

## INTERMEDIATE SOCIAL STATISTICS CLASSES

### WEEK 3: BINARY LOGIT AND PROBIT

Dr. Dave Armstrong ([david.armstrong@politics.ox.ac.uk](mailto:david.armstrong@politics.ox.ac.uk))

Dr. Michelle Jackson ([michelle.jackson@politics.ox.ac.uk](mailto:michelle.jackson@politics.ox.ac.uk))

**Objectives:** Running and interpreting logit and probit models. Interpretation and presentation of results.

**STATA Commands:** `logit`, `logistic`, `probit`, `test`, `prvalue`, `prgen`, `prchange`.

**Dataset:** American National Election Studies (ANES) Cumulative Data file 1948-2002: (`anes.dta`)

[http://www.politics.ox.ac.uk/teaching/res\\_meths/reading\\_lists/intermstats/datasets.asp](http://www.politics.ox.ac.uk/teaching/res_meths/reading_lists/intermstats/datasets.asp)

Codebook can be found at: <http://www.umich.edu/~nes/studypages/cdf/cdf.htm>

---

The term ‘categorical variable’ refers to a variable that is binary, ordinal, or nominal (including event count data). When the dependent variable is categorical, the ordinary least squares (OLS) method is no longer the appropriate estimator; in other words, OLS is biased and inefficient. Consequently, researchers have developed various categorical dependent variable models. This week we are going to discuss logit and probit models for binary variables. The data that we are using in this class is the ANES Cumulative Data File, which is a pooled dataset of each of the American National Election Studies conducted from 1948 through 2002.

#### 1. BINARY LOGIT AND PROBIT

When you have a binary dependent variable it is rarely advisable to use a simple linear probability model (i.e. OLS with binary dependent variable). There are several reasons for this. First, the distribution of residual error is heteroscedastic, which violates one of the assumptions of OLS. Second, when the dependent variable is binary, OLS estimates of the sum of squares are misleading, and this can lead to inaccurate standard errors. Yet another problem comes in the form of the possibility of predictions below 0 and above 1, which makes little sense. Further, the linear probability model implies a linear relationship between  $x$  and  $\Pr(Y=1)$ , but logistic models imply a non-linear relationship between  $x$  and  $\Pr(Y=1)$  such that as  $\Pr(Y=1)$  gets closer to its upper (lower) bound, the marginal impact of one unit increase (decrease) in  $x$  is smaller. Logistic and Probit regressions overcome these problems.

Remember from the notes that with all probability models for binary data, there is some function that relates the systematic part of the equation to the probability that the dependent variable is equal to one. In general, this is:

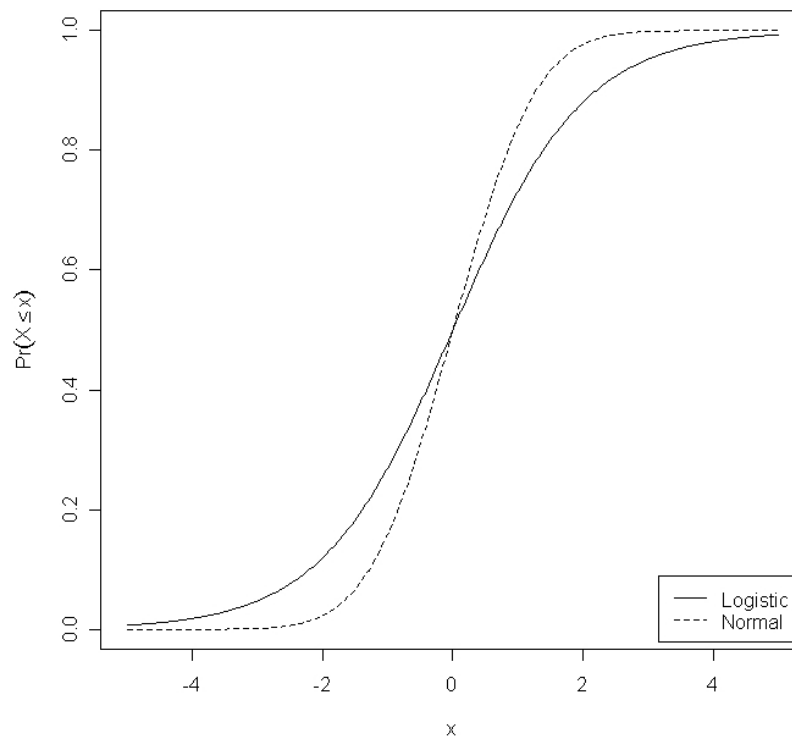
$$\Pr(y = 1 | x) = F(x\beta)$$

Probit and logit differ in the function  $F()$  that maps the linear predictor into the  $[0,1]$  interval, using known and common probability distributions. Specifically, the cumulative distribution function (CDF) is used. The CDF tells us the probability of finding a number smaller than or equal to the number in question. We represent the logistic (notice **not** 'logistical') CDF with  $\Lambda$  (upper-case lambda) and the normal CDF is represented with  $\Phi$  (upper case phi). Thus, we can represent the two different models as follows:

$$\text{Logit : } \Pr(y = 1 | x) = \Lambda(x\beta)$$

$$\text{Probit : } \Pr(y = 1 | x) = \Phi(x\beta)$$

While the CDF's do not have exactly the same form, they have similar features (e.g., the same general pattern).



## 1.1 BINARY LOGIT

We usually don't calculate probit probabilities by hand because that would be pretty difficult. However, we can calculate logit probabilities by hand as follows

$$\Pr(y = 1 | x) = \frac{1}{1 + e^{-a + b_1x_1 + b_2x_2 + \dots + b_nx_n}}$$

This can then be transformed to get back to the linear equation:

$$\log\left(\frac{\Pr(y = 1)}{1 - \Pr(y = 1)}\right) = a + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

The expression on the left hand side is the log of the odds of  $y=1$ , also known as the 'logit'. This model estimates the probability that each case (e.g. individuals) will reside in each category of the dependent variable.

We want to test the theoretical propositions that better economic perceptions, higher class, and union membership should all increase the probability of an incumbent party vote, while left-right self-placement and living in the Southwest should decrease the same probability. To test these hypotheses, we run the following model:

**Table 1: Logit Regression Coefficients for Incumbent Vote**

	Coefficient (S.E.)
Economic Perceptions (3-point) [better, middle, worse]	-0.606* (0.127)
Class (5-point) [lower, lower-mid, mid, upper-mid, upper]	-0.411* (0.108)
Union Membership (0/1)	0.466 (0.243)
Southwest Region (0/1)	-1.242* (0.494)
Urban (0/1)	0.398 (0.218)
Left-Right Self-Placement (10-point)	-0.163* (0.045)
Intercept	1.706* (0.438)

Dependent variable is vote for Vote for Incumbent (labour) Party (0/1)  
 \* p<0.05  
 Coefficients are logistic regression coefficients  
 Standard errors are in parentheses  
 N=607, Log-Likelihood=-297.080

Just looking at the significance and direction of the coefficients, we can see that the results corroborate our expectations with regards to each of the characteristics. The coefficients are on the log-odds scale, which means that, for example, a one unit increase in the variable **union** leads to an increase of 0.466 in the log odds of voting for the incumbent, assuming

that all of the other independent variables are held constant. Since `union` is a dummy variable, we can say that the difference in the log odds of voting for the incumbent for union members compared with non-union members is 0.466. It may be helpful to translate the log odds into an odds ratio. We can do this by exponentiating the log-odds coefficient:  $e^{0.466}$  [or  $\exp(0.466)$  in *STATA*] = 1.593, which means that the odds of a union member voting for the incumbent party are about 1.6 times the odds of a non-union member doing the same.

