TWINSPAN : a FORTRAN program for arranging multivariate data in an ordered two-way table by classification of the individuals and attributes / M.O. Hill

Hill, M. O.

Ithaca, N.Y. : Section of Ecology and Systematics, Cornell University, [1979?]

https://hdl.handle.net/2027/coo.31924000189385

TWINSPAN

A FORTRAN Program for Arranging Multivariate Data in an Ordered Two-way Table by Classification of the Individuals and Attributes

M. O. Hill

Ecology and Systematics
Cornell University
Ithaca, New York 14850

July 1979

```
                    11  1
                    21980763 1245

 9 SOLI SEM         211-3-------    0000
 5 PHRA COM         53415-------    0001
 3 IVA  FRU         452113------    0010
11 SPAR PAT         3254-5------    0011
 8 SCIR OLN         -1---53-----    01
 2 DIST SPI         -5414545-424    100
 1 ATRI PAT         231--3221411    101
 6 SALI EUR         ----22-23411    1100
10 SPAR ALT         --142-135553    1101
 7 SALI VIR         ----------34    1110
12 SUAE MAR         ----------54    1110
 4 JUNC GER         -----151----    1111

                    000000111111
                    000001001111
                    00011 010011
                    00101   0101
                    01
```

TWINSPAN -- A FORTRAN program for arranging multivariate data in an ordered two-way table by classification of the individuals and attributes

M. O. Hill

Section of Ecology and Systematics

Cornell University

Ithaca, New York 14850

July 1979

PREFACE

TWINSPAN is program CEP-41 in the Cornell Ecology Programs series, written by Mark O. Hill. TWINSPAN is a FORTRAN program for two-way indicator species analysis, which is an improvement upon the original indicator species analysis (Journal of Ecology 63:597-613) in that species are classified as well as samples. This two-way classification makes possible a tabular matrix arrangement which approximates the result of Braun-Blanquet tablework. TWINSPAN also constructs a key to the sample classification by identifying one to several species which are particularly diagnostic of each division in the classification.

TWINSPAN is a polythetic divisive method of classification; details of the algorithm are presented in this documentation. Although developed and discussed in a phytosociological context where the data matrix is one of species abundances in samples, the method is applicable to a wide range of data matrices expressing attributes of individuals (for example, vocabulary of persons, market preferences of customers, and types of artifact in archeological sites). TWINSPAN contains about 2300 cards.

The computer time required by TWINSPAN has a linear dependence on the amount of data, so large problems can be tackled readily. This is in marked contrast to most classification methods with requirements rising with the second or higher power of the number of samples. Furthermore, TWINSPAN does not store zero values of the data matrix (which usually predominate in large data matrices), and it does not produce or store a secondary matrix of sample similarities. Hence TWINSPAN is also very efficient with computer memory.

Copies of this documentation and magnetic tape copies of the program are available at cost. A catalog to the Cornell Ecology Programs series is available upon request. TWINSPAN is in the public domain and may be copied. TWINSPAN is offered with no warranty, express or implied, and with no legal liability or responsibility.

In many cases it is useful to parallel TWINSPAN classification with an ordination. In some cases if the data set is very large preliminary

clustering of essentially replicate samples may reduce the data set to manageable proportions, facilitating study of subsequent computer outputs. Other programs in the Cornell Ecology Programs series may be useful for these functions.

TWINSPAN requires that the input data matrix be supplied in a particular format which simplifies program code and permits fast data input. This format is essentially the condensed format in sample order of ORDIFLEX (an earlier program in the Cornell Ecology Programs series). However, in some cases this format may not be convenient for the initial keypunching of a data set. In order to allow greater flexibility in data matrix format, TWINSPAN is supplied with a utility program, CONDENSE, which reads data matrices in any format accepted by ORDIFLEX, and copies them into the restricted format used by TWINSPAN. CONDENSE is supplied without additional charge. This same format is also used by DECORANA, a Cornell Ecology Program by Mark O. Hill for detrended correspondence analysis and reciprocal averaging.

Hugh G. Gauch, Jr.

Section of Ecology and Systematics
Cornell University
Ithaca, New York 14850

July 1979

RRATA.

Bugs in TWINSPAN and DECORANA

To date, no bugs of note have appeared in DECORANA. Several bugs have appeared in TWINSPAN, most of which will only appear when it is attempted to run large problems, where there are problems of dimensioning. Those that are known to me at present are as follows.

✓ Lines 479,480 should read MM,MMAX, not M,MMAX. At present, if the allowed number of samples is exceeded, a bad error message is obtained. Changing MM to M puts the error message right.

✓ Line 987 should read
IF(IP.LE.NMAX) INDPOT(IP)=JJJ
The point here is that it should not be possible to write outside the bounds of the array INDPOT. An error message will appear shortly, with an indication of the required dimensioning (cf. lines 1022-1026).

✓ Line 552 should read
CALL PARA(MMIN,2,10000,5)
The poiht here is that groups smaller than 2 should not be attempted to be divided. The program gets into difficulties if it tries to do this.

✓ Line 1246 should read
1004 FORMAT((1X,7(A4,1X,A4,I1,'(',A1,')',3X)))
i.e. the FORMAT statement should have an extra pair of bbackets round the outside of it. If this is not put in, and if more than 7 indicators are asked for, the first letter of the 8th indicator is regarded as a control character and omitted from the print-out.

Note: These four changes are already made on magnetic tape copies of TWINSPAN made since June 1981 -HP

✓ Lines 1076-1083 Common blocks are in wrong order; this leads to diagnostic messages being put out by some compilers, but has never to my knowledge been regarded as a fatal error.

✓ Line 388 Numerous users have had trouble because the number read by the program as NITEM (=8 in diagram on p. 18) was wrongly positioned in the data. The awkward position of this number is a relic from days when the data were routinely submitted on cards. Users may think it worth their while to test that the value of NITEM lies between suitable limits (e.g. 4 to 20); this would eliminate almost all cases of wrongly submitted data. If users do want to test the value of NITEM, then they should insert a test statement immediately following line 388.

✓ Fig. 1 (on page 2). The lines indicating the dichotomy are omitted. Do not be confused by this.

Restrictions not indicated in existing write-up are as follows

NMAX must exceed (2**(LEVMAX+1)-1)*3) (=381 in default case). This will not cause trouble unless it is desired to use a large number for LEVMAX, i.e. maximum level of divisions - cf. p. 13.

Bugs in TWINSPAN, continued

Dimensions of TOT, TOTJ must exceed
(2**(LEVMAX+1)-1) (=127 in default case).
The dimensions supplied on line 35, of 511 (=2**9-1)
allow 8 levels of division.

In addition to the restrictions given on p.33, an additional restriction is that the data when held in non-pseudospecies form must not occupy more than half the data matrix. If they do occupy more than this, the species classification may not be possible. This can be tested for following line 103 of the program, by demanding that
2*MDAT.LT.NDAT

If (as is usually the case) it is not wanted to run the program unless the species classification is possible, then it may be sensible to put in an instruction to terminate the program following lines 116, 1016 if the condition is not satisfied, and following line 103 (addition).

GENERAL COMMENT

None of the above matters will cause any trouble with a standard analysis of a normal-sized problem. The bugs are pretty unimportant in general. On the other hand, the restrictions can be serious when trying to analyse big problems. Provided that the level of the divisions is not altered, the main difficulty is in ensuring that there is enough room in the main data matrix. This amounts to ensuring that

(i) the data when in pseudospecies form do not have length greater than $\frac{1}{2}$(NDAT)

(ii) the data when in non-pseudospecies form do not have length greater than $\frac{1}{2}$(NDAT).

Provided that these two conditions can be assured, there will probably not be too much difficulty in running large problems.

It is of course unfortunate that the matter of dimensioning is not made clearer by the program. This is one aspect of the fact that the program would be better if it could be simplified. However, I do not see myself having time to make a simplification for some years, and users must content themselves with its present form for the time being.

M.O. Hill
ITE, Penrhos Road,
Bangor, Gwynedd LL57 2LQ
Wales, U.K.

26 March 1980

7 April 1980

COLLEGE OF ARTS & SCIENCES
NEW YORK STATE COLLEGE OF AGRICULTURE & LIFE SCIENCES
DIVISION OF BIOLOGICAL SCIENCES
CORNELL UNIVERSITY
SECTION OF ECOLOGY & SYSTEMATICS

- [X] For your information
- [ ] For appropriate action
- [ ] Please comment and return
- [ ] For your files or disposition
- [ ] Please answer direct
- [ ] Need not return

To: TWINSPAN Users:

Mark Hill has supplied this list of bugs and restrictions for TWINSPAN, and we are circulating it to TWINSPAN users. Should problems or questions arise, it would ordinarily be best to write to Dr. Hill directly.

From Hugh G. Gauch, Jr.
Hugh G. Gauch, Jr.

CONTENTS

1. INTRODUCTION

1.1 TWINSPAN and indicator species analysis

TWINSPAN (TWo-way INdicator SPecies ANalysis) is a computer program in FORTRAN designed primarily for ecologists and phytosociologists who have collected data on the occurrence of a set of species in a set of samples. The samples may be stands, relevés, stomach contents, island faunas, or whatever is appropriate to the study. TWINSPAN is a development of a method already published under the name "indicator species analysis" (Hill, Bunce & Shaw, 1975). However, it is much more flexible than previous programs for indicator species analysis, and involves many new features.

The most significant new feature is that the program first constructs a classification of the samples, and then uses this classification to obtain a classification of the species according to their ecological preferences. The two classifications are then used together to obtain an ordered two-way table that expresses the species' synecological relations as succinctly as possible.

1.2 Ordered two-way tables

Consider the table in Fig. 1. This is an ordered two-way table, and is the sort of structure that TWINSPAN aims to exhibit. (In fact it was obtained by TWINSPAN, but with non-standard parameter settings.)

The table has been ordered to exhibit the relation between the species and the samples as clearly as possible. The top five species are more abundant on the left side of the primary division than on the right side. The bottom five species are more abundant on the right side of the primary division than on the left side. The middle two species are somewhat indifferent, occurring widely on both sides.

For the most part, the species form a fairly well defined sequence, from those that prefer the left side of the primary dichotomy to those that prefer the right side of the dichotomy. However, one species "4 JUNC GER" appears out of sequence, at the end of the list. This is

SEE ERRATA

```
                    11  1
                    21980763 1245

 9 SOLI SEM         211-3-------   0000
 5 PHRA COM         53415-------   0001
 3 IVA  FRU         452113------   0010
11 SPAR PAT         3254-5------   0011
 8 SCIR OLN         -1---53-----   01
 2 DIST SPI         -5414545-424   100
 1 ATRI PAT         231--3221411   101
 6 SALI EUR         ----22-23411   1100
10 SPAR ALT         --142-135553   1101
 7 SALI VIR         ----------34   1110
12 SUAE MAR         ----------54   1110
 4 JUNC GER         -----151----   1111

                    000000111111
                    000001001111
                    00011 010011
                    00101   0101
                    01
```

Fig. 1 Two-way table ordered and classified by TWINSPAN (data of L. F. M. Fresco discussed in Gauch, 1977, pp. 70-74). Species names are shown at the left; sample numbers along the top. The classifications of species and samples are indicated along the right and bottom margins (notation is explained in Section 3.3 and need not concern us here). The main dichotomy for the species is indicated by a horizontal line, and the main dichotomy for the samples is indicated by a vertical line. Values indicate a scale of abundance, with absence of a species represented by the symbol "-".

not a mistake of the program, but is a result of the fact that this species behaves differently from the others, occurring with high abundance in a sample (number 6) where the commoner species are rather poorly represented.

The ordered table, therefore, is not simply an ordination. If it were, then the odd species "4 JUNC GER" would be put in an intermediate position. Rather, the aim of the arrangement is to throw the salient features of the data into sharp relief, by grouping like species with like, and like samples with like. In the example, the odd species "4 JUNC GER" is unlike any other (though it obviously has some relation to "8 SCIR OLN") and is therefore placed in an anomalous position.

1.3 Differential species

A differential species is one with clear ecological preferences, so that its presence can be used to identify particular environmental conditions. In Fig. 1 the species "3 IVA FRU" is a fine differential species. Its presence exactly demarcates the group of samples to the left of the primary division, and is strongly indicative of conditions in which the other three of the top four species are likely to be found. On the other hand the species "2 DIST SPI" is a poor differential species, not showing any marked affinity for a particular group of samples.

1.4 Strategy of analysis in TWINSPAN

TWINSPAN is designed to construct ordered two-way tables, and the method of doing so is by identification of differential species. In this respect it closely resembles the "hand" method of classification outlined by Mueller-Dombois & Ellenberg (1974, Chapter 9). It differs, however, in its treatment of the species. In the method outlined by Mueller-Dombois & Ellenberg, the species are classified at the same time as the samples. In TWINSPAN, on the other hand, the samples are classified first, and the species are classified second, using the classification of the samples as a basis.

The basic structure of TWINSPAN is as follows.

1. Classify the samples in a divisive hierarchy, dividing them first into 2 subsets, then 4, 8, 16, etc.
2. Convert the sample classification into an ordering.
3. Using the groups of samples as a basis, construct attributes for the species. For example, in Fig. 1, species 9, 5, 3, 11, 8 would be described as "preferential to the left side of the major division," and would be registered as possessing the attribute "preferential to the left side of the major division."
4. Classify the species in the same way as the samples, but with the difference that whereas the species were treated as attributes of the samples, the species have attributes of the kind indicated above.
5. Convert the species classification also into an ordering.
6. Print out the resulting ordered two-way table.

1.5 Making a dichotomy

The basic activity in TWINSPAN is to make a dichotomy. Indeed, as indicated in Section 1.4, all that the program does, in effect, is to divide up the samples into groups by repeated dichotomization, and then to do the same for the species.

It can be argued that this is a bad way to organize a table because dichotomies do not arise naturally. Indeed, Mueller-Dombois & Ellenberg (1974) recommend dividing the species into three categories, those that are preferential to one side of the division, those that are preferential to the other, and those that are indifferent. It was noted above (Section 1.2) that the species in Fig. 1 can be put into three such categories, comprising the first five species (preferential to the left), the next two (indifferent), and the last five (preferential to the right). But more careful inspection will show that these three categories are nearly as arbitrary as the two indicated in the figure. Indifference and preferentiality are a matter of degree; a sharp natural distinction between the two is often not found. Indeed, the dichotomy of the species in Fig. 1 shows that the species that are preferential to the left form a more clearly defined category than the species that are preferential to

the right--a fact that would be missed if the species were initially divided into three categories of equal status.

Given that a table can be structured satisfactorily by successive dichotomization, how are dichotomies to be made? The method suggested by Mueller-Dombois & Ellenberg is very roughly as follows.

1. Identify a direction of variation in the data--i.e. find a group of differential species that tend to occur together in the samples studied.
2. Construct an ordination by assigning each of these species a score of +1 and adding together the scores of the species in the sample.
3. Divide the ordination at an appropriate point into distinguishable communities.
4. Divide the species into categories according as to whether they are preferential or not to these communities.

This description of the method of "hand" classification is deliberately oversimplified, but does nevertheless encapsulate some of its essential features.

Now consider how this method might be applied to the table in Fig. 1. Suppose, for example, that it has been observed that species 5, 9, and 11 tend to occur together. Then these species can be declared as differential species and the samples can be ordinated thus:

Samples 9, 11, 12 - score +3
Samples 8, 10 - score +2
Sample 7 - score +1
Remainder - score 0.

If the samples are now arranged in order of descending score, it will rapidly become apparent that species 3 is also a differential species of the same group. If this species is now also given a score of +1, then the scores would work out like this:

Samples 9, 11, 12 - score +4
Samples 8, 10 - score +3
Sample 7 - score +2
Remainder - score 0.

Now there is a discontinuity, suggesting that the samples that score 0 should be distinguished from the rest.

TWINSPAN makes its dichotomies in a broadly similar manner to that described above. There are many complications, but the basic idea is the same. The stages of making a dichotomy in TWINSPAN are as follows.

1. Identify a direction of variation in the data by ordinating the samples. This ordination is referred to below as the "primary" ordination, and is made by the method of reciprocal averaging (Hill, 1973).
2. Divide the ordination at its middle to get a crude dichotomy of the samples.
3. Identify differential species that are preferential to one side or the other of the crude dichotomy.
4. Construct an improved ordination (referred to below as the "refined" ordination), using the differential species as a basis.
5. Divide the refined ordination at an appropriate point to derive the desired dichotomy.

