GOV 2001/ 1002/ Stat E-200 Section 8
Ordered Probit and Zero-Inflated Logit

Solé Prillaman

Harvard University

March 25, 2015

Logistics

Reading Assignment- Becker and Kennedy (1992), Harris and Zhao (2007) (sections 1 and 2), and UPM ch 5.4-5.10.

Re-replication- Due by 6pm Wednesday, April 1 on Canvas.

Re-replication

Re-replication- Due **April 1** at 6pm.

- You will receive all of the replication files from another team.
- It is your responsibility to hand-off your replication files. (Re-replication teams are posted on Canvas.)
- Go through the replication code and try to improve it in any way you can.
- Provide a short write-up of thoughts on the replication and ideas for their final paper.
- Aim to be helpful, not critical!

Outline

The Ordered Probit Model

Zero-Inflated Logistic Regression

Binomial Model

Ordered Categorical Variables

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?

AN EXAMPLE: COOPERATION ON SANCTIONS

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

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

ORDERED PROBIT: COOPERATION ON 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: Cooperation on 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 sanctions
- `mil` which is a measure of whether or not there is military action in addition to the sanction

Maximum Likelihood Estimation

Steps to finding the MLE:

1. **Write out the model.**
2. Calculate the likelihood (L($\theta|y$)) for all observations.
3. Take the log of the likelihood ($\ell(\theta|\mathbf{Y})$).
4. Plug in the systematic component for θ_i.
5. Bring in observed data.
6. Maximize $\ell(\theta|y)$ with respect to θ and confirm that this is a maximum.
7. Find the variance of your estimate.
8. Calculate quantities of interest.

1. The Ordered Probit Model

How can we derive the ordered probit?

- Suppose there is a latent (unobserved) data distribution, $Y^* \sim f_N(\mu_i, 1)$.
- 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.

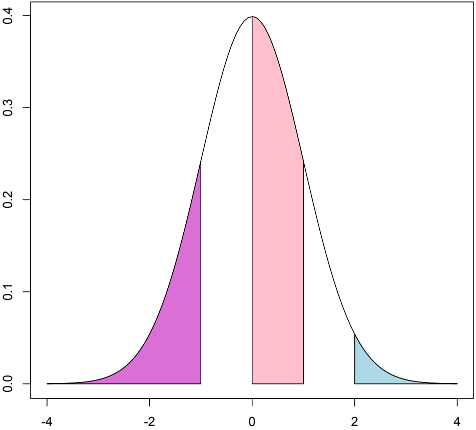
The Model:

$$Y^* \sim f_N(\mu_i, 1)$$

$$\mu_i = x_i\beta$$

1. The Ordered Probit Model

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



1. The Ordered Probit Model

In equation form,

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

What does **Y** look like in our example?

$$\mathbf{Y} = \begin{bmatrix} y_{1,1} & y_{1,2} & y_{1,3} & y_{1,4} \\ y_{2,1} & y_{2,2} & y_{2,3} & y_{2,4} \\ y_{3,1} & y_{3,2} & y_{3,3} & y_{3,4} \\ y_{4,1} & y_{4,2} & y_{4,3} & y_{4,4} \end{bmatrix}$$

1. The Ordered Probit Model

In equation form,

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

Our stochastic component for y_{ij} is still Bernoulli, so:

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

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

1. The Ordered Probit Model

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

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

$$\begin{aligned}
\pi_{ij} = Pr(Y_{ij} = 1) &= Pr(\tau_{j-1} < y_i^* < \tau_j) \\
&= \int_{\tau_{j-1}}^{\tau_j} f_N(y_i^* | \mu_i, \sigma^2) dy_i^* \\
&= \int_{\tau_{j-1}}^{\tau_j} f_N(y_i^* | \mu_i = x_i\beta, \sigma^2 = 1) dy_i^* \\
&= F_N(\tau_j | \mu = x_i\beta, \sigma^2 = 1) - F_N(\tau_{j-1} | \mu = x_i\beta, \sigma^2 = 1) \\
&= \Phi(\tau_j - x_i\beta) - \Phi(\tau_{j-1} - x_i\beta)
\end{aligned}$$

where F is the cumulative density of Y_i^* and Φ is the CDF of the standardized normal.

1. Ordered Probit Model

The latent model:

1. $Y_i^* \sim f_{\text{stn}}(y_i^*|\mu_i)$.
2. $\mu_i = X_i\beta$
3. Y_i^* and Y_j^* are independent for all $i \neq j$.

