GOV 2001/ 1002/ Stat E-200 Section 7
Probit Models and Quantities of Interest

Solé Prillaman

Harvard University

March 11, 2015

Logistics

Reading Assignment- King and Roberts (2014)

Problem Set 5- Due by 6pm Wednesday, March 25 on Canvas.

Assessment Question- Due by 6pm Wednesday, March 25 on Canvas. You must work alone and only one attempt.

Paper Replication- Due by 6pm on Wednesday, March 25 on Canvas.

Paper Topic Survey- Due by Friday, March 20 on Canvas.

Office Hours

Friday March 13 - Appointments by Email

Monday March 23 - Stephen 11am-1pm in CGIS Café

Tuesday March 24 - Solé 10am-12pm in CGIS Café

Replication

Replication- Due **March 25** at 6pm.

- All data required to do the replication
- An R script that replicates all relevant tables, graphs, and numbers in the paper
- A PDF of the original paper
- A PDF of all the tables, graphs, and numbers that your replicated (These should look as close to what's in the original paper as possible)
- A brief description of what you plan to write your final paper about

Submit to Canvas and to Re-replication team!

RE-REPLICATION

Re-replication- Due **April 1** (no joke) 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 to be posted on Canvas by the 25th.
- More next Section!

Outline

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.

Maximum Likelihood Estimation

Steps to finding the MLE:

1. **Write out the model.**
2. Calculate the likelihood ($\text{L}(\theta|y)$) 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.

THE LOGIT MODEL

Let's assume Y is a Bernoulli random variable with parameter π, the probability of success.

The model:

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

$$\ell(\beta) = \sum_{i=1}^{n} y_i X_i\beta - X_i\beta - \ln(1 + e^{-X_i\beta})$$

1. The Probit Model

Let's assume Y is a Bernoulli random variable with parameter π, the probability of success.

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. (**Inverse Link Function**)
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 correct scale.

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

1. The Probit Model

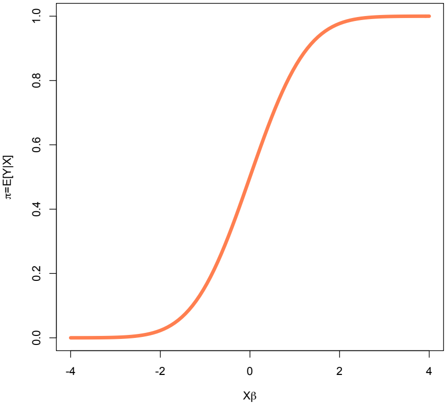
Simply, the model we've just defined:

$$\begin{aligned} Y_i &\overset{iid}{\sim} \text{Bernoulli}(\pi_i) \\ \pi_i &= \Phi(X_i\beta) \end{aligned}$$

What are the parameters in this model? β

How many parameters are in this model?

1. The Probit Model



What assumptions are we making with $\pi_i = \Phi(X_i\beta)$?
increasing, symmetric

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.

2-4. Log-likelihood: The Probit Model

We can then derive the log-likelihood for β:

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

2-4. Log-likelihood: The Probit Model

$$\ell(\beta) = \sum_{i=1}^{n} y_i \ln(\Phi(X_i\beta)) + (1 - y_i)\ln(1 - \Phi(X_i\beta))$$

```
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)
}
```

- Uses a logical test to check that an intercept column has been added
- The STN CDF is evaluated with pnorm. R's pre-programmed log of the CDF has greater range than
- If `lower.tail = FALSE` gives $Pr(Z \geq z)$.

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.

5. The Data: Congressional Elections

We are going to use the same data from last week's section on House general elections.

`votes0408.dta` is the data that we're going to use to estimate our model.

```
setwd("c:/.../your working directory")
install.packages("foreign")
require(foreign)
votes <- read.dta("votes0408.dta")
```

5. Election Data

What's in the data?

```
head(votes)
  incwin open freshman incpres
1      1    0        1   64.77
2      1    0        0   66.97
3      1    0        1   58.59
4      1    0        0   71.73
5      1    0        0   39.77
6      1    0        1   64.53

apply(votes, 2, summary)
        incwin    open freshman incpres
Min.    0.0000 0.00000   0.0000   30.10
1st Qu. 1.0000 0.00000   0.0000   52.98
Median  1.0000 0.00000   0.0000   58.09
Mean    0.9393 0.08824   0.1233   59.08
3rd Qu. 1.0000 0.00000   0.0000   64.26
Max.    1.0000 1.00000   1.0000   95.00
```

