Chapter 9
ESTIMATION OF QUALITATIVE DEPENDENT VARIABLE MODELS AND GENERAL ML ESTIMATION

Economists frequently encounter a research problem where the dependent variable of the structural model is not directly observed. For example, the actual value of the variable may be observed only part of the time; whether or not it is observed may depend on its value or on the values of other variables. Alternatively, you may observe a variable that takes on several discrete values related to an underlying unobserved dependent variable. This is the case, for example, when modeling the choice among alternatives such as different types of products or different modes of transportation. For these models, ordinary least squares or other standard econometric estimators are not appropriate, because of the limited or qualitative nature of the observed dependent variable.

The estimators described in this chapter are useful for the following situations:

TOBIT -- The dependent variable is observed only when it lies above some threshold value.

PROBIT -- Only the sign (+/- or 1/0) of the dependent variable is observed, or the dependent variable only takes on two values (binary probit).

SAMPSEL -- The dependent variable is not observed when another unobserved variable in the model lies below a threshold value (this is a generalization of Tobit).

LOGIT -- The dependent variable is the index of a choice among several discrete alternatives (e.g., 1, 2, or 3). There is a "value equation" for each alternative, and the one chosen has the highest value, although actual values are not observed. TSP will estimate models involving characteristics of the choice (conditional logit), the chooser (multinomial logit) or both (mixed logit).

ML -- ML performs maximum likelihood estimation of any model for which the likelihood function can be written in a FRML, or evaluated in a PROC. It includes all of the above estimators as special cases.

The first four procedures are designed for structural equations where the underlying dependent variable is linear in the parameters. Nonlinear structural equations can be specified for the models estimated by these procedures by writing the likelihood explicitly and using ML. All of the models are estimated by maximum likelihood, using common nonlinear maximization algorithms (see Chapter 10 for a discussion of nonlinear methods).

TOBIT, PROBIT, and SAMPSEL models have normally distributed structural disturbances. Logit disturbances have the Generalized Extreme Value distribution (see Manski and McFadden (1981), Maddala (1983), or Train (1986) for further information about this distribution). Estimates obtained from the binary logit model are extremely close to those obtained by probit, up to the implied standard deviation of the disturbances (approximately 1.6 versus 1.0).

Standard errors robust to misspecification of the error term are calculated from first and second derivatives of the likelihood function. They can be obtained for all models by including the HCOV=W or HCOV=NBW option. However, be aware that recent Monte-Carlo work (Calzolari and Fiorentini (1990); Griffiths, Hill, and Pope (1987); Calzolari and Panettoni (1988)) has shown that neither the outer product (HCOV=B) nor the robust estimate (HCOV=W) are as good estimates as the matrix of second derivatives (HCOV=N) in finite sample (approximately 100 observations) when the disturbances are truly normal. The outer product of the analytic gradients tends to overestimate the variance, and the robust estimate to underestimate it. For details, consult the articles cited.

9.1. TOBIT

The Tobit estimator was proposed by James Tobin (1958) when he was analyzing household expenditure on automobiles. The observed expenditure for many households was zero, implying that the desired quantity of automobile services was below the minimum price of a car. Estimation of this model with ordinary least squares will result in coefficients biased toward zero.

TOBIT is specified like OLSQ, with the dependent variable and a list of independent variables. For example,

TOBIT CAREXP C,INCOME,MARSTAT;

The output includes an estimate of SIGMA, the standard deviation of the residuals, since this parameter is estimated jointly with the regression coefficients.

TOBIT assumes that any observations for which the dependent variable takes on a zero or negative value are observations not observed. You can easily estimate a model where the cutoff value is different from zero, or where observations with large values are those not observed, by transforming the dependent variable before estimation. For example, if your dependent variable is Y, to use a cutoff value of THRESH rather than zero, enter

Y = Y-THRESH ;

before the Tobit estimation, and all values of Y that are less than THRESH will be treated as missing. The value of the intercept will be measured relative to THRESH rather than zero, but this can be easily corrected when you report the estimates.

If you want a ceiling for Y rather than a floor, then use the transformation

Y = THRESH-Y ;

before the Tobit statement. THRESH can be zero, in which case the positive values of Y will be treated as though they are missing. In this case, all the slope estimates will have signs opposite to those implied by the original model, and you will need to keep that in mind.

TOBIT produces and stores the same output as OLSQ, except for some of the summary statistics (R-squared, Durbin-Watson, and F-statistic). The residuals for the nonmissing dependent variable observations are stored as @RES and may be printed or plotted.

See Maddala (1983), Chapter 6, and Amemiya (1986), Chapter 10, for further information on the Tobit model. The model estimated by the TOBIT procedure is Amemiya's Type I Tobit. Also see the *Reference Manual* for information on doing Tobit estimation with grouped data using the WEIGHT option.

