# POLI 8501
# Binary Logit & Probit, II

The topic du jour is interpretation of binary response models (i.e. logit and probit estimates).

- Yes, their interpretation is harder (more involved, acutally) than OLS, BUT
- It's not *that* much harder.

We'll talk about a number of different approaches to interpreting these models. But, for now, remember three key points:

1. Nearly all of these approaches require one to be cognizant of "where we are on the curve".
2. When it comes to any kind of interpretation, a picture really is much more valuable than text or tables.
3. With very rare exceptions, it is never a good idea to present quantities of interest without their associated measures of uncertainty.

## A Running Example: House Voting on NAFTA

To motivate the discussion, we'll use a running example: The U.S. House of Representatives vote on the North American Free Trade Agreement (NAFTA). In 1993, the House voted to approve ratification of NAFTA by a margin of $234 - 200$. Our example data thus contain 435 observations and five variables:

1. `vote` - Whether ($= 1$) or not ($= 0$) the House member in question voted in favor of NAFTA.
2. `democrat` - Whether the House member in question is a Democrat($= 1$) or a Republican ($= 0$).
3. `pcthispc` - The percentage of the House member's district who are of Latino/Hispanic origin.
4. `cope93` - The 1993 AFL-CIO (COPE) voting score of the member in question; this variable ranges from 0 to 100, with higher values indicating more pro-labor positions.
5. `DemXCOPE` - The multiplicative interaction of `democrat` and `cope93`.

Our expectations are that:

- Higher COPE scores will correspond to lower probabilities of voting for NAFTA,
- Members from districts with higher numbers of Latinos will have higher probabilities of voting for NAFTA, but
- The effect of the former will be moderated by political party. In particular, the (negative) effect of COPE scores on pro-NAFTA voting will be greater for Democrats than for Republicans.

The relevant model, then, looks like:

$$\begin{aligned}\Pr(\texttt{vote}_i = 1) &= f[\beta_0 + \beta_1(\texttt{democrat}_i) + \beta_2(\texttt{pcthispc}_i) + \\ &\quad \beta_3(\texttt{cope93}_i) + \beta_4(\texttt{democrat}_i \times \texttt{cope93}_i) + u_i] \end{aligned} \tag{1}$$

The data look like this:

```
. su

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
        vote |       434    .5391705    .4990386          0          1
    democrat |       435     .583908    .4934767          0          1
    pcthispc |       435    8.786207    14.28133          0         83
      cope93 |       435    60.03908     39.2254          0        100
    DemXCOPE |       435    51.52644    45.56422          0        100
```

and we can estimate this model using (e.g.) the -logit- and/or -probit- commands in Stata :

```
. logit vote democrat pcthispc cope93 DemXCOPE

Logistic regression                               Number of obs   =        434
                                                  LR chi2(4)      =     162.16
                                                  Prob > chi2     =     0.0000
Log likelihood = -218.41388                       Pseudo R2       =     0.2707
------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    democrat |   6.865556   1.547357     4.44   0.000     3.832792    9.898319
    pcthispc |   .0209106    .007941     2.63   0.008     .0053466    .0364747
      cope93 |  -.0365007   .0075976    -4.80   0.000    -.0513917   -.0216097
    DemXCOPE |  -.0670544   .0182039    -3.68   0.000    -.1027334   -.0313754
       _cons |    1.79164   .2754383     6.50   0.000     1.251791    2.331489
------------------------------------------------------------------------------
```

For the rest of the class, we'll talk about a host of means for interpreting models, using these data as a running example.

## "Signs-N-Significance"

One alternative for interpretation is what I call "signs-n-significance": talk about the sign(s) of the estimated coefficient(s), and whether (and to what extent) that coefficient is statistically differentiable from zero.

- For all of the models we've discussed, this approach is no different than for OLS:
  - A positive estimate for $\hat{\beta}_X$ mean that increases in $X$ correspond to increases in $\Pr(Y = 1)$.
  - Likewise, a negative estimate for $\hat{\beta}_X$ mean that increases in $X$ correspond to decreases in $\Pr(Y = 1)$.
- Similarly, the ratio of $\hat{\beta}$ to its standard error is a $z$-score that can be used for hypothesis testing, etc.

So, in the example, we might note that (for Republicans), the estimate of the effect of `cope93` is negative (as expected), and that it is "statistically significant" (because its $z$-score is -4.77).

Note also that, because we have an interaction term in the model, the "direct effects" have a conditional interpretation. So, for example (referencing Equation 1), the estimated coefficient for the `cope93` variable when `democrat = 1` is:

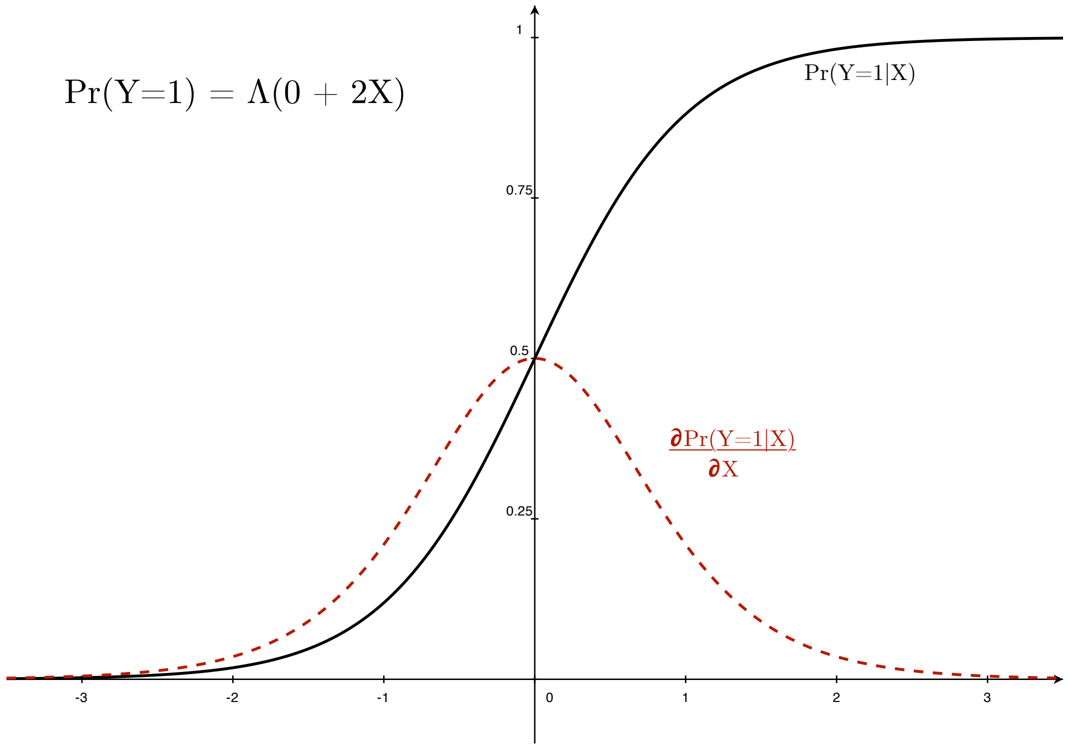
$$\hat{\beta}_{\texttt{cope93|democrat=1}} = \hat{\beta}_3 + \hat{\beta}_4$$

As with linear regression, we can obtain these "conditional" coefficient estimates – and their standard errors, $z$-scores, and confidence intervals – using the `-lincom-` command:

```
. lincom cope93 + DemXCOPE

 ( 1)  cope93 + DemXCOPE = 0

------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |  -.1035551   .0165809    -6.25   0.000    -.1360531   -.0710572
------------------------------------------------------------------------------
```

Thus, we can say that the effect of pro-union ideology – which was negative and significant for Republicans – is also negative, also significant, and roughly three times larger for Democrats.