`incwin`: **dependent variable**; did the incumbent (or their party) win this election? {0,1}

`open`: is the incumbent running for reelection? {0,1}

`freshman`: is the incumbent in his/her first term? {0,1}

`incpres`: in the last election, what percentage of votes did the presidential candidate from the incumbent's party receive in this district? (0,100)

5. Election Data

We want to model whether or not the incumbent won using `open`, `freshman`, and `incpres`.

```
# Specify our data for the model
y <- votes$incwin
X <- as.matrix(votes[,2:4])
```

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.**

6-7. Maximize the Likelihood

```
## Estimate the MLE:
opt <- optim(par = rep(0, ncol(X) + 1),
             fn = ll.probit,
             y = y,
             X = X,
             control = list(fnscale = -1),
             hessian = T,
             method = "BFGS")

# Extract the coefficients from optim
coefs <- opt$par
# Calculate the variance
varcov <- -solve(opt$hessian)
# Calculate the standard errors
ses <- sqrt(diag(varcov))
```

Maximize the Likelihood

	Coefficient	SE
Intercept	-1.4731	0.4619
Open Seat	-1.0958	0.1769
Freshman	-0.1515	0.1990
Pres Vote	0.0587	0.0087

Table: DV: Incumbent Election Win

This is what we should report, right?

OUTLINE

UNCERTAINTY

$$\begin{aligned} Y_i &\sim f(\theta_i) \\ \theta_i &= g(X_i\beta) \end{aligned}$$

Estimation uncertainty: Lack of knowledge of θ because of *sampling*.Vanishes as n gets larger.

Fundamental uncertainty: Represented by the stochastic component.Even if we knew β we couldn't perfectly predict Y!

Getting Quantities of Interest

General steps to calculating quantities of interest:

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$

   $$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$

3. Choose one value for each explanatory variable and denote the vector of values X_C
4. Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$

   $$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$

5. Use $\widetilde{\pi}$ to simulate from the stochastic component

   $$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

Getting Quantities of Interest

Where is our uncertainty?

Getting Quantities of Interest

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$
$$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$
3. Choose one value for each explanatory variable and denote the vector of values X_C
4. Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$
$$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$
5. Use $\widetilde{\pi}$ to simulate from the stochastic component
$$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

Getting Quantities of Interest

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. **Estimation Uncertainty:** Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$

   $$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$

3. Choose one value for each explanatory variable and denote the vector of values X_C
4. Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$

   $$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$

5. Use $\widetilde{\pi}$ to simulate from the stochastic component

   $$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

GETTING QUANTITIES OF INTEREST

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. **Estimation Uncertainty:** Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$
   $$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$
3. Choose one value for each explanatory variable and denote the vector of values X_C
4. Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$
   $$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$
5. **Fundamental Uncertainty:** Use $\widetilde{\pi}$ to simulate from the stochastic component
   $$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

QUANTITIES OF INTEREST

- **Predicted Values:** estimates of Y given some values of the covariates.

  eg. Estimates of an incumbent win (0 or 1) given some values of the covariates.

- **Expected values:** estimates of the expected value of Y (or any other feature of Y's distribution) given some values of the covariates.

  eg. Estimates of the predicted probability of an incumbent wingiven some values of the covariates.

- **First Differences:** estimates of the difference between two expected values for different values of the covariates.

  eg. Estimates of the *difference* in the predicted probability of an incumbent win for freshmen v. non-freshmen, given some values of the covariates.

Getting Quantities of Interest

General steps to calculating quantities of interest:

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. **Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$**

   $$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$

3. Choose one value for each explanatory variable and denote the vector of values X_C
4. Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$

   $$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$

5. Use $\widetilde{\pi}$ to simulate from the stochastic component

   $$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

2. Simulate $\widetilde{\beta}$

```
#1. We have our estimates of Betahat and its variance

#2. Draw our beta tildes from a multivariate normal distribution
#     with mean and variance = our estimates
#     to account for estimation uncertainty
#   Note: make sure to use the entire variance covariance matrix!
library(mvtnorm)
set.seed(1234)
sim.betas <- rmvnorm(n = 10000,
                     mean = coefs,
                     sigma = varcov)

head(sim.betas)
dim(sim.betas) ## 10,000 by 4 matrix of simulated betas
```

Getting Quantities of Interest

General steps to calculating quantities of interest:

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$
$$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$
3. **Choose one value for each explanatory variable and denote the vector of values X_C**
4. Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$
$$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$
5. Use $\widetilde{\pi}$ to simulate from the stochastic component
$$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

