Mgmt 469

Discrete Dependent Variables

Limitations of OLS Regression

A key implicit assumption in OLS regression is that the dependent variable is continuous. This

is usually a pretty good assumption. For example, costs, profits and sales are all essentially

continuous. But some dependent variables are discrete – that is, they take on a relatively small

number of integer values. Examples include annual sales of Boeing 777 airframes, the number of

auto dealers in a town, and the number of football games won by the Northwestern Wildcats in a

given year. Sometimes, the dependent variable equals zero for many observations (e.g., the

number of restaurant customers who order a particular bottle of wine on a given day, the number

of hours of television viewed nightly by Kellogg students). Last but not least, many interesting

research studies involve dependent variables that represent the answers to "yes/no" questions

(e.g., Did a survey respondent buy a car? Does a firm have women on its Board of Directors?)

As we will see, OLS regression is not always well-suited for analyzing these kinds of

dependent variables. Fortunately, there are number of alternatives to OLS that are easy to

implement. This note discusses some of the most commonly used methods for dealing with

discrete dependent variables -- variables that take on a limited range of values.

Dichotomous dependent variables

There are lots of times when the dependent variable of interest takes on one of two

values. For example, suppose you are studying car purchase decisions. You survey consumers

to find out if they have purchased a car in the past year. Your observations can be coded 1 (if the

1

consumer bought a car) or 0 (if the consumer did not).   When your dependent variable can take

on one of two values, then you have a *dichotomous dependent variable* and the resulting models

are sometimes called 0/1 models.

Continuing this example, you might state the research question as follows:

"What factors help predict whether a consumer will buy a car?"

Expressed this way, it becomes apparent that you are estimating *probabilities*. Specifically, you

seek to identify factors that influence the probability that a consumer will buy a car. You can use

OLS regression to estimate 0/1 models. This is known as a *linear probability model.*

Unfortunately, it can be difficult to interpret the results of linear probability models, as can be

seen by continuing the car purchase example.

You wish to identify factors that predict whether a person will buy a car. Your LHS

variable is **carbuy**, which equals 1 if the individual bought a car and 0 if not. The RHS variables

are **income** (in thousands of dollars) and **schooling** (in years completed).

You could estimate the following linear probability model in Stata:

regress carbuy income schooling

Suppose you got the following coefficients:

$$B_0 = -.70, \ B_{income} = .01, \text{ and } B_{schooling} = .025.$$

The following table gives the resulting predicted probabilities of car purchase for 3 individuals

with different levels of income and schooling:

Individual	Income	Schooling	Probability[*]
1	50	12	0.10
2	200	16	1.70
3	20	12	-0.20

[*]You should be able to verify that these are the predicted values.

The predicted probability that the first individual buys a car is .10. This seems plausible. But the predicted probabilities for the second and third individuals are 1.70 and -0.20. These are nonsensical predictions! Probabilities should be bounded between 0 and 1. Linear probability models do not provide such bounds. This is potentially a major problem.

The next table gives predicted purchase probabilities as income increases:

Individual	Income	Schooling	Probability*
1	100	16	0.70
2	110	16	0.80
3	120	16	0.90
4	130	16	1.00
5	140	16	1.10

It is no surprise that the purchase probability increases linearly with income. As the purchase probability approaches 1.00, the probabilities stop making sense. There ought to be diminishing returns – as income increases, the probability of buying a car increases, but at a decreasing rate. There should be a symmetric effect as the probability gets close to 0. This is a second potentially important problem and it cannot be easily fixed using OLS.

The third problem with the linear probability model is that the errors are often highly heteroscedastic and difficult to correct, particularly when the predictions from the regression fall outside the range of 0 to 1 and the sample size is small.[1]

---

[1] The reason for this is tied to the first two problems – observations with outlier values of key predictors will tend to have predictions that are much bigger than 1 or much smaller than 0; these will naturally be the biggest prediction errors. Thus, the prediction errors are correlated with the predictors, a violation of homoscedasticity.

Taken together, we conclude that it is acceptable to run a linear probability model if:

· The predicted values all fall comfortably between 0 and 1

· The estimated effects of changing the values of X variables do not push the predictions close to 0 or 1.

· The sample size is large (hundreds of observations or more).

If these are potential problems for your model, you can instead estimate a *Logit* or *Probit* model. These are nonlinear models designed to avoid the potential problems described above.

Logit and Probit Models

Logit and *probit* models constrain predictions to fall within the 0-1 interval. Logit and probit make slightly different assumptions about the distribution of the error term and use different formulae to compute the predicted values. Even so, after proper conversion the two models usually give nearly identical results. I will discuss the simpler logit formulation; you can easily run both logit and probit in Stata.