If this were all, TWINSPAN would be relatively easy to describe. However, yet a third ordination is also constructed, the "indicator" ordination. This is based on a small number of the most strongly differential species, and is designed to provide a simple criterion for re-identification of the groups. So a further stage must be added.

6. Construct a simplified ordination, the "indicator" ordination, based on a few of the most highly preferential species. See whether the dichotomy suggested by the refined ordination can be reproduced by a division of the indicator ordination.

To summarize, TWINSPAN makes its dichotomies by dividing ordinations in half. There are three ordinations involved:

1. The primary ordination (reciprocal averaging), which is divided to obtain an initial, crude dichotomy;
2. The refined ordination, which is derived from the primary ordination through the identification of differential species; and
3. The indicator ordination.

With the exception of borderline cases (see Section 3.3), the refined

ordination is the one that is used to determine the dichotomy. The indicator ordination is essentially an appendage, put there for the convenience of users who want a succinct characterization of the dichotomy.

1.6 Warning about nomenclature

Unfortunately, the name "indicator species analysis" can cause much confusion. The method of division in TWINSPAN closely resembles the method published as "indicator species analysis" by Hill, Bunce & Shaw (1975). But, as already stressed, the indicator ordination is only an appendage, not the real basis of the method. "Differential species analysis" would possibly be a better name, but even this is not strictly accurate. Perhaps the best name would be "dichotomized ordination analysis," which would be a generic term to describe a wide variety of similar types of method.

Therefore, dear reader, please do not be confused by names. The basic method in TWINSPAN is the division of ordinations, and not the selection of indicator species.

1.7 Pseudospecies

The idea of a differential species is essentially qualitative, and to be effective with quantitative data must be replaced by a quantitative equivalent. This equivalent is the "pseudospecies" (Hill, Bunce & Shaw, 1975; Hill, 1977). The essential idea is that much of the quantitative information can be retained by expressing it on a relatively crude scale such as the Braun-Blanquet scale of cover-abundance (Mueller-Dombois & Ellenberg, 1974, p. 82; Westhoff & Maarel, 1973, p. 639). The levels of abundance that are used in TWINSPAN to define the crude scale are here termed "pseudospecies cut levels."

Consider, for the sake of example, the Braun-Blanquet scale. Ignoring the distinction between 1, +, and r, the scale is as follows:

1	0 - 4%	4	51 - 75%
2	5 - 25%	5	76 - 100% .
3	26 - 50%		

If quantitative data with cover expressed on a percentage scale are entered into TWINSPAN, then the Braun-Blanquet scale can be used by entering the pseudospecies cut levels:

0 5 26 51 76 .

Consider now a species, e.g. Poa annua, whose cover in one sample is 18% and in another is 36%. Although these values differ by a factor of 2, the samples have an important feature in common, namely that they share a moderate abundance of Poa annua. Given the Braun-Blanquet scale defined above, this fact can be expressed in terms of pseudospecies by saying that the samples contain the following pseudospecies:

Sample with Poa annua 18% - P. annua 1, P. annua 2;
Sample with Poa annua 36% - P. annua 1, P. annua 2, P. annua 3 .

Hence, using the Braun-Blanquet scale, the samples are registered as having two pseudospecies in common, and one different. In spite of the rather large difference in cover, the samples are registered as having more in common than by way of difference.

The method of pseudospecies allows quantitative values to be used as differential "species" and as indicators. Thus, in TWINSPAN, instead of a differential species P. annua, it is possible to have a differential species P. annua 2, which--with the pseudospecies cut levels defined above--occurs when and only when P. annua has cover 5% or greater.

In practice, users of TWINSPAN who are mainly interested in a good tabular arrangement will hardly concern themselves with the technical detail of psuedospecies. For them it suffices to know that they are using a particular scale of abundance. However, there is one important practical point, namely that information on the occurrence of each pseudospecies is stored separately in the computer. Use of very numerous pseudospecies may therefore be undesirable with larger problems, as they take up space in the computer and use up additional computer time. The amount of space that they take up is printed in the output (see Section 3.2).

2. DATA INPUT AND OUTPUT

2.1 Input and output devices

Input for TWINSPAN is composed of two parts:

1. The data matrix itself, entered from input device 5; and
2. Parameters determining details of the analysis and output, entered from device 3.

If the operating system is interactive, then device 3 would ordinarily be a computer terminal.

Output from TWINSPAN is composed of two, or optionally three, parts:

1. Output to the lineprinter using the FORTRAN word PRINT;
2. Monitoring to output device 4, using the FORTRAN phrase WRITE(4,...); and
3. Optional output to device 7, using the FORTRAN phrase WRITE(7,...).

Output to devices 4 and 7 is intended to be construed as 1 line-control character and 79 other characters. No confusion will arise, however, if the control character is output explicitly. Indeed, in the case of output to device 7, the control character is always blank, so that the output will not even be unsightly.

Output to the printer uses 1 line-control character and a maximum of 124 other characters. Users with narrower lineprinters will need to modify the following FORMAT statements:

In subroutine QUIKIN: 1102, 1103, 1105, 1107;
In subroutine CLASS: 1010, 1004;
In subroutine REPORT: 1050;
In subroutine XLIST: 1000, 1001;
In subroutine YLIST: 1000;
In subroutine TABOUT: 1000, 2000.

Most of the changes are easy enough to make. In XLIST and YLIST it is necessary to change the values of NITEM (numbers of items printed out per line) as well as the FORMAT statements. In TABOUT, small economies of field-width can be achieved by omitting either the species names or the species numbers. Larger economies can be achieved by reducing the number of samples that can be included on a page of the table. At

present this number is 100, but it can be reduced by changing the value of the parameter NUMTAB (line 53 of source code).

The monitoring to device 4 is intended for users with an interactive operating system, and would ordinarily be sent to the console of a computer terminal. It is quite dispensable, and its main use is to allow checking of input parameters as they are entered. Furthermore, it is reproduced almost exactly in the output to the lineprinter, so that once the program has been run to completion, the output to device 4 serves no further purpose.

The output to device 7 is a copy of the sample and species classifications in machine-readable form. It is intended for users who wish to subject the data to further automatic data-processing (see Section 3.7 below). It is optional, and is not supplied unless requested by the user.

2.2 Entering the input parameters (standard analysis)

To get the standard analysis, 11 cards (or card images), each with a -1 on it, should be entered on device 3. All input parameters are read in free format. If the operating system is interactive, then device 3 would ordinarily be a computer terminal, and the user is then prompted by monitoring on device 4 (which would be the console of the terminal).

The standard analysis should suffice in many applications, but in others the user will want to vary the parameters. In particular, the pseudospecies cut levels provided by the standard analysis assume that either the data are quantitative with quantities given as percentages (percent cover, percent frequency, percent biomass, etc.), or the data are presence-and-absence data, in which case the quantities must all be 1. (It is permissible for the quantities all to be 0, but then there is a distinction between a species present with quantity 0--i.e. present but with unspecified quantity arbitrarily represented as 0--and species that are genuinely absent.)

To vary the parameters, values other than -1 must be supplied.

2.3 Omitting samples

The first action of the program is to read in the data matrix from device 5 (see Section 2.14 for the correct format). Then a message appears on the console (i.e. is written to device 4):

```
DO YOU WISH TO OMIT SOME SAMPLES?
ENTER NUMBERS (NOT NAMES) OF ITEMS TO BE OMITTED
ONE PER CARD, ENDING LIST WITH A -1.
OTHER NEGATIVE NUMBERS DENOTE SEQUENCES.  FOR EXAMPLE
A 4 FOLLOWED BY A -8 OMITS ITEMS 4 THROUGH 8.
```

The standard response of -1 omits no samples. Typing

```
 5
-29
 35
-1
```

omits samples 5-29, and 35. Omission of items is the one case where the monitoring on the terminal is not reproduced exactly in the output to the lineprinter. For ease of comprehension, the input considered above would be written

```
OMIT ITEMS     5    -29
OMIT ITEM     35
```

in the output to the lineprinter.

2.4 Pseudospecies cut levels

Message on console:

```
ENTER NUMBER (NOT EXCEEDING 9) OF PSEUDOSPECIES CUT LEVELS
OR TYPE -1 FOR DEFAULT CUT LEVELS, WHICH ARE  0  2  5  10  20.
```

If a value other than -1 is entered, then appropriate cut levels must be supplied. They must be supplied on the next card, and not on the same card as the number of cut levels. Correctly entered parameters would be (for example)

```
4
0.0   0.5   1   5
```

The cut levels need not all be supplied on the same card; equally correct would be

```
4
0.0   0.5
1     5
```

A point to note about free-format input is that the program must know in advance how many values to expect. Hence the necessity of typing the number of cut levels first, on a separate card.

Levels should be chosen so as to reflect typical values of abundance, e.g. present, a little, a lot, more-or-less dominant. It is important, however, not to over-weight the effect of dominance by including many relatively high cut levels. With percentage data the default cut levels have proved very effective, and the user is recommended to adhere to them until he has some experience with the program.

2.5 Minimum group size for division

Message on console:

```
ENTER MINIMUM GROUP SIZE FOR DIVISION
TYPE -1 FOR DEFAULT VALUE (= 5); OTHERWISE TYPE VALUE REQUIRED,
WHICH MUST NOT LIE OUTSIDE LIMITS   0   10000
```

Groups smaller than this value are not divided further. Two points should be noted.

1. Groups much smaller than the specified size will often be formed, e.g. when the natural structure of the data is one large group and a few outliers.
2. The same minimum size for division is applied to the species classification as to the sample classification.

In general it is better to control the maximum level of divisions (Section 2.8) than to control the size of groups generated; though there certainly does come a size below which it is not worth dividing groups further.

2.6 Maximum number of indicators

Message on console:

```
ENTER MAXIMUM NUMBER OF INDICATORS PER DIVISION
TYPE -1 FOR DEFAULT VALUE (= 7); OTHERWISE TYPE VALUE REQUIRED,
WHICH MUST NOT LIE OUTSIDE LIMITS   0   15
```

This determines the maximum number of species that can be used in the indicator ordination (see Section 3.3 for a further explanation of what this involves). If no indicator ordination is desired, then this parameter can be set to zero. If a particular dichotomy can be specified precisely

by an indicator ordination based on a smaller number of indicators than the maximum, then the smaller number will be used.

2.7 Maximum number of species in final tabulations

Message on console:

```
ENTER MAXIMUM NUMBER OF SPECIES IN FINAL TABULATION
TYPE -1 FOR DEFAULT VALUE (= 100); OTHERWISE TYPE VALUE REQUIRED,
WHICH MUST NOT LIE OUTSIDE LIMITS   0   1000
```

In the two-way table at the end of the output, it is often inconvenient to clutter the diagram with rarer species. Therefore only the N commonest species are printed, where N is the value supplied. The default value is 100, which occupies about two pages of standard lineprinter output. If the value zero is supplied here, the final tabulation is omitted.

2.8 Maximum level of divisions

Message on console:

```
ENTER MAXIMUM LEVEL OF DIVISIONS
TYPE -1 FOR DEFAULT VALUE (= 6); OTHERWISE TYPE VALUE REQUIRED,
WHICH MUST NOT LIE OUTSIDE LIMITS   0   15
```

Even with large data sets there is little purpose in continuing to subdivide groups beyond a certain limit. Six levels of division are apt to produce about 64 groups. If there are larger numbers of groups, then interpretation can be difficult. Users will soon get a feel for how much division they require with any problem. Rarely will they want fewer than 4 levels, and almost never more than 7.

2.9 Diagrams of divisions

Message on console:

```
TYPE 1 IF DIAGRAMS OF DIVISIONS ARE WANTED.
TYPE -1 FOR DEFAULT VALUE (= 0); OTHERWISE TYPE VALUE REQUIRED,
WHICH MUST NOT LIE OUTSIDE LIMITS   0   1
```

In some applications it may be of interest to see how the indicator ordination relates to the refined ordination, and in particular to see whether the misclassifications are approximately borderline cases. If the value 1 is supplied, a scatter diagram relating the two ordinations

is printed out for each dichotomy (see Section 3.3 for what the diagrams mean).

2.10 Machine-readable copy of solution

Message on console:

```
TYPE 1 IF MACHINE-READABLE COPY OF SOLUTION TO BE WRITTEN TO DEVICE 7.
TYPE -1 FOR DEFAULT VALUE (= 0); OTHERWISE TYPE VALUE REQUIRED,
WHICH MUST NOT LIE OUTSIDE LIMITS   0   1
```

This refers to the optional output to device 7 mentioned in Section 2.1. Both the sample and species classifications are printed, together with the classes that items belong to in both binary and decimal notation. (See Section 3.3 for an explanation of the class numbering.)

2.11 Weights for levels of pseudospecies

Message on console:

```
ENTER WEIGHTS FOR LEVELS OF PSEUDOSPECIES.
FOR EXAMPLE WEIGHTS  1 2 2 2  SIGNIFY THAT PSEUDOSPECIES
CORRESPONDING TO 3 HIGHER CUT LEVELS ARE TO BE GIVEN TWICE
THE WEIGHT OF PSEUDOSPECIES AT THE LOWEST LEVEL.
TYPE -1 FOR DEFAULT VALUES (I.E. IF ALL VALUES ARE TO BE SET TO 1)
OR TYPE 0 IF NON-DEFAULT VALUES ARE TO BE ENTERED
```

If a 0 is typed at this stage, a further message appears:

```
NOW  n  INTEGER VALUES MUST BE SUPPLIED
WHICH MUST NOT LIE OUTSIDE LIMITS   0   1000000
```

The message is largely self-explanatory. The idea is that in some applications it may (for example) be desirable to pay particularly strong attention to species with high abundance. If so, then weights 1 1 1 1 3 could be supplied. Note that the number of values supplied should coincide with the number of cut levels, i.e. if there are 5 cut levels (as in the default case), then 5 weights should be supplied.

In many applications it is not necessary to use weights. However, with vegetation data taken from very large stands, it may sometimes be advisable to give presences and absences relatively low weight, as they may be due to anomalous small patches of terrain that occupy little of the stand area.

2.12 Indicator potential for cut levels

Message on console:

```
ENTER INDICATOR POTENTIALS FOR CUT LEVELS.
FOR EXAMPLE INDICATOR POTENTIALS  1 0 0 1 0  SIGNIFY
THAT PSEUDOSPECIES AT LEVELS 1 AND 4 CAN BE USED AS
INDICATORS, BUT THAT THOSE AT OTHER LEVELS CANNOT.
IN THE DEFAULT CASE, ALL PSEUDOSPECIES ARE AVAILABLE
AS INDICATORS.
TYPE -1 FOR DEFAULT VALUES (I.E. IF ALL VALUES ARE TO BE SET TO 1)
OR TYPE 0 IF NON-DEFAULT VALUES ARE TO BE ENTERED
```

If a 0 is typed at this stage, a further message appears:

```
NOW  n  INTEGER VALUES MUST BE SUPPLIED
NONE OF WHICH MUST LIE OUTSIDE LIMITS   0   1
```

The most common variation required here is that all the indicators should be real species and not pseudospecies. In this case, values 1 0 0 0 0 should be entered.

2.13 Omission of species from list of potential indicators

Message on console:

```
DO YOU WISH TO OMIT SOME SPECIES FROM LIST OF
POTENTIAL INDICATORS?  SPECIES OMITTED FROM THIS LIST
ARE USED IN THE CALCULATION, BUT CANNOT APPEAR AS INDICATORS.
ENTER NUMBERS (NOT NAMES) OF ITEMS TO BE OMITTED
ONE PER CARD, ENDING LIST WITH A -1.
OTHER NEGATIVE NUMBERS DENOTE SEQUENCES.  FOR EXAMPLE
A 4 FOLLOWED BY A -8 OMITS ITEMS 4 THROUGH -8.
```

This facility is to prevent particular species from being used in the indicator ordination. For example, if the indicator criterion is to be applied by relatively inexperienced field workers, it may be more useful if difficult taxonomic groups are omitted. Sequences of species are handled in the same way as sequences of stands (compare Section 2.3). Note that when species are barred from being indicators, they are nevertheless carried in the analysis, and will be printed out as usual in the final tabulation.

2.14 Format of the data matrix

The format used for input to TWINSPAN is the same as that for DECORANA (Hill, 1979), and is essentially the same as the "restricted condensed

format, input by samples" of ORDIFLEX (Gauch, 1977). Any matrix already in this format can be used directly by TWINSPAN, and any matrix in another format compatible with ORDIFLEX can be converted to the required format by the Cornell Ecology Program CONDENSE (Singer & Gauch, 1979).