Figure 1: $\Pr(Y=1)$ and $\frac{\partial \Pr(Y=1)}{\partial X}$ versus $X$ for $Y = \Lambda(0+2X)$



## Predicted Probabilities

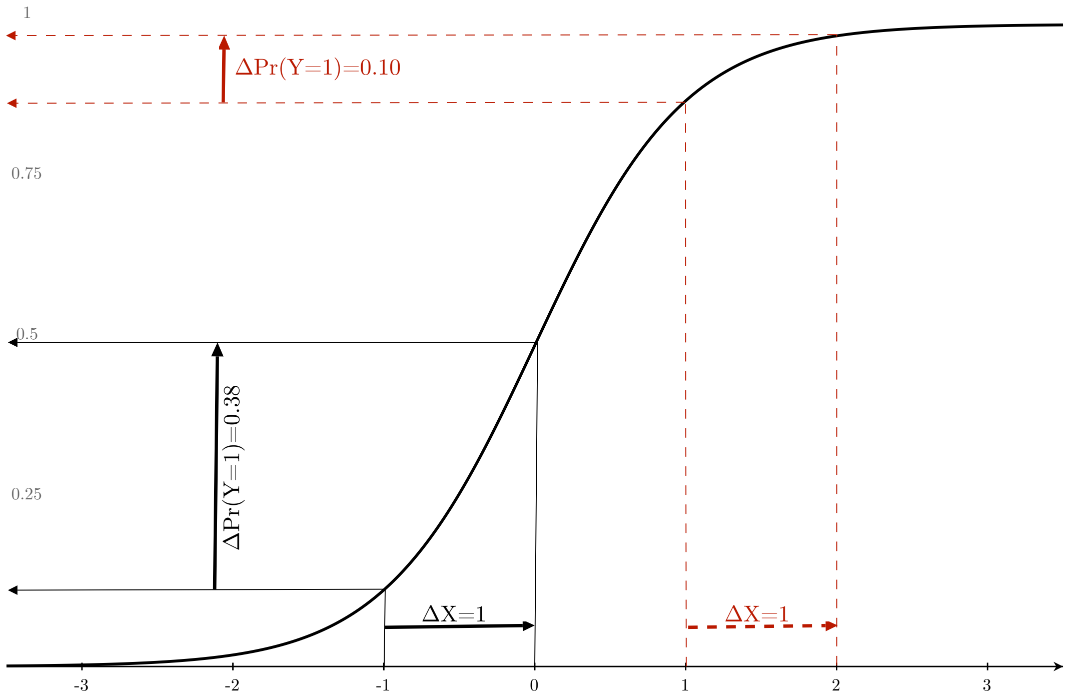
"Signs-n-Significance" is – to put it mildly – a rotten way to interpret this or any statistical model. What we really care about is, in most cases, the effect of changes in $\mathbf{X}$ on $\Pr(Y=1)$ – that is, on the probability of the actual event of interest. To get at this is a bit more involved than in the OLS case.

In all the binary-response models we've discussed, the effect of covariates is *linear in the latent variable* (that is, $Y^*$), but not in $Y$. *The real net effect of a change in $X$ depends critically on the values of the other $X$s and parameter estimates, and on the constant;* This is because the model is *nonlinear*. We can see this by noting that – unlike in a linear regression model – the first derivative of a logit/probit function depends on the value(s) of $X$ and $\hat{\beta}$. So, for example, in the case of a binary logit:

$$\frac{\partial \Pr(\hat{Y}_i = 1)}{\partial X_k} \equiv \lambda(X) = \frac{\exp(X_i\hat{\beta})}{[1+\exp(X_i\hat{\beta})]^2}\hat{\beta}_k \tag{2}$$

This non-constant first derivative is illustrated in Figure 1. As a practical matter, this means that if you're interested in the effect of a one-unit change in $X$ on $\Pr(Y_i = 1)$, how much change there is depends critically on "where you are on the curve."

Figure 2: Changes in $\Pr(Y = 1)$ for One-Unit Changes in $X$, for $Y = \Lambda(0 + 2X)$



To illustrate this, consider first a model with a single continuous covariate $X$:

$$\Pr(Y_i = 1) = \Lambda(\beta_0 + \beta_1 X_i) \tag{3}$$

where we have an estimated $\hat{\beta}_1 = 2.0$ and no intercept ($\hat{\beta}_0 = 0$). The change in $\Pr(Y = 1)$ associated with a one-unit change in $X$ varies depending on the "baseline" value of $X$, as well as on any other covariates' values in the model. So, for example, if we change $X = -1$ to $X = 0$, then the associated change in the predicted probability is:

$$\begin{aligned}
\frac{\exp(2 \times 0)}{1 + \exp(2 \times 0)} - \frac{\exp(2 \times -1)}{1 + \exp(2 \times -1)} &= \frac{\exp(0)}{1 + \exp(0)} - \frac{\exp(-2)}{1 + \exp(-2)} \\
&= \frac{1}{2} - \frac{0.14}{1.14} \\
&= 0.50 - 0.12 \\
&= \mathbf{0.38}
\end{aligned}$$

On the other hand, if we are going from $X = 1$ to $X = 2$, we get:

$$\begin{aligned}\frac{\exp(2\times 2)}{1+\exp(2\times 2)} - \frac{\exp(2\times 1)}{1+\exp(2\times 1)} &= \frac{\exp(4)}{1+\exp(4)} - \frac{\exp(2)}{1+\exp(2)} \\ &= \frac{7.39}{8.39} - \frac{1}{2} \\ &= 0.98 - 0.88 \\ &= \mathbf{0.10}\end{aligned}$$

These changes are illustrated graphically in Figure 2.

More generally, the change in $\Pr(Y = 1)$ associated with a (possibly multivariate) change in the values of the covariate vector $\mathbf{X}$ from $\mathbf{X}_A$ to $\mathbf{X}_B$ is:

$$\Delta\Pr(Y = 1)_{\mathbf{X}_A \to \mathbf{X}_B} = \frac{\exp(\mathbf{X}_B\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{X}_B\hat{\boldsymbol{\beta}})} - \frac{\exp(\mathbf{X}_A\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{X}_A\hat{\boldsymbol{\beta}})} \tag{4}$$

As a practical matter, all this means that you need to be careful of the values of the other variables in the model when assessing one variable's impact.

## Getting Predicted Probabilities

All predicted probabilities – whether in-sample or out-of-sample – take on the same general form:

$$\begin{aligned}\Pr(Y_i = 1) &= \frac{\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}})}{1+\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}})} \text{ for logit,} \\ &= \Phi(\mathbf{X}_i\hat{\boldsymbol{\beta}}) \text{ for probit.}\end{aligned}$$

Stata makes it very easy to obtain these predictions after estimating your regression. After running either model, just type:

```
. predict <varname>
```

...and Stata will create a variable called `varname` that contains the predicted probability of $Y_i = 1$ for each of the observations in whatever data are in memory. (Stata will also give that new variable a generic label, which you'll almost certainly want to change). Also useful is the fact that we can generate the predicted *index value* $\mathbf{X}_i\hat{\boldsymbol{\beta}}$ for each observation in the data, using:

```
. predict <varname>, xb
```

Likewise, we can calculate the *standard error of the linear prediction* – that is, the standard error of $\mathbf{X}_i\hat{\boldsymbol{\beta}}$ – using:

```
. predict <varname>, stdp
```

As we'll see, the latter are important when we go to use these predictions.[1]

## What Do We Do With Predictions?

First off, note that there are two types of predictions that we typically care about:

- *In-sample predictions* are simply the predicted probabilities of $Y_i = 1$ for the observations in the data on which the model was estimated.
- *Out-of-sample predictions* are predictions for cases that don't necessarily exist in the data, but which might be of interest (e.g., hypothetical cases).

