TOBIT ANALYSIS USING SAS

Cathryn Gust, Senate Computer Center
Rebecca Klemm, Georgetown University

"What do you mean less than nothing?" replied Wilbur. "I don't think there is any such thing as less than nothing. Nothing is absolutely the limit of nothingness. It's the lowest you can go. It's the end of the line. How can something be less than nothing? If there were something that was less than nothing than nothing would not be nothing, it would be something--even though it's just a very little bit of something. But if nothing is nothing, then nothing has nothing that is less than it is."

E. B. White, Charlotte's Web (New York: Harper, 1952) p. 28

INTRODUCTION

Tobit analysis, first proposed by James Tobin (5), estimates the parameters of a regression model when the dependent variable is limited by some upper or lower bound. This limit is often zero, as in the case of the amount annually expended by a family for an expensive durable good. Every year many families refrain from purchasing this good. Thus the limiting value of the dependent variable not only exists as a possibility, but is often observed. Under such conditions, Tobit estimation can be viewed as an extension of Probit analysis; families are not merely classified as purchasers or non-purchasers, but the magnitude of the expenditure is maintained as relevant.

Tobit estimates were originally computed by various modifications of Newton's method. These are discussed in Tobin (5) and Amemiya (1). Ray Fair (3) proposed an alternative procedure which has reduced the computing time for many problems although it generally requires more iterations than the procedures based on Newton's method. Program steps and notation presented in this paper closely follow the procedure outlined by Fair.

THE MODEL

The model is

$$y_t = \beta_o{}'X_t + u_t \text{ if RHS} > 0$$
$$= 0 \text{ otherwise} \qquad t = 1, 2, \ldots T$$

where β_o is a Kx1 vector of unknown coefficients, X_t is a Kx1 vector of values of the explanatory variables for observation t, and the error term u_t is independently distributed $N(0,\sigma^2)$.

Y_t is positive for R observations and zero for S=T-R observations.

THE PROGRAM

The SAS program presented in this paper iteratively calculates the Tobit estimates of β_o and σ^2 using the MATRIX procedure in SAS. The estimation procedure is considered to have achieved convergence when the magnitude of the greatest change in the estimate of β_o on successive iterations is less than the tolerance level of .001 (SAS Statement 98). There is no guarantee that the procedure will converge and the program ceases processing if the convergence criterion has failed to be met in 100 iterations.

The authors are currently investigating the optimal starting values for β_o. This program begins with either a vector of zeros or the ordinary least squares estimates for β_o for the R observations with positive y_t. The vector of zeros is selected if the number of observations with y_t equal to zero exceeds the number of observations with nonzero y_t (SAS Statements 78 and 79).

Fair (3) suggests some value between .3 and .5 for the damping factor (lambda). This program initially sets the value of lambda at .5 and reduces it at the rate of .05 for each successive round of 20 iterations (SAS Statement 83).

The SAS program presented in this paper outputs the number of iterations required for convergence, the variance-covariance matrix of (β_o, σ^2), and the standard table of regression estimates with their standard errors. Sample output is displayed in Figure 1.

The matrix procedure outputs a SAS data set containing the observed y_t values, the predicted y_t values, the residuals and the X_t values. This facilitates further investigation of the residuals via SAS graphic and analytical procedures and also allows comparisons with the residuals obtained using ordinary least squares methods.

ACKNOWLEDGEMENTS

The authors wish to thank Jerry Gordon and Ron Oaxaca for introducing Tobit analysis to them and for providing the test example mentioned in this paper.

REFERENCES

1. Amemiya, Takeshi: "Regression Analysis When the Dependent Variable is Truncated Normal," Econmetrica, 41(1973), 997-1016.

2. Dolk, Dan: "Tobit Program Version 1.0," FORTRAN documentation, University of Arizona, (1979).

3. Fair, Ray: "A Note on the Computation of the Tobit Estimator," Econometrica, 45(1977), 1723-1727.

4. Poirier, Dale: The Use of the Box-Cox Transformation in Limited Dependent Variable Models," Journal of the American Statistical Association, 73(1978), 284-287.