How do we get this result? In essence, we want to change the value of `union` from 0 to 1 holding all other variables constant. We know that the odds are expressed as follows:

$$\frac{\Pr(\text{Incumbent} | X)}{\Pr(\sim \text{Incumbent} | X)} = \exp(1.706 - .606E - .411C + .466Un - 1.242SW + .398Ur - .163LR)$$

The odds for non-union members are then:

$$\frac{\Pr(\text{Incumbent} | X)}{\Pr(\sim \text{Incumbent} | X)} = \exp(1.706 - .606E - .411C + .466(0) - 1.242SW + .398Ur - .163LR)$$

And for union members:

$$\frac{\Pr(\text{Incumbent} | X)}{\Pr(\sim \text{Incumbent} | X)} = \exp(1.706 - .606E - .411C + .466(1) - 1.242SW + .398Ur - .163LR)$$

We know that  $\exp(a + b) = \exp(a)\exp(b)$ , so we can use that to re-express the odds for non-union members:

$$\exp(1.706) \exp(-.606E) \exp(-.411C) \exp(.466(0)) \exp(-1.242SW) \exp(.398Ur) \exp(-.163LR)$$

and for union members:

$$\exp(1.706) \exp(-.606E) \exp(-.411C) \exp(.466(1)) \exp(-1.242SW) \exp(.398Ur) \exp(-.163LR)$$

Then, we can take the ratio of union members to non-union members,

$$\frac{\exp(1.706) \exp(-.606E) \exp(-.411C) \exp(.466(1)) \exp(-1.242SW) \exp(.398Ur) \exp(-.163LR)}{\exp(1.706) \exp(-.606E) \exp(-.411C) \exp(.466(0)) \exp(-1.242SW) \exp(.398Ur) \exp(-.163LR)}$$

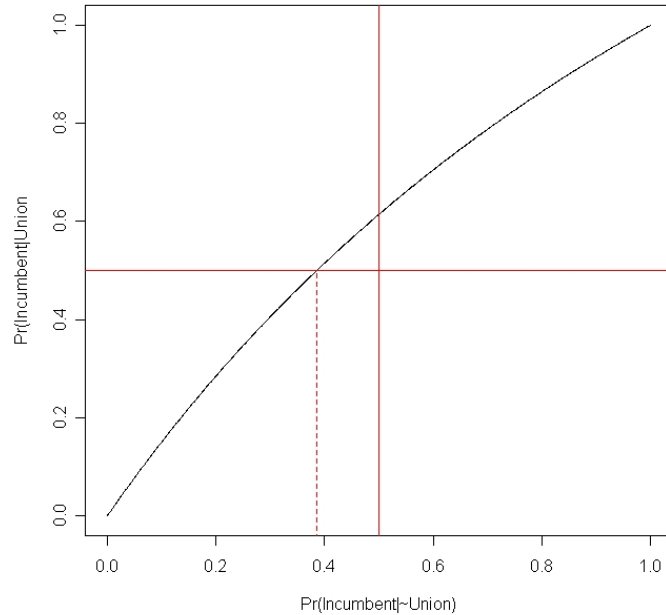
As you can see, most of these things cancel, and we're left with, the odds ratio of:

$$\frac{\exp(.466(1))}{\exp(.466(0))} = \frac{\exp(.466)}{\exp(0)} = \frac{1.593}{1}$$

So, the odds of voting for the incumbent party by union members are 1.5 times the odds of voting incumbent for non-union members. Now, we may very well be still one step removed from an intuitive figure. What does this mean for the *probability* of voting for the incumbent party? If we know one probability (either that relating to union members or to non-union members), then we can calculate the other probability. In general,

$$\Pr(y = 1 | X = x + \delta) = \frac{Or \Pr(y = 1 | X = x)}{1 + (Or - 1) \Pr(y = 1 | X = x)}$$

where  $\delta$  is the amount by which  $x$  is augmented to generate the odds ratio (this will usually be one, as in the example above). Let's take the example above. For any given probability of voting incumbent for non-union members, we can calculate the probability of voting incumbent for union members.



In the figure above, we are holding all of the other variables in the model constant, and illustrating the probabilities of voting for the incumbent. The figure shows that when the probability of a non-union member voting for the incumbent is between .39 and .5 (indicating a vote for a party other than the incumbent), union members would be expected to vote for the incumbent.