3. Set X_C

Let's identify a set of values for our covariates that we care about:

```
# Say when people are up for reelection
# They were a freshmen
# And their presidential party got 48% of the vote
# Don't forget about the intercept!
Xc <- c(1,1,1,48)

# Alternatively, we could just set every value at its median
Xc <- apply(X = cbind(1,X), MARGIN = 2, FUN = median)
```

Predicted Values

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$

   $$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$

3. Choose one value for each explanatory variable and denote the vector of values X_C
4. **Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$**

   $$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$

5. **For each $\widetilde{\pi}$ draw one simulation from the stochastic component**

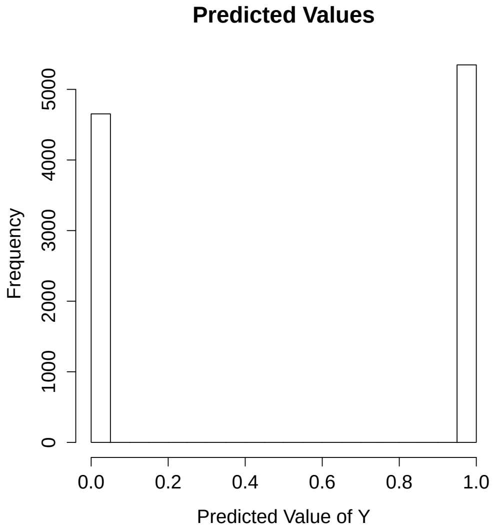
   $$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$

4-5. Calculate Predicted Values

```
#4. Calculate the pi. systematic component
# and
#5. Draw our Y tildes from the stochastic component
#     to account for fundamental uncertainty
pv.ests <- c() # Create an empty vector
for(i in 1:10000){
  pi.tilde <- pnorm(Xc%*%sim.betas[i,])
  pv.ests[i] <- rbinom(1, 1, pi.tilde)
}
head(pv.ests)

# Look at the results
hist(pv.ests,
     main = "Predicted Values",
     xlab = "Predicted Value of Y")
```

PREDICTED VALUES



Expected Values

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$
   $$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$
3. Choose one value for each explanatory variable and denote the vector of values X_C
4. **Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$**
   $$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$
5. **For each $\widetilde{\pi}$ draw m simulations from the stochastic component**
   $$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$
6. **Store the mean of these simulations for each $\widetilde{\pi}$, $E[Y_C]$**

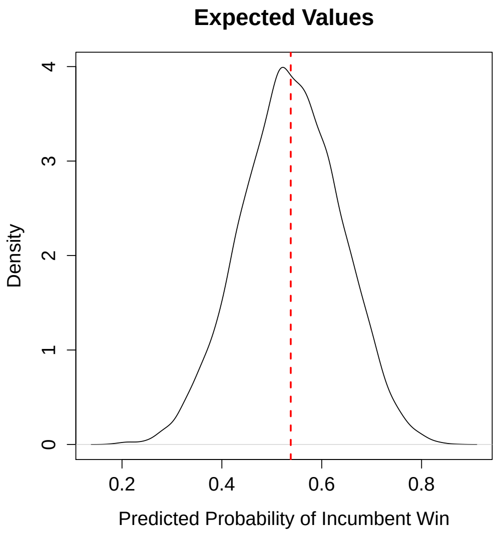
4-6. Calculate Expected Values

```
#4. Calculate the pi. systematic component
# and
#5. Draw our Y tildes from the stochastic component
#     to account for fundamental uncertainty
#6. Calculate the expected value of Y
ev.ests <- c() # Create an empty vector
for(i in 1:10000){
  pi.tilde <- pnorm(Xc%*%sim.betas[i,])
  y.ests <- rbinom(10000, 1, pi.tilde)
  ev.ests[i] <- mean(y.ests)
}
head(ev.ests)

# Look at the results
hist(ev.ests,main = "Expected Values",
     xlab = "Predicted Probability of Incumbent Win")

mean(ev.ests)
quantile(ev.ests, c(.025,.975))
```