9.2. PROBIT

Probit is used for analyzing the determinants of a choice between two discrete alternatives, such as working/not working. Since the dependent variable is not continuous, ordinary least squares is not appropriate. Instead, the dependent variable may be treated as an indicator of the sign of a latent continuous dependent variable. That is,

$$\begin{aligned} \text{Latent} &= Xb + e \\ Y &= 1 \text{ if Latent} > 0 \\ Y &= 0 \text{ if Latent} \leq 0 \end{aligned}$$

Often this latent variable has a meaningful interpretation, such as the net value of being in choice 1 versus choice 0. Since the numerical scale of the latent variable is unobservable, the model is identified by normalizing the standard deviation (SIGMA) of the disturbance (e) to one. Because the coefficients do not directly transform the independent variables into the observed (0/1) dependent variable, PROBIT also stores a matrix, @DPDX, that contains the sample mean of the derivative of the probabilities Y=0 and Y=1 with respect to each independent variable. For example:

PROBIT WORKING C SCHOOL EXPER RURAL IND;

Output from PROBIT is similar to that for most of the regression procedures except that a quantity called “deviance R-squared” is also printed. This quantity, known as the Kullbock-Leibler R-squared, is a generalized measure of explanatory power for a wide class of nonlinear models. It is computed as 1-logL (fitted model) / log L (model with intercept only).

PROBIT can also estimate the selectivity bias in any kind of selection model, so that a correction can be computed for a two-stage estimator of such a model. See the example in the SAMPSEL section below.

9.3. Sample Selection: SAMPSEL

The sample selection model is a generalization of the Tobit model when observability of the dependent variable (and possibly the independent variables) in the regression equation is affected by factors other than the value of the dependent variable. In the typology of Amemiya, this is a Type 2 Tobit model. In the original examples developed by Heckman (1974), Griliches, Hall, and Hausman (1978), Hanoch (1980), and others, the regression equation is the log of wages on schooling and experience. Obviously wages are only observed for people in the labor force, and labor force participation is governed by a separate probit equation describing the reservation wage. This equation is also called the selection equation; when its dependent variable is equal to one, the regression equation is observed.

The sample selection model connects the two equations by estimating a correlation ρ between their disturbances. This correlation is implied, for example, when an unobserved characteristic of the individual enters both the wage and selection equations. When ρ is nonzero, estimation of the regression equation in the selected sample where it is observed would result in biased coefficients. To be precise, the model estimated by SAMPSEL is

$$
\begin{array}{lll}
y_1 = & X_1 b_1 + e_1 & (y_1 \text{ not observed}) \\
d_1 = & 1 & \text{if } y_1>0 \\
 & 0 & \text{if } y_1 \leq 0 \\
y_2 = & X_2 b_2 + e_2 & \text{if } d_1=1 \\
 & \text{not observed} & \text{if } d_1=0 \\
\multicolumn{2}{l}{\mathrm{Var}(e_1) = 1 \quad \mathrm{Var}(e_2) = \sigma^2} & \mathrm{Corr}(e_1,e_2) = \rho
\end{array}
$$

The selection (probit) equation and the regression equation are both specified on the SAMPSEL command, separated by a vertical bar (|), with the probit equation first. The wage and labor force participation example mentioned above would be specified as:

```
SAMPSEL  INLABF C SCHOOL EXP MARSTAT | LWAGE C SCHOOL EXP ;
```

The σ (SIGMA) and ρ (RHO) parameters are estimated jointly with the coefficients of both equations. SAMPSEL stores the fitted probabilities and implied inverse Mills ratio for the probit equation, as well as the residuals for the observed subset of observations. See the *Reference Manual* for further information about SAMPSEL's output.

Consistent estimates of the sample selection model can also be obtained with Heckman's two-stage method. This method involves estimating the selection equation with PROBIT, and using the inverse Mills ratio function of the probit residuals (stored by PROBIT in @MILLS) as an extra variable in a regression over the selected sample. For example, the model mentioned above could be estimated by the following commands:

```
PROBIT INLABF C SCHOOL EXP MARSTAT;
SMPLIF INLABF;
OLSQ(ROBUST) LWAGE C SCHOOL EXP @MILLS;
```

Unfortunately, the conventionally estimated standard errors for the coefficients obtained by OLSQ are not consistent estimates. This problem can be avoided, however, by using the ROBUST option in OLSQ to compute heteroskedastic-consistent standard errors. Keep in mind that the maximum likelihood procedure SAMPSEL will produce more efficient estimates (under the assumption of normal disturbances). See Section 8.8 for an example of testing for heteroskedasticity with SAMPSEL.

In practice, extra identifying variables in the selection equation are useful. Without them, the @MILLS term in the regression equation can be a proxy for left-out nonlinear functions of the right-hand-side variables. Accordingly, a correct functional form specification of the regression equation is quite important.

It is also possible to use SAMPSEL to estimate models where only the selected sample is observed (the dependent variable in the selection model is always one). See Bloom and Killingsworth (1985). However, identification of the parameters of the selection equation comes entirely from the implied weighting of the observed data, since there is no longer a separate probit equation. In practice, this makes it difficult to obtain convergence.

9.4. Multinomial and conditional logit: LOGIT

When the dependent variable involves two or more discrete choices, the logit model can be a good way of examining the determinants of these choices. In the case of only two choices, it provides an alternative to the probit model; estimates from the two models will be very similar (see below). Logit models have different names, depending on whether the data or coefficients are choice-specific or chooser-specific. Multinomial logit has chooser-specific data and coefficients vary over the choices. Conditional logit has choice-specific data and coefficients equal over all choices. Mixed logit involves both types of data and coefficients.