Here is the secret behind logit. Suppose you had some value ρ that could range from $-\infty$ to $+\infty$. Now consider what happens if you plug ρ into the "logit formula:"

(1) $$f(\rho) = e^{\rho}/(1+e^{\rho})$$

This "transformation" of ρ is just what you want.[2] Although ρ can take on any value, $f(\rho)$ is bound between 0 and 1. Moreover, as ρ increases, $f(\rho)$ follows an S-shape, displaying exactly the kind of nonlinearity we are looking for. (See figure on next page.)

---

[2] There is a similar algorithm for Probit but the prediction formula is more complex than the logit equation (1).

Logit Curve

$f(\rho)$
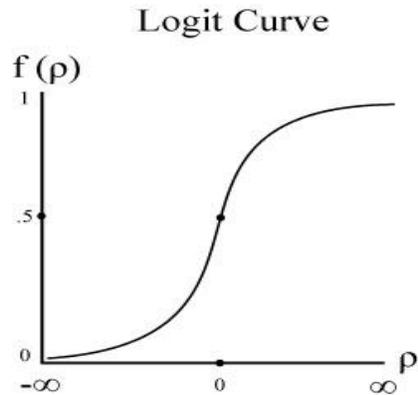
In a logit model, you first estimate the coefficients of $\underline{BX}$. Once you obtain $\underline{BX}$, you plug the results into the logit equation (1) to obtain the predictions $f(\underline{BX})$. The computer uses maximum likelihood to obtain the values of $\underline{B}$ that generate the best predictions $f(\underline{BX})$.

There are some parallels between logit and OLS:

· In OLS, you compute $Y=\underline{BX}$. In logit, you compute $\rho=\underline{BX}$.

· In OLS, your predictions equal $Y=\underline{BX}$. In logit, your predictions equal $f(\rho) = f(\underline{BX})$.[3]

These are the predicted probabilities of scoring a 1.

Now that you know how to estimate a logit model, let me remind you one more time of the distinction between logit and OLS. In OLS, the predicted value of Y is $\underline{BX}$. In logit, the predicted value of Y is $f(\underline{BX}) = e^{BX}/(1+e^{BX})$.

---

[3]Logit and Probit models are less prone to heteroscedasticity. One can use Stata to test for and correct heteroscedasticity in Probit but not Logit models. The details may be found by using the Stata command **help hetprob**. This is beyond the scope of this lecture but we will revisit the issue when we discuss heteroscedasticity in a later lecture.

Computing the effect of X on Y in a logit model

There are three ways to compute the effect of a change in X on the probability of scoring a 1 in a

logit/probit model. You should learn the "derivative" technique so that you understand what

Stata does when you implement the "lazybones" technique.

Here is the "derivative" technique for Logit. (The formula for probit is much more

complex.) Throughout, I will use ρ to refer to <u>BX</u>.

1) Choose a change in X that is of interest. Call this ΔX.

2) Recall the formula $\Delta Y = \Delta X \cdot (\partial Y / \partial X)$. You need to compute $\partial Y / \partial X$, or, in this case, $\partial f(\rho) / \partial X$, because $Y = f(\rho)$.

3) If you have no interactions or exponents, you differentiate $f(\rho)$ with respect to X to get:

$\partial f(\rho) / \partial X = B_x \cdot f(\rho) \cdot (1 - f(\rho))$, or

$\partial f(\rho) / \partial X = $ (regression coefficient) $\cdot$ (prob. of scoring a 1) $\cdot$ (prob. of scoring a 0).

4) It follows that $\Delta Y = \Delta X \cdot B_x \cdot f(\rho) \cdot (1 - f(\rho))$.

5) You can use the derivative formula from earlier in the course if your model has interactions or exponents. For example, if your model includes $B_x X$ and $B_{x2} X^2$, then

$\partial f(\rho) / \partial X = (B_x + 2B_{x2}X) \cdot f(\rho) \cdot (1 - f(\rho))$ and $\Delta Y = \Delta X \cdot (B_x + 2B_{x2}X) \cdot f(\rho) \cdot (1 - f(\rho))$.

6) Note that due to the S-shape of the $f(\rho)$ function, the change in the probability of scoring a 1 depends on the initial, baseline probability of scoring a 1. This implies that to compute the derivative, *you must specify a baseline value of f(ρ).*

7) Choose a baseline value that would be of some interest to your audience, such as the mean probability of scoring a 1 in the sample. Once you select the baseline value, you can ask how the probability of scoring a 1 changes as X changes.

