Division of Research
Graduate School of Business Administration
The University of Michigan

December 1983

THE 1979-80 UNIVERSITY OF MICHIGAN HEATING PLANT AND UTILITIES COST ALLOCATION STUDY

WORKING PAPER #352

Roger L. Wright and Karen Oberg
The University of Michigan

Table of Contents

This paper documents the allocation by the University of Michigan of heating plant and utilities cost to organized research and other functions for the fiscal year ending on 6-30-80. It describes in detail the procedures which the University followed in its allocation and the rationale for using these procedures. The University's allocation process consists of two main steps. The first involves estimating the utilities cost per square foot of building space and the second involves allocating these costs according to the percent of space which was used for organized research. A brief description of the program which was used for the allocation will be given, and a copy of the program and its output is attached.

1. Background

The total space of any building can be divided into structural space and net (non-structural) space. Only net space was used in the allocation process. In previous years the University has divided the actual utilities cost per building by the net square feet per building to get the average utilities cost per square foot for each building. However, this procedure assumed that the actual utilities cost per square foot is constant throughout a building, which is very unlikely. The University thought that a more equitable allocation could be constucted by dividing net space into areas which incur approximately the same amount of utilities cost per square foot. In particular, the University Office of Audits believed that lab space has a higher utility cost per square foot than non-lab space. Lab space is probably used more frequently than non-lab space and lab space is usually more heavily equipped with appliances and special machinery than non-lab space. Also, air turnover is necessarily higher in labs, leading to high ventilation requirements.

To get some idea of whether they were proceeding in the right direction, the Office of Audits conducted a short analysis of this idea. In their analysis they chose nine buildings which contain no lab space and eighteen buildings which contain both lab and non-lab space. The buildings were all chosen from the central campus area because it was thought that this is most representative of the University. An equivalent percent of air conditioned buildings was chosen for each sample (five out of nine and ten out of eighteen). The sum over the nine buildings of the actual utilities cost for the fiscal year ending on 6-30-79 was divided by the sum over the nine buildings of the assignable square feet.[1] This gave an average cost per square foot of assignable space in the non-lab buildings of $1.5012. The same computations were performed on the eighteen buildings, yielding an average cost per square foot of assignable space of $2.8596. In order to separate out

---

[1]Assignable square feet is equal to net square feet minus unassignable square feet. Unassignable space consists mainly of corridors and custodial areas and will be described more completely below.

lab costs from the eighteen buildings, the sum over the eighteen buildings of all the non-lab space was multiplied by the estimated cost per square foot of non-lab space ($1.5012). This was subtracted from the total utilities cost of the eighteen buildings to get an estimate of the total utilities cost of the lab space. Finally, this difference was divided by the total lab space in the eighteen buildings in order to arrive at an estimate of the utilities cost per square foot of lab space. This was $4.2397, approximately 2.8 times greater than the estimated cost per square foot of non-lab space.

2. The Database

In order to refine the preceding classification of space, the University decided that assignable space should be divided into one more category, that of parking space. The utilities for parking space consist primarily of lighting which is set at a lower level than the lighting within a building, and of course there are minimal heating costs. Unassignable space was left in a category by itself. Unassignable space is space that isn't assigned to any particular person's or group's use. It consists generally of circulation, custodial and mechanical space. These four categories: non-lab space, lab space, parking space and unassignable space, represented subdivisions of net building space into major groups within which actual utilities cost per square foot was believed to be comparatively similar and among which actual utilities cost per square foot was believed to be significantly different.

The data used for the allocation came from three sources: the Engineering Department, the Plant Department and the Office of Space Analysis (part of the Office of Cost Reimbursement). The Engineering Department supplied the gross size (in square feet) of each building. It also supplied the following information for each of the rooms in each building: a description, the size in net interior square feet and the "type code". The "type codes" for rooms are taken from national definitions for higher education facilities of room types. (For example, "100" is the type code for classroom facilities, "200" is the type code for laboratory facilities and "300" is the type code for office facilities.) In its allocation the University grouped the following type codes together into lab space: 210- Class Laboratory, 215- Class Laboratory Service, 220- Special Class Laboratory, 225- Special Class Laboratory Service, 250- Non Class Laboratory, 255- Non Class Laboratory Service, 570- Animal Quarters ("A room that houses laboratory animals maintained for the institution for research and/or instruction purposes"[2]), 575- Animal Quarters Service and 840- Surgery. Also included in lab space were those rooms with type code 850- Treatment ("A

[2]Higher Education Facilities Inventory and Classification Manual, Appendix 6.2, p. 65.

room used for diagnostic and therapeutic treatment"[3]), which were used for radiology treatment only. The following room types were grouped together under the category unassignable space: 010- Custodial Area, 020- Circulation Area, 030- Mechanical Area, 050- Inactive Area ("Rooms that are available for assignment to an organizational unit or activity but are unassigned at the time of the inventory"[4]) , 060- Alteration or Conversion Area ("Rooms that are temporarily out of use"[5]), and 070- Unfinished Area ("All potentially assignable areas in new buildings or additions to existing buildings that are not completely finished at the time of the inventory"[6]). 740- Vehicle Storage Facility, was the type code for parking space. Non-lab space was defined as all assignable space which was not lab space or parking space. A database of the information supplied by the Engineering Department is kept by the Office of Space Analysis.

The Plant Department supplied the actual utilities cost for each building for the fiscal year ending on 6-30-79.

The Office of Space Analysis supplied the Annual Space Study. Following is an excerpt from the Office of Space Analysis' description of their space study:

---

[3]Ibid, p. 77.

[4]Ibid, p. 81.

[5]Ibid, p. 81.

[6]Ibid, p. 81.

The space study procedures are as follows: Annually, each dean, director, or department head is given a list(s) of all the rooms for which he is responsible (separate lists for each building). This list (titled Departmental Space Inventory Survey) shows: room number, room type code, room description, and square feet. Space is provided to report the functional use and the percentage for each function if there is more than one use. Accompanying the survey form are instructions including definitions of functions (instruction, organized research, departmental administration, etc.). The size of each room is stated in net square feet as determined and checked by the University Engineering Department. The dean, director, or department head is asked to determine and report the current functional use of each room and to verify or correct the room type code. When the completed survey forms are received in the Office of Cost Reimbursement, they are reviewed for completeness and consistency. Questionable responses are discussed with the units and resolved. The information is entered into a data bank. Two reports are issued to the Cost Reimbursement Office for use in allocating space related costs in the indirect cost study: (1) Space Study, Summary by Building, and (2) Space Study, Summary of Departments. The first report shows: building name, building number, square feet by function and by department. The second report summarizes square feet assigned to functions by department.

The total number of buildings given by the Engineering Department was 251. However, not all of these buildings were used as data in the linear regression which was implemented to estimate the utilities cost per square foot of lab, non-lab, parking, or unassignable space. Forty-eight buildings were excluded from the analysis, leaving 203 buildings. Thirty-six of the forty-eight were leased or rented and in these cases the utilities either weren't paid or identifiable by the University. Two of the forty-eight were under construction and two more of the forty-eight were undergoing renovation. The remaining eight of the forty-eight had unusual utilities requirements due to the functions for which they were used. The eight buildings were the Botanical Gardens, Ferry Field, the Food and Chemical Stores, the Laundry, North Campus Computing Center, Radrick Farms, the Seismograph Station and the Sheep Facility. Ferry Field is a football practice field. It was excluded because the actual building associated with Ferry Field is only a small building at the edge of the field, yet the cost of lighting the entire field is attributed to it.

3. Description of Data

The total utilities cost for the fiscal year ending on 6-30-80 of the 203 buildings was $22,460,517. Excluding structural space and unassignable space, the buildings had a total size of 12,525,435 square feet, of which 69.8 percent was nonlab, 13.7 percent was lab, and 16.4 percent was parking. Organized research used 1,164,260 square feet, of which 36.7 percent was nonlab and 63.2 percent was lab. The amount of parking space used by federally sponsored research was negligible.

The total utilities cost for the 251 buildings was $24,650,090.

4. Modeling

Since the University believed that utilities cost varies within a building according to the four basic categories of space, the first step toward an allocation was to show that the utilities cost of a building was directly proportional to the amounts of lab, non-lab, parking and unassignable space that are contained within that building and to estimate the average cost factors for each of these space types. Linear multiple regression analysis was performed on the data from the 203 buildings.

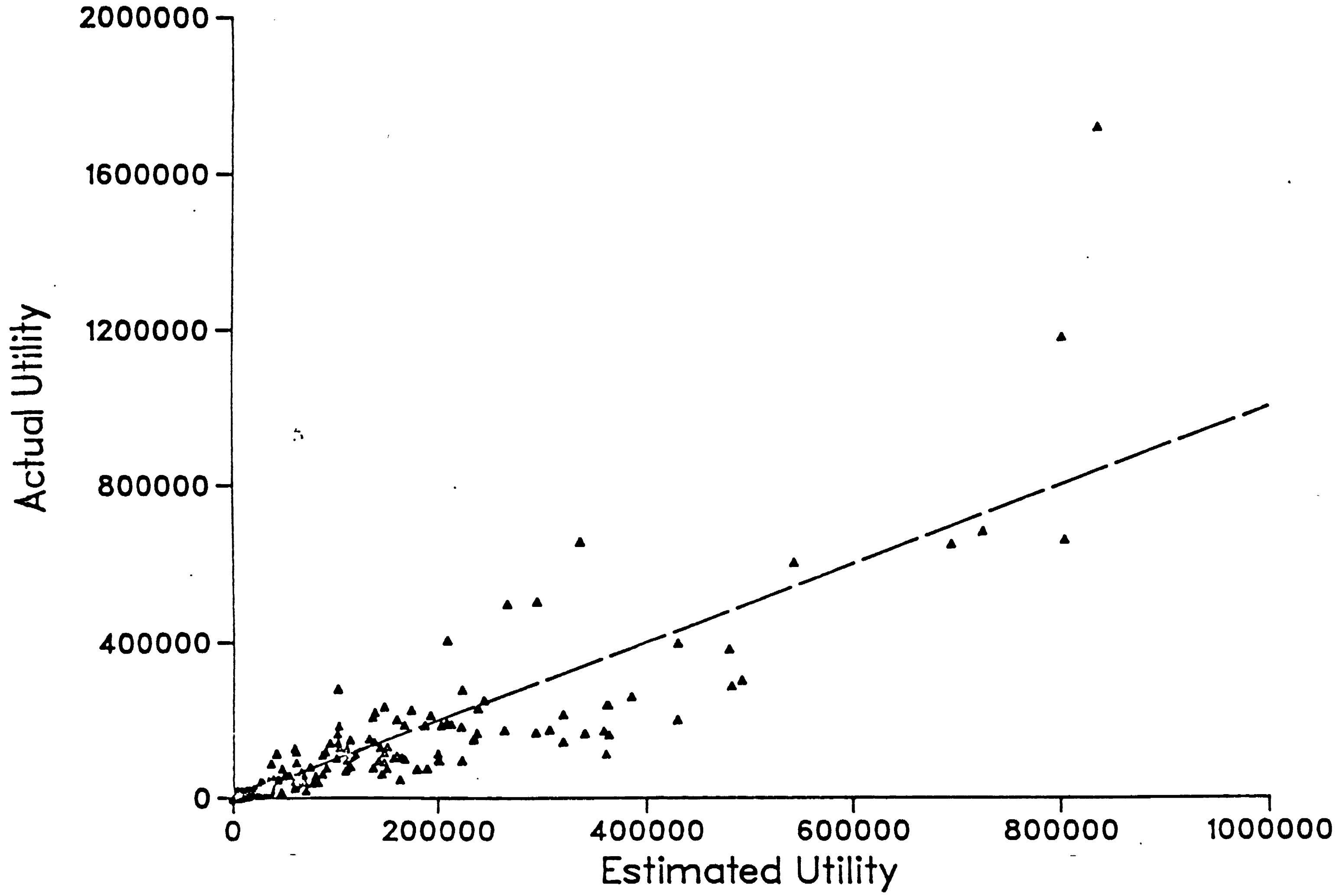
The dependent variable Y was the total utilities cost of each building. The independent variables were the square feet of each building for each of the four types of space. X_1 represented non-lab space, X_2 represented lab space, X_3 represented parking space and X_4 represented unassignable space. The regression coefficients represented the average utilities cost per square foot for each type of space.

$$Y_i = \beta_1 X_{i1} + \beta_2 X_{i2} + \beta_3 X_{i3} + \beta_4 X_{i4} + \varepsilon_i$$

where the "error term" ε_i was assumed to have a mean of zero. ε_i represented the difference between the actual utilities cost of a building and the utilities cost which could have been predicted if all the coefficients were known. The subscript i denotes the ith building. In this case ε_i accounted for factors specific to building i which caused building i's utilities cost to be higher or lower than the average building of the same size. These factors included architecture, construction and insulation.

If the variance of ε had been assumed to be constant in the above equation, the regression coefficients could have been estimated using ordinary least squares procedures. However, when scatter plots of the data were examined heteroscedasticity was suspected. Figure 1 shows

Figure 1. Estimated vs. Actual Utility



actual versus estimated utility prior to the correction for heteroscedasticity. Heteroscedasticity exists when the variance of ε isn't constant. There are serious problems involved with the validity of regresssion analysis if heteroscedasticity exists and the regression equation is not corrected for it:

(1) predictive intervals are misleading,

(2) the estimators of the coefficients are unbiased and consistent but are not efficient or asymptotically efficient and

(3) since the standard errors of the estimated coefficients are incorrect, hypothesis tests about and confidence intervals for the coefficients are incorrect.

One of the most common and most general ways of dealing with heteroscedasticity is to assume that the standard deviation of ε_i, σ_i, is related to another variable Z by the equation $\sigma_i = \sigma_0 Z_i^{\gamma}$, where σ_0 is constant. This assumption is useful because if the original model is divided by the factor Z^{γ}, the resulting transformed equation displays homoscedasticity and can be estimated.

$$(Y_i/Z_i^{\gamma}) = \beta_1(X_{i1}/Z_i^{\gamma}) + \beta_2(X_{i2}/Z_i^{\gamma}) + \beta_3(X_{i3}/Z_i^{\gamma}) + \beta_4(X_{i4}/Z_i^{\gamma}) + (\varepsilon_i/Z_i^{\gamma}).$$

The choice of Z which best accounts for the heteroscedasticity is not usually obvious. The choice for Z which was implemented in the cost allocation was the net size of each building in square feet. Figure 2 shows the data after it has been corrected for heteroscedasticity.

There are six unknown parameters in the above model; β_1, β_2, β_3, β_4, σ_0 and γ. An iterative weighted least squares algorithm was used in their estimation [7]. The results were as follows (the estimated standard deviations of the estimated coefficients are in parentheses):

$$Y_i = .65441X_{i1} + 3.1905X_{i2} + .14138X_{i3} + 2.5661X_{i4},$$

$$(.15482) \quad (.38540) \quad (.28081) \quad (.38454)$$

$$s_i = 1.146Z_i^{.97200},$$

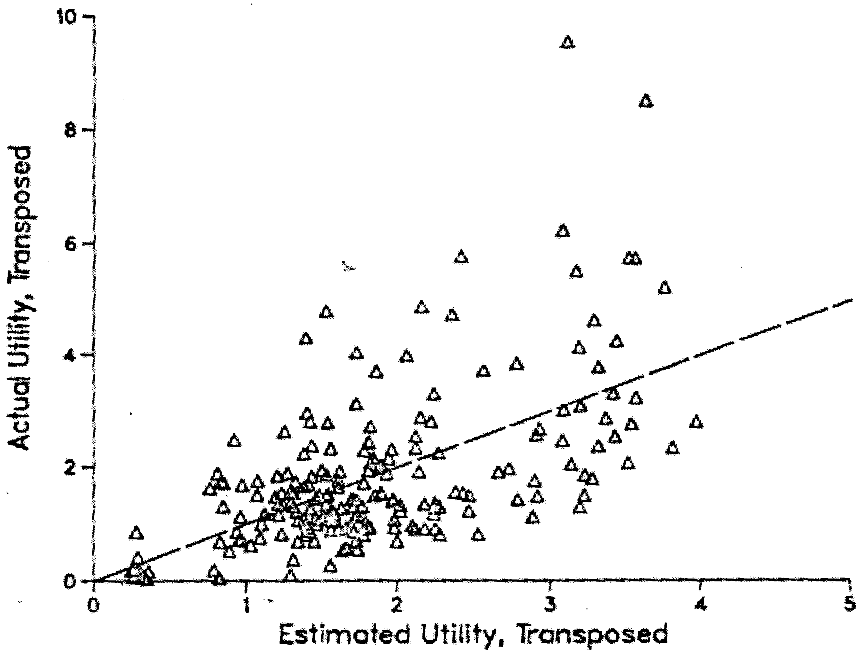
where Z_i denotes net square feet.

5. The Allocation

Costs of the 203 buildings used in the regression analysis were allocated to research based on the estimated use by research of lab, non-lab and parking space. First, ratios of the estimated coefficients of the average cost of lab to non-lab space, parking to non-lab space and unassignable to non-lab space were computed. These ratios were used as weights and non-lab space was given a weight of one. The square feet of the four room types were summed, each total was multiplied by its respective weight, and the cost was allocated to each of the four room types according

[7] A. C. Harvey, "Estimating Regression Models With Multiplicative Heteroscedasticity," Econometrica, Vol. 44, No. 3, 1976, pp. 461-465.

Figure 2. Estimated vs. Actual Utility, Corrected for Heteroscedasticity



to their fraction of the total weighted space, thus giving the initial allocation (see Table 1). This was only an initial allocation since the cost of unassignable space couldn't be attributed directly to any particular use. Therefore the cost initially allocated to unassignable space was reallocated to lab, non-lab and parking space according to their respective fractions of assignable space. The total allocation to lab, non-lab and parking space finally consisted of the initial allocation plus a fraction of the cost of unassignable space. To determine the allocation to organized research, the percent of non-lab space used for organized research as determined by the Office of Space Analysis' Annual Space Study was multiplied by the allocation to non-lab space, and likewise for lab and parking space. This procedure was followed separately for each of the 203 buildings, and then the allocated utilities costs were summed across the 203 buildings. The allocation to organized research of the costs of the 203 buildings was $4,301,426. The standard error was 1.1 percent of the allocation. The method for determining the standard error is described in a technical paper [8].

Costs of the forty-eight buildings excluded from the regression analysis were allocated in various ways. None of the utilities costs of the four buildings which were under

---

[8]R. L. Wright, "Measuring the Precision of Statistical Cost Allocations," Journal of Business and Economic Statistics, Vol. 1, No. 2, April 1983, pp. 93-100.

construction or renovation were allocated to organized research. Costs of the Botanical Gardens, Ferry Field, the Food and Chemical Stores, the Laundry, Radrick Farms, the Seismograph Station and the Sheep Facility were allocated by multiplying the cost of each building by the percent of space used by organized research. The cost of the North Campus Computing Center was allocated to organized research according to the recharge rates by which everyone in the University was charged with the services they used. To the extent that the University could identify the utilities costs of the thirty-six buildings which they leased or rented, these costs were allocated by multiplying the identifiable cost of each building by the percent of space used by organized research.

The total costs of the 48 buildings excluded from the regression analysis amounted to $2,189,573. Of this, $487,955 was allocated to organized research, bringing the total allocation to $4,789,381.

Table 1

ALLOCATION OF THE COST OF A BUILDING USED IN THE REGRESSION ANALYSIS

Room Type	Total Square Feet	Weight	Initial Allocation	Allocation of Cost of Unassignable Space	Total Allocation
			(1)	(2)	
Lab	L	4.9	$\frac{(\$)\ 4.9*L*C}{4.9*L+1.0*N+.2*P+3.9*U}$	$\frac{(\$)\ L*Y}{L+N+P}$	(1)+(2)
Non-Lab	N	1.0	$\frac{(\$)\ 1.0*N*C}{4.9*L+1.0*N+.2*P+3.9*U}$	$\frac{(\$)\ N*Y}{L+N+P}$	(1)+(2)
Parking	P	.2	$\frac{(\$)\ .2*P*C}{4.9*L+1.0*N+.2*P+3.9*U}$	$\frac{(\$)\ P*Y}{L+N+P}$	(1)+(2)
Unassignable	U	3.9	$\frac{(\$)\ 3.9*U*C}{4.9*L+1.0*N+.2*P+3.9*U}$	-($) Y	0

C = 1979-1980 utilities cost of this particular building
L+N+P+U = net square feet of this building
Y = allocation of this building's cost to unassignable space

APPENDIX A. DIFFERENT MODELS

In the technical paper [1], a different variable was used to correct for heteroscedasticity than was used in the actual allocation. Also, only 187 buildings were used in the paper. The variable used to correct for heteroscedasticity in the paper was the expected value of the total utilities cost of each building. This was because upon further reflection and after trying other choices for Z, it was found that expected total utilities cost produces a better adjustment for heteroscedasticity than does net square feet. However, since the second choice for Z changed the cost allocation only slightly, the net square feet model was retained in the allocation. If the costs of the 203 buildings were allocated using the expected cost of each building to correct for heteroscedasticity, the allocation to organized research would increase by $158 (see Table 2).

Three of the forty-eight buildings which were excluded from the analysis have been questioned by the federal auditors, so an allocation was performed which was identical in every respect to the actual allocation except for the inclusion of the Food and Chemical Stores, the Laundry and North Campus Computing Center. The allocation of the costs of the buildings used in the regression analysis based on these 206 buildings decreased the allocation to organized research of utilities cost by 1.5 percent, from $4,301,426 to $4,239,184 (see Table 2).

---

[1]Ibid, p. 99.

Table 2

ALLOCATION TO RESEARCH OF THE COSTS OF THE BUILDINGS USED IN THE REGRESSION ANALYSIS UNDER DIFFERENT MODELS

Number of Buildings	Heteroscedasticity Option	Total Allocation to Research	Standard Error of Total Allocation	Coefficient of Determination Based on Maximum Likelihood
187	expected cost of building	$4,301,540	$66,326	.9822
203*	net size of building	$4,301,426	$47,569	.9821
203	expected cost of building	$4,301,584	$63,576	.9869
206	net size of building	$4,239,184	$67,862	.9718

*This is the model which was actually used.

APPENDIX B. OUTPUT FROM PROGRAM

DEMONSTRATION OF STATISTICAL COST ALLOCATION
COST OF UTILITIES SPECIAL STUDY
THE UNIVERSITY OF MICHIGAN

PREPARED BY ROGER L. WRIGHT AND KAREN OBERG

APRIL 1, 1983

THIS COMPUTER RUN LISTS THE DATABASE FOR THE 203 BUILDINGS, AND SHOWS THE MODEL WHICH WAS USED IN THE ALLOCATION. THE OUTPUT INCLUDES THE UTILITY COSTS ALLOCATED TO ORGANIZED RESEARCH FOR EACH OF THE 203 BUILDINGS.