### In-Sample Predictions

As we note above, Stata will automatically generate in-sample predictions and their associated measures of uncertainty using `-predict-`. For our NAFTA voting data, for example, we might use:

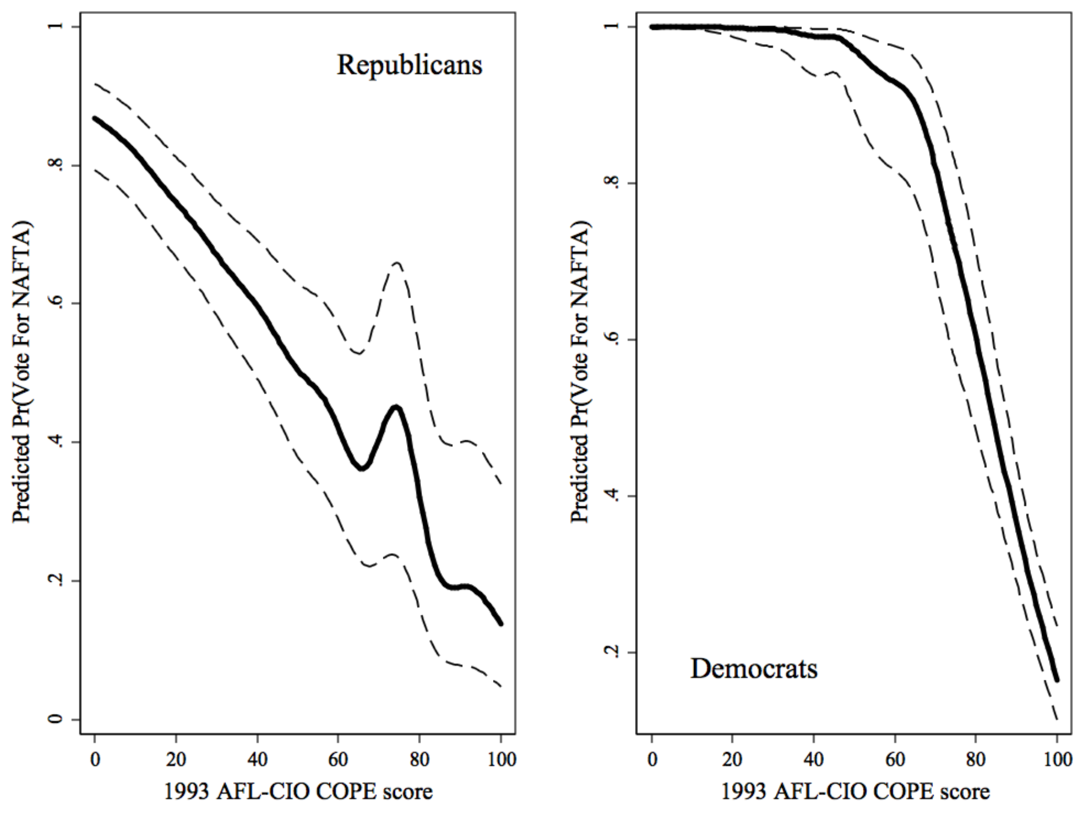
```
. logit ...

(output omitted)

. predict probhat
(option p assumed; Pr(vote))

. predict xbeta, index

. predict sehat, stdp

. gen Lindex=xbeta-invnorm(0.975)*sehat

. gen Uindex=xbeta+invnorm(0.975)*sehat

. gen L_prob=exp(Lindex) / (1+exp(Lindex))

. gen U_prob=exp(Uindex) / (1+exp(Uindex))
```

These can then be plotted,[2] as in Figure 3:

[1] As a matter of fact, there are lots of other things that `-predict-` will calculate for you after estimation of probit or logit models; these include various kinds of residuals, influence statistics (to check for outliers), score values, etc. We won't go into those here; check `-help predict-` in Stata if you're curious...

[2] For the interested, the Stata commands to do so are available at the end of these notes.

Figure 3: Smoothed In-Sample $\widehat{\Pr(Y=1)}$s and 95% Confidence Intervals



These plots are based on the actual data, and so are very "noisy;" notice that I use a median spline smoother to make the plots smoother (and prettier). The plot in Figure 3 is informative; for example, it clearly reflects the finding that the effect of changes in `cope93` are greater for Democrats than for Republicans.

## Out-Of-Sample Predictions

We also use the `-predict-` command to do out-of-sample predictions. The key to dealing with out-of-sample predictions is in selecting the variables of most interest to us, and then holding the other variables in the model constant at some particular level. To do this, we usually choose the mean for continuous variables, and the median or mode for others (since the mean of a dichotomous variable is a nonsensical value). We then use these to calculate an *index value*: the value of the combination of the covariate mean/medians and the estimated coefficients (i.e., $\sum_{k=1}^{K} \overline{X}_k \hat{\beta}_k$); to get the estimated mean probability of a positive outcome, we take the logit transform of this.

Presenting out-of-sample predictions usually involves one of two methods: either we report the predictions in *tabular* format, or we *plot* the predicted probabilities. 99 percent of the time, the latter is better than the former, but we'll discuss both here.

*Tables of Predictions*

The method for generating tables of predicted probabilities is pretty straightforward:

1. Calculate the "index value" for a "mean" (typical) observation.
2. Choose some amount by which you want to change the value of the independent variable in question.
    - Standard deviations are good – they make results across covariates a little more comparable. But,
    - They may also be some substantively interesting value(s).
3. Calculate the "index value" for two (or more) alternative observations – say, one $\sigma$ above and below the mean.
4. From these, calculate the associated predicted probabilities, and report them.

All of this can be done (e.g.) in a spreadsheet, to make your life easier.

*Graphs of Predicted Probabilities*

Generally, plots of predicted probabilities are a (much) better option than tables. First, pick out a variable of interest: Let's assume here that we are interested in plotting graphically the different effects of changes in COPE scores on Democrats' and Republicans' votes for NAFTA. To do this, we start with a "dummy" (simulated) dataset that contains all the variables used in the original analysis:

```
. estat summ

  Estimation sample logit               Number of obs =    434
  -------------------------------------------------------------
      Variable |         Mean     Std. Dev.       Min        Max
  -------------+-----------------------------------------------
          vote |     .5391705     .4990386          0          1
      democrat |     .5852535     .4932468          0          1
      pcthispc |     8.799539      14.2951          0         83
        cope93 |     60.17742     39.16429          0        100
      DemXCOPE |     51.64516     45.54939          0        100
  -------------------------------------------------------------
```

```
. clear

. set obs 101
obs was 0, now 101

. gen democrat=0

. gen pcthispc=8.8

. gen cope93=_n-1

. expand 2
(101 observations created)

. sort cope93

. quietly by cope93 : replace democrat=democrat[_n-1]+1 if democrat[_n-1]~=.

. gen DemXCOPE = democrat*cope93

. summarize

    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+-----------------------------------------------------
    democrat |       202          .5    .5012422          0          1
    pcthispc |       202         8.8           0        8.8        8.8
      cope93 |       202          50    29.22719          0        100

. save NAFTAsim
file NAFTAsim.dta saved
```

These simulated data contain all the (independent) variables in the original analysis. Note that we have set the `pcthispc` variable to its mean value. We also have 101 observations in which `democrat`=0 and `cope93` ranges from 0 to 100, and another 101 observations in which `democrat`=1 and `cope93` ranges from 0 to 100.