Here is the *lazybones way*.

After you estimate your Logit model in Stata, type **mfx compute**. Stata computes the derivative and reports the effect of a one unit change in each predictor X on the probability of scoring a 1. Stata does this for a specific baseline probability. The default choice of baseline probability is when each predictor variable equals its mean value. But you can specify any baseline by typing **mfx compute, at …** See the Stata help file (**help mfx compute**) for more details. As always, you cannot use mfx compute if you have exponents or interactions.

There is an even easier way to do this in probit. Use the command **dprobit** to estimate your regression model. Stata will skip the preliminaries and simply report the estimated magnitudes of changes in X (and their standard errors.)

Here is the "brute force" technique for estimating magnitudes (*optional*):

1) Estimate the model and recover the values of <u>B</u>

2) Take some "baseline" starting point. This could be the mean for the sample. Another popular baseline is $f(\rho) = .50$; i.e., the probability that Y = 1 is .50.

3) Compute the value of ρ corresponding to this baseline value. Thus, if $f(\rho) = .50$, you need to find the value of ρ such that $f(\rho) = e^{\rho}/(1+e^{\rho}) = .50$. In this case, $\rho = 0$.

4) Compute the new value of ρ if X increases by ΔX. Call this ρ'. It is easy to see that

$$\rho' = \rho + (B_x \cdot \Delta X).$$

5) Compute $f(\rho')$. This tells you the new predicted probability that Y = 1.

Reviewing the Options for Estimating 0/1 Models

This section presents results from three alternative estimates of a 0/1 model. I have data on whether women undergo vaginal delivery or have caesarian sections. I will estimate a simple model on a small subsample of the data. First the summary statistics:

Variable	Obs	Mean	Std. Dev.	Min	Max
caesarian	237	.278481	.4492001	0	1
age	237	24.64135	5.693631	15	43

I will regress caesarian on age. First the linear probability (OLS) results:

caesarian	Coef.	Std. Err.	t	P>|t|
age	.0153808	.0050478	3.05	0.003
_cons	-.1005222	.1276483	-0.79	0.432

Now the Logit model followed by mfx compute (computed at the mean value of age):

caesarian	Coef.	Std. Err.	z	P>|z|
age	.0741422	.0253224	2.93	0.003
_cons	-2.814475	.6631677	-4.24	0.000

. mfx compute

Marginal effects after logit
     y = Pr(caesarian) (predict)
        = .27140392

variable	dy/dx	Std. Err.	z	P>|z|	[
age	.0146612	.00496	2.96	0.003	.0

Finally, the dprobit estimate:

caesar~n	dF/dx	Std. Err.	z	P>|z|
age	.0148914	.0050939	2.92	0.004
obs. P	.278481			
pred. P	.2729924	(at x-bar)		

Note from the OLS model that the effect of *age* is significant but modest in magnitude. Each increase in age causes the caesarian probability to increase by 1.54 percent. Given that *age* ranges from 15 to 43, the predicted probabilities fall comfortably within 0 and 1. We suspect that the linear probability model is acceptable. The logit and dprobit results confirm this. The magnitude and significance of the *age* effect (observed from the mfx compute in the logit and directly from the dprobit) are virtually unchanged from the OLS model.

Although logit and probit are technically preferred to the OLS/linear probability model, the latter is easier and faster when sample sizes are very large. In addition, the interpretation of fixed effects is a bit easier in the OLS/linear probability model because it is a linear model. A safe approach is to estimate the OLS/linear probability model for most of your 0/1 analyses (provided the predicted probabilities fall well within 0/1 and you have at least several hundred observations) but also estimate a logit or probit to test for robustness. If predicted probabilities are outside 0/1 or you have at most a few hundred observations, you should stick to logit/probit.

Logistic regression (optional)

Biostatisticians (i.e. statisticians in the biomedical sciences) use a version of the logit model called the logistic regression. Logistic and logit regressions are identical in every way except for how they report their results.

The logistic regression works with *odds ratios*.

- Odds ratio = Prob(1)/Prob(0)

For example, in late 1997, the average county in the United States had a probability of having a local Internet Service Provider of about .43. Thus, the odds ratio for the average county was

.43/.57 = .754.

The coefficients that are reported by Stata when you run a logistic regression tell you how the odds ratio changes when the X variable increases by one unit. For example, suppose the coefficient on population is 1.04, where population is measured in thousands. Then a one unit change in population causes the odds ratio to increase by a factor of 1.04. Thus, the odds ratio on having an ISP increases from .754 to $(1.04 \cdot .754) = .785$.