VARIABLES USED IN THE ANALYSIS:

V01 - 1.0 (USED TO INCLUDE AN INTERCEPT)

V02 - BUILDING NUMBER

V03 - UTILITY COST

V06 - NONLAB SPACE (IN SQUARE FEET)

V07 - LAB SPACE (IN SQUARE FEET)

V08 - PARKING SPACE (IN SQUARE FEET)

V09 - UNASSIGNABLE SPACE (IN SQUARE FEET)

V11 - NONLAB SPACE USED IN RESEARCH (SQ FT)

V12 - LAB SPACE USED IN RESEARCH (SQ FT)

V13 - PARKING SPACE USED IN RESEARCH (SQ FT)

V14 - UNASSIGNABLE SPACE USED IN RESEARCH (SQ FT)
V14 = [(V11+V12+V13)/(V06+V07+V08)]*V9

V15 - NET SIZE OF BUILDING (IN SQUARE FEET)

V16 - THE COST ALLOCATED TO EACH BUILDING, (CALCULATED BY THIS PROGRAM)

V99 - CASE NUMBER

READ IN THE DATABASE

SAMPLE DATA BASE PARAMETERS

```
MAXIMUM NUMBER OF CASES:      300
ACTUAL NUMBER OF CASES:       203
MAXIMUM NUMBER OF VARIABLES:   15
ACTUAL NUMBER OF VARIABLES:    11
LABELS:   2  3  6  7  8  9 11 12 13 14 15
FORMAT FOR INPUT: (11A8)
INPUT DEVICE:  1
```

LISTING OF THE ANALYSIS DATABASE

V99	V2	V3	V6	V7	V8	V9	V15
		ACTUAL	---------TOTAL SPACE---------				
CASE	BLDG	UTILITY COSTS	NNLAB	LAB	PRKG	UNAS	NET SIZE
1.	5.	4100.	2867.	0.	0.	18.	2885.
2.	40.	79427.	76538.	0.	0.	26059.	102597.
3.	51.	19241.	22489.	0.	0.	6178.	28667.
4.	52.	39093.	36667.	0.	0.	11554.	48221.
5.	53.	116530.	112691.	0.	0.	37931.	150622.
6.	54.	164493.	187664.	6163.	0.	67590.	261417.
7.	55.	11436.	12283.	0.	0.	3405.	15688.
8.	57.	8855.	6810.	0.	0.	2077.	8887.
9.	59.	100924.	117010.	0.	0.	37506.	154516.
10.	60.	216934.	174042.	0.	0.	61781.	235823.
11.	61.	75801.	75850.	0.	0.	37239.	113089.
12.	62.	15894.	20167.	0.	0.	5212.	25379.
13.	63.	262817.	232758.	615.	0.	71231.	304604.
14.	64.	48891.	81408.	0.	0.	32245.	113653.
15.	65.	59718.	24001.	851.	0.	14179.	39031.
16.	66.	167295.	192995.	235.	0.	64603.	257833.
17.	82.	145957.	124819.	0.	0.	20317.	145136.
18.	100.	119771.	18184.	9653.	0.	15080.	42917.
19.	101.	78353.	3205.	15198.	0.	13694.	32097.
20.	105.	7146.	9381.	0.	0.	1694.	11075.
21.	108.	131467.	93785.	0.	0.	25701.	119486.
22.	120.	399921.	175362.	0.	0.	90532.	265894.
23.	125.	222477.	7044.	26679.	0.	18503.	52226.
24.	136.	24555.	8022.	0.	0.	5022.	13044.
25.	145.	188584.	75233.	4718.	0.	40659.	120610.
26.	146.	51205.	15458.	0.	0.	9336.	24794.
27.	149.	96845.	35634.	0.	0.	32590.	68224.
28.	150.	76981.	69018.	6679.	0.	36514.	112211.
29.	151.	115476.	28567.	0.	0.	7191.	35758.
30.	152.	167847.	95722.	6317.	0.	45987.	148026.
31.	154.	71270.	40521.	0.	0.	23526.	64047.
32.	155.	17847.	6245.	2267.	0.	6888.	15400.
33.	156.	97856.	75339.	2698.	0.	47009.	125046.
34.	158.	384498.	37078.	97417.	0.	59609.	194104.
35.	159.	22328.	13612.	0.	0.	2151.	15763.
36.	162.	650393.	85747.	110427.	0.	101613.	297787.
37.	164.	20411.	12294.	0.	0.	6432.	18726.
38.	165.	151108.	40037.	23116.	0.	41726.	104879.
39.	166.	303915.	92636.	83850.	0.	64634.	241120.
40.	167.	154595.	71921.	33252.	0.	31463.	136636.
41.	168.	45376.	2133.	8103.	0.	3594.	13830.
42.	169.	21179.	14382.	0.	0.	1339.	15721.
43.	170.	145121.	91982.	17739.	0.	61075.	170796.
44.	171.	143038.	2279.	20773.	0.	13595.	36647.
45.	172.	194279.	69787.	0.	0.	45817.	115604.
46.	175.	41635.	27870.	0.	0.	14671.	42541.

47.	176.	67661.	37760.	2548.	0.	11056.	51364.
48.	177.	105052.	44669.	306.	0.	24095.	69070.
49.	178.	14785.	8611.	3374.	0.	1672.	13657.
50.	179.	117162.	51941.	1518.	0.	31285.	84744.
51.	180.	102820.	24443.	0.	0.	23499.	47942.
52.	181.	406616.	96632.	0.	0.	42754.	139386.
53.	182.	499478.	31455.	35526.	0.	43723.	110704.
54.	183.	12456.	11846.	0.	0.	5570.	17416.
55.	184.	186555.	73180.	0.	0.	18024.	91204.
56.	185.	203424.	138149.	0.	0.	24674.	162823.
57.	186.	6928.	3405.	0.	0.	3186.	6591.
58.	188.	176008.	35674.	64013.	0.	35653.	135340.
59.	189.	75929.	27644.	23699.	0.	20989.	72332.
60.	190.	605240.	55463.	92307.	0.	77194.	224964.
61.	191.	235925.	55350.	0.	0.	31916.	87266.
62.	193.	172511.	44904.	71741.	0.	43300.	159945.
63.	194.	8988.	7969.	1398.	0.	2949.	12316.
64.	195.	116589.	3332.	5262.	0.	7386.	15980.
65.	196.	31606.	24266.	0.	0.	12893.	37159.
66.	197.	95962.	47726.	3216.	0.	44757.	95699.
67.	198.	3328.	468.	1368.	0.	1362.	3198.
68.	200.	1179729.	48443.	105470.	0.	137537.	291450.
69.	201.	4577.	3263.	939.	0.	2946.	7148.
70.	202.	77825.	35238.	21770.	0.	11374.	68382.
71.	203.	15048.	11278.	0.	0.	0.	11278.
72.	204.	183885.	32095.	42989.	0.	27306.	102390.
73.	205.	4022.	3780.	0.	0.	1019.	4799.
74.	207.	213092.	56338.	12002.	0.	36032.	104372.
75.	208.	100699.	19455.	34988.	0.	19537.	73980.
76.	210.	89597.	1077.	8019.	0.	4620.	13716.
77.	211.	239906.	49185.	66563.	0.	47265.	163013.
78.	212.	30974.	2079.	6619.	0.	5349.	14047.
79.	214.	6777.	6696.	0.	0.	1911.	8607.
80.	215.	97146.	58841.	0.	0.	21988.	80829.
81.	216.	19956.	14426.	247.	0.	4784.	19457.
82.	221.	174523.	102071.	3903.	0.	53974.	159948.
83.	222.	6064.	4935.	0.	0.	1863.	6798.
84.	224.	6063.	3565.	0.	0.	1471.	5036.
85.	226.	231338.	107601.	13224.	0.	40831.	161656.
86.	227.	208669.	97398.	0.	0.	23761.	121159.
87.	230.	110548.	5344.	11182.	0.	15346.	31872.
88.	234.	505024.	73703.	17990.	0.	56974.	148667.
89.	235.	10665.	10510.	0.	0.	3442.	13952.
90.	253.	20700.	525.	0.	141026.	2236.	143787.
91.	254.	59368.	0.	0.	360829.	9202.	370031.
92.	255.	10783.	85.	0.	221767.	4248.	226100.
93.	257.	25370.	4012.	4188.	203720.	3959.	215879.
94.	258.	7029.	80.	0.	129376.	5326.	134782.
95.	259.	24860.	245.	0.	155905.	3399.	159549.
96.	261.	19242.	5671.	0.	6173.	1450.	13294.
97.	269.	66.	522.	0.	1312.	0.	1834.
98.	279.	4093.	2269.	0.	0.	742.	3011.
99.	313.	141175.	25964.	0.	0.	21737.	47701.
100.	400.	102883.	7354.	18915.	0.	28408.	54677.
101.	402.	24961.	13827.	0.	0.	4005.	17832.
102.	403.	50359.	13048.	7019.	0.	15142.	35209.
103.	404.	108467.	6990.	19357.	0.	17696.	44043.
104.	406.	23612.	24030.	0.	0.	2611.	26641.
105.	407.	240222.	16179.	89870.	0.	36379.	142428.
106.	409.	12765.	9097.	0.	4032.	1717.	14846.

107.	414.	57607.	6374.	15059.	0.	10956.	32389.
108.	415.	40051.	11181.	11651.	0.	13134.	35966.
109.	418.	17341.	12813.	0.	0.	612.	13425.
110.	421.	58084.	1406.	23014.	0.	3150.	27570.
111.	424.	31273.	10992.	1626.	0.	4177.	16795.
112.	427.	119623.	43983.	0.	0.	10686.	54669.
113.	428.	28804.	4774.	6607.	0.	3218.	14599.
114.	429.	49746.	20716.	0.	0.	7729.	28445.
115.	430.	17109.	44051.	0.	0.	189.	44240.
116.	432.	288747.	37789.	96809.	0.	60908.	195506.
117.	435.	226600.	30085.	22717.	0.	27727.	80529.
118.	438.	1554.	713.	0.	0.	42.	755.
119.	439.	41315.	24660.	0.	0.	4495.	29155.
120.	440.	187795.	57025.	7089.	0.	37636.	101750.
121.	441.	149884.	12269.	21436.	0.	15139.	48844.
122.	442.	91565.	26371.	0.	0.	12984.	39355.
123.	443.	19329.	21538.	665.	0.	15587.	37790.
124.	444.	131731.	22007.	22251.	0.	22901.	67159.
125.	445.	8718.	10817.	721.	0.	4690.	16228.
126.	449.	3531.	6677.	0.	0.	576.	7253.
127.	450.	58379.	51390.	0.	0.	7988.	59378.
128.	457.	165467.	143556.	0.	0.	9023.	152579.
129.	498.	152920.	140653.	0.	0.	17439.	158092.
130.	510.	108482.	91948.	0.	0.	30533.	122481.
131.	515.	104159.	93333.	0.	0.	31608.	124941.
132.	554.	1111.	668.	0.	0.	185.	853.
133.	555.	200859.	216496.	102.	0.	85604.	302202.
134.	601.	279139.	467120.	0.	0.	0.	467120.
135.	700.	168580.	99340.	0.	0.	64683.	164023.
136.	703.	35290.	63666.	0.	0.	1892.	65558.
137.	705.	26574.	12284.	3176.	0.	2555.	18015.
138.	709.	81087.	53443.	0.	0.	23507.	76950.
139.	710.	28644.	28708.	0.	0.	4377.	33085.
140.	711.	10620.	7512.	0.	0.	7710.	15222.
141.	712.	110.	679.	0.	0.	0.	679.
142.	719.	126870.	69364.	0.	0.	20604.	89968.
143.	720.	2280.	1820.	0.	0.	504.	2324.
144.	726.	446.	1147.	0.	0.	318.	1465.
145.	728.	6221.	4292.	0.	0.	841.	5133.
146.	730.	66.	591.	0.	0.	163.	754.
147.	731.	16136.	11894.	0.	0.	6357.	18251.
148.	732.	44685.	11820.	0.	0.	14463.	26283.
149.	733.	338.	6225.	0.	3964.	956.	11145.
150.	735.	8980.	15539.	0.	0.	6872.	22411.
151.	800.	188256.	130822.	0.	0.	27604.	158426.
152.	807.	5092.	5822.	0.	0.	714.	6536.
153.	808.	20776.	1152.	0.	30702.	619.	32473.
154.	813.	15251.	6843.	0.	0.	1923.	8766.
155.	814.	2814.	2276.	0.	0.	1585.	3861.
156.	815.	281926.	61063.	0.	0.	19484.	80547.
157.	816.	5417.	3582.	0.	1281.	382.	5245.
158.	818.	362.	260.	0.	0.	0.	260.
159.	819.	1949.	1219.	0.	0.	337.	1556.
160.	825.	4884.	2575.	0.	0.	2476.	5051.
161.	830.	92022.	55681.	7126.	0.	27338.	90145.
162.	831.	61596.	30320.	0.	0.	34413.	64733.
163.	832.	19394.	14983.	0.	0.	2241.	17224.
164.	834.	9616.	3816.	0.	0.	811.	4627.
165.	838.	2475.	2351.	0.	0.	1531.	3882.
166.	845.	25758.	27198.	699.	0.	12207.	40104.

167.	847.	716.	2304.	0.	0.	1021.	3325.
168.	850.	2030.	2996.	100.	0.	1661.	4757.
169.	851.	21233.	20915.	0.	0.	5233.	26148.
170.	853.	133.	2964.	0.	0.	0.	2964.
171.	857.	2440.	4559.	0.	0.	0.	4559.
172.	858.	20074.	10535.	3477.	0.	5600.	19612.
173.	861.	11904.	9608.	0.	0.	2181.	11789.
174.	874.	2561.	3867.	0.	0.	466.	4333.
175.	875.	2890.	1767.	1364.	0.	1306.	4437.
176.	886.	10588.	6876.	0.	0.	3676.	10552.
177.	890.	37998.	20222.	4700.	0.	14931.	39853.
178.	897.	8804.	5886.	0.	0.	5909.	11795.
179.	898.	22128.	14436.	0.	0.	5207.	19643.
180.	899.	2469.	3377.	0.	0.	1175.	4552.
181.	976.	1678.	2374.	0.	0.	656.	3030.
182.	1001.	682141.	252458.	54988.	256035.	120921.	684402.
183.	1450.	2905.	1328.	1245.	0.	392.	2965.
184.	1501.	60983.	39196.	0.	0.	18173.	57369.
185.	1601.	661302.	263194.	28354.	105956.	158765.	556269.
186.	2080.	112304.	42827.	20810.	0.	76325.	139962.
187.	2701.	251385.	511112.	0.	0.	0.	511112.
188.	302.	112817.	45612.	37.	0.	25642.	71291.
189.	304.	41622.	14413.	0.	0.	5274.	19687.
190.	305.	16014.	1331.	2457.	0.	3001.	6789.
191.	306.	1717878.	304523.	61061.	0.	142639.	508223.
192.	307.	78583.	39985.	987.	0.	14259.	55231.
193.	308.	189489.	62204.	4676.	0.	45221.	112101.
194.	309.	129011.	26942.	4994.	0.	20995.	52931.
195.	310.	20531.	16405.	436.	0.	7614.	24455.
196.	312.	656875.	74363.	15386.	0.	69285.	159034.
197.	314.	74746.	3404.	5864.	0.	8550.	17818.
198.	319.	40546.	21669.	0.	0.	7473.	29142.
199.	320.	127918.	112.	0.	432963.	11003.	444078.
200.	325.	46364.	17528.	426.	0.	9037.	26991.
201.	326.	14639.	13284.	0.	0.	4124.	17408.
202.	327.	5011.	3525.	1017.	0.	2514.	7056.
203.	328.	14381.	9438.	113.	0.	11219.	20770.

SUMMARY STATISTICS N = 203

VARIABLE	MINIMUM	MAXIMUM	MEAN	STD DEV
3	65.65	0.1718E+07	0.1106E+06	0.1872E+06
6	0.0	0.5111E+06	0.4310E+05	0.6856E+05
7	0.0	0.1104E+06	8480.	0.2065E+05
8	0.0	0.4330E+06	0.1012E+05	0.5092E+05
9	0.0	0.1588E+06	0.2016E+05	0.2675E+05
11	0.0	0.7209E+05	2106.	6266.
15	260.0	0.6844E+06	0.8186E+05	0.1080E+06

MODEL: STANDARD DEVIATION PROPORTIONAL TO NET SIZE RAISED TO THE POWER .972

"ALLOCATED COST" INDICATES COST ALLOCATED TO ORGANIZED RESEARCH

STARTING GAMMAS

VARIABLE	GAMMA
15	0.972000000000

ESTIMATED MODEL FOR VARIABLE 3
ASSUMING THE INITIAL VALUES OF GAMMA

ASSUMED SECONDARY EQUATION FOR HETEROSCEDASTICITY

VARIABLE	GAMMA
15	0.9720

SIGMA 0 = 1.146

PRIMARY EQUATION USING BLU REGRESSION

VARIABLE	BETA	ST ERROR	T-STAT	SIGNIF
6	0.6544	0.1548	4.23	0.0000
7	3.190	0.3854	8.28	0.0000
8	0.1414	0.2808	0.50	0.6152
9	2.566	0.3845	6.67	0.0000

RSQ-WLS = 0.7616 RSQ-MLE = 0.9821

F-STAT = 158.90 SIGNIF = 0.0000

CHI SQ = 816.22 SAMPLE N = 203

THE ALLOCATED COST IS 4301426.08

THE STANDARD ERROR IS 47568.81

THE ALLOCATION VALUE HAS BEEN SAVED
THIS VARIABLE HAS THE LABEL 16

LISTING OF THE ALLOCATION DATABASE

V99	V2	V3	V11	V12	V13	V14	V16
		ACTUAL UTILITY	------RESEARCH SPACE---------				ALLOC
CASE	BLDG	COSTS	NNLAB	LAB	PRKG	UNAS	COST
1.	5.	4100.	0.	0.	0.	0.	0.
2.	40.	79427.	0.	0.	0.	0.	0.
3.	51.	19241.	0.	0.	0.	0.	0.
4.	52.	39093.	0.	0.	0.	0.	0.
5.	53.	116530.	0.	0.	0.	0.	0.
6.	54.	164493.	0.	0.	0.	0.	0.
7.	55.	11436.	0.	0.	0.	0.	0.
8.	57.	8855.	0.	0.	0.	0.	0.
9.	59.	100924.	0.	0.	0.	0.	0.
10.	60.	216934.	0.	0.	0.	0.	0.
11.	61.	75801.	0.	0.	0.	0.	0.
12.	62.	15894.	0.	0.	0.	0.	0.
13.	63.	262817.	0.	0.	0.	0.	0.
14.	64.	48891.	0.	0.	0.	0.	0.
15.	65.	59718.	7312.	413.	0.	4407.	18973.
16.	66.	167295.	1037.	59.	0.	366.	1032.
17.	82.	145957.	0.	0.	0.	0.	0.
18.	100.	119771.	12218.	9084.	0.	11540.	97987.
19.	101.	78353.	3205.	15198.	0.	13694.	78353.
20.	105.	7146.	0.	0.	0.	0.	0.
21.	108.	131467.	0.	0.	0.	0.	0.
22.	120.	399921.	0.	0.	0.	0.	0.
23.	125.	222477.	4877.	26589.	0.	17265.	214559.
24.	136.	24555.	1632.	0.	0.	1022.	4995.
25.	145.	188584.	72085.	4718.	0.	39058.	181685.
26.	146.	51205.	739.	0.	0.	446.	2448.
27.	149.	96845.	0.	0.	0.	0.	0.
28.	150.	76981.	90.	0.	0.	43.	82.
29.	151.	115476.	0.	0.	0.	0.	0.
30.	152.	167847.	1820.	0.	0.	820.	2755.
31.	154.	71270.	8181.	0.	0.	4750.	14389.
32.	155.	17847.	0.	0.	0.	0.	0.
33.	156.	97856.	2124.	912.	0.	1829.	4929.
34.	158.	384498.	130.	2447.	0.	1142.	8527.
35.	159.	22328.	0.	0.	0.	0.	0.
36.	162.	650393.	5122.	16524.	0.	11212.	82461.
37.	164.	20411.	792.	0.	0.	414.	1315.
38.	165.	151108.	5343.	7096.	0.	8219.	34470.
39.	166.	303915.	14271.	25041.	0.	14397.	77625.
40.	167.	154595.	4771.	15746.	0.	6138.	45679.
41.	168.	45376.	0.	2284.	0.	802.	11627.
42.	169.	21179.	0.	0.	0.	0.	0.
43.	170.	145121.	2689.	820.	0.	1953.	4981.
44.	171.	143038.	1806.	17246.	0.	11236.	118492.
45.	172.	194279.	1294.	0.	0.	850.	3602.
46.	175.	41635.	213.	0.	0.	112.	318.

47.	176.	67661.	0.	0.	0.	0.	0.
48.	177.	105052.	0.	0.	0.	0.	0.
49.	178.	14785.	1298.	500.	0.	251.	2207.
50.	179.	117162.	2222.	0.	0.	1300.	4712.
51.	180.	102820.	0.	0.	0.	0.	0.
52.	181.	406616.	0.	0.	0.	0.	0.
53.	182.	499478.	4170.	28260.	0.	21169.	298749.
54.	183.	12456.	1645.	0.	0.	773.	1730.
55.	184.	186555.	1397.	0.	0.	344.	3561.
56.	185.	203424.	0.	0.	0.	0.	0.
57.	186.	6928.	0.	0.	0.	0.	0.
58.	188.	176008.	2838.	9378.	0.	4369.	23714.
59.	189.	75929.	892.	5200.	0.	2490.	12126.
60.	190.	605240.	7107.	49849.	0.	29753.	274697.
61.	191.	235925.	0.	0.	0.	0.	0.
62.	193.	172511.	7369.	66019.	0.	27242.	133270.
63.	194.	8988.	1215.	374.	0.	500.	1706.
64.	195.	116589.	2489.	3844.	0.	5443.	85653.
65.	196.	31606.	0.	0.	0.	0.	0.
66.	197.	95962.	4630.	1138.	0.	5068.	12070.
67.	198.	3328.	0.	0.	0.	0.	0.
68.	200.	1179729.	9970.	65815.	0.	67722.	638482.
69.	201.	4577.	719.	383.	0.	773.	1325.
70.	202.	77825.	6668.	18486.	0.	5019.	48741.
71.	203.	15048.	0.	0.	0.	0.	0.
72.	204.	183885.	9085.	34347.	0.	15795.	125739.
73.	205.	4022.	42.	0.	0.	11.	45.
74.	207.	213092.	98.	0.	0.	52.	250.
75.	208.	100699.	10056.	32869.	0.	15404.	87127.
76.	210.	89597.	817.	6758.	0.	3847.	75091.
77.	211.	239906.	21797.	47278.	0.	28206.	155733.
78.	212.	30974.	1040.	6619.	0.	4710.	28990.
79.	214.	6777.	180.	0.	0.	51.	182.
80.	215.	97146.	0.	0.	0.	0.	0.
81.	216.	19956.	0.	0.	0.	0.	0.
82.	221.	174523.	14383.	218.	0.	7436.	23396.
83.	222.	6064.	0.	0.	0.	0.	0.
84.	224.	6063.	3565.	0.	0.	1471.	6063.
85.	226.	231338.	618.	2884.	0.	1183.	13454.
86.	227.	208669.	0.	0.	0.	0.	0.
87.	230.	110548.	1227.	10502.	0.	10892.	87616.
88.	234.	505024.	16826.	12621.	0.	18297.	196993.
89.	235.	10665.	9370.	0.	0.	3069.	9508.
90.	253.	20700.	0.	0.	0.	0.	0.
91.	254.	59368.	0.	0.	0.	0.	0.
92.	255.	10783.	0.	0.	0.	0.	0.
93.	257.	25370.	0.	0.	0.	0.	0.
94.	258.	7029.	0.	0.	0.	0.	0.
95.	259.	24860.	0.	0.	0.	0.	0.
96.	261.	19242.	0.	0.	0.	0.	0.
97.	269.	66.	522.	0.	1312.	0.	66.
98.	279.	4093.	0.	0.	0.	0.	0.
99.	313.	141175.	1908.	0.	0.	1597.	10374.
100.	400.	102883.	2825.	7034.	0.	10662.	38490.
101.	402.	24961.	0.	0.	0.	0.	0.
102.	403.	50359.	3223.	5816.	0.	6821.	27542.
103.	404.	108467.	2718.	17681.	0.	13701.	90612.
104.	406.	23612.	0.	0.	0.	0.	0.
105.	407.	240222.	5230.	44350.	0.	17008.	115948.
106.	409.	12765.	0.	0.	0.	0.	0.