Card 1 is a title card. This is reproduced at the front of the printed output and also at the front of the output to device 7.

Card 2 specifies two items:

1. The FORTRAN format in which the data have been written, using columns 1-60; and
2. The maximum number of couplets per card, this number being right-justified to column 70.

Users who are unfamiliar with FORTRAN format statements should consult a programmer who has used the language, or failing this any FORTRAN primer, or the documentation for ORDIFLEX (Gauch, 1977).

Card 3 is the beginning of the data matrix. The first item on each card is the index number of the sample. These index numbers must be in ascending order, except that an index number of zero terminates the matrix. Each species is identified by a number, and this is followed by a quantity. For example, the input

```
FUNNY LITTLE EXAMPLE
(I2,2(I2,F4.1))
 1 1 1.3 3 1.0
 1 4 1.0
 2 1  .1 2  .2
 0
SPEC 1  SPEC 2  SPEC 3  SPEC 4
SAMP 1  SAMP 2
```

2 in column 70 of card 2 ——————————> 2

signifies that sample 1 contains species 1 with abundance 1.3, species 3 with abundance 1.0, and species 4 with abundance 1.0. Sample 2 contains species 1 with abundance 0.1 and species 2 with abundance 0.2. Written out as a matrix, the above data are as follows:

Species	1	2	3	4
Sample				
1	1.3	0	1.0	1.0
2	0.1	0.2	0	0

Species names must be supplied next. These are abbreviated to binomials, each consisting of a 4-letter generic name and a 4-letter specific name, and are punched without spaces, so that ten 8-letter species names occupy an 80-column card.

Sample names are the last item to be supplied. Each of these is 8 letters long, so 10 sample names occupy an 80-column card.

Species names and sample names have been included in their correct place in the example above. A more substantial example is given in Fig. 2.

With the default analysis the lowest cut level is zero. In this case species entered with quantity zero are interpreted as present, but in small quantity, which is arbitrarily represented by a zero. A clear distinction needs to be maintained, therefore, between species that are absent--i.e. which are not listed among the species present in the sample--and species that are present but in small quantity that is represented by a zero.

2.15 Restrictions on reading the data matrix

As supplied, the program is dimensioned for up to 500 samples, 800 species, and 14500 non-zero items in the data matrix. These values can be increased by altering the declaration statements at the front of the main program (see Section 4.1).

The sample numbers must be ascending but need not be sequential. Samples can therefore be edited out of the matrix without making other changes. (But beware that the number of species names supplied still coincides with the highest species number in the data matrix.) Note well: this freedom may not apply to other Cornell Ecology Programs; for many of them the sample numbers must be sequential.

Species with no occurrences are permissible, and cause no difficulty. They do, however, use up array space and slightly slow down the calculation.

```
   12   24TENNESSEE WATERSHED (G & G 1972 ORNL-IBP-71-10).            TCTENA** S
(I2,2X,8(I3,F6.2))                                                          8
 1     1 60.32  8  1.65  9  1.75 12 30.13
 2     1 72.95  8  1.58  9  0.92 12  0.79
 3     1 81.21 10  0.75 12  6.94
 4     1 90.23  2  0.87  8  1.70 10  3.80 12  7.07
 5     1 57.50  2  1.36  7  3.98  8  4.79  9  4.12 10  1.96 12  7.07
 6     1 74.63  2  0.20  4  0.87  5  8.69  6  1.01  8  6.15 10  8.95 11  0.35
 7     2 14.48  3  7.00  4  3.36  5  7.21  6  3.28  8 13.75 10 15.07 12 33.05
 8     2  5.51  6 11.80  9  0.96 11  5.32 12 37.05
 9     1  0.83  2 12.95  4  5.05  6  7.71  8  1.40 10  1.41 12 64.03
10     2 17.31  3  3.98  5  4.19  6 14.29  9  2.21 11  1.79 12 41.83
11     2  4.06  8  0.35 10 16.11 11  2.52 12 36.44
12     2  4.26  4  5.14  5  8.25  6  0.75  8  4.94 10  1.01 12 19.09
13     2 17.52  5 12.56  8  5.59  9  1.84 10  9.44 11  2.96 12  5.60
14     2 19.43  3  2.82  4  0.83  5 14.91  6 16.45  9  0.55 10 16.40 11  9.19
15     2 16.41  5 32.02  6 52.83  8  4.53 10  5.86 11  4.98
16     2 34.57  3  3.65  4  1.10  5 15.23  6  9.69  8  4.51  9  8.08 10 11.24
16    11  4.47 12  3.49
17     2 50.76  3 18.27  4  2.18  5  6.13  6 10.22  8  4.24  9  2.21 10  2.78
17    11  9.00
18     2 11.75  3  3.89  5 13.20  6 22.79  8 10.27  9  1.81 10 10.79 11 19.86
18    12  9.32
19     1  8.44  2  5.50  5 46.06  6  7.25  8  1.97  9  7.55 10  8.04 11 12.75
20     1  3.82  2  2.27  3  7.88  5 39.30  6 14.14  8  8.38  9  0.35 11 18.07
21     2  1.77  5 65.58  6  3.67  8  1.87 10  1.92 11  3.54
22     1  7.70  2  6.21  3  3.32  5 36.27  6  4.26 11  3.95
23     1  1.10  2  9.95  3  7.11  4 19.21  5 40.07  6  1.77  8  4.91  9  1.27
23    10  7.86 11 19.53
24     3 10.04  5 46.32  7 13.45 11  0.92
00
PINU ECHCARY SPPQUER VELQUER RUBQUER PRIQUER ALBQUER COCNYSS SYLCORN FLOOXYD ARB
ACER RUBLIRI TUL
PINE  01PINE  02PINE  03PINE  04PINE  05PINE  06YELPR 07YELPR 08YELPR 09YELPR 10
YELPR 11YELPR 12OAKHK 13OAKHK 14OAKHK 15OAKHK 16OAKHK 17OAKHK 18CHNOK 19CHNOK 20
CHNOK 21CHNOK 22CHNOK 23CHNOK 24
```

Fig. 2 Data matrix in format acceptable to both TWINSPAN and ORDIFLEX.

3. INTERPRETATION OF THE OUTPUT

3.1 Reading the data matrix

The first part of the output (Fig. 3) concerns the input of the data matrix and the omission of samples. It is self-explanatory except for the print-out of the data matrix. Only the beginning and end of the matrix are printed, and all quantities are multiplied by 1000 and rounded to the nearest integer. Values of -1 are used to indicate the end of a sample. Thus a record

```
1  46000  5  2500  -1  1  56000  6  4500  -1
```

would refer to two samples, one with species 1 having quantity 46.0 and species 5 having quantity 2.5, and the other with species 1 having quantity 56.0 and species 6 having quantity 4.5. The matrix is printed out in the form that is held in the computer in order to remind the user of two things:

1. How the length of the raw data array relates to the number of non-zero items in the data matrix; and
2. That quantities smaller than 0.001 are lost in roundoff.

3.2 Entry of input parameters

This part of the output is also self-explanatory, and if the program is run interactively it merely repeats what was printed out on the console at run time. If the program is not being run interactively, then the input parameters should be checked carefully for correctness.

After the input parameters, three important statistics on the data are printed out:

1. Length of data array after defining pseudospecies;
2. Total number of species and pseudospecies; and
3. Number of species, excluding pseudospecies and ones with no occurrences.

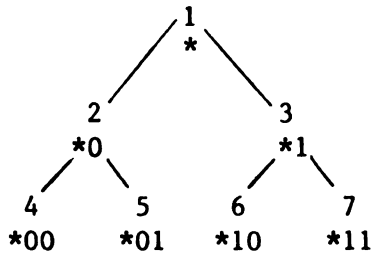
With big problems these statistics may be relevant to determining whether a species classification can be made as well as the sample classification (see Section 4.2), or whether the user can afford to use more cut levels for the pseudospecies (Section 4.1).

```
94   25MEADOW RELEVES (MUELLER-DOMBOIS & ELLENBERG 1974 182-183).  TCMEAD** S

NUMBER OF SAMPLES    25

NUMBER OF SPECIES    94

LENGTH OF RAW DATA ARRAY  1601
        1      6000        7       100       10       1000       11      50000       12       100       13       1000
       16      2000       20       100       21        100       23       1000       26      5000       27       1000
       32      2000       34     15000       37       3000       44       1000       45       100       48       3000
       52       100       57      1000       67       1000       68        100       69      4000       75        100
       77       100       81       100       85        100       86       1000       88       100       89        100
       91       100       -1         1      100          2      100          7     1000       10      1000         20
     1000        21      100        23      100         25      100         26     5000       27      1000         32
     1000        33     5000        37     2000         44     1000         51     1000       59       100         67
     1000        68      100        69    74000         75      100         81     2000       84      2000         86
      100        88      100        90     2000         -1        1       3000        7      2000        8         100
     ..........................................................................................................
     2000        -1        1      3000         3      100          7     30000        9      5000       11       10000
       12       100       13       100        20     2000         23      6000       25       100       26        8000
       33      2000       34      2000        37     3000         39       100       44      1000       45         100
       54       100       67       100        68      100         69     20000       74       100       75         100
       79      4000       85       100        88     2000         90      2000       91       100       92         100
       94       100       -1         1      100          2      100          4     1000        5       100          6
      100         7     35000        13      100         15     2000         17     1000       24      3000         26
    18000        33      3000        37     2000         41      100         45      100       46     15000         55
      100        56       100        58     4000         67      100         69    10000       71      4000         75
      100        77      1000        78      100         85      100         89      100       91      1000         -1

SPECIES NAMES
   1 ACHI MILL|   2 AJUG REPT|   3 ALCH VULG|   4 ALOP PRAT|   5 ANGE SILV|   6 ANTH SILV|   7 ARRH ELAT|   8 BELL PERE
   9 BRAC PINN|  10 BRIZ MEDI|  11 BROM EREC|  12 CAMP GLOM|  13 CAMP ROTU|  14 CARD PRAT|  15 CARE ACUT|  16 CARE FLAC
  17 CARE GRAC|  18 CARE HIRT|  19 CARE PANI|  20 CENT JACE|  21 CERA CAES|  22 CHEN ALBU|  23 CHRY LEUC|  24 CIRS OLER
  25 CREP BIEN|  26 DACT GLOM|  27 DAUC CARO|  28 DESC CAES|  29 DIAN SUPE|  30 EUPH ODON|  31 FEST ARUN|  32 FEST OVIN
  33 FEST PRAT|  34 FEST RUBR|  35 FILI ULMA|  36 GALI BORE|  37 GALI MOLL|  38 GALI ULIG|  39 GALI VERU|  40 GERA PRAT
  41 GEUM RIVA|  42 GLEC HEDE|  43 GLYC FLUI|  44 HELI PUBE|  45 HERA SPHO|  46 HOLC LANA|  47 KNAU ARVE|  48 KOEL PYRA
  49 LAMI ALBU|  50 LATH PRAT|  51 LEON HISP|  52 LINU CATH|  53 LOLI PERE|  54 LOTU CORN|  55 LYCH FLOS|  56 LYSI NUMM
  57 MEDI LUPU|  58 MELA DIUR|  59 MYOS ARVE|  60 MYOS PALU|  61 PAST SATI|  62 PHAL ARUN|  63 PHLE PRAT|  64 PHRA COMM
  65 PIMP MAGN|  66 PIMP SAXI|  67 PLAN LANC|  68 PLAN MEDI|  69 POA  PRAT|  70 POA  TRIV|  71 POLY BIST|  72 POLY CONV
  73 POTE REPT|  74 PRUN VULG|  75 RANU ACER|  76 RANU REPE|  77 RUME ACET|  78 RUME CRIS|  79 SALV PRAT|  80 SANG OFFI
  81 SCAB COLU|  82 SENE JACO|  83 SILA PRAT|  84 SILE INFL|  85 TARA OFFI|  86 THYM SERP|  87 TRAG PRAT|  88 TRIF PRAT
  89 TRIF REPE|  90 TRIS FLAV|  91 VERO CHAM|  92 VICI CRAC|  93 VICI SEPI|  94 VIOL HIRT|

SAMPLE NAMES
   1     1  |   2     2  |   3     3  |   4     4  |   5     5  |   6     6  |   7     7  |   8     8
   9     9  |  10    10  |  11    11  |  12    12  |  13    13  |  14    14  |  15    15  |  16    16
  17    17  |  18    18  |  19    19  |  20    20  |  21    21  |  22    22  |  23    23  |  24    24
  25    25  |

DO YOU WISH TO OMIT SOME SAMPLES?
ENTER NUMBERS (NOT NAMES) OF ITEMS TO BE OMITTED
ONE PER CARD, ENDING LIST WITH A -1.
OTHER NEGATIVE NUMBERS DENOTE SEQUENCES.  FOR EXAMPLE
A 4 FOLLOWED BY A -8 OMITS ITEMS 4 THROUGH 8.
-1
```

Fig. 3 Lineprinter output derived from reading data matrix into TWINSPAN. Sample names are here the same as sample numbers, but there is no necessity for this.

3.3 Classification of the samples

Divisions are made successively, according to the scheme below.



Each group is represented by two numbers, one in decimal, one in binary notation. If, in the hierarchy above, the symbol * is replaced by the number 1, then the decimal and binary numbers can be seen to be identical. Thus 10 is the representation of 2 in binary code, 110 represents 6, etc.

In general, group n is divided to obtain group 2n (the "negative" group) and group 2n + 1 (the "positive" group). The binary representation of the nodes of the hierarchy is more directly interpretable than the decimal representation, 0 denoting a left arm and 1 denoting a right arm.

Consider the output for a dichotomy line by line (refer to Fig. 4).

Line 1 gives the number of the group to be divided, both in decimal and in binary notation, together with the number of elements that it contains.

Line 2 gives information on the primary ordination (reciprocal averaging). Each iteration involves eight passes of the data.