You can use the new odds ratio to determine the new prob(1). You know that Prob(1)/Prob(0) =.785, which is equivalent to saying Prob(1)/[1-Prob(1)] = .785. A little manipulation yields Prob(1) = .440.[4] Thus, when the population increases by 1000, the probability of an Internet service provider in the market increases from .430 to .440. (Note: because this is a nonlinear model, if you choose a different starting probability, you will get a slightly different increase.)

A Reminder about Reporting Magnitudes

Whether using logit, probit, or logistic regression, you will probably want to report the magnitudes of the estimated effects. In all of these models, *the magnitudes of the estimated effects depend on the baseline probability of scoring a 1*. Pick a baseline probability that seems interesting, such as the probability of scoring a 1 in the median market. Then use the derivative, brute force, or **mfx compute** method to compute the magnitude of the estimated effect at each of these interesting probabilities.

---

[4] In general, if P/(1-P) = x, then P = 1/[(1/x +1)].

General Regression Models

The logit model is an example of a class of models that I call "general regression models" or

GRMs.[5] OLS is also a GRM. In GRMs, the computer keeps track of two values: the **"*regression*

*score"*** and the **"*real world score."*** The regression score is defined to be $\underline{BX}$. The real world

score is the predicted value of the dependent variable, which is a function of $\underline{BX}$, or f($\underline{BX}$).

    In OLS models, the regression score and the real world score are identical. In other

words, f($\underline{BX}$) = $\underline{BX}$. In logit, the regression score does not equal the real world score. The logit

regression score is $\rho = \underline{BX}$. But the real world score is $f(\rho) = e^{\rho}/(1+e^{\rho}) = e^{\underline{BX}}/(1+e^{\underline{BX}})$.

    In GRMs, the computer estimates $\underline{B}$'s that give the "best" predictions of the real world

scores, where "best" implies maximizing the likelihood score. (If OLS is properly specified, then

minimizing SSE and maximizing the likelihood score should yield identical results.)

    All of the models encountered in this class are GRMs. The concepts of real world score

and regression score are fundamental to understanding them. Remember, the regression

coefficients will generate a regression score $\underline{BX}$. But to get to the real world score, you will

need to transform the regression score by some function f($\underline{BX}$)

---

[5] This is not a "term of the trade", but I find it helpful for understand the difference between OLS and other models.

Hypothesis testing after Maximum Likelihood Estimation

You use t-tests to assess the significance of individual predictors in OLS. When assessing significance after maximum likelihood regression, you will use the z-statistic. Recall that the t-statistic $= B_x/\sigma_x$, where σ_x is the standard error of the estimate B_x. Recall also that significance levels are based on the t-distribution. The z-statistic also equals β_x/σ_x, but the significance levels are now based on the normal distribution.[6] You do not really need to know the difference; Stata will report the correct significance levels for any GRM.

Recall that you used a partial-F (Chow) test to assess the significance of groups of predictors in OLS. You probably didn't notice, but this test used information about the variances and covariances of predictors.[7] The comparable test for other GRMs is known as the *Wald test*.[8] You perform a Wald test in Stata using exactly the same syntax that you used to perform a Chow test. After you estimate your model, type **test varlist**, where **varlist** is a list of variables you are jointly testing. By now, you should know how to interpret the results.

---

[6] Statisticians use z to refer to goings-on with the normal distribution—hence the terms z-statistic and z-score.

[7] Covariances are important because if two coefficients move together, there is less independent information to confirm significance.

[8] The Wald test compares the squared values of a group of coefficients against their variances and covariances. This is very much like the z-statistic. The z-test statistic is B_x/σ_x. If you square this value, you have a Wald test statistic: B_x^2/σ_x^2. To assess the significance level of the Wald test statistic, you use the Chi-square distribution. This is because the Chi-square equals the square of the normal distribution! Like the Chow test, the Wald test for several variables adjusts for both variances and covariances. The formula is beyond the scope of this course.

Another test (optional)

The Wald test and Chow test do not have identical interpretations. Recall that the Chow test determines if the added variables *add significant predictive power*. The Wald test determines if the added variables have *nonzero coefficients*. In OLS, the two tests turn out to be identical. But in other GRMs, the two tests may slightly disagree. On occasion, you can reject the hypothesis that the coefficients are zero but you cannot reject the hypothesis that the new variables add no predictive power, or *vice-versa*. To determine if the new variables add predictive power to a maximum likelihood model, you must perform a *Likelihood Ratio* test (LR test). The LR test has slightly better statistical properties than the Wald test but it usually gives similar results and is a bit more complex to use. I recommend you use the LR test only when the Wald test is ambiguous. If you are interested, I can show you the syntax or you can try to wade through the discussion in Stata by typing **help lrtest**.