107.	414.	57607.	563.	4386.	0.	2530.	14955.
108.	415.	40051.	2837.	317.	0.	1814.	3854.
109.	418.	17341.	0.	0.	0.	0.	0.
110.	421.	58084.	1406.	19868.	0.	2744.	50277.
111.	424.	31273.	2012.	407.	0.	801.	6322.
112.	427.	119623.	0.	0.	0.	0.	0.
113.	428.	28804.	2515.	5548.	0.	2280.	22358.
114.	429.	49746.	0.	0.	0.	0.	0.
115.	430.	17109.	8019.	0.	0.	34.	3114.
116.	432.	288747.	2307.	1924.	0.	1915.	7404.
117.	435.	226600.	15617.	22717.	0.	20130.	186413.
118.	438.	1554.	0.	0.	0.	0.	0.
119.	439.	41315.	0.	0.	0.	0.	0.
120.	440.	187795.	0.	0.	0.	0.	0.
121.	441.	149884.	9030.	16961.	0.	11674.	117002.
122.	442.	91565.	0.	0.	0.	0.	0.
123.	443.	19329.	0.	0.	0.	0.	0.
124.	444.	131731.	20401.	22251.	0.	22070.	128822.
125.	445.	8718.	0.	0.	0.	0.	0.
126.	449.	3531.	0.	0.	0.	0.	0.
127.	450.	58379.	0.	0.	0.	0.	0.
128.	457.	165467.	0.	0.	0.	0.	0.
129.	498.	152920.	0.	0.	0.	0.	0.
130.	510.	108482.	0.	0.	0.	0.	0.
131.	515.	104159.	0.	0.	0.	0.	0.
132.	554.	1111.	0.	0.	0.	0.	0.
133.	555.	200859.	0.	0.	0.	0.	0.
134.	601.	279139.	0.	0.	0.	0.	0.
135.	700.	168580.	0.	0.	0.	0.	0.
136.	703.	35290.	0.	0.	0.	0.	0.
137.	705.	26574.	0.	0.	0.	0.	0.
138.	709.	81087.	0.	0.	0.	0.	0.
139.	710.	28644.	0.	0.	0.	0.	0.
140.	711.	10620.	0.	0.	0.	0.	0.
141.	712.	110.	0.	0.	0.	0.	0.
142.	719.	126870.	0.	0.	0.	0.	0.
143.	720.	2280.	0.	0.	0.	0.	0.
144.	726.	446.	0.	0.	0.	0.	0.
145.	728.	6221.	0.	0.	0.	0.	0.
146.	730.	66.	0.	0.	0.	0.	0.
147.	731.	16136.	0.	0.	0.	0.	0.
148.	732.	44685.	0.	0.	0.	0.	0.
149.	733.	338.	0.	0.	0.	0.	0.
150.	735.	8980.	0.	0.	0.	0.	0.
151.	800.	188256.	0.	0.	0.	0.	0.
152.	807.	5092.	0.	0.	0.	0.	0.
153.	808.	20776.	0.	0.	0.	0.	0.
154.	813.	15251.	0.	0.	0.	0.	0.
155.	814.	2814.	0.	0.	0.	0.	0.
156.	815.	281926.	0.	0.	0.	0.	0.
157.	816.	5417.	0.	0.	0.	0.	0.
158.	818.	362.	0.	0.	0.	0.	0.
159.	819.	1949.	1219.	0.	0.	337.	1949.
160.	825.	4884.	0.	0.	0.	0.	0.
161.	830.	92022.	3945.	7126.	0.	4819.	26813.
162.	831.	61596.	1120.	0.	0.	1271.	2275.
163.	832.	19394.	0.	0.	0.	0.	0.
164.	834.	9616.	3146.	0.	0.	669.	7928.
165.	838.	2475.	0.	0.	0.	0.	0.
166.	845.	25758.	22266.	643.	0.	10024.	21240.

167.	847.	716.	576.	0.	0.	255.	179.
168.	850.	2030.	885.	100.	0.	528.	699.
169.	851.	21233.	0.	0.	0.	0.	0.
170.	853.	133.	0.	0.	0.	0.	0.
171.	857.	2440.	0.	0.	0.	0.	0.
172.	858.	20074.	0.	0.	0.	0.	0.
173.	861.	11904.	0.	0.	0.	0.	0.
174.	874.	2561.	0.	0.	0.	0.	0.
175.	875.	2890.	0.	62.	0.	26.	86.
176.	886.	10588.	0.	0.	0.	0.	0.
177.	890.	37998.	6229.	4660.	0.	6524.	20377.
178.	897.	8804.	4690.	0.	0.	4708.	7015.
179.	898.	22128.	0.	0.	0.	0.	0.
180.	899.	2469.	0.	0.	0.	0.	0.
181.	976.	1678.	0.	0.	0.	0.	0.
182.	1001.	682141.	0.	0.	0.	0.	0.
183.	1450.	2905.	664.	623.	0.	196.	1454.
184.	1501.	60983.	0.	0.	0.	0.	0.
185.	1601.	661302.	1931.	751.	0.	1071.	6186.
186.	2080.	112304.	4207.	4666.	0.	10642.	17390.
187.	2701.	251385.	0.	0.	0.	0.	0.
188.	302.	112817.	0.	0.	0.	0.	0.
189.	304.	41622.	0.	0.	0.	0.	0.
190.	305.	16014.	0.	0.	0.	0.	0.
191.	306.	1717878.	0.	0.	0.	0.	0.
192.	307.	78583.	0.	0.	0.	0.	0.
193.	308.	189489.	0.	0.	0.	0.	0.
194.	309.	129011.	0.	0.	0.	0.	0.
195.	310.	20531.	0.	0.	0.	0.	0.
196.	312.	656875.	0.	0.	0.	0.	0.
197.	314.	74746.	0.	0.	0.	0.	0.
198.	319.	40546.	0.	0.	0.	0.	0.
199.	320.	127918.	0.	0.	0.	0.	0.
200.	325.	46364.	0.	0.	0.	0.	0.
201.	326.	14639.	0.	0.	0.	0.	0.
202.	327.	5011.	0.	0.	0.	0.	0.
203.	328.	14381.	43.	0.	0.	51.	64.

SUMMARY STATISTICS N = 203

VARIABLE	MINIMUM	MAXIMUM	MEAN	STD DEV
3	65.65	0.1718E+07	0.1106E+06	0.1872E+06
11	0.0	0.7209E+05	2106.	6266.
12	0.0	0.6602E+05	3623.	0.1027E+05
13	0.0	1312.	6.463	91.86
14	0.0	0.6772E+05	2913.	7577.
16	0.0	0.6385E+06	0.2119E+05	0.6399E+05

STATISTICAL COST ALLOCATION
APRIL 27, 1983 VERSION

PREPARED FOR THE UNIVERSITY OF MICHIGAN

BY ROGER L. WRIGHT
ASSOCIATE PROFESSOR OF STATISTICS
GRADUATE SCHOOL OF BUSINESS ADMINISTRATION
THE UNIVERSITY OF MICHIGAN