Expected Values



Expected Values: A Shortcut

What step could we skip?

```
set.seed(1234)
sim.betas <- rmvnorm(n = 10000,
                     mean = coefs,
                     sigma = varcov)
ev.ests2 <- pnorm(Xc%*%t(sim.betas))

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

Why does this work? because $E[y|X_C] = \pi_C$

We averaged over the fundamental uncertainty!

Expected Values: A Shortcut

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?

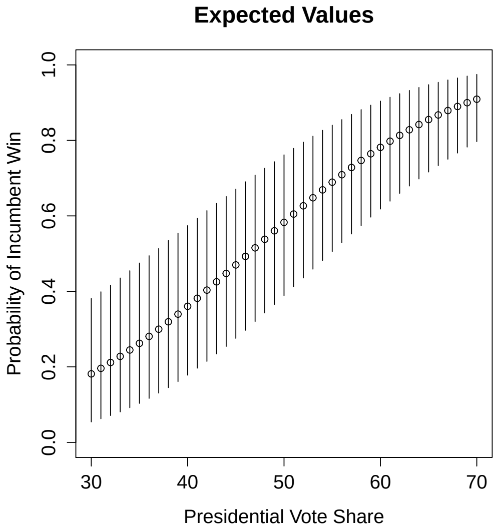
```
#Let's do this for president vote shares from 30% to 70%

presvals <- 30:70
ev.ests <- matrix(data = NA, ncol = length(presvals),
                  nrow=10000)

# Loop over all values of presvals
for(j in 1:length(presvals)){
  Xc.new <- Xc
  Xc.new[4] <- presvals[j]
  ev.ests[,j] <- pnorm(Xc.new%*%t(sim.betas))
}

plot(presvals, apply(ev.ests,2,mean), ylim=c(0,1))
segments(x0 = presvals, x1 = presvals,
         y0 = apply(ev.ests, 2, quantile, .025),
         y1 = apply(ev.ests, 2, quantile, .975))
```

More Expected Values



First Differences

1. Write out your model and estimate $\hat{\beta}_{MLE}$ and the Variance-Covariance Matrix, $\hat{V}(\hat{\beta})$
2. Simulate $\widetilde{\beta}$ from the sampling distribution of $\hat{\beta}_{MLE}$
$$\widetilde{\beta} \sim MVN(\hat{\beta}, \hat{V}(\hat{\beta}))$$
3. Choose one value for each explanatory variable and denote the vector of values X_C
4. **Using X_C and $\widetilde{\beta}$, calculate the systematic component $\widetilde{\theta} = g(X\beta)$**
$$\widetilde{\pi} = \Phi(X_C\widetilde{\beta})$$
5. **For each $\widetilde{\pi}$ draw m simulations from the stochastic component**
$$\widetilde{Y_C} \sim Bern(\widetilde{\pi})$$
6. **Store the mean of these simulations for each $\widetilde{\pi}$, $E[Y_C]$**
7. **Repeat for different values of the covariates, X'_C**
8. **Calculate the difference between $E[Y|X_C]$ and $E[Y|X'_C]$**

4-8. Calculate First Differences

Let's calculate the first difference for freshmen as compared to not being a freshman.

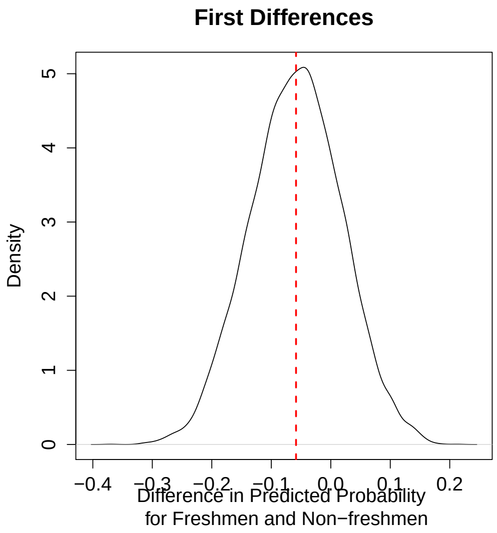
```
# Xc already is set for freshmen
Xc.fresh <- Xc
Xc.nofresh <- Xc
Xc.nofresh[3] <- 0

# Calculate the first differences as the
# difference in the expected values
fd.ests <- pnorm(Xc.fresh%*%t(sim.betas)) -
  pnorm(Xc.nofresh%*%t(sim.betas))

# Look at the results
hist(fd.ests,
     main = "First Differences",
     xlab = "Difference in Predicted Probability \n
     for Freshmen and Non-freshmen")

mean(fd.ests)
quantile(fd.ests, c(.025,.975))
```

First Differences



Outline

USING ZELIG

`Zelig` has three basic steps which coincide with the procedure that we have been using

1. Fit the model: `zelig`
2. Choose some levels of the explanatory variables: `setx`
3. Simulate predicted values, expected values, first differences: `sim`

Using Zelig

```
# We are going to be using Zelig 5.0, which is still in
beta release
# Direct any questions you have about the software to:
# https://groups.google.com/forum/#!forum/zelig-statistical-software
# Report any bugs you find to:
# https://github.com/IQSS/Zelig/issues