To run the same model, but display the odds ratio instead of the log odds, use the `logistic` command or specify the “or” option after the comma in the `logit` command:

```
. logit incumvote retnat class union swreg urbrur1 lrself, or
```

## 1.2 PREDICTIONS: INDIVIDUAL AND IDEAL

Predicted probabilities are the quantities with which we are generally most comfortable. There are a few different types of predictions that are useful to us. We can either look at predicted probabilities for each individual or, alternatively, we might construct ‘ideal types’ and calculate probabilities for those types. One thing that we have to think about is the degree of uncertainty around our probability predictions. These predictions are estimates and should be treated as such. If some individual’s predicted probability of voting Labour is 0.55, we should not expect that individual to vote Labour unless we can say that this 0.55 is

statistically significantly different from 0.5 (of course any probability below 0.5 indicates that the individual will *not* vote Labour).

The first, and perhaps most interesting, thing that we can do is to calculate the predicted probability for some hypothetical individual with characteristics that are theoretically interesting. This amounts to plugging interesting values into the prediction equation to obtain the resulting probability. We could very easily do this by hand, although that is time consuming. Alternatively, we can use the *SPost* suite of commands in *STATA* (see Long and Freese 2006) to accomplish this goal.

The first command of interest is `prvalue`. This command allows us to specify a set of values for the variables from which *STATA* will calculate probabilities.

```
. prvalue, rest(median)

logit: Predictions for labvote

Confidence intervals by delta method

          95% Conf. Interval
Pr(y=labour|x):      0.2420 [ 0.1931, 0.2909]
Pr(y=other|x):      0.7580 [ 0.7091, 0.8069]

      retnat   lrself   class   union  urbrurl   swreg
x=         2         5         2         0         0         0
```

The `rest(median)` option specifies that all other variables not mentioned in the command (in this case all of the variables) should be held at their median values while calculating the predicted probability of voting Labour. The results suggest that the predicted probability of voting labour is 0.242 with a confidence interval of 0.19 to 0.29. By default, *SPost* obtains the confidence intervals by what is called an ‘end-point transformation’. It calculates the following:

$$\hat{y}^* = 1.706 - .606E - .411C + .466Un - 1.242SW + .398Ur - .163LR$$

It then calculates  $\sigma_{\hat{y}^*}$  and then calculates the confidence interval for  $\hat{y}^*$ :  $\hat{y}^* \pm z_\alpha \sigma_{\hat{y}^*}$ .

After that, it calculates the predicted effect as:

$$\Pr(y = 1) = \frac{\exp(\hat{y}^*)}{1 + \exp(\hat{y}^*)}$$

And the confidence interval for the probability in the following way:

$$\Pr(y = 1) = \frac{\exp(\hat{y}^* \pm z_\alpha \sigma_{\hat{y}^*})}{1 + \exp(\hat{y}^* \pm z_\alpha \sigma_{\hat{y}^*})}$$

We can modify the behaviour of the `prvalue` command by assigning particular  $x$  values.

```
. prvalue, x(retnat 1 lrself 10 class 3 union 0 urbrur1 1 swreg 0)
```

```
logit: Predictions for labvote
```

```
Confidence intervals by delta method
```

			95% Conf. Interval	
Pr(y=labour x):	0.2037		[ 0.0969,	0.3105]
Pr(y=other x):	0.7963		[ 0.6895,	0.9031]

	retnat	lrself	class	union	urbrur1	swreg
x=	1	10	3	0	1	0

An individual with these specified characteristics has a probability of voting for the incumbent of around 0.2.