APPENDIX C. SAMPLE RUN OF PROGRAM

THE FOLLOWING IS A SAMPLE RUN OF THE PROGRAM, WHICH IS PROVIDED TO ILLUSTRATE HOW THE PROGRAM IS USED. THE PROGRAM IS INTERACTIVE, WITH USER RESPONSES OCCURING ON EVERY LINE DIRECTLY FOLLOWING THE PROMPT "& ?". EXTRA COMMENTS HAVE BEEN INSERTED TO HELP EXPLAIN CERTAIN POINTS. THESE COMMENTS ARE ON LINES BEGINNING WITH "***". ALL LINES OTHER THAN THE TWO TYPES JUST DESCRIBED ARE LINES WHICH HAVE BEEN PRINTED BY THE PROGRAM ITSELF.

```
#RUN SRY6:ALLOCATE 1=SRY6:DATA 7=*SINK* 8=-TRACE
#Execution begins

      STATISTICAL COST ALLOCATION



 CURRENT PROGRAM LIMITS:
 NUMBER OF VARIABLES IN SAMPLE DATA BASE:        15
 NUMBER OF CASES IN SAMPLE DATA BASE:           300

 CORRECT? (00=YES,01=NO)
& ?
 00

 ENTER ANALYSIS OPTION:
   01 - ENTER A COMMENT
   02 - READ IN A SAMPLE DATA BASE
   03 - SELECT CASES FOR ANALYSIS
   04 - WRITE OUT DATA
   05 - DESCRIBE THE DATA
   06 - ESTIMATE THE MODEL
   10 - STOP
& ?
*** A COMMENT MAY BE ENTERED BY THE USER TO DOCUMENT THE RUN.
 01

 ANALYSIS OPTION:   1

 CORRECT? (00=YES,01=NO)
& ?
 00

 NUMBER OF LINES (01 TO 10)
& ?
 06

 LINE   1
& ?
 1

 LINE   2
& ?
 -      SESSION TO DEMONSTRATE THE USE OF THE ALLOCATION PROGRAM

 LINE   3
```

```
& ?
 -                        AUGUST 1, 1983

 LINE  4
& ?
 -

 LINE  5
& ?
 THIS SESSION USES 203 BUILDINGS IN THE REGRESSION ANALYSIS AND USES THE

 LINE  6
& ?
 MODEL SIGMA=(SIGMA O)*(NET SIZE**.972) TO CORRECT FOR HETEROSCEDASTICITY.
```

```
       SESSION TO DEMONSTRATE THE USE OF THE ALLOCATION PROGRAM

                           AUGUST 1, 1983


 THIS SESSION USES 203 BUILDINGS IN THE REGRESSION ANALYSIS AND USES THE
 MODEL SIGMA=(SIGMA O)*(NET SIZE**.972) TO CORRECT FOR HETEROSCEDASTICITY.

 CORRECT? (OO=YES,O1=NO)
& ?
 OO
```

```
        SESSION TO DEMONSTRATE THE USE OF THE ALLOCATION PROGRAM

                          AUGUST 1, 1983


 THIS SESSION USES 203 BUILDINGS IN THE REGRESSION ANALYSIS AND USES THE
 MODEL SIGMA=(SIGMA O)*(NET SIZE**.972) TO CORRECT FOR HETEROSCEDASTICITY.

 ENTER ANALYSIS OPTION:
   01 - ENTER A COMMENT
   02 - READ IN A SAMPLE DATA BASE
   03 - SELECT CASES FOR ANALYSIS
   04 - WRITE OUT DATA
   05 - DESCRIBE THE DATA
   06 - ESTIMATE THE MODEL
   10 - STOP
& ?
*** THE FIRST STEP IS TO READ IN THE DATABASE.
 02

 ANALYSIS OPTION:  2

 CORRECT? (00=YES,01=NO)
& ?
 00


 SAMPLE DATA BASE PARAMETERS

 ENTER DATA BASE PARAMETERS

 ACTUAL NUMBER OF CASES  E.G. 210.
& ?
 203.

 ENTER LABELS FOR VARIABLES  E.G. 02 03 04
 UP TO 20 LABELS PER LINE

 LABELS
& ?
*** LABELS MUST BE GIVEN TO ALL OF THE VARIABLES WHICH ARE GOING TO BE USED- THIS
*** INCLUDES THE VARIABLES USED IN THE PRIMARY REGRESSION EQUATION, THE VARIABLES
*** WHICH ALLOCATE THE COST, AND THE VARIABLE USED TO CORRECT FOR HETEROSCEDASTICITY.
 02 03 06 07 08 09 11 12 13 14 15

 DATA FORMAT FOR INPUT  E.G. (1F10.3,1F5.0)
& ?
 (11A8)

 INPUT DEVICE  E.G. 01
& ?
 01

 MAXIMUM NUMBER OF CASES:       300
 ACTUAL NUMBER OF CASES:        203
```

```
 MAXIMUM NUMBER OF VARIABLES:    15
 ACTUAL NUMBER OF VARIABLES:     11
 LABELS:    2  3  6  7  8  9 11 12 13 14 15
 FORMAT FOR INPUT: (11A8)
 INPUT DEVICE:  1

 CORRECT? (OO=YES,O1=NO)
& ?
*** ENTERING "O1" AT THIS POINT WOULD ALLOW THE USER TO RE-ENTER ALL OF THE
*** ABOVE INFORMATION, IF ANY OF IT HAS BEEN MISSPECIFIED.
 OO

  2O3 CASES HAVE BEEN READ

 ENTER ANALYSIS OPTION:
   O1 - ENTER A COMMENT
   O2 - READ IN A SAMPLE DATA BASE
   O3 - SELECT CASES FOR ANALYSIS
   O4 - WRITE OUT DATA
   O5 - DESCRIBE THE DATA
   O6 - ESTIMATE THE MODEL
   1O - STOP
& ?
*** ANALYSIS OPTION 4 ENABLES THE USER TO VERIFY THAT THE DATA HAS BEEN READ
*** IN CORRECTLY.
 O4

 ANALYSIS OPTION:  4

 CORRECT? (OO=YES,O1=NO)
& ?
 OO

 ENTER VARIABLES TO BE WRITTEN OUT

 VARIABLES
& ?
*** ANY SUBSET OF THE VARIABLES MAY BE DISPLAYED.
 O2 O3 O6 O7 O8 O9 15

 NUMBER OF CASES TO BE WRITTEN OUT  E.G. 1O.
& ?
*** ALL OF THE CASES MAY BE WRITTEN OUT IF DESIRED.
 2O.

 DATA FORMAT FOR OUTPUT  E.G. (2F1O.2)
& ?
 (1F7.O,1F9.O,4F8.O,1F8.O)

 OUTPUT DEVICE E.G. O6
& ?
*** DEVICE 7 WAS SPECIFIED AS THE USER'S TERMINAL IN THE RUN COMMAND.
 O7

 WRITE OUT
 VARIABLES     2  3  6  7  8  9 15
 FOR THE FIRST   2O OF THE  2O3 SELECTED CASES
 USING THE FORMAT (1F7.O,1F9.O,4F8.O,1F8.O)
 ON THE DEVICE  7
```

```
 CORRECT? (OO=YES,O1=NO)
& ?
 OO
     5.    4100.    2867.       O.       O.      18.    2885.
    40.   79427.   76538.       O.       O.   26059.  102597.
    51.   19241.   22489.       O.       O.    6178.   28667.
    52.   39093.   36667.       O.       O.   11554.   48221.
    53.  116530.  112691.       O.       O.   37931.  150622.
    54.  164493.  187664.    6163.       O.   67590.  261417.
    55.   11436.   12283.       O.       O.    3405.   15688.
    57.    8855.    6810.       O.       O.    2077.    8887.
    59.  100924.  117010.       O.       O.   37506.  154516.
    60.  216934.  174042.       O.       O.   61781.  235823.
    61.   75801.   75850.       O.       O.   37239.  113089.
    62.   15894.   20167.       O.       O.    5212.   25379.
    63.  262817.  232758.     615.       O.   71231.  304604.
    64.   48891.   81408.       O.       O.   32245.  113653.
    65.   59718.   24001.     851.       O.   14179.   39031.
    66.  167295.  192995.     235.       O.   64603.  257833.
    82.  145957.  124819.       O.       O.   20317.  145136.
   100.  119771.   18184.    9653.       O.   15080.   42917.
   101.   78353.    3205.   15198.       O.   13694.   32097.
   105.    7146.    9381.       O.       O.    1694.   11075.

   20 CASES HAVE BEEN WRITTEN

 ENTER ANALYSIS OPTION:
   O1 - ENTER A COMMENT
   O2 - READ IN A SAMPLE DATA BASE
   O3 - SELECT CASES FOR ANALYSIS
   O4 - WRITE OUT DATA
   O5 - DESCRIBE THE DATA
   O6 - ESTIMATE THE MODEL
   10 - STOP
& ?
*** ANALYSIS OPTION 5 ALLOWS THE USER TO HAVE SOME DESCRIPTIVE STATISTICS
*** CALCULATED FOR ANY OF THE VARIABLES.
 O5

 ANALYSIS OPTION:  5

 CORRECT? (OO=YES,O1=NO)
& ?
 OO

 ENTER DESCRIBE OPTION:
   O1 - SUMMARY STATISTICS
   O2 - FREQUENCY TABLE
   O3 - NEW ANALYSIS OPTION
& ?
 O1

 DESCRIBE OPTION:   1

 CORRECT? (OO=YES,O1=NO)
& ?
 OO

 ENTER VARIABLES TO BE DESCRIBED
```

```
 VARIABLES
& ?
 03 06 07 08 09

 VARIABLES
  3  6  7  8  9

 CORRECT? (00=YES,01=NO)
& ?
 00


 SUMMARY STATISTICS       N =  203

 VARIABLE  MINIMUM       MAXIMUM         MEAN       STD DEV
     3        65.65       0.1718E+07   0.1106E+06   0.1872E+06
     6       0.0          0.5111E+06   0.4310E+05   0.6856E+05
     7       0.0          0.1104E+06    8480.       0.2065E+05
     8       0.0          0.4330E+06   0.1012E+05   0.5092E+05
     9       0.0          0.1588E+06   0.2016E+05   0.2675E+05

 ENTER DESCRIBE OPTION:
   01 - SUMMARY STATISTICS
   02 - FREQUENCY TABLE
   03 - NEW ANALYSIS OPTION
& ?
 03

 DESCRIBE OPTION:   3

 CORRECT? (00=YES,01=NO)
& ?
 00

 ENTER ANALYSIS OPTION:
   01 - ENTER A COMMENT
   02 - READ IN A SAMPLE DATA BASE
   03 - SELECT CASES FOR ANALYSIS
   04 - WRITE OUT DATA
   05 - DESCRIBE THE DATA
   06 - ESTIMATE THE MODEL
   10 - STOP
& ?
 06

 ANALYSIS OPTION:  6

 CORRECT? (00=YES,01=NO)
& ?
 00

 ENTER HETEROSCEDASTICITY OPTION:
   01 - EXPONENTIAL MODEL FOR SIGMA
   02 - SIGMA PROPORTIONAL TO THE EXPECTED VALUE
& ?
*** IN THE EXPONENTIAL MODEL FOR SIGMA, SIGMA IS ASSUMED TO BE PROPORTIONAL
*** TO A VARIABLE RAISED TO THE "GAMMA" POWER.
 01

 HETEROSCEDASTICITY OPTION:  1
```

```
 CORRECT? (OO=YES,O1=NO)
& ?
 OO

 SPECIFY VARIABLES

 DEPENDENT VARIABLE
& ?
*** THE DEPENDENT VARIABLE IS UTILITY COST PER BUILDING.
 O3

 PRIMARY EXPLANATORY VARS
& ?
*** THE PRIMARY EXPLANATORY VARIABLES DENOTE TOTAL NON-LAB, LAB, PARKING, AND
*** UNASSIGNABLE SPACE WITHIN A BUILDING.
 O6 O7 O8 O9

 SECONDARY EXPL. VARS.
& ?
*** UNDER HETEROSCEDASTICITY OPTION 1, THE USER MAY SPECIFY ANY VARIABLE WHICH
*** HE/SHE HAS READ IN TO CORRECT FOR HETEROSCEDASTICITY.
 15

 ALLOCATION VARIABLES
& ?
*** THE ALLOCATION VARIABLES DENOTE TOTAL NON-LAB, LAB, PARKING, AND UNASSIGNABLE
*** SPACE USED FOR ORGANIZED RESEARCH WITHIN A BUILDING.
 11 12 13 14

 DEPENDENT VARIABLE:  3
 PRIMARY EXPLANATORY VARIABLES:    6  7  8  9
 SECONDARY EXPLANATORY VARIABLES: 15
 ALLOCATION VARIABLES:            11 12 13 14

 CORRECT? (OO=YES,O1=NO)
& ?
 OO

 DO YOU WANT NONZERO STARTING GAMMAS?
 (O1=NO,OO=YES)
& ?
*** GAMMA IS ESTIMATED ITERATIVELY.
 OO

 DO YOU WANT SPECIAL TERMINATION LIMITS FOR ESTIMATION?
 (O1=NO,OO=YES)
& ?
 O1

 ESTIMATED MODEL FOR VARIABLE  3

 SECONDARY EQUATION FOR HETEROSCEDASTICITY

 VARIABLE     GAMMA       ST ERROR     DELGAMMA       SIGNIF
     O        O.1380      O.3477
    15        O.9709      O.3295E-O1  O.3344E-O2      O.OOOO

 SIGMA O  =    1.148
```

```
ITERATIONS:   6            LIMIT:   10
F-STAT:     0.0022         LIMIT:  0.0100


PRIMARY EQUATION USING BLU REGRESSION

VARIABLE      BETA        ST ERROR          T-STAT      SIGNIF
    6        0.6539       0.1549             4.22       0.0000
    7         3.188       0.3853             8.27       0.0000
    8        0.1424       0.2804             0.50       0.6152
    9         2.569       0.3846             6.68       0.0000

RSQ-WLS =        0.7617     RSQ-MLE   =    0.9821

F-STAT =        159.01      SIGNIF   =    0.0000

 CHI SQ =        816.22    SAMPLE N   =       203


 THE ALLOCATED COST IS      4301112.61

 THE STANDARD ERROR IS        54062.82


DO YOU WANT TO SAVE ANY VARIABLES USING THIS MODEL?

OPTIONS:
  01 - THE EXPECTED VALUE PREDICTED BY THE PRIMARY EQUATION
  02 - THE SIGMA PREDICTED BY THE SECONDARY EQUATION
  03 - THE RESIDUAL (ACTUAL - EXPECTED VALUE)
  04 - THE STANDARDIZED RESIDUAL (RESIDUAL / SIGMA)
  05 - THE ALLOCATED VALUE
  06 - NO
& ?
 05

SAVE OPTION:   5

CORRECT? (00=YES,01=NO)
& ?
 00

THE ALLOCATION VALUE HAS BEEN SAVED

ENTER THE LABEL TO BE USED  (E.G. 09)
& ?
 16

THE DESIRED LABEL IS 16

CORRECT? (00=YES,01=NO)
& ?
 00
THIS VARIABLE HAS THE LABEL 16


DO YOU WANT TO SAVE ANY VARIABLES USING THIS MODEL?

OPTIONS:
  01 - THE EXPECTED VALUE PREDICTED BY THE PRIMARY EQUATION
```

```
   02 - THE SIGMA PREDICTED BY THE SECONDARY EQUATION
   03 - THE RESIDUAL (ACTUAL - EXPECTED VALUE)
   04 - THE STANDARDIZED RESIDUAL (RESIDUAL / SIGMA)
   05 - THE ALLOCATED VALUE
   06 - NO
& ?
*** MORE THAN ONE VARIABLE MAY BE SAVED IF DESIRED.
 06

 SAVE OPTION:   6

 CORRECT? (00=YES,01=NO)
& ?
 00

 ENTER ANALYSIS OPTION:
   01 - ENTER A COMMENT
   02 - READ IN A SAMPLE DATA BASE
   03 - SELECT CASES FOR ANALYSIS
   04 - WRITE OUT DATA
   05 - DESCRIBE THE DATA
   06 - ESTIMATE THE MODEL
   10 - STOP
& ?
 04

 ANALYSIS OPTION:  4

 CORRECT? (00=YES,01=NO)
& ?
 00

 ENTER VARIABLES TO BE WRITTEN OUT

 VARIABLES
& ?
 16

 NUMBER OF CASES TO BE WRITTEN OUT  E.G. 10.
& ?
 15.

 DATA FORMAT FOR OUTPUT  E.G. (2F10.2)
& ?
 (1F9.0)

 OUTPUT DEVICE E.G. 06
& ?
 07

 WRITE OUT
 VARIABLES    16
 FOR THE FIRST    15 OF THE  203 SELECTED CASES
 USING THE FORMAT (1F9.0)
 ON THE DEVICE  7

 CORRECT? (00=YES,01=NO)
& ?
 00
*** THESE ARE THE ALLOCATIONS TO ORGANIZED RESEARCH FOR THE FIRST 15 OF THE 203 BUILDINGS.
```

```
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
      0.
  18973.

  15 CASES HAVE BEEN WRITTEN

 ENTER ANALYSIS OPTION:
   01 - ENTER A COMMENT
   02 - READ IN A SAMPLE DATA BASE
   03 - SELECT CASES FOR ANALYSIS
   04 - WRITE OUT DATA
   05 - DESCRIBE THE DATA
   06 - ESTIMATE THE MODEL
   10 - STOP
& ?
 10

 ANALYSIS OPTION: 10

 CORRECT? (00=YES.01=NO)
& ?
 00
```

```
STATISTICAL COST ALLOCATION
      APRIL 27, 1983 VERSION

PREPARED FOR  THE UNIVERSITY OF MICHIGAN

         BY ROGER L. WRIGHT
            ASSOCIATE PROFESSOR OF STATISTICS
            GRADUATE SCHOOL OF BUSINESS ADMINISTRATION
            THE UNIVERSITY OF MICHIGAN

#Execution terminated
```

```
C  *****  STATISTICAL COST ALLOCATION - MAIN PROGRAM
C    FILE - SRY6:ALLOCATE
C
C ***********************************************************
C SECTION 1 PROGRAM SETUP
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION TITLE(9,10),FMT(5),AY(4),REQN(5),TBS(3,17),
     *ITB(14),PAR(6,2)
      DATA TBS/
     *'DEPENDEN','T VARIAB','LE      ',
     *'PRIMARY ','EXPLANAT','ORY VARS',
     *'SECONDAR','Y EXPL. ','VARS.   ',
     *'WEIGHT V','ARIABLE ','        ',
     *'INCLUSIO','N PROBAB','ILITIES ',
     *'A-VARIAB','LE FOR P','OP CHAR ',
     *'MAX NUMB','ER OF IT','ERATIONS',
     *'MINIMUM ','F-STATIS','TIC     ',
     *'PRECISIO','N       ','        ',
     *'RELIABIL','ITY     ','        ',
     *'Z-VALUE ','        ','        ',
     *'STRATIFI','CATION V','ARIABLE ',
     *'PPS-VARI','ABLE    ','        ',
     *'CASE SEL','ECTOR VA','RIABLES ',
     *'VARIABLE','S       ','        ',
     *'LABELS  ','        ','        ',
     *'ALLOCATI','ON VARIA','BLES    '/
C
C  THE FOLLOWING PARAMETERS ARE USED IN DIMENSIONING ARRAYS:
C  NS  =  MAXIMUM SAMPLE SIZE
C  NP = MAXIMUM NUMBER OF CASES IN THE POPULATION DATA BASE
C  NB = MAXIMUM NUMBER OF CASES IN THE BUFFER
C  NOTE: NB MUST BE AT LEAST MAX(NS,NP)
C  NVTS = MAXIMUM NUMBER OF VARIABLES IN THE SAMPLE DATA BASE
C  NVTP = MAXIMUM NUMBER OF VARIABLES IN THE POPULATION DATA BASE
C  NOTE: BOTH OF THESE DATA BASES USE ONE COLUMN FOR HANDLING CASE SELECTION
C  NVTB = MAXIMUM NUMBER OF VARIABLES IN THE BUFFER ARRAY
C  NOTE: NVTB MUST BE AT LEAST MAX(NVTS,NVTP,2)
C  NSTT  =  MAXIMUM NUMBER OF STRATA
C  NL   =   NVTB+5
C  NSV   =   NVTB*3
C  THE GENERAL DIMENSION STATEMENT IS THE FOLLOWING:
C     DIMENSION DS(NS,NVTS), DP(1,1), DB(NB,NVTB),
C    *G(NVTB), SG(NVTB), DG(NVTB),
C    *B1(NVTB), B(NVTB), SB(NVTB), TB(NVTB), PB(NVTB), AX(NVTB),
C    *PI(NB),Q(NB),ISQP(NB),ISQP1(NB),
C    *NPS(NSTT),FPS(NSTT),RPS(NSTT),RSRPS(NSTT),YMPS(NSTT),SDPS(NSTT),WPS(NSTT),
C    *LD(NL,12),SL(NL,2),SV(NSV)
C
      DIMENSION DS(300,16),DP(1,1),DB(300,16),
     *G(16),SG(16),DG(16),
     *B1(16),B(16),SB(16),TB(16),PB(16),AX(16),
     *D(16),C(16,16),
     *PI(300),Q(300),ISQP(300),ISQP1(300),
     *NPS(20),FPS(20),RPS(20),RSRPS(20),YMPS(20),SDPS(20),WPS(20),
     *LD(21,12),SL(21,2),SV(48)
C
C SET PROGRAM LIMITS
C  IF THE PROGRAM IS REDIMENSIONED THE FOLLOWING PARAMETERS
C  MUST BE REVISED
```

```
C
      NS=300
      NP=300
      NB=300
      NVTS=16
      NVTP=16
      NVTB=16
      NSTT=20
      NL=21
C  NO OTHER CHANGES SHOULD BE REQUIRED FOR REDIMENSIONING
C  ****************************************************************
C
      LD(2,1)=NVTS
      LD(2,2)=NVTP
      LD(2,3)=NVTB
      LD(2,5)=NVTB
      LD(2,6)=NVTB
      LD(2,11)=NVTB
      LD(2,12)=NVTB
      LD(1,1)=0
      LD(1,2)=0
      LD(2,7)=1
      LD(2,8)=1
      LD(2,9)=1
      LD(2,10)=1
      DO 110 K=1,12
 0110 LD(3,K)=0
      AY(1)=0.D0
C RESERVE V1 AS THE UNIT VARIABLE WITH TRANS CODE 1:
      LD(3,3)=1
      LD(4,3)=1
      LD(4,4)=1
C RESERVE V99 AS THE CASE NUMBER VARIABLE WITH TRANS CODE 3:
      LD(3,3)=2
      LD(5,3)=99
      LD(5,4)=3
      DO 112 K=4,NL
  112 SL(K,1)=-1.0D75
      DO 114 K=4,NL
  114 SL(K,2)=1.0D75
      NVTS1=NVTS-1
      NVTP1=NVTP-1
      WRITE(7,0116)
 0116 FORMAT(1X/'       STATISTICAL COST ALLOCATION './1X/1X/1X/)
      WRITE(7,0120) NVTS1,NS
 0120 FORMAT(1X/' CURRENT PROGRAM LIMITS:'./,
     *' NUMBER OF VARIABLES IN SAMPLE DATA BASE:       ',I4,/,
     *' NUMBER OF CASES IN SAMPLE DATA BASE:           ',I4)
      CALL CHECK(&0130,&0140)
 0130 WRITE(7,0135)
 0135 FORMAT(1X/' TO CHANGE THE LIMITS YOU MUST EDIT THE PROGRAM')
 0140 WRITE(8,0145)
 0145 FORMAT(' 0  OK')
 0500 WRITE(7,0510)
 0510 FORMAT(1X/' ENTER ANALYSIS OPTION:',/,
     *'    01 - ENTER A COMMENT',/,
     *'    02 - READ IN A SAMPLE DATA BASE',/,
     *'    03 - SELECT CASES FOR ANALYSIS',/,
     *'    04 - WRITE OUT DATA',/,
     *'    05 - DESCRIBE THE DATA',/,
```

```
     *'   06 - ESTIMATE THE MODEL',/,
     *'   10 - STOP',/,
     *'&?')
      READ(5,0520) IOPT
 0520 FORMAT(I2)
      WRITE(7,0525) IOPT
 0525 FORMAT(1X/' ANALYSIS OPTION: ',I2)
      CALL CHECK(&0500,&0530)
 0530 GO TO(1000,2000,3000,4000,5000,6000,7000,8000,9000,10000), IOPT
      WRITE(7,0540) IOPT
 0540 FORMAT(1X/' ANALYSIS OPTION ',I3,' IS NOT RECOGNIZED')
      GO TO 0500
C ************************************************************
C SECTION 1 COMMENT
 1000 WRITE(8,1010) IOPT
 1010 FORMAT(I2,'  COMMENT',/,
     *' 0  OK')
      CALL COMNT(TITLE)
      GO TO 0500
C ************************************************************
C SECTION 2 READ IN DATA
 2000 WRITE(8,2010) IOPT
 2010 FORMAT(I2,'  READ IN SAMPLE DATA BASE',/,
     *' 0  OK')
 2015 WRITE(6,2020)
 2020 FORMAT(1X/1X/' SAMPLE DATA BASE PARAMETERS')
      CALL DBASE(LD,NL,DS,NS,1,DB,NB,SL,FMT,TBS)
      GO TO 0500
C ************************************************************
C SECTION 3 SELECT CASES FOR ANALYSIS
 3000 WRITE(8,3010) IOPT
 3010 FORMAT(I2,'  SELECT CASES',/,
     *' 0  OK')
      WRITE(6,3020)
 3020 FORMAT(1X/1X/' CASE SELECTION')
 3030 CALL SELECT(LD,NL,DS,NS,1,DB,NB,SL,TBS,1,IER)
      CALL SELECT(LD,NL,DS,NS,1,DB,NB,SL,TBS,3,IER)
      IF(IER.EQ.9) GO TO 3030
      CALL SELECT(LD,NL,DP,NP,2,DB,NB,SL,TBS,3,IER)
      IF(IER.EQ.9) GO TO 3030
      GO TO 0500
C ************************************************************
C SECTION 4 WRITE OUT DATA
 4000 WRITE(8,4010) IOPT
 4010 FORMAT(I2,'  WRITE OUT DATA',/,
     *' 0  OK')
      CALL WRITE(LD,NL,DS,NS,DP,NP,DB,NB,FMT,TBS)
      GO TO 0500
C ************************************************************
C SECTION 5 DESCRIBE DATA
 5000 WRITE(8,5010) IOPT
 5010 FORMAT(I2,'  DESCRIBE DATA',/,
     *' 0  OK')
      CALL DESC(LD,NL,DS,NS,DP,NP,DB,NB,Q,NPS,FPS,RPS,TBS)
      GO TO 0500
C ************************************************************
C SECTION 6 ESTIMATE MODEL
 6000 WRITE(8,6010) IOPT
 6010 FORMAT(I2,'  ESTIMATE MODEL',/,
     *' 0  OK')
```

```
      CALL MODEL(LD,NL,DS,NS,DP,NP,DB,NB,SL,TBS,ITB,PAR,
     *SO,G,SG,DG,B,SB,TB,PB,SV,Q,C,D)
      GO TO 0500
C ***********************************************************
C SECTION 10 STOP
 7000 CONTINUE
 8000 CONTINUE
 9000 CONTINUE