The observed model:

1. $Y_{ij} \sim f_{\text{bern}}(y_{ij}|\pi_{ij})$.
2. $\pi_{ij} = \Phi(\tau_j - X_i\beta) - \Phi(\tau_{j-1} - X_i\beta)$

Note: for the ordered logit Y_i^* is distributed logistic and $\pi_{ij} = \frac{e^{\tau_j - X_i\beta}}{1+e^{\tau_j - X_i\beta}} - \frac{e^{\tau_{j-1} - X_i\beta}}{1+e^{\tau_{j-1} - X_i\beta}}$

MAXIMUM LIKELIHOOD ESTIMATION

Steps to finding the MLE:

1. Write out the model.
2. **Calculate the likelihood ($L(\theta|y)$) for all observations.**
3. **Take the log of the likelihood ($\ell(\theta|Y)$).**
4. **Plug in the systematic component for θ_i.**
5. Bring in observed data.
6. Maximize $\ell(\theta|y)$ with respect to θ and confirm that this is a maximum.
7. Find the variance of your estimate.
8. Calculate quantities of interest.

2-4. Ordered Probit: Deriving the likelihood

We want to generalize to all observations and all categories

$$\begin{aligned}
L(\tau, \beta|y) &= \prod_{i=1}^{n} Pr(Y_{ij}|\pi) \\
&= \prod_{i=1}^{n} \pi_{i1}^{y_{i1}} \pi_{i2}^{y_{i2}} \pi_{i3}^{y_{i3}} \ldots \\
&= \prod_{i=1}^{n} \left\{ \prod_{j=1}^{m} [\pi_{ij}]^{y_{ij}} \right\} \\
&= \prod_{i=1}^{n} \left\{ \prod_{j=1}^{m} \left[\Phi(\tau_j - X_i\beta) - \Phi(\tau_{j-1} - X_i\beta) \right]^{y_{ij}} \right\}
\end{aligned}$$

3-4. Ordered Probit: Deriving the likelihood

Then we take the log to get the log-likelihood

$$
\begin{aligned}
\ell(\tau, \beta | y) &= \ln\left(\prod_{i=1}^{n}\left\{\prod_{j=1}^{m}\left[\Phi(\tau_j - X_i\beta) - \Phi(\tau_{j-1} - X_i\beta)\right]^{y_{ij}}\right\}\right) \\
\ell(\tau, \beta | y) &= \sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} ln[\Phi(\tau_j - X_i\beta) - \Phi(\tau_{j-1} - X_i\beta)]
\end{aligned}
$$

How many parameters are there to estimate in this model?

j-1+k

3-4. Ordered Probit: Deriving the likelihood

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]))
}
```

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,

$$\begin{aligned} P(y_{i1} = 1) &= Pr(X\beta + \beta_0 \leq \tau_1) \\ P(y_{i2} = 1) &= Pr(\tau_1 \leq X\beta + \beta_0 \leq \tau_2) \end{aligned}$$

Or equivalently we could write this:

$$\begin{aligned} P(y_{i1} = 1) &= Pr(X\beta \leq \tau_1 - \beta_0) \\ P(y_{i2} = 1) &= Pr(\tau_1 - \beta_0 \leq X\beta \leq \tau_2 - \beta_0) \end{aligned}$$

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

Maximum Likelihood Estimation

Steps to finding the MLE:

1. Write out the model.
2. Calculate the likelihood ($L(\theta|y)$) for all observations.
3. Take the log of the likelihood ($\ell(\theta|\mathbf{Y})$).
4. Plug in the systematic component for θ_i.
5. **Bring in observed data.**
6. Maximize $\ell(\theta|y)$ with respect to θ and confirm that this is a maximum.
7. Find the variance of your estimate.
8. Calculate quantities of interest.

5. Ordered Probit: Example Data

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

```
y <- sanction$coop
# Find all of the unique categories of y
y0 <- sort(unique(y))
m <- length(y0)
Z <- matrix(NA, nrow(sanction), m)
# Fill in our matrix with logical values if
# the observed value is in each category
# Remember R can treat logical values as 0/1s
for (j in 1:m){Z[,j] <- y==y0[j]}
X <- cbind(sanction$target, sanction$cost, sanction$mil)
```

Maximum Likelihood Estimation

Steps to finding the MLE:

1. Write out the model.
2. Calculate the likelihood ($L(\theta|y)$) for all observations.
3. Take the log of the likelihood ($\ell(\theta|\mathbf{Y})$).
4. Plug in the systematic component for θ_i.
5. Bring in observed data.
6. **Maximize $\ell(\theta|y)$ with respect to θ and confirm that this is a maximum.**
7. **Find the variance of your estimate.**
8. Calculate quantities of interest.

6-7. Ordered Probit: Calculate MLE

Optimize

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

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

sqrt(diag(solve(-out$hessian)))
[1] 0.1879563 0.1907797 0.4165843 0.4434676 0.4645836 0.5249454
```