# Make sure you have all of the necessary dependencies
installed

# To install Zelig 5.0, you must type exactly:
install.packages("Zelig", type = "source", repos =
"http://r.iq.harvard.edu/")
library(Zelig)

# If you want more detailed instructions, go to
http://docs.zeligproject.org/en/latest/installation_quickstart.html
```

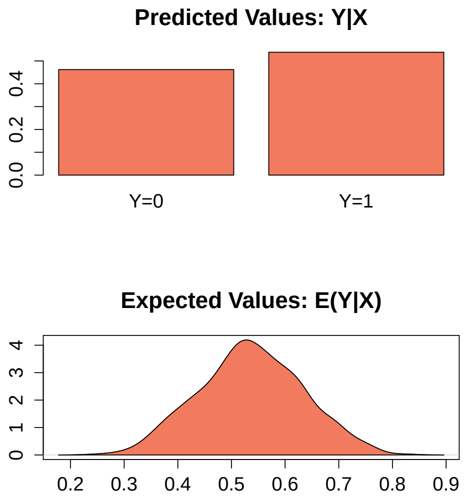
USING ZELIG

```
# Now let's run our model
# This is very similar to the lm function
# but you specify the model as probit
zelig.out <- zelig(incwin ~ open + freshman + incpres,
                   data = votes, model = "probit")
summary(zelig.out)