Lines 3-5 specify the indicator ordination. Each sample is given an "indicator score" found by adding +1 for each positive indicator and -1 for each negative indicator that it contains. The sign (+ or -) of each indicator follows immediately after its name in the list of indicators. The division derived from the indicator ordination is specified on line 5. The two values printed are the maximum indicator score for a sample to be assigned to the negative group, and the minimum score for it to be assigned to the positive group. If these values are A and B respectively, then B is always 1 + A.

```
DIVISION    1  (N=  25)          I.E. GROUP *
EIGENVALUE 0.308  AT ITERATION   1
INDICATORS, TOGETHER WITH THEIR SIGN
CIRS OLER1(-)    BROM EREC1(+)    HOLC LANA1(-)    GEUM RIVA1(-)
MAXIMUM INDICATOR SCORE FOR NEGATIVE GROUP   -1      MINIMUM INDICATOR SCORE FOR POSITIVE GROUP    0
 1 2 3 4 5 6 7** 8 91011**121314151616****
**************************************
 0 0 0 0 0 0 0** 0 0 0 0** 0 0 0 1 6** 1
 0 0 0 0 0 0 0** 0 0 0 0** 0 0 1 0 0** 0
**************************************
 0 0 0 2 0 0 0** 0 0 0 1** 0 0 0 0 0**-1
 0 1 2 2 0 0 0** 0 0 0 0** 0 0 0 0 0**-2
 6 3 0 0 0 0 0** 0 0 0 0** 0 0 0 0 0**-3

ITEMS IN NEGATIVE GROUP   2  (N=   17)          I.E. GROUP *0
     5          6          7          8         11         12         13         14         16         17         18         19
    20         21         22         23         25

BORDERLINE NEGATIVES    (N=    1)
    16

ITEMS IN POSITIVE GROUP   3  (N=    8)          I.E. GROUP *1
     1          2          3          4          9         10         15         24

NEGATIVE PREFERENTIALS
AJUG REPT1(  14,  2) ALOP PRAT1(  9,  0) ANGE SILV1(  6,  0) CARE ACUT1(  5,  0) CIRS OLER1(  14,  0) DESC CAES1(  7,  0)
FILI ULMA1(   4,  0) GEUM RIVA1(  14,  0) GLEC HEDE1(  6,  0) HOLC LANA1(  12,  0) LATH PRAT1(  9,  1) LYCH FLOS1(  7,  0)
LYSI NUMM1(   8,  0) MELA DIUR1(  11,  0) PIMP MAGN1(  4,  0) POLY BIST1(  4,  0) RUME ACET1(  16,  3) TRAG PRAT1(  6,  1)
ALOP PRAT2(   8,  0) CARE ACUT2(  4,  0) CIRS OLER2(  8,  0) DESC CAES2(  6,  0) HELI PUBE2(  9,  1) HOLC LANA2(  6,  0)
PLAN LANC2(   8,  0) RUME ACET2(  4,  0) TARA OFFI2(  4,  0) ALOP PRAT3(  5,  0) CIRS OLER3(  4,  0) DESC CAES3(  4,  0)
GALI MOLL3(  10,  2) HELI PUBE3(  4,  0) TRIS FLAV3(  9,  1) ARRH ELAT4(  12,  2) CIRS OLER4(  4,  0) DACT GLOM4(  12,  1)
FEST PRAT4(   8,  0) ARRH ELAT5(  7,  1)

POSITIVE PREFERENTIALS
BRIZ MEDI1(   1,  4) BROM EREC1(  0,  7) CAMP GLOM1(  2,  5) CAMP ROTU1(  6,  7) CARE FLAC1(  3,  3) DIAN SUPE1(  2,  2)
FEST OVIN1(   0,  4) GALI BORE1(  2,  2) GALI VERU1(  0,  2) KNAU ARVE1(  1,  2) KOEL PYRA1(  0,  4) LEON HISP1(  2,  2)
LINU CATH1(   1,  4) LOTU CORN1(  2,  2) PIMP SAXI1(  0,  2) SALV PRAT1(  0,  5) SCAB COLU1(  0,  5) SILE INFL1(  1,  2)
THYM SERP1(   0,  5) VICI CRAC1(  2,  3) VICI SEPI1(  3,  4) VIOL HIRT1(  1,  5) BROM EREC2(  0,  7) FEST OVIN2(  0,  2)
FEST RUBR2(   4,  6) KOEL PYRA2(  0,  4) SALV PRAT2(  0,  4) VIOL HIRT2(  0,  2) BROM EREC3(  0,  7) CHRY LEUC3(  0,  2)
FEST RUBR3(   0,  2) BROM EREC4(  0,  7) POA  PRAT4(  5,  5) BROM EREC5(  0,  6) POA  PRAT5(  1,  2)

NON-PREFERENTIALS
ACHI MILL1(  13,  8) ANTH SILV1(  4,  1) ARRH ELAT1(  17,  8) BELL PERE1(  8,  3) CENT JACE1(  9,  5) CERA CAES1(  8,  5)
CHRY LEUC1(  15,  8) CREP BIEN1(  12,  3) DACT GLOM1(  17,  8) DAUC CARO1(  13,  7) FEST PRAT1(  17,  6) FEST RUBR1(  9,  6)
GALI MOLL1(  17,  8) HELI PUBE1(  11,  5) HERA SPHO1(  11,  3) MEDI LUPU1(  11,  6) PLAN LANC1(  17,  8) PLAN MEDI1(  8,  6)
POA  PRAT1(  17,  8) PRUN VULG1(  9,  3) RANU ACER1(  16,  7) SENE JACO1(  4,  2) TARA OFFI1(  13,  6) TRIF PRAT1(  12,  6)
TRIF REPE1(  11,  3) TRIS FLAV1(  10,  5) VERO CHAM1(  15,  7) ACHI MILL2(  7,  6) ARRH ELAT2(  17,  6) CARE FLAC2(  3,  2)
CENT JACE2(   6,  2) CHRY LEUC2(  5,  4) DACT GLOM2(  16,  8) FEST PRAT2(  17,  6) GALI MOLL2(  16,  7) POA  PRAT2(  15,  8)
RANU ACER2(   4,  1) TRIS FLAV2(  10,  5) ACHI MILL3(  4,  2) ARRH ELAT3(  13,  4) DACT GLOM3(  15,  7) FEST PRAT3(  10,  3)
POA  PRAT3(  10,  7) GALI MOLL4(  4,  1)

   END OF LEVEL   1
```

Fig. 4 First dichotomy resulting from application of TWINSPAN to the meadow data considered by Mueller-Dombois & Ellenberg (1974, pp. 182-183). The indicators have been underlined. It may seem surprising that the pseudospecies GALI BORE1(2, 2) is a positive preferential. The reason for this is that it occurs in 2/8 = 25% of the samples in the positive group, and in 2/17 = 11.8% of the samples in the negative group. Hence it is more than twice as likely to occur in the positive group than in the negative group.

In the example, only those samples with an indicator score of -1 or less are included in the negative group. Referring to the two-way table (Fig. 7), it can be seen that the borderline sample (number 16) contains only one indicator, Geum rivale. This gives it an indicator score of -1, assigning it to the negative group. Sample 2, on the other hand, contains no indicators at all, and hence has an indicator score of 0, assigning it to the positive group. At the other extreme, sample 8 contains all three of the negative indicators and hence has a score of -3.

The indicators are listed in an approximate order of effectiveness. Thus in Fig. 4, Cirsium oleraceum (negative) is a better indicator than Bromus erectus (positive), etc. It may seem strange that Geum rivale, dividing (14,0), is a worse indicator than Holcus lanatus, dividing (12,0). The reason for this is that the measure of effectiveness pays less attention to samples near the center of the refined ordination than to those at the extremes, and furthermore, that it measures effectiveness for a division at the center of the refined ordination rather than for the slightly off-center dividing line that is actually chosen (see below). Technically, the effectiveness of indicators is measured by the absolute value of the "preference index," defined in the remarks under parameter FEEBLE in Section 4.10.

Scatter diagram relating indicator ordination to refined ordination is not printed out in the default case, but is included here to show what is available. The lineprinter output is highly schematic, and corresponds to the relation shown in Fig. 5. To interpret the lineprinter output in Fig. 4, the following points should be noted.

1. For ease of computing the ordination is divided into segments. These have no special significance, but are merely a convenient way of calculating where to locate the zone of indifference. There are five possible positions for the zone of indifference such that it lies entirely within the critical zone (see Fig. 5). The location of the zone of indifference is selected to minimize the number of misclassified samples (see below for how these are defined).
2. Segments of the refined ordination are shown along top of scatter

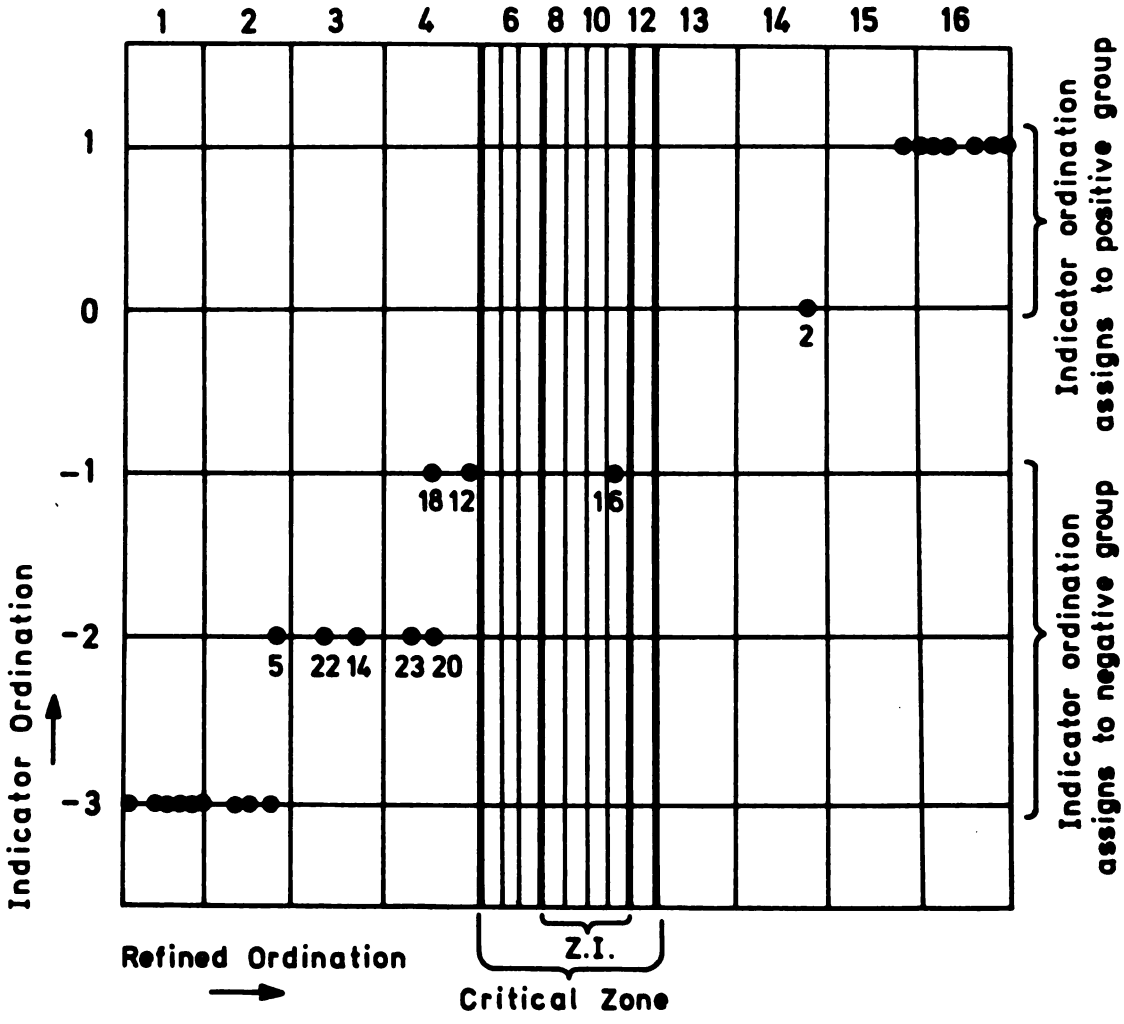


Fig. 5 Relation between indicator ordination and refined ordination (same dichotomy as Fig. 4). The segments of the refined ordination are indicated along the top of the diagram, and should be compared with the scatter diagram in Fig. 4. Segments 5-12 constitute the "critical zone," and segments 8-11 constitute the "zone of indifference" (abbreviated Z.I.) in the diagram. Samples are designated as "borderline cases" if they lie in the zone of indifference. In this example, only sample 16 is borderline.

diagram, which is not to scale. The critical zone (segments 5-12) is 20% of the length of the whole ordination. The length of the segments within the critical zone is one-quarter of the length of the segments outside it.

3. Indicator scores are shown along the right of the diagram. Indicator scores for assignment to the positive group are separated from those for assignment to the negative group by a row of stars.
4. The zone of indifference (here comprising segments 8-11) is indicated by columns of stars.
5. Values printed in the diagram indicate numbers of samples in each category. Thus the value 6 appears at the bottom left of the scatter diagram, corresponding to the six samples shown in Fig. 5 that occur in segment 1 and have indicator score -3.

Items in negative group are listed. The samples are specified by their names, which in this case are the same as their numbers.

Borderline negatives are those items that are in the negative group and which also lie in the zone of indifference (see Figs. 5 and 6). In Fig. 5, sample 16 is a borderline negative, and is assigned to the negative group because it has an indicator score of -1. The refined ordination is highly polarized, so that there are few borderline cases. In this respect, TWINSPAN differs from the version of indicator species analysis published by Hill, Bunce & Shaw (1975; see especially p. 602). In the original version there was no process of refinement, the "refined" and primary ordinations being identical; borderline cases were accordingly more numerous. A borderline case is assigned to its class according to its indicator score, the refined ordination being held in this case to be indecisive.

Misclassified negatives are those samples lying to the left of the zone of indifference but whose indicator score would assign them to the positive arm of the dichotomy (Fig. 6). This is regarded as a failure on the part of the indicator ordination to reproduce the dichotomy accurately, and hence as a "misclassification."

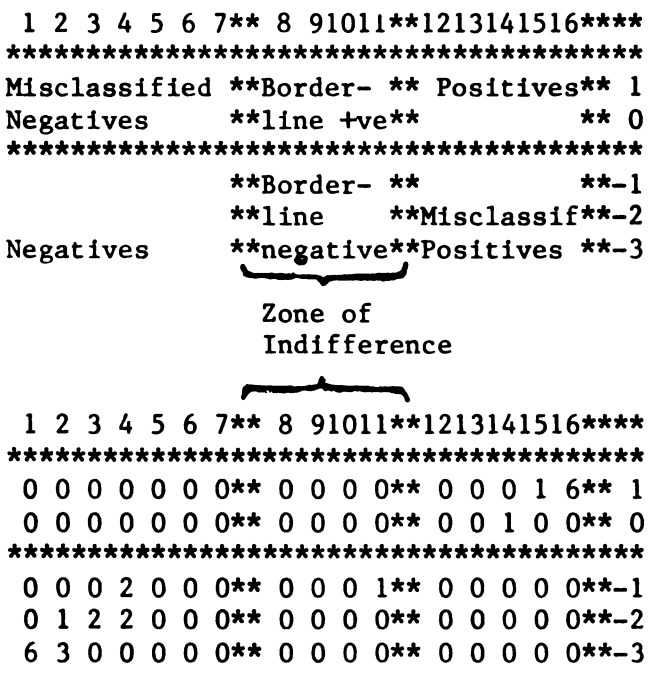


Fig. 6 Categories of samples in relation to indicator scores and zones of the refined ordination. The scatter diagram in Fig. 4 is reproduced here for the sake of comparison. Indicator scores are shown along the right-hand margin. Samples with indicator score -1 or less are assigned to the negative group unless they lie to the right of the zone of indifference, in which case they are categorized as "misclassified positives." Samples in the zone of indifference are categorized according to their indicator score.

Items in positive group, borderline positives, and misclassified positives are defined as for negatives.

Negative preferentials are those pseudospecies and species that are at least twice as likely to occur on the negative side as on the positive side. Only those that occur in at least 20% of the samples on the negative side are listed. Values given in brackets are actual numbers of occurrences, so that the entry

FEST PRAT4(8, 0)

signifies that Festuca pratensis occurs with abundance 4 or more (i.e. with cover at least 10%) in 8 samples on the negative side of the dichotomy and in no samples on the positive side of the dichotomy. This can be checked by referring to the two-way table in Fig. 7. It should be noted that when there is a very uneven split, negative preferentials can easily occur in more samples on the positive side of the dichotomy than on the negative side. Suppose, for example, that the split is into 2 samples on the negative side and 10 on the positive. Then a species that occurs in 2 samples on the negative side and 4 samples on the positive side is a good negative preferential. It has 100% frequency on the negative side and 40% frequency on the positive side, and is therefore 2.5 times as likely to occur on the negative side as on the positive side.

Positive preferentials are defined as for negative preferentials; non-preferentials are pseudospecies and species that are reasonably common and which are not preferentials. Here again, only those that achieve 20% frequency on one side or the other are listed.

3.4 Classification of the species

Species are classified by TWINSPAN in much the same way as samples. However, there is an important difference in that the species classification is made in the light of the sample classification, and not using the raw data. In fact, the species classification is made on the basis of fidelity, i.e. using the degree to which species are confined to particular groups of samples. For example, in Fig. 7, Cirsium oleraceum

and Phalaris arundinacea are both completely faithful to group *0 of the sample hierarchy--i.e. they have no occurrences outside this group. However, Phalaris is also completely faithful to group *00 of the sample hierarchy, whereas Cirsium is not.

Species are assigned attributes according to differing degrees of fidelity to sample groups (see Section 4.4 for details). For example, both Cirsium oleraceum and Phalaris arundinacea will be registered as having the attribute "very highly faithful to group *00," but Phalaris is also very highly faithful to group *00, whereas Cirsium is not. Indeed, Cirsium occurs widely outside group *00, in group *01.

Because the species classification is made on the basis of fidelity to groups defined by the sample classification, altering the level of division of the classification (cf. Section 2.8) may alter the species classification even if no other changes are made.

The species classification differs from the sample classification in that "indicator characters" are of little interest. It is scarcely of much significance to know that (for example) "low level of fidelity to sample group X" is a preferential attribute of a particular group of species. Consequently, the output has been shortened for the species classification, to present only the dichotomies, without the additional information that is supplied during the sample classification.

If the data set is very large, there may be no room for the new matrix of species' attributes as well as the original matrix. If so, then the species classification is abandoned (see Section 4.2 for conditions under which this will happen).

3.5 Order of species and samples

These orderings are printed out for two purposes. In the first place, the final table may not contain all the less common species, so that the orderings may be convenient as an indication of where the rarer species would come in the arrangement. Secondly and more important, the final tabulation refers to the samples by number, not by name as elsewhere in

the program. The reason for this is that it is difficult to write the names vertically, owing to the poor character-manipulation of the FORTRAN language. A previous version of TWINSPAN that did write the names vertically was found to be machine-specific, and was therefore unsuitable for issue as a general-purpose program.

