0.26437778 0.05225064 0.173653596 0.36940997
3 0.17097079 0.04377157 0.093288499 0.26643960
4 0.03145590 0.01714663 0.007172707 0.07420374
```

.
.
.

Ordered Probit: Using Zelig

And then you can use the `plot(s.out)` command to visualize



Ordered probit: Evaluating cooperation for sanctions

Make a matrix for the y's indicating what category it is in:

```
y <- sanction$coop
y0 <- sort(unique(y))
m = 4
Z <- matrix(NA, nrow(sanction), m)
for (j in 1:m)
        Z[,j] <- y==y0[j]
X <- cbind(sanction$target, sanction$cost, sanction$mil)
```

Ordered probit: Evaluating cooperation for sanctions

Create the log-likelihood function

```
ll.oprobit <- function(par, Z, X){
      beta <- par[1:ncol(X)]
      tau <- par[(ncol(X)+1):length(par)]
      ystarmu <- X%*%beta
      m <- length(tau) + 1
      probs =cprobs = matrix(nrow=length(ystarmu), ncol=m)
      for (j in 1:(m-1))
                cprobs[,j] <- pnorm(tau[j]- ystarmu)
      probs[,m] <- 1-cprobs[,m-1]
      probs[,1] <- cprobs[,1]
      for (j in 2:(m-1))
                probs[,j] <- cprobs[,j] - cprobs[,(j-1)]
      sum(log(probs[Z]))
}
```

Ordered probit: Evaluating cooperation for sanctions

Optimize

```
par <- c(rep(1,3),0,1,2)
optim(par, ll.oprobit, Z=Z, X=X,
      method="BFGS", control=list(fnscale=-1))
```

Output

```
out$par
[1] -0.1602698  0.7672435  0.6389932
    1.1196181  1.8708925  2.9159060
```