Multinomial choice models (optional but valuable)

Suppose you have survey data that identify whether an individual purchased a sport utility vehicle, minivan, sedan, sports car, or made no car purchase. In this case, the dependent variable can take on several values.   To analyze this decision, you could assign numerical values to the different choices. For example define **carcode** as follows:

> 0 = no purchase
> 1 = sport utility vehicle
> 2 = minivan
> 3 = sedan
> 4 = sports car

Carcode is neither cardinal nor ordinal. It makes no sense to add scores together (e.g., two minivans do not equal a sports car). Nor does it make sense to rank them (e.g., sedan is not greater than sports utility vehicle). You cannot use GLMs to study these choices.

What you need is some way to determine the *relative probabilities* with which individuals select different types of cars. If there were just two options, you could use a logit or probit model to compute the probability of each option. When there are more than two options, you have several models to choose from, all related to logit and probit models. Because the probit models get very complex as the number of options increases, I will focus on the logit-based models.

The two logit-based models are known as the *multinomial logit* model and the *conditional choice* model. In the multinomial logit model, the predictor variables are characteristics of the decision maker (e.g., income, schooling). The estimated coefficients allow you to assess how changing the values of a predictor affects the probability of each choice. For example, the coefficients might indicate that as income increases, the probability of selecting an SUV or sports car increases, while the probability of selecting a sedan decreases and the probability of selecting a minivan is unchanged. By examining the coefficients on income, schooling, etc., it is possible to predict the choices of different individuals.

In conditional choice models, the predictor variables are characteristics of the choices (e.g., price, quality) and the statistical model indicates how changing these characteristics affects the choice probabilities. For example, the model might indicate that increasing the price of SUVs leads to a reduction in the sales of SUVs and an increase in the sales of minivans and sedans, but does not affect the sales of sports cars. By examining the coefficients on price,

14

quality, etc., one can determine the potential market share for a product with any specific characteristics.

Mixed models incorporate characteristics of the decision makers and the products. Due to the way in which the data are organized, you estimate mixed models as if they were conditional choice models, with a few wrinkles thrown in. Mixed models are very versatile and are widely used by economists to study product market differentiation. (E.g., you can study how the effects of product quality vary by customer segment.)

These valuable empirical methods are facilitated by the transaction-specific data (e.g. scanner data). Stata and other advanced regression packages estimate these models, though you will need to put in considerable time organizing the data. You will also need to pay attention to how the data are structured. You will need good mix of consumer and product attributes. For example, if you have race and sex, be sure to have lots of men and women in each race. You also have to pay attention to how you characterize the choices. For example, dividing the SUV category into two categories -- large SUVs and small SUVs -- can profoundly affect your results in ways that you might find unattractive. If one of these models seems to be appropriate, seek out an expert for further advice on how to proceed.

Count models

Consider trying to estimate the number of individuals in Chicago who will purchase a Bentley automobile on a given day. The unit of observation is the day. The modal value for the dependent variable is 0. There are a few 1's, fewer 2's and an occasional 3, 4, or even 5. (I call the latter aspect the "long upper tail.") This is a "count" variable. There are many other

examples of *real world processes* whose outcomes are count variables, including the number of people who enter a store in a given hour, the number of firms to enter a market in a given month, or the number of wins for the Northwestern football team in a given year. In most cases, you are counting up the number of times something happens in a given period of time. But any data set that has a similar distribution (no negative values, lots of 0's and 1's, and a long upper tail) can be estimated using count data methods.

Just as you could use OLS to study 0/1 data, you can use OLS to study count data. If the number of "arrivals" in each period tends to be large, then OLS is probably safe.[9] Your OLS predictions will almost surely be nonnegative and with a long upper tail. If the number of arrivals is frequently small, and especially if there are a preponderance of 0's and 1's, then OLS is no longer acceptable. In particular, you will probably get negative predictions. You may also be unable to reproduce the long upper tail. Thus, you may want to find a model that is better tailored to count data.

One such model is the *Poisson* (pronounced pwah'-sonn) model. The key parameter in the Poisson process is denoted by λ. The arrival rate is the expected number of times that the event under consideration (e.g. a customer enters a store) will occur in a given period of time, as given by the Poisson formula:

$$\text{Arrival rate} = e^{\lambda}.$$

Note that with this formula, the expected number of arrivals is never negative. You may also have a few periods in which there are quite a few arrivals, thus generating the long upper tail.