Users with programming ability, who wish to print sample names in the table, should use the vectors INAME1, INAME2 instead of INFLAG in subroutine TABOUT.

3.6 Tabular arrangement of species and samples

The tabular arrangement (Fig. 7) needs little explanation. The hierarchy is shown in binary notation (omitting the *'s) along the right-hand and bottom margins. Numbers and names of species, and numbers of the samples, are shown along the left-hand and top margins.

The samples in Fig. 7 are rather homogeneous, and there are only two major groups. Sample 19 is anomalous in having several rare species abundant in it. Group *010 differs from *011 in having more of Carex acutiformis, Cirsium oleraceum, Deschampsia caespitosa, and Holcus lanatus, and less of Achillea millefolium and Trisetum flavescens; but the difference is a small one. Likewise, group *10 differs little from *11, the absence of Koeleria pyramidata being the main distinguishing feature. Further discussion of this set of data can be found in Maarel, Janssen & Louppen (1978).

3.7 Output to device 7

This output is intended to be read by machines rather than people. It lists the number, name, and class of each sample and each species. The classes are specified in both decimal and binary notation. (See Section 3.3 for an explanation of the class numbering.)

The purpose of writing out the solutions in machine-readable form is twofold. In some applications it may be desirable to subject a group derived from the classification to further analysis, e.g. to an ordination. Members of the selected group can be picked out by specifying their class

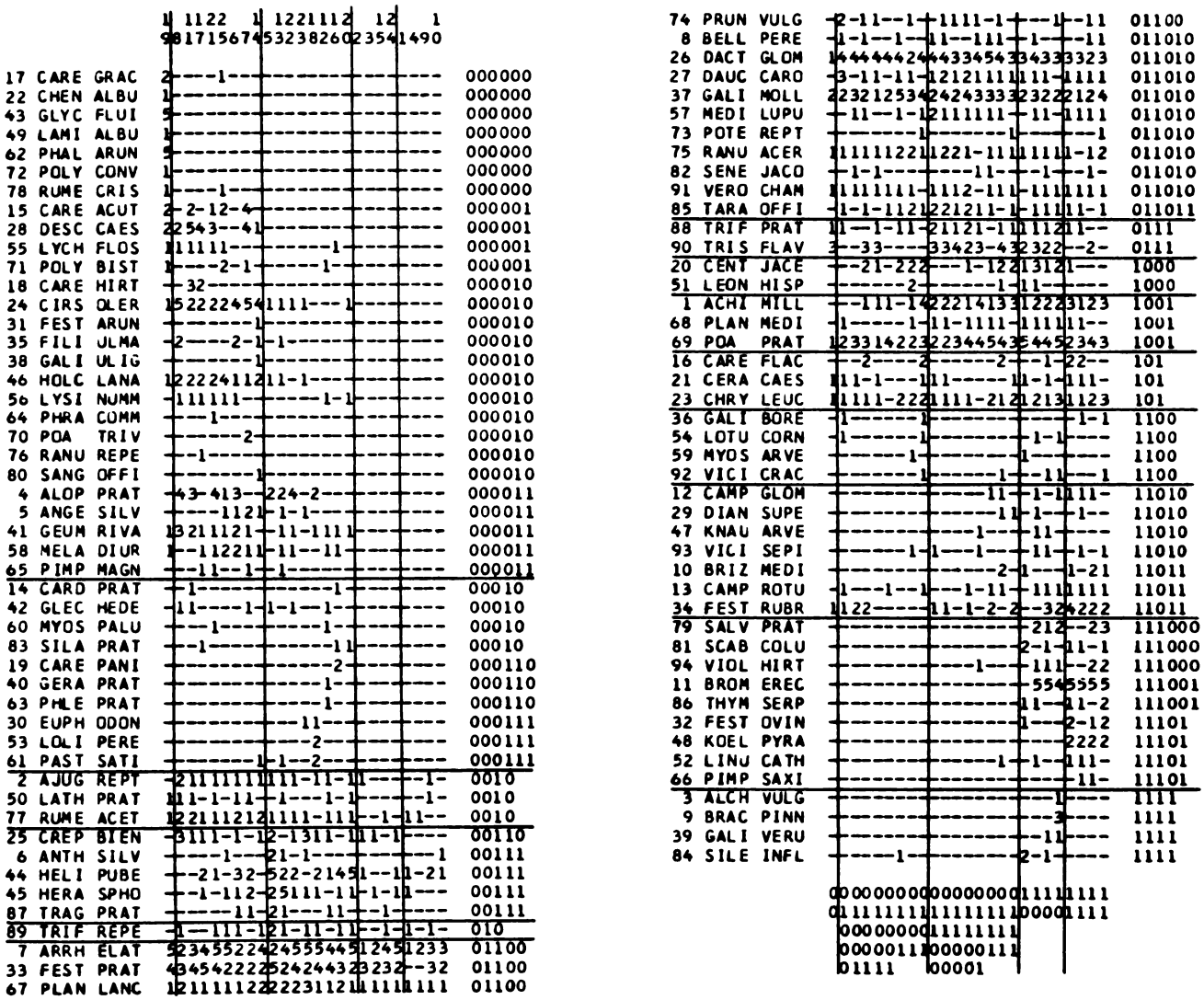


Fig. 7 Two-way table resulting from an application of TWINSPAN to the meadow data considered by Mueller-Dombois & Ellenberg (1974, pp. 182-183). Values denote categories of abundance defined by the pseudospecies cut levels. In this case the cut levels were the default values 0,2,5,10,20, and the raw data specified abundance by percent biomass. Values therefore denote categories of percent biomass:

1, 0-1% 2, 2-4% 3, 5-9%

4, 10-19% 5, 20-100%.

Vertical lines separate classes of samples at level 3; horizontal lines separate classes of species at level 4. Note that the arrangement is approximately on the positive diagonal.

number and scanning the list of samples and classes. Alternatively, it may be desired to tabulate the classification in another form. The list of classes can be used to do this.

4. TECHNICAL DETAILS OF THE PROGRAM AND ALGORITHM

4.1 Dimensioning

All subroutines have variable dimensioning so that only the main program needs to be modified. There are three dimensions that may need to be modified:

NDAT (= 30000)
MMAX (= 500)
NMAX (= 800).

NDAT is the length of the data array. The data are stored as indicated in Fig. 3, and are never expressed as a two-dimensional array except in the final tabulation. Hence, the magnitude of the calculation increases only linearly with the amount of data supplied. It depends mainly on the number of non-zero items in the matrix, and to a much lesser extent on the number of species or samples. It does, however, depend also on the number of pseudospecies, as information on each pseudospecies is stored separately. With large problems it is possible to overrun the array storage by asking for a large number of pseudospecies. In such cases an error message is printed, and the user has two options: either to reduce the number of pseudospecies, or to increase the value of NDAT.

To increase the value of NDAT it is necessary to modify lines 22, 37, and 49 of the program. It will be seen in line 22 that JDAT is half the length of IDAT; this proportionality should be maintained. Also, in line 37, JDAT(1) is equivalenced to IDAT(NDAT/2+1), and this relationship should be maintained. The intention here is to allow the sample data to overrun into space that is not required until the species classification is worked on. In some cases it is possible for NDAT to be big enough to allow the samples to be classified, but not big enough to allow the species classification to be made as well (see Section 4.2 for details).

MMAX is the maximum number of samples that can be considered. In the listing supplied here, MMAX is set to 500. Users with larger problems should modify lines 23-25, 32, 33, 50. If the number of species is greater

than MMAX, then the species classification is impossible. In this case a warning message is printed, and the calculation will terminate at the end of the sample classification.

NMAX is the maximum number of species and pseudospecies that can be handled. For example, if a problem has 700 species, and if the user wishes to use 3 levels of pseudospecies, then he will probably need to set NMAX to about 1200. The theoretical maximum of 2100 will not be required, as not all species will attain the abundance demanded by the top cut level. If the program needs a larger value of NMAX, then an appropriate error message is printed. To modify NMAX, lines 26-28, 34, and 51 will need to be altered.

NMAX must also exceed the maximum species number in the data matrix. This may be relevant if there are many species with no occurrences.

4.2 Dimensions necessary for species classification

To make the species classification, three conditions must be satisfied.

1. There must be enough room in the array JDAT for a secondary data matrix specifying fidelity attributes of the species (see Section 4.4). This condition is rarely likely to be violated.
2. The data when held in pseudospecies form must not occupy more than half the array IDAT. If they do occupy more than this, a warning message is printed.
3. The number of species (after eliminating those with no occurrences) must not exceed MMAX, the upper bound on the number of samples. If it does exceed MMAX, a warning message is printed.

In addition, a fourth condition must be met if the full sample classification is to be used in classifying the species.

4. Let the largest number designating a group of samples be ICMAX. Usually, ICMAX is approximately 2**(LEVMAX+1), where LEVMAX is the maximum level of division. Then NMAX must be greater than ICMAX*3. If this condition is not met, the lower-level groups of the sample classification are automatically amalgamated. This should not affect the output or the species classification

too seriously, but it will mean that the lowest-level sample groups are lost in the final tabulation.

Conditions 2 and 3 are the most likely to be violated, and the user will then get a warning. (Note that this is printed before the sample classification.) However, if the species classification is not made and no warning is printed, then condition 1 may be suspected of having failed, and NDAT should be increased. However, a simpler and better strategy is probably to make separate analyses of subsets of the data, using the output to device 7 to divide the samples into appropriate subsets (cf. Section 3.7).

4.3 Structure of the main program

The main program performs the following activities.

1. Values are set for a number of parameters (cf. Section 4.10).
2. The MM × NSPEC data matrix A = $[a_{ij}]$ is entered (subroutine QUIKIN). There are MM samples and NSPEC species. The matrix is stored in a linear array IDAT, in the form indicated in Section 3.1.
3. Parameters are supplied by the user (subroutine ENTER).
4. Quantities are converted to pseudospecies using cut levels supplied by the user (subroutine PSEUDO; cf. Section 4.5). The total number of pseudospecies is NN, and NSPEC is revised downwards if there are species with no occurrences in the matrix.
5. The samples are classified (subroutine CLASS; see Section 4.6).
6. Attributes are defined for the species, using the degree of fidelity (sensu Braun-Blanquet) of species to the groups of samples derived by the sample classification (see Section 4.4 for details). A data matrix for the species is put in JDAT, and a version of the original data matrix is retained in IDAT.
7. The species are classified (second application of subroutine CLASS).
8. The species and sample classifications are converted to orderings. This may involve reversing the order of the species, so as to get the data matrix on the positive diagonal.
9. The data matrix is rearranged so that within each sample the species appear in the correct order obtained at Activity 8 above (subroutine JORDER).
10. The final two-way table is printed out (subroutine TABOUT).

4.4 Conversion of data matrix for species classification

Activity 6 above is basic to the species classification. The rationale behind using fidelity is that species should be considered as similar or dissimilar not according to whether they are common or rare, but according to whether they occur in similar sample groups. Of course, an exceedingly common species such as Arrhenatherum elatius in Fig. 7 will not be highly faithful to any group, and will therefore not appear similar to any exceedingly rare species. However, A. elatius is classified in the group 011 along with several relatively uncommon species such as Potentilla reptans and Senecio jacobaea. These species resemble Arrhenatherum in that they show a poor relation to the classification of the samples.

Fidelity is defined as follows. Consider the class of samples whose index is IC (refer to Section 3.3 for an explanation of the index numbers of the classes). Then the fidelity of species J to class IC is defined by the ratio

$$RAT(IC,J) = \frac{\text{mean occurrence of J in class IC}}{\text{mean occurrence of J in individuals not in IC}} .$$

In this context, the mean occurrence of J is defined to be the unweighted mean value of b_{ij} for those samples i that occur in the group under consideration; where the matrix elements b_{ij} are those printed out in the final tabulation (i.e. b_{ij} = C if species j achieves cut level C in sample i).

Using the fidelity ratio RAT, it is possible to define attributes for the species. Species J is deemed to have attribute F(IC,K) if RAT(IC,J) is greater than or equal to the limit SPE(K). Three values of SPE are used here (lines 100-102 of source code), and are:

SPE1 = 0.8; SPE2 = 2.0; SPE3 = 6.0.

In the species classification, both the species and their attributes are given differing weights, as follows.

1. Extra weight for high fidelity. A parameter WTHIGH, set at the beginning of the program to 2.0, is used to give extra weight to

the higher fidelities. In this case, attributes of the form F(IC,2) and F(IC,3) are given twice the weight of F(IC,1).

2. Extra weight for commoner species. In most applications it is undesirable to have numerous small groups of similar rare species segregating off at an early stage of the classification. This can be avoided by weighting the species according to their abundance, in which case the rarer species are unlikely to dominate the classification, and will tend to align themselves with commoner ones. The weighting used here is

$$w_j = \sum_i b_{ij} \ ;$$

where the matrix elements b_{ij} are (as in the definition of RAT) those that appear in the final tabulation.

3. Extra weight for larger groups and for the higher levels of the hierarchy in the sample classification. Weights of the attributes F(IC,K) corresponding to the class IC are multiplied by

$$w_{IC} = 2^{-L/2} \sum_{i \epsilon IC} \sum_j b_{ij} \ ,$$

where the summation $i \epsilon IC$ is taken over those samples i that belong to group IC and L is the level of group IC. In effect, this means that each level down the hierarchy is given weight $\sqrt{2}$ less than that above, and that less attention is paid to fidelity to small groups than to large ones.

4.5 Subroutine PSEUDO

This constructs pseudospecies for the sample classification (cf. Section 1.7), and needs little comment except to note that the real species to which the pseudospecies are attached are identified by the vectors JNFLAG and INDPOT. Specifically, JNFLAG(INDPOT(J)) is the number of the real species corresponding to pseudospecies J. If J is a bottom-level pseudospecies (usually this will mean that it is a real species), then INDPOT(J) = J. INDPOT is so called because it is used to determine the indicator potential of a species for the indicator ordination. If either

INDPOT(J) or INDPOT(INDPOT(J)) is negative, then J is barred from being used in the indicator ordination. This device is used to ensure that no one real species is used more than once in the indicator ordination-- i.e. if Poa annua 2 is an indicator, then Poa annua 1, P. annua 3, etc., are barred from being indicators in the same dichotomy. There is no constraint preventing a species from being used as an indicator in many different dichotomies.

4.6 Subroutine CLASS

This subroutine makes the classification. It may be divided into the following activities.

1. A parameter IEND is set equal to 2**(LEVMAX+2), which is bigger than the largest index number of any class that can result from divisions down to level LEVMAX. Any individuals that lack all attributes are identified at the outset and assigned to class IEND. Later on in the program, IEND is added to the index number of any class that is not to be divided further.
2. Various housekeeping activities are undertaken (subroutines DECODE, UPDATE, and RECODE). The name of subroutine UPDATE expresses its function well. It takes a newly made dichotomy, updates the class register ICLASS to allow for the new dichotomy, and looks for a new class to be divided. If no class remains to be divided, a switch ISTOP is set to 1. The subroutines RECODE and DECODE establish and remove a temporary numbering of the individuals and attributes for use during a particular dichotomy. The attributes are recoded so that they run consecutively from 1 to N, and the individuals are renumbered to run from 1 to M.
3. The rarer attributes are downweighted according to the following rules:
   (a) All attributes with frequency greater than FRQLIM (= 0.2) are given unit weight;
   (b) Attributes with frequency less than FRQLIM are given weights on a linear sliding scale in proportion to their frequency, starting with weight CWTMIN (= 0.01) for very rare attributes and rising to weight 1.0 for attributes with frequency FRQLIM.

   The purpose of this downweighting is to reduce the effect of the

rarer attributes. It is possible to modify the degree of down-weighting by changing parameter values (see Section 4.10).

4. The individuals are ordinated by reciprocal averaging (Hill, 1973) in subroutine RA. Note that in addition to weights applied at Activity 3, the reciprocal averaging is weighted by weights RRWT for the individuals and CCWT for the attributes. In the sample classification the weights RRWT are set to 1.0, as are the weights CCWT in the standard analysis. However, CCWT may be varied by the user so as to be different for different cut levels (see Section 2.11). In the classification of the species, both RRWT and CCWT are set to non-unit values (see Section 4.4).
5. The primary ordination is refined by application of subroutine POLISH (see Section 4.7).
6. The order of the two groups resulting from the dichotomy is considered, and is arranged to fit in as well as possible with the rest of the table (for details see Section 4.8).
7. The indicator ordination is defined in accordance with the ideas presented by Hill, Bunce & Shaw (1975)--subroutines ZONEUP, INDSCO, TOPIND, CODESC, TABLE, and FIND. If the maximum number of indicators is set to zero (cf. Section 2.6), then several of these subroutines are omitted. They are always omitted in the classification of the species.
8. Information on the dichotomy is printed out (subroutine REPORT).