5. Tobin, James: "Estimation of Relationships for Limited Dependent Variables," Econometrica, 26(1958), 24-36.

SAS LOG OF TOBIT ANALYSIS

```
          OPTIONS NODATE NONUMBER;
          DATA ALL NONZEROS ZEROS;
            INPUT YEAR UM E ENC SL PR;
            URATIO = UM / E;
            CRATIO = SL / ENC;
            IF (CRATIO >  0) THEN  OUTPUT ALL NONZEROS;
            ELSE OUTPUT ALL ZEROS;
            CARDS;

NOTE: DATA SET WORK.ALL HAS 50 OBSERVATIONS AND 8 VARIABLES. 191 OBS/TRK.
NOTE: DATA SET WORK.NONZEROS HAS 38 OBSERVATIONS AND 8 VARIABLES. 191 OBS/TRK.
NOTE: DATA SET WORK.ZEROS HAS 12 OBSERVATIONS AND 8 VARIABLES. 191 OBS/TRK.
NOTE: THE DATA STATEMENT USED 0.74 SECONDS AND 116K.

          PROC MATRIX;                                       * NOTATION FOLLOWS FAIR (3);
           SETUP:
            FETCH Y DATA=NONZEROS(KEEP = CRATIO);              * Y: NONZERO OBS OF DEP VAR;
            FETCH NONZEROS
                  DATA=NONZEROS(KEEP = PR URATIO);
            K = NCOL(NONZEROS) + 1;                            * K: # OF EXPLANATORY VARS;
            R = NROW(NONZEROS);                                * R: # OF NONZERO OBS;
            X = J(R,1) || NONZEROS;                            * X: X MATRIX FOR NONZERO OBS;
            FETCH ZEROS
                  DATA=ZEROS(KEEP = PR URATIO);
            S = NROW(ZEROS);                                   * S: # OF ZERO OBS;
            XBAR = J(S,1) || ZEROS;                            * XBAR: X MATRIX FOR ZERO OBS;
            FREE ZEROS NONZEROS;
            T = R + S;                                         * T: TOTAL # OF OBS;
           STEP1:
            XCOV = INV((X') * X);
            LSBETA = XCOV * (X') * Y;                          * LSBETA: OLSQ BETA FOR NON-
                                                                 ZERO OBS;
            XCOVXBAR = XCOV * (XBAR');
           INIT:;                                              * INITIAL VALUE FOR BETA;
            IF (S > R) THEN BETA = J(K,1,0);
            ELSE BETA = LSBETA;
            NORMCOEF = 1 #/ (SQRT(2 # ARCOS(-1)));             * ARCOS(-1) = PI;
           STEP2:
            DO ITRCT = 1 TO 100;                               * ITRCT: ITERATION COUNTER;
            LAMBDA = .50 - (INT(ITRCT #/ 20) # .05);           * LAMBDA: DAMPING FACTOR;
            SIGMA2 = ((Y') * (Y - (X * BETA))) #/ R ;
            IF (SIGMA2 < 0) THEN SIGMA2 = .5 ;                 * SET SIGMA2 = SMALL + VALUE
                                                                 IF INITAL VALUE < 0;
            SIGMA = SQRT(SIGMA2);
           STEP3:
            ZBAR = (XBAR * BETA) #/ SIGMA;
            PHI = NORMCOEF # EXP(-.5 # (ZBAR ## 2));
            PHICAP = PROBNORM(ZBAR);
            GAMMABAR = PHI #/ (1 - PHICAP);
           STEP4:
            BETASQGL = LSBETA -
             SIGMA # (XCOVXBAR * GAMMABAR);                    * BETASQGL: INTERMEDIATE BETA;
            BETANEW = BETA + (LAMBDA # (BETASQGL - BETA));     * BETANEW: NEW BETA ESTIMATE;
           TEST:
            IF (MAX(ABS(BETANEW - BETA)) < .001)  THEN DO;     * TEST FOR CONVERGENCE;
              BETAHAT = BETANEW;
              GO TO PREDCALC;
            END;
            ELSE BETA = BETANEW;
           END;
           OUTLOOP:
            NOTE FAILURE TO REACH CONVERGENCE;                 * 100 ITERATION LIMIT;
            STOP;
           PREDCALC:;                                          * CALCULATE PREDICTED VALUES;