---

[9] Yogurt sales follow a Poisson process, but the number sold per week is so large that we could safely use OLS.

You will probably want to determine how various predictor variables affect the arrival rate. This requires estimating a *Poisson regression*. Like other GRM models, you specify the predictor variables $\underline{X}$ and the computer estimates $\underline{B}$. From this, the computer obtains a regression score $\lambda = \underline{BX}$. The computer takes λ and calculates a predicted real world score: $Y = e^{\lambda} = e^{\underline{BX}}$. The computer finds those values of $\underline{B}$ that maximize the likelihood score.

There are three ways to use the coefficients $\underline{B}$ to predict how a change in X will affect Y. To use the derivative method, recall that $\Delta Y = \Delta X \cdot \partial Y / \partial X$.

1) Choose a value for ΔX.

2) If your model has no interactions or exponents, then **the derivative of the predicted value with respect to a change in X equals $\partial e^{\lambda} / \partial X = B_x e^{\lambda}$.**

3) This is sort of reminiscent of the formula for the logit – the derivative equals the baseline number of arrivals (e^{λ}) times the coefficient. Thus, you will need to choose a baseline number of arrivals (usually the mean for the sample.)

4) Of course, if you have interactions or exponents, the formula is a bit more complex.

Q: Can you recall how to alter the formula when there are interactions and exponents?

To use the "brute force" method (*optional*):

1) Estimate the model, recover the values of <u>B</u> and select a change in X, ΔX.

2) Take some "baseline case" as a starting point. For example, take the baseline case where the estimated number of arrivals is 5.   The question you will answer: how does the expected number of arrivals change if the value of X changes by ΔX?

3) Compute the value of λ that generates this baseline value. Thus, find the value of λ such that $e^{\lambda} = 5$. My calculator tells me that λ = 1.61.

4) Compute the value of λ if X increases by one unit (or by some reasonable change). This is easily determined from the coefficient on X. Let the new value of λ be $\lambda' = \lambda + \Delta X \cdot B_x$

5) Compute $e^{\lambda'}$.   This is the new predicted number of arrivals.

The third way to estimate the effect of X on Y is to use **mfx compute**.

Software packages such as Stata estimate Poisson regressions as easily as they do OLS. If you have data on the number of bankruptcies per month and the monthly level of unemployment, then in Stata type

> **poisson bankrupt unemployment**

You can now predict monthly bankruptcies. Your predictions will never be less than 0 and will have a long right tail.

A Problem with Poisson

A key requirement of a Poisson process is that the conditional mean (i.e., the expected outcome if the predictors equal their mean values) should equal the conditional variance (i.e., the variance of the expected outcome). Sometimes, the dependent variable is "overdispersed," so that the conditional variance exceeds the conditional mean. (This means that the data show a lot more variation than you would expect from a Poisson process.) If this occurs, the Poisson model does not do an ideal job of fitting the real world process that you are studying.

To test for the "goodness of fit" of the Poisson model, run your Poisson regression and then type **poisgof**. If the test statistic is significant, then the assumption about the conditional mean equaling the conditional variance is violated and the Poisson results may be problematic.

Fortunately, there is another process called the *negative binomial* model that is almost identical to Poisson but does not require the conditional variance to equal the conditional mean. While the coefficients may differ, the formula for making predictions is the same (the expected outcome is e^λ). Use the negative binomial if you failed the **poisgof** test.

You can easily estimate a negative binomial model by typing **nbreg** instead of **poisson**. For example, you could estimate

nbreg bankrupt unemployment

The resulting output will look virtually identical to the output of any other GRM model – some likelihood scores followed by a table of regression coefficients, standard errors, and z-statistics. *You may interpret the regression coefficients in a negative binomial model in the same way you interpret them in a Poisson model.*

19

The output of the negative binomial model includes the value of a parameter called alpha. Alpha is a measure of the dispersion of the predictions. You should examine the "likelihood ratio" test reported by Stata to see if alpha is significantly greater than 0. (See the Stata screen shot below.) If so, then there is too much dispersion in the Poisson model and you should stick with the negative binomial. If not, you can revert to the more parsimonious Poisson model.

```
Negative binomial regression                    Number of obs   =       3072
                                                LR chi2(3)      =    1757.81
Dispersion     = mean                           Prob > chi2     =     0.0000
Log likelihood = -4713.585                      Pseudo R2       =     0.1572


number_of_~s |     Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]

populatio~92 |   5.45e-06   3.00e-07    18.14   0.000     4.86e-06    6.04e-06
   income_92 |   .0000419   .0000129     3.24   0.001     .0000165    .0000672
pct_colle~90 |   .1235296   .0077456    15.95   0.000     .1083485    .1387108
       _cons |  -2.437858   .1896297   -12.86   0.000    -2.809526   -2.066191

    /lnalpha |   .5841157   .0451108                      .4957002    .6725311

       alpha |   1.793404   .0809018                      1.641647     1.95919

Likelihood-ratio test of alpha=0:  chibar2(01) = 1.3e+04  Prob>=chibar2 = 0.000
```