4.7 Refinement of the ordination

Refinement of the ordination is performed by subroutine POLISH. Essentially, this amounts to the application of a transfer algorithm (Gower, 1974; synonymously an iterative relocation algorithm--see Cormack, 1971), using a discriminant function to make the transfers. The primary ordination is converted to a dichotomy by dividing it at its center of gravity, and the frequencies of attributes on the positive and negative sides of the dichotomy are then used to construct a new ordination (and hence a new dichotomy). There is allowance both for borderline cases and for prior weighting of the individuals (for details see Section 4.10 and subroutine INDSCO), but this is ignored in the description given below.

The discriminant function is derived by adding together two ordinations. The first of these is additive, the second is gotten by taking a mean. The first ordination will generally be the dominant partner at the lower levels of the hierarchy; the second will be dominant at the higher levels.

The first ordination is gotten by adding together the "preference scores" of the commoner and more preferential species. A preference score of +1 is given to each attribute that is at least three times more frequent on the positive side as on the negative side, and which is commoner than a certain minimum. Common negative preferentials are treated similarly, and are given a score of -1. Attributes that are rarer, or which are less markedly preferential, are downweighted accordingly. In symbols, let an attribute J have frequency AYY on the positive side, and frequency AY on the negative. Define

PREF = (AYY - AY)/(AYY + AY)

FREQ = AYY + AY

Then the score given to the attribute is

Preference score = (FREQ/FRQLIM) * (PREF/PRLIM)**5 .

Note that this downweighting is only applied when either FREQ falls below the threshold FRQLIM (= 0.2; this is--unnecessarily--the same parameter as is used in Section 4.6, Activity 3), or when the absolute value of PREF falls below PRLIM (= (RATLIM - 1)/(RATLIM + 1), where RATLIM is the frequency-ratio limit, here set to 3.0). Clearly, rare attributes are mildly downweighted, but attributes with an insufficient preference ratio are greatly so.

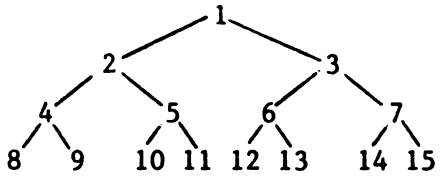
The first ordination is completed by standardizing the values so that the maximum absolute value is 1.

The other ordination in POLISH is gotten simply by taking the mean value of PREF/PRLIM (truncated to a maximum absolute value of 1) in the individual. This allows rarer attributes to express an opinion as well.

Finally, the refined ordination is gotten by adding these two. In practice, this process polarizes the ordination quite strongly, so that there are few borderline cases (cf. Section 3.3).

4.8 Ordering of dichotomies

Consider the following hierarchy, and in particular the question of how to order the pairs 10,11 and 14,15.



To get as smooth an arrangement as possible, we want 10 to be like 9, and 11 to be like 12; and we want 14 to be like 13, whereas 15 should be relatively unlike 13. One way to achieve this aim would be first to complete all the divisions at a particular level of the hierarchy, and then to try swivelling the dichotomies to see which ordering is the most satisfactory. An alternative approach, adopted here, is to try to see whether 10 or 11 more closely resembles 4, and whether 14 or 15 more closely resembles 6. This has the advantage that groups at level 4 are compared with those at level 3, so that the correct order can be determined at the time each dichotomy is formed. It has the added advantage that it compares the new groups with what are likely to be relatively large groups, so that order is determined more by general relationships than by particular accidents.

The basic ordering criterion in TWINSPAN, then, is to take two groups (2n) and (2n + 1) and to compare them with (n - 1) if n is odd and with (n + 1) if n is even. The comparison is made by setting up a discriminant function to distinguish (2n) from (2n + 1), and then asking where this discriminant would place the comparison group (which is either (n - 1) or (n + 1), according as n is odd or even). This comparison is made by subroutine CLOSER.

This simple ordering criterion is complicated by the fact that the groups (2n) and (2n + 1) are compared not only with one or other of (n - 1)

and (n+1), but also with one or other of (n/2-1) and (n/2+1). That is to say, not only is the local order considered, but also the relation to groups two stages back in the hierarchy. Let YSCOR1, YSCOR2 be the values of the discriminant function for comparisons one and two levels back in the hierarchy. Then the final discriminant is

YSCORE = YSCOR1 + W*YSCOR2 ,

where W is a weight function set to +0.5 or -0.5 (the sign depending on the remainder when n is divided by 4). This device pushes responsibility for the order even further back in the hierarchy, with the aim of achieving greater stability in the ordering. In particular, if the division at level 2 was effectively orthogonal to that at level 3, then the order dictated by relations at level 2 may be irrelevant to those at level 3; whereas relations at level 1 may still indicate a clear preference.

The discriminant function used in subroutine CLOSER is as follows. Consider the ratio

PREF = ABS(AY0 - AY1)/(AY0 + AY1) ,

where AY0 is the frequency of the attribute on the negative side of the dichotomy, and AY1 is the frequency on the positive side. Now take

PREF = PREF/0.3 ,

and set all those values of PREF that are bigger than 1 to 1. In this way we obtain a function that is 1 for all attributes that have a frequency ratio greater than about 1.9, and is less than 1 for lower values, dropping to 0 for complete indifference. (The value 1.9 arises because $0.3 \approx (1.9-1)/(1.9+1)$.) Now each attribute is scored for negativeness (PNEG), indifference (PIND), or positiveness (PPOS), as follows.

Case 1. The attribute is more frequent on the negative side.
Then PNEG = PREF**4; PIND = 1 - PNEG; PPOS = 0.
Case 2. The attribute is more frequent on the positive side.
Then PNEG = 0; PIND = 1 - PPOS; PPOS = PREF**4.

Using this scoring system, any individual or group of individuals can be assessed for negativeness, positiveness, or indifference; and in

particular, the positive and negative sides of the dichotomy can be so assessed. Let the mean scores per individual on the negative side of the dichotomy be XONEG, XOIND, and XOPOS; and let the mean scores per individual on the positive side be X1NEG, X1IND, and X1POS. The idea is now to assign scores that pay as little attention as possible to the indifferent attributes, while still requiring that the mean score on the negative side should be equal and opposite to that on the positive side. This is achieved by scoring -1 for negativeness, +1 for positiveness, and adjusting the score for indifference to make up the difference. In symbols, the score for indifference is

$$(XONEG + X1NEG - XOPOS - X1POS)/(XOIND + X1IND) .$$

The only additional requirement is that the score for indifference should not exceed +1 or be less than -1. If it does exceed these limits, then it is reduced to +1 or -1 as appropriate.

There is a small additional feature that needs to be noticed, namely that in the discriminant function for ordering groups of samples, the various pseudospecies corresponding to a given real species are treated merely as additional occurrences of the one real species. (They are recognized as belonging to one another through the vector INDPOT; see Section 4.5.) For the species classification, on the other hand,

$$INDPOT(J) = J$$

for all values of J. Hence the differing degrees of fidelity are not recognized as belonging to one another.

The score for an individual is calculated by adding

$$PPOS - PNEG + (\text{Score for indifference} * PIND)$$

for all the attributes possessed by the individual. Finally, the score for a whole group is gotten by adding together the scores of all the individuals that it contains.

4.9 Subroutine RA

The algorithm used here for reciprocal averaging is a direct iteration algorithm comparable to that used by Hill (1973), but less susceptible

to round-off error. The essence of the algorithm is to represent the two-way averaging iteration approximately by a 4×4 symmetric tridiagonal matrix. The top eigenvector of the 4-dimensional approximation is then sought, and is used as a trial vector for the next iteration.

4.10 Significance of parameter values

Lines 53-101 of the program are devoted to setting the values of various parameters. Many of these parameters have already been mentioned, but are considered briefly again here to explain the result of varying them.

NUMTAB determines the number of samples printed out per page of the two-way table. This may need to be reduced for lineprinters with less than 124 characters per line (see Section 2.1).

RARE determines which species are to be included in the listing of preferential species at each dichotomy (see Section 3.3). At present this list is rather long (cf. Fig. 4), and it could be shortened by setting RARE to 0.4. In this case, only species and pseudospecies occurring in at least 40% of the samples on one or other side of the dichotomy would be listed.

FEEBLE is used for rejecting indicators that are too feeble. An initial dichotomy is made by dividing the refined ordination at its center. Attributes are then graded on the basis of their preference index, where

> Preference index = chance of occurring on positive side of dichotomy - chance of occurring on negative side of dichotomy.

For example, an attribute that occurs in 30% of samples on the positive side of the dichotomy and in 10% of samples on the negative side has a preference index of $0.3 - 0.1 = 0.2$. With the setting FEEBLE = 0.1 as supplied, such an attribute is then available as an indicator. Two points should be noted.

1. The absolute value of the preference index is used in applying FEEBLE, as the intention is to identify all good indicators, whether positive or negative.

2. Some allowance is made for borderline samples, giving them less weight than samples that are definitely positive or negative. For samples outside the borderline zone defined by CRCUT (see below), however, the criterion described above is applied exactly.

FRQLIM is a limit used for downweighting rare attributes, both in subroutine WEIGHT (see Section 4.6, Activity 3), and in subroutine POLISH (see Section 4.7). Lowering the value of FRQLIM will mean that fewer attributes are downweighted. This will make the analysis more inclined to separate small, deviant groups of samples at an early stage in the hierarchy.

TOL is the tolerance for deciding whether an eigenvector has been obtained in subroutine RA. Let x be a trial vector, standardized to unit length, and let Lx be the result of applying the two-way averaging of reciprocal averaging, also standardized to unit length. Then x is counted as an eigenvector when length of (x - Lx) is less than TOL. There is little point in working to a high accuracy with the primary ordination, and the supplied value of TOL (= 0.003) appears to be quite adequate.

RATLIM is a limit used in subroutine POLISH (see Section 4.7). As supplied, it is set to 3.0, so that attributes that are less than three times as frequent on one side of the dichotomy as on the other are given less weight in refining the ordination.

REPLIM is a similar limit used in subroutine REPORT. As supplied it is set to 2.0, so that attributes that are less than twice as frequent on one side of the dichotomy as on the other are counted as non-preferentials (cf. Section 3.3). Raising the value of REPLIM will reduce the number of species and pseudospecies that are counted as preferentials, but will not affect the number of these that are printed out in all (unlike RARE; see comments above).

IREWT is the number of times that subroutine POLISH is applied per dichotomy. The value of 2 used here was found to be the best empirically,

but if the method of refining the ordination is changed, there is no reason why another value of IREWT might not be better.

ICWEXP is power of downweighting for species not up to limit FRQLIM. Rarer species can be more heavily downweighted by raising the value of ICWEXP (see lines 1530 and 1751 of source code).

IPREXP is a similar exponent for preferentiality in POLISH (see Section 4.7). In fact the power of downweighting used in POLISH is one greater than IPREXP, which explains why an exponent **5 is mentioned in Section 4.7, whereas IPREXP is set to 4.

CWTMIN is a minimum weight to be given to attributes in subroutine WEIGHT. See Section 4.6, Activity 3.

CRCUT determines the length of the zone used in defining borderline cases for the purposes of refining the ordination. (It was mentioned in Section 4.7 that borderline cases are given special treatment.) As supplied, CRCUT is set to 0.2. This means that samples in the middle 20% of the range of the ordination are downweighted in determining preferentiality, from zero weight at the center of the ordination, rising on a linear scale to unit weight at distances 40% and 60% from the end.

CRLONG determines the length of the critical zone, for the purposes of defining misclassifications, etc. As supplied, CRLONG is 0.2. This is why the critical zone in Fig. 4 (Section 3.3) is 20% of the total length of the ordination. If it is felt that too many samples are being classified as "borderline cases" then this can be corrected by lowering the value of CRLONG.

MZCRIT, MZIND, and MZOUT are best interpreted by reference to Fig. 4. MZCRIT is the number of segments in the critical zone (here set to 8). MZIND is the number of segments in the zone of indifference (here set to 4), and MZOUT is the number of segments on each side of the diagram outside the critical zone (here set to 4 also). One way to reduce the number of samples that are classified as "borderline cases" would be to

reduce the value of MZIND from 4 to 2, halving the length of the zone of indifference, so that it drops from 10% to 5% of the total length of the ordination.

WTHIGH, SPE1, SPE2, and SPE3 are parameters used in converting the data matrix for the species classification (see Section 4.4 for details).

4.11 Outlook

TWINSPAN is long and rather complicated. It would be a better program if it were simpler, concentrating more directly on users' particular requirements. For example, many users will be more interested in getting a good two-way table than in the niceties of the indicator ordination. The indicator ordination requires several hundred lines of programming code, and could perhaps be omitted in a future edition.

Another feature that could perhaps be improved is the process of refining the ordination. The discriminant function used to order the dichotomies is more elegant than that used in subroutine POLISH, and could well be used also to refine the ordination. This modification, also, would save programming code.

It would be desirable to look more than two levels back in determining the order of dichotomies (cf. Section 4.8). Thus, if two small and aberrant groups are successively split off a larger group, and if the larger group is split again, then the ordering of the two groups resulting from the third dichotomy will be determined by their similarity to the two small groups split off previously, and not to the general trend of the two-way table.

For the present, however, the program is being allowed to stand. A more careful study of discriminant functions needs to be made before any substantial progress is possible.

5. ACKNOWLEDGMENTS

TWINSPAN was written during a year of paid leave at Cornell University, granted to me by my employer, the Natural Environment Research Council of Great Britain. I am grateful to them, and especially to the institute for which I work (Institute of Terrestrial Ecology) for making this opportunity possible. Funds for computing and travelling were provided by the U.S. National Science Foundation Grant DEB 78-09340.

Dr. R. H. Whittaker invited me to visit Cornell, and encouraged me to pursue the possibilities of numerical classification. I am indebted to him, to H. G. Gauch, Jr., and to Dr. P. L. Marks for the provision of test data and much helpful criticism.

Finally I should like to thank S. B. Singer for his expert advice about the Cornell computer and for writing several utility programs which have been indispensable in preparing TWINSPAN.

6. REFERENCES

Cormack, R. M. (1971). A review of classification. J. R. Statist. Soc., Series A, 134, 321-367.

Gauch, H. G. (1977). ORDIFLEX -- A flexible computer program for four ordination techniques: weighted averages, polar ordination, principal components analysis, and reciprocal averaging, Release B. Ecology and Systematics, Cornell University, Ithaca, New York 14850.

Gower, J. C. (1974). Maximal predictive classification. Biometrics 30, 643-654.

Hill, M. O. (1973). Reciprocal averaging: an eigenvector method of ordination. J. Ecol. 61, 237-249.

Hill, M. O. (1977). Use of simple discriminant functions to classify quantitative phytosociological data. First International Symposium on Data Analysis and Informatics, Versailles 7-9 Sept. 1977 (Ed. by E. Diday, L. Lebart, J. P. Pages, & R. Tomassone), Vol. 1, pp. 181-199. Institut de Recherche d'Informatique et d'Automatique, Domaine de Voluceau, Rocquencourt, B.P. 105, 78150 le Chesnay, France.

Hill, M. O. (1979). DECORANA -- A FORTRAN program for detrended correspondence analysis and reciprocal averaging. Ecology and Systematics, Cornell University, Ithaca, New York 14850.

Hill, M. O., Bunce, R. G. H., & Shaw, M. W. (1975). Indicator species analysis, a divisive polythetic method of classification, and its application to a survey of native pinewoods in Scotland. J. Ecol. 63, 597-613.

Maarel, E. van der, Janssen, J. G. M., & Louppen, J. M. W. (1978). TABORD, a program for structuring phytosociological tables. Vegetatio 38, 143-156.

Mueller-Dombois, D. & Ellenberg, H. (1974). Aims and Methods of Vegetation Ecology. Wiley, New York.

Singer, S. B. & Gauch, H. G. (1979). CONDENSE -- Convert data matrices from any ORDIFLEX format into a condensed format by samples. Ecology and Systematics, Cornell University, Ithaca, New York 14850.

Westhoff, V. & Maarel, E. van der. (1973). The Braun-Blanquet approach. Ordination and Classification of Communities (Ed. by R. H. Whittaker), pp. 617-726. Junk, The Hague.

