ORNL/TM-10069

ornl

OAK RIDGE
NATIONAL
LABORATORY

MARTIN MARIETTA

RUMS, A PC-Based FORTRAN Program for Estimating Consumer Surplus Changes Using Multinomial Logit and Hedonic Demand Models

David L. Greene

OPERATED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

Printed in the United States of America. Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road, Springfield, Virginia 22161
NTIS price codes—Printed Copy: A05 Microfiche A01

ORNL/TM-10069

Energy Division

RUMS, A PC-BASED FORTRAN PROGRAM FOR ESTIMATING CONSUMER SURPLUS CHANGES USING MULTINOMIAL LOGIT AND HEDONIC DEMAND MODELS

David L. Greene

Date Published: August 1986

Prepared by the
OAK RIDGE NATIONAL LABORATORY
Oak Ridge, Tennessee 37831
operated by
MARTIN MARIETTA ENERGY SYSTEMS, INC.
for the
U.S. DEPARTMENT OF ENERGY
under Contract No. DE-AC05-84OR21400

CONTENTS

	Page
ACKNOWLEDGEMENTS	v
ABSTRACT	vii
1. INTRODUCTION	1
2. THEORY	3
3. IMPLEMENTATION	7
4. USER INSTRUCTIONS	17
5. REFERENCES	21
Appendix A. RUMS Program Listing	A-1
Appendix B. Sample Model Runs	B-1

ACKNOWLEDGEMENTS

The author thanks Dr. Carmen Difiglio of the Office of Policy Integration, U.S. Department of Energy for his encouragement and support. Thanks are due to the many colleagues who patiently discussed this problem with me, especially Bruce Peterson, Frank Southworth, Dan Hamblin, David Trumble, N. Scott Cardell, and Jin-Tan Liu. Thanks are also due to Stephanie Floyd, who prepared this manuscript.

ABSTRACT

This report describes a computer program for calculating the value, in dollars per consumer, of the difference in consumer satisfaction between two different sets of alternatives from which consumers must choose. Multinomial Logit and Hedonic Demand formulations of discrete choice, random utility models are used to measure the difference in consumer surplus between the two sets of alternatives, which may differ both in number and characteristics. The user must supply attribute weights for the characteristics of interest, as well as quantitative measures of the characteristics themselves. The program also has limited capability to predict changes in market shares resulting from changes in vehicle attributes. Consumer surplus change in the Hedonic Demand Model is estimated by means of Monte Carlo integration. The program is written in FORTRAN 77 for an IBM PC, with an 8087 math coprocessor chip.

RUMS, A PC-BASED FORTRAN PROGRAM FOR ESTIMATING CONSUMER SURPLUS CHANGES USING MULTINOMIAL LOGIT AND HEDONIC DEMAND MODELS

David L. Greene

1. INTRODUCTION

Random utility models of consumer choice can be used to calculate the change in consumer surplus resulting from a change in the set of choices of goods and their characteristics available to consumers. Within the context of random utility theory, precise and elegant measures of consumer surplus change can be derived. These measures are sensitive to changes in both the number of choices and the attributes of choices available.

RUMS (Random Utility Models Surplus) is a program for computing consumer surplus changes for fixed coefficient and random coefficient (Hedonic Demand) variants of the multinomial logit model. Given the characteristics of choices in a base and a new choice set, and the values of the respective attribute weights, the model computes the difference in consumer surplus between the two choice sets. The model also has limited capabilities for predicting the effects of changes in characteristics on market shares. Its prediction capability is suited to cases where only part of the typical consumer utility function is known, and where the "new" choice set is a subset of the original. The program is written in Microsoft (TM) FORTRAN 77 for MS- and PC-DOS operating systems.

2. THEORY

Random utility models of consumer choice address the consumer's decision to purchase one discrete item from an array of possible choices which satisfy the same need. The consumer is presumed to select that option which offers him the greatest utility, or satisfaction. The utility (U) of an alternative is assumed to be a stochastic function of the attributes (x_i, i=1,N) of the alternative and a set of taste parameters (a_i, i=1,N), which describe the weight attached to each attribute in determining utility.

$$U = U(x_i, a_i) \ . \tag{1}$$

Typically, U is assumed to be linear. The simplest stochastic specification of U is to assume an additive random term (e_{tj}) for each individual, t, and alternative, j.

$$U_{tj} = \sum_i a_i x_{ij} + e_{tj} \ . \tag{2}$$

If the e_{tj}'s follow the Weibull extreme value distribution, then the probability of consumer t choosing alternative j is given by the multinomial logit (MNL) model (Hensher and Johnson, 1981).

$$P_{tj} = \frac{\exp\{U_{tj}\}}{\sum_k \exp\{U_{tk}\}} \ . \tag{3}$$

The functions exp{ } and ln{ } indicate exponentiation and natural logarithms. If, in equation (2), the parameters a_i are also assumed to be randomly distributed over the population of consumers according to lognormal distributions (e_{tj}'s still distributed extreme value) then the Hedonic Demand Model (HDM) results (Beggs, 1981). The HDM is considerably more complex, since evaluating the probability of choice of j requires integrating over the probability distribution of all N attribute weights. This involves a multiple integral with order equal to the number of attributes.

The concept of consumer's surplus recognizes that when the price of a good changes, the consumer will experience a change in utility which can be translated into dollars (e.g., Varian, 1978, p. 209). Consider a consumer with a demand function for good j a function of prices of all goods, p, and income, y. By using Roy's identity x_1 can be expressed in terms of the consumer's indirect utility function U,

$$x_1(p, y) = - \frac{\partial U(p, y)}{\partial p_1} \bigg/ \frac{\partial U(p, y)}{\partial y} . \tag{4}$$

If p_1 changes, the consumer's surplus will equal the area under the demand curve between the initial, p_1^i, and final, p_1^f, prices,

$$\begin{aligned} \Delta CS &= \int_{p_1^f}^{p_1^i} x_1(p, y) dp_1 \\ &= \int_{p_1^f}^{p_1^i} - \frac{\partial U(p, y)}{\partial p_1} \bigg/ \frac{\partial U(p, y)}{\partial y} \, dp_2 . \end{aligned} \tag{5}$$

If the marginal utility of income is constant (as in the linear utility function (2) above), the integral (5) becomes

$$\Delta CS = \frac{1}{a_0} [U(p^f, y) - U(p^i, y)] , \tag{6}$$

where a_0 is the marginal utility of income.

It is intuitively clear that when the number of choices available to consumers, or the attributes of choices available are changed, the levels of satisfaction which can be reached may (generally, will) also change. In a discrete choice situation, the demand for an alternative, x_j, can be written,

$$x_j(p, y) = P_j[U^1(p, y), U^2(p, y)] \cdot 1 , \tag{7}$$

since only one unity of x_j will be consumed if x_j is chosen, and it will be chosen with probability P_j, a function of the utility levels of j and all other alternatives. This demand function can be integrated over changes in utility brought about by price changes, or any other changes which cause the utility of alternatives to change. For the MNL model, it can be shown (Williams, 1977; Small and Rosen, 1981) that the precise change in the consumer surplus (satisfaction measured in dollars) is,

$$\$CSL = \sum_t (1/a_0)[\ln\{(\sum_{j:M2} \exp\{U_{tj}(a,x')\})/\sum_{j:M1} \exp\{U_{tj}(a,x)\}\}] . \tag{8}$$

For the HDM, the corresponding consumer surplus change is given by (Cardell and Dunbar, 1980),

$$\$CSL = T \int \cdots \int (1/a_0) \ln\{.....\} f(a|m,\Sigma) da , \tag{9}$$

where M_1 and M_2 represent the base and new choice sets with x' being the new characteristics of choice set M_2. The expression ln{...} represents the same log term as in equation (8). The probability distribution of the a_i's is $f(a{:}m,\Sigma)$, m representing the vector of means of the a_i's and Σ their variance-covariance matrix (assumed here to be diagonal). Finally, a_0 is the weight of the price of the alternatives, so that $(1/a_0)$ translates the utility measure U into dollars. The integral (9) must be computed by numerical techniques, and since the number of integrals is typically greater than four, Monte Carlo methods are most appropriate.

In order to compute the change in consumer surplus, one must know the attribute values for elements of the choice sets M_1 and M_2, and the relevant attribute weight parameters. For the HDM, this includes both the mean and variance of each attribute weight. When the probability distribution of the attribute weights is assumed to be lognormal, the signs of the weights must also be specified. Given these data, RUMS can be used to calculate the change in consumer surplus.

3. IMPLEMENTATION

The calculation of consumer surplus using the multinomial logit and hedonic demand models has been implemented for microcomputers using Microsoft (TM) FORTRAN 77. Structured programming techniques were used, so that the program would be as clear as possible. The essence of the program is as follows:

1. A "setup" routine interactively prompts the user to determine the type of model (MNL or HDM), the type of operation (consumer surplus calculation or prediction), and to describe the format of the input data files. There are two input data files which are assumed to be on disk drive B: 1) PARAMS.DAT, which contains the relevant model parameters, and 2) CHARS.DAT, which contains the characteristics of each choice set. These data sets are explained in more detail below.
2. Subroutines read the parameter (PAR1, PAR2) and characteristics (CHOICE) data sets.
3. Subroutines (WEIGHTS, SHRS, SUM) compute weights if any are to be used in the calculation of consumer surplus. If the model is to be used for prediction, other subroutines (CALIB,BSUM1, BSUM2,SUM) calibrate the model to the actual market shares of the base year choice set. This step is especially important when characteristics and attribute weights are available for only a subset of the full complement of relevant characteristics.

4. If the model is MNL, subroutines are called which compute the consumer surplus change (SURPLS,ACCUM). If the operation is prediction, these are skipped and the MNL prediction subroutine (PRED1) is called.
5. If the model is HDM, calculations must be done by means of Monte Carlo integration. For consumer surplus, subroutine HDMSUR is called. HDMSUR calls RANSET, which requests the number of iterations and a seed for the subroutines which generate pairs of independent normal random numbers (RANDCO, NORMAL,RAN1) and then use these to generate a set of log-normally distributed random coefficients. Each set of coefficients is, in turn, passed to the subroutines which calculate consumer surplus change (SURPLS,ACCUM), or predict the changes in market shares (PRED2). The average change over all iterations is then computed.
6. Finally, a subroutine (SURPRN or PREPRN) is called which prints the results.

The flow chart in Figure 1 illustrates the above process. While the calculation of consumer surplus in the MNL model is essentially instantaneous, calculations for the HDM can be time consuming. The amount of time required is primarily a function of the number of characteristics (attributes) rather than the number of choices since the number of characteristics determines the number of random numbers which must be generated. For a small choice set (six choices) and six attributes, a Monte Carlo simulation of 1000 iterations requires twenty minutes of run time on an IBM-PC (TM) without the faster 8087 math coprocessor chip.