Another useful command called `prchange` generates one-unit or one-standard deviation changes for each variable in turn.

```
. prchange, rest(median)
```

```
logit: Changes in Probabilities for labvote
```

	min->max	0->1	++1/2	++sd/2	MargEfct
retnat	-0.2208	-0.1483	-0.1109	-0.0870	-0.1111
lrself	-0.2558	-0.0390	-0.0299	-0.0691	-0.0299
class	-0.2399	-0.0957	-0.0753	-0.0748	-0.0754
union	0.0952	0.0952	0.0854	0.0329	0.0855
urbrur1	0.0801	0.0801	0.0729	0.0322	0.0729
swreg	-0.1576	-0.1576	-0.2262	-0.0665	-0.2278

	other	labour
Pr(y x)	0.7580	0.2420

	retnat	lrself	class	union	urbrur1	swreg
x=	2	5	2	0	0	0
sd(x)=	.783403	2.31417	.993087	.385517	.440944	.291936

The five columns give different, interesting pieces of information. First, the 'min -> max' column shows the change in the probability of voting Labour given a maximal change in the indicated variable while holding all other variables constant (at values set by you or *STATA*). This is a way of comparing the maximal effects of the variables, however, all of the standard caveats regarding discussions of which variable is *most important* apply here, and perhaps even more so in this case. As you can see, retrospective economic judgements, left-right self placement and class all generate about the same maximal change in voting Labour. The '0 -> 1' column shows the change in probability expected for a change from zero to one in the indicated variable. For some variables, like *retnat*, this column is not meaningful because *retnat* cannot theoretically take the value of zero. The '-+1/2' and '-+sd/2' columns give centred changes of 1 unit and 1 standard deviation around the specified value. The marginal effect is the coefficient multiplied by the probability density function (PDF) evaluated at  $\hat{y}^*$ . The PDF is different from the CDF because the PDF is approximately zero at either extreme (positive or negative) and has its maximum in the middle, while the CDF is approximately zero at the negative extreme and approximately one at the positive extreme.

and has its maximum at positive infinity. The intuition here is that variables have their biggest effect around the middle of the distribution of the predicted values.

A final command that might be useful is `prgen`. This generates predictions and confidence intervals for continuous variables.

```
. prgen lrself, gen(lrs) rest(median) ci
```

logit: Predicted values as lrself varies from 1 to 10.

```

      union    swreg   lrself  urbrur1    class   retnat
x=      0         0         5         0         2         2

```

This command also generates a number of new variables:

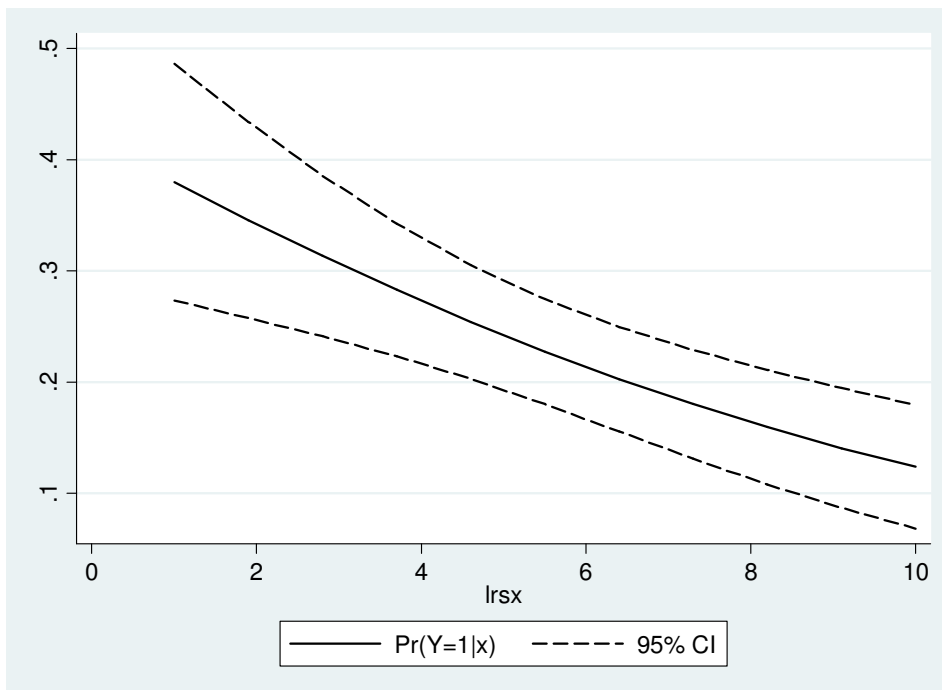
```

lrsx          float   %9.0g
lrsp0         float   %9.0g          pr(other)=Pr(0)
lrsp1         float   %9.0g          pr(labour)=Pr(1)
lrsp0lb       float   %9.0g          LB pr(other)=Pr(0)
lrsp1lb       float   %9.0g          LB pr(labour)=Pr(1)
lrsp0ub       float   %9.0g          UB pr(other)=Pr(0)
lrsp1ub       float   %9.0g          UB pr(labour)=Pr(1)