McFadden (1973, 1976, 1985) pioneered the use of the logit model in econometrics. The underlying model for most econometric applications involves latent value (utility) equations for each choice. For example, the following latent value equations apply to a model with three choices, with multinomial variable X that has coefficients (b_1,b_2,b_3) and conditional variables (Z_1,Z_2,Z_3) which have coefficients g:

$$V_1 = a_1 + X*b_1 + Z_1*g + e_1$$
$$V_2 = a_2 + X*b_2 + Z_2*g + e_2$$
$$V_3 = a_3 + X*b_3 + Z_3*g + e_3$$

The latent values V_1,V_2,V_3 are not observed, but the chosen alternative is the one with the highest value. If alternate 2 is chosen, for example, we know that $V_2 > V_1$ and $V_2 > V_3$. If the disturbances e_1,e_2,e_3 have the Generalized Extreme Value distribution, the observed choice probabilities have the form

$$\text{Prob}(i) = \exp(a_i+Xb_i+Z_ig) \,/\, \sum_j(\exp(a_j+Xb_j+Z_jg))$$

In this case, the likelihood function is computationally simple, and the estimation method converges rapidly.

The LOGIT command resembles a regression command with the dependent variable followed by a list of the choice-specific variables, and then a list of chooser-specific variables. The two lists are separated by a |. For example, the model outlined above is estimated with the command:

```
LOGIT(NCHOICE=3,COND) Y Z | C X ;
```

When multinomial variables are used, the model is identified by normalizing the multinomial coefficients of the first choice to zero. In the case of two alternatives, this provides compatibility to the PROBIT command, which effectively uses the same normalization; although the coefficients are larger in LOGIT by a factor of approximately 1.6, due to the larger standard deviation of the G. E. V. disturbances.

Conditional logit models may have a variable number of choices per chooser, since the data varies for each choice, and the coefficients are fixed across choices. When setting up data for this model, use one observation for each choice, instead of one observation per chooser. An additional variable, either case number, or number of choices, is used to keep the observations for each chooser together. See the *Reference Manual* for details. Incidentally, the constant C is not a legal conditional variable since it doesn't vary across choices.

Like the binary probit model, logit pertains to a set of mutually exclusive choices (exactly one must be chosen). It is theoretically possible to set up likelihood functions where several alternatives are chosen. It is also possible to extend the probit model to more than two alternatives, with correlations between the disturbances of each value function. However, the multivariate normal integrals required become difficult to compute when the number of choices is greater than four.

The multinomial probit model would be a useful alternative to logit, since the basic logit model does not imply correlations among choices. This shortcoming is the well-known "Independence of Irrelevant Alternatives" property of the multinomial logit model, sometimes called the "Blue Bus/Red Bus" problem: the ratio of probabilities between any two choices is unaffected by the availability of a third choice. See Hausman and McFadden (1984) for a test for this property. There are two possible solutions to this problem within the logit framework.

The nested logit model is one way to structure the alternatives so that some are "closer" to each other than others. Although this model is not currently implemented in TSP, it can be consistently estimated by using the conventional LOGIT procedure in stages. The first stage would estimate the bottom branch of a set of choices, and the succeeding stages would include a predicted "inclusive value" based on the estimates of lower stages in the set of independent variables. See Train (1986) or Manski and McFadden (1982) for details. Also see Section 9.6.10 for an example of how to estimate this model using ML and the TSP examples (on the web site) for a more complex nested logit example.

Another method to avoid the IIA problem uses the fact that logit models can be used to describe *any* set of choice probabilities, not just those implied by the utility model and the GEV distribution (McFadden 1975). In particular, to allow correlation among choices, include variables describing other alternatives in the latent value function for one alternative. The defect of this approach is that it does not specify how many or which variables should be added, and may estimate a large number of parameters. Having a model of the possible relationships among choices will help here.

9.5. General Maximum Likelihood Estimation: ML, EQSUB

The ML procedure provides a very general estimation method. You write the likelihood function for an observation in a FRML statement, and TSP maximizes it with respect to the parameters it contains, using the data in the current sample. ML is quite powerful. It can compute analytic first and second derivatives for use in iteration and in computing standard error estimates. For example, the following commands will estimate a binary probit model (an improved version of this example is discussed later in this section):

```
FRML EQL LOGL = LOG( (1-WORKING)*CNORM(-A-B*X) + WORKING*(1-CNORM(-A-B*X)) );
PARAM A,B;
ML EQL;
```

The automatic differentiation in ML is a great advantage over coding the derivatives by hand in Fortran, Guass, or Matlab (and especially in debugging them). The disadvantage is that execution time is slower and numerical error handling more difficult, so take care when writing the likelihood function to minimize these problems.

Execution time can be reduced by simplifying the FRML as much as possible, and eliminating repeated terms. Repeated terms are best handled with the EQSUB (equation substitution) command, which guarantees that the repeated term will be evaluated only once. EQSUB is also useful for changing the list of explanatory variables in different parts of the likelihood function, so that it is not necessary to rewrite the general equation. See the *Reference Manual* for a fuller discussion of the power of EQSUB. Numerical errors are often reduced by simplifying the function, but be careful to avoid operations like taking the log of zero.