ORNL-DWG 86-13554

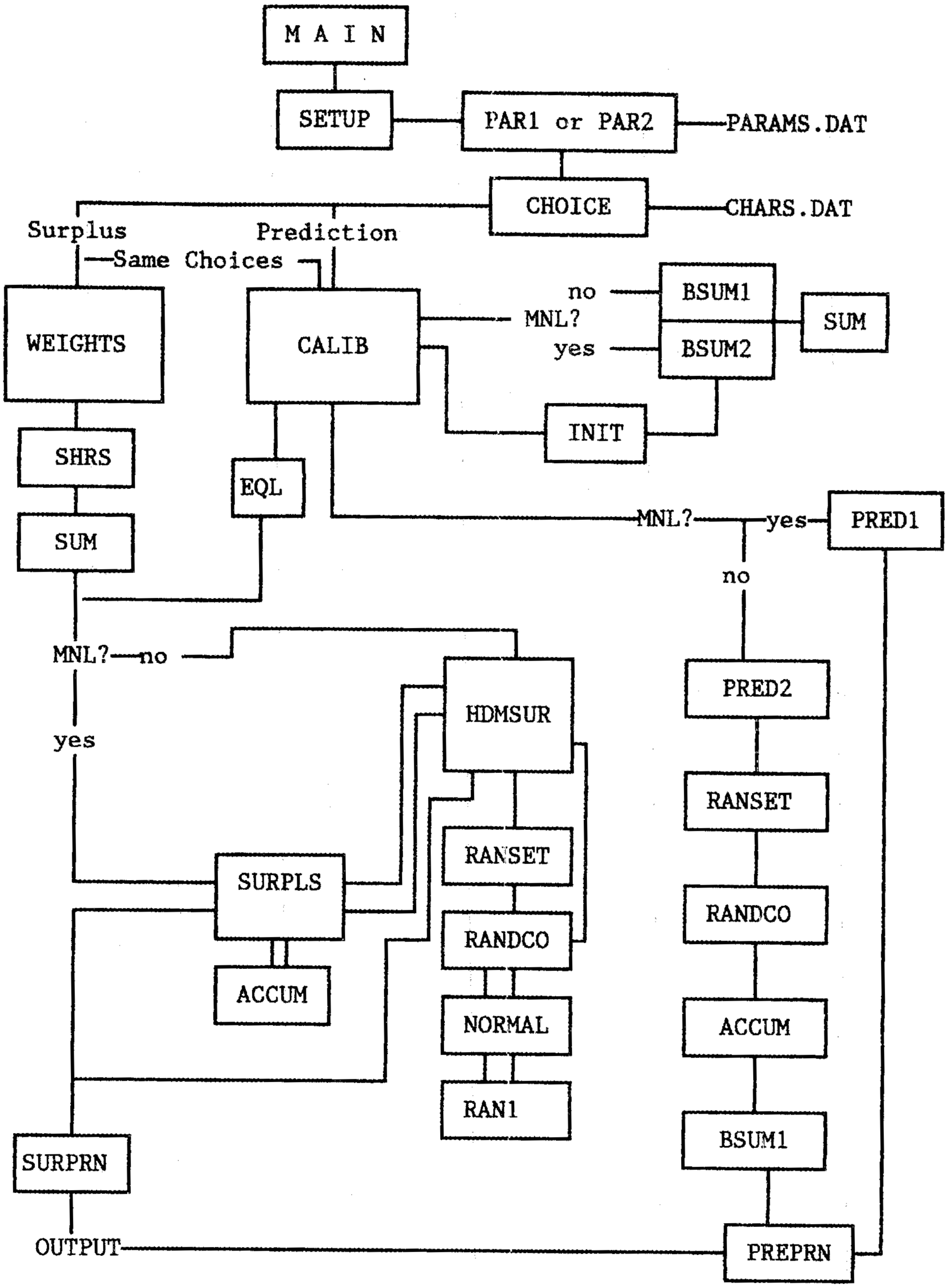

Fig. 1. Flow chart of RUMS3.

The same problem requires less than two minutes on an IBM PC equipped with an 8087 chip. Increasing the number of attributes would increase the run time roughly proportionately.

Three choices are available for computing consumer surplus using either the MNL or HDM models: 1) unweighted, 2) market share weighted, or 3) with calibration constants for the special case where the characteristics of alternative choices change but the membership of choice sets 1 and 2 is the same. If the utility function described by the available characteristics is considered to be complete, that is, either no attributes are missing or those that are missing are uncorrelated with those present, then unweighted calculation is appropriate. In many cases, however, a large number of relevant attributes may be missing. If this is believed to be true then it may be preferable to compute a market shares weighted consumer surplus. In the special case that exactly the same alternative choices are available in both sets, but with somewhat changed characteristics, a calibration method can be used to complete the utility value for each alternative.

The method of market shares weighting is novel and, thus, needs some explanation. The linear utility function can be partitioned into unknown (A) and known (B) parts,

$$U_j = A_j + B_j \ . \tag{10}$$

The unknown part, A, represents, in effect, a weight for the known part in the logsum terms of equations (4) and (5).

$$\ln\{\sum_j W_j \exp\{B_j\}\} \ , \ W_j = \exp\{A_j\} \ . \tag{11}$$

If A_j is very large relative to B_j, then the market share of alternative j, P_j, can be used as an approximation to W_j.

It can be shown that the estimated change in consumer surplus obtained by using P_j as a weight is the consumer surplus change with B_j multiplied by 2 minus the actual consumer surplus change. By inserting the definition of p_j from (3) into (11), and using the partitioning of the utility function in (10), one obtains,

$$\begin{aligned} \ln \sum_{j:M} p_j e^{B_j} &= \ln \sum_{j:M} \left(e^{U_j+B_j} / \sum_{i:M} e^{U_i}\right) \\ &= \ln\left[\left(\sum_{i:M} e^{U_i}\right)^{-1} \left(\sum_{j:M} e^{A_j+2B_j}\right)\right] . \end{aligned} \tag{12}$$

This, of course, is half of the formula for consumer's surplus. The full expression is,

$$\begin{aligned} \Delta CS &= \frac{1}{a_0} \ln\left[\left(\sum_{i:M2} e^{U_i}\right)^{-1} \left(\sum_{j:M2} e^{A_j+2B_j}\right) / \right. \\ &\qquad \left. \left(\sum_{k:M1} e^{U_k}\right)^{-1} \left(\sum_{\ell:M1} e^{A_\ell+2B_\ell}\right)\right] \\ &= (1/a_0)[\ln\{\sum_{j:M2} \exp\{A_j + 2B_j\} / \sum_{j:M1} \exp\{A_j + 2B_j\}\} - \\ &\qquad \ln\{\sum_{j:M2} \exp\{A_j + B_j\} / \sum_{j:M1} \exp\{A_j + B_j\}\}] . \end{aligned} \tag{13}$$

The second line of equation (13) is the total consumer surplus change as shown in equation (8). The first part is also a kind of total consumer surplus change but with the values of the observed variables doubled.

The difference is an approximation to the consumer surplus change due to changes in observed variables alone.

A heuristic interpretation of (13) can be based on equation (6) which states that consumer's surplus is equal to a change in utility levels translated into dollars. Using the partitioning of the utility function into known and unknown parts, together with the fact that the utility functions are linear, we can see that in equation (13) we are subtracting a ΔA plus ΔB (the actual change in utility) from a ΔA plus 2Δ B. This would give exactly ΔB were it not for the fact that probabilities of choosing alternatives changes (in a nonlinear way) as their utility levels change. As a result, (13) is an approximation only, and likely to be good only for small changes in utility levels.

If B_j is very large relative to A_j, then B_j approaches the total utility function and no weights are needed. If A_j is very large relative to B_j, then shares-weighting should be better. Since neither of these two conditions is likely to be exactly true, it will usually be wise to carry out the estimation with and without weighting in order to bound the inaccuracy caused by an incomplete representation of the utility function.

In the special case that there are no changes in the membership of choice sets 1 and 2 and (only the observed characteristics of the alternatives change, a more precise method can be used. This third method calculates constants A_j such that, given the observed characteristics and attribute weights of choice set 1, the MNL (or HDM) model will predict the actual market shares exactly. The method of calculating the A_j is the same as that used when RUMS is used for shares prediction and is described below. The A_j from choice set 1 are added to the B_j for choice sets 1 and 2 in computing consumer's surplus.

The market shares weighting method treats the missing part of the utility function as nonstochastic. This is actually not consistent with the HDM, where the entire utility function is stochastic. Unfortunately, there are insufficient degrees of freedom to estimate both the weights and their variances. Thus, when the HDM is used and the variance of the missing portion of the utility function is believed to be important, one must interpret the computed consumer surplus changes with caution.

The inclusion of weighting factors in the model accomplishes two things: 1) it weights the changes in utility for individual choices according to their market share, so that changes in very popular models will count more than changes in rare ones, and when market shares weighting is used, 2) it allows for changes in other, unmeasured attributes from one time period to the next. These refinements should increase the accuracy of the estimated changes in consumer surplus.

When RUMS is used for prediction, it must be calibrated so that it accurately predicts the base year shares. Calibration of the model consists of computing constants for each member of the choice sets such that the multinomial logit or hedonic demand models predict exactly the observed market shares for each alternative. Again, consider the representative utility function as being composed of two parts: 1) the utility derived from the attributes for which we have measures, and 2) the utility derived from all other factors. The linear utility function of equation (2) can be partitioned into the calibration constant, A_j, and the remaining sum of attributes times attribute weights:

$$U_j = A_j + \sum_i a_i x_{ij} \ . \qquad (14)$$

Equation (14) can be substituted in equation (3), and the probability of individual t chooosing j replaced by the observed market share of j. This gives a set of M (the number of alternatives) equations in M unknowns (the A_j). However, the sum of shares must equal 1, so that one of the A_j must be chosen arbitrarily to satisfy this constraint.

A particulary convenient estimating equation can be derived by first taking the ratio of the shares of alternatives j and k:

$$P_j/P_k = (\exp\{A_j\}\exp\{B_j\})/(\exp\{A_k\}\exp\{B_k\}) \ , \qquad (15)$$

where B_k represents the summation term of equation (6) for alternative k. Taking logarithms and rearranging gives:

$$A_j = \ln\{P_j\} - \ln\{P_k\} + (B_k - B_j) + A_k \ . \qquad (16)$$

Now if we express each share relative to alternative 1 and choose $A_1 = -B_1$, then the remaining A_j's can be computed using:

$$A_j = \ln\{P_j\} - \ln\{P_1\} - B_j \ . \qquad (17)$$

It can be easily shown that A's computed in this way will yield shares which sum to one for any B's.