# Set the values of the covariates
x.evs <- setx(zelig.out, open = 1, freshman = 1, incpres = 48)
x.evs

# Simulate predicted probabilities
set.seed(1234)
zelig.sim <- sim(zelig.out, x=x.evs)
summary(zelig.sim)
# Plot our simulations
par(mar=c(2.5,2.5,2.5,2.5)) # Set our plot margins
plot(zelig.sim)
```

ZELIG'S PLOT


Predicted Values: Y|X
0.0
0.2
0.4
Y=0
Y=1
Expected Values: E(Y|X)
0
1
2
3
4
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9

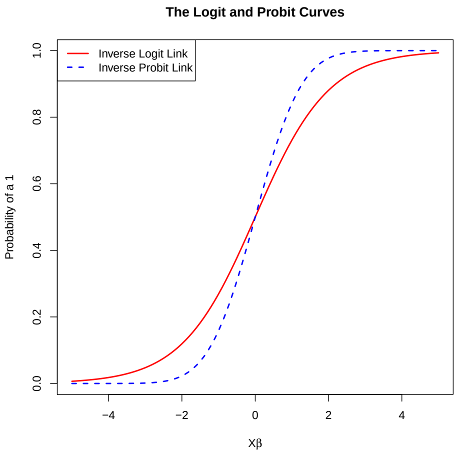
USING `ZELIG`

It's important that you know how `Zelig` is working under the hood. It is basically doing the same thing we are doing.

1. Zelig can handle Expected Values, First Differences, Risk Ratios and Predicted Values
2. Extensive Documentation on the Zelig website (http://zeligproject.org/)
3. Many kinds of models all which use the same basic syntax!

OUTLINE

Logit v. Probit



Logit v. Probit

Note that the coefficients are different:

	Probit	Logit
Intercept	-1.4731	-2.9064
Open Seat	-1.0958	-2.1267
Freshman	-0.1515	-0.3568
Pres Vote	0.0587	0.1112

LOGIT V. PROBIT

But the inferences are the same. If we calculate the first difference between presidential vote share of 45% and 55%:

	Probit	Logit
Mean	0.2192	0.3785
2.5% Quantile	0.1467	0.2066
97.5% Quantile	0.2875	0.5072

Table: First Differences

Logit v. Probit

What if we move out in the tails though? If we calculate the move from 30% vote share to 90%.

	Probit	Logit
Mean	0.8098	0.9590
2.5% Quantile	0.6094	0.7946
97.5% Quantile	0.9385	0.9994

Table: First Differences

Even in the tails it isn't too different.

OUTLINE

Errors and Residuals

Errors are the vertical distances between observations and the unknown Conditional Expectation Function. Therefore, they are unknown.

Residuals are the vertical distances between observations and the estimated regression function. Therefore, they are known.

Notation

Errors represent the difference between the outcome and the true mean.

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{u}$$
$$\mathbf{u} = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}$$

Residuals represent the difference between the outcome and the estimated mean.

$$\mathbf{y} = \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{u}}$$
$$\hat{\mathbf{u}} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$$

VARIANCE OF $\hat{\boldsymbol{\beta}}$

Variance of $\hat{\beta}$ depends on the errors!

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{y} \\ &= \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'(\mathbf{X}\boldsymbol{\beta}+\mathbf{u}) \\ &= \boldsymbol{\beta} + \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}\end{aligned}$$

Remember that because $\hat{\boldsymbol{\beta}}$ is unbiased, then it must be that

$E[(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{u}] = 0$

Variance Reminder

Remember that:

$$\begin{aligned} V(X) &= E\left[(X - E[X])^2\right] \\ &= E(X^2) - E(X)^2 \end{aligned}$$

In Matrix notation:

$$V(X) = E\left[(X - E[X])(X - E[X])'\right]$$

Variance of $\hat{\beta}$

Variance of $\hat{\beta}$ depends on the errors!

$$
\begin{aligned}
V[\hat{\boldsymbol{\beta}}] &= V[\boldsymbol{\beta}] + V[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}] \\
&= \mathbf{0} + V[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}] \\
&= E[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}] - E[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}]E[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}]' \\
&= E[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{u}\mathbf{u}'\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}] - \mathbf{0} \\
&= \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'E[\mathbf{u}\mathbf{u}']\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1} \\
&= \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\boldsymbol{\Sigma}\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}
\end{aligned}
$$

VARIANCE OF $\hat{\boldsymbol{\beta}}$

Under standard OLS assumptions, the model can be written as:

1. $Y_i \sim N(\mu_i, \sigma^2)$.
2. $\mu_i = \mathbf{X}\boldsymbol{\beta}$
3. Y_i and Y_j are independent for all $i \neq j$.

Which implies:

$$\mathbf{u} \sim N_n(0, \boldsymbol{\Sigma})$$

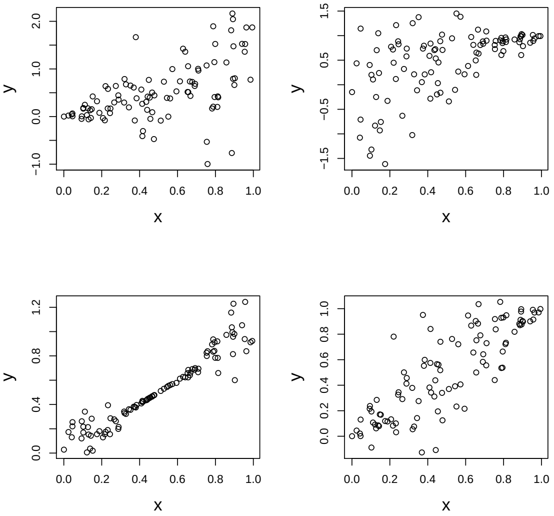
Constant Error Variance

Under standard OLS assumptions, $\mathbf{u} \sim N_n(0, \mathbf{\Sigma})$ with

$$\mathbf{\Sigma} = E[\mathbf{u}\mathbf{u}'] = Var(\mathbf{u}) = \begin{bmatrix} \sigma^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \sigma^2 \end{bmatrix}$$

Why is $\mathbf{\Sigma} = Var(\mathbf{u})$? Why are the off-diagonal elements 0?

Evidence of Non-constant Error Variance



Notation

The constant error variance assumption sometimes called homoskedasiticity states that

$$\boldsymbol{\Sigma} = Var(\mathbf{u}) = \begin{bmatrix} \sigma^2 & 0 & 0 & \ldots & 0 \\ 0 & \sigma^2 & 0 & \ldots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \ldots & \sigma^2 \end{bmatrix}$$

What if we actually have non-constant error variance, or heteroskedasticity?

$$\boldsymbol{\Sigma} = Var(\mathbf{u}) = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \ldots & 0 \\ 0 & \sigma_2^2 & 0 & \ldots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \ldots & \sigma_n^2 \end{bmatrix}$$

Consequences of non-constant error variance

- The $\hat{\sigma}^2$ will not be unbiased for σ^2.
- For "α" level tests, probability of Type I error will not be α.