Ordered Dependent Variables

The *ordered Probit* model is a useful variant of the Poisson model that has grown in popularity among statisticians and economists. As with the Poisson model, the dependent variable may be thought of as a count variable, such as the number of firms in a market. The ordered Probit is especially useful when the LHS variable is ordered, *but the intervals between the scores are arbitrary*. (I will give you an example in a moment.) Ordered probit also has a very useful application to studies of firm entry, as we will see in the hospital services project.

It is easiest to explain ordered probit through an example. Suppose the LHS variable is the response to a survey of customers about their new car purchases. The response categories

are: 1=not at all satisfied, 2=somewhat dissatisfied, 3=neutral, 4=somewhat satisfied, 5=very

satisfied.[10] You may be tempted to use OLS to analyze this LHS variable, and, indeed, this is

what a lot of folks do.

There are several reasons not to use OLS. First, the OLS predictions will range from

below 1 to above 5. Second, when you use OLS (and even if you use Poisson), you implicitly

assume that the data is cardinal. In other words, the interval between any pair of categories (e.g.

between 1 and 2) is of the same magnitude as the interval between any other pair (e.g. between 4

and 5). In the context of survey responses, this is a tenuous assumption. Perhaps it does not take

much for customers to move from "not at all satisfied" to "somewhat satisfied", but it takes a lot

for consumers to jump from "somewhat" to "very" satisfied.

It is far better to treat the data as *ordinal* rather than cardinal. With ordinal data, each

higher category represents a higher degree of satisfaction, but respondents do not necessarily treat

the intervals between adjacent categories as equal. Ordered models, including ordered probit, are

designed to estimate ordinal data.

Ordered probit resembles other GRMs: the regression generates coefficients <u>B</u> from

which you can compute the regression score <u>BX</u>. But getting from the regression score to the

real world score is a bit different than in other GRMs, so pay attention!

In addition to reporting <u>B</u>, the computer also reports "thresholds" or "cutpoints" (labeled

κ) to help identify the range of each ordered category. If the dependent variable falls into one of

N categories, then the computer reports N-1 cutpoints: $\kappa_1, \kappa_2, \ldots \kappa_{n-1}$.

---

[10] Such a scale is known as a five point *Likert scale*.

Here is how to make your predictions. Compute the regression score $\underline{BX}$. Now compare the regression score to the cutpoints. If $\underline{BX} < \kappa_1$, then you predict the observation will fall in the lowest ordered category. If $\kappa_1 < \underline{BX} < \kappa_2$, the observation is predicted to fall in the next lowest category, and so forth. If $\underline{BX} > \kappa_{n-1}$, the observation is predicted to lie in the highest category. These predictions are not guaranteed, of course. You may also want to compute the probability that the observation falls into each of the N categories.

Let's see how this works in Stata. Suppose that you are trying to determine the factors that affect the number of Internet service providers (**nisp**) in each county in the year 1997, where that number varies from 0 to 6. Your key predictors are population size (**pop92**) and per capita income (**pcinc92**) in 1992. In Stata, you estimate:

      oprobit nisp pop92 pcinc92

Here is part of the results screen:

```
. oprobit nisp pop92  pcinc92 if nisp<6

Iteration 0:    log likelihood = -2830.2083
Iteration 1:    log likelihood = -2659.6169
Iteration 2:    log likelihood = -2615.0194
Iteration 3:    log likelihood = -2603.2241
Iteration 4:    log likelihood = -2602.0934

Ordered probit regression                          Numb
                                                   LR c
Log likelihood =            .                      Prob

       nisp |      Coef.   Std. Err.      z    P>|z|
      pop92 |   8.10e-06   5.88e-07    13.78   0.000
    pcinc92 |   .0000454   8.24e-06     5.52   0.000

       /cut1 |   1.383055    .129824
       /cut2 |   2.123809   .1328186
       /cut3 |   2.520137   .1362879
       /cut4 |   2.875023   .1411154
       /cut5 |   3.403832    .152577
```

Note that you have coefficients on each predictor and 5 cutpoints.