Calibration of the Hedonic Demand Model differs slightly from the Multinomial Logit in that there is no single value for an attribute weight. Instead, the mean is used. Since the parameters are lognormally distributed, the means are,

$$\exp\{m_i + 0.5s_i^2\} \text{ ,} \tag{18}$$

where m_i and s_i are parameters of the lognormal distribution. Clearly, there are insufficient degrees of freedom to compute both A's and variances for the A's. Thus, the A's are constants which produce constant weighting factors for each alternative. This is not an exact representation of the HDM, since it allows random coefficients only for those attributes which are explicitly included. Still, it appears to be the best that can be done, and is undoubtedly superior to ignoring the calibration constants altogether.

Computation of consumer surplus is accomplished according to the formulas described in Section 2. Generation of normally distributed random numbers is accomplished in two steps. First, uniform random numbers on the interval [0,1] are generated using an algorithm proposed by Monro (1982, pp. 52-53). Next, two successive, independent uniform random numbers (R_n, R_{n+1}) are transformed to produce a pair of independent normal (0,1) random numbers (S_n, S_{n+1}) using a method proposed by Wilf (1967, p. 255). Wilf's tranformation equations are as follows:

$$\begin{aligned} S_n &= (-2\ln\{R_n\})^{1/2}\cos\{2(\pi)R_{n+1}\} \\ S_{n1} &= (-2\ln\{R_n\})^{1/2}\sin\{2(\pi)R_{n+1}\} \text{ .} \end{aligned} \tag{19}$$

These normal random variates are used to produce lognormally distributed attribute weights using the given lognormal parameters, m and s, and the transformation,

$$a_i = \text{sgn}(i)\exp\{m_i+S_n s_i\} \text{ .} \tag{20}$$

In (20), the term sgn(i) represents + or - 1 depending on the sign of the attribute (i.e., whether more of it is good or bad). When the parameters are assumed to be distributed lognormally, the sign of each must be specified a priori. For the HDM, then, the parameter data set (PARAMS.DAT) contains three values for each attribute: the lognormal parameters m and s, and the sign of the attribute weight (+/-1).

4. USER INSTRUCTIONS

This section describes how to set up the model on an IBM (TM) PC with two disk drives. The program will run without an 8087 coprocessor chip, but this will slow the execution time greatly. This is followed by a sample run including sample data sets of parameters and attributes based on an automobile choice model by Boyd and Mellman (1980). Finally, common problems which may be encountered are discussed.

The first step is to start the computer and install the disk operating system (DOS). The program RUMS.EXE should be available on diskette or hard disk. It does not matter in which drive this diskette is placed, however, because the program will prompt the user for the names of required data files. The user must supply a complete pathname for each data file. If the user has a hard disk or can set up a ramdisk, it may be worthwhile to read data from disk c: to increase the program's speed. If the number of alternatives is large (say 100 or more) this can significantly reduce execution time.

Two separate data files are required:

1. The first file must contain the necessary parameter values (in the examples this is called PARAMS.DAT). For the MNL model, the program will read the i attribute weights, in order, using an implied DO loop. <u>The first parameter must always be the attribute weight for the price of the good</u>. For the HDM, the program consecutively reads the lognormal parameters m (the logarithm of the median), s (the standard deviation of m), and sign, before proceeding to the next attribute weight.

2. the file containing the attributes of all elements in both the base and new choice sets (called CHARS.DAT in the sample rums). Each alternative constitutes one record. The order of attributes in this file must be the same as that in PARAMS.DAT. The first variable in each record must be the alternative's price, the last variable must be either the total sales or the market share of that alternative. All base choice set elements and attributes must appear first, followed by the attributes of the new choice set. The model will assume that the first M1 records are the base alternatives and the following M2 records are the new alternatives attributes.

The user will be prompted to specify the following information:

1. type of model (MNL or HDM),
2. type of operation (Consumer's Surplus or Prediction),
3. numbers of elements in the base and new choice sets,
4. number of characteristics (including price but not sales or shares),
5. record formats (FORTRAN format statements) for the parameter and characteristics data files (both may be printed, if desired), and
6. the type of weighting to be used if consumer surplus change is to be computed.

The sample runs in Appendix A illustrate these prompts. The user responses are in lower case letters. The user must hit the carriage return (or ENTER key) following each response. The parameters and data used in the samples have been borrowed from Boyd and Mellman (1980).

The program may fail to operate as expected for several reasons. The most common will probably be that the attribute weight parameters specified are unrealistic. The result is most often a real math overflow error, although an underflow can also result. This is most likely to be a problem with the HDM model, where the Monte Carlo simulation may produce values too large to exponentiate (the current version of the program uses double precision real numbers for such calculations). The values of m may be too large, or it may be that the variance of m, s, is too large and occasionally produces a very large number when a large random number is generated. Similar errors can also occur when attribute data are unrealistic (e.g., cause the utility function to take on values as large as 80 or higher). Such problems can usually only be solved by correcting errors in or changing the values of the attribute weight parameters or characteristics data, although simply changing the random number seed may provide a quick fix on rare occasions.

Since these kinds of errors are indicative of problems with the model parameters being used (because they suggest that unreasonably large, or small, values are being placed on some attributes), it is generally unwise to ignore them. When overflow errors are encountered, the user should carefully examine the model parameters and data being used.

5. REFERENCES

1. Beggs, S. D., 1981. "The demand for electric automobiles," EA-2072, Research Project 1145-1, for the Electric Power Research Institute, Palo Alto, California, October 1981.
2. Boyd, J. H. and R. E. Mellman, 1980. "The effect of fuel economy standards on the U.S. automotive market: an hedonic demand analysis," Transportation Research, vol. 14A, no. 5-6, pp. 367-378, October-December 1980.
3. Cardell, N. S. and F. C. Dunbar, 1980. "Measuring the societal impacts of downsizing," Transportation Research, vol. 14A, no. 5-6, pp. 423-434, October-December 1980.
4. Hensher, D. A. and L. W. Johnson, 1981. Applied Discrete Choice Modeling, John Wiley, New York.
5. Microsoft Corporation, 1983. Microsoft FORTRAN, Reference Manual and User's Guide, Bellevue, Washington.
6. Monro, D. M., 1982. FORTRAN 77, Edward Arnold, London.
7. Moshman, J., 1967. "Random number generation," in Wilf, H.S., Mathematical Methods for Digital Computers, John Wiley & Sons, Inc., New York.
8. Small, K. A. and H. S. Rosen, 1981. "Applied Welfare Economics with Discrete Choice Models," Econometrica, vol. 49, no. 1, pp. 105-130.
9. Varian, H. R., 1978. Microeconomic Analysis, W. W. Norton and Co., N.Y.

10. Williams, H.C.W.L, 1977. "On the formation of travel demand models and economic evaluation measures of user benefit," Environment and Planning A, vol. 9, pp. 285-344.

Appendix A

RUMS Program Listing

Appendix A

RUMS Program Listing


```
D Line# 1     7                         Microsoft FORTRAN77 V3.31 August 1985
C**************************************************************
C
C
C RUMS  IS A PROGRAM FOR CALCULATING CONSUMER SURPLUS CHANGE
C AND PREDICTING MARKET SHARE CHANGES USING MULTINOMIAL LOGIT
C AND HEDONIC DEMAND MODELS.
C  CREATED BY D L GREENE ORNL 6-09-1986
C
C
C******************************************************************
C
C
C
      PROGRAM RUMS
      REAL*8 B(15),SAL1(200),SAL2(200),A1(200),A2(200),BLN(15),
     1 RAN(16),CONSUR,E(15),SGN(15),X1(200,10),X2(200,10)
      CHARACTER*2 IDSK
C  OPEN DATA FILES FOR PARAMETERS AND CHARACTERISTICS
C
C
      WRITE (*,5)
5     FORMAT(1H0,'PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS')
      WRITE (*,6)
6     FORMAT(1H )
      OPEN(11,FILE=' ')
      WRITE (*,6)
      WRITE (*,7)
7     FORMAT(1H0,'PLEASE ENTER FILENAME FOR CHARACTERISTICS')
      OPEN(12,FILE=' ')
C
C
C  DEFAULTS STATUS=OLD ACCESS=SEQUENTIAL FORM=FORMATTED
C
C  PROGRAM SETUP SUBROUTINE MNL OR HDM
      CALL SETUP (MODEL,M1,M2,N,IPRE)
C
C  READ PARAMETERS
      IF (MODEL .EQ. 1) CALL PAR1 (N,B)
      IF (MODEL .EQ. 2) CALL PAR2 (N,B,E,SGN)
C
C  READ CHOICE SET CHARACTERISTICS
      CALL CHOICE (M1,M2,N,X1,X2,SAL1,SAL2)
C
C  CALIBRATE SHARE WEIGHTS OR UTILITY FUNCTION CONSTANTS
      IF (IPRE .EQ. 1) CALL WEIGHTS (M1,M2,SAL1,SAL2,A1,A2)
      IF (IPRE .EQ. 2) CALL CALIB (M1,N,B,E,X1,X2,SAL1,SAL2,A1,SGN,
     1 MODEL,M2,IPRE)
      IF (IPRE .EQ. 3) CALL CALIB (M1,N,B,E,X1,X2,SAL1,SAL2,A1,SGN,
     1 MODEL,M2,IPRE)
      IF (IPRE .EQ. 3) CALL EQL (M1,A1,A2)
      IF (IPRE .EQ. 2) GOTO 99
C
C  CALCULATE CONSUMER SURPLUS CHANGE
      IF (MODEL .EQ. 1) CALL SURPLS (M1,M2,N,B,X1,X2,CONSUR,A1,A2)
      IF (MODEL .EQ. 2) CALL HDMSUR (M1,M2,N,B,E,X1,X2,CONSUR,SGN,A1,A2)
99    CONTINUE
```

```
C
C OUTPUT RESULTS
        IF (IPRE .EQ. 1) CALL SURPRN (M1,M2,N,MODEL,CONSUR)
        IF (IPRE .EQ. 2) CALL PREPRN (MODEL,M1,SAL1,SAL2,M2)
        IF (IPRE .EQ. 3) CALL SURPRN (M1,M2,N,MODEL,CONSUR)
C
        END
```