For example, the above probit could be rewritten in two ways. The first involves minimum execution time, but may fail because of an attempt to take the log of zero. It is the model above, with the CNORM(-A-B*X) term defined in another equation, and its coefficient (called WORKYN) evaluated outside the FRML since it depends only on the data and not on the parameters. The EQSUB command is not strictly necessary here, but it is useful for changing the variables in the model and avoiding repeated evaluation.

```
WORKYN = (1-WORKING) - WORKING ;
FRML CNXB CNORM(-A-B*X);
FRML EQL LOGL = LOG( WORKING + CNXB*WORKYN );
EQSUB EQL CNXB;                    ? Substitute the expression for CNXB into the model.
PARAM A,B;
ML(HCOV=N) EQL;
```

On a 386 16 Mhz computer, using a dataset with 385 observations and 8 variables, this example took 98.1 CPU seconds, compared with 5.9 seconds for the PROBIT command.

The second way of writing the LOGL equation avoids the log of zero, but may take slightly more execution time. Generally, this approach is preferred for larger models where extreme values can result in negative arguments to the

LOG function. In this example, the function LCNORM() = LOG(CNORM()) and the relationship CNORM(e) = 1-CNORM(-e) are used.

```
NOWORK = WORKING = 0;
FRML XB A+B*X;
FRML EQL LOGL = NOWORK*LCNORM(-XB) + WORKING*LCNORM(XB);
EQSUB EQL XB;
PARAM A,B;
ML(HCOV=N) EQL;
```

Several other considerations for writing LOGL equations for ML are discussed in the *Reference Manual*.

As in LSQ, the likelihood function equation for the ML procedure may contain logical expressions (such as WAGE>0 or GAMMA*(1- BETA)<=0). However, be aware that if these logical expressions contain parameters to be estimated, the likelihood function will not be differentiable over the whole parameter space, and will not satisfy the usual regularity conditions that guarantee the consistency and asymptotic normality of maximum likelihood estimates. Therefore, take extra care in computing estimates with this type of likelihood function (try several sets of starting values, etc.).

9.5.1 ML PROC

Some likelihood functions are difficult to write in terms of a single FRML, especially time series models with recursive state variables, like GARCH and models with MA(q) residuals. Other models in this class are the Kalman Filter with hyperparameter estimation, simulation estimators like Multivariate Probit, and models with concentrated likelihood functions (although these can often be done in unconcentrated form). For these models, you write a PROC which evaluates the log likelihood, taking as many commands as are necessary. You then tell ML the name of the PROC and give it a list of the parameters to be estimated. It uses numeric derivatives, so it will not be as fast as the FRML method above, but at least it is feasible. For example, the PROBIT model above can be estimated (rather slowly) with:

```
PARAM A,B;
ML MYPROB A,B;
PROC MYPROB;
  LOGL = LOG( WORKING + CNORM(-A-B*X))*WORKYN );
  MAT @LOGL = SUM(LOGL);
ENDPROC;
```

Please see the TSP web page for more advanced examples of ML PROC. There is also a simple ML PROC example in the *TSP Reference Manual* under KALMAN, which shows one way to impose inequality constraints in the Kalman Filter model.

9.6. ML examples

This section presents some examples of using ML. Some of these examples are already built into TSP, like OLSQ and TOBIT. The examples are intended to give an idea of the power of ML, and to provide hints about how to write the likelihood function. See the previous section for an extensive probit example.

9.6.1. OLS

```
? OLS model  (not a recommended way to do OLSQ!)
FRML EQ1 LOGL = LOG(SIGI) + LNORM((Y-XB)*SIGI);
FRML EQXB1 XB = B0 + B1*X;
EQSUB(NAME=OLS) EQ1 EQXB1;
PARAM B0 B1 SIGI; SET SIGI = 1;
ML(HITER=N,HCOV=NBW) OLS;
```

9.6.2. Box-Cox Transformation

```
FRML  RESID  (Y**LAM - 1)/LAM - (A + B*X);          ? Unnormalized equation for Box-Cox
?  Note: it is easiest to do Box-Cox via FIML (which includes the same Jacobian term as above):
FIML(ENDOG=Y) RESID;
?  To use the ML command instead, you need to write the log likelihood, including the Jacobian:
LY = LOG(Y);
FRML EQB LOGL = LOG(SIGI) + LNORM(RESID*SIGI) + (LAM-1)*LY;
EQSUB EQB RESID;
PARAM A,1  B,1  LAM,.5  SIGI,1;
ML(HITER=N,HCOV=NBW) EQB;
```

9.6.3. ARCH(3) model

```
? ARCH (AutoRegressive Conditional Heteroskedasticity) model reference:  Engle (1982) pp. 987-1008.
? See ARCH in Chapter 10, which estimates this model automatically using: ARCH (NAR=3) Y C X ;
? To estimate GARCH models, use the ARCH command, or ML PROC (see web page www.tspintl.com ).
?
? ----- Simulated data -----
SET N=100;
SMPL 1,N;
RANDOM E; U = E;
SMPL 4,N;
U = E*SQRT(1 + .8*U(-1)**2 + .6*U(-2)**2 + .4*U(-3)**2);
SMPL 1,N;
PARAM A 4 B 4;
RANDOM X;
Y = A + B*X + U;
? ----- End of simulated data -----
?
? Estimation model (coefficients in the volatility equation are guaranteed nonnegative by the method of squaring)
? Note: example of the use of a lagged equation (RES in EQH).
FRML RES Y - (A+B*X) ;                                                   ? residual
FRML EQH H = A0**2 + A1**2*RES(-1)**2 + A2**2*RES(-2)**2 + A3**2*RES(-3)**2 ;  ?ARCH(3) model.
FRML ARCH LOGL = -LOG(H)/2 + LNORM(RES/SQRT(H));                         ?volatility eqn.
? declare PARAMs before using EQSUB, so that it knows they do not have to be lagged
? (as opposed to the Y and X variables, which do get lagged when the eqn. RES is laggd)
PARAM A B A0-A3;
EQSUB ARCH EQH RES;                                    ?substitute in for H and residuals
?
? Starting values from OLS, small lag coefficients in volatility equation.
SMPL 4,N;
OLSQ Y C,X;
UNMAKE @COEF A B;
SET A0 = @S; PARAM A1,.1 A2,.1 A3,.1;
?
? Estimation by ML
ML ARCH;
?
? Generate the estimated residuals RES and fitted variance series H
GENR RES UHAT;
EQSUB EQH RES;   GENR EQH;
```