7. LISTING OF SOURCE CODE

```
C CORNELL ECOLOGY PROGRAM TWINSPAN - WRITTEN BY M.O. HILL, JULY 1979.    TWI     1
C SOURCE CODE AND ACCOMPANYING DOCUMENTATION ARE AVAILABLE FROM          TWI     2
C HUGH G. GAUCH, JR., ECOLOGY AND SYSTEMATICS, CORNELL UNIVERSITY        TWI     3
C ITHACA, NEW YORK 14850.                                                TWI     4
C                                                                        TWI     5
C PERFORMS TWO-WAY INDICATOR SPECIES ANALYSIS, ARRANGING SPECIES AND     TWI     6
C SAMPLES IN A TWO-WAY TABLE AT THE END.                                 TWI     7
C                                                                        TWI     8
C DIMENSIONED FOR 500 SAMPLES, 800 SPECIES, 14500 NON-ZERO ITEMS IN      TWI     9
C DATA-MATRIX.  IDAT IS DIMENSIONED 30000=2*(14500+500).  JDAT IS        TWI    10
C DIMENSIONED WITH HALF THE LENGTH OF IDAT, AND THE FRONT OF IT IS       TWI    11
C EQUIVALENCED TO THE MIDDLE OF IDAT.                                    TWI    12
C                                                                        TWI    13
C INPUT DEVICES ARE 3 (INPUT PARAMETERS - SHOULD BE A COMPUTER           TWI    14
C TERMINAL IF POSSIBLE), 5 (DATA-MATRIX).                                TWI    15
C                                                                        TWI    16
C OUTPUT DEVICES ARE 4 (MONITORING - SHOULD BE CONSOLE OF COMPUTER       TWI    17
C TERMINAL IF POSSIBLE; OTHERWISE THIS OUTPUT IS DISPENSABLE);           TWI    18
C 7 (COPY OF SOLUTION IN MACHINE-READABLE FORM - OPTIONAL AND ONLY       TWI    19
C SUPPLIED IF REQUESTED);  AND LINEPRINTER USING FORTRAN WORD PRINT.      TWI    20
C                                                                        TWI    21
      INTEGER IDAT(30000),JDAT(15000)                                    TWI    22
      INTEGER IX(500),ICLASS(500),IIROW(500),IADDR(500)                  TWI    23
      INTEGER INFLAG(500),INAME1(500),INAME2(500),IZONE(500)             TWI    24
      INTEGER IX1(500),IX2(500),IX3(500)                                 TWI    25
      INTEGER IY(800),JJCOL(800),INDPOT(800),INDORD(800)                 TWI    26
      INTEGER JNFLAG(800),JNAME1(800),JNAME2(800),JNAM(800)              TWI    27
      INTEGER IY0(800),IY1(800),IY2(800)                                 TWI    28
      INTEGER ITEM(200)                                                  TWI    29
      INTEGER LCUT(10),LWGT(10),LIND(10)                                 TWI    30
      INTEGER INDSIG(20),IPICT(104,25)                                   TWI    31
      REAL X(500),XX(500),RTOT(500),RRWT(500),ROWWGT(500)                TWI    32
      REAL X3(500),X4(500),X5(500)                                       TWI    33
      REAL Y(800),YY(800),CTOT(800),CCWT(800),COLWGT(800)                TWI    34
      REAL TOT(511),TOTJ(511)                                            TWI    35
      REAL TITLE(20)                                                     TWI    36
      EQUIVALENCE (IDAT(15001),JDAT(1))                                  TWI    37
      EQUIVALENCE (IX1(1),X3(1)),(IX2(1),X4(1)),(IX3(1),X5(1))           TWI    38
      EQUIVALENCE (IY0(1),COLWGT(1)),(IY1(1),CTOT(1)),(IY2(1),CCWT(1))   TWI    39
      COMMON/PICT/MZ,MS,IZD,IIZD,ISD,ISHIFT,MZIND,MZCRIT,                TWI    40
     1MZOUT,NIND                                                         TWI    41
      COMMON/LIMS/RARE,FEEBLE,FRQLIM,TOL,RATLIM,REPLIM                   TWI    42
      COMMON/SWITCH/IFAIL,IDIAGR,ISTOP,IREWT                             TWI    43
      COMMON/IARBS/ICWEXP,IEND,MMIN,IPREXP,LEVMAX                        TWI    44
      COMMON/ARBS/CWTMIN,CRLONG,CRCUT                                    TWI    45
      COMMON/WORK/ITEM                                                   TWI    46
      COMMON/SECOND/ISEC                                                 TWI    47
      COMMON/SPESPE/WTHIGH,SPE1,SPE2,SPE3                                TWI    48
      NDAT=30000                                                         TWI    49
      MMAX=500                                                           TWI    50
      NMAX=800                                                           TWI    51
C THIS IS THE END OF DECLARATIONS AND SETTING OF DIMENSIONS.             TWI    52
      NUMTAB=100                                                         TWI    53
C NUMTAB IS MAXIMUM NUMBER OF SAMPLES PER PAGE IN FINAL TABULATION       TWI    54
      RARE=0.2                                                           TWI    55
```

```
C RARE IS LIMIT OF RARITY FOR USE IN LISTING PREFERENTIAL SPECIES       TWI   56
      FEEBLE=0.1                                                        TWI   57
C INDICATORS WITH PREFERENCE INDEX LESS THAN FEEBLE ARE REJECTED        TWI   58
      FRQLIM=0.2                                                        TWI   59
C FRQLIM IS LIMIT OF RARITY FOR DOWNWEIGHTING. SPECIES OCCURRING IN     TWI   60
C A PROPORTION OF SAMPLES LESS THAN FRQLIM ARE DOWNWEIGHTED IN THE      TWI   61
C ORDINATION.                                                           TWI   62
      TOL=0.003                                                         TWI   63
C TOL IS TOLERANCE FOR DECIDING WHETHER A TRIAL VECTOR IS EIGENVECTOR   TWI   64
      RATLIM=3.0                                                        TWI   65
C RATLIM IS THE FREQUENCY-RATIO LIMIT USED IN SUBROUTINE POLISH         TWI   66
      REPLIM=2.0                                                        TWI   67
C REPLIM IS THE FREQUENCY-RATIO LIMIT USED IN SUBROUTINE REPORT         TWI   68
      IREWT=2                                                           TWI   69
C IREWT IS NUMBER OF TIMES SUBROUTINE POLISH TO BE CALLED PER DIVISION  TWI   70
      ICWEXP=1                                                          TWI   71
C ICWEXP IS POWER (I.E. EXPONENT) OF DOWNWEIGHTING FOR SPECIES NOT UP   TWI   72
C TO RARITY LIMIT FRQLIM.                                               TWI   73
      IPREXP=4                                                          TWI   74
C IPREXP IS POWER (I.E. EXPONENT) OF DOWNWEIGHTING USED FOR             TWI   75
C PREFERENTIALITY IN SUBROUTINE POLISH.                                 TWI   76
      CWTMIN=0.01                                                       TWI   77
C CWTMIN IS SMALLEST MULTIPLIER THAN CAN BE USED IN DOWNWEIGHTING.      TWI   78
C NO SPECIES CAN BE DOWNWEIGHTED BY A FACTOR MORE THAN CWTMIN.          TWI   79
      CRCUT=0.2                                                         TWI   80
C CRCUT IS LENGTH OF CRITICAL ZONE USED IN SUBROUTINE POLISH            TWI   81
      CRLONG=0.2                                                        TWI   82
C CRLONG IS PROPORTION OF TOTAL LENGTH OF ORDINATION OCCUPIED BY        TWI   83
C CRITICAL ZONE, FOR PURPOSES OF DEFINING MISCLASSIFICATIONS ETC.       TWI   84
      MZCRIT=8                                                          TWI   85
C MZCRIT IS NUMBER OF SEGMENTS IN CRITICAL ZONE.                        TWI   86
      MZIND=4                                                           TWI   87
C MZIND IS THE NUMBER OF SEGMENTS IN THE ZONE OF INDIFFERENCE           TWI   88
      MZOUT=4                                                           TWI   89
C MZOUT IS THE NUMBER OF SEGMENTS TO BE RECOGNIZED OUTSIDE CRITICAL     TWI   90
C ZONE.  HENCE NUMBER OF ZONES IN DIAGRAM OF DIVISION (AVAILABLE        TWI   91
C AS AN OPTION) IS 2*MZOUT+MZCRIT.                                      TWI   92
      WTHIGH=2.0                                                        TWI   93
C WTHIGH IS WEIGHTING GIVEN TO CHARACTERISTICS OF SPECIES THAT          TWI   94
C CORRESPOND TO HIGHER LEVELS OF FIDELITY - I.E. TO THOSE               TWI   95
C CHARACTERISTICS CORRESPONDING TO FIDELITY LEVELS SPE2,SPE3.           TWI   96
      SPE1=0.8                                                          TWI   97
      SPE2=2.0                                                          TWI   98
      SPE3=6.0                                                          TWI   99
C SPEC1,2,3 ARE THRESHOLDS OF FIDELITY USED IN DEFINING THE             TWI  100
C ATTRIBUTES OF SPECIES FOR THE SPECIES-CLASSIFICATION.                 TWI  101
      CALL QUIKIN(MMAX,NMAX,NDAT,MM,NSPEC,INAME1,                       TWI  102
     1INAME2,JNAME1,JNAME2,IDAT,IADJR,MDAT,TITLE,INFLAG)                TWI  103
      CALL ENTER(NSPEC,NL,LCUT,MMIN,MIND,JMAX,LEVMAX,                   TWI  104
     1IDIAGR,IPUNCH,LWGT,LIND,INDORD)                                   TWI  105
      MZ=MZCRIT+2*MZOUT                                                 TWI  106
      MMZ=MZ+4                                                          TWI  107
      MMS=MIND+4                                                        TWI  108
      DO 50 J=1,NSPEC                                                   TWI  109
   50 JNFLAG(J)=J                                                       TWI  110
```

```
      CALL PSEUDO(MM,NN,NMAX,NL,NDAT,NSPEC,IDAT,LCUT,                    TWI  111
     1JNFLAG,JNAME1,JNAME2,JNAM,INDPOT,IY)                               TWI  112
      IF(MMIN.GT.MM) STOP                                                TWI  113
      IF (LEVMAX.EQ.0) STOP                                              TWI  114
      IF(NSPEC.GT.MMAX) PRINT 1000,MMAX                                  TWI  115
      IF(NSPEC.GT.MMAX) WRITE(4,1000) MMAX                               TWI  116
 1000 FORMAT(//1X,'*** BEWARE ***  NUMBER OF SPECIES EXCEEDS',           TWI  117
     1' MMAX (=',I4,').'/                                                TWI  118
     21X,'NO SPECIES CLASSIFICATION IS POSSIBLE.')                       TWI  119
      DO 60 II=1,MM                                                      TWI  120
   60 RRWT(II)=1.0                                                       TWI  121
      DO 165 JJ=1,NN                                                     TWI  122
      IL=JNAM(JJ)                                                        TWI  123
  165 CCWT(JJ)=FLOAT(LWGT(IL))+1.0E-5                                    TWI  124
      DO 170 JJ=1,NN                                                     TWI  125
      J=IABS(INDPOT(JJ))                                                 TWI  126
      JJJ=JNFLAG(J)                                                      TWI  127
  170 IF(INDORD(JJJ).EQ.0) INDPOT(JJ)=-J                                 TWI  128
      PRINT 1011                                                         TWI  129
 1011 FORMAT('1')                                                        TWI  130
      ISEC=1                                                             TWI  131
      IF(IPUNCH.EQ.1) WRITE(7,1200) (TITLE(IT),IT=1,20)                  TWI  132
 1200 FORMAT(1X,19A4,A3//1X,'SAMPLE CLASSIFICATION')                     TWI  133
      CALL CLASS(MM,NN,NDAT,MIND,MMZ,MMS,IX,ICLASS,IIROW,IADDR,          TWI  134
     1INDPOT,INDORD,IZONE,IY,JJCOL,IDAT,INDSIG,IPICT,X,XX,RTOT,          TWI  135
     2RRWT,ROWWGT,Y,YY,CTOT,CCWT,COLWGT,INAME1,INAME2,JNAME1,            TWI  136
     3JNAME2,JNAM,X3,X4,X5,LIND,INFLAG,IPUNCH)                           TWI  137
      ID=IADDR(MM)                                                       TWI  138
  415 IF(IDAT(ID).EQ.-1) GOTO 420                                        TWI  139
      ID=ID+1                                                            TWI  140
      GOTO 415                                                           TWI  141
  420 NDAT=NDAT/2                                                        TWI  142
      IF (ID.GT.NDAT) GOTO 999                                           TWI  143
      IID=NDAT                                                           TWI  144
      DO 422 J=1,NSPEC                                                   TWI  145
  422 Y(J)=0.0                                                           TWI  146
  425 IDAT(IID)=IDAT(ID)                                                 TWI  147
      IID=IID-1                                                          TWI  148
      ID=ID-1                                                            TWI  149
      IF(ID.EQ.0) GOTO 426                                               TWI  150
      GOTO 425                                                           TWI  151
  426 JJJ=0                                                              TWI  152
      ICMAX=0                                                            TWI  153
      DO 431 II=1,MM                                                     TWI  154
      IC=ICLASS(II)                                                      TWI  155
  427 IF(IC*3.LE.NMAX) GOTO 428                                          TWI  156
      IC=IC/2                                                            TWI  157
      GOTO 427                                                           TWI  158
  428 ICLASS(II)=IC                                                      TWI  159
  431 IF(IC.GT.ICMAX) ICMAX=IC                                           TWI  160
      IF (NSPEC.GT.MMAX) GOTO 999                                        TWI  161
      DO 432 IC=1,ICMAX                                                  TWI  162
  432 TOT(IC)=0.0                                                        TWI  163
      JD=0                                                               TWI  164
      IBIG=10000                                                         TWI  165
```

```
      DO 440 II=1,MM                                                    TWI  166
      IC=ICLASS(II)                                                     TWI  167
  433 IID=IID+1                                                         TWI  168
      JJ=IDAT(IID)                                                      TWI  169
      IF(JJ.EQ.-1) GOTO 437                                             TWI  170
      J=IABS(INDPOT(JJ))                                                TWI  171
      JD=JD+1                                                           TWI  172
      IF(JD.GT.NDAT) GOTO 999                                           TWI  173
      JDAT(JD)=J*IBIG+IC                                                TWI  174
      TOT(IC)=TOT(IC)+1.0                                               TWI  175
      IF(J.EQ.JJJ) GOTO 435                                             TWI  176
      JJJ=J                                                             TWI  177
      Y(J)=Y(J)+CCWT(J)*RRWT(II)                                        TWI  178
      ID=ID+1                                                           TWI  179
      IDAT(ID)=J                                                        TWI  180
      ID=ID+1                                                           TWI  181
      IDAT(ID)=1                                                        TWI  182
      IF(ID.GE.IID) GOTO 999                                            TWI  183
      GOTO 433                                                          TWI  184
  435 IDAT(ID)=IDAT(ID)+1                                               TWI  185
      GOTO 433                                                          TWI  186
  437 ID=ID+1                                                           TWI  187
      IDAT(ID)=JJ                                                       TWI  188
      JJJ=0                                                             TWI  189
  440 CONTINUE                                                          TWI  190
      DO 441 J=1,NSPEC                                                  TWI  191
  441 RRWT(J)=Y(J)+0.00001                                              TWI  192
C WE NOW WORK ON SPECIES CLASSIFICATION                                 TWI  193
      IIBIG=IBIG-1                                                      TWI  194
      DO 442 J=1,NSPEC                                                  TWI  195
      JD=JD+1                                                           TWI  196
      IF(JD.GT.NDAT) GOTO 999                                           TWI  197
  442 JDAT(JD)=J*IBIG+IIBIG                                             TWI  198
      CALL ISORT(JDAT,JD)                                               TWI  199
C MOVE MATRIX UP TO BACK OF ARRAY                                       TWI  200
      JJD=NDAT                                                          TWI  201
  445 JDAT(JJD)=JDAT(JD)                                                TWI  202
      JJD=JJD-1                                                         TWI  203
      JD=JD-1                                                           TWI  204
      IF(JD.EQ.0) GOTO 450                                              TWI  205
      GOTO 445                                                          TWI  206
C WE NOW RECONSTITUTE THE MATRIX AND PROCEED TO ACCUMULATE              TWI  207
C THE TOTALS FOR THE SAMPLE CATEGORIES                                  TWI  208
  450 IC=ICMAX                                                          TWI  209
  455 ICC=IC/2                                                          TWI  210
      TOT(ICC)=TOT(ICC)+TOT(IC)                                         TWI  211
      IC=IC-1                                                           TWI  212
      IF(IC.GT.1) GOTO 455                                              TWI  213
      DO 470 JJJ=1,NSPEC                                                TWI  214
      DO 456 IC=1,ICMAX                                                 TWI  215
  456 TOTJ(IC)=0.0                                                      TWI  216
  458 JJD=JJD+1                                                         TWI  217
      JJ=JDAT(JJD)                                                      TWI  218
      J=JJ/IBIG                                                         TWI  219
      IC=JJ-J*IBIG                                                      TWI  220
```

```