Next, we rerun the original model, then use `-predict-` on our simulated data to generate predicted probabilities and their standard errors, in a manner analogous to what we did in-sample:

```
. use NAFTA.dta
. logit vote democrat pcthispc cope93 DemXCOPE

(output omitted...)

. use NAFTAsim

. predict Probhat
(option p assumed; Pr(vote))

. predict XBeta, xb

. predict seXB, stdp

. gen Lindex=XBeta-invnorm(0.975)*seXB

. gen Uindex=XBeta+invnorm(0.975)*seXB

. gen L_prob=exp(Lindex) / (1+exp(Lindex))

. gen U_prob=exp(Uindex) / (1+exp(Uindex))
```

We now have the predicted probabilities and their 95 percent confidence intervals for the range of `cope93` values, for both Democrats and Republicans. We can use these data to create plots, like the one in Figure 4.[3]

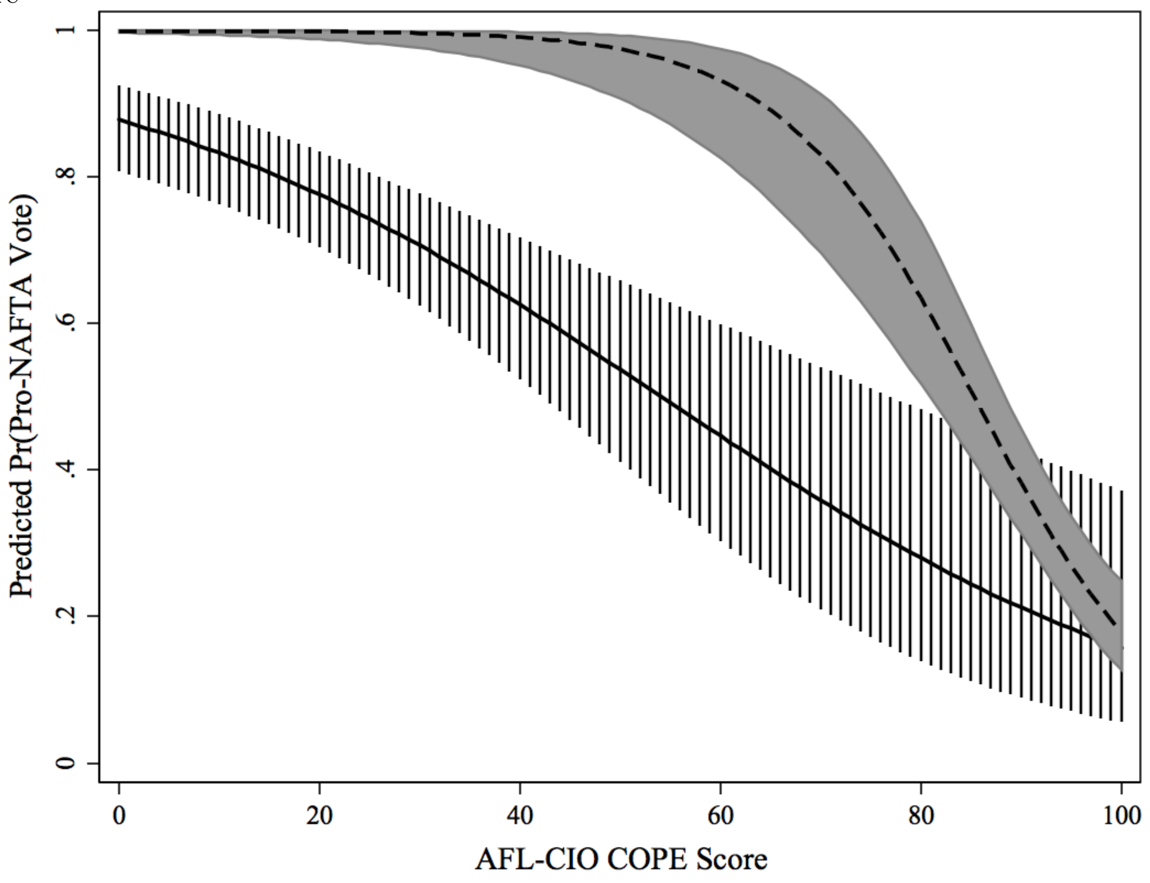
Figure 4 tells us that, with all other variables at their mean values, the predicted probability of a pro-NAFTA vote decreases significantly as a member's COPE score increases. However, the magnitude of that decrease is significantly larger for Democrats than for Republicans. One interpretation of this, then, is that Democrats are (as one might expect) more responsive to district-level union concerns than are Republicans.

The confidence intervals in Figure 4 are also useful. They tell us that, over most of the range of COPE scores, Democrats and Republicans are quite different in their probability of voting for NAFTA. At high levels of COPE scores, however, they converge. Thus, pro-union Democrats and Republicans tended to vote alike (that is, against NAFTA), while those less favorable toward unions behaved differently.

There are other things one can do with predicted probabilities, including 3-D plots (which are very useful for interactions of two continuous variables). But that's probably enough for now; let's move on.

[3]Once again, the command for this graph is in the Appendix at the end of these notes.

Figure 4: Predicted Probabilities of a Pro-NAFTA Vote, by Party Identification and COPE Score



Note: Solid line is predicted probabilities for Republicans; dashed line for Democrats. Regions indicate 95% pointwise confidence intervals for the predictions.

## Odds Ratios and the Logit Model

**Odds ratios** are an easy way of substantively interpreting a logit model. Consider the **"odds"** of $Y = 1$ for a given observation with some values of $\mathbf{X}$:

$$\Omega(\mathbf{X}) = \frac{\Pr(Y = 1|\mathbf{X})}{\Pr(Y = 0|\mathbf{X})} = \frac{\Pr(Y = 1|\mathbf{X})}{1 - \Pr(Y = 1|\mathbf{X})} = \frac{\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}}{1 - \frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}} \tag{5}$$

So, if $\Pr(Y = 1) = 0.50$, then the odds are 1 to 1; for $\Pr(Y = 1) = 0.75$, the odds are 3 to 1, etc. Note that the odds range from zero to infinity (but will never be negative).

If $\Omega$ is the odds, then $ln\Omega$ is the "log-odds" (which is also known as the *logit*), which ranges from negative infinity to infinity. This means that:

$$ln\Omega(\mathbf{X}) = ln\left[\frac{\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}}{1-\frac{\exp(\mathbf{X}\boldsymbol{\beta})}{1+\exp(\mathbf{X}\boldsymbol{\beta})}}\right] = \mathbf{X}\boldsymbol{\beta} \tag{6}$$

That is – and as we noted last class – in the logit model, the log-odds are linear in $\mathbf{X}$. This means that if we consider the effect of a change in $\mathbf{X}$ on the log-odds of $Y$, we get:

$$\frac{\partial ln\Omega}{\partial \mathbf{X}} = \boldsymbol{\beta} \tag{7}$$

In other words, the estimate $\hat{\beta}_k$ from our logit equation tells us the change in the log-odds which accompanies a one-unit change in $X_k$.

But normal people don't think in terms of log-odds,[4] they think in terms of odds. (7) means that the change in the odds of $Y = 1$ associated with a one-unit change in $X_k$ is:

$$\frac{\Omega(X_k + 1)}{\Omega(X_k)} = \exp(\hat{\beta}_k) \tag{8}$$

More generally,

$$\frac{\Omega(X_k + \delta)}{\Omega(X_k)} = \exp(\hat{\beta}_k\delta) \tag{9}$$

As a practical matter, Equation (8) means that we can interpret the exponentiated coefficients of a logit model as the change in the odds of $\Pr(Y = 1)$ associated with a one-unit change in $X_k$. This translates easily to a percentage change in the odds as well:

$$\text{Percentage Change} = 100[\exp(\hat{\beta}_k\delta) - 1] \tag{10}$$

Thus, for example:

- For a logit estimate of $\hat{\beta} = 2.3$, a unit change in $X$...
    - ...corresponds to an increase in the log-odds of $Y = 1$ of 2.3, or
    - ...a change in the odds that $Y = 1$ of $\exp(2.3) = 9.974$, or
    - ...a percentage change in the odds that $Y = 1$ of $100[\exp(\hat{\beta}) - 1] = 897$ percent.
- For a logit estimate of $\hat{\beta} = -0.22$, an 11–unit change in $X$...
    - ...corresponds to a $-0.22 \times 11 = -2.42$ decrease in the log-odds of $Y = 1$, or
    - ...a change in the odds that $Y = 1$ of $\exp(-0.22 \times 11) = \exp(-2.42) = 0.089$, or

[4] If I took you to the track, and told you the log-odds of a horse coming in win, place or show were -1.95, would you bet on it? (That's 7 to one, for you race fans...).

- ...a percentage change in the odds that $Y = 1$ of $100[\exp(-0.22 \times 11) - 1] = 100(0.089 - 1) = -91.1$ percent.