9.6.4. Frontier production model

```
? y = f(x,b) + e,  e = -u + v,  v ~ N(0,σ²),  u is non-negative
? For a frontier cost model, e = u + v.
? See Maddala (1983) pp. 194-196.
? Note that the regularity conditions for asymptotic normality are not satisfied by this model.
?
FRML RESID E = Y - A - B*X;
PARAM A,B;
FRML FRONTP LOGL = LOG(2) + LOG(SIGI) + LNORM(E*SIGI) + LCNORM(-E*LAMBDA*SIGI);
? for cost model, there is no - sign here.
PARAM LAMBDA,SIGI;
EQSUB FRONTP RESID;
? crude starting values from OLS -- there are better formulas available
OLSQ(SILENT) Y C X;  UNMAKE @COEF A,B;
SET SIGI = 1/@S;  SET LAMBDA = .1;
?
ML FRONTP;
```

9.6.5. Basic Tobit Model

```
? Tobit model, starting values from OLS
Y0 = Y <= 0; Y1 = Y > 0;                    ? create dummies for obs. And unobs. Y.
FRML EQ1 LOGL = Y0*LCNORM(-XB/SIG) + Y1*[-LOG(SIG) + LNORM((Y-XB)/SIG)];
FRML EQXB1 XB = B0 + B1*X;                  ? define regression model
EQSUB(NAME=TOBIT) EQ1 EQXB1;
PARAM B0 B1 SIG;
OLSQ Y C X;  UNMAKE @COEF B0 B1;  SET SIG = @S;
ML(HITER=N,HCOV=NBW) TOBIT;
```

9.6.6. Tobit reparametrized for global concavity

```
FRML EQ1 LOGL = Y0*LCNORM(-XB) + Y1*[LOG(SIGI) + LNORM((Y-XB)*SIGI)];
FRML EQXB1 XB = R0 + R1*X;
EQSUB(NAME=TOBITGC) EQ1 EQXB1;
PARAM R0 R1 SIGI;
OLSQ Y C X;  UNMAKE @COEF B0 B1;  SET SIGI = 1/@S;
SET R0 = B0*SIGI;
SET R1 = B1*SIGI;
ML(HITER=N,HCOV=N) TOBITGC;
FRML F1 B0 = R0/SIGI;
FRML F2 B1 = R1/SIGI;
FRML F3 SIG = 1/SIGI;
ANALYZ F1-F3;              ? ANALYZ obtains values and standard errors of original parameters
```

9.6.7. Multinomial Logit

```
?  Multinomial Logit (X=characteristics of the chooser)
?
?  Simulate data with 3 choices and 2 X's.
CONST N 1000 ;                      ? Sample size.
SMPL 1 N ;
?
```

```
DOT 1 2 ;                                   ? X's are uniform random variables on (0,1).
  RANDOM (UNIFORM) X. ;
ENDDOT ;
DOT 1 2 3 ;                                 ? 3 choices.
  RANDOM (UNIFORM) F;                       ? inverse distribution funcion
  U. = -LOG(-LOG(F)) ;                      ? for Type I extreme value distn.
ENDDOT ;
?
Y1 =              U1 ;                      ? all coefs. are zero for 1st alt.
Y2 = -1 + X1 + X2 + U2 ;                    ? coefs. are (-1,1,1).
Y3 = -1 + X1 + X2 + U3 ;                    ? coefs. are (-1,1,1).
Y_1 = Y1>Y2 & Y1>Y3 ;
Y_2 = Y2>Y1 & Y2>Y3 ;
Y_3 = Y3>Y1 & Y3>Y2 ;
Y = 1*Y_1 +2*Y_2 +3*Y_3 ;                   ? y is dep var (choice).
?
?  Define the logit model for ML procedure and estimate it.
fRML LOGIT Y_2*XB2 +Y_3*XB3 - LOG(1 + EXP(XB2) + EXP(XB3)) ;
FRML XB2 B20+B21*X1+B22*X2 ;
FRML XB3 B30+B31*X1+B32*X2 ;
EQSUB LOGIT XB2 XB3 ;
PARAM B20-B22 B30-B32 ;
ML (HITER=N,HCOV=NBW) LOGIT ;               ? Use Newton-Raphson iteration.
```