            FETCH ALL DATA=ALL(KEEP = PR URATIO)
             COLNAME=INDNAMES;
            XT = J(T,1) || ALL;                                * XT: X MATRIX FOR ALL OBS;
            FETCH YOBS DATA=ALL(KEEP = CRATIO)
             COLNAME=DEP_VAR;                                  * YOBS: DEP VAR FOR ALL OBS;
            FETCH YID DATA=ALL(KEEP=YEAR) COLNAME=YIDNAME;
            FREE ALL;
            YHAT = (XT * BETAHAT);                             * YHAT: PREDICTED Y VALUES;
            S2HAT = ((Y') * (Y - (X * BETAHAT))) #/ R;         * S2HAT: EST OF SIGMA ** 2;
            SIGMAHAT = SQRT(S2HAT);
            SSQTOTAL = SSQ(Y) - ((SUM(Y) ## 2) #/ T);          * SSQTOTAL: TOTAL SUM OF SQS;
            ZT = (YHAT) #/ SIGMAHAT;
            RESIDUAL =  YOBS - YHAT;
           VARCALC:;                                           * VAR-COV MATRIX CALCULATION;
            PHIT = NORMCOEF # EXP(-.5 # (ZT ## 2));
            PHICAPT = PROBNORM(ZT);
            A=(((PHIT ## 2) #/ (1 - PHICAPT)) +
             PHICAPT - (PHIT # ZT)) #/ S2HAT;
            B=(((ZT ## 2) # PHIT) + PHIT -
             ((ZT # (PHIT ## 2)) #/ (1 - PHICAPT)))
             #/ (2 # (SIGMAHAT ## 3));
            C=((((ZT ## 2) # (PHIT ## 2) +
             (2 # PHICAPT)) #/ (1 - PHICAPT))
             - ((ZT # PHIT) + ((ZT ## 3) # PHIT)))
             #/ (4 # (S2HAT ## 2));
            SUMAXXT = J(K,K,0);
```

```
        DO I = 1 TO T;
          SUMAXXT = SUMAXXT +
            (A(I,1) # ((XT(I,*)') * XT(I,*)));
        END;
        VCOV = INV((SUMAXXT       || ((XT') * B))
                // (((B') * XT) || (SUM(C))));                * VCOV: VAR-COV MATRIX FOR
                                                                 (BETA, SIGMA ** 2);
       PRTOUT:
        CONSTANT = '         ';
        NOTE CONVERGENCED IN ITRCT ITERATIONS;
        PRINT ITRCT COLNAME=CONSTANT ROWNAME=CONSTANT;
        PRINT FORMAT=$CHAR8. DEP_VAR COLNAME=CONSTANT
                                     ROWNAME=CONSTANT;
        CONSTANT = 'CONSTANT';
        RLABEL1 = CONSTANT // (INDNAMES');
        CLABEL1 =  'ESTIMATE' || 'T-STAT   ' ||
                   'PR > |T|' || 'STDERROR' ;
        BETASE = SQRT(VECDIAG(VCOV(1:K,1:K)));
        TSTATS = BETAHAT #/ BETASE;
        TPROB = (1 - PROBT(ABS(TSTATS),J(K,1,R))) # 2;
        IND_VAR = BETAHAT || TSTATS || TPROB || BETASE;
        PRINT IND_VAR FORMAT=16.8 ROWNAME = RLABEL1
                                  COLNAME = CLABEL1;
        CONSTANT = 'SIGMA**2';
        RLABEL2 = RLABEL1 // CONSTANT;
        CLABEL2 = (RLABEL1') || CONSTANT;
        NOTE VAR-COV MATRIX FOR (BETA, SIGMA**2): ;
        PRINT VCOV FORMAT=16.8 ROWNAME=RLABEL2
                               COLNAME=CLABEL2;
        CONSTANT = '         ' ;
        PRINT SIGMAHAT S2HAT SSQTOTAL ROWNAME=CONSTANT
                                      COLNAME=CONSTANT;
        YLABEL = 'Y_OBS    ' || 'Y_HAT    ' || 'RESIDUAL';
        CLABEL3 = YIDNAME || DEP_VAR || YLABEL(*,2:3);
        YPRT = YID || YOBS || YHAT || RESIDUAL ;
        PRINT YPRT FORMAT=12.6 COLNAME=CLABEL3;
        CLABOUT = INDNAMES || CLABEL3;
        YOUT = XT(*,2:K) || YPRT;
        OUTPUT YOUT OUT=TOBITOUT(DROP=ROW)
         COLNAME=CLABOUT;
      TITLE1 TOBIT ANALYSIS USING THE SAS MATRIX PROCEDURE;

NOTE: DATA SET WORK.TOBITOUT HAS 50 OBSERVATIONS AND 6 VARIABLES. 250 OBS/TRK.
NOTE: THE PROCEDURE MATRIX USED 2.98 SECONDS AND 166K AND PRINTED PAGES 1 TO 2.
```