10000 WRITE(8,10010) IOPT
10010 FORMAT(I2,'  STOP',/,
     *' O  OK')
      CALL STOP
      STOP
      END
C ***********************************************************
      SUBROUTINE MODEL(LD,NL,DS,NS,DP,NP,DB,NB,SL,TBS,ITB,PAR,
     *SO,G,SG,DG,B,SB,TB,PB,SV,Q,C,D)
C
C PURPOSE: TO SPECIFY AND ESTIMATE A DESIRED MODEL.
C LD IS THE HOUSEKEEPING ARRAY WITH EXACTLY NL ROWS.
C DS IS THE SAMPLE DATA BASE WITH EXACTLY NS ROWS.
C DB IS THE BUFFER DATA ARRAY WITH NB ROWS.
C TBS IS A ARRAY OF VARIABLE DESCRIPTIVE PHRASES.
C ITB AND PAR ARE USEDTO INPUT THE TERMINATION PARAMETERS.
C SO, G, SG, AND DG ARE PARAMETERS OF THE SECONDARY MODEL:
C NOTE THAT THE INPUT VALUES OF G ARE USED AS INITIAL
C VALUES IN THE ESTG ALGORITHM.
C SO IS THE SCALE PARAMETER, G IS THE VECTOR OF COEFFICIENTS,
C SG IS THE VECTOR OF ASYMPTOTIC STANDARD ERRORS, DG IS THE
C LAST CHANGE IN G IN THE ITERATIVE ESTIMATION ALGORITHM.
C B, SB, TB, AND PB ARE  PARAMETERS OF THE PRIMARY MODEL:
C B IS THE VECTOR OF COEFFICIENTS, SB THE STANDARD ERRORS,
C TB THE T-STATS, AND PB THE SIGNIFICANT LEVELS.
C SV IS A VECTOR WHICH ISUSED FOR SINGULAR VALUES AND WORK SPACE.
C Q IS A VECTOR USED TO STORE WLS WEIGHTS.
C C IS THE VARIANCE MATRIX OF B.
C D IS THE VECTOR USED TO CALCULATE THE PRECISION OF THE COST ALLOCATION.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),DS(NS,100),DP(NP,100),DB(NB,100),TBS(3,17),
     *ITB(14),PAR(6,2),G(NL),SG(NL),DG(NL),B(NL),SB(NL),TB(NL),PB(NL),
     *SV(NL),Q(NS),SL(NL,2),C(NL,NL),D(NL)
      IF (LD(1,1) .EQ. 0) WRITE(7,100)
  100 FORMAT(1X/' YOU HAVE NO DATA TO ANALYZE',/,
     *' YOU MUST READ IN YOUR SAMPLE DATA')
      IF (LD(1,1) .NE. 0) GO TO 150
      RETURN
  150 WRITE(7,152)
  152 FORMAT(1X/' ENTER HETEROSCEDASTICITY OPTION:',/,
     *'    01 - EXPONENTIAL MODEL FOR SIGMA ',/,
     *'    02 - SIGMA PROPORTIONAL TO THE EXPECTED VALUE',/,
     *'&?')
      READ(5,154) IOPT
  154 FORMAT(I2)
      WRITE(7,156) IOPT
  156 FORMAT(1X/' HETEROSCEDASTICITY OPTION: ',I2)
      CALL CHECK(&150,&165)
  165 GO TO(180,180), IOPT
      WRITE(7,167) IOPT
  167 FORMAT(1X/' HETEROSCEDASTICITY OPTION ',I3,' IS NOT RECOGNIZED')
```

```
      GO TO 150
  180 WRITE(8,182) IOPT
  182 FORMAT(I2,'  HETEROSCEDASTICITY OPTION',/,' O  OK')
  200 KLIM=O
      WRITE(7,201)
  201 FORMAT(1X/' SPECIFY VARIABLES ')
      CALL INVAR(LD,NL,3,7,TBS(1,1),3,IER)
      CALL INVAR1(LD,NL,7,1,&200,&202)
  202 IF ( LD(3,7).GT.O ) GO TO 206
      WRITE(7,203)
  203 FORMAT(1X/' CANCEL ALL VARIABLES IN THE MODEL')
      CALL CHECK(&200,&204)
  204 WRITE(8,205)
  205 FORMAT('OO  NO VARIABLES',/,' O  OK')
      LD(3,6)=O
      LD(3,5)=O
      RETURN
  206 IF (IER .EQ. O ) GO TO 212
      WRITE(7,211)
  211 FORMAT(1X/' THIS VARIABLE IS NOT AVAILABLE')
      GO TO 200
  212 CALL INVAR(LD,NL,3,5,TBS(1,2),3,IER)
      NMAX=LD(2,5)-1
      CALL INVAR1(LD,NL,5,NMAX,&212,&213)
  213 IF (IER .EQ. O) GO TO 215
      WRITE(7,214)
  214 FORMAT(1X/' ONE OF THESE VARIABLES IS UNAVAILABLE',/,
     *' OR YOU HAVE SPECIFIED TOO MANY VARIABLES')
      GO TO 212
  215 IF (IOPT .EQ. 1) GO TO 216
C SET UP AN INDICATOR FOR THE SPECIAL HETEROSCEDASTICITY OPTION:
      LD(3,6)=O
      LD(4,6)=1
      GO TO 218
  216 CALL INVAR(LD,NL,3,6,TBS(1,3),3,IER)
      NMAX=LD(2,6)-1
      CALL INVAR1(LD,NL,6,NMAX,&216,&217)
  217 IF(LD(3,5).EQ.O) LD(3,6)=O
      IF (IER .EQ. O) GO TO 218
      WRITE(7,214)
      GO TO 216
  218 CALL INVAR(LD,NL,3,11,TBS(1,17),3,IER)
      IF (IER .EQ. O) GO TO 220
      WRITE(7,214)
      GO TO 218
  220 NX=LD(3,5)
      NXO=LD(3,11)
      IF (NXO .EQ. O .OR. NXO .EQ. NX) GO TO 225
      WRITE(7,   2)
    2  FORMAT('OTHE NUMBER OF ALLOCATION VARIABLES MUST EQUAL THE',
     *' NUMBER OF EXPLANATORY VARIABLES')
      GO TO 212
  225 WRITE(7,230) LD(4,7)
  230 FORMAT(1X/' DEPENDENT VARIABLE: ',I2)
      IF(NX.EQ.O) WRITE(7,231)
  231 FORMAT(' NO PRIMARY EXPLANATORY VARIABLES')
      JX=NX+3
      SO=1.DO
      IF(NX.EQ.O) GO TO 233
      WRITE(7,232) (LD(J,5),J=4,JX)
```

```
232   FORMAT(' PRIMARY EXPLANATORY VARIABLES:   ',15(1X,I2))
233   NT=LD(3,6)
      JT=NT+3
      IF( NT .GT. O) GO TO 260
      WRITE(7,250)
250   FORMAT(' NO SECONDARY EXPLANATORY VARIABLES')
      GO TO 265
260   WRITE(7,264) (LD(J,6),J=4,JT)
264   FORMAT(' SECONDARY EXPLANATORY VARIABLES:',15(1X,I2))
265   IF( NXO .GT. O) GO TO 268
      WRITE(7,266)
266   FORMAT(' NO ALLOCATION VARIABLES')
      GO TO 270
268   WRITE(7,269) (LD(J,11),J=4,JX)
269   FORMAT(' ALLOCATION VARIABLES:                 ',15(1X,I2))
270   CALL CHECK(&200,&300)
300   WRITE(8,310) LD(4,7)
310   FORMAT(20(I2,1X))
312   WRITE(8,310) (LD(J,5),J=4,JX)
      J3=JX-20*(JX/20)
314   IF(J3.EQ.O) WRITE(8,315)
      IF(IOPT.LQ.2) GO TO 316
      WRITE(8,310) (LD(J,6),J=4,JT)
      J3=JT-20*(JT/20)
      IF(J3.EQ.O) WRITE(8,315)
315   FORMAT('OO')
316   JXO=NXO+3
      WRITE(8,310) (LD(J,11),J=4,JXO)
      J3=JXO-20*(JXO/20)
317   IF(J3.EQ.O) WRITE(8,315)
      WRITE(8,320)
320   FORMAT(' O  OK')
      CALL SELECT(LD,NL,DS,NS,1,DB,NB,SL,TBS,2,IER)
      IF(IER.EQ.4 ) RETURN
 325  CALL SELECT(LD,NL,DP,NP,2,DB,NB,SL,TBS,2,IER)
      IF(NX.EQ.O) RETURN
 329  IF(NT.EQ.O .AND. IOPT .EQ. 1) GO TO 650
 330  WRITE(7,331)
 331  FORMAT(1X/' DO YOU WANT TO BEGIN THE ESTIMATION  ',/,
     *' AT GAMMA = O? ',/,
     *' (OO=YES,O1=NO)'/'&?')
      READ(5,332) IOPT2
 332  FORMAT(I2)
      IF(IOPT2 .EQ. 1) GO TO 340
      IF(IOPT2 .EQ. O) GO TO 390
      WRITE(7,335)
 335  FORMAT(1X/' INVALID RESPONSE')
      GO TO 330
 340  WRITE(8,341)
 341  FORMAT(' 1  STARTING VALUES FOR GAMMA')
 342  WRITE(7,343)
 343  FORMAT(1X/' ENTER STARTING GAMMA FOR EACH VARIABLE  E.G. .5')
      DO 350 J=4,JT
      J1=J-3
      WRITE(7,344) LD(J,6)
 344  FORMAT(I4,/,'&?')
      READ(5,346) G(J1)
 346  FORMAT(G20.12)
 350  CONTINUE
      WRITE(6,352)
```

```
352 FORMAT(1X/1X/' STARTING GAMMAS ',/1X/,
   *' VARIABLE  GAMMA')
    DO 360 J=4,JT
    J1=J-3
    WRITE(6,354) LD(J,6),G(J1)
354 FORMAT(3X,I2,5X,G20.12)
360 CONTINUE
    CALL CHECK(&342,&370)
370 DO 380 J=4,JT
    J1=J-3
    WRITE(8,374) G(J1)
374 FORMAT(G20.12)
380 CONTINUE
    WRITE(8,382)
382 FORMAT(' O  OK')
    GO TO 400
390 WRITE(8,392)
392 FORMAT(' O  USE DEFAULT GAMMAS')
    DO 394 J=4,JT
    J1=J-3
394 G(J1)=0.DO
400 WRITE(7,402)
402 FORMAT(1X/' DO YOU WANT THE ESTIMATION TO PROCEED ',
   *' WITHOUT SPECIAL TERMINATION LIMITS? ',/,
   *' (OO=YES,O1=NO)'/'&?')
    READ(5,404) IOPT2
404 FORMAT(I2)
    IF(IOPT2 .EQ. 1) GO TO 410
    IF(IOPT2 .EQ. 0) GO TO 500
    GO TO 400
410 WRITE(8,411)
411 FORMAT(' 1  SPECIAL TERMINATION LIMITS')
412 WRITE(7,413)
413 FORMAT(1X/' MAXIMUM NUMBER OF ITERATIONS   E.G. 05',/,
   *'&?')
    READ(5,414) KLIM
414 FORMAT(I2)
    WRITE(7,416)
416 FORMAT(1X/' MINIMUM F-STAT  E.G. .01',/,'&?')
    READ(5,418) FLIM
418 FORMAT(G20.12)
    WRITE(7,420) KLIM,FLIM
420 FORMAT(1X/' MAXIMUM NUMBER OF ITERATIONS: ',I4,/,
   *' MINIMUM FSTAT:',F6.4)
    CALL CHECK(&412,&430)
430 WRITE(8,432) KLIM,FLIM
432 FORMAT(I2,/,G20.12,/,' O  OK')
    GO TO 600
500 KLIM=10
    FLIM=.01
    WRITE(8,510)
510 FORMAT(' O  DEFAULT TERMINATION LIMITS')
600 K=KLIM
    F=FLIM
    IF(IOPT.EQ.2) GO TO 6122
    IF (NT.GT.0) GO TO 6000
650 DO 700 I=1,NS
700 Q(I)=1.DO
    WRITE(6,710) LD(4,7)
710 FORMAT(1X/1X/' ESTIMATED MODEL FOR VARIABLE ',I2,/,
```

```
     *' ASSUMING THAT SIGMA IS CONSTANT')
      GO TO 6205
 6000 IF(KLIM .GT. 0) GO TO 6090
      WRITE(6,6010) LD(4,7)
 6010 FORMAT(1X/1X/' ESTIMATED MODEL FOR VARIABLE ',I2,/,
     *' ASSUMING THE INITIAL VALUES OF GAMMA',/,
     *1X/' ASSUMED SECONDARY EQUATION FOR HETEROSCEDASTICITY',/,
     *1X/' VARIABLE     GAMMA')
      DO 6020 J=1,NT
      J1=J+3
 6020 WRITE(6,6022) LD(J1,6),G(J)
 6022 FORMAT(' ',3X,I2,5X,1G12.4)
      GO TO 6150
 6090 CALL ESTG(LD,NL,DS,NS,DB,NB,Q,SV,B,G,SG,DG,SO,SE,K,F,1,IER)
      IF(IER.NE.0) RETURN
      NC=LD(1,3)
      NDF=NC-NX
      IF(IER .NE. 0) RETURN
      WRITE(6,6100) LD(4,7)
 6100 FORMAT(1X/1X/' ESTIMATED MODEL FOR VARIABLE ',I2,/,
     *1X/' SECONDARY EQUATION FOR HETEROSCEDASTICITY',/,
     *1X/' VARIABLE    GAMMA        ST ERROR      DELGAMMA        SIGNIF')
      J=NT+1
      WRITE(6,6105) G(J),SG(J)
 6105 FORMAT('      0      ',2G12.4)
      DO 6110 J=1,NT
      J1=J+3
      TSQ=(G(J)/SG(J))**2
      CALL MDFD(TSQ,1,NDF,SIGG,IER)
      SIGG=1.D0 - SIGG
 6110 WRITE(6,6120) LD(J1,6),G(J),SG(J),DG(J),SIGG
 6120 FORMAT(' ',3X,I2,5X,3G12.4,F12.4)
      GO TO 6124
 6122 CALL ESTG1(LD,NL,DS,NS,DB,NB,Q,SV,B,G,SO,K,F,1,IER)
 6124 WRITE(6,6130) SO,K,KLIM,F,FLIM
 6130 FORMAT(1X/' SIGMA O  =  ',1G10.4,/,
     *1X/' ITERATIONS: ',1I3,'           LIMIT: ',I3,/,
     *' F-STAT: ',1F10.4,'        LIMIT: ',1F7.4)
 6150 CALL SIGMA(LD,NL,DS,NS,1,DB,NB,G,B,SO,Q,IER)
      NC=LD(1,3)
      DO 6200 I=1,NC
 6200 Q(I)=1.D0/Q(I)**2
 6205 CALL ESTB(LD,NL,DS,NS,DB,NB,Q,SV,B,SB,TB,PB,SE,SE1,CHISQ,
     *RSQ,RSQ1,FSTAT,SIGF,NC1,1,IER)
      IF (IOPT .EQ. 2) GO TO 6208
      IF( NT .GT. 0 .AND. KLIM .GT. 0 ) GO TO 6208
      WRITE(6,6206) SE
 6206 FORMAT(1X/' SIGMA O  =  ',1G10.4)
 6208 WRITE(6,6210)
 6210 FORMAT(1X/1X/' PRIMARY EQUATION USING BLU REGRESSION',/,
     *1X/' VARIABLE     BETA        ST ERROR          T-STAT      SIGNIF')
      DO 6220 J=1,NX
      J1=J+3
 6220 WRITE(6,6230) LD(J1,5),B(J),SB(J),TB(J),PB(J)
 6230 FORMAT(' ',3X,I2,5X,2G12.4,F12.2,F12.4)
      WRITE(6,6240) RSQ,RSQ1,FSTAT,SIGF,CHISQ,NC1
 6240 FORMAT(1X/
     *' RSQ-WLS = ',1F12.4,'     RSQ-MLE   =  ',1F8.4,/1X/
     *' F-STAT = ',1F12.2,'       SIGNIF   =  ',1F8.4,/1X/
     *'  CHI SQ = ',1F12.2,'    SAMPLE N   =  ',I8)
```

```
      IF (NT.GT.O .AND. KLIM .GT. O ) GO TO 6600
      SO=SE
 6600 IF (NXO .EQ. O) GO TO 7000
C CALCULATE THE VARIANCE MATRIX OF B
      SOSQ=SO*SO
      DO 6640 I=1,NX
      DO 6640 J=1,NX
      C(I,J)=O.DO
      DO 6620 K=1,NX
 6620 C(I,J)=C(I,J) + DB(I,K)*DB(J,K)/SV(K)**2
 6640 C(I,J)=C(I,J)*SOSQ
      JXO=NX+1
      JY=2*NX+1
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JXO),NB,11,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
C CALCULATE THE TOTAL ALLOCATED COST AC, AND THE VECTOR D
C USED TO CALCULATE THE PRECISION OF THE ALLOCATION
C HERE XB = X(I)'B AND XOB = XO(I)'B WHERE X(I) IS THE X-VECTOR
C DESCRIBING BUILDING I, XO IS THE VECTOR DESCRIBING RESEARCH
C IN BUILDING I, AND B IS THE VECTOR OF ESTIMATED COEFFICIENTS
C OF THE MODEL.  THUS THE ALLOCATION FACTOR FOR BUILDING I IS
C A = XOB/XB AND THE ALLOCATED COST FOR BUILDING I IS A*Y
C IN SIMILAR NOTATION, THE D-VECTOR IS THE SUM ACROSS ALL BUILDINGS
C OF THE VECTOR  Y*(XO - A*X)/XB
      NCB=LD(1,3)
      AC=O.DO
      DO 6650 J=1,NX
 6650 D(J)=O.DO
      DO 6700 I=1,NCB
      XOB=O.DO
      XB=O.DO
      DO 6670 J=1,NX
      XB=XB + B(J)*DB(I,J)
      J1=J+NX
 6670 XOB=XOB + B(J)*DB(I,J1)
      A=XOB/XB
      AC=AC+A*DB(I,JY)
C CALCULATE EACH ENTRY OF D FOR THIS BUILDING
      DO 6680 J=1,NX
      J1=J+NX
 6680 D(J)=D(J) + DB(I,JY)*(DB(I,J1) - A*DB(I,J))/XB
 6700 CONTINUE
C CALCULATE THE STANDARD DEVIATION, SAC,  OF THE COST ALLOCATION,
C AS D'C D, WHERE D IS THE VECTOR CALCULATED ABOVE AND C IS THE
C VARIANCE MATRIX OF B
      SAC=O.DO
      DO 6720 J=1,NX
      DO 6720 K=1,NX
 6720 SAC=SAC + D(J)*C(J,K)*D(K)
      SAC=DSQRT(SAC)
      WRITE(6,6750) AC,SAC
 6750 FORMAT(/1X/1X,' THE ALLOCATED COST IS',F15.2,
     */,     /1X,' THE STANDARD ERROR IS',F15.2)
 7000 WRITE(7,7010)
 7010 FORMAT(1X/1X/' DO YOU WANT TO SAVE ANY VARIABLES USING ',
     *'THIS MODEL?',/,/,
     *' OPTIONS:',/,
     *'    O1 - THE EXPECTED VALUE PREDICTED BY THE PRIMARY ',
     *'EQUATION',/,
```

```
     *'    02 - THE SIGMA PREDICTED BY THE SECONDARY EQUATION',/,
     *'    03 - THE RESIDUAL (ACTUAL - EXPECTED VALUE)',/,
     *'    04 - THE STANDARDIZED RESIDUAL (RESIDUAL / SIGMA)',/,
     *'    05 - THE ALLOCATED VALUE',/,
     *'    06 - NO',/,'&?')
      READ(5,7020) IOPT2
 7020 FORMAT(I2)
      WRITE(7,7030) IOPT2
 7030 FORMAT(1X/' SAVE OPTION:  ',I2)
      CALL CHECK(&7000,&7040)
 7040 GO TO (7100,7200,7300,7400,7500,7600),IOPT2
      WRITE(7,7050) IOPT2
 7050 FORMAT(1X/' SAVE OPTION ',I2,' IS NOT RECOGNIZED')
      GO TO 7000
 7100 WRITE(8,7110)
 7110 FORMAT(' 1  SAVE EXPECTED VALUE',/,' 0  OK')
      WRITE(6,7112)
 7112 FORMAT(1X/' THE EXPECTED VALUE PREDICTED BY THIS MODEL',
     *' HAS BEEN SAVED')
      CALL RESID(LD,NL,DS,NS,1,DB,NB,B,Q,2,IER)
      IF (IER.NE.0) GO TO 7600
      CALL SAVE(LD,NL,DS,NS,1,Q,L,&7600,&7000)
 7200 WRITE(8,7210)
 7210 FORMAT(' 2  SAVE SIGMA',/,' 0  OK')
      WRITE(6,7212)
 7212 FORMAT(1X/' THE SIGMA PREDICTED BY THIS MODEL',
     *' HAS BEEN SAVED')
      CALL SIGMA(LD,NL,DS,NS,1,DB,NB,G,B,SO,Q,IER)
      IF (IER .NE. 0) GO TO 7600
      CALL SAVE(LD,NL,DS,NS,1,Q,L,&7600,&7000)
 7300 WRITE(8,7310)
 7310 FORMAT(' 3  SAVE RESIDUAL',/,' 0  OK')
      WRITE(6,7312)
 7312 FORMAT(1X/' THE RESIDUAL OF THIS MODEL HAS BEEN SAVED')
      CALL RESID(LD,NL,DS,NS,1,DB,NB,B,Q,1,IER)
      IF(IER.NE.0) GO TO 7600
      CALL SAVE(LD,NL,DS,NS,1,Q,L,&7600,&7000)
 7400 WRITE(8,7410)
 7410 FORMAT(' 4  SAVE STANDARDIZED RESIDUAL',/,' 0  OK')
      WRITE(6,7412)
 7412 FORMAT(1X/' THE STANDARDIZED RESIDUAL OF THIS MODEL',
     *' HAS BEEN SAVED')
      CALL RESID(LD,NL,DS,NS,1,DB,NB,B,Q,1,IER)
      IF (IER .NE. 0) GO TO 7600
      CALL SIGMA(LD,NL,DS,NS,1,DB(1,2),NB,G,SO,DB,IER)
      IF (IER .NE. 0) GO TO 7600
      NC=LD(1,3)
      DO 7420 I=1,NC
 7420 Q(I)=Q(I)/DB(I,1)
      CALL SAVE(LD,NL,DS,NS,1,Q,L,&7440,&7000)

 7440 WRITE(7,7460)
 7460 FORMAT(1X/' UNABLE TO SAVE THIS VARIABLE')
      GO TO 7000
 7500 WRITE(8,7510)
 7510 FORMAT(' 5  ALLOCATION VALUE SAVED',/,' 0  OK')
      IF(NXO.GT.0) GO TO 7515
      WRITE(7,7512)
 7512 FORMAT(1X/' UNABLE TO CALCULATE THE ALLOCATION',/,
     *' YOU HAVE SPECIFIED NO ALLOCATION VARIABLES IN THE MODEL')
```

```
      GO TO 7000
 7515 WRITE(6,7516)
 7516 FORMAT(1X/' THE ALLOCATION VALUE HAS BEEN SAVED')
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JXO),NB,11,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
      NCB=LD(1,3)
      DO 7550 I=1,NCB
      XOB=0.D0
      XB=0.D0
      DO 7520 J=1,NX
      XB=XB + B(J)*DB(I,J)
      J1=J+NX
 7520 XOB=XOB + B(J)*DB(I,J1)
      A=XOB/XB
 7550 Q(I)=A*DB(I,JY)
      CALL SAVE(LD,NL,DS,NS,1,Q,L,&7600,&7000)
 7600 WRITE(8,7610)
 7610 FORMAT(' 6  NO VARIABLE SAVED',/,' 0  OK')
      RETURN
      END
C *****************************************************************
      SUBROUTINE STOP
      WRITE(6,99020)
99020 FORMAT('1'/1X/1X/1X/1X/1X/
     *' STATISTICAL COST ALLOCATION',/,
     *'          APRIL 27, 1983 VERSION',/,1X/
     *' PREPARED FOR  THE UNIVERSITY OF MICHIGAN',1X/1X/
     *'              BY ROGER L. WRIGHT ',/,
     *'                   ASSOCIATE PROFESSOR OF STATISTICS',/,
     *'                   GRADUATE SCHOOL OF BUSINESS ADMINISTRATION',/,
     *'                   THE UNIVERSITY OF MICHIGAN',/1X/1X)
      STOP
      END
C ***********************************************************
      SUBROUTINE DBASE(LD,NL,D,N,K,DB,NB,SL,FMT,TBS)
C
C PURPOSE: TO READ A DATA MATRIX INTO THE DATA ARRAY D.
C LD IS THE HOUSEKEEPING ARRAY WITH EXACTLY NL ROWS.
C THE INPUT DATA ARRAY D MAY BE EITHER THE SAMPLE DATA BASE DS
C OR THE POPULATION DATA BASE DP. N IS ITS EXACT ROW DIMENSION,
C EITHER NS OR NP.  K IS THE CORRESPONDING COLUMN OF LD, EITHER
C 1 FOR THE SAMPLE OR 2 FOR THE POPULATION.
C THE LABELS OF THE VARIABLES ARE ENTERED IN ROW K OF LD.
C THE LABELS ARE ALSO ENTERED IN THE LIST OF LOGICAL VARIABLES
C STORED IN THE 3RD COLUMN OF LD; THEY ARE GIVEN THE ASSIGNMENT
C TRANSFORMATION CODE "2" IN COLUMN 4.
C THE LABEL OF THE WEIGHT VARIABLE IS ALSO ENTERED IN LD;
C USING COLUMN 8 FOR THE SAMPLE WEIGHT, 9 FOR THE POPULATION.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100),SL(NL,2),FMT(5),TBS(3,16)
      NVT=LD(2,K)-1
  990 WRITE(7,1000)
 1000 FORMAT(1X/' ENTER DATA BASE PARAMETERS')

 1005 WRITE(7,1010)
 1010 FORMAT(1X/' ACTUAL NUMBER OF CASES  E.G. 210.   '/'&?')
      READ(5,1020) RNC
 1020 FORMAT(G12.5)
```

```
      NC=RNC
      IF ( NC  GT. O ) GO TO 1024
      WRITE(7,1022) NC
 1022 FORMAT(1X/' CANNOT READ ',I5,' CASES')
      WRITE(8,1023) RNC
 1023 FORMAT(G12.5,'    ILLEGAL NUMBER OF CASES')
      RETURN
 1024 IF ( NC .LE. N ) GO TO 1028
      WRITE(7,1025) RNC,N
 1025 FORMAT(1X/' THE NUMBER OF CASES ',G12.5,/,
     *' MUST BE NO GREATER THAN ',I4,/,
     *' BE SURE TO INCLUDE A DECIMAL POINT')
      GO TO 1005
 1028 WRITE(7,1030)
 1030 FORMAT(1X/' ENTER LABELS FOR VARIABLES  E.G. 02 03 04  ',/,
     *' UP TO 20 LABELS PER LINE')
      CALL INVAR(LD,NL,O,K,TBS(1,16),2,IER)
      CALL INVAR1(LD,NL,K,NVT,&1028,&1048)
 1048 NV=LD(3,K)
      WRITE(7,1050)
 1050 FORMAT(1X/' DATA FORMAT FOR INPUT  E.G. (1F10.3,1F5.0)  '/'&?')
      READ(5,1060) (FMT(I), I=1,5)
 1060 FORMAT(5A8)
      WRITE(7,1064)
 1064 FORMAT(1X/' INPUT DEVICE  E.G. 01    '/'&?')
      READ(5,1066) IDEV
 1066 FORMAT(1I2)
      WRITE(6,1080) N,NC,NVT,NV
 1080 FORMAT(1X/' MAXIMUM NUMBER OF CASES        ',I4,/,
     *' ACTUAL NUMBER OF CASES:        ',I4,/,
     *' MAXIMUM NUMBER OF VARIABLES: ',I4,/,
     *' ACTUAL NUMBER OF VARIABLES:   ',I4)
      NV1=NV+3
      WRITE(6,1090) (LD(J,K), J=4,NV1)
 1090 FORMAT(' LABELS: ',20(1X,I2))
      WRITE(6,1095) (FMT(I),I=1,5)
 1095 FORMAT(' FORMAT FOR INPUT: ',5A8)
      WRITE(6,1097) IDEV
 1097 FORMAT(' INPUT DEVICE: ',I2)
      CALL CHECK(&990,&1100)
 1100 WRITE(8,1102) RNC
 1102 FORMAT(G12.5,'  NUMBER OF CASES IN DATA BASE (/VARIABLE LIST)')
      WRITE(8,1104) (LD(J,K),J=4,NV1)
 1104 FORMAT(20(I2,1X))
      J3=NV-20*(NV/20)
      IF( J3 .EQ. O) WRITE(8,1105)
 1105 FORMAT('OO')
      WRITE(8,1106) (FMT(I),I=1,5)
 1106 FORMAT(5A8,'  INPUT FORMAT')
      WRITE(8,1108) IDEV
 1108 FORMAT(I2,'  INPUT DEVICE')
      WRITE(8,1110)
 1110 FORMAT(' O  OK')
 1200 DO 1250 J=4,NV1
      L=LD(J,K)
      CALL INDX(LD,NL,3,L,JL,&1210,&1220)
 1210 NVB=LD(3,3)
      IF ( NVB+3 .LT. NL) GO TO 1215
      NL2=NL-3
      WRITE(7,1212) NL2
```

```