Odds ratios are thus an easy, intuitive way to interpret logit coefficients. Moreover, for Stata users out there, the software will report odds ratios rather than $\hat{\boldsymbol{\beta}}$s – along with their standard errors, $z$-scores, and confidence intervals – automatically; all you have to do is ask:

```
. logit, or

Logistic regression                               Number of obs   =        434
                                                  LR chi2(4)      =     162.16
                                                  Prob > chi2     =     0.0000
Log likelihood = -218.41388                       Pseudo R2       =     0.2707

------------------------------------------------------------------------------
        vote | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    democrat |   958.6783   1483.417     4.44   0.000     46.19134    19896.89
    pcthispc |   1.021131   .0081088     2.63   0.008     1.005361    1.037148
      cope93 |   .9641574   .0073253    -4.80   0.000     .9499065    .9786221
    DemXCOPE |   .9351443   .0170233    -3.68   0.000     .9023675    .9691117
------------------------------------------------------------------------------
```

This also extends to ancilliary/post-estimation commands, such as `-test-` and `-lincom-`, e.g.:

```
. lincom cope93 + DemXCOPE, eform

 ( 1)  cope93 + DemXCOPE = 0

------------------------------------------------------------------------------
        vote |     exp(b)   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
         (1) |   .9016263   .0149498    -6.25   0.000     .8727963    .9314086
------------------------------------------------------------------------------
```

Finally, in using odds ratios, remember that:

- Percentage decreases in odds are bounded at 100 (naturally), but have no upper bound. That means that...
- ...if $\exp(\hat{\beta}_k \delta) < 1$, the we would say that the odds of $Y = 1$ are "only $100[\exp(\hat{\beta}_k \delta) - 1]$ percent of those for cases with $X = X_0 + \delta$, versus those with $X_0$."
- If $\exp(\hat{\beta}_k \delta) > 1$, the we would say that the odds of $Y = 1$ are "$100[\exp(\hat{\beta}_k \delta) - 1]$ percent of those for cases with $X = X_0 + \delta$, versus those with $X_0$."

## Clarify

### What it is

Clarify is a program (actually, a set of programs) written by political scientists Gary King, Mike Tomz, and Jason Wittenberg. It is implemented in Stata , and is designed to allow researchers to easily interpret regression-like models in a more intuitive (and, we hope, correct) way. Clarify does this through the use of simulations of parameters and other quantities of interest; the relevant citation is:

King, Gary, Michael Tomz and Jason Wittenberg. 2000. "Making the Most of Statistical Analyses: Improving Interpretation and Presentation." *American Journal of Political Science*, 44(April): 341-55.

You can check out the documentation yourself at

`http://gking.harvard.edu/clarify/docs/clarify.html`.

### How to use it

There are three main commands that one uses, almost always in this sequence:

1. `-estsimp-` is a "wrapper" that goes before a standard regression-like command, and causes Stata to follow estimating the model by generating 1000 (or however many you choose) draws from the asymptotic posterior distribution of the parameter estimates (typically, a multivariate Normal distribution with means equal to the estimates and variance-covariance matrix equal to the estimated VCV matrix – that is, it draws 1000 vectors $\tilde{\theta}$ from $\text{MVN}(\hat{\theta}, \widehat{\mathbf{VCV}(\hat{\theta})})$). You can then (e.g.) plot these simulations to get an idea of what your parameter estimates "look like" (remember: $\hat{\beta}$ is a *random variable* – so you ought to be (at least) as interested in its dispersion around the point estimate as you are in the point estimate's value itself...).
2. `-setx-` allows you to set the values of the various independent variables – to their means, medians, etc. or even to specific values you choose.
3. `-simqi-` **sim**ulates (and generates) **q**uantities of **i**nterest – things like expected values of $Y$, predicted probabilities of various kinds, and so forth.

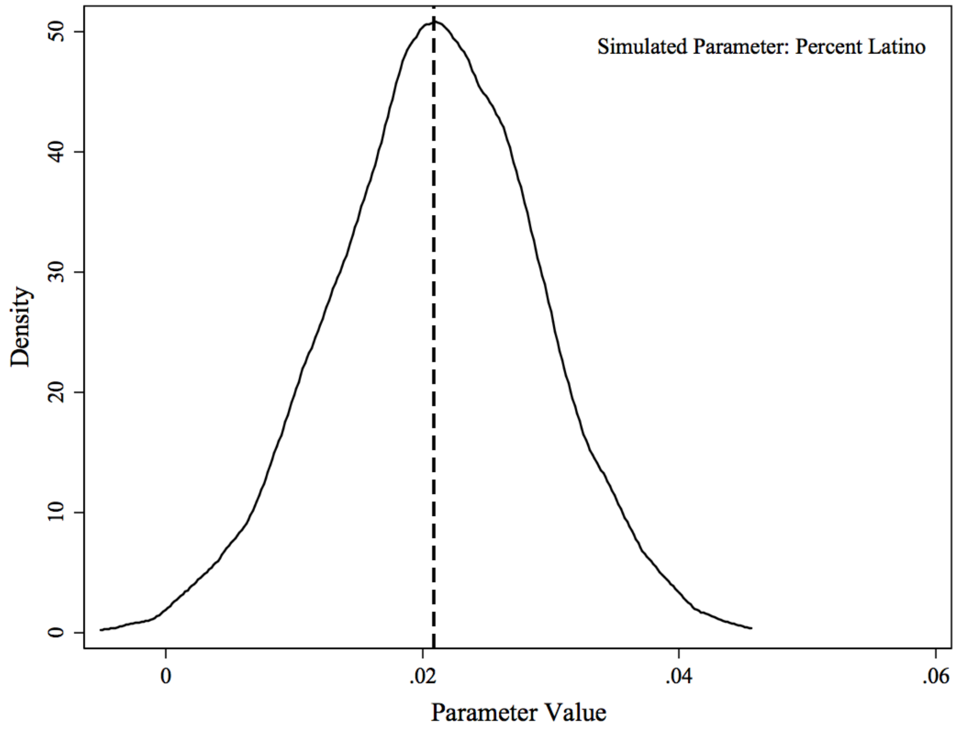
### An Example

We'll go back to our NAFTA voting example again...

First, estimate the same logit model we did before, but this time with the `-estsimp-` wrapper on it:

```
. estsimp logit vote democrat pcthispc cope93 DemXCOPE

Logistic regression                               Number of obs   =        434
                                                  LR chi2(4)      =     162.16
                                                  Prob > chi2     =     0.0000
Log likelihood = -218.41388                       Pseudo R2       =     0.2707
------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    democrat |   6.865556   1.547357     4.44   0.000     3.832792    9.898319
    pcthispc |   .0209106    .007941     2.63   0.008     .0053466    .0364747
      cope93 |  -.0365007   .0075976    -4.80   0.000    -.0513917   -.0216097
    DemXCOPE |  -.0670544   .0182039    -3.68   0.000    -.1027334   -.0313754
       _cons |    1.79164   .2754383     6.50   0.000     1.251791    2.331489
------------------------------------------------------------------------------

Simulating main parameters.  Please wait....

Note: Clarify is expanding your dataset from 435 observations to 1000
observations in order to accommodate the simulations.  This will append
missing values to the bottom of your original dataset.

% of simulations completed: 20% 40% 60% 80% 100%

Number of simulations  : 1000
Names of new variables : b1 b2 b3 b4 b5

. su
    Variable |       Obs        Mean    Std. Dev.       Min        Max
-------------+--------------------------------------------------------
          b1 |      1000     6.86644    1.550493   1.928434   11.33249
          b2 |      1000     .0210454   .0078827  -.0051356   .0456498
          b3 |      1000    -.0362176   .0075131   -.055486  -.0100977
          b4 |      1000    -.0673823   .0184587  -.1221915  -.0107905
          b5 |      1000     1.791742   .2713336    .856936   2.580538
-------------+--------------------------------------------------------
        vote |       434     .5391705   .4990386          0          1
    democrat |       435      .583908   .4934767          0          1
    pcthispc |       435     8.786207   14.28133          0         83
      cope93 |       435     60.03908    39.2254          0        100
    DemXCOPE |       435     51.52644   45.56422          0        100
-------------+--------------------------------------------------------
```

Figure 5: Simulated `pcthispc` Parameters, Using 𝔆larify



Notice that 𝔆larify *changed your data* (keep this in the very front of your mind whenever you use 𝔆larify). The different *b*s are simulated parameters for the variables in the `-logit-` (notice how they have more-or-less the same means and standard deviations as the estimates?). Sometimes it's keen to graph these things, like in Figure 5; there, the logit point estimate is indicated by the dashed line.