Name	Type	Offset	P	Class
A1	REAL*8	35824		
A2	REAL*8	37424		
B	REAL*8	16		
BLN	REAL*8	35456		
CONSUR	REAL*8	39154		
E	REAL*8	35704		
IDSK	CHAR*2	*****		
IPRE	INTEGER*4	39150		
M1	INTEGER*4	39138		
M2	INTEGER*4	39142		
MODEL	INTEGER*4	39134		
N	INTEGER*4	39146		
RAN	REAL*8	35576		
SAL1	REAL*8	136		
SAL2	REAL*8	1736		
SGN	REAL*8	3336		
X1	REAL*8	3456		
X2	REAL*8	19456		

```
C
C
C*****************************************************************
C
C         SUBROUTINE  SETUP (MODEL, M1, M2, N, IPRE)
C
C         > OBTAIN PROBLEM DESCRIPTION FROM USER
C
C         M1: NUMBER OF OBSERVATIONS IN FIRST CHOICE SET
C         M2: NUMBER OF OBSERVATIONS IN SECOND CHOICE SET
C         N:  NUMBER OF ATTRIBUTES
C         IPRE: INDICATOR FOR TYPE OF CALCULATION
C         MODEL: INDICATOR FOR TYPE OF MODEL
C
C*****************************************************************
C
          SUBROUTINE SETUP (MODEL,M1,M2,N,IPRE)
C
C  READ TYPE OF MODEL FROM KEYBOARD
        WRITE (*,10)
10      FORMAT(1H )
          WRITE (*,12)
12        FORMAT(1H0,'ESTIMATION OF CONSUMER SURPLUS IN RUMS')
          WRITE (*,14)
14        FORMAT(1H0,'WHAT TYPE OF MODEL?  ENTER:')
          WRITE (*,16)
```

```
16       FORMAT(1H0,'     1 FOR MULTINOMIAL LOGIT')
         WRITE (*,18)
18       FORMAT(1H0,'     2 FOR HEDONIC DEMAND MODEL')
         READ (*,52) MODEL
52       FORMAT (I1)
         WRITE (*,20) MODEL
20       FORMAT(1H0,'YOU CHOSE MODEL ',I1,'.')
        WRITE (*,100)
100     FORMAT(1H0,'WHAT TYPE OF ESTIMATION?   ENTER:')
        WRITE (*,110)
110     FORMAT(1H0,'     1 FOR CONSUMER SURPLUS CHANGE')
        WRITE (*,120)
120     FORMAT(1H0,'     2 FOR SHARES PREDICTION')
        WRITE (*,130)
130     FORMAT(1H0,'     3 FOR C S CHANGE, CORRESPONDING SET MEMBERS')
        READ (*,52) IPRE
        IF (IPRE .EQ. 3) WRITE (*,140)
140     FORMAT(1H0,'FOR OPTION 3  SETS NUMBER AND ORDER MUST AGREE')
C
C  READ NUMBER OF CHOICES AND NUMBER OF CHARACTERISTICS
         WRITE (*,22)
22       FORMAT(1H0,'ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -')
         READ (*,54) M1
54       FORMAT(I3)
         WRITE (*,23)
23       FORMAT(1H0,'ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -')
         READ (*,54) M2
         WRITE (*,24)
24       FORMAT(1H0,'ENTER NUMBER OF CHARACTERISTICS -')
         READ (*,56) N
56       FORMAT(I2)
         WRITE (*,26) M1,M2,N
26       FORMAT(1H0,'THERE ARE ',I3,' & ',I3,' CHOICES & ',I2,' CHARS.')
        IF ((IPRE .EQ. 3) .AND. (M1 .NE. M2)) WRITE (*,200)
200     FORMAT(1H0,'ERROR   M1 NE M2: CHOICE SETS DO NOT CORRESPOND')
C
         END
```

Name	Type	Offset	P	Class
IPRE	INTEGER*4	16	*	
M1	INTEGER*4	4	*	
M2	INTEGER*4	8	*	
MODEL	INTEGER*4	0	*	
N	INTEGER*4	12	*	

```
C
C
C*******************************************************************
C
C        SUBROUTINE PAR1 (N, B)
C
C       > READ MNL PARAMETERS
C
C       B: ATTRIBUTE COEFFICIENTS
```

```
C      N: NUMBER OF ATTRIBUTES
C
C*******************************************************************
             SUBROUTINE PAR1 (N,B)
C  READ MNL MODEL PARAMETERS
             REAL*8 B(N)
             CHARACTER*25 AFMT
             WRITE (*,100)
 100         FORMAT(1H0,' READING MNL MODEL PARAMETERS . . .')
             WRITE (*,99)
 99          FORMAT(1H0,'ENTER RECORD FORMAT FOR PARAMS.DAT.')
             READ (*,9) AFMT
 9           FORMAT(A25)
             READ (11,AFMT) (B(I),I=1,N)
           WRITE (*,105)
 105       FORMAT(1H0,'ENTER 1 TO PRINT PARAMETERS')
           READ (*,107) IPRN
 107       FORMAT(I1)
           IF (IPRN .EQ. 1) WRITE (*,111) (B(I),I=1,N)
 111       FORMAT(1X,E15.9)
            RETURN
            END
```

Name	Type	Offset	P	Class
AFMT	CHAR*25	39964		
B	REAL*8	4	*	
I	INTEGER*4	39996		
IPRN	INTEGER*4	40040		
N	INTEGER*4	0	*	

```
C
C
C*******************************************************************
C
C          SUBROUTINE PAR2 (N, B, E, SGN)
C
C        > READ HDM PARAMETERS
C
C             B: ATTRIBUTE COEFFICIENT
C             E: STANDARD DEVIATION
C             SGN: SIGN OF THE PARAMETER VALUE
C
C*******************************************************************
C
           SUBROUTINE PAR2 (N,B,E,SGN)
           REAL*8 B(N),E(N),SGN(N)
           CHARACTER*25 AFMT
C  READ HDM PARAMETERS
           WRITE (*,100)
 100       FORMAT(1H0,' READING HDM MODEL PARAMETERS . .  .')
           WRITE (*,99)
 99        FORMAT(1H0,'ENTER RECORD FORMAT FOR PARAMS.DAT.')
           READ (*,9) AFMT
 9         FORMAT(A25)
```

```
D Line# 1      7                               Microsoft FORTRAN77 V3.31 August 1985
                READ (11,AFMT) (B(I),E(I),SGN(I),I=1,N)
               WRITE (*,110)
         110   FORMAT(1H0,'ENTER 1 TO PRINT PARAMETERS')
               READ (*,112) IPRN
         112   FORMAT(I1)
               IF (IPRN .EQ. 1) WRITE (*,114) (B(I),E(I),SGN(I),I=1,N)
         114   FORMAT(1X,3E15.9)
                RETURN
                END

Name      Type          Offset P Class

AFMT     CHAR*25         40148
B        REAL*8              4 *
E        REAL*8              8 *
I        INTEGER*4       40180
IPRN     INTEGER*4       40224
N        INTEGER*4           0 *
SGN      REAL*8             12 *


         C
         C***************************************************************
         C
         C         SUBROUTINE CHOICE (M1, M2, N, X1, X2, SAL1, SAL2)
         C
         C         > READ CHARACTERISTICS FOR BOTH CHOICE SETS
         C
         C         M1: NUMBERS IN BASE CHOICE SET
         C         M2: NUMBERS IN NEW CHOICE SET
         C         X1: CAR CHARACTERISTICS IN BASE YEAR
         C         X2: CAR CHARACTERISTICS IN NEW YEAR
         C         SAL1: SALES IN BASE YEAR
         C         SAL2: SALES IN NEW YEAR
         C         IPRN: INDICATOR FOR PRINTING CHARACTERISTICS DATA
         C
         C*****************************************************************
         C
         C
               SUBROUTINE CHOICE (M1,M2,N,X1,X2,SAL1,SAL2)
         C  READ CHOICE CHARACTERISTICS, NEW AND BASE
               CHARACTER*25 AFMT,BFMT
               REAL*8 SAL1(M1),SAL2(M2),X1(M1,N),X2(M2,N)
         C  READ BASE RECORDS, FIRST M1
               WRITE (*,200)
         200   FORMAT(1H0,'READING CHARACTERISTICS & SALES DATA')
               WRITE (*,99)
         99    FORMAT(1H0,'ENTER RECORD FORMAT FOR CHARS.DAT.')
               READ (*,9) AFMT
         9     FORMAT(A25)
               WRITE (*,105)
         105   FORMAT(1H0,'ENTER 1 TO PRINT CHARACTERISTICS DATA')
               READ (*,107) IPRN
         107   FORMAT(I1)
               IF (IPRN .EQ. 1) WRITE (*,109)
         109   FORMAT(1H0,'ENTER OUTPUT FORMAT, E.G. 1X,INPUT FORMAT')
```

```
D Line# 1       7                                    Microsoft FORTRAN77 V3.31 August 1985
                IF (IPRN .EQ. 1) READ (*,9) BFMT
                DO 20 I=1,M1
1                 READ (12,AFMT)   (X1(I,J),J=1,N),SAL1(I)
1                 IF (IPRN .EQ. 1) WRITE(*,BFMT) (X1(I,J),J=1,N),SAL1(I)
1         20    CONTINUE
          C
          C  READ NEW RECORDS M1+1 TO M1+M2
                DO 40 I=1,M2
1                 READ (12,AFMT)   (X2(I,J),J=1,N),SAL2(I)
1                 IF (IPRN .EQ. 1) WRITE (*,BFMT) (X2(I,J),J=1,N),SAL2(I)
1         40    CONTINUE
          C
                END
```

Name	Type	Offset	P	Class
AFMT	CHAR*25	40331		
BFMT	CHAR*25	40466		
I	INTEGER*4	40492		
IPRN	INTEGER*4	40408		
J	INTEGER*4	40500		
M1	INTEGER*4	0	*	
M2	INTEGER*4	4	*	
N	INTEGER*4	8	*	
SAL1	REAL*8	20	*	
SAL2	REAL*8	24	*	
X1	REAL*8	12	*	
X2	REAL*8	16	*	

```
      C
      C
      C
      C  CALIBRATION SUBROUTINES
      C
      C*****************************************************************
      C
      C        SUBROUTINE  WEIGHTS (M1, M2, SAL1, SAL2, A1, A2)
      C
      C        > COMPUTE SALES WEIGHTS FOR CONSUMER SURPLUS CALCULATION
      C          WHEN SETS 1 AND 2 DO NOT HAVE CORRESPONDING MEMBERSHIPS
      C
      C        A1: UTILITY FUNCTION INTERCEPTS FOR CHOICE SET 1
      C        A2: UTILITY FUNCTION INTERCEPTS FOR CHOICE SET 2
      C
      C          CALL  SUBROUTINE SHRS (M1, SAL1, A1)
      C          CALL  SUBROUTINE SHRS (M1, SAL2, A2)
      C
      C*****************************************************************
      C
            SUBROUTINE WEIGHTS (M1,M2,SAL1,SAL2,A1,A2)
            REAL*8 SAL1(M1),SAL2(M2),A1(M1),A2(M2)
      C
            WRITE (*,100)
        100 FORMAT(1H0,'WHAT TYPE OF WEIGHTING FOR LOGSUM TERMS?  ENTER:')
            WRITE (*,110)
```

```
D Line# 1      7                        Microsoft FORTRAN77 V3.31 August 1985
        110    FORMAT(1H0,'     1 FOR SHARES')
               WRITE (*,120)
        120    FORMAT(1H0,'     2 FOR NO WEIGHTS')
               READ (*,200) IWT
        200    FORMAT(I1)
        C
               DO 10 I=1,M1
1               A1(I)=0.0
1       10     CONTINUE
               DO 20 I=1,M2
1               A2(I)=0.0
1       20     CONTINUE
        C
               IF (IWT .EQ. 1) CALL SHRS (M1,SAL1,A1)
               IF (IWT .EQ. 1) CALL SHRS (M2,SAL2,A2)
        C
               END
```