9.6.8. Sample Selection

```
?  Sample Selection model using starting values from OLS and PROBIT
?  Note: |rho| > 1 can cause problems -- one way to avoid this is to reparameterize  rho = tan(theta) instead.
?  The SAMPSEL command uses a similar reparameterization during iterations.
?
?  SAMPSEL IY C Z | Y C X;
?
IY0 = IY = 0;                                       ? Dummy for Y not observed.
FRML EQ1 LOGl = IY0*LCNORM(-ZD) + IY*[-LOG(SIGMA) + LNORM(YXB/SIGMA) +
        LCNORM( [ZD + RHO*YXB/SIGMA]*[1-RHO*RHO]**(-.5) ) ];
FRML ZD1 ZD = D_C + D_Z*Z;
FRML YXB1 YXB = Y - B_C - B_X*X;
EQSUB(NAME=SAMPSEL) EQ1 YXB1 ZD1;
PARAM D_C D_Z B_C B_X SIGMA RHO;
PROBIT IY C Z; UNMAKE @COEF D_C D_Z;                ? Probit for parameter starting values.
COPY @SMPL FULL; SMPLIF IY;                         ? OLSQ over observed data for starting values.
   OLSQ Y C X; UNMAKE @COEF B_C B_X; SET SIGMA=@S;
SMPL FULL;
ML(HCOV=NBW) SAMPSEL;                               ? ML on sample selection model over full sample.
```

9.6.9. Ordered Probit

```
?  Ordered Probit model  (4 states)
?  Q = XB + U   (Q unobserved)
?  Y =1  for        Q <= 0       (          U <= -XB   )
?     2  for  0  <  Q <= A1 (   -XB    <    U <= A1-XB      )
?     3  for A1  <  Q <= A2 ( A1-XB  <    U <= A2-XB      )
?     4  for A2  <  Q            ( A2-XB  <        U          )
?
```

```
? To insure that the ordering satisfies  0 <= A1 <= A2  , parametrize as A1 = D1**2, A2 = A1 + D2**2.
? However, if D1 or D2 tends to zero, numerical problems will arise due to taking the log of zero.
?
Y1 = Y=1; Y2 = Y=2; Y3 = Y=3; Y4 = Y=4;
?
?FRML EQ1 LOGL = LOG{
?  Y1*CNORM(-XB) +
?  Y2*[CNORM(A1-XB) - CNORM(-XB)] +
?  Y3*[CNORM(A2-XB) - CNORM(A1-XB)] +
?  Y4*[1 - CNORM(A2-XB)]
?  };
Y12 = Y1 - Y2; Y23 = Y2 - Y3; Y34 = Y3 - Y4;
FRML EQ1 LOGL = LOG{
  Y12*CNORM(-XB) + Y23*CNORM(A1-XB) +
  Y34*CNORM(A2-XB) + Y4
  };
?  The model above is more efficient for estimation.
FRML EXB XB = B0 + B1*X1 + B2*X2;
FRML EA1 A1 = D1*D1;
FRML EA2 A2 = A1 + D2*D2;
PARAM B0 B1 B2 D1 .1  D2 .1 ;
?
? Use Probit on Y=1 vs. Y>1 to obtain starting values for B coefficients.
?
YY = Y > 1;
PROBIT YY C X1 X2;
UNMAKE @COEF B0 B1 B2;
EQSUB(NAME=ORDPROB) EQ1 EA2 EA1 EXB;                    ? note substitution order
ML(HITER=N,HCOV=NBW) ORDPROB;
```