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"` for the ordered logit 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 = "opr
    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
```

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

Maximum Likelihood Estimation

Steps to finding the MLE:

1. Write out the model.
2. Calculate the likelihood ($\mathrm{L}(\theta|y)$) for all observations.
3. Take the log of the likelihood ($\ell(\theta|\mathbf{Y})$).
4. Plug in the systematic component for θ_i.
5. Bring in observed data.
6. Maximize $\ell(\theta|y)$ with respect to θ and confirm that this is a maximum.
7. Find the variance of your estimate.
8. **Calculate quantities of interest.**

8. Ordered Probit: Using Zelig for QoIs

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

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

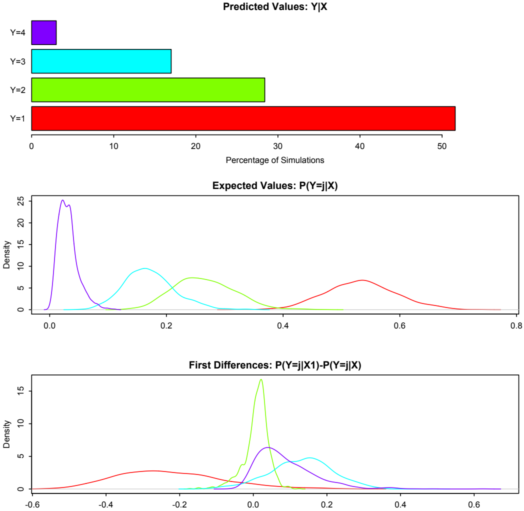
8. Ordered Probit: Using Zelig for QoIs

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
```

8. Ordered Probit: Using Zelig for QoIs

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



OUTLINE

The Ordered Probit Model

Zero-Inflated Logistic Regression

Binomial Model

WHAT IS ZERO-INFLATION?

Let's return to binary data.

- What if we knew that something in our data was mismeasured?
- For example, what if we thought that some of our data were sytematically zero rather than randomly zero? This could be when:
    1. Some data are spoiled or lost
    2. Survey respondents put "zero" to an ordered answer on a survey just to get it done.

If our data are mismeasured in some systematic way, our estimates will be off.

A Working Example: Fishing

You're trying to figure out the probability of catching a fish in a lake from a survey. People were asked:

- How many children were in the group
- How many people were in the group
- Whether they caught a fish.

A Working Example: Fishing

The problem is, some people didn't even fish! These people have systematically zero fish.

Zero-Inflated Logit Model

We're going to assume that whether or not the person fished is the outcome of a Bernoulli trial.

$$Y_i = \begin{cases} 0 & \text{with probability } \psi_i \\ \text{Logistic} & \text{with probability } 1 - \psi_i \end{cases}$$

ψ_i is the probability that you do not fish.

This is a mixture model because our data is a mix of these two types of groups each with their own data generation process.

Zero-Inflated Logit Model

Given that you fished, the logistical model is what we have done before:

1. $Y_i \sim f_{\text{bern}}(y_i|\pi_i)$.
2. $\pi_i = \frac{1}{1+e^{-X_i\beta}}$
3. Y_i and Y_j are independent for all $i \neq j$.

So the probability that Y is 0:

$$P(Y_i = 0|\text{ fished}) = 1 - \frac{1}{1 + e^{-X_i\beta}}$$

and the probability that Y is 1:

$$P(Y_i = 1|\text{ fished}) = \frac{1}{1 + e^{-X_i\beta}}$$

Zero-Inflated Logit Model

Given that you *did not* fish, what is the model?

So the probability that Y is 0:

$$P(Y_i = 0 | \text{ not fished}) = 1$$

and the probability that Y is 1:

$$P(Y_i = 1 | \text{ not fished}) = 0$$

ZERO-INFLATED LOGIT MODEL

Let's say that there is a ψ_i probability that you did not fish. We can write out the distribution of Y_i as (stochastic component):

$$P(Y_i = y_i|\beta, \psi_i) \begin{cases} \psi_i + (1-\psi_i)\left(1 - \frac{1}{1+e^{-X\beta}}\right) & \text{if } y_i = 0 \\ (1-\psi_i)\left(\frac{1}{1+e^{-X\beta}}\right) & \text{if } y_i = 1 \end{cases}$$