1212 FORMAT(1X/' THE NUMBER OF LOGICAL VARIABLES MUST BE NO ',
    *'GREATER THAN ',I4)
     GO TO 990
1215 NVB=NVB+1
     LD(3,3)=NVB
     JL1=NVB+3
     LD(JL1,3)=L
     LD(JL1,4)=2
     GO TO 1250
1220 JL1=JL+3
     IF ( LD(JL1,4) .EQ. 2 ) GO TO 1250
     WRITE(7,1230) L
1230 FORMAT(1X/' VARIABLE ',I2,' IS TO BE REDEFINED')
     CALL CHECK(&1238,&1240)
1238 WRITE(8,1239)
1239 FORMAT(' 1  NO')
     GO TO 990
1240 WRITE(3,1242)
1242 FORMAT(' O  OK')
     LD(JL1,4)=2
1250 CONTINUE
     DO 1260 I=1,NC
     READ(IDEV,FMT,END=1262,ERR=1264) (D(I,J),J=1,NV)
1260 CONTINUE
     WRITE(7,1261) NC
1261 FORMAT(1X/,I5,' CASES HAVE BEEN READ')
     GO TO 1280
1262 I1=I-1
     WRITE(6,1263) I1,NC
1263 FORMAT(1X/,I5,' CASES HAVE BEEN READ OUT OF ',I5,/,
    *' BEFORE THE END OF THE FILE WAS REACHED')
     NC=I1
     GO TO 1280
1264 I1=I-1
     WRITE(7,1265) I1,NC,I
1265 FORMAT(1X/,I5,' CASES HAVE BEEN READ OUT OF ',I5,/,
    *' UNABLE TO READ CASE ',I5)
     GO TO 990
1280 LD(3,K)=NV
     LD(1,K)=NC
     CALL SELECT(LD,NL,D,N,K,DB,NB,SL,TBS,2,IER)
     IF ( K .EQ. 1 ) KW=8
     IF ( K .EQ. 2 ) KW=9
1300 LD(3,8)=1
     LD(4,8)=1
     RETURN
     END
C ************************************************************************
C   **************************************************************
C
      SUBROUTINE DESC(LD,NL,DS,NS,DP,NP,DB,NB,Q,NPS,FPS,RPS,TBS)
C  PURPOSE: TO DESCRIBE THE DATA USING WEIGHTS IN Q.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),DS(NS,100),DP(NP,100),DB(NB,100),TBS(3,15)
     *,Q(NB),NPS(20),FPS(20),RPS(20)
      CALL DESC1(LD,NL,DS,NS,1,DB,NB,Q,NPS,FPS,RPS,TBS)
      RETURN
      END
C   **********************************************************
```

```
C
      SUBROUTINE DESC1(LD,NL,D,N,K,DB,NB,Q,NPS,FPS,RPS,TBS)
C
C  PURPOSE:  TO DESCRIBE THE DATA IN D
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,100),D(N,100),DB(NB,100),Q(NB),NPS(20),
     *FPS(20),RPS(20),TBS(3,15)
      IF( LD(1,K) .GT. 0) GO TO 10
      WRITE(7,8)
    8 FORMAT(1X/' YOU HAVE NO DATA TO DESCRIBE',/,
     *' YOU MUST READ IN YOUR DATA FIRST')
      RETURN
   10 IF (K .EQ. 1) CALL TRANS(LD,NL,D,N,K,Q,NB,8,IER)
      IF (K .EQ. 2) CALL TRANS(LD,NL,D,N,K,Q,NB,9,IER)
   11 WRITE(7,12)
   12 FORMAT(1X/' ENTER DESCRIBE OPTION:',/,
     *'    01 - SUMMARY STATISTICS',/,
     *'    02 - FREQUENCY TABLE',/,
     *'    03 - NEW ANALYSIS OPTION',/,'&?')
   13 READ(5,14) IOPT
   14 FORMAT(I2)
      WRITE(7,16) IOPT
   16 FORMAT(1X/' DESCRIBE OPTION:   ',I2)
      CALL CHECK(&11,&20)
   20 GO TO (100,200,300), IOPT
      WRITE(7,22) IOPT
   22 FORMAT(1X/' DESCRIBE OPTION ',I2,' NOT RECOGNIZED')
      GO TO 10
  100 WRITE(8,110) IOPT
  110 FORMAT(I2,'  SUMMARY STATISTICS',/,' 0  OK')
  112 WRITE(7,113)
  113 FORMAT(1X/' ENTER VARIABLES TO BE DESCRIBED')
      CALL INVAR(LD,NL,3,11,TBS(1,15),1,IER)
      IF( IER .GT. 0 ) GO TO 120
      CALL INVAR1(LD,NL,11,LD(2,3),&112,&115)
  115 NV=LD(3,11)
      CALL TRANS(LD,NL,D,N,K,DB,NB,11,IER)
      NC=LD(1,3)
      IF ( IER .NE. 0 ) GO TO 120
      NC=LD(1,3)
      IF(NC .GT. 0) GO TO 140
      WRITE(7,116)
  116 FORMAT(1X/' YOU HAVE NO DATA TO DESCRIBE',/,
     *' YOU MUST CHANGE YOUR CASE SELECTION')
      RETURN
  120 WRITE(7,122)
  122 FORMAT(1X/' ONE OF THESE VARIABLES DOES NOT EXIST')
      GO TO 112
  140 WRITE(6,142) NC
  142 FORMAT(1X/1X/' SUMMARY STATISTICS      N = ',I4,/,1X/,
     *' VARIABLE  MINIMUM      MAXIMUM          MEAN       STD DEV')
      DO 190 J=1,NV
      J1=J+3
      C1=DB(1,J)
      C2=DB(1,J)
      SUM1=0.D0
      SUM2=0.D0
      SUM3=0.D0
      DO 150 I=1,NC
```

```
      F=Q(I)
      C=DB(I,J)
      IF ( C .LT. C1 ) C1=C
      IF ( C .GT. C2 ) C2=C
      SUM1=SUM1+F
      SUM2=SUM2+C*F
  150 SUM3=SUM3+C*C*F
      SUM2=SUM2/SUM1
      SUM3=(SUM3/SUM1) - SUM2*SUM2
      IF ( SUM3 .GT. O.DO ) GO TO 160
      SUM3=O.DO
      GO TO 170
  160 SUM3=DSQRT(SUM3)
  170 WRITE(6,172) LD(J1,11),C1,C2,SUM2,SUM3
  172 FORMAT(4X,I2,3X,4G13.4)
  190 CONTINUE
      GO TO 10
  200 WRITE (8,210) IOPT
  210 FORMAT(I2,'  FREQUENCY TABLE',/,' O  OK')
  212 WRITE(7,213)
  213 FORMAT(1X/' ENTER VARIABLES OR OO TO STOP')
      CALL INVAR(LD,NL,3,11,TBS(1,15),4,IER)
      IF ( IER .GT. O) GO TO 220
      CALL INVAR1(LD,NL,11,LD(2,3),&212,&215)
  215 NV=LD(3,11)
      IF(NV .LE. O) GO TO 10
      CALL TRANS(LD,NL,D,N,K,DB,NB,11,IER)
      IF ( IER .NE. O ) GO TO 220
      NC=LD(1,3)
      IF(NC .GT. O) GO TO 240
      WRITE(7,216)
  216 FORMAT(1X/' YOU HAVE NO DATA TO DESCRIBE',/,
     *' YOU MUST CHANGE YOUR CASE SELECTION')
      RETURN
  220 WRITE(7,222)
  222 FORMAT(1X/' ONE OF THESE VARIABLES DOES NOT EXIST')
      GO TO 212
  240 WRITE(7,242)
  242 FORMAT(1X/' NUMBER OF INTERVALS. O1 TO 10',/,
     *' OR ENTER OO TO STOP',/,'&?')
  243 READ(5,244) NINT
  244 FORMAT(I2)
      IF ( NINT .GT. 10 ) GO TO 240
      WRITE(7,245) NINT
  245 FORMAT(1X/' NUMBER OF INTERVALS: ',I2)
      CALL CHECK(&243,&246)
  246 WRITE(8,247) NINT
  247 FORMAT(I2,'  NUMBER OF INTERVALS',/,' O  OK')
      IF( NINT .EQ. O ) GO TO 212
      WRITE(7,248)
  248 FORMAT(1X/' ENTER UPPER LIMIT FOR EACH INTERVAL')
      DO 260 L=1,NINT
      WRITE(7,250) L
  250 FORMAT(1X/' UPPER LIMIT FOR INTERVAL ',I2,/,'&?')
      READ(5,252) RPS(L)
  252 FORMAT(1G20.12)
      WRITE(8,254) RPS(L),L
  254 FORMAT(G20.12,'  UPPER LIMIT FOR INTERVAL ',I2)
  260 CONTINUE
      WRITE(6,262) NC, (RPS(L1), L1=1,NINT)
```

```
262 FORMAT(1X/1X/' FREQUENCY TABLE          N = ',I4,/1X/
   *' VARIABLE                 UPPER LIMIT OF INTERVAL',/,
   *'                          ',10G10.3)
    DO 280 J=1,NV
    J1=J+3
    DO 264 L=1,NINT
    NPS(L)=0
    FPS(L)=0.D0
264 FPS(L)=0.D0
    SUM=0.D0
    DO 270 I=1,NC
    SUM=SUM+Q(I)
    DO 265 L=1,NINT
    IF ( DB(I,J) .GT. RPS(L) ) GO TO 265
    NPS(L)=NPS(L)+1
    FPS(L)=FPS(L)+Q(I)
    GO TO 270
265 CONTINUE
270 CONTINUE
    DO 271 L=1,NINT
271 FPS(L)=100.D0*FPS(L)/SUM
    WRITE(6,272) LD(J1,11),(FPS(L), L=1,NINT)
272 FORMAT(1X/,4X,I2,4X,' % IN POP:',10F10.1)
    WRITE(6,274) (NPS(L), L=1,NINT)
274 FORMAT(10X,'# IN SAMP:',10I10)
280 CONTINUE
    GO TO 240
300 WRITE(8,310) IOPT
310 FORMAT(I2,'  NEW ANALYSIS OPTION',/,' 0  OK')
    RETURN
    END
    SUBROUTINE WRITE(LD,NL,DS,NS,DP,NP,DB,NB,FMT,TBS)
C
C  PURPOSE: TO WRITE OUT DATA
C
    IMPLICIT REAL*8(A-H,O-Z)
    DIMENSION LD(NL,12),DS(NS,100),DP(NP,100),DB(NB,100),
   *TBS(3,15),FMT(5)

    CALL WRIT1(LD,NL,DS,NS,1,DB,NB,FMT,TBS)
    RETURN
    END
    SUBROUTINE ESTG1(LD,NL,DS,NS,DB,NB,Q,SV,B,B1,SO,KA,F,IOPT,IER)
C
C PURPOSE: TO CALCULATE WLS ESTIMATES OF A HETEROSCEDASTIC REGRESSION MODEL
C ASSUMING THAT SIGMA IS PROPORTIONAL TO E(Y), USING MADDALA'S
C APPROACH - FILL IN REFERENCE.
C LD IS THE HOUSEKEEPING ARRAY WITH EXACTLY NK ROWS.
C DS IS THE SAMPLE DATA BASE WITH NS ROWS, AND DB IS THE BUFFER
C DATASET WITH NB ROWS.  Q IS A VECTOR USED FOR BLU WEIGHTS AND
C SV IS A VECTOR USED TO STORE SINGULAR VALUES AND FOR WORKSPACE
C FOR THE SINGULAR VALUE DECOMPOSITION ALGORITHM.
C ON OUTPUT, B IS THE VECTOR OF PRIMARY COEFFICIENTS, B1 IS THE
C VECTOR OF COEFFICIENTS FROM THE PRECEEDING ITERATION,

C SO IS THE SCALE PARAMETER OF THE SECONDARY MODEL.
C ON INPUT KA IS THE MAXIMUM NUMBER OF ITERATIONS; ON OUTPUT KA
C IS THE ACTUAL NUMBER OF ITERATIONS.
C ON INPUT, F IS A SECOND LIMIT FOR THE NUMBER OF ITERATIONS
C BASED ON THE F-STATISTIC MEASURING THE SIGNIFICANCE OF THE
```

```
C CHANGE IN THE BETA COEFFICIENTS.
C THE ITERATIONS STOP WHENEVER THIS F-STATISTIC IS LESS THAN F.
C ON OUTPUT F IS THE LAST VALUE OF THIS F-STATISTIC.
C FOR IOPT=1,INTERMEDIATE ESTIMATES ARE NOT PRINTED;
C FOR IOPT=2, THEY ARE PRINTED.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),DS(NS,100),DB(NB,100),Q(NS),SV(NL),
     *B(NL),B1(NL)
C INITIALIZE
      IER=0
      KLIM=KA
      KA=1
      FCRIT=F
      NX=LD(3,5)
      JY=NX+1
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
      NC=LD(1,3)
      DO 50 J=1,NX
   50 B1(J)=0.D0
      GO TO 110
C START LOOP
  100 KA=KA+1
      DO 101 J=1,NX
  101 B1(J)=B(J)
C PRIMARY (WLS) REGRESSION
      CALL RESID(LD,NL,DS,NS,1,DB,NB,B,Q,2,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
      DO 102 I=1,NC
      DO 102 J=1,JY
  102 DB(I,J)=DB(I,J)/Q(I)
  110 CALL MINFIT(NB,NC,NX,DB,SV,1,DB(1,JY),IER,SV(NX+1))
      IF (IER .EQ. 0) GO TO 116
  112 WRITE(6,114) IER
  114 FORMAT(1X/' ERROR MESSAGE ',I6,' WAS OBTAINED FROM ',/,
     *' THE SINGULAR VALUE DECOMPOSITION IN THE PRIMARY EQUATION',/,
     *' PROBABLY DUE TO MULTICOLINEARITY')
      IER=1
      RETURN
  116 DO 118 I=1,NX
      IF (SV(I) .GT. 0.D0) GO TO 118
      WRITE(6,117)
  117 FORMAT('PERFECT MULTICOLINEARITY IN THE PRIMARY EQUATION')
      IER=1
      RETURN
  118 CONTINUE
      SSR=0.D0
      DO 119 I=1,NX
      SSR=SSR+DB(I,JY)**2
      B(I)=0.D0
      DO 119 J=1,NX
  119 B(I)=B(I)+DB(I,J)*DB(J,JY)/SV(J)
      SST=0.D0
      DO 120 I=1,NC
  120 SST=SST+DB(I,JY)**2
      SO=DSQRT((SST-SSR)/(NC-NX))
C  CALCULATE THE FSTAT TESTING HO:TRUE COEFFICIENTS EQUAL THE
C  ESTIMATES OBTAINED ON THE PRECEDING ITERATION, IE THAT THE
C  REVISIONS ARE NOT SIGNIFICANTLY DIFFERENT FROM ZERO.
```

```
C  REF:JOHNSON AND WICHERN, APPLIED MULTIVARIATE STATISTICS,
C      PRENTICE HALL, 1982, P. 304.
      F=0.D0
      DO 130 I=1,NX
      DO 130 J=1,NX
      DO 130 K=1,NX
  130 F=F+(B(I)-B1(I))*(B(J)-B1(J))*DB(I,K)*DB(J,K)*SV(K)**2
      F=F/(NX*SO**2)
      IF (IOPT .EQ. 1 ) GO TO 230
      WRITE(6,220) KA
  220 FORMAT(1X/1X/' ITERATION  ',I4)
      DO 222 I=1,NX
  222 WRITE(6,224) I,B(I)
  224 FORMAT(' BETA FOR VARIABLE ',I3,' IS ',G14.6)
      WRITE(6,226) SO,F
  226 FORMAT(' SO = ',G14.6,'  F-STAT = ',G14.6)
C CHECK STOPPING RULE
  230 IF(KA.EQ.1) GO TO 100
      IF(KA .GE. KLIM) RETURN
      IF (F .LT. FCRIT) RETURN
      GO TO 100
      END
C *************************************************************
      SUBROUTINE SIGMA(LD,NL,D,N,K,DB,NB,G,B,SO,S,IER)
C
C PURPOSE: TO CALCULATE THE MODEL-BASED STANDARD DEVIATION S(I)
C USING THE COEFFICIENTS SPECIFIED IN G.
C THIS VERSION IS MODIFIED TO HANDLE THE SPECIAL HETEROSCEDASTICITY
C MODEL IN WHICH SIGMA IS EQUAL TO SO*E(Y).  THIS OPTION IS INDICATED
C BY LD(3,6)=0 AND LD(4,6)=1.
C LD IS THE HOUSEKEEPING ARRAY WITH NL ROWS.
C D IS THE DATABASE WHICH CAN BE EITHER THE SAMPLE OR THE POPULATION
C N IS THE ROW DIMENSION OF D AND K IS THE COLUMN OF LD
C CORRESPONDING TO D.  DB IS THE BUFFER ARRAY WITH NB ROWS.
C G IS THE VECTOR OF COEFFICIENTS OF THE SECONDARY MODEL AND
C SO IS THE SCALOR COEFFICIENT OF THE MODEL. S IS THE OUTPUT
C VECTOR FOR THE STANDARD DEVIATIONS.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100),G(NL),B(NL),S(N)
      NT=LD(3,6)
      NC=LD(1,3)
      IF(NT.EQ.0 .AND. LD(4,6).EQ.1) GO TO 300
      IER=0
      IF ( NT .GT. 0 ) CALL TRANS(LD,NL,D,N,K,DB,NB,6,IER)
      IF(IER.EQ.0) GO TO 50
      WRITE(7,30)
   30 FORMAT(1X/' UNABLE TO EVALUATE THE STANDARD DEVIATION OF THE ',
     *'MODEL')
      RETURN
   50 DO 200 I=1,NC
      S(I)=SO
      IF (NT .EQ. 0) GO TO 200
      DO 100 J=1,NT
  100 S(I)=S(I)*DB(I,J)**G(J)
  200 CONTINUE
      RETURN
  300 CALL RESID(LD,NL,D,N,K,DB,NB,B,S,2,IER)
      DO 310 I=1,NC
  310 S(I)=SO*S(I)
```

```
      RETURN
      END
C ***********************************************************************
      SUBROUTINE ESTG(LD,NL,DS,NS,DB,NB,Q,SV,
     *B,G,SG,DG,SO,SE,KA,F,IOPT,IER)
C
C PURPOSE: TO CALCULATE MAXIMUM LIKELIHOOD ESTIMATES OF HETEROSCEDASTICITY
C COEFFICIENTS FOR A MULTIVARIATE LINEAR MODEL.
C LD IS THE HOUSEKEEPING ARRAY WITH EXACTLY NK ROWS.
C DS IS THE SAMPLE DATA BASE WITH NS ROWS, AND DB IS THE BUFFER
C DATASET WITH NB ROWS.  Q IS A VECTOR USED FOR BLU WEIGHTS AND
C SV IS A VECTOR USED TO STORE SINGULAR VALUES AND FOR WORKSPACE
C FOR THE SINGULAR VALUE DECOMPOSITION ALGORITHM.
C ON OUTPUT, B IS THE VECTOR OF PRIMARY COEFFICIENTS, G IS THE
C VECTOR OF SECONDARY COEFFICIENTS, SG IS THEIR ASYMPTOTIC
C STANDARD ERRORS, AND DG IS THEIR MOST RECENT ADJUSTMENT.
C SO IS THE SCALE PARAMETER OF THE SECONDARY MODEL.
C SE IS THE STANDARD ERROR OF THE PRIMARY REGRESSION, STANDARDIZED
C FOR COMPARABILITY BETWEEN HETEROSCEDASTICITY MODELS.
C ON INPUT KA IS THE MAXIMUM NUMBER OF ITERATIONS; ON OUTPUT KA
C IS THE ACTUAL NUMBER OF ITERATIONS.
C ON INPUT, F IS A SECOND LIMIT FOR THE NUMBER OF ITERATIONS
C BASED ON THE F-STATISTIC OF THE SECONDARY REGRESSION -
C THE ITERATIONS STOP WHENEVER THIS F-STATISTIC IS LESS THANF.
C ON OUTPUT F IS THE LAST VALUE OF THIS F-STATISTIC.
C FOR IOPT=1,INTERMEDIATE ESTIMATES ARE NOT PRINTED;
C FOR IOPT=2, THEY ARE PRINTED.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),DS(NS,100),DB(NB,100),Q(NS),SV(NL),
     *B(NL),G(NL),SG(NL),DG(NL)
C INITIALIZE
      IER=0
      KLIM=KA
      KA=0
      FCRIT=F
      NX=LD(3,5)
      NT=LD(3,6)
      JY=NX+1
      JYT=NT+1
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,G,IER)
      NC=LD(1,3)
      DO 25  I=1,NC
      DO 25  J=1,NT
      IF (DB(I,J) .GT. 0.D0) GO TO 25
      WRITE(6,24)
   24 FORMAT(1X/' SECONDARY VARIABLE IS NOT POSITIVE')
      IER=1
      RETURN
  25  DB(I,J)=DLOG(DB(I,J))
      DO 30 J=1,NT
  30  SG(J)=VM(NC,DB(1,J))
C START LOOP
  100 KA=KA+1
C PRIMARY (WLS) REGRESSION
      SO=0.D0
      DO 101 J=1,NT
 101  SO=SO-G(J)*SG(J)
      SO=DEXP(SO)
      CALL SIGMA(LD,NL,DS,NS,1,DB,NB,G,B,SO,Q,IER)
```

```
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
      DO 102 I=1,NC
      DO 102 J=1,JY
  102 DB(I,J)=DB(I,J)/Q(I)
      CALL MINFIT(NB,NC,NX,DB,SV,1,DB(1,JY),IER,SV(NX+1))
      IF (IER .EQ. O) GO TO 116
  112 WRITE(6,114) IER
  114 FORMAT(1X/' ERROR MESSAGE ',I6,' WAS OBTAINED FROM ',/,
     *' THE SINGULAR VALUE DECOMPOSITION IN THE PRIMARY EQUATION',/,
     *' PROBABLY DUE TO MULTICOLINEARITY')
      IER=1
      RETURN
  116 DO 118 I=1,NX
      IF (SV(I) .GT. O.DO) GO TO 118
      WRITE(6,117)
  117 FORMAT('PERFECT MULTICOLINEARITY IN THE PRIMARY EQUATION')
      IER=1
      RETURN
  118 CONTINUE
 200  DO 210 I=1,NX
      B(I)=O.DO
      DO 210 J=1,NX
  210 B(I)=B(I)+DB(I,J)*DB(J,JY)/SV(J)
      IF (IOPT .EQ. 1 ) GO TO 230
      WRITE(6,220) K
  220 FORMAT(1X/1X/' ITERATION  ',I4)
      DO 222 I=1,NX
  222 WRITE(6,224) I,B(I)
  224 FORMAT(' BETA FOR VARIABLE ',I3,' IS ',G14.6)
C CALCULATE SQUARED RESIDUALS OF WLS REGRESSION
 230  CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
      DO 240 I=1,NC
      U=DB(I,JY)
      DO 235 J=1,NX
  235 U=U-DB(I,J)*B(J)
  240 DB(I,JYT)=(U/Q(I))**2

C CALCULATE DEPENDENT VARIABLE FOR SECONDARY REGRESSION
      T=VM(NC,DB(1,JYT))
      IF(T.GT.O.DO) GO TO 300
      WRITE(7,290)
  290 FORMAT(1X/' ERROR: UNABLE TO ESTIMATE THE SECONDARY EQUATION')
      IER=1
      RETURN
  300 SE=DSQRT(T)
      T=2.DO*T
      DO 310 I=1,NC
  310 DB(I,JYT)=DB(I,JYT)/T
C RUN THE SECONDARY REGRESSION AND REDBISE G
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,6,IER)
      DO 320 I=1,NC
      DO 320 J=1,NT
  320 DB(I,J)=DLOG(DB(I,J))-SG(J)
      CALL MINFIT(NB,NC,NT,DB,SV,1,DB(1,JYT),IER,SV(NT+1))
      IF (IER .EQ. O) GO TO 416
  412 WRITE(6,414) IER
  414 FORMAT(1X/' ERROR MESSAGE ',I6,' OBTAINED FROM SINGULAR VALUE',
     *' DECOMPOSITION IN THE SECONDARY EQUATION',/,
```

```
     *' PROBABLY DUE TO MULTICOLINEARITY')
      IER=1
      RETURN
  416 DO 418 I=1,NT
      IF (SV(I) .GT. O.DO) GO TO 418
      WRITE(6,417)
  417 FORMAT('PERFECT MULTICOLINEARITY OBTAINED IN THE SECONDARY ',
     *'EQUATION')
      IER=1
      RETURN
  418 CONTINUE
  419 SSR=O.DO
      DO 425 I=1,NT
      SSR=SSR+DB(I,JYT)**2
      DG(I)=O.DO
      DO 422 J=1,NT
  422 DG(I)=DG(I)+DB(I,J)*DB(J,JYT)/SV(J)
  425 G(I)=G(I)+DG(I)
      IF ( IOPT .EQ. 1 ) GO TO 437
      DO 430 I=1,NT
  430 WRITE(6,435) I,G(I),DG(I)
  435 FORMAT(' FOR VARIABLE ',I2,' GAMMA IS ',G14.6,' DELGAM IS ',G14.6)
C TEST FOR TERMINATION
  437 SST=O.DO
      DO 440 I=1,NC
  440 SST=SST+DB(I,JYT)**2
      F=(SSR/NT)/((SST-SSR)/(NC-NT))
      IF (KA .GE. KLIM) GO TO 500
      IF (F .LT. FCRIT) GO TO 500
      GO TO 100
C CALCULATE GO
 500  T=DLOG(SE)
      DO 510 J=1,NT
 510  T=T-G(J)*SG(J)
      G(NT+1)=T
      SO=DEXP(T)
C CALCULATE THE STANDARD ERROR OF GO
      T=1.DO/NC
      DO 520 J=1,NT
      DO 520 K=1,NT
      DO 520 L=1,NT
 520  T=T+SG(J)*SG(K)*DB(J,L)*DB(K,L)/SV(L)**2
      SG(NT+1)=DSQRT(T/2.DO)
C CALCULATE THE STANDARD ERRORS OF G
      DO 550 J=1,NT
      T=O.DO
      DO 540 L=1,NT
 540  T=T+(DB(J,L)/SV(L))**2
 550  SG(J)=DSQRT(T/2.DO)
      RETURN
      END
C ***************************************************************
      SUBROUTINE TRANS(LD,NL,D,N,K,DB,NB,KB,IER)
C
C PURPOSE: TO EXPAND THE VARIABLES AVAILABLE FOR ANALYSIS
C BY CREATING TRANSFORMATIONS OF THE VARIABLES IN THE DATA ARRAY D.
C THE RESULT VARIABES ARE PLACED IN THE BUFFER ARRAY DB.
C
C TRANS WILL PUT OUT A SUBSET OF CASES BASED ON THE LAST COLUMN
C OF D. IF THIS COLUMN'S ENTRY IS O.DO THE CASE IS SKIPPED.
```

```
C OTHERWISE, IT IS MOVED TO DB.  LD(1,3) RECORDS THE ACTUAL
C NUMBER OF CASES IN DB.
C
C THIS VERSION CREATES NO NEW VARIABLES EXCEPT THE UNIT VARIABLE 1
C AND THE CASE NUMBER VARIABLE V99.
C
C LD IS THE HOUSEKEEPING ARRAY AND NL IS ITS EXACT ROW DIMENSION.
C D IS THE INPUT DATA ARRAY, EITHER DS OR DP, AND N IS ITS
C ROW DIMENSION, EITHER NS OR NP.  K IS THE COLUMN OF LD
C CORRESPONDING TO D, EITHER 1 OR 2.
C DB IS THE OUTPUT DATA ARRAY AND NB IS ITS ROW DIMENSION.
C KB IS THE COLUMN OF LD IDENTIFYING THE DESIRED VARIABLES, E.G.
C USE K=5 TO RETRIEVE THE PRIMARY EXPLANATORY VARIABLES.
C IER IS O IF ALL VARIABLES ARE SUCCESSFULLY RETRIEVED; 1,2, OR 3
C DEPENDING ON THE PROBLEM IF THE RETRIEVAL IS UNSUCCESSFUL.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100)
      IER=0
      NC=LD(1,K)
      NVB=LD(3,KB)
      IF(NVB.EQ.O) RETURN
      JD=LD(2,K)
      DO 1000 JB=1,NVB
      IB=0
      LVB=LD(JB+3,KB)
      CALL INDX(LD,NL,3,LVB,J,&2,&4)
 2    IER=1
C     ERROR: THE VARIABLE WAS NOT FOUND AMONG THE LOGICAL VARIABLES.
      RETURN
 4    IT=LD(J+3,4)
      GO TO (10,20,30), IT
      IER=2