9.6.10. Nested Logit

```
? Nested Logit using ML, including example of IIA Testing.
? This example is for the simplest nested logit model, where the top branch is a choice between alternative 1 (with
? characteristics X1) and the lower branch. The lower branch is a choice between alternative 2 (with char. X2) and
? alternative 3 (with char. X3). Alternatives 2 and 3 are correlated;  the inclusive value parameter is denoted lambda.
? Note that the coefficients of the lower branch must be multiplied by lambda to obtain estimates that can be
? compared to ordinary MN logit.
?
? The likelihood for an individual observation is
? LOGL = D(1 chosen)L(1|X) + D(2 chosen)L(2|X) + D(3 chosen)L(3|X)
? where D(.) is a zero-one variable denoting which choice was made, and L(i|X) is the likelihood of that choice
? conditional on the Xs.
?
? References:
? Maddala (1983), p. 71 (with lambda=1-sigma),  McFadden (1987), pp. 63-82.
?
? Simulate data with 3 choices and 2 X's; data satisfy IIA.
? CONST N 1000 ;           ? Sample size.
SMPL 1 N ;
?
DOT 1 2 3 ;                ? 3 choices.
    DOT 1 2 ;              ? X's are uniform random variables on (0,1).
        RANDOM (UNIFORM) X.. ;
    ENDDOT ;
```

```
        RANDOM (UNIFORM) F;                     ? inverse distribution function
        U. = -LOG(-LOG(F)) ;                    ? for Type I extreme value distn.
    ENDDOT ;
    Y1 =            U1 ;                        ? normalize coefs. to zero for 1st choice.
    Y2 = -1 + X21 + X22 + U2 ;                  ? coefs. are (-1,1,1)
    Y3 = -1 + X31 + X32 + U3 ;                  ? coefs. are (-1,1,1)
    Y_1 = Y1>Y2 & Y1>Y3 ;                       ? y_1, y_2, y_3 are a set of dummies
    Y_2 = Y2>Y1 & Y2>Y3 ;                       ? equal to one if this obs made this
    Y_3 = Y3>Y1 & Y3>Y2 ;                       ? choice, zero otherwise.
    ?
    ? Define Nested Logit Model.
    FRML XB1N BETA01 ;
    FRML XB2N (BETA02+BETA1*X21+BETA2*X22)/LAMBDA ;
    FRML XB3N (BETA03+BETA1*X31+BETA2*X32)/LAMBDA ;
    FRML SUM23 EXP(XB2N) + EXP(XB3N) ;
    FRML DENOM EXP(XB1N) + SUM23**LAMBDA ;
    FRML NLOGIT -LOG(DENOM) + Y_1*XB1N
          + Y_2*((LAMBDA-1)*LOG(SUM23)+XB2N)
          + Y_3*((LAMBDA-1)*LOG(SUM23)+XB3N) ;
    EQSUB NLOGIT DENOM SUM23 XB1N-XB3N ;
    ?
    TITLE "Multinomial Logit using Nested Logit Likelihood' ;
    PARAM BETA02 -1 BETA03 -1 BETA1 1 BETA2 1 ;
    CONST BETA01 0 ;                            ? One of the intercepts must be set to zero.
    CONST LAMBDA 1 ;                            ? Collapses to MN Logit model when lambda=1.
    ML NLOGIT ;                                 ? These estimates are for the null hypothesis.
    COPY @LOGL LOGL0 ;                          ? They are efficient if IIA holds (lambda=1).
    COPY @COEF BETAE ; COPY @VCOV VCOVE ;       ? Save for the Hausman-McFadden test below.
    ?
    TITLE 'Nested Logit Model with upper branch 1, (2,3)' ;
    PARAM LAMBDA ;
    ML NLOGIT ;
    COPY @LOGL LOGL1 ;
    ?
    TITLE 'Wald-type test for IIA' ;
    FRML WALD4IIA 1-LAMBDA ;
    ANALYZ WALD4IIA ;
    ?
    TITLE 'Likelihood ratio test for IIA' ;
    SET LR4IIA = -2*(LOGL0-LOGL1) ;
    CDF (CHISQ,DF=1) LR4IIA ;
    ?
    ? Hausman-McFadden Test for IIA (dropping first alternative).
    ? Note that proc haustest adjusts for coefficients that are not identified in the reduced (consistent) model. (beta02)
    ? First we estimate the lower branch of the model; estimates are consistent even if IIA property does not hold.
    FRML NLOGIT2 -LOG(SUM23) + Y_2*XB2N + Y_3*XB3N ;
    EQSUB NLOGIT2 SUM23 XB2N XB3N ;
    CONST LAMBDA 1 ;
    SELECT Y>1 ;                                ? Sample: subset that chooses 2 or 3.
    ML (SILENT) NLOGIT2 ;
    ?
    TITLE 'Hausman-McFadden Test for IIA' ;
    COPY @COEF BETAC ; COPY @VCOV VCOVC ;
    HAUSTEST BETAE VCOVE BETAC VCOVC ;          ? See section 13.4.1 for this procedure.
```

9.6.11. Switching regression

```
? A Disequilibrium model: the observed quantity, Y, is the minimum of the quantity supplied (equation 1) and the
? quantity demanded (equation 2).  The sample selection is unknown, i. e., for any given observation on price and
? quantity, it is not known whether it is supply or demand constrained.  See Maddala (1983), p.298 (10.21).
?
FRML U1EQ U1 = (Y-(B01-B11*P-B21*X1)) ;
FRML U2EQ U2 = (Y-(B02-B21*P-B22*X2)) ;
FRML LOGLEQ LOGL = LOG((NORM(U2/SIG2)/SIG2) * (1-CNORM(U1/SIG1))
                    +  (NORM(U1/SIG1)/SIG1) * (1-CNORM(U2/SIG2))) ;
EQSUB LOGLEQ U1EQ U2EQ ;
?
?  Use OLS to obtain starting values for ML estimation.
OLSQ Y C P X1 ;
UNMAKE @COEF B01 B11 B12 ;
SET SIG1 = @S ;
OLSQ Y C P X2 ;
UNMAKE @COEF B02 B21 B22 ;
SET SIG2 = @S ;
?
? Now do MLE on switching regression model.
PARAM SIG1 SIG2 B01 B02 B11 B12 B21 B22 ;
ML LOGLEQ ;
```

9.6.12. Poisson and negative binomial models

```
? Note: an extended version of these examples, including use in a panel data setting is given in our web site.
? The likelihood function for a Poisson model is L = exp(-lambda(i)) lambda(i)**y(i) / y(i)!
?
FRML LAMBEQ LAMBI=EXP(ALPHA+BETA*X) ;
FRML FISHEQ LOGL = -LAMBI+Y*LOG(LAMBI) - LFACT(Y) ;
EQSUB FISHEQ LAMBEQ ;
PARAM ALPHA .1 BETA .05 ;
ML FISHEQ ;
?
? The negative binomial likelihood function is a generalization of the Poisson which allows for a variance
? different from the mean.  Note that this function has been set up so that the same regression function
? (lambi) will be used for both the Poisson and the negative binomial.  See Hausman, Hall & Griliches (1984).
?  See also the discussion of RANDOM(NEGBIN) in the TSP Reference Manual.
?
FRML NBEQ LOGL = LGAMFN(LAMBI+Y) - LGAMFN(LAMBI) - LGAMFN(Y+1) +
                                  LAMBI*LOG(DELTA) - (LAMBI+Y)*LOG(1+DELTA) ;
EQSUB NBEQ LAMBEQ ;
PARAM DELTA .5 ;
ML NBEQ ;
```