So, we can rewrite this as:

$$P(Y_i|\beta, \psi_i) = \left[\psi_i + (1-\psi_i)\left(1 - \frac{1}{1+e^{-X_i\beta}}\right)\right]^{1-Y_i} \left[(1-\psi_i)\left(\frac{1}{1+e^{-X_i\beta}}\right)\right]^{Y_i}$$

And we can put covariates on ψ (systematic component):

$$\psi = \frac{1}{1+e^{-z_i\gamma}}$$

ZERO-INFLATED LOGIT: DERIVING THE LIKELIHOOD

The likelihood function is proportional to the probability of Y_i:

$$
\begin{aligned}
L(\beta,\gamma|Y_i) &\propto P(Y_i|\beta,\gamma) \\
&= \left[\psi_i + (1-\psi_i)\left(1-\frac{1}{1+e^{-X_i\beta}}\right)\right]^{1-Y_i} \\
&\quad \left[(1-\psi_i)\left(\frac{1}{1+e^{-X_i\beta}}\right)\right]^{Y_i} \\
&= \left[\frac{1}{1+e^{-z_i\gamma}} + \left(1-\frac{1}{1+e^{-z_i\gamma}}\right)\left(1-\frac{1}{1+e^{-X_i\beta}}\right)\right]^{1-Y_i} \\
&\quad \left[\left(1-\frac{1}{1+e^{-z_i\gamma}}\right)\left(\frac{1}{1+e^{-X_i\beta}}\right)\right]^{Y_i}
\end{aligned}
$$

Zero-Inflated Logit: Deriving the Likelihood

Multiplying over all observations we get:

$$
\begin{aligned}
L(\beta,\gamma|Y) \quad = \quad & \prod_{i=1}^{n}\left[\frac{1}{1+e^{-z_i\gamma}}+\left(1-\frac{1}{1+e^{-z_i\gamma}}\right)\left(1-\frac{1}{1+e^{-X_i\beta}}\right)\right]^{1-Y_i} \\
& \left[\left(1-\frac{1}{1+e^{-z_i\gamma}}\right)\left(\frac{1}{1+e^{-X_i\beta}}\right)\right]^{Y_i}
\end{aligned}
$$

Taking the log we get:

$$
\begin{aligned}
\ell(\beta,\gamma) \quad = \quad & \sum_{i=1}^{n}\left\{(1-Y_i)\ln\left[\frac{1}{1+e^{-z_i\gamma}}+\left(1-\frac{1}{1+e^{-z_i\gamma}}\right)\left(1-\frac{1}{1+e^{-X_i\beta}}\right)\right]\right. \\
& \left.+Y_i\ln\left[\left(1-\frac{1}{1+e^{-z_i\gamma}}\right)\left(\frac{1}{1+e^{-X_i\beta}}\right)\right]\right\}
\end{aligned}
$$

How many parameters do we need to estimate?

LET'S PROGRAM THIS IN R

Load and get the data ready:

```
fish <- read.table("http://www.ats.ucla.edu/stat/data/fish.csv",
sep=",", header=T)
X <- fish[c("child", "persons")]
Z <- fish[c("persons")]
X <- as.matrix(cbind(1,X))
Z <- as.matrix(cbind(1,Z))
y <- ifelse(fish$count>0,1,0)
```

LET'S PROGRAM THIS IN R

Write out the Log-likelihood function

```
ll.zilogit <- function(par, X, Z, y){
beta <- par[1:ncol(X)]
gamma <- par[(ncol(X)+1):length(par)]
psi <- 1/(1+exp(-Z%*%gamma))
pie <- 1/(1+exp(-X%*%beta))
sum(y*log((1-psi)*pie) + (1-y)*(log(psi + (1-psi)*(1-pie))))
}
```

LET'S PROGRAM THIS IN R

Optimize to get the results

```
par <- rep(1,(ncol(X)+ncol(Z)))
out <- optim(par, ll.zilogit, Z=Z, X=X,y=y, method="BFGS",
      control=list(fnscale=-1), hessian=TRUE)

out$par
[1]  0.9215608 -2.6952940  0.5814261  1.5657467 -1.2020805

varcv.par <- solve(-out$hessian)
```

PLOTTING TO SEE THE RELATIONSHIP

These numbers don't mean a lot to us, so we can plot the predicted probabilities of a group having not fished (i.e. predict ψ).