```

*lrsx* indicates the value of the variable, in this case *lrself*, at which the prediction is generated. *lrsp0* and *lrsp1* are the probabilities that  $y=0$  and  $y=1$  respectively, for given values of left-right self placement. *lrsp0lb* and *lrsp0ub* are the lower and upper 95% confidence bounds for the probability that  $y=0$  and *lrsp1lb* and *lrsp1ub* are the lower and upper 95% confidence bounds for the probability that  $y=1$ . These can then be used to produce a graph:

```
. twoway (line lrsp1 lrsx, lcolor(black)) (line lrsp1lb lrsx, color(black)
lpattern(dash)) (line lrsp1ub lrsx, lcolor(black) lpattern(dash)),
xscale(range(1 10.)) legend(order(1 "Pr(Y=1|x)" 2 "95% CI"))
```



## EXERCISE

Add previous vote for labour (*votelab2001*) to the model, and interpret the coefficient. What is the effect of a previous vote for Labour on the probability of voting for Labour?

### 1.3 BINARY PROBIT

Probit is an alternative to the log-linear approach to handling binary (and other categorical) dependent variables. The main difference is in the distribution of the error. The probit model assumes normally distributed errors on the latent continuous variable underlying the binary variable. Logit assumes a standard logistic distribution of these errors which has fatter tails than the normal distribution (see Long 1997, p43). There are other differences, but for the most part they won't be relevant for this class. Like logit, the researcher focuses on a transformation of the probability that  $Y=1$ . The following formula describes probit function.

$$\Phi = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\alpha+\beta X} \exp\left(-\frac{1}{2}Z^2\right) dZ$$

Don't worry – you don't need to memorise this! The capital phi,  $\Phi$ , is used to designate the *probit* link function. Instead of the log transformation of the predicted scores, the probit transformation is used. With probit analysis there is no odds ratio obtained. Probit and logistic regression will usually produce results that are very similar substantively, especially with large sample sizes. Hence, logit models are used more frequently than probit (the coefficients may be easier to interpret because they are odds ratios). To illustrate the similarity between the two models, we will run the same model, but this time using the probit link function, by simply using the `probit` command.

```
.probit incumvote retnat class union swreg urbrur1 lrself
```

As you can see, our logit and probit analyses arrive at the same general conclusions, but the logit and probit coefficients differ in magnitude. Logit coefficients are about 1.8 times the corresponding probit coefficients. The regression coefficients of the probit model are effects on a cumulative normal function of the probabilities that  $Y = 1$  (i.e. the probability that a person is a democrat). They are 'Z scores' or standard normal scores. Probit coefficients (z scores) can be translated into predicted probabilities, using any table of the standard normal distribution.

### 1.4 MODEL FIT AND SELECTION

There are a number of measures of fit for logit and probit models and even though there are pseudo- $R^2$  measures, they **do not** correspond to the percentage of variance in the dependent variable accounted for by the independent variables. The pseudo- $R^2$  measures attempt to give the spirit of variance explained, but these models are not trying to 'explain variance' in the usual sense of  $R^2$ . As such, you should be extremely cautious about making statements to that effect about the pseudo- $R^2$  measures. Further, there are a number of different pseudo- $R^2$  measures and you should always be explicit about which one you are reporting if you

choose to report one at all (see Long 1997 or Long and Freese 2006, for definitions and relationship to the OLS  $R^2$ ).