9.6.13. Bivariate probit model

The following approximation for the bivariate normal cumulative distribution function is useful for computing two-equation selection-type models:

```
CDF(XB1,XB2,rho) = cnorm(XB1)*cnorm(XB2) + norm(XB1)*norm(XB2)*
                   (rho + rho**2*XB1*XB2/2  +  rho**3*(XB1**2 - 1)*(XB2**2 - 1)/6 )
```

approximates the following CDF

```
CDF(XB1,XB2,rho) = Pr (e1 < XB1 and e2 < XB2)
                    where E[e1*e1] = 1, E[e2*e2] = 1, and E[e1*e2] = rho .
```

It is just a Taylor expansion around rho for small rho. Be careful with the sign of XB1 and XB2.

The bivariate normal distribution can be used to estimate a bivariate probit model, for example:

```
FRML PR00 CNORM(XB1)*CNORM(XB2) + NORM(XB1)*NORM(XB2)*
      (RHO + RHO**2*XB1*XB2/2 + RHO**3*(XB1**2 - 1)*(XB2**2 - 1)/6 );
FRML XB1 -(A1 + X*B1 + Z*G1);
FRML XB2 -(A2 + X*B2 + Z*G2);
PARAM A1 A2 B1 B2 G1 G2 RHO;
?
? Bivariate probit
?  y1 = A1 + X*B1 + Z*G1
?     = -XB1 + e1
?     if y1 < 0, e1 < XB1 and D1 = 0
?  y2 = -XB2 + e2
?     if y2 > 0, e2 > XB2 and D2 = 1
?  D01 = 1 if D1=0 and D2=1
?
FRML BVPROB LOGL = D00*LOG(PR00) + D01*LOG(CNORM(XB1) - PR00) +
                                   D10*LOG(CNORM(XB2) - PR00) +
                                   D11*LOG(1 - (CNORM(XB1) + CNORM(XB2) - PR00));
EQSUB BVPROB PR00 XB1 XB2;
?
? Starting values from binary probit
D1X = D10 + D11;
PROBIT D1X C X Z;
UNMAKE @COEF A1 B1 G1;
DX1 = D01 + D11;
PROBIT DX1 C X Z;
UNMAKE @COEF A2 B2 G2;
?
? Watch out for probabilities that are exactly equal to zero and one.
GENR BVPROB;  SELECT ^MISS(LOGL);
ML BVPROB;
SELECT 1;
ML BVPROB;
```

The table below shows the average absolute error of this Taylor expansion approximation (Abramowitz and Stegun, formula 26.3.29), based on 100 trials

RHO	Average absolute error
.1	.0000025
.2	.0000092
.3	.0000389
.4	.0001527
.5	.0003845
.6	.0009554
.7	.0020995
.8	.0040833
.9	.0088709

9.6.14. Hazard function

```
? This log-linear hazard rate example is a variation of the Weibull model.  See Lancaster () for further information.
?
? P(X,t) is the conditional probability that an individual who is employed at time t is still employed at time t+1.
?       P(X,t) = 1/[1+exp(Xb + a*t)]
? For a person who become unemployed at time T, the likelihood function is
?       P(X,1)*P(X,2)*...*P(X,T-1)*[1-P(X,T)]
? Toss some sample data. There is a single covariate x, normally distributed with mean zero and variance 1. Assume
? there are no time-varying covariates (no t term) for data generation.  Therefore true value of the parameter A
? (below) will be zero.
?
CONST NOBS 100 ;
SMPL 1 NOBS ;
RANDOM X ;
?
PARAM A 0 B0 .5 B1 1 ;
DO IOBS = 1 TO NOBS ;
      SMPL IOBS IOBS ;
      SET LOGTHET = LOG(1+EXP(X(IOBS)*B1+B0))-.5772 ;
      RANDOM(MEAN=LOGTHET,STDEV=1.6449) LOGT ;
ENDDO ;
SMPL 1 NOBS ;
T = INT(EXP(LOGT)+.999) ;
MSD T LOGT X ;
HIST (DISCRETE,NBINS=100,WIDTH=1) T ;
SMPLIF T>=0 & T<21 ;
?
? The program below shows how to automate the likelihood function construction if you know the maximum
?  number of periods of observation.
?
?FRML HAZARD LOGL = TERMS +(XB + A*T) - LOG[1+EXP(XB + A*T)];     ? TERMS + LOG[1-P(X,T)]
FRML TRICK MORE = TERMS;
SET TI = 0;
DOT 1-20;                                                ? E.g., the maximum for T is 20
 SET TI = TI+1;
 SET T. = TI;                                            ? T1=1, T2=2, etc.
 FRML TERMS -(T>T.)*LOG(1+EXP(XB + A*T.)) + MORE;        ? this term is non-zero if t > t.
 EQSUB HAZARD TERMS TRICK;                               ? the trick turns more back into terms
ENDDOT;
PRINT HAZARD ;
FRML XB B0 + B1*X ;                                      ? Xb formula -- user-defined
                                                         ? t is the "dependent variable"
FRML LAST TERMS = 0;
EQSUB HAZARD LAST XB;
?
?  Maximum likelihood estimation.
?
ML (MAXIT=50) HAZARD;
```