Next, we use `-setx-` to set the variables to some values of interest (like, say, their means or medians):

```
. setx democrat median pcthispc mean cope93 mean DemXCOPE mean
```

This doesn't appear to do anything, but its actually a very important step.

Finally, we use `-simqi-` to generate some things we care about, and/or that make interpreting the model easier and more fun. This is where 𝔆larify really becomes useful: once you've finalized your model, its really easy to generate all kinds of useful pieces of information about your model results using `-simqi-`. For example:

- Predicted probabilities, at the values of $X$ at which you `-setx-`:

```
. simqi

  Quantity of Interest |     Mean       Std. Err.    [95% Conf. Interval]
-----------------------+--------------------------------------------------
            Pr(vote=0) |   .0485171     .0334052     .0113376     .138122
            Pr(vote=1) |   .9514829     .0334052      .861878    .9886624
```

- The difference in the predicted probability of a pro-NAFTA vote between Democrats and Republicans (assuming `cope93` $= 0$):

```
. simqi fd(pr) changex(democrat 0 1)

First Difference: democrat 0 1

  Quantity of Interest |     Mean       Std. Err.    [95% Conf. Interval]
-----------------------+--------------------------------------------------
         dPr(vote = 0) |  -.9167421     .0660248    -.9839994   -.7350486
         dPr(vote = 1) |   .9167421     .0660248     .7350486    .9839994
```

- The change in $\widehat{\Pr(Y=1)}$, changing the `pcthispc` from scores of 5 to 40:

```
. simqi fd(pr) changex(pcthispc 5 40)

First Difference: pcthispc 5 40

  Quantity of Interest |     Mean       Std. Err.    [95% Conf. Interval]
-----------------------+--------------------------------------------------
         dPr(vote = 0) |  -.0250473     .0181051    -.0727688   -.0044629
         dPr(vote = 1) |   .0250473     .0181051     .0044629    .0727688
```

- The change in $\widehat{\Pr(Y=1)}$ one gets from going from a 0% pro-labor Republican to a 100% pro-labor Democrat:

```
. simqi fd(pr) changex(democrat 0 1 cope93 0 100)

First Difference: democrat 0 1 cope93 0 100

  Quantity of Interest |     Mean       Std. Err.    [95% Conf. Interval]
-----------------------+--------------------------------------------------
         dPr(vote = 0) |  -.6005153     .2683254    -.9361235    .0607074
         dPr(vote = 1) |   .6005153     .2683254    -.0607074    .9361235
```

There are lots of other possible uses for 𝔆larify; give it a look...

## Goodness-of-Fit

As in linear models, it's useful to present measures of how well (or poorly) the model fits the data in the aggregate. We'll talk about three general ways of conveying this information:

- Pseudo-$R^2$,
- Proportional reduction in error (PRE), and
- ROC curves.

### Pseudo-$R^2$

- There are several different flavors of these...
- They can be based on likelihood ratios, predicted values, etc.
- They are not always all that useful (see the Hagle and Mitchell article), in that they can have odd statistical properties and rarely have an intuitive substantive interpretation.
- IMO, PRE (see below) is generally a better (and more intuitive) summary for goodness-of-fit in these models.

### Proportional Reduction in Error (PRE)

The intuition behind PRE is to answer the question, "How much better does my model do than dumb guessing?" First, let's use our predicted probabilities to generate some (binary) predictions about how members will vote on NAFTA:

```
. gen VoteHat=.
(435 missing values generated)

. replace VoteHat = 0 if probhat<.5 & probhat~=.
(197 real changes made)

. replace VoteHat = 1 if probhat>.5 & probhat~=.
(238 real changes made)
```

Now we can compare our predictions to the actual values, to see how well we did:

```
. tab2 vote VoteHat, col

-> tabulation of vote by VoteHat

1=vote for |
    NAFTA, |
    0=vote |        VoteHat
    agains |         0          1 |     Total
-----------+----------------------+----------
         0 |       148         52 |       200
           |     75.13      21.94 |     46.08
-----------+----------------------+----------
         1 |        49        185 |       234
           |     24.87      78.06 |     53.92
-----------+----------------------+----------
     Total |       197        237 |       434
           |    100.00     100.00 |    100.00
```

Now think about how our model is doing...

- If we were to take a guess at $\hat{Y}$ *without any information on the $X$s*, the best we could do would be to always guess the median category.
  - We'd guess that *everyone* was going to vote for NAFTA.
  - We'd be right $\frac{234}{434} = 53.5\%$ of the time.
  - BUT we could do a lot better – think of the 200 votes we would get wrong as our "room for improvement".
- Our model predicts 148 of the 200 "nos" correctly (74.0%), and 185 of the 234 "yess" (79.1%).
- All together, our model gets (148+185 =) 333 (or $\frac{333}{434} = 76.7\%$) of the votes correct, or (333 - 234 =) 99 *more* votes right than the "null model."
- So, our proportional reduction in error is $\frac{333-234}{200} = \frac{99}{200} = \mathbf{49.5\%}$.
- That is, our model eliminated nearly 50% of the error, relative to the "null model."

More generally, the formula for PRE is:

$$\text{PRE} = \frac{N_{MC} - N_{NC}}{N - N_{NC}} \tag{11}$$

where $N_{NC}$ is the number correct under the "null model," $N_{MC}$ is the number correct under the estimated model, and $N$ is the total number of observations.

Note a few things about this...

- Its *possible* (but unlikely) to have a negative PRE...
- The maximum PRE is (of course) 100 percent.
- The PRE doesn't *necessarily* say anything about aggregate outcomes, though obviously the better your PRE, the closer your model will come to getting the aggregate outcome "right.".

It is commonplace in the social sciences to report PRE as a substitute for $R^2$ in binary-response models, usually along with the overall proportion of observations correctly classified. Moreover, when more than one model is involved (e.g., different specifications of covariates), PRE can be a useful way to get an intuitive sense of which model(s) are performing better or worse, in predictive terms.

As it happens, Stata will automatically calculate a number of the statistics we just discussed (though not PRE itself), via the post-estimation command `-estat clas-`:

```
. estat clas

Logistic model for vote

              -------- True --------
Classified |         D            ~D  |      Total
-----------+--------------------------+-----------
     +     |       185            52  |        237
     -     |        49           148  |        197
-----------+--------------------------+-----------
   Total   |       234           200  |        434

Classified + if predicted Pr(D) >= .5
True D defined as vote != 0
--------------------------------------------------
Sensitivity                     Pr( +| D)   79.06%
Specificity                     Pr( -|~D)   74.00%
Positive predictive value       Pr( D| +)   78.06%
```

```
Negative predictive value       Pr(~D| -)   75.13%
--------------------------------------------------
False + rate for true ~D        Pr( +|~D)   26.00%
False - rate for true D         Pr( -| D)   20.94%
False + rate for classified +   Pr(~D| +)   21.94%
False - rate for classified -   Pr( D| -)   24.87%
--------------------------------------------------
Correctly classified                        76.73%
--------------------------------------------------
```

This introduces some useful jargon for us:

- *Sensitivity* is the probability of a positive prediction given a positive (actual) outcome; that is, the proportion of all "1s" that the model classifies as / predicts to be "1." These are also called "true positives," or the "true positive fraction;" as we noted, in our example, this is $185/234 \approx 0.79$.
- *Specificity* is the same thing, but for negative (zero) responses: the proportion of (actual) zero outcomes that the model classifies as zeros. These are known as "true negatives," or the "true negative fraction;" in our example, this is $148/200 \approx 0.74$.
- Given these definitions, the quantity $1-$*Sensitivity* is therefore the fraction of "false positives" (predicted "1s" with actual "0s"), and likewise $1-$*Specificity* is the fraction of "false positives" (predicted "0s" with actual "1s").