First, we have to simulate our gammas:

```
library(mvtnorm)
sim.pars <- rmvnorm(10000, out$par, varcv.par)
# Subset to only the parameters we need (gammas)
# Better to simulate all though
sim.z <- sim.pars[,(ncol(X)+1):length(par)]
```

Plotting to See the Relationship

We then generate predicted probabilities of not fishing for different sized groups.

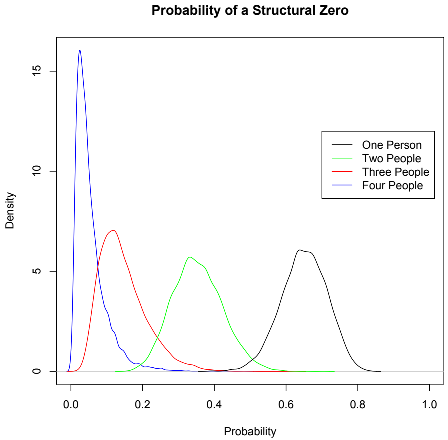
```
person.vec <- seq(1,4)
Zcovariates <- cbind(1, person.vec)
exp.holder <- matrix(NA, ncol=4, nrow=10000)
for(i in 1:length(person.vec)){
exp.holder[,i] <- 1/(1+exp(-Zcovariates[i,]%*%t(sim.z)))
}
```

Plotting to See the Relationship

Using these numbers, we can plot the densities of probabilities, to get a sense of the probability and the uncertainty.

```
plot(density(exp.holder[,4]), col="blue", xlim=c(0,1),
    main="Probability of a Structural Zero", xlab="Probability")
lines(density(exp.holder[,3]), col="red")
lines(density(exp.holder[,2]), col="green")
lines(density(exp.holder[,1]), col="black")
legend(.7,12, legend=c("One Person", "Two People",
   "Three People", "Four People"),
     col=c("black", "green", "red", "blue"), lty=1)
```

PLOTTING TO SEE THE RELATIONSHIP



OUTLINE

Binomial Model

Suppose our dependent variable is the number of successes in a series of *independent* trials. For example:

- The number of heads in 10 coin flips.
- The number of times you voted in the last six elections.
- The number of Supreme Court cases the government won in the last ten decisions.

We can use a generalization of the binary model to study these processes.

Binomial Model

- Stochastic Component

$$
\begin{aligned}
Y_i &\sim Binomial(y_i|\pi_i) \\
P(Y_i = y_i|\pi_i) &= \binom{N}{y_i}\pi_i^{y_i}(1-\pi_i)^{N-y_i}
\end{aligned}
$$

  - $\pi_i^{y_i}$: There are y_i successes each with probability of π_i
  - $(1-\pi_i)^{N-y_i}$: There are $N-y_i$ failures each with probability $1-\pi_i$
  - $\binom{N}{y_i}$: Number of ways to distribute y_i successes in N trials; order of successes does not matter.

Binomial Model

- Systematic Component

$$\pi_i = \frac{1}{1 + e^{-x_i \beta}}$$

Why?

Binomial Model

Derive the likelihood:

$$
\begin{aligned}
L(\pi_i|y_i) &= P(y_i|\pi_i) \\
&= \prod_{i=1}^{n} \binom{N}{y_i} \pi_i^{y_i} (1-\pi_i)^{N-y_i} \\
lnL(\pi_i|y_i) &= \sum_{i=1}^{n} \left[ln \binom{N}{y_i} + ln\pi_i^{y_i} + ln(1-\pi_i)^{N-y_i} \right] \\
&= \sum_{i=1}^{n} [y_i ln\pi_i + (N-y_i) ln(1-\pi_i)]
\end{aligned}
$$

Binomial Model

We can operationalize this in `R` by coding the log likelihood up ourselves. First, let's make up some data to play with:

```
x1 <- rnorm(1000,0,1)
x2 <- rnorm(1000,9,.5)
pi <- inv.logit(-5 + .4*x1 +.6*x2)
y <- rbinom(1000,10,pi)
```

Binomial Model

Write out the Log-likelihood function

```
ll.binom <- function(par, N, X, y){
            pi <- 1/(1 + exp(-1*X%*%par))
            out <- sum(y * log(pi) + (N - y)*log(1-pi))
            return(out)
           }
```

Binomial Model

Optimize to get the results

```
my.optim <- optim(par = c(0,0,0), fn = ll,
    y = y, X = cbind(1,x1,x2), N = 10,
    method = "BFGS", control=list(fnscale=-1), hessian=T)

my.optim$par
[1] -4.6132799  0.3836413  0.5590359
```

Given that

```
pi <- inv.logit(-5 + .4*x1 +.6*x2)
```

the output doesn't look too bad.