C     ERROR: THE TRANS CODE IS GREATER THAN THE NUMBER OF ENTRIES
C            IN THE "GO TO" LIST.
      RETURN
 10   DO 12 I=1,NC
      IF ( D(I,JD) .EQ. 0.DO ) GO TO 12
      IB=IB+1
      DB(IB,JB)=1.DO
   12 CONTINUE
      GO TO 1000
 20   CALL INDX(LD,NL,K,LVB,J,&22,&24)
 22   IER=3
C     ERROR: THE VARIABLE WAS NOT FOUND IN THE INPUT DATA BASE.
      RETURN
 24   DO 26 I=1,NC
      IF ( D(I,JD) .EQ. 0.DO ) GO TO 26
      IB=IB+1
      DB(IB,JB)=D(I,J)
   26 CONTINUE
      GO TO 1000
   30 DO 32 I=1,NC
      IF ( D(I,JD) .EQ. 0.DO ) GO TO 32
      IB=IB+1
      DB(IB,JB)=I
   32 CONTINUE
      GO TO 1000
1000  CONTINUE
      LD(1,3)=IB
```

```
      RETURN
      END
C *************************************************************
      SUBROUTINE INDX(LD,NL,K,L,J,*,*)
C PURPOSE: TO LOCATE THE VARIABLE L IN THE KTH VARIABLE LIST OF LD.
C LD IS THE HOUSEKEEPING ARRAY AND NL IS ITS EXACT ROW DIMENSION.
C K IS THE COLUMN OF LD TO BE SEARCHED.  L IS THE LABEL OF THE
C VARIABLE TO BE FOUND. ON OUTPUT, J IS O IF L IS NOT FOUND.
C OTHERWISE, J IS THE INDEX OF L IN THE LIST K; I.E. J=1 IF L IS
C THE FIRST VARIABLE IN THE LIST, 2 FOR THE SECOND VARIABLE, ETC.
C RETURN 1 IS USED IF THE SEARCH IS UNSUCCESSFUL, OETHERWISE RETURN
C RETURN 2 IS USED.
C
      DIMENSION LD(NL,12)
      NV=LD(3,K)
      DO 5 J=1,NV
      IF (LD(J+3,K) .EQ. L) GO TO 10
 5    CONTINUE
      J=O
      RETURN 1
 10   RETURN 2
      END
C *************************************************************
      SUBROUTINE ESTB(LD,NL,DS,NS,DB,NB,Q,SV,
     *B,SB,TB,PB,SE,SE1,CHISQ,RSQ,RSQ1,F,SIGF,NC1,IOPT,IER)
C
C PURPOSE: TO CALCULATE A WLS REGRESSION OF Y ON X USING WEIGHT Q.
C THE ESTIMATES THAT ARE PRODUCED ARE BLU ASSUMING THAT
C THE VARIANCE OF THE RESIDUAL IS PROPORTIONAL TO 1/Q.
C MOST PARAMETERS ARE THE SAME AS IN ESTG. HOWEVER SE IS
C THE ORDINARY STANDARD ERROR OF THE TRANSFORMED REGRESSION WHILE
C SE1 IS THE STANDARD ERROR OF THE REGRESSION WITH GEOMETRICALLY
C STANDARDIZED WEIGHTS.
C NC1 IS THE NUMBER OF CASES ACTUALLY USED IN THE ANALYSIS;
C I.E. THOSE WITH POSITIVE WEIGHT Q.
C CHISQ IS THE CHI SQUARED STATISTIC FOR TESTING THE HYPOTHESIS THAT
C BOTH BETA AND GAMMA COEFFICIENTS ARE ALL EQUAL TO ZERO.
C RSQ IS THE CONVENTIONAL COEFFICIENT OF DETERMINATION FOR THE WLS,
C I.E. THE R-SQUARED OF THE TRANSFORMED REGRESSION;
C RSQ1 IS THE LIKELIHOOD-BASED STATISTIC FOR THE WLS REGRESSION.
C USE IOPT=1 FOR THE FULL OUTPUT.
C USE IOPT=2 IF THE FSTAT AND SB ARE NOT REQUIRED.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),DS(NS,100),DB(NB,100),Q(NS),SV(NL),
     *B(NL),SB(NL),TB(NL),PB(NL)
      NX=LD(3,5)
      JY=NX+1
      JX=NX+3
      IINT=O
      DO 50 J=4,JX
  50  IF ( LD(J,5) .EQ. 1 ) IINT=1
C IINT=O IF NO INTERCEPT IS USED, 1 OTHERWISE
      CALL TRANS(LD,NL,DS,NS,1,DB,NB,5,IER)
      CALL TRANS(LD,NL,DS,NS,1,DB(1,JY),NB,7,IER)
      NC=LD(1,3)
      NC1=O
      GM=O.DO
      YBAR=O.DO
      YBAR1=O.DO
```

```
      SST=O.DO
      SST1=O.DO
      DO 101 I=1,NC
      IF (Q(I) .GT. O.DO) GO TO 90
      Q2=O.DO
      GO TO 95
   90 Q2=DSQRT(Q(I))
      NC1=NC1+1
      YBAR1=YBAR1+DB(I,JY)
      SST1=SST1+DB(I,JY)**2
      GM=GM+DLOG(Q2)
   95 DO 100 J=1,JY
  100 DB(I,J)=DB(I,J)*Q2
      YBAR=YBAR+DB(I,JY)
  101 SST=SST+DB(I,JY)**2
      YBAR=YBAR/NC1
      YBAR1=YBAR1/NC1
      SST=SST-IINT*NC1*YBAR**2
      SST2=SST1-IINT*NC1*YBAR1**2
      GM=DEXP(2.DO*GM/NC1)
C NC1 IS THE NUMBER OF CASES WITH NONZERO WEIGHT Q
C YBAR1 IS THE MEAN OF THE UNWEIGHTED Y
C SST1 IS THE SUM OF SQUARES OF THE UNWEIGHTED Y
C SST2 IS THE UNWEIGHTED SUM OF SQUARES, ADJUSTED FOR THE INTERCEPT
C YBAR IS THE MEAN OF THE WEIGHTED Y, Y*SQRT(Q)
C SST IS THE SUM OF SQUARES OF THE WEIGHTED Y
C GM IS THE QEOMETRIC MEAN OF Q
C NDFN AND NDFD AND THE NUMERATOR AND DENOMINATOR DEGREES OF FREEDOM
      NDFN=NX-IINT
      NDFD=NC1-NX
      CALL MINFIT(NB,NC,NX,DB,SV,1,DB(1,JY),IER,SV(NX+1))
      IF (IER .EQ. O) GO TO 116
  112 WRITE(6,114) IER
  114 FORMAT(1X/' ERROR MESSAGE ',I6,' OBTAINED FROM SINGULAR VALUE',
     *' DECOMPOSITION',/,
     *'PROBABLY DUE TO MULTICOLINEARITY')
      IER=1
      RETURN
  116 DO 118 I=1,NX
      IF (SV(I) .GT. O.DO) GO TO 118
      WRITE(6,117)
  117 FORMAT('PERFECT MULTICOLINEARITY OBTAINED')
      IER=1
      RETURN
  118 CONTINUE
      SSR=O.DO
      DO 119 I=1,NX
      SSR=SSR+DB(I,JY)**2
      B(I)=O.DO
      DO 119 J=1,NX
  119 B(I)=B(I)+DB(I,J)*DB(J,JY)/SV(J)
      SSR=SSR-IINT*NC1*YBAR**2
      SSE=SST-SSR
      SE=DSQRT(SSE/NDFD)
      IF (IOPT .EQ. 2) RETURN
      IF ( NDFN .GT. O .AND. NDFD .GT. O ) GO TO 126
      F=O.DO
      SIGF=1.DO
      GO TO 129
  126 IF ( SSE .GT. O .AND. SSR .GT. O ) GO TO 128
```

```
      F=9999.
      SIGF=1.DO
      GO TO 129
  128 F=(SSR/NDFN)/(SSE/NDFD)
      CALL MDFD(F,NDFN,NDFD,SIGF,IER)
      SIGF=1.DO-SIGF
  129 DO 140 I=1,NX
      SB(I)=O.DO
      DO 130 J=1,NX
  130 SB(I)=SB(I)+(DB(I,J)/SV(J))**2
      IF(SB(I).GT.O.DO) GO TO 135
      TB(I)=9999.
      PB(I)=1.DO
      GO TO 140
  135 SB(I)=DSQRT(SB(I))*SE
      TB(I)=B(I)/SB(I)
      FB=TB(I)**2
      CALL MDFD(FB,1,NDFD,PB(I),IER)
      PB(I)=1.DO-PB(I)
  140 CONTINUE
      IF (SE .GT. O.DO ) GO TO 145
      CHISQ=9999.
      SE1=O.DO
      RSQ=O.DO
  145 SE1=SE/DSQRT(GM)
      CHISQ=-NC1*DLOG(SSE/(SST1*GM))
      RSQ= 1.DO - SSE/SST
      RSQ1= 1.DO - SSE/(SST2*GM)
      RETURN
      END
C *****************************************************************
      SUBROUTINE INVAR(LD,NL,K1,K2,TBS,IOPT,IER)
C
C PURPOSE: TO READ IN A LIST OF VARIABLES AND TO CHECK THEIR
C AVAILABILITY IN THE DATA BASE.
C LD IS THE HOUSEKEEPING ARRAY WITH NL ROWS. K1 IS THE COLUMN OF LD
C THAT SPECIFIES THE AVAILABLE VARIABLES.  USE K1=O FOR NO CHECK.
C K2 IS THE COLUMN OF LD  THAT IS TO CONTAIN THE NEW LIST.
C TBS IS A DESCRIPTIVE PHRASE FOR THE VARIABLES
C INITIALIZED IN THE MAIN PROGRAM.
C USE IOPT=1 TO CONFIRM THE ACCURACY OF THE LIST; IOPT=2
C TO SKIP THE CONFIRMATION. IER=O ON OUTPUT IF ALL VARIABLES
C ARE FOUND; IER=1 IF NOT.
C USE IOPT=3 TO ALLOW NO VARIABLE SPECIFICATION AND NO CONFORMATION.
C USE IOPT=4 TO ALLOW NO VARIABLES AND TO CONFIRM.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),TBS(3)
      IER=O
      NVT=LD(2,K2)
    1 J2=NVT+3
      DO 2 J=4,J2
    2 LD(J,K2)=O
      NBLK=(NVT-1)/20 + 1
      JPR=O
      DO 10 IBLK=1,NBLK
      J3=NVT-JPR
      IF ( J3 .EQ. O ) GO TO 12
      IF ( J3 .GT. 20 ) J3=20
      WRITE(7,5) (TBS(K), K=1,3)
```

```
    5 FORMAT(1X/' ',3A8,/,'&?')
      J1=JPR+4
      J2=JPR+J3+3
      JPR=JPR+J3
      READ(5,8) (LD(J,K2),J=J1,J2)
    8 FORMAT(20(I2,1X))
      IF ( LD(J2,K2).EQ. O ) GO TO 12
   10 CONTINUE
   12 CONTINUE
      NV=O
      DO 15 J=4,J2
      IF(LD(J,K2) .EQ. O) GO TO 20
 15   NV=NV+1
 20   LD(3,K2)=NV
      IF ( IOPT .LE. 2 .AND. NV .EQ. O ) GO TO 1
      IF ( IOPT.EQ.4 .AND. NV.EQ.O ) GO TO 35
      IF ( IOPT.EQ.3 .AND. NV.EQ.O ) GO TO 39
      IF( K1 .EQ. O ) GO TO 41
      J1=LD(3,K1)+3
      J2=NV+3
      DO 34 I=4,J2
      DO 30 J=4,J1
      IF ( LD(I,K2) .EQ. LD(J,K1) ) GO TO 34
 30   CONTINUE
      IER=1
 34   CONTINUE
      GO TO 41
   35 WRITE(7,36)
   36 FORMAT(1X/' NO VARIABLE SPECIFIED')
      CALL CHECK(&1,&37)
   37 WRITE(8,38)
   38 FORMAT(' O',/,' O  OK')
   39 RETURN
   41 IF ( IOPT .EQ. 1 .OR. IOPT .EQ. 4 ) GO TO 42
      RETURN
 42   WRITE(7,45) (TBS(K), K=1,3)
 45   FORMAT(1X/' ',3A8)
      WRITE(7,50) (LD(J,K2),J=4,J2)
 50   FORMAT(20(1X,I2))
      CALL CHECK(&1,&100)
 100  WRITE(8,110) (LD(J,K2),J=4,J2)
 110  FORMAT(20(I2,1X))
      J3=NV-20*(NV/20)
      IF(J3.EQ.O) WRITE(8,115)
  115 FORMAT('OO')
      WRITE(8,120)
 120  FORMAT(' O  OK')
      RETURN
      END
C ***************************************************************
      SUBROUTINE CHECK(*,*)
 10   WRITE(7,20)
 20   FORMAT(1X/' CORRECT? (OO=YES,O1=NO)'/'&?')
      READ(5,40) ICHECK
 40   FORMAT(I2)
      IF(ICHECK .EQ. O) RETURN 2
      IF(ICHECK .EQ. 1) RETURN 1
      WRITE(7,60)
 60   FORMAT('UNRECOGNIZABLE RESPONSE TO "CORRECT?"')
      GO TO 10
```

```
      END
C ***************************************************************
C *************************************************************
C     'SUBROUTINE' VM
      REAL FUNCTION VM*8(N,X)
C
C  PURPOSE: TO CALCULATE THE MEAN OF THE N ENTRIES IN THE VECTOR X
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(N)
      VM=O.DO
      DO 1 I=1,N
      VM=VM+X(I)
    1 CONTINUE
      VM=VM/N
      RETURN
      END
C ******************************************************************
C
      SUBROUTINE COMNT(TITLE)
C
C PURPOSE: TO ENTER A COMMENT INTO THE OUTPUT
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION TITLE(9,10)
  100 WRITE(7,110)
  110 FORMAT(1X/' NUMBER OF LINES (01 TO 10)',/,'&?')
      READ(5,120) N
  120 FORMAT(I2)
      IF(N .GE. 1 .AND. N .LE. 10) GO TO 150
      WRITE(7,130) N
  130 FORMAT(1X/' ',I3,' LINES ARE NOT ALLOWED',/,
     *' ENTER BETWEEN 01 AND 10')
      GO TO 100
  150 DO 170 J=1,N
      WRITE(7,160) J
  160 FORMAT(1X/' LINE ',I2,/,'&?')
  170 READ(5,200) (TITLE(I,J),I=1,9)
  200 FORMAT(9A8)
      WRITE(7,200) ((TITLE(I,J),I=1,9),J=1,N)
      CALL CHECK(&100,&300)
  300 WRITE(8,310) N
  310 FORMAT(I2,'  NUMBER OF LINES')
      WRITE(8,200) ((TITLE(I,J),I=1,9),J=1,N)
      WRITE(8,320)
  320 FORMAT(' O  OK')
      WRITE(6,200) ((TITLE(I,J),I=1,9),J=1,N)
      RETURN
      END
C ******************************************************************
      SUBROUTINE MINFIT(NM,M,N,A,W,IP,B,IERR,RV1)
C
      INTEGER I,J,K,L,M,N,II,IP,I1,KK,K1,LL,L1,M1,NM,ITS,IERR
      REAL*8 A(NM,N),W(N),B(NM,IP),RV1(N)
      REAL*8 C,F,G,H,S,X,Y,Z,EPS,SCALE,MACHEP
      REAL*8 DSQRT,DMAX1,DABS,DSIGN
C
C     THIS PROCEDURE IS A TRANSLATION OF THE ALGOL PROCEDURE MINFIT,
C     NUM. MATH. 14, 403-420(1970) BY GOLUB AND REINSCH.
C     HANDBOOK FOR AUTO. COMP., VOL II-LINEAR ALGEBRA, 134-151(1971).
```

```
C
C     THIS PROCEDURE DETERMINES, TOWARDS THE SOLUTION OF THE LINEAR
C                                                             T
C     SYSTEM AX=B, THE SINGULAR VALUE DECOMPOSITION A=USV  OF A REAL
C                                                T
C     M BY N RECTANGULAR MATRIX, FORMING U B RATHER THAN U.  HOUSEHOLDER
C     BIDIAGONALIZATION AND A VARIANT OF THE QR ALGORITHM ARE USED.
C
C     ON INPUT:
C
C        NM MUST BE SET TO THE ROW DIMENSION OF TWO-DIMENSIONAL
C          ARRAY PARAMETERS AS DECLARED IN THE CALLING PROGRAM
C          DIMENSION STATEMENT.  NOTE THAT NM MUST BE AT LEAST
C          AS LARGE AS THE MAXIMUM OF M AND N;
C
C        M IS THE NUMBER OF ROWS OF A AND B;
C
C        N IS THE NUMBER OF COLUMNS OF A AND THE ORDER OF V;
C
C        A CONTAINS THE RECTANGULAR COEFFICIENT MATRIX OF THE SYSTEM;
C
C        IP IS THE NUMBER OF COLUMNS OF B.  IP CAN BE ZERO;
C
C        B CONTAINS THE CONSTANT COLUMN MATRIX OF THE SYSTEM
C          IF IP IS NOT ZERO.  OTHERWISE B IS NOT REFERENCED.
C
C     ON OUTPUT:
C
C        A HAS BEEN OVERWRITTEN BY THE MATRIX V (ORTHOGONAL) OF THE
C          DECOMPOSITION IN ITS FIRST N ROWS AND COLUMNS.  IF AN
C          ERROR EXIT IS MADE, THE COLUMNS OF V CORRESPONDING TO
C          INDICES OF CORRECT SINGULAR VALUES SHOULD BE CORRECT;
C
C        W CONTAINS THE N (NON-NEGATIVE) SINGULAR VALUES OF A (THE
C          DIAGONAL ELEMENTS OF S).  THEY ARE UNORDERED.  IF AN
C          ERROR EXIT IS MADE, THE SINGULAR VALUES SHOULD BE CORRECT
C          FOR INDICES IERR+1,IERR+2,...,N;
C
C                                       T
C        B HAS BEEN OVERWRITTEN BY U B.  IF AN ERROR EXIT IS MADE,
C                          T
C          THE ROWS OF U B CORRESPONDING TO INDICES OF CORRECT
C          SINGULAR VALUES SHOULD BE CORRECT;
C
C        IERR IS SET TO
C          ZERO       FOR NORMAL RETURN,
C          K          IF THE K-TH SINGULAR VALUE HAS NOT BEEN
C                     DETERMINED AFTER 30 ITERATIONS;
C
C        RV1 IS A TEMPORARY STORAGE ARRAY.
C
C     QUESTIONS AND COMMENTS SHOULD BE DIRECTED TO B. S. GARBOW,
C     APPLIED MATHEMATICS DIVISION, ARGONNE NATIONAL LABORATORY
C
C     ------------------------------------------------------------------
C
C     :::::::::: MACHEP IS A MACHINE DEPENDENT PARAMETER SPECIFYING
C                THE RELATIVE PRECISION OF FLOATING POINT ARITHMETIC.
C                MACHEP = 16.0D0**(-13) FOR LONG FORM ARITHMETIC
C                ON S360 ::::::::::
```

```
      DATA MACHEP/Z3410000000000000/
C
      IERR = 0
C     :::::::::: HOUSEHOLDER REDUCTION TO BIDIAGONAL FORM ::::::::::
      G = 0.0D0
      SCALE = 0.0D0
      X = 0.0D0
C
      DO 300 I = 1, N
         L = I + 1
         RV1(I) = SCALE * G
         G = 0.0D0
         S = 0.0D0
         SCALE = 0.0D0
         IF (I .GT. M) GO TO 210
C
         DO 120 K = I, M
  120    SCALE = SCALE + DABS(A(K,I))
C
         IF (SCALE .EQ. 0.0D0) GO TO 210
C
         DO 130 K = I, M
            A(K,I) = A(K,I) / SCALE
            S = S + A(K,I)**2
  130    CONTINUE
C
         F = A(I,I)
         G = -DSIGN(DSQRT(S),F)
         H = F * G - S
         A(I,I) = F - G
         IF (I .EQ. N) GO TO 160
C
         DO 150 J = L, N
            S = 0.0D0
C
            DO 140 K = I, M
  140       S = S + A(K,I) * A(K,J)
C
            F = S / H
C
            DO 150 K = I, M
               A(K,J) = A(K,J) + F * A(K,I)
  150    CONTINUE
C
  160    IF (IP .EQ. 0) GO TO 190
C
         DO 180 J = 1, IP
            S = 0.0D0
C
            DO 170 K = I, M
  170       S = S + A(K,I) * B(K,J)
C
            F = S / H
C
            DO 180 K = I, M
               B(K,J) = B(K,J) + F * A(K,I)
  180    CONTINUE
C
  190    DO 200 K = I, M
  200    A(K,I) = SCALE * A(K,I)
```

```
C
  210     W(I) = SCALE * G
          G = 0.0D0
          S = 0.0D0
          SCALE = 0.0D0
          IF (I .GT. M .OR. I .EQ  N) GO TO 290
C
          DO 220 K = L, N
  220     SCALE = SCALE + DABS(A(I,K))
C
          IF (SCALE .EQ. 0.0D0) GO TO 290
C
          DO 230 K = L, N
             A(I,K) = A(I,K) / SCALE
             S = S + A(I,K)**2
  230     CONTINUE
C
          F = A(I,L)
          G = -DSIGN(DSQRT(S),F)
          H = F * G - S
          A(I,L) = F - G
C
          DO 240 K = L, N
  240     RV1(K) = A(I,K) / H
C
          IF (I .EQ. M) GO TO 270
C
          DO 260 J = L, M
             S = 0.0D0
C
             DO 250 K = L, N
  250        S = S + A(J,K) * A(I,K)
C
             DO 260 K = L, N
                A(J,K) = A(J,K) + S * RV1(K)
  260     CONTINUE
C
  270     DO 280 K = L, N
  280     A(I,K) = SCALE * A(I,K)
C
  290     X = DMAX1(X,DABS(W(I))+DABS(RV1(I)))
  300 CONTINUE
C     :::::::::: ACCUMULATION OF RIGHT-HAND TRANSFORMATIONS.
C                FOR I=N STEP -1 UNTIL 1 DO -- ::::::::::
      DO 400 II = 1, N
         I = N + 1 - II
         IF (I .EQ. N) GO TO 390
         IF (G .EQ. 0.0D0) GO TO 360
C
         DO 320 J = L, N
C     :::::::::: DOUBLE DIVISION AVOIDS POSSIBLE UNDERFLOW ::::::::::
  320    A(J,I) = (A(I,J) / A(I,L)) / G
C
         DO 350 J = L, N
            S = 0.0D0
C
            DO 340 K = L, N
  340       S = S + A(I,K) * A(K,J)
C
            DO 350 K = L, N
```

```
            A(K,J) = A(K,J) + S * A(K,I)
  350    CONTINUE
C
  360    DO 380 J = L, N
            A(I,J) = 0.0D0
            A(J,I) = 0.0D0
  380    CONTINUE
C
  390    A(I,I) = 1.0D0
         G = RV1(I)
         L = I
  400 CONTINUE
C
      IF (M .GE. N .OR. IP .EQ. 0) GO TO 510
      M1 = M + 1
C
      DO 500 I = M1, N
C
         DO 500 J = 1, IP
            B(I,J) = 0.0D0
  500 CONTINUE
C     :::::::::: DIAGONALIZATION OF THE BIDIAGONAL FORM ::::::::::
  510 EPS = MACHEP * X
C     :::::::::: FOR K=N STEP -1 UNTIL 1 DO -- ::::::::::
      DO 700 KK = 1, N
         K1 = N - KK
         K = K1 + 1
         ITS = 0
C     :::::::::: TEST FOR SPLITTING.
C                FOR L=K STEP -1 UNTIL 1 DO -- ::::::::::
  520    DO 530 LL = 1, K
            L1 = K - LL
            L = L1 + 1
            IF (DABS(RV1(L)) .LE. EPS) GO TO 565
C     :::::::::: RV1(1) IS ALWAYS ZERO, SO THERE IS NO EXIT
C                THROUGH THE BOTTOM OF THE LOOP ::::::::::
            IF (DABS(W(L1)) .LE. EPS) GO TO 540
  530    CONTINUE
C     :::::::::: CANCELLATION OF RV1(L) IF L GREATER THAN 1 ::::::::::
  540    C = 0.0D0
         S = 1.0D0
C
         DO 560 I = L, K
            F = S * RV1(I)
            RV1(I) = C * RV1(I)
            IF (DABS(F) .LE. EPS) GO TO 565
            G = W(I)
            H = DSQRT(F*F+G*G)
            W(I) = H
            C = G / H
            S = -F / H
            IF (IP .EQ. 0) GO TO 560
C
            DO 550 J = 1, IP
               Y = B(L1,J)
               Z = B(I,J)
               B(L1,J) = Y * C + Z * S
               B(I,J) = -Y * S + Z * C
  550       CONTINUE
C
```

```
  560    CONTINUE
C     :::::::::: TEST FOR CONVERGENCE ::::::::::
  565    Z = W(K)
         IF (L .EQ. K) GO TO 650
C     :::::::::: SHIFT FROM BOTTOM 2 BY 2 MINOR ::::::::::
         IF (ITS .EQ. 30) GO TO 1000
         ITS = ITS + 1
         X = W(L)
         Y = W(K1)
         G = RV1(K1)
         H = RV1(K)
         F = ((Y - Z) * (Y + Z) + (G - H) * (G + H)) / (2.0D0 * H * Y)
         G = DSQRT(F*F+1.0D0)
         F = ((X - Z) * (X + Z) + H * (Y / (F + DSIGN(G,F)) - H)) / X
C     :::::::::: NEXT QR TRANSFORMATION ::::::::::
         C = 1.0D0
         S = 1.0D0
C
         DO 600 I1 = L, K1
            I = I1 + 1
            G = RV1(I)
            Y = W(I)
            H = S * G
            G = C * G
            Z = DSQRT(F*F+H*H)
            RV1(I1) = Z
            C = F / Z
            S = H / Z
            F = X * C + G * S
            G = -X * S + G * C
            H = Y * S
            Y = Y * C
C
            DO 570 J = 1, N
               X = A(J,I1)
               Z = A(J,I)
               A(J,I1) = X * C + Z * S
               A(J,I) = -X * S + Z * C
  570       CONTINUE
C
            Z = DSQRT(F*F+H*H)
            W(I1) = Z
C     :::::::::: ROTATION CAN BE ARBITRARY IF Z IS ZERO ::::::::::
            IF (Z .EQ. 0.0D0) GO TO 580
            C = F / Z
            S = H / Z
  580       F = C * G + S * Y