As we'll see in a bit, these values are useful for calculating ROC curves, which are an alternative (graphical) indication of goodness-of-fit.

### ROC Curves

ROC stands for "receiver operating characteristic."[5] It can be thought of as a plot of the true positive rate against the false positive rate for a particular classification model. It is also, however, a useful way to assess model fit for binary response models.

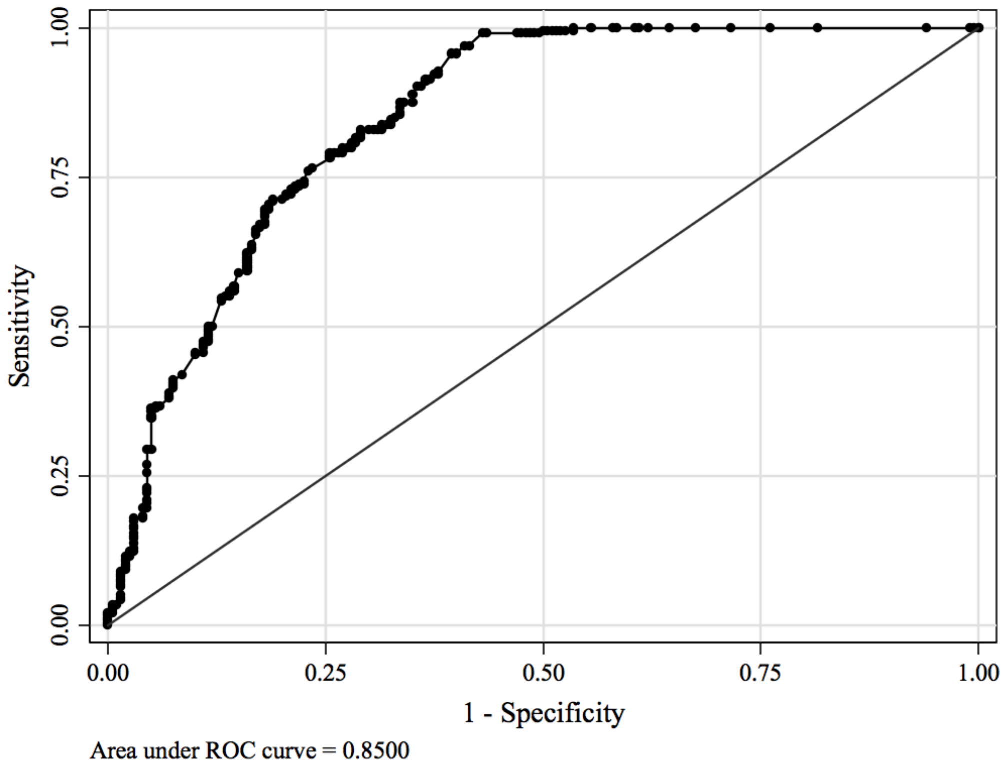
As an example, we can plot the ROC curve for our NAFTA model:

```
. lroc, recast(connected) msymbol(smcircle) mcolor(black) msize(medsmall)
```

which yields

[5]The name comes from signal detection theory; ROCs were developed during World War II, by mathematicians and statisticians working with radar operators. A nifty little web-based introduction to ROC curves can be found at http://www.anaesthetist.com/mnm/stats/roc/Findex.htm.

Figure 6: ROC Curve, NAFTA Logit Analysis



Note that there is necessarily a tradeoff between true positives and false positives, but that that tradeoff depends on the quality of the model at classifying outcomes. We can see this tradeoff by picking a level/fraction of positives we wish to find, and then tracing out the corresponding level of false positives the model will give us:

- If we chose *Sensitivity* = 0.50, then we can expect the model to have a false-positive rate of roughly 0.12.
- If we increase the *Sensitivity* to 0.75, the corresponding false positive rate is about 0.24.

A "perfect" test – one that classified all the actual zeros as zeros, and the actual ones as ones – would have an ROC curve that went up the left-hand side (at $1-\textit{Specificity} = 0$) and then straight across the top (at *Sensitivity* = 1). Conversely, a model that did no better than chance (i.e., a 50-50 model) would have an ROC that was a 45-degree line, since all "true positives" would also have a correspondingly equal number of "false positives."

This suggests a role for the *area under the ROC curve* (sometimes called the "aROC"). The area under the ROC curve is necessarily bounded between 0.5 (the area of the $\Re_2$ unit space under the diagonal line) and 1.0 (for a test that perfectly predicts both positive and negative outcomes). The area measures *discriminatory power* of the model; the "better" the model does at classifying both positive and negative outcomes, the higher the amount of area under the ROC curve.

A mathematically identical (but somewhat more intuitive) interpretation of the area under an ROC curve is as the average concordance between the model predictions and reality in repeated sampling. In our example, imagine randomly picking one observation from the pro-NAFTA voters, and one from the anti-NAFTA group. If the model is a good one, the pro-NAFTA House member should have a higher predicted probability of voting for NAFTA than the anti-NAFTA representative. The area under the ROC curve is the total percentage of randomly drawn pairs for which this is true (that is, that the test correctly classifies the two observations in the random pair).

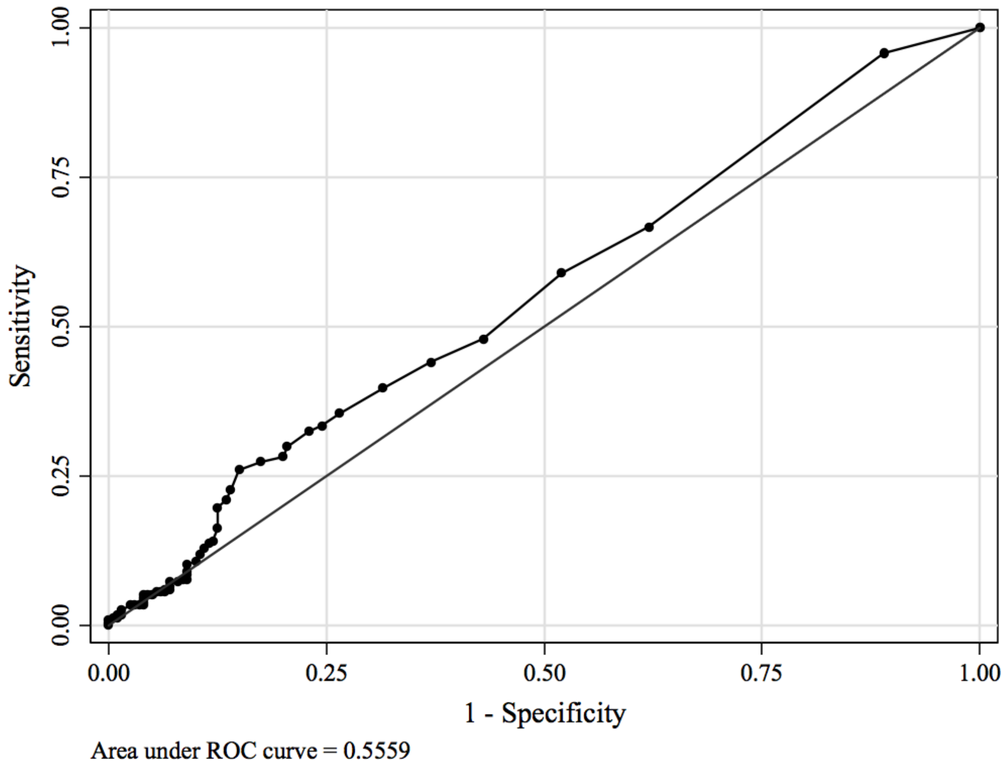
Given this interpretation, it's easy to see how the area under the ROC curve is useful as a summary measure of model fit. A good set of rules of thumb for substantively interpreting the area under the ROC curve is:

- Area under ROC = 0.90-1.00 $\rightarrow$ Excellent (A)
- Area under ROC = 0.80-0.90 $\rightarrow$ Good (B)
- Area under ROC = 0.70-0.80 $\rightarrow$ Fair (C)
- Area under ROC = 0.60-0.70 $\rightarrow$ Poor (D)
- Area under ROC = 0.50-0.60 $\rightarrow$ Total Failure (F)

By these criteria, our NAFTA model is a good-fitting model; by contrast, a model that included only the `pcthispc` variable as a covariate fares much worse:

```
. logit vote pcthispc

Logistic regression                               Number of obs   =        434
                                                  LR chi2(1)      =       1.19
                                                  Prob > chi2     =     0.2748
Log likelihood =  -298.8964                       Pseudo R2       =     0.0020
------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    pcthispc |   .0074779   .0069253     1.08   0.280    -.0060955    .0210513
       _cons |   .0919126   .1132522     0.81   0.417    -.1300576    .3138828
------------------------------------------------------------------------------

. predict hispxbeta, xb

. lroc, recast(connected) msymbol(smcircle) mcolor(black) msize(medsmall)
```

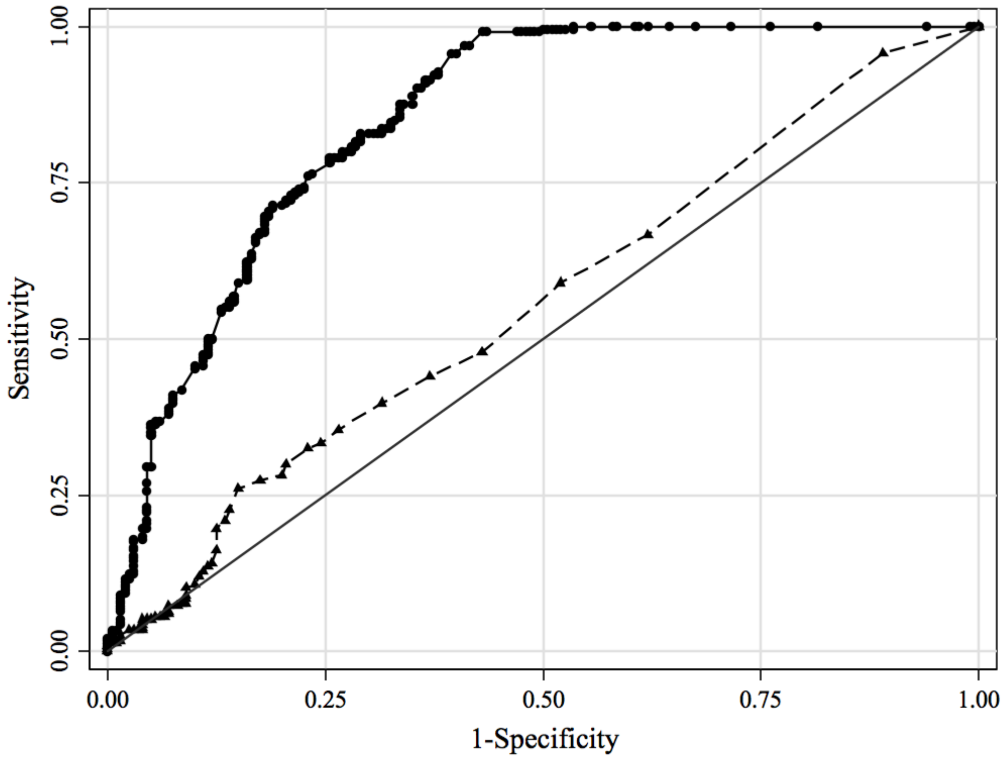
Figure 7: ROC Curve, NAFTA Logit Analysis Using `pcthispc` Only



Given the interpretation of the area under the ROC curve described above, it seems like it ought to be possible to do a formal statistical test of whether the areas under the ROC curve for two different models are statistically differentiable or not. And, in fact, we can (and it's not even that hard). The mathematical details of how this is done depend on whether the ROC (and thus the area under it) is calculated via trapezoidal approximation or using MLE; we need not go into that too much here.

To accomplish the comparison, we can use the aptly-named `-roccomp-` (for "ROC comparison") command:

Figure 8: Comparing ROC Curves, NAFTA Data



```
. roccomp vote xbeta hispxbeta, graph summary plot1opts(recast(connected)
msymbol(smcircle) mcolor(black) msize(medsmall)) plot2opts(recast(connected)
lcolor(black) lpattern(dash) msymbol(smtriangle) mcolor(black) msize(small))
legend(off)
                               ROC                    -Asymptotic Normal--
                    Obs       Area     Std. Err.      [95% Conf. Interval]
-------------------------------------------------------------------------
xbeta               434     0.8500       0.0190        0.81283     0.88725
hispxbeta           434     0.5559       0.0273        0.50233     0.60948
-------------------------------------------------------------------------
Ho: area(xbeta) = area(hispxbeta)
    chi2(1) =    64.51       Prob>chi2 =   0.0000
```

This tells us that we can confidently reject the null hypothesis that the area under the ROC (and thus the predictive fit) of the two models is equal.

## A Few Final Things

Note that this summary hardly exhausts the possible approaches for interpreting binary-response models. To mention just a few things we have not – thanks to the need for speed – covered:

- **Marginal Effects**. That is,

$$\frac{\partial \Pr(Y = 1)}{\partial X_k} \tag{11}$$

  which, as we noted above, is *not* simply equal to $\hat{\beta}_k$. These can be a flexible, useful way of discussing covariate effects; and, as it happens, there's a simple post-estimation command in Stata (`-mfx-`) for calculating them. Read up on that if it sounds interesting.

- **Additional Goodness-of-Fit Tests**. These include those by Pearson, Hosmer and Lemeshow (1989), and many others, as well as the standard LR, AIC, and BIC tests.

- **Residual Analysis**, including influence statistics, outlier detection, etc.

**Next time**: Models for funny-looking binary responses of various kinds...

## Appendix: Stata Commands for Figures 3 and 4

### Figure 3

```
. twoway (mspline probhat cope93 if democrat==0, lcolor(black) lpattern(solid)
lwidth(thick)) (mspline U_prob cope93 if democrat==0, lcolor(black) lpattern(dash))
(mspline L_prob cope93 if democrat==0, lcolor(black) lpattern(dash)), ytitle(Predicted
Pr(Vote For NAFTA)) note(Republicans, size(large) position(2) ring(0)
margin(medlarge)) legend(off)

. graph save "GOPinsample.gph"

. twoway (mspline probhat cope93 if democrat==1, lcolor(black) lpattern(solid)
lwidth(thick)) (mspline U_prob cope93 if democrat==1, lcolor(black) lpattern(dash))
(mspline L_prob cope93 if democrat==1, lcolor(black) lpattern(dash)), ytitle(Predicted
Pr(Vote For NAFTA)) note(Democrats, size(large) position(7) ring(0) margin(large))
legend(off)

. graph save "DEMinsample.gph"

. graph combine "GOPinsample.gph" "DEMinsample.gph"
```

### Figure 4

```
. twoway (rspike U_prob L_prob cope93 if democrat==0, lcolor(black) lpattern(solid))
(line Probhat cope93 if democrat==0, lcolor(black) lpattern(solid) lwidth(medthick))
(rarea U_prob L_prob cope93 if democrat==1, fcolor(gs8)) (line Probhat cope93 if
democrat==1, lcolor(black) lpattern(dash) lwidth(medthick)), ytitle(Predicted
Pr(Pro-NAFTA Vote)) xtitle(AFL-CIO COPE Score) legend(off)
graphregion(margin(vsmall))
```