- "$1-\alpha$" confidence intervals will not have $1-\alpha$ coverage probability.
- The LS estimator is no longer BLUE.

However,

- The degree of the problem depends on the amount of heteroskedasticity.
- $\hat{\boldsymbol{\beta}}$ is still unbiased for $\boldsymbol{\beta}$

Robust Standard Errors

The $Var(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{\Sigma}\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$ and suppose

$$V[\mathbf{u}] = \boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & 0 & 0 & \dots & 0 \\ 0 & \sigma_2^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \sigma_n^2 \end{bmatrix}$$

Huber (and then White) showed that

$$(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}' \begin{bmatrix} \hat{\mathbf{u}}_1^2 & 0 & 0 & \dots & 0 \\ 0 & \hat{\mathbf{u}}_2^2 & 0 & \dots & 0 \\ & & & \vdots & \\ 0 & 0 & 0 & \dots & \hat{\mathbf{u}}_n^2 \end{bmatrix} \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

is a **consistent** estimator of $Var(\hat{\boldsymbol{\beta}})$.

Robust Standard Errors

Things to note about this approach:

1. Requires larger sample size
   - large enough for each estimate (e.g., large enough in both treatment and baseline groups)
   - large enough for consistent estimates (e.g., need $n \geq 250$ for Stata default when highly heteroskedastic (Long and Ervin 2000)).
2. Doesn't make $\hat{\beta}$ BLUE
3. What are you going to do with predicted probabilities?

WHAT ABOUT OTHER MODELS?

Huber (1967) developed a general way to find the standard errors for models *that are specified in the wrong way.*

Under certain conditions, you can get the standard errors, even if your model is misspecified.

These are the robust standard errors that scholars now use for other glm's, and that happen to coincide with the linear case.

Robust Standard Errors for GLMs

To derive robust standard errors in the general case, we assume that

$$y_i \sim f_i(y_i|\theta)$$

Then our likelihood function is given by

$$L(\theta) = \prod_{i=1}^{n} f_i(y_i|\theta)$$

and thus the log-likelihood is

$$\ell(\theta) = \sum_{i=1}^{n} \ln f_i(y_i|\theta)$$

Robust Standard Errors for GLMs

Remember that we denote the first and second partial derivatives of $\ell(\theta)$ as:

$$\begin{aligned}
\ell'(\theta) &= s(\theta) = \sum_{i=1}^{n} g_i(Y_i|\theta) = \sum_{i=1}^{n} \frac{\partial \ln f_i(y_i|\theta)}{\partial \theta} \\
\ell''(\theta) &= H(\theta) = \sum_{i=1}^{n} h_i(Y_i|\theta) = \sum_{i=1}^{n} \frac{\partial^2 \ln f_i(y_i|\theta)}{\partial \theta^2}
\end{aligned}$$

This shouldn't be too unfamiliar. Remember, the Fisher

information matrix is $-E\left[\frac{\partial^2 \ell(\theta)}{\partial \theta^2}\right]$.

Robust Standard Errors for GLMs

- Let's assume the model is correct – there is a true value θ_0 for θ.
- Then we can use the Taylor approximation for the log-likelihood function to estimate what the likelihood function looks like around θ_0:

$$L(\theta) = L(\theta_0) + L'(\theta_0)(\theta - \theta_0) + \frac{1}{2}(\theta - \theta_0)^T L''(\theta_0)(\theta - \theta_0)$$

Robust Standard Errors for GLMs

- We can use the Taylor approximation to approximate $\hat{\theta} - \theta_0$ and therefore the variance covariance matrix.
- We want to find the maximum of the log-likelihood function, so we set $L'(\theta) = 0$:

$$L'(\theta_0) + (\theta - \theta_0)^T L''(\theta_0) = 0$$

$$\hat{\theta} - \theta_0 = [-L''(\theta_0)]^{-1} L'(\theta_0)^T$$

$$Var(\hat{\theta}) = [-L''(\theta_0)]^{-1}[Cov(L'(\theta_0))][-L''(\theta_0)]^{-1}$$

Remember, before we had $Var(\hat{\boldsymbol{\beta}}) = \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{X}'\boldsymbol{\Sigma}\mathbf{X} \left(\mathbf{X}'\mathbf{X}\right)^{-1}$.

Robust Standard Errors for GLMs

It's the sandwich estimator.

$$Var(\hat{\theta}) = [-L''(\theta_0)]^{-1}[Cov(L'(\theta_0))][-L''(\theta_0)]^{-1}$$

$$= \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta})\right]^{-1} \left[\sum_{i=1}^{n} g_i(Y_i|\hat{\theta})' g_i(Y_i|\hat{\theta})\right] \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta})\right]^{-1}$$

Robust Standard Errors for GLMs

It's the sandwich estimator.

$$= \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta}) \right]^{-1} \left[\sum_{i=1}^{n} g_i(Y_i|\hat{\theta})' g_i(Y_i|\hat{\theta}) \right] \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta}) \right]^{-1}$$

Bread: $\left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta}) \right]^{-1}$ Meat: $\left[\sum_{i=1}^{n} g_i(Y_i|\hat{\theta})' g_i(Y_i|\hat{\theta}) \right]$

Robust Standard Errors for GLMs

It's the sandwich estimator.

$$= \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta})\right]^{-1} \left[\sum_{i=1}^{n} g_i(Y_i|\hat{\theta})' g_i(Y_i|\hat{\theta})\right] \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta})\right]^{-1}$$

Bread: Negative inverse hessian Meat: Dot product of the score.

Cluster-Robust Standard Errors

Using clustered-robust standard errors, the meat changes.

Instead of summing over each individual, we first sum over the groups.

$$\left[\sum_{j}\sum_{i\in c_j} g_i(Y_i|\hat{\theta})' g_i(Y_i|\hat{\theta})\right]$$

IMPLEMENTING IN R

- There are lots of different ways to implement these standard errors in R.
- Sometimes it's difficult to figure out what is going on in Stata.
- But by really understanding what is going on in R, you will be able to implement once you know the equation for Stata.

SOME DATA

I’m going to use data from the Gov 2001 Code library.