            X = -S * G + C * Y
            IF (IP .EQ. 0) GO TO 600
C
            DO 590 J = 1, IP
               Y = B(I1,J)
               Z = B(I,J)
               B(I1,J) = Y * C + Z * S
               B(I,J) = -Y * S + Z * C
  590       CONTINUE
C
  600    CONTINUE
C
         RV1(L) = 0.0D0
```

```
      RV1(K) = F
      W(K) = X
      GO TO 520
C    :::::::::: CONVERGENCE ::::::::::
 650  IF (Z .GE. 0.0D0) GO TO 700
C    :::::::::: W(K) IS MADE NON-NEGATIVE ::::::::::
      W(K) = -Z
C
      DO 690 J = 1, N
 690  A(J,K) = -A(J,K)
C
 700 CONTINUE
C
     GO TO 1001
C    :::::::::: SET ERROR -- NO CONVERGENCE TO A
C               SINGULAR VALUE AFTER 30 ITERATIONS ::::::::::
1000 IERR = K
1001 RETURN
C    :::::::::: LAST CARD OF MINFIT ::::::::::
     END
C **********************************************************
     SUBROUTINE MDFD(Z,N1,N2,POFF,IER)
C
C SOURCE - KENNEDY, WILLIAM J. JR. AND GENTLE JAMES E. (1980)
C        - STATISTICAL COMPUTING; NEW YORK: MARCEL DEKKER
C        - P. 114-115
C
C FUNCTION       - COMPUTE F PROBABILITY (1 - P-VALUE) USING METHOD
C                - SUGGESTED BY DAVIS AND KHALIL TO EVALUATE F CDF.
C
C PARAMETERS: N1- NUMERATOR DEGREES OF FREEDOM (INPUT)
C             N2- DENOMINATOR DEGREES OF FREEDOM (INPUT)
C              Z- UPPER LIMIT OF INTEGRATION
C           POFF- COMPUTED PROBABILITY (OUTPUT)
C            IER- ERROR INDICATOR
C               - 0 IF NORMAL TERMINATION
C               - 1 IF Z<0 (POFF=0)
C               - 2 IF N1 OR N2 IS NOT POSITIVE (POFF=0)
C               - 3 IF COMPUTED POFF IS >1 OR <0 (1 OR 0 IS RETURNED)
C
     IMPLICIT REAL*8 (A-H,O-Z)
     IER=0
     IF (Z .GT. 0.D0) GO TO 5
     POFF=0.D0
     IER=1
     RETURN
   5 IF (N1.GT.0 .AND. N2.GT.0) GO TO 10
     IER=2
     POFF=0.D0
     RETURN
  10 CONTINUE
     AN1=N1
     AN2=N2
     A=AN1*Z/(AN1*Z+AN2)
     A1=1.D0-A
     IF(A1 .LT. 0.1D-36) A1=0.1D-36
     D1=AN1*0.5D0
     D2=AN2*0.5D0
     D3=D1+D2-1.D0
     R=0.D0
```

```
      S1=O.DO
      S2=O.DO
      DEL=1.DO
      XM=1 DO
      XK=1.DO
      C=O.25DO
      PI=3.141592653589793DO
      N=N2
C NOTE BEGINNING OF MAJOR LOOP
   15 CONTINUE
C
C TO SEE IF DEGREES OF FREEDOM ARE ODD OR EVEN
C
      M=D2
      M=2*M
      IF (M .NE. N) GO TO 30
      N=D2-1
C
C IF DEGREES OF FREEDOM ARE EVEN SET N=(DF/2)-1
C
      IF (N .EQ. O) GO TO 25
      DO 20 I=1,N
      S1=DEL+S1*R
      D2=D2-1.DO
      D3=D3-1.DO
      TEM=A1/D2
      R=D3*TEM
      S2=(R+TEM)*S2
   20 CONTINUE
   25 S1=DEL+S1*R
      DEL=O.DO
      T=-1.DO
      D3=-1.DO
      S2=A*S2
      C=C+O.5DO
      GO TO 45
C
C IF DEGREES OF FREEDOM ARE ODD SET N=(DF-1)/2
C IF DEGREES OF FREEDOM EQUAL 1. DO NOT EXIT LOOP
C
   30 N=D2
      IF (N .EQ. O) GO TO 40
      DO 35 I=1,N
      S1=DEL+S1*R
      D2=D2-1.DO
      D3=D3-1.DO
      TEM=A1/D2
      R=D3*TEM
      S2=(R+TEM)*S2
   35 CONTINUE
   40 S1=XK*S1
      S2=XK*S2
      ART=DSQRT(A1)
      XM=XM*ART
      T=(XM-ART)/A1
      D3=-O.5DO
      XK=2.DO/PI
      C=C*2.ODO
   45 IF (C .GT. O.875DO) GO TO 50
      D2=D1
```

```
      D3=D2+D3
      S2=S1
      S1=0.D0
      A1=A
      IF (A1 .LT. 0.1D-36) A1=0.1D-36
      N=N1
      GO TO 15
C
C NOTE END OF MAJOR LOOP
C
   50 IF (C .LT. 1.125D0) DEL=4.D0/PI*DATAN(T)
      POFF=XM*(S2-S1)-DEL
      IF (0.D0 .LE. POFF .AND. 1.D0 .GE. POFF) RETURN
      IF (POFF .LT. 0.D0) POFF=0.D0
      IF (POFF .GT. 1.D0) POFF=1.D0
      IER=3
      RETURN
      END
C ****************************************************************
      SUBROUTINE SELECT(LD,NL,D,N,K,DB,NB,SL,TBS,IOPT,IER)
C
C PURPOSE: TO SELECT CASES TO BE USED IN ALL ANALYSIS OF THE
C DATA ARRAY D.  A CASE IS INCLUDED IN THE ANALYSIS ONLY IF EACH
C OF A SET OF SELECTOR VARIABLES IS BETWEEN SPECIFIED LIMITS.
C THIS ROUTINE HELPS THE USER TO SPECIFY THESE LIMITS, ALSO
C ALLOWS THE USER TO SPECIFY SPECIAL SELECTOR VARIABLES TO
C BE USED IN ADDITION TO THE REGULAR SELECTOR VARIABLES.
C
C THE REGULAR SELECTOR VARIABLES ARE ALL EXPLANATORY VARIABLES
C INCLUDED IN THE LATEST MODEL.  IF D IS THE SAMPLE DATA BASE DS,
C THEN THE DEPENDENT VARIABLE IS ALSO A REGULAR SELECTOR.
C
C ONCE THE SPECIAL SELECTOR VARIABLES AND LIMITS ARE SET, AN
C INDICATOR VARIABLE IS STORED IN THE LAST COLUMN OF D.  THIS
C INDICATOR VARIABLE IDENTIFIES THE CASES TO BE USED BY TAKING
C THE VALUE 1.D0 FOR THESE CASES AND 0.D0 FOR ALL CASES TO BE
C DROPPED.  THIS INDICATOR VARIABLE IS RECALCULATED WHENEVER
C A NEW MODEL IS SPECIFIED.  IT IS ALSO USED BY TRANS TO PUT
C THE DESIRED CASES INTO THE BUFFER.
C
C MOST OF THE ARGUMENTS OF SELECT ARE IDENTICAL TO THE ARGUMENTS
C OF TRANS.  SL IS THE ARRAY IN WHICH LIMITS ARE STORED.
C IOPT IS 1 IF THE ROUTINE IS TO OBTAIN SELECTOR VARIABLES OR
C LIMITS FROM THE USER, OR 2 IF THE ROUTINE IS TO UPDATE THE
C INDICATOR VARIABLE FOR A NEW MODEL.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100),SL(NL,2),TBS(3,14)
      IER=0
      IF ( IOPT .GE. 2 ) GO TO 2000
      IF(LD(1,1) .GT. 0 ) GO TO 50
      WRITE(7,20)
   20 FORMAT(1X/' YOU MUST READ IN A SAMPLE DATA BASE')
      RETURN
   50 CONTINUE
      J1=LD(3,3)+3
  100 WRITE(7,110)
  110 FORMAT(1X/1X/' ENTER CASE SELECTOR OPTION',/,
     *'    01 - SPECIFY A LOWER LIMIT FOR ALL VARIABLES',/,
     *'    02 - SPECIFY AN UPPER LIMIT FOR ALL VARIABLES',/,
```

```
     *'    03 - SPECIFY LIMITS FOR INDIVIDUAL VARIABLES',/,
     *'    04 - DISPLAY LIMITS',/,
     *'    05 - SPECIFY SPECIAL SELECTOR VARIABLES',/,
     *'    06 - NEW ANALYSIS OPTION',/,
     *'&?')
      READ(5,115) IOPT1
  115 FORMAT(I2)
      GO TO (1100,1200,1300,1400,1500,1600),IOPT1
      WRITE(7,120) IOPT1
  120 FORMAT(1X/' SELECTOR OPTION ',I2,' IS NOT RECOGNIZED')
      GO TO 100
 1100 WRITE(8,1102)
 1102 FORMAT(' 1  LOWER LIMIT FOR ALL VARIABLES')
      WRITE(7,1110)
 1110 FORMAT(1X/' ENTER LOWER LIMIT  (E.G. 0.)',/,'&?')
      READ(5,1120) C1
      WRITE(8,1120) C1
 1120 FORMAT(1G20 12)

      DO 1130 J=4,J1
 1130 SL(J,1)=C1
      GO TO 100
 1200 WRITE(8,1202)
 1202 FORMAT(' 2  UPPER LIMIT FOR ALL VARIABLES')
      WRITE(7,1210)
 1210 FORMAT(1X/' ENTER UPPER LIMIT  (E.G. 9999.)',/,'&?')
      READ(5,1220) C2
      WRITE(8,1220) C2
 1220 FORMAT(1G20.12)
      DO 1230 J=4,J1
 1230 SL(J,2)=C2
      GO TO 100
 1300 WRITE(8,1301)
 1301 FORMAT(' 3  LIMITS FOR INDIVIDUAL VARIABLES')
      DO 1380 J=4,J1
 1302 WRITE(7,1305)
 1305 FORMAT(1X/' ENTER VARIABLE  (ENTER 00 TO STOP)',/,'&?')
      READ(5,1310) L
      WRITE(8,1310) L
 1310 FORMAT(I2)
      IF (L .EQ. 0) GO TO 100
      CALL INDX(LD,NL,3,L,JB,&1330,&1340)
 1330 WRITE(7,1332) L
 1332 FORMAT(1X/' VARIABLE ',I2,' IS UNDEFINED')
      GO TO 1302
 1340 WRITE(7,1341)
 1341 FORMAT(1X/' ENTER.LOWER LIMIT  (E.G. 0.)',/,'&?')
      READ(5,1342) C1
      WRITE(8,1342) C1
 1342 FORMAT(1G20.12)
      WRITE(7,1343)
 1343 FORMAT(1X/' ENTER UPPER LIMIT  (E.G. 99999.)',/,'&?')
      READ(5,1342) C2
      WRITE(8,1342) C2
      IF (C1.LT.C2) GO TO 1350
      WRITE(7,1345) C1,C2
 1345 FORMAT(1X/' YOUR LOWER LIMIT',G20.12,/,
     *'SHOULD BE LESS THAN YOUR UPPER LIMIT',G20.12)
      GO TO 1302
 1350 SL(JB+3,1)=C1
```

```
      SL(JB+3,2)=C2
1380  CONTINUE
      GO TO 100
1400  WRITE(8,1402)
1402  FORMAT(' 4  DISPLAY LIMITS')
      WRITE(6,1410)
1410  FORMAT(1X/1X/' CASE SELECTION LIMITS ',/1X/,
     *' VARIABLE',4X,'LOWER LIMIT',9X,'UPPER LIMIT')
      DO 1430 J=4,J1
1430  WRITE(6,1432) LD(J,3),SL(J,1),SL(J,2)
1432  FORMAT(3X,I2,3X,2G20.12)
      GO TO 100
1500  WRITE(8,1502)
1502  FORMAT(' 5  SPECIAL SELECTOR VARIABLES')
      CALL INVAR(LD,NL,3,12,TBS(1,14),4,IER)
      JS=LD(3,12)+3
      IF (JS .GT. 3) GO TO 1530
      WRITE(6,1510)
1510  FORMAT(1X/1X/' NO SPECIAL SELECTOR VARIABLES ARE BEING USED')
      GO TO 100
1530  WRITE(6,1540) (LD(J,12), J=4,JS)
1540  FORMAT(1X/1X/
     *' SELECTOR VARIABLES: ',20(I2,1X))
      GO TO 100
1600  WRITE(8,1602)
1602  FORMAT(' 6  NEW ANALYSIS OPTION')
      RETURN
2000  NC=LD(1,K)
      IF( NC .GT. 0 ) GO TO 2008
      IF(IOPT.EQ.3) GO TO 2001
      RETURN
2001  IF( K .EQ. 2) GO TO 2004
      WRITE(6,2002)
2002  FORMAT(1X/' THE SAMPLE DATA BASE IS EMPTY')
      RETURN
2004  WRITE(6,2006)
2006  FORMAT(1X/' THE POPULATION DATA BASE IS EMPTY')
      RETURN
2008  JD=LD(2,K)
      JB=LD(2,3)
      DO 2010 I=1,NC
      D(I,JD)=1.DO
2010  DB(I,JB)=1.DO
      CALL SEL1(LD,NL,D,N,K,DB,NB,5,SL,IPOS,IER)
2020  CALL SEL1(LD,NL,D,N,K,DB,NB,6,SL,IPOS,IER)
      IF (IPOS .EQ. 0) GO TO 2030
      IF(K.EQ.1)WRITE(7,2015)
2015  FORMAT(1X/' ONE OF THE SECONDARY VARIABLES OF YOUR MODEL ',
     */,'IS NOT POSITIVE IN THE SAMPLE DATA BASE',/,
     *' YOU MUST SELECT CASES GREATER THAN ZERO OR SPECIFY NEW ',
     *'VARIABLES')
      IF(K.EQ.2)WRITE(7,2017)
2017  FORMAT(1X/' ONE OF THE SECONDARY VARIABLES OF YOUR MODEL ',
     */,'IS NOT POSITIVE IN THE POPULATION DATA BASE',/,
     *' YOU MUST SELECT CASES GREATER THAN ZERO OR SPECIFY NEW ',
     *'VARIABLES',/,
     *' BEFORE DESIGNING A NEW SAMPLING PLAN')
      IER=4
2030  IF (K .EQ. 1) CALL SEL1(LD,NL,D,N,K,DB,NB,7,SL,IPOS,IER1)
      CALL SEL1(LD,NL,D,N,K,DB,NB,12,SL,IPOS,IER1)
```

```
      SUM=0.D0
      DO 2040 I=1,NC
      SUM=SUM+DB(I,JB)
 2040 D(I,JD)=DB(I,JB)
      IF (IOPT.EQ  2) RETURN
      IF ( SUM .GT. 0.D0 ) GO TO 2060
      WRITE(7,2050)
 2050 FORMAT(1X/' YOU HAVE NO CASES FOR ANALYSIS ')
      IF (K.EQ.1) WRITE(7,2052)
 2052 FORMAT(' IN THE SAMPLE DATA BASE')
      IF (K.EQ.2) WRITE(7,2054)
 2054 FORMAT(' IN THE POPULATION DATA BASE')
      GO TO 2070
 2060 I=SUM
      WRITE(6,2062) I
 2062 FORMAT(1X/,I5,' CASES HAVE BEEN SELECTED')
      IF (K.EQ.1) WRITE(6,2064)
 2064 FORMAT(' FROM THE SAMPLE DATA BASE')
      IF (K EQ.2) WRITE(6,2066)
 2066 FORMAT(' FROM THE POPULATION DATA BASE')
 2070 CALL CHECK(&2075,&2080)
 2075 WRITE(8,2077)
 2077 FORMAT(' 1  NO')
      IER=9
      RETURN
 2080 WRITE(8,2082)
 2082 FORMAT(' 0  OK')
      IER=0
      RETURN
      END
C ********************************************************************
      SUBROUTINE SEL1(LD,NL,D,N,K,DB,NB,KB,SL,IPOS,IER)
C
C PURPOSE: TO IDENTIFY ANY CASES TO BE DELETED BASED ON THE
C SELECTOR VARIABLES IN COLUMN KB OF LD.  THIS IDENTIFIER VARIABLE
C IS STORED IN THE LAST COLUMN OF DB.  THIS ROUTINE IS USED
C REPEATEDLY BY 'SELECT' FOR DIFFERENT CHOICES OF KB.  IER IS
C RETURNED FROM TRANS, IPOS IS 0 IF ALL VALUES ARE POSITIVE,
C 1 OTHERWISE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100),SL(NL,2)
      IPOS=0
      IER=0
      NC=LD(1,K)
      JB=LD(2,3)
      NVB=LD(3,KB)
      IF (NVB .EQ. 0) RETURN
      IF (NVB .LT  JB ) GO TO 150
      WRITE(7,120)
  120 FORMAT(1X/' NOT ENOUGH BUFFER SPACE')
      IER=5
      RETURN
  150 CALL TRANS(LD,NL,D,N,K,DB,NB,KB,IER)
      J3=NVB+3
      DO 300 J=4,J3
      J2=J-3
      L=LD(J,KB)
      CALL INDX(LD,NL,3,L,J1,&160,&170)
  160 WRITE(7,165)
```

```
  165 FORMAT(1X/' VARIABLE NOT FOUND')
      IER=6
      RETURN
  170 C1=SL(J1+3,1)
      C2=SL(J1+3,2)
      DO 200 I=1,NC
      C=DB(I,J2)
      IF ( C.GT.C1 .AND. C.LE.C2 ) GO TO 190
      DB(I,JB)=0.D0
      GO TO 200
  190 IF (C .LE. 0.D0) IPOS=1
  200 CONTINUE
  300 CONTINUE
      RETURN
      END
C ************************************************************
      SUBROUTINE INVAR1(LD,NL,K,N,*,*)
C
C PURPOSE: TO VERIFY THAT THE NUMBER OF VARIABLES SPECIFIED IN
C COLUMN K OF LD DOES NOT EXCEED N.  THE FIRST RETURN IS USED IF
C FALSE; THE SECOND IF TRUE.
C
      DIMENSION LD(NL,12)
      NACT=LD(3,K)
      IF (NACT .LE. N) RETURN 2
      WRITE(7,110) NACT,N
  110 FORMAT(1X/' YOU HAVE SPECIFIED TOO MANY VARIABLES',/,
     *I3,' VARIABLES WERE SPECIFIED ',/,
     *' BUT THE LIMIT IS ',I3)
      RETURN 1
      END
C ************************************************************
      SUBROUTINE SAVE(LD,NL,D,N,K,Q,L,*,*)
C
C  PURPOSE: TO SAVE THE VARIABLE Q IN THE DATA BASE D.
C  LD, NL, D, N, K ARE DEFINED IN DBASE
C  ON OUTPUT, L IS THE LABEL OF THE SAVED VARIABLE.
C  RETURN 2 IS USED IF THE SAVE IS SUCCESSFUL; RETURN 1 OTHERWISE.
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,100),D(N,100),Q(N)
  100 WRITE (7,110)
  110 FORMAT(1X/' ENTER THE LABEL TO BE USED  (E.G. 09)',/,'&?')
      READ (5,115) L
  115 FORMAT(I2)
      WRITE(7,118) L
  118 FORMAT(1X/' THE DESIRED LABEL IS ',I2)
      CALL CHECK(&100,&120)
  120 CALL INDX(LD,NL,K,L,JD,&150,&130)
  130 WRITE(7,135) L
  135 FORMAT(1X/' VARIABLE ',I2,' IS ALREADY IN THE DATA BASE',/,
     *' DO YOU WANT TO CHANGE ITS VALUES?')
      CALL CHECK(&100,&140)
  140 WRITE(8,142) L
  142 FORMAT(I2,'  LABEL FOR SAVED VARIABLE',/,' O  OK',/,
     *' O  OK TO CHANGE VALUES')
      GO TO 500
  150 WRITE(8,152) L
  152 FORMAT(I2,'  LABEL FOR SAVED VARIABLE',/,' O  OK')
      NVD=LD(3,K)+1
```

```
      NVTD=LD(2,K)
      NVB=LD(3,3)+1
      NVTB=NL-3
      IF ( (NVD .LT. NVTD) .AND. (NVB .LT. NVTB) ) GO TO 400
      WRITE(7,160)
  160 FORMAT(1X/' NOT ENOUGH ROOM, YOU MIGHT WANT TO REUSE AN',
     *' EXISTING VARIABLE')
      RETURN 2
  400 CALL INDX(LD,NL,3,L,JB,&420,&410)
  410 WRITE(7,415) L
  415 FORMAT(1X/' YOU MUST USE A DIFFERENT LABEL')
      GO TO 100
  420 CONTINUE
  440 JD=NVD
      LD(3,K)=LD(3,K)+1
      LD(3,3)=LD(3,3)+1
      JB=LD(3,3)+3
      JD3=JD+3
      LD(JD3,K)=L
      LD(JB,3)=L
      LD(JB,4)=2
  500 WRITE(6,510) L
  510 FORMAT(' THIS VARIABLE HAS THE LABEL ',I2)
      NCB=LD(1,3)
      NCD=LD(1,K)
      IF ( NCB .LT. NCD ) GO TO 600
      DO 530 I=1,NCB
  530 D(I,JD)=Q(I)
      RETURN 2
  600 WRITE(7,610)
  610 FORMAT(1X/' ENTER A VALUE FOR THE OMITTED CASES (E.G. -999.)')
      READ(5,615) C
  615 FORMAT(1G20.12)
      WRITE(6,620) C
  620 FORMAT(' THIS VARIABLE HAS BEEN GIVEN THE VALUE',G20.12,/,
     *' FOR THE OMITTED CASES')
      WRITE(8,625) C
  625 FORMAT(1G20.12,'  MISSING VALUE')
      JDF=LD(2,K)
      IB=0
      DO 640 I=1,NCD
      IF ( D(I,JDF) .EQ. 0.D0 ) GO TO 630
      IB=IB+1
      D(I,JD)=Q(IB)
      GO TO 640
  630 D(I,JD)=C
  640 CONTINUE
      IF ( IB .EQ. NCB ) RETURN 2
      WRITE(7,650) IB,NCD
  650 FORMAT(1X/,I5,' CASES HAVE BEEN SAVED INSTEAD OF ',I5)
      RETURN 1
      END
C     ***************************************************************
      SUBROUTINE RESID(LD,NL,D,N,K,DB,NB,B,Q,IOPT,IER)
C
C     PURPOSE: TO CALCULATE THE PERDICTED VALUE (IOPT=2) OR
C     RESIDUAL (IOPT=1) DETERMINED BY THE PRIMARY EQUATIONS
C     OF THE MODEL USING THE COEFFICIENTS STORED IN B.  Q IS
C     THE OUTPUT VECTOR AND LD,NL, ... ,NB ARE THE SAME AS IN
C     THE ROUTINE SIGMA.
```

```
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100),B(NL),Q(N)
      IER=0
      NX=LD(3,5)
      IF (NX .GT. 0) GO TO 200
      WRITE(7,110)
  110 FORMAT(1X/' THERE ATE NO VARIABLES IN THE MODEL')
      IER=1
      RETURN
  200 CALL TRANS(LD,NL,D,N,K,DB,NB,5,IER)
      IF (IER .NE. 0) RETURN
      NC=LD(1,3)
      DO 300 I=1,NC
      Q(I)=0.D0
      DO 300 J=1,NX
  300 Q(I)=Q(I)+B(J)*DB(I,J)
      IF (IOPT .EQ. 2) RETURN
      CALL TRANS(LD,NL,D,N,K,DB,NB,7,IER)
      IF(IER .NE. 0) RETURN
      DO 400 I=1,NC
  400 Q(I)=DB(I,1)-Q(I)
      RETURN
      END
C  *************************************************
      SUBROUTINE WRIT1(LD,NL,D,N,K,DB,NB,FMT,TBS)
C
C  PURPOSE: TO WRITE OUT DATA
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION LD(NL,12),D(N,100),DB(NB,100),TBS(3,15),FMT(5)
      IF ( LD(1,K) .GT. 0 ) GO TO 100
      WRITE(7,50)
   50 FORMAT(1X/' THIS DATA BASE CONTAINS NO DATA',/,
     *' YOU MUST READ IN THE DATA FIRST')
      RETURN
  100 WRITE(7,101)
  101 FORMAT(1X/' ENTER VARIABLES TO BE WRITTEN OUT')
      CALL INVAR(LD,NL,3,11,TBS(1,15),2,IER)
      CALL INVAR1(LD,NL,11,LD(2,3),&100,&102)
  102 IF (IER .EQ. 0) GO TO 110
  103 WRITE(7,105)
  105 FORMAT(1X/' ONE OF THESE VARIABLES DOES NOT EXIST')
      GO TO 100
  110 CALL TRANS(LD,NL,D,N,K,DB,NB,11,IER)
      IF (IER.GT.0) GO TO 103
      NCB=LD(1,3)
      WRITE(7,115)
  115 FORMAT(1X/' NUMBER OF CASES TO BE WRITTEN OUT  E.G. 10.',
     */,'&?')
  118 READ(5,120) RNC
  120 FORMAT (G12.5)
      NC=RNC
      IF (NC .GT. 0) GO TO 125
      WRITE(8,124) RNC
  124 FORMAT(' 1',/,1G12 5,'  ILLEGAL NUMBER OF CASES')
      RETURN
  125 IF ( NC .LE. NCB ) GO TO 140
      IF ( NC .LE. N ) GO TO 135
      WRITE(7,130) NC,NCB
```

```
 130 FORMAT(1X/' UNABLE TO WRITE ',I10,' CASES',/,
    *' SINCE THE DATA BASE CONTAINS ONLY ',I4,' SELECTED CASES',/,
    *' ENTER THE NUMBER OF CASES TO BE WRITTEN OUT',/,
    *' (BE SURE TO INCLUDE A DECIMAL POINT)',/,'&?')
     GO TO 118
 135 WRITE(7,137) NC,NCB,NCB
 137 FORMAT(1X/' UNABLE TO WRITE ',I4,' CASES',/,
    *' SINCE THE DATA BASE CONTAINS ONLY ',I4,' SELECTED CASES',/,
    *' INSTEAD ALL ',I4,' CASES WILL BE WRITTEN')
     NC=NCB
 140 WRITE(7,145)
 145 FORMAT(1X/' DATA FORMAT FOR OUTPUT  E.G. (2F10.2)',/,'&?')
     READ(5,150) (FMT(I), I=1,5)
 150 FORMAT(5A8)
     WRITE(7,160)
 160 FORMAT(1X/' OUTPUT DEVICE E.G. 06',/,'&?')
     READ(5,170) IDEV
 170 FORMAT(I2)
     NV=LD(3,11)
     NV1=NV+3
     WRITE(7,180) (LD(J,11), J=4,NV1)
 180 FORMAT(1X/' WRITE OUT',/,' VARIABLES  ',20(1X,I2))
     WRITE (7,190) NC,NCB
 190 FORMAT('FOR THE FIRST ',I4,' OF THE ',I4,
    *' SELECTED CASES')
     WRITE (7,194) (FMT(I), I=1,5)
 194 FORMAT(' USING THE FORMAT ',5A8)
     WRITE(7,196) IDEV
 196 FORMAT(' ON THE DEVICE ',I2)
     CALL CHECK(&100,&200)
 200 WRITE(8,202) (LD(J,11),J=4,NV1)
 202 FORMAT(20(I2,1X))
     J3=NV-20*(NV/20)
     IF( J3.EQ.0) WRITE(8,203)
 203 FORMAT('00')
     WRITE(8,204) RNC
 204 FORMAT(1G12.5,'  NUMBER OF CASES')
     WRITE(8,206) (FMT(I), I=1,5)
 206 FORMAT(5A8,'  OUTPUT FORMAT')
     WRITE(8,208) IDEV
 208 FORMAT(I2,'  DEVICE',/,' 0  OK')
     DO 250 I=1,NC
 250 WRITE(IDEV,FMT,ERR=300) (DB(I,J), J=1,NV)
     WRITE(7,260) NC
 260 FORMAT(1X/,I5,' CASES HAVE BEEN WRITTEN')
     RETURN
 300 WRITE(7,310) I
 310 FORMAT(1X/' UNABLE TO WRITE CASE ',I4,
    *' USING THE SPECIFIED FORMAT')
     WRITE (7,320) (DB(I,J),J=1,NV)
 320 FORMAT(' THE VALUES FOR THIS CASE ARE:',/,10G20.12)
     GO TO 100
     END
C    *****************************************************
C
```