Name	Type	Offset	P	Class
A1	REAL*8	16	*	
A2	REAL*8	20	*	
I	INTEGER*4	40684		
IWT	INTEGER*4	40620		
M1	INTEGER*4	0	*	
M2	INTEGER*4	4	*	
SAL1	REAL*8	8	*	
SAL2	REAL*8	12	*	

```
C
C
C*****************************************************************
C
C      SUBROUTINE CALIB (M1,N,B,E,X1,X2,SAL1,SAL2,A1,SGN,MODEL,M2,IPRE)
C
C      > COMPUTE CONSTANT TERMS OF UTILITY FUNCTION WHEN:
C             1. PREDICTING SHARES
C             2. CHOICE SETS DON'T HAVE CORRESPONDING MEMBERSHIPS
C
C         CALL  SHRS (M1, SAL1, A1)
C         CALL  BSUM1(M1, X1, B, N, BS1)
C         CALL  BSUM1(M2, X2, B, N, BS2)
C         CALL  BSUM2(M1, X1, B, E, N, BS1, SGN)
C         CALL  PRED1(M2, A1, BS2, SAL2)
C         CALL  PRED2(M2, A1, N, B, E, SGN, X2, BS2, SAL2)
C
C*****************************************************************
C
C
       SUBROUTINE CALIB (M1,N,B,E,X1,X2,SAL1,SAL2,A1,SGN,MODEL,M2,IPRE)
C  CALIBRATE UTILITY CONSTANTS FOR MNL AND HDM MODELS
C
       REAL*8 SAL1(M1),SAL2(M2),BS1(200),BS2(200),A1(M1),B(N),
      1 E(N),SGN(N),X1(M1,N),X2(M2,N)
C
```

```
      CALL SHRS (M1,SAL1,A1)
C
      IF (MODEL .EQ. 1) CALL BSUM1 (M1,X1,B,N,BS1)
      IF (MODEL .EQ. 1) CALL BSUM1 (M2,X2,B,N,BS2)
C
      IF (MODEL .EQ. 2) CALL BSUM2 (M1,X1,B,E,N,BS1,SGN)
C
      CALL INIT (M1,BS1,SAL1,A1)
C
      IF (IPRE .EQ. 3) GOTO 99
      IF (MODEL .EQ. 1) CALL PRED1    (M2,A1,BS2,SAL2)
      IF (MODEL .EQ. 2) CALL PRED2 (M2,A1,N,B,E,SGN,X2,BS2,SAL2)
99    CONTINUE
C
      END
```

Name	Type	Offset	P	Class
A1	REAL*8	32	*	
B	REAL*8	8	*	
BS1	REAL*8	40696		
BS2	REAL*8	42296		
E	REAL*8	12	*	
IPRE	INTEGER*4	48	*	
M1	INTEGER*4	0	*	
M2	INTEGER*4	44	*	
MODEL	INTEGER*4	40	*	
N	INTEGER*4	4	*	
SAL1	REAL*8	24	*	
SAL2	REAL*8	28	*	
SGN	REAL*8	36	*	
X1	REAL*8	16	*	
X2	REAL*8	20	*	

```
C
C*******************************************************************
C
C        SUBROUTINE SHRS (M, SHR, A)
C
C       > COMPUTE SALES SHARES FROM SALES DATA
C
C       SHR: SALES SHARES (FRACTION OF TOTAL)
C
C           CALL SUM (M, S, SHR)
C
C********************************************************************
C
      SUBROUTINE SHRS (M,SHR,A)
C  RETURNS SHARES IN PLACE OF SALES
C
      REAL*8 S,SHR(M),A(M)
C
      CALL SUM (M,S,SHR)
C
C
```

```
D Line# 1       7
                 DO 10 I=1,M
1                  SHR(I)=SHR(I)/S
1                 A(I)=DLOG(SHR(I))
1         10     CONTINUE
          C
                END
```

Name	Type	Offset	P	Class
A	REAL*8	8	*	
DLOG				INTRINSIC
I	INTEGER*4	43904		
M	INTEGER*4	0	*	
S	REAL*8	43896		
SHR	REAL*8	4	*	

```
          C
          C
          C*******************************************************************
          C
          C       SUBROUTINE SUM (NN, SS, XX)
          C
          C       > COMPUTE THE SUM OF AN ARRAY
          C
          C       NN: SIZE OF ARRAY
          C       XX: ARRAY TO BE SUMMED
          C       SS: SUM
          C
          C*******************************************************************
          C
                  SUBROUTINE SUM (NN,SS,XX)
          C
                  REAL*8 SS,XX(NN)
          C
          C  INITIALIZE
                  SS=0
                   DO 10 I=1,NN
1                    SS=SS+XX(I)
1         10       CONTINUE
          C
                  END
```

Name	Type	Offset	P	Class
I	INTEGER*4	43912		
NN	INTEGER*4	0	*	
SS	REAL*8	4	*	
XX	REAL*8	8	*	

```
          C
          C
          C*******************************************************************
          C
          C   SUBROUTINE  BSUM1 (M, X, B, N, BS)
```

```
D Line# 1      7
      C
      C     > COMPUTE UTILITIES OF ALTERNATIVES FOR MNL MODEL
      C
      C     BS: UTILITY VALUE = SUM OF COEFFS*ATTRIBUTES
      C
      C*****************************************************************
      C
            SUBROUTINE BSUM1 (M,X,B,N,BS)
      C
            REAL*8 BS(M),B(N),X(M,N)
      C
            DO 10 I=1,M
1             BS(I)=0
1            DO 20 J=1,N
2              BS(I)=BS(I)+(B(J)*X(I,J))
2       20   CONTINUE
1       10  CONTINUE
      C
            END
```

Name	Type	Offset	P	Class
B	REAL*8	8	*	
BS	REAL*8	16	*	
I	INTEGER*4	43920		
J	INTEGER*4	43928		
M	INTEGER*4	0	*	
N	INTEGER*4	12	*	
X	REAL*8	4	*	

```
      C
      C
      C*****************************************************************
      C
      C     SUBROUTINE BSUM2 (M, X, B, E, N, BS, SGN)
      C
      C     > COMPUTE UTILITIES OF ALTERNATIVES FOR HDM MODEL
      C
      C*****************************************************************
      C
            SUBROUTINE BSUM2 (M,X,B,E,N,BS,SGN)
      C
            REAL*8 BS(M),B(N),E(N),SGN(N),X(M,N)
      C
            DO 10 I=1,M
1             BS(I)=0
1            DO 20 J=1,N
2              BS(I)=BS(I)+SGN(J)*DEXP(B(J)+0.5*E(J)**2)*X(I,J)
2       20   CONTINUE
1       10  CONTINUE
      C
            END
```

Name	Type	Offset	P	Class

```
D Line# 1      7
B       REAL*8              8 *
BS      REAL*8             20 *
DEXP                            INTRINSIC
E       REAL*8             12 *
I       INTEGER*4       43936
J       INTEGER*4       43944
M       INTEGER*4           0 *
N       INTEGER*4          16 *
SGN     REAL*8             24 *
X       REAL*8              4 *
```

```
      C
      C****************************************************************
      C
      C          SUBROUTINE  INIT (B, M, SAL, A)
      C
      C        > COMPUTE INTERCEPTS OF UTILITY FUNCTIONS USING KNOWN SHARES
      C
      C****************************************************************
      C
            SUBROUTINE INIT (M,B,SAL,A)
      C
            REAL*8 A(M),SAL(M),B(M)
      C
              A(1)=-B(1)
            DO 10 I=2,M
1           IF ((SAL(I) .LE. 0) .OR. (SAL(1) .LE. 0)) CALL SERR (M,B,SAL,I)
1             A(I)=DLOG(SAL(I))-DLOG(SAL(1))-B(I)
1        10 CONTINUE
      C
            END
```

```
Name     Type         Offset P Class

A       REAL*8             12 *
B       REAL*8              4 *
DLOG                            INTRINSIC
I       INTEGER*4       43952
M       INTEGER*4           0 *
SAL     REAL*8              8 *
```

```
      C
      C
      C****************************************************************
      C
      C        SUBROUTINE EQL (M,A1,A2)
      C
      C        > EQUATES ONE VECTOR TO ANOTHER
      C
      C****************************************************************
      C
            SUBROUTINE EQL (M,A1,A2)
      C
            REAL*8 A1(M),A2(M)
```

```
D Line# 1      7
                 DO 10 I=1,M
1                  A2(I)=A1(I)
1         10     CONTINUE
          C
               END
```

Name	Type	Offset	P	Class
A1	REAL*8	4	*	
A2	REAL*8	8	*	
I	INTEGER*4	43960		
M	INTEGER*4	0	*	

```
      C
      C
      C*****************************************************************
      C
      C       SUBROUTINE  SURPLS (M1,M2,N,B,X1,X2,CONSUR,A1,A2)
      C
      C       > CALCULATE CONSUMER SURPLUS CHANGE FOR MNL
      C
      C       CONSUR: CONSUMER'S SURPLUS CHANGE
      C
      C          CALL ACCUM (M1, N, B, X1, BASE, A1)
      C          CALL ACCUM (M2, N, B, X2, ANEW, A2)
      C          CALL AERR  (ANEW, BASE)
      C
      C*******************************************************************
      C
      C
            SUBROUTINE SURPLS (M1,M2,N,B,X1,X2,CONSUR,A1,A2)
      C
            REAL*8 ANEW,BASE,A1(M1),A2(M2),B(N),CONSUR,X1(M1,N),X2(M2,N)
      C
      C  COMPUTE DENOMINATOR
            CALL ACCUM (M1,N,B,X1,BASE,A1)
      C
      C  COMPUTE NUMERATOR
            CALL ACCUM (M2,N,B,X2,ANEW,A2)
      C
      C  COMPUTE CONSUMER SURPLUS CHANGE
      C    B(1) MUST BE PRICE COEFFICIENT
            IF ((ANEW .LE. 0) .OR. (BASE .LE. 0)) CALL AERR (ANEW,BASE)
            CONSUR=(DLOG(ANEW)-DLOG(BASE))/(-B(1))
      C
            END
```