```
load("Gov2001CodeLibrary.RData")
```

This particular dataset is from the paper ”When preferences and commitments collide: The effect of relative partisan shifts on International treaty compliance.” in *International Organization* by Joseph Grieco, Christopher Gelpi, and Camber Warren.

THE MODEL

First, let's run their model:

```
fmla <- as.formula(restrict ~ art8 + shift_left + flexible +
   gnpcap + regnorm + gdpgrow + resgdp + bopgdp +
   useimfcr + surveil + univers + resvol + totvol +
   tradedep + military + termlim + parli + lastrest +
   lastrest2 + lastrest3)

# GLM is a function which performs our
# maximum likelihood estimation
fit <-glm(fmla, data=treaty1,
      family=binomial(link="logit"))
```

The Meat and Bread

$$\left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta})\right]^{-1} \left[\sum_{i=1}^{n} g_i(Y_i|\hat{\theta})' g_i(Y_i|\hat{\theta})\right] \left[-\sum_{i=1}^{n} h_i(Y_i|\hat{\theta})\right]^{-1}$$

The bread of the sandwich estimator is just the variance covariance matrix.

```
bread <-vcov(fit)
```

For the meat, we are going to use the `estfun` function to calculate the score.

```
library(sandwich)
est.fun <- estfun(fit)
```

Note: if estfun doesn't work for your glm, there is a way to do it using numericGradient().

The Sandwich

So we can create the sandwich

```
meat <- t(est.fun)%*%est.fun
sandwich <- bread%*%meat%*%bread
```

And put them back in our table

```
library(lmtest)
coeftest(fit, sandwich)
```

Note: For the linear case, estfun() is doing something a bit different than in the logit, so use:

```
robust <- sandwich(fit, meat=crossprod(est.fun)/N)
```

Clustered-Robust Standard Errors

First, we have to identify our clusters:

```
fc <- treaty1$imf_ccode
m <- length(unique(fc))
k <- length(coef(fit))
```

Then, we sum the meat by cluster

```
u <- estfun(fit)
u.clust <- matrix(NA, nrow=m, ncol=k)
for(j in 1:k){
u.clust[,j] <- tapply(u[,j], fc, sum)
}
dim(u) # n x k
dim(u.clust) # m x k
```

CLUSTERED-ROBUST STANDARD ERRORS

Last, we can make our cluster robust matrix:

```
cl.vcov <- bread %*% ((m / (m-1)) * t(u.clust)
    %*% (u.clust)) %*% bread
```

And add to our table

```
coeftest(fit, cl.vcov)
```

A couple notes

- There are easier ways to do this in R (see for example hccm).
- But it's good to know what is going on, especially when you are replicating.
- Beware: degrees of freedom corrections.

Quantities of Interest

Robust standard errors constrain your quantities of interest.

- Say you fit a linear model, but you think there is heteroskedasticity
- You want to simulate a quantity of interest, which means you have to simulate the y's.
- You *could* use the robust variance matrix to simulate the betas. (Although this seems weird)
- But how would you simulate fundamental uncertainty? You have the wrong stochastic component.

Coefficients vs. Standard Errors

We care more about the coefficients than the standard errors.

- If you think in theory you need robust standard errors, then your coefficients are probably off.
- In our field, we generally care about coefficients (especially their sign!) more than standard errors.
- Even more than coefficients, we care about other quantities of interest we can't get to if we use robust SE's.