To predict the most likely number of ISPs in each county, you compare $\underline{BX}$ against the cutpoints. For example, a county with 100,000 residents and a per capita income of 25,000 would have $\underline{BX} = 1.945$ (see if you can confirm this for yourself.) This falls between cutpoints 1 and 2, suggesting that this county would have 1 ISP.

You can determine the probability that a given observation falls into any of the satisfaction categories by running your oprobit model and then typing:

predict p0 p1 p2 p3 p4 p5

Stata saves the probabilities as new variables named p0, p1, p2, p3, p4, and p5. For example, the new variable p1 gives the probability that the county has 1 ISP.

Using Ordered Probit to forecast the effects of changing X on the value of Y

Here is how to predict the effect of changing X on Y.

1) Take a baseline category for Y. You should *not* select the lowest category for this exercise. (E.g., category isp=1)

2) Find the value of κ that corresponds to that category. (κ=1.383)

3) Determine the amount that κ must increase to move to the next highest category. Call this difference $\Delta\kappa$. ($\Delta\kappa$ = 2.123-1.383 = 0.74)

4) If Y is to jump from one category to the next, then X must increase by $\Delta\kappa/B_x$. Thus, *the value $\Delta\kappa/B_x$ is a useful measure of the effect of X on Y*

   (Thus, for pop92, we get 0.74/.0000081 = 91,358. That is, it would take 91,358 more residents to go from 1 to 2 ISPs, holding income constant.

   For pcinc92, it would take 0.74/.0000454 = $16,300 additional income, holding population constant.)

Report your results in these terms: "How many more X does it take to get one more Y?"

Ordered Probit and Market Entry

Suppose you have data on lots of markets. Your dependent variable is *firms*, which equals the number of firms in the market. You also have the market *population* and some additional predictors $\underline{X}$.   If you want to determine how these predictors affect the number of firms, you could run a Poisson or negative binomial regression. However, oprobit will allow some economic insights that might not be possible with these other models.   Let me explain.

Stepping back from regression for a moment, consider some basic economics. Suppose that a market requires only 10,000 residents to support one firm, all else equal. You should not expect a market with 20,000 people to support 2 firms. The reason is that competition between the two firms would make the market less profitable on a per resident basis. The additional population required to support 2 firms depends on the degree of competition between them. Oprobit can tell you how big the market must be to support a second or third firm, thereby informing you about oligopolistic interactions. Here is how:

1) Start by regressing: **oprobit firms population $\underline{X}$**

2) Set all the X variables excluding population equal to their mean values. Compute $\underline{BX}$ for all of these variables (again excluding population).

3) Compute $K_1 = \kappa_1 - \underline{BX}$. K_1 is the "gap" in the regression score required to reach cutpoint κ_1 in a market where all the X variables (except population) equal their means.

4) Compute $Pop_1 = K_1/B_{population}$. Pop_1 is the population required to "fill in the gap." With this population, you can expect one firm to enter the market (assuming all other X variables equal their mean values.)

5) Compute $Pop_2 = K_2/B_{population}$, $Pop3 = K_3/B_{population}$, and so forth.   This will give you the *additional* population required for 2 firms, 3 firms, etc. This should provide useful information about market structure and competition.

You will explore these methods in your hospital services project.

Two provisos about using oprobit

1) If you have N possible outcomes, the computer reports N-1 cutpoints. Stata may balk if the LHS variable can take on too many different values. If you still want to run Oprobit, you should group your LHS variable data to reduce the number of categories (e.g. create a new variable whose values represent different ranges of the data, 1=0-10, 2=11-20, etc.)

2) The computer only notices the ordinal ranking of the scores, and not the actual values. If your LHS variable takes on values 0, 2, 3, and 4, then Stata will treat this as if there are four ordered categories. Stata does not create categories for scores for which there are no observations.

Hazard Models (optional)

In a *hazard* model, time passes until some event transpires, and then the outcome is realized. This is common in biology, where a creature is alive until it is dead. In biology, hazard models are also called survival models. The *hazard rate* gives the probability that the event (e.g. death) will occur in any given time period. The measured outcome (e.g., life span) will have a long tail.

The hazard process is common in business. The time to fill an order, the length of time waiting to be served, and the time until the FDA approves a new drug can all be expressed as hazard models. Stata can easily estimate how independent variables affect the hazard rate. However, there are a number of considerations when estimating hazard rates, such as the possibility that the rate changes over time. Full discussion of these considerations is beyond the scope of this course.