Name	Type	Offset	P	Class
A1	REAL*8	28	*	
A2	REAL*8	32	*	
ANEW	REAL*8	43976		
B	REAL*8	12	*	
BASE	REAL*8	43968		
CONSUR	REAL*8	24	*	

```
D Line# 1      7                        Microsoft FORTRAN77 V3.31 August 1985
DLOG                             INTRINSIC
M1       INTEGER*4          0 *
M2       INTEGER*4          4 *
N        INTEGER*4          8 *
X1       REAL*8            16 *
X2       REAL*8            20 *
```

```
      C
      C***************************************************************
      C
      C         SUBROUTINE ACCUM (M, N, B, X, ACC, A)
      C
      C        > COMPUTE SUMS OF EXPONENTIATED UTILITY
      C
      C        ACC: SUM OF EXPONENTIATED UTILITY
      C
      C***************************************************************
      C
      C
               SUBROUTINE ACCUM (M,N,B,X,ACC,A)
      C   COMPUTE SUMS OF EXPONENTIATED COMPOSITE COSTS
               REAL*8 BSUM,ACC,A(M),B(N),X(M,N)
      C
      C   INITIALIZE SUM OF EXPONENTIALS
               ACC=0.0
      C   COMPUTE SUM
               DO 20 I=1,M
1     C   INITIALIZE COMPOSITE PRICE SUM
1              BSUM=A(I)
1     C   COMPUTE COMPOSITE PRICE
1                DO 10 J=1,N
2                 BSUM=BSUM+(B(J)*X(I,J))
2      10        CONTINUE
1              ACC=ACC+DEXP(BSUM)
1      20      CONTINUE
      C
               END
```

```
Name     Type          Offset P Class

A        REAL*8            20 *
ACC      REAL*8            16 *
B        REAL*8             8 *
BSUM     REAL*8         43992
DEXP                              INTRINSIC
I        INTEGER*4      43984
J        INTEGER*4      44000
M        INTEGER*4          0 *
N        INTEGER*4          4 *
X        REAL*8            12 *
```

```
      C***************************************************************
      C
      C        SUBROUTINE HDMSUR (M1, M2, N, B, E, X1, X2, CONSUR, SGN, A1,A2)
```

```
      C
      C        > CALCULATE CONSUMER'S SURPLUS CHANGE IN HDM
      C
      C          CALL    RANSET (ITER, K, NEW1, N)
      C          CALL    RANDCO (N, B, E, BLN, K, NEW1, SGN)
      C          CALL    SURPLS (M1, M2, N, BLN, X1, X2, CON, A1, A2)
      C
      C*****************************************************************
      C
                SUBROUTINE HDMSUR (M1,M2,N,B,E,X1,X2,CONSUR,SGN,A1,A2)
      C
                INTEGER*4 NEW1
                INTEGER*2 K
                REAL*8 A1(M1),A2(M2),B(N),BLN(16),CON,CONSUR,E(N),SGN(N),
               1 X1(M1,N),X2(M2,N)
      C
      C  CALCULATE CONSUMER SURPLUS IN HDM
      C
                CALL RANSET (ITER,K,NEW1,N)
                CONSUR=0.0
                DO 10 I=1,ITER
1     C
1                CALL RANDCO (N,B,E,BLN,K,NEW1,SGN)
1               CALL SURPLS (M1,M2,N,BLN,X1,X2,CON,A1,A2)
1                CONSUR=CONSUR+(CON/ITER)
1       10       CONTINUE
      C
                END
```

Name	Type	Offset	P	Class
A1	REAL*8	36	*	
A2	REAL*8	40	*	
B	REAL*8	12	*	
BLN	REAL*8	44008		
CON	REAL*8	44154		
CONSUR	REAL*8	28	*	
E	REAL*8	16	*	
I	INTEGER*4	44146		
ITER	INTEGER*4	44136		
K	INTEGER*2	44140		
M1	INTEGER*4	0	*	
M2	INTEGER*4	4	*	
N	INTEGER*4	8	*	
NEW1	INTEGER*4	44142		
SGN	REAL*8	32	*	
X1	REAL*8	20	*	
X2	REAL*8	24	*	

```
      C
      C*****************************************************************
      C
      C          SUBROUTINE RANSET (ITER, K, NEW1, N)
      C
      C        > OBTAIN PARAMETERS OF MONTE CARLO INTEGRATION
```

```
C
C      ITER: NUMBER OF ITERATIONS
C      K: NUMBER OF PAIRS OF RANDOM VARIABLES
C      NEW1: RANDOM NUMBER SEED
C      N: NUMBER OF ATTRIBUTES
C
C*******************************************************************
C
      SUBROUTINE RANSET (ITER,K,NEW1,N)
      INTEGER*4 ISEED,NEW1
      INTEGER*2 K
C
C  OBTAIN DESIRED NUMBER OF ITERATIONS
      WRITE (*,100)
100   FORMAT(1H0,'HOW MANY ITERATIONS FOR HDM MONTE CARLO?')
      READ (*,110) ITER
110   FORMAT(I4)
C  MONTE CARLO INTEGRATION OF HDM SURPLUS
C
      K=INT(N/2)
      KK=N-2*K
      IF (KK .GT. 0) K=K+1
C  OBTAIN SEED FOR RANDOM NUMBER GENERATOR
      WRITE (*,200)
200   FORMAT(1H0,'ENTER A NUMBER FROM 0 TO 46300')
      READ (*,210) ISEED
210   FORMAT(I5)
      NEW1=ISEED
C
      END
```

Name	Type	Offset	P	Class
INT				INTRINSIC
ISEED	INTEGER*4	44314		
ITER	INTEGER*4	0	*	
K	INTEGER*2	4	*	
KK	INTEGER*4	44270		
N	INTEGER*4	12	*	
NEW1	INTEGER*4	8	*	

```
C*************************************************************
C
C      SUBROUTINE  RANDCO (N,B,E,BLN,K,NEW1,SGN)
C
C       > GENERATE RANDOM COEFFICIENTS
C
C       BLN: RANDOM COEFFICIENTS
C
C              CALL NORMAL (NEW1,RAN,K)
C
C*************************************************************
C
C
      SUBROUTINE RANDCO (N,B,E,BLN,K,NEW1,SGN)
```

```
D Line# 1      7                                Microsoft FORTRAN77 V3.31 August 1985
                INTEGER*4 NEW1
                INTEGER*2 K
                REAL*8 BLN(N),RAN(16),B(N),E(N),SGN(N)
           C
           C  CALL SUBROUTINE FOR 2K NORMAL 0,1 RANDOM VARIABLES
                 CALL NORMAL (NEW1,RAN,K)
           C  COMPUTE LOG-NORMALLY DISTRIBUTED COEFFICIENTS
                DO 10 I=1,N
1                BLN(I)=DEXP(B(I)+RAN(I)*E(I))*SGN(I)
1          10   CONTINUE
                RETURN
                END
```

Name	Type	Offset	P	Class
B	REAL*8	4	*	
BLN	REAL*8	12	*	
DEXP				INTRINSIC
E	REAL*8	8	*	
I	INTEGER*4	44450		
K	INTEGER*2	16	*	
N	INTEGER*4	0	*	
NEW1	INTEGER*4	20	*	
RAN	REAL*8	44322		
SGN	REAL*8	24	*	

```
           C***************************************************************
           C
           C
           C      SUBROUTINE  NORMAL  (NEW1, RAN, K)
           C
           C       > GENERATE A VECTOR OF NORMAL RANDOM VARIABLES
           C
           C               CALL  RAN1  (SN,SN1,NEW1)
           C
           C       RAN: NORMALLY DISTRIBUTED RANDOM VARIABLES
           C       SN: FIRST OF PAIR OF NORMAL RANDOM VARIABLES
           C       SN1: SECOND OF PAIR OF NORMAL RANDOM VARIABLES
           C
           C  SUBROUTINE FOR PAIRS OF INDEPENDENT NORMAL 0,1 VARIATES
           C  BASED ON P.F. HULTQUIST, PC TECH JOUR., 8/84, AND J MOSHMAN
           C  RANDOM NUMBER GENERATION, IN H.S. WILF, V.2 MATHEMATICAL METHODS
           C  FOR DIGITAL COMPUTERS
           C
           C***************************************************************
           C
                SUBROUTINE NORMAL (NEW1,RAN,K)
                INTEGER*4 NEW1
                INTEGER*2 G,K
                REAL*8 SN,SN1,RAN(16)
           C
                J=1
                DO 15 I=1,K
1               CALL RAN1 (SN,SN1,NEW1)
1               RAN(J)=SN1
```

```
D Line# 1      7
1              RAN(J+1)=SN
1              J=J+2
1         15   CONTINUE
               END
```

Name	Type	Offset	P	Class
G	INTEGER*2	*****		
I	INTEGER*4	44462		
J	INTEGER*4	44458		
K	INTEGER*2	8	*	
NEW1	INTEGER*4	0	*	
RAN	REAL*8	4	*	
SN	REAL*8	44470		
SN1	REAL*8	44478		

```
C
C
C***************************************************************
C
C          SUBROUTINE  RAN1  (SN,SN1,NEW1)
C
C        > GENERATE PAIRS OF NORMAL RANDOM NUMBERS
C
C***************************************************************
        SUBROUTINE RAN1  (SN,SN1,NEW1)
        INTEGER*4 NEW1,NEW,M
        INTEGER*2 G
        REAL*8 Z,SN,SN1,UNIF,UNIF1
C
C  GENERATE UNIFORM RANDOM NUMBERS
        G=0
        M=46300
        NEW=9261*NEW1+3
        NEW=MOD(NEW,M)
        UNIF=NEW/46300.0
1       CONTINUE
C
C  TRANSFORM UNIFORM PAIRS TO 0,1 NORMAL PAIRS
        IF (G-1) 7,6,7
7       G=G+1
        NEW1=NEW
        UNIF1=UNIF
        GOTO 1
        IF (UNIF1 .LE. 0) CALL UERR (NEW1,UNIF1,K)
6       Z=DSQRT(-2.0*DLOG(UNIF1))
        SN1=Z*DCOS(6.28318*UNIF)
        SN=Z*DSIN(6.28318*UNIF)
        G=0
        NEW1=NEW
        RETURN
        END
```