There are a few different ways of considering whether your model is a good model. One common test is to ask whether this model is better than one with only an intercept, as assessed through a likelihood ratio test. A model can be thought of as either being constrained or unconstrained. In an unconstrained model, we estimate all of the effects of interest. In a constrained model, not surprisingly, we constrain one or a number of the parameters. In the example here, we might want to test the constraint that all of the coefficients except for the constant are simultaneously zero. We estimate two models: one where all of the coefficients are estimated and one where only a constant is estimated (constraining the other parameters to be zero). The likelihood ratio statistic is -2 times the constrained log-likelihood minus the unconstrained log-likelihood. This statistic has a  $\chi^2$  distribution, with degrees of freedom equal to the number of constraints put on the model. The log-likelihood of the constrained model in this case is -334.874 and for the unconstrained model it is -297.080.

$$LR = -2(-334.874 - -297.080) = 75.588$$

We can compare this to a  $\chi^2$  distribution with six degrees of freedom (the number of coefficients estimated in the unconstrained model). We can do this in *STATA* as follows:

```
. dis chi2tail( 6, 75.588)
2.904e-14
```

You can compare this to the model output and see that it is the same. The chi-square statistic is analogous to an F-test in regression in that it assesses the joint significance of all the model parameters. Hypothesis tests of individual or multiple coefficients can also be computed with `test` for the Wald test and with `lrtest` for the corresponding likelihood-ratio tests. To test that the effect of being Catholic or Jewish or Atheist on voting Democrat are all equal to 0, we fit the full model and then,

```
.test class union
```

Thus, we can reject that the hypothesis that the effect of each variable is simultaneously zero. `test` can also be used to test the equality of coefficients.

```
.test class = union
```

You may also want to be able to test whether one model is statistically 'better' than another proposed model. This can be accomplished through a couple of different processes. Both can be done using the `fitstat` command in *STATA*. You need to make sure that whichever two models you compare, those models have been computed for exactly the same set of observations (although one need not be nested within the other). The procedure is as follows:

1. Run the first model
2. Run the second model adding the if statement `if e(sample)`, this will restrict it to the observations in the first model.
3. Type: `gen tempsamp=e(sample)`
4. Run the first model adding the if condition `if tempsamp == 1`.

5. Save the results of the first model by typing `fitstat, saving(mod1)`
6. Run the second model adding the if condition `if tempsamp == 1`.
7. Type `fitstat, using(mod1)`

Imagine that we wanted to see whether class or union was a better variable, but we didn't want a model that uses them both. We would have to use the following set of commands:

```
logit incumvote retnat union swreg urbrur1 lrself
logit incumvote retnat class swreg urbrur1 lrself if e(sample)
gen tempsamp = 1 if e(sample)
logit incumvote retnat union swreg urbrur1 lrself if tempsamp == 1
fitstat, saving(mod1)
logit incumvote retnat class swreg urbrur1 lrself if tempsamp == 1
fitstat, using(mod1)
```

The likelihood-ratio test is only valid if the models are nested, which these ones are not. In our case, we could rely on the Bayesian Information Criterion (see Long 1997, 109-112, for a discussion). There is a nice guide on the bottom of the output that suggests which of the two models (current or saved) has support. Generally, BIC differences over 2 are considered as positive support and differences over 6 are considered as strong support for the model with the BIC statistic closer to zero.

#### USEFUL REFERENCES

Alvarez, R. Michael, and Jonathan Nagler (1998) 'When Politics and Models Collide: Estimating Models of Multi-Candidate Elections.' *American Journal of Political Science* 42:55-96.

King, G., M. Tomz, and J. Wittenberg (2000): 'Making the Most of Statistical Analyses: Improving Interpretation and Presentation,' *American Journal of Political Science*, 44(2), 347-361.

Long, J. Scott (1997) *Regression Models for Categorical and Limited Dependent Variables*. Thousand Oaks, CA: Sage Publications.

Long, J. S., and J. Freese (2006) *Regression Models for Categorical Dependent Variables Using Stata*. Stata Press.