FIGURE 1. SAMPLE OUTPUT FROM TOBIT ANALYSIS PROCEDURE USING SAS

TOBIT ANALYSIS USING THE SAS MATRIX PROCEDURE

CONVERGENCED IN ITRCT ITERATIONS

ITRCT

7

DEP_VAR

CRATIO

IND_VAR	ESTIMATE	T-STAT	PR > |T|	STDERROR
CONSTANT	0.26765447	1.11505199	0.27183047	0.24003766
PR	-0.75376441	-0.77340500	0.44406740	0.97460504
URATIO	0.57245086	0.96618464	0.34006082	0.59248599

VAR-COV MATRIX FOR (BETA, SIGMA**2):

VCOV	CONSTANT	PR	URATIO	SIGMA**2
CONSTANT	0.05761808	-0.19208061	-0.12381475	-0.00001945
PR	-0.19208061	0.94985498	0.28502116	-0.00008774
URATIO	-0.12381475	0.28502116	0.35103964	0.00004712
SIGMA**2	-0.00001945	-0.00008774	0.00004712	0.00009031

SIGMAHAT

0.330691

S2HAT

0.109357

SSQTOTAL

3.62327

YPRT	YEAR	CRATIO	Y_HAT	RESIDUAL
ROW1	21.000000	0.000000	0.293363	-0.293363
ROW2	22.000000	0.000000	0.267600	-0.267600
ROW3	23.000000	0.000000	0.411741	-0.411741
ROW4	24.000000	0.000000	0.186800	-0.186800
ROW5	25.000000	0.000000	0.205410	-0.205410
ROW6	26.000000	0.000000	0.281675	-0.281675
ROW7	27.000000	0.000000	0.196689	-0.196689
ROW8	28.000000	0.000000	0.382339	-0.382339
ROW9	29.000000	0.000000	0.245974	-0.245974
ROW10	30.000000	0.000000	0.251800	-0.251800
ROW11	31.000000	0.000000	0.304945	-0.304945
ROW12	32.000000	0.000000	0.296836	-0.296836
ROW13	33.000000	0.409091	0.432315	-0.023224
ROW14	34.000000	0.142857	0.292298	-0.149441
ROW15	35.000000	0.153409	0.204767	-0.051358
ROW16	36.000000	0.273890	0.385495	-0.111605
ROW17	37.000000	0.148571	0.309154	-0.160583
ROW18	38.000000	0.903061	0.351632	0.551430
ROW19	39.000000	0.707317	0.361164	0.346153
ROW20	40.000000	0.584906	0.271191	0.313714
ROW21	41.000000	0.573770	0.394035	0.179736
ROW22	42.000000	0.444444	0.192783	0.251662
ROW23	43.000000	0.495268	0.389293	0.105975
ROW24	44.000000	0.658654	0.313161	0.345493
ROW25	45.000000	0.529412	0.273321	0.256091
ROW26	46.000000	0.512346	0.378707	0.133639
ROW27	47.000000	0.741935	0.362298	0.379637
ROW28	48.000000	0.683495	0.441137	0.242358
ROW29	49.000000	0.694245	0.322553	0.371692
ROW30	50.000000	0.351563	0.197664	0.153899
ROW31	51.000000	0.660000	0.423777	0.236223
ROW32	52.000000	0.491228	0.300840	0.190388
ROW33	53.000000	0.746063	0.368278	0.377785
ROW34	54.000000	0.329114	0.266448	0.062666
ROW35	55.000000	0.450604	0.437701	0.012903
ROW36	56.000000	0.465257	0.223117	0.242140
ROW37	57.000000	0.316667	0.180909	0.135758
ROW38	58.000000	0.170803	0.415397	-0.244594
ROW39	59.000000	0.488764	0.249585	0.239179
ROW40	60.000000	0.145161	0.361554	-0.216393
ROW41	61.000000	0.387782	0.418735	-0.030953
ROW42	62.000000	0.741935	0.357225	0.384711
ROW43	63.000000	0.770492	0.101254	0.669238
ROW44	64.000000	0.680851	0.249482	0.431369
ROW45	65.000000	0.146067	0.338177	-0.192109
ROW46	66.000000	0.405405	0.278563	0.126842
ROW47	67.000000	0.434959	0.412414	0.022545
ROW48	68.000000	0.427184	0.456400	-0.029215
ROW49	69.000000	0.379205	0.342545	0.036660
ROW50	70.000000	0.186047	0.242908	-0.056861