Name   Type   Offset P Class

```
D Line# 1       7                                      Microsoft FORTRAN77 V3.31 August 1985
DCOS                                INTRINSIC
DLOG                                INTRINSIC
DSIN                                INTRINSIC
DSQRT                               INTRINSIC
G       INTEGER*2       44486
K       INTEGER*4       44512
M       INTEGER*4       44488
MOD                                 INTRINSIC
NEW     INTEGER*4       44492
NEW1    INTEGER*4           8 *
SN      REAL*8              0 *
SN1     REAL*8              4 *
UNIF    REAL*8          44496
UNIF1   REAL*8          44504
Z       REAL*8          44516
```

```
      C
      C
      C**************************************************************
      C
      C      SUBROUTINE  PRED1   (M,A,BS,SAL)
      C
      C       > PREDICT NEW SHARES FOR MNL MODEL
      C
      C**************************************************************
      C
      C
             SUBROUTINE PRED1     (M,A,BS,SAL)
      C
             REAL*8 A(M),BS(M),SAL(M),DEN
      C
             DEN=0.0
             DO 10 I=1,M
1              DEN=DEN+DEXP(A(I)+BS(I))
1     10     CONTINUE
             DO 20 I=1,M
1              SAL(I)=DEXP(A(I)+BS(I))/DEN
1     20     CONTINUE
      C
             END
```

```
Name     Type          Offset P Class

A       REAL*8              4 *
BS      REAL*8              8 *
DEN     REAL*8          44524
DEXP                                INTRINSIC
I       INTEGER*4       44532
M       INTEGER*4           0 *
SAL     REAL*8             12 *
```

```
      C
      C**************************************************************
      C
```

```
D Line# 1      7                                  Microsoft FORTRAN77 V3.31 August 1985
          C      SUBROUTINE  PRED2  (M,A,N,B,E,SGN,X,BS2,SAL2)
          C
          C       > PREDICT NEW SHARES FOR HDM MODEL
          C
          C         CALL  RANSET (ITER,K,NEW1,N)
          C         CALL RANDCO (N,B,E,BLN,K,NEW1,SGN)
          C         CALL ACCUM  (M,N,BLN,X,DEN,A)
          C         CALL BSUM1 (M,X,BLN,N,BS2)
          C
          C************************************************************
          C
                 SUBROUTINE PRED2    (M,A,N,B,E,SGN,X,BS2,SAL2)
          C
                 REAL*8 A(M),B(N),DEN,BS2(M),SAL2(M),BLN(16),CONSUR,PSHR(200),
                1 E(N),SGN(N),X(M,N)
                 INTEGER*4 ISEED,NEW1
                 INTEGER*2 K
          C
                 CALL RANSET (ITER,K,NEW1,N)
          C INITIALIZE PREDICTED SHARES
                 DO 100 I=1,M
1                  SAL2(I)=0.0
1         100    CONTINUE
          C BEGIN MONTE CARLO ITERATIONS
                 DO 20 J=1,ITER
1                CALL RANDCO (N,B,E,BLN,K,NEW1,SGN)
1                CALL ACCUM (M,N,BLN,X,DEN,A)
1                CALL BSUM1 (M,X,BLN,N,BS2)
1                DO 10 I=1,M
2                  PSHR(I)=DEXP(A(I)+BS2(I))/DEN
2                  SAL2(I)=SAL2(I)+(PSHR(I)/ITER)
2         10     CONTINUE
1         20     CONTINUE
          C
                 END
```

Name	Type	Offset	P	Class
A	REAL*8	4	*	
B	REAL*8	12	*	
BLN	REAL*8	46144		
BS2	REAL*8	28	*	
CONSUR	REAL*8	*****		
DEN	REAL*8	46298		
DEXP				INTRINSIC
E	REAL*8	16	*	
I	INTEGER*4	46282		
ISEED	INTEGER*4	*****		
ITER	INTEGER*4	46272		
J	INTEGER*4	46290		
K	INTEGER*2	46276		
M	INTEGER*4	0	*	
N	INTEGER*4	8	*	
NEW1	INTEGER*4	46278		
PSHR	REAL*8	44544		
SAL2	REAL*8	32	*	

```
SGN     REAL*8          20 *
X       REAL*8          24 *
```

```
C
C***************************************************************
C
C       SUBROUTINE  SURPRN (M1,M2,N,MODEL,CONSUR)
C
C        > PRINT RESULTS OF CONSUMER SURPLUS CALCULATION
C
C***************************************************************
C
C
C
        SUBROUTINE SURPRN (M1,M2,N,MODEL,CONSUR)
C  PRINT RESULTS OF CONSUMER SURPLUS CALCULATIONS
        REAL*8 CONSUR
C
        WRITE (*,100)
 100    FORMAT(1H0,'        CALCULATIONS COMPLETED')
        IF (MODEL .EQ. 1) WRITE (*,110)
 110    FORMAT(1H0,'MULTINOMIAL LOGIT MODEL - CONSUMER SURPLUS')
        IF (MODEL .EQ. 2) WRITE (*,120)
 120    FORMAT(1H0,'HEDONIC DEMAND MODEL - CONSUMER SURPLUS')
        WRITE (*,130) M1,M2
 130    FORMAT(1H0,'BASE CHOICES = ',I3,'  NEW CHOICES = ',I3)
        WRITE (*,140) CONSUR
 140    FORMAT(1H0,'SURPLUS CHANGE PER CONSUMER = $',E15.9)
C
        END
```

```
Name     Type        Offset P Class

CONSUR  REAL*8          16 *
M1      INTEGER*4        0 *
M2      INTEGER*4        4 *
MODEL   INTEGER*4       12 *
N       INTEGER*4        8 *
```

```
C
C****************************************************************
C
C       SUBROUTINE  PREPRN (MODEL,M1,SAL1,SAL2,M2)
C
C        > PRINT RESULTS OF SHARES PREDICTION
C
C****************************************************************
C
        SUBROUTINE PREPRN (MODEL,M1,SAL1,SAL2,M2)
C
        REAL*8 SAL1(M1),SAL2(M2)
C
        WRITE (*,100)
 100    FORMAT(1H0,'BASE YEAR AND PREDICTED MARKET SHARES')
```

```
D Line# 1      7
        C
               WRITE (*,110)
        110    FORMAT(1H0,'                  BASE YEAR          PREDICTED')
               DO 10 I=1,M1
1                WRITE (*,200) I,SAL1(I),SAL2(I)
1       10     CONTINUE
        200    FORMAT(4X,I3,3X,2E15.8)
               END
```

Name	Type	Offset	P	Class
I	INTEGER*4	46664		
M1	INTEGER*4	4	*	
M2	INTEGER*4	16	*	
MODEL	INTEGER*4	0	*	
SAL1	REAL*8	8	*	
SAL2	REAL*8	12	*	

```
      C***************************************************************
      C
      C       SUBROUTINE  UERR (NEW1,UNIF1,K)
      C
      C****************************************************************
      C
      C  ERROR CHECKING SUBROUTINES
      C
      C
              SUBROUTINE UERR (NEW1,UNIF1,K)
              INTEGER*4 NEW1,UNIF1
              INTEGER*2 K
              WRITE (*,100) NEW1,UNIF1,K
        100   FORMAT(1H0,'ERROR IN RAN1.  NEW1=',E15.9,'UNIF1=',E15.9,'K=',I3)
              END
```

Name	Type	Offset	P	Class
K	INTEGER*2	8	*	
NEW1	INTEGER*4	0	*	
UNIF1	INTEGER*4	4	*	

```
      C
      C****************************************************************
      C
      C        SUBROUTINE  SERR (M,B,SAL,I)
      C
      C****************************************************************
      C
              SUBROUTINE SERR (M,B,SAL,I)
              REAL*8 B(M),SAL(M)
              WRITE (*,100) M,SAL(1),SAL(I),I
        100   FORMAT(1H0,'ERROR IN SUM.  M=',I3,'SAL(1)=',E15.9,'SAL(I)=',E15.9
             1 ,'I=',I3)
              END
```

D Line# 1 7

Name	Type	Offset	P	Class
B	REAL*8	4	*	
I	INTEGER*4	12	*	
M	INTEGER*4	0	*	
SAL	REAL*8	8	*	

```
  C
  C
  C*************************************************************
  C
  C       SUBROUTINE  AERR (ANEW,BASE,B,N)
  C
  C*************************************************************
          SUBROUTINE AERR (ANEW,BASE,B,N)
          REAL*8 B(N),ANEW,BASE
          WRITE (*,100) ANEW,BASE
      100 FORMAT(1H0,'ERROR IN SURPLS.  ANEW=',E15.9,'BASE=',E15.9)
          DO 10 I=1,N
1           WRITE (*,200) I,B(I)
1      10 CONTINUE
      200 FORMAT(1H0,'B',I3,' = ',E15.9)
          END
```

Name	Type	Offset	P	Class
ANEW	REAL*8	0	*	
B	REAL*8	8	*	
BASE	REAL*8	4	*	
I	INTEGER*4	46874		
N	INTEGER*4	12	*	

```
  C
```

Name	Type	Size	Class
ACCUM			SUBROUTINE
AERR			SUBROUTINE
BSUM1			SUBROUTINE
BSUM2			SUBROUTINE
CALIB			SUBROUTINE
CHOICE			SUBROUTINE
EQL			SUBROUTINE
HDMSUR			SUBROUTINE
INIT			SUBROUTINE
NORMAL			SUBROUTINE
PAR1			SUBROUTINE
PAR2			SUBROUTINE
PRED1			SUBROUTINE
PRED2			SUBROUTINE
PREPRN			SUBROUTINE
RAN1			SUBROUTINE
RANDCO			SUBROUTINE

RANSET	SUBROUTINE
RUMS	PROGRAM
SERR	SUBROUTINE
SETUP	SUBROUTINE
SHRS	SUBROUTINE
SUM	SUBROUTINE
SURPLS	SUBROUTINE
SURPRN	SUBROUTINE
UERR	SUBROUTINE
WEIGHT	SUBROUTINE

Pass One No Errors Detected
840 Source Lines

Appendix B

Sample Model Runs

Appendix B

Sample Model Runs

System configuration. IBM PC with one floppy disk and one hard disk drive. Program RUMS.EXE resides on the hard disk and data are read from drive a:.

The first six runs illustrate the various options for estimating consumer's surplus using multinomial logit and hedonic demand models. The last two illustrate market shares predictions with the two models.

```
rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:pold.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
1

YOU CHOSE MODEL 1.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
1

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE    6 &    6 CHOICES &   6 CHARS.

 READING MNL MODEL PARAMETERS . . .

ENTER RECORD FORMAT FOR PARAMS.DAT.
```

```
(f9.6)

ENTER 1 TO PRINT PARAMETERS
1
-.286000000E-03
-.339000000E+00
 .374000000E-01
 .137000000E+01
 .375000000E+00
 .462000000E+00

READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA
1

ENTER OUTPUT FORMAT, E.G. 1X,INPUT FORMAT
(1x,7f7.2)
3390.00   3.60  17.10   4.38   3.85   2.76  22.90
3930.00   5.25  11.20   5.00   3.92   3.43  18.80
4190.00   5.77  17.30   5.43   4.18   3.95  17.10
4630.00   5.96  17.50   5.34   4.35   4.54  20.50
8560.00   6.56  18.00   5.52   4.50   4.47   5.20
4700.00   5.72  13.80   5.45   4.27   3.78  15.60
3390.00   2.92  17.10   4.38   3.47   2.76  24.92
3930.00   3.45  11.20   4.75   3.53   3.43  21.15
4190.00   4.46  17.30   5.16   3.77   3.95  15.74
4630.00   4.81  17.50   5.07   3.91   4.54  17.67
9560.00   5.29  18.00   5.52   4.50   4.47   5.99
6700.00   4.23  13.80   5.45   4.27   3.78  14.54

WHAT TYPE OF WEIGHTING FOR LOGSUM TERMS?  ENTER:

    1 FOR SHARES

    2 FOR NO WEIGHTS
2

       CALCULATIONS COMPLETED

MULTINOMIAL LOGIT MODEL - CONSUMER SURPLUS

BASE CHOICES =    6  NEW CHOICES =    6

SURPLUS CHANGE PER CONSUMER = $-.116255301E+03
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:pold.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
1

YOU CHOSE MODEL 1.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
1

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE   6 &   6 CHOICES &  6 CHARS.

 READING MNL MODEL PARAMETERS . . .

ENTER RECORD FORMAT FOR PARAMS.DAT.
```

```
(f9.6)

ENTER 1 TO PRINT PARAMETERS


READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


WHAT TYPE OF WEIGHTING FOR LOGSUM TERMS?  ENTER:

    1 FOR SHARES

    2 FOR NO WEIGHTS
1

       CALCULATIONS COMPLETED

MULTINOMIAL LOGIT MODEL - CONSUMER SURPLUS

BASE CHOICES =    6  NEW CHOICES =    6

SURPLUS CHANGE PER CONSUMER = $-.292400481E+03
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:pold.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
1

YOU CHOSE MODEL 1.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
3

FOR OPTION 3  SETS NUMBER AND ORDER MUST AGREE

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE    6 &    6 CHOICES &   6 CHARS.

 READING MNL MODEL PARAMETERS . . .

ENTER RECORD FORMAT FOR PARAMS.DAT.
```

```
(f9.6)

ENTER 1 TO PRINT PARAMETERS


READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


       CALCULATIONS COMPLETED

MULTINOMIAL LOGIT MODEL - CONSUMER SURPLUS

BASE CHOICES =    6  NEW CHOICES =    6

SURPLUS CHANGE PER CONSUMER = $ .900960068E+01
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:params.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
2

YOU CHOSE MODEL 2.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
1

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE    6 &    6 CHOICES &   6 CHARS.

 READING HDM MODEL PARAMETERS . . .

ENTER RECORD FORMAT FOR PARAMS.DAT.
```

```
(3f6.3)

ENTER 1 TO PRINT PARAMETERS
1
-.794000000E+01 .118000000E+01-.100000000E+01
-.128000000E+01 .100000000E-02-.100000000E+01
-.410000000E+01 .999000000E+00 .100000000E+01
 .569000000E+00 .622000000E+00 .100000000E+01
-.175000000E+01 .134000000E+01 .100000000E+01
-.263000000E+01 .234000000E+01 .100000000E+01

READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


WHAT TYPE OF WEIGHTING FOR LOGSUM TERMS?  ENTER:

    1 FOR SHARES

    2 FOR NO WEIGHTS
1

HOW MANY ITERATIONS FOR HDM MONTE CARLO?
1000

ENTER A NUMBER FROM 0 TO 46300
3265

       CALCULATIONS COMPLETED

HEDONIC DEMAND MODEL - CONSUMER SURPLUS

BASE CHOICES =    6  NEW CHOICES =    6

SURPLUS CHANGE PER CONSUMER = $-.123954701E+04
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:params.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
2

YOU CHOSE MODEL 2.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
3

FOR OPTION 3  SETS NUMBER AND ORDER MUST AGREE

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE   6 &   6 CHOICES &  6 CHARS.

 READING HDM MODEL PARAMETERS . .  .

ENTER RECORD FORMAT FOR PARAMS.DAT.
```

```
(3f6.3)

ENTER 1 TO PRINT PARAMETERS


READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


HOW MANY ITERATIONS FOR HDM MONTE CARLO?
1000

ENTER A NUMBER FROM 0 TO 46300
3265

       CALCULATIONS COMPLETED

HEDONIC DEMAND MODEL - CONSUMER SURPLUS

BASE CHOICES =    6  NEW CHOICES =    6

SURPLUS CHANGE PER CONSUMER = $-.564065774E+03
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:params.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
2

YOU CHOSE MODEL 2.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
1

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE    6 &    6 CHOICES &   6 CHARS.

 READING HDM MODEL PARAMETERS . . .

ENTER RECORD FORMAT FOR PARAMS.DAT.
(f9  3f6.3)

ENTER 1 TO PRINT PARAMETERS
```

```
READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


WHAT TYPE OF WEIGHTING FOR LOGSUM TERMS?  ENTER:

    1 FOR SHARES

    2 FOR NO WEIGHTS
0

HOW MANY ITERATIONS FOR HDM MONTE CARLO?
1000

ENTER A NUMBER FROM 0 TO 46300
3265

       CALCULATIONS COMPLETED

HEDONIC DEMAND MODEL - CONSUMER SURPLUS

BASE CHOICES =    6  NEW CHOICES =    6

SURPLUS CHANGE PER CONSUMER = $-.942366764E+03
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:pold.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
1

YOU CHOSE MODEL 1.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
2

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE   6 &   6 CHOICES &  6 CHARS.

 READING MNL MODEL PARAMETERS . . .

ENTER RECORD FORMAT FOR PARAMS.DAT.
(f9.6)

ENTER 1 TO PRINT PARAMETERS
```

```
READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


BASE YEAR AND PREDICTED MARKET SHARES

            BASE YEAR          PREDICTED
     1    .22877123E+00   .24917757E+00
     2    .18781219E+00   .21151919E+00
     3    .17082917E+00   .15735861E+00
     4    .20479520E+00   .17668769E+00
     5    .51948052E-01   .59871313E-01
     6    .15584416E+00   .14538563E+00
```

```
C:\FORT>rums

PLEASE ENTER COMPLETE FILENAME FOR PARAMETERS
File name missing or blank - Please enter name

UNIT 11? a:params.dat


PLEASE ENTER FILENAME FOR CHARACTERISTICS
UNIT 12? a:chars.dat


ESTIMATION OF CONSUMER SURPLUS IN RUMS

WHAT TYPE OF MODEL?  ENTER:

    1 FOR MULTINOMIAL LOGIT

    2 FOR HEDONIC DEMAND MODEL
2

YOU CHOSE MODEL 2.

WHAT TYPE OF ESTIMATION?   ENTER:

    1 FOR CONSUMER SURPLUS CHANGE

    2 FOR SHARES PREDICTION

    3 FOR C S CHANGE, CORRESPONDING SET MEMBERS
2

ENTER NUMBER OF ELEMENTS IN BASE CHOICE SET -
6

ENTER NUMBER OF ELEMENTS IN NEW CHOICE SET -
6

ENTER NUMBER OF CHARACTERISTICS -
6

THERE ARE   6 &    6 CHOICES &   6 CHARS.

 READING HDM MODEL PARAMETERS . .  .

ENTER RECORD FORMAT FOR PARAMS.DAT.
(3f6.3)

ENTER 1 TO PRINT PARAMETERS
```

```
READING CHARACTERISTICS & SALES DATA

ENTER RECORD FORMAT FOR CHARS.DAT.
(7f7.2)

ENTER 1 TO PRINT CHARACTERISTICS DATA


HOW MANY ITERATIONS FOR HDM MONTE CARLO?
1000

ENTER A NUMBER FROM 0 TO 46300
3265

BASE YEAR AND PREDICTED MARKET SHARES

              BASE YEAR         PREDICTED
     1     .22877123E+00   .38661450E+00
     2     .18781219E+00   .927540B0E-01
     3     .17082917E+00   .77710762E-01
     4     .20479520E+00   .11016109E+00
     5     .51948052E-01   .24268173E+00
     6     .15584416E+00   .90077840E-01
```

INTERNAL DISTRIBUTION

1-2. Center for Energy and Environmental Information
3. J. Christian
4. G. A. Dailey
5. T. Dinan
6. S. D. Floyd
7. W. Fulkerson
8-12. D. L. Greene
13. V. J. Harley
14. L. J. Hill
15. E. L. Hillsman
16. R. B. Honea
17. C. R. Kerley
18. R. Lee
19. F. C. Maienschein
20. V. C. Mei
21. C. G. Rizy
22. R. B. Shelton
23. R. A. Stevens
24. D. Vogt
25. T. J. Wilbanks
26. G. P. Zimmerman
27-28. Central Research Library
29. Document Reference Section
30. Laboratory Records
31-32. Laboratory Records - RC
33. ORNL Patent Office

EXTERNAL DISTRIBUTION

34. Office of Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations Office, Oak Ridge, TN 37831.
35. Jaime G. Carbonell, Associate Professor of Computer Science, Carnegie-Mellon University, Pittsburgh, PA 15213.
36. S. Malcolm Gillis, Dean, Graduate School, Duke University, 4875 Duke Station, Durham, NC 27706.
37. Fritz R. Kalhammer, Vice President, Electric Power Research Institute, Post Office Box 10412, Palto Alto, CA 94303.
38. Roger E. Kasperson, Professor of Government and Geography, Graduate School of Geography, Clark University, Worcester, MA 01610.
39. Martin Lessen, Consulting Engineer, 12 Country Club Drive, Rochester, NY 14618.
40-63. Transportation Group Distribution, Energy Division.
64-90. Technical Information Center, P.O. Box 62, Oak Ridge, TN 37831.