      IF(IC.EQ.IIBIG) GOTO 460                                          TWI  221
      TOTJ(IC)=TOTJ(IC)+1.0                                             TWI  222
      GOTO 458                                                          TWI  223
  460 IC=ICMAX                                                          TWI  224
  462 IIC=IC/2                                                          TWI  225
      TOTJ(IIC)=TOTJ(IIC)+TOTJ(IC)                                      TWI  226
      IC=IC-1                                                           TWI  227
      IF(IC.GT.1) GOTO 462                                              TWI  228
      DO 464 IC=1,ICMAX                                                 TWI  229
      RAT=TOTJ(IC)*(TOT(1)-TOT(IC))/(TOT(IC)*(TOTJ(1)-                  TWI  230
     1TOTJ(IC))+1.0E-7)                                                 TWI  231
      IF (RAT.LT.SPE1) GOTO 464                                         TWI  232
      JD=JD+1                                                           TWI  233
      JDAT(JD)=IC*3-2                                                   TWI  234
      IF (RAT.LT.SPE2) GOTO 464                                         TWI  235
      JD=JD+1                                                           TWI  236
      JDAT(JD)=IC*3-1                                                   TWI  237
      IF (RAT.LT.SPE3) GOTO 464                                         TWI  238
      JD=JD+1                                                           TWI  239
      JDAT(JD)=IC*3                                                     TWI  240
  464 CONTINUE                                                          TWI  241
      JD=JD+1                                                           TWI  242
      IF(JD.GE.JJD) GOTO 999                                            TWI  243
      JDAT(JD)=-1                                                       TWI  244
  470 CONTINUE                                                          TWI  245
C JDAT NOW CONTAINS DESIRED SPECIES INFORMATION.  IT IS NOW A           TWI  246
C MATTER OF FIXING UP WEIGHTS AND CLASSIFYING SPECIES                   TWI  247
      DO 480 IC=1,ICMAX                                                 TWI  248
      WEIGHT=1.0                                                        TWI  249
      IIC=IC                                                            TWI  250
  473 IIC=IIC*2                                                         TWI  251
      IF(IIC.GT.1023) GOTO 476                                          TWI  252
      WEIGHT=WEIGHT*1.414                                               TWI  253
      GOTO 473                                                          TWI  254
  476 WEIGHT=WEIGHT*TOT(IC)                                             TWI  255
      IIC=3*IC                                                          TWI  256
      CCWT(IIC-2)=WEIGHT                                                TWI  257
      INDPOT(IIC-2)=IIC-2                                               TWI  258
      CCWT(IIC-1)=WEIGHT*WTHIGH                                         TWI  259
      INDPOT(IIC-1)=IIC-1                                               TWI  260
      CCWT(IIC)=WEIGHT*WTHIGH                                           TWI  261
      INDPOT(IIC)=IIC                                                   TWI  262
  480 CONTINUE                                                          TWI  263
      PRINT 1011                                                        TWI  264
      ISEC=2                                                            TWI  265
      IF(IPUNCH.EQ.1) WRITE(7,1300)                                     TWI  266
 1300 FORMAT(//1X,'SPECIES CLASSIFICATION')                             TWI  267
      CALL CLASS(NSPEC,3*ICMAX,NDAT,0,MMZ,MMS,                          TWI  268
     1IX,JNAM,IIROW,IADDR,INDPOT,INDORD,IZONE,IY,JJCOL,JDAT,            TWI  269
     2INDSIG,IPICT,X,XX,RTOT,RRWT,ROWWGT,Y,YY,CTOT,CCWT,COLWGT,         TWI  270
     3JNAME1,JNAME2,INAME1,INAME2,INAME1,X3,X4,X5,LIND,JNFLAG,IPUNCH)   TWI  271
      CALL CLORD(MM,LEVMAX,ICLASS,IX)                                   TWI  272
      CALL CLORD(NSPEC,LEVMAX,JNAM,IY)                                  TWI  273
C WE HAVE NOW ESSENTIALLY GOT SAMPLES AND SPECIES ORDERED, BUT          TWI  274
C THE SPECIES ORDER MAY NEED TO BE REVERSED TO GET THINGS ON THE        TWI  275
```

```
C POSITIVE DIAGONAL.  ALSO WE ACCUMULATE SPECIES TOTALS IN INDPOT.      TWI  276
      DO 500 II=1,MM                                                   TWI  277
      I=IX(II)                                                         TWI  278
      IX1(II)=INFLAG(I)                                                TWI  279
      IX2(II)=INAME1(I)                                                TWI  280
      IX3(II)=INAME2(I)                                                TWI  281
  500 X(I)=FLOAT(II)                                                   TWI  282
      DO 510 JJ=1,NSPEC                                                TWI  283
      INDPOT(JJ)=0                                                     TWI  284
      J=IY(JJ)                                                         TWI  285
  510 Y(J)=FLOAT(JJ)                                                   TWI  286
      TOT1=0.0                                                         TWI  287
      TOTX=0.0                                                         TWI  288
      TOTY=0.0                                                         TWI  289
      TOTXY=0.0                                                        TWI  290
      ID=0                                                             TWI  291
      DO 520 II=1,MM                                                   TWI  292
      AX=X(II)                                                         TWI  293
  512 ID=ID+1                                                          TWI  294
      J=IDAT(ID)                                                       TWI  295
      IF(J.EQ.-1) GOTO 520                                             TWI  296
      AY=Y(J)                                                          TWI  297
      ID=ID+1                                                          TWI  298
      JQ=IDAT(ID)                                                      TWI  299
      INDPOT(J)=INDPOT(J)+JQ                                           TWI  300
      Q=FLOAT(JQ)                                                      TWI  301
      TOT1=TOT1+Q                                                      TWI  302
      TOTX=TOTX+Q*AX                                                   TWI  303
      TOTY=TOTY+Q*AY                                                   TWI  304
      TOTXY=TOTXY+Q*AX*AY                                              TWI  305
      GOTO 512                                                         TWI  306
  520 CONTINUE                                                         TWI  307
      TOTXY=TOTXY-TOTX*TOTY/TOT1                                       TWI  308
      IF(TOTXY.GT.0.0) GOTO 537                                        TWI  309
C IN THIS CASE TOTXY IS NEGATIVE, I.E. IX AND IY ARE ORDERS            TWI  310
C THAT ARE NEGATIVELY CORRELATED, AND IY MUST BE REVERSED              TWI  311
      DO 530 J=1,NSPEC                                                 TWI  312
  530 JJCOL(J)=IY(NSPEC-J+1)                                           TWI  313
      DO 535 J=1,NSPEC                                                 TWI  314
  535 IY(J)=JJCOL(J)                                                   TWI  315
  537 JMIN=1                                                           TWI  316
      DO 540 J=1,NSPEC                                                 TWI  317
      JJ=IY(J)                                                         TWI  318
      IF(INDPOT(JJ).EQ.0) JMIN=JMIN+1                                  TWI  319
      IY0(J)=JNFLAG(JJ)                                                TWI  320
      IY1(J)=JNAME1(JJ)                                                TWI  321
  540 IY2(J)=JNAME2(JJ)                                                TWI  322
C WE WANT TO PRINT OUT ONLY JMAX COMMONEST SPECIES                     TWI  323
      PRINT 1500                                                       TWI  324
 1500 FORMAT(////1X,'ORDER OF SPECIES INCLUDING RARER ONES')           TWI  325
      PRINT 1501,(IY0(J),IY1(J),IY2(J),J=JMIN,NSPEC)                   TWI  326
 1501 FORMAT((1X,7(I4,1X,A4,1X,A4,'|'),I4,1X,A4,1X,A4))                TWI  327
      PRINT 1502                                                       TWI  328
 1502 FORMAT(////1X,'ORDER OF SAMPLES')                                TWI  329
      PRINT 1503,(IX1(II),IX2(II),IX3(II),II=1,MM)                     TWI  330
```

```
 1503 FORMAT((1X,7(I4,1X,2A4,' |'),I4,1X,2A4))                          TWI  331
      IF(JMAX.EQ.0) STOP                                                TWI  332
      IBIG=5000                                                         TWI  333
      DO 542 J=1,NSPEC                                                  TWI  334
      JJCOL(J)=IBIG*NSPEC                                               TWI  335
      Y(J)=1.0                                                          TWI  336
  542 INDPOT(J)=-IBIG*INDPOT(J)-IBIG+J                                  TWI  337
      CALL ISORT(INDPOT,NSPEC)                                          TWI  338
      IF(JMAX.GT.NSPEC-JMIN+1) JMAX=NSPEC-JMIN+1                        TWI  339
      DO 546 JJ=1,JMAX                                                  TWI  340
      J=-INDPOT(JJ)                                                     TWI  341
      J=(J/IBIG)*IBIG-J+IBIG                                            TWI  342
  546 JJCOL(J)=0                                                        TWI  343
      DO 547 JJ=1,NSPEC                                                 TWI  344
      J=IY(JJ)                                                          TWI  345
  547 IY(JJ)=JJ*IBIG+J+JJCOL(J)                                         TWI  346
      CALL ISORT(IY,NSPEC)                                              TWI  347
      DO 548 JJ=1,NSPEC                                                 TWI  348
      J=IY(JJ)                                                          TWI  349
  548 IY(JJ)=J-(J/IBIG)*IBIG                                            TWI  350
  550 CALL JORDER(MM,NSPEC,NDAT,IDAT,                                   TWI  351
     1IADDR,IY,JJCOL,INDPOT)                                            TWI  352
      ITIMES=MM/NUMTAB                                                  TWI  353
      IF(MM.NE.ITIMES*NUMTAB)ITIMES=ITIMES+1                            TWI  354
      DO 600 IT=1,ITIMES                                                TWI  355
      IMIN=(IT-1)*NUMTAB+1                                              TWI  356
      IMAX=IT*NUMTAB                                                    TWI  357
      IF(IMAX.GT.MM) IMAX=MM                                            TWI  358
  600 CALL TABOUT(MM,NSPEC,IMIN,IMAX,1,JMAX,NDAT,IDAT,                  TWI  359
     1IADDR,IX,IY,INFLAG,JNFLAG,JNAME1,JNAME2,ICLASS,JNAM)              TWI  360
      STOP                                                              TWI  361
  999 PRINT 1999                                                        TWI  362
      WRITE (4,1999)                                                    TWI  363
 1999 FORMAT (1X,'** STOP **  INSUFFICIENT ARRAY SPACE TO CONTINUE')    TWI  364
      STOP                                                              TWI  365
      END                                                               TWI  366
C                                                                       TWI  367
      SUBROUTINE QUIKIN(MMAX,NMAX,NDAT,MM,N,INAME1,                     TWI  368
     1INAME2,JNAME1,JNAME2,IDAT,IBEGIN,NID,TITLE,INFLAG)                TWI  369
C READS IN DATA MATRIX BY SAMPLES, FOLLOWED BY ROW AND COLUMN NAMES IN  TWI  370
C SPECIFIED FORMAT.  THE SAMPLES MUST BE IN ASCENDING NUMERICAL         TWI  371
C ORDER.  SECOND HALF OF SUBROUTINE CONSISTS OF AN OPTION TO DELETE     TWI  372
C UNWANTED SAMPLES FROM THE ANALYSIS.                                   TWI  373
      INTEGER INAME1(MMAX),INAME2(MMAX),JNAME1(NMAX),                   TWI  374
     1JNAME2(NMAX),IDAT(NDAT)                                           TWI  375
      INTEGER IBEGIN(MMAX),INFLAG(MMAX)                                 TWI  376
      REAL AITEM(100),FMT(15),TITLE(20)                                 TWI  377
      INTEGER ITEM(200)                                                 TWI  378
      COMMON/WORK/ITEM                                                  TWI  379
      PRINT 1002                                                        TWI  380
      WRITE(4,1002)                                                     TWI  381
      READ(5,1000) (TITLE(IT),IT=1,20)                                  TWI  382
 1000 FORMAT(20A4)                                                      TWI  383
      PRINT 2000,(TITLE(IT),IT=1,20)                                    TWI  384
      WRITE(4,2001) (TITLE(IT),IT=1,20)                                 TWI  385
```

```
 2001 FORMAT(//1X,19A4,A3//)                                          TWI  386
 2000 FORMAT(//1X,20A4//)                                             TWI  387
      READ(5,1001) (FMT(IT),IT=1,15),NITEM                           TWI  388
 1001 FORMAT(15A4,8X,I2)                                              TWI  389
 1002 FORMAT(///1X,'READING DATA MATRIX FROM DEVICE 5')               TWI  390
      DO 10 II=1,MMAX                                                 TWI  391
   10 INFLAG(II)=0                                                    TWI  392
      ID=0                                                            TWI  393
      N=0                                                             TWI  394
      III=1                                                           TWI  395
      ITEM(NITEM+1)=0                                                 TWI  396
   40 READ(5,FMT) II,(ITEM(IT),AITEM(IT),IT=1,NITEM)                  TWI  397
      IF(II.EQ.III) GOTO 50                                           TWI  398
      ID=ID+1                                                         TWI  399
      IDAT(ID)=-1                                                     TWI  400
      IF(II.EQ.0) GOTO 100                                            TWI  401
      IBEGIN(II)=ID+1                                                 TWI  402
      IF(II.LT.III) GOTO 999                                          TWI  403
      III=II                                                          TWI  404
   50 IT=0                                                            TWI  405
   55 IT=IT+1                                                         TWI  406
      J=ITEM(IT)                                                      TWI  407
      IF(J) 997,40,56                                                 TWI  408
   56 IF(J.GT.N) N=J                                                  TWI  409
      INFLAG(II)=1                                                    TWI  410
      ID=ID+1                                                         TWI  411
      IDAT(ID)=J                                                      TWI  412
      AIJ=AITEM(IT)                                                   TWI  413
      IF(AIJ.LT.0.0) GOTO 997                                         TWI  414
      ID=ID+1                                                         TWI  415
      IDAT(ID)=IFIX(AIJ*1000.0+0.5)                                   TWI  416
      IF(ID.GT.NDAT-3) GOTO 996                                       TWI  417
      GOTO 55                                                         TWI  418
  100 MM=III                                                          TWI  419
      IF(MM.GT.MMAX) GOTO 995                                         TWI  420
      IF(N.GT.NMAX) GOTO 994                                          TWI  421
      NID=ID                                                          TWI  422
      IBEGIN(1)=1                                                     TWI  423
C WE NOW READ THE SPECIES NAMES FOLLOWED BY THE SAMPLE NAMES          TWI  424
      READ (5,1100) (JNAME1(J),JNAME2(J),J=1,N)                       TWI  425
      READ (5,1100) (INAME1(I),INAME2(I),I=1,MM)                      TWI  426
 1100 FORMAT(20A4)                                                    TWI  427
      WRITE(4,1101) MM,N,NID                                          TWI  428
      PRINT 1101,MM,N,NID                                             TWI  429
 1101 FORMAT('0NUMBER OF SAMPLES',I6/                                 TWI  430
     1'0NUMBER OF SPECIES',I6/                                        TWI  431
     2'0LENGTH OF RAW DATA ARRAY',I6)                                 TWI  432
      IID=120                                                         TWI  433
      IF(NID.LT.IID) IID=NID                                          TWI  434
      PRINT 1102,(IDAT(ID),ID=1,IID)                                  TWI  435
 1102 FORMAT(1X,12I10)                                                TWI  436
      IF(NID.LE.120) GOTO 150                                         TWI  437
      IF(NID.LE.240) GOTO 140                                         TWI  438
      PRINT 1103                                                      TWI  439
 1103 FORMAT(1X,6X,28('....'),'..')                                   TWI  440
```

```
 140 IID=NID-119                                                        TWI  441
     IF(IID.LT.121) IID=121                                             TWI  442
     PRINT 1102,(IDAT(ID),ID=IID,NID)                                   TWI  443
 150 PRINT 1104                                                         TWI  444
1104 FORMAT('OSPECIES NAMES')                                           TWI  445
     PRINT 1105,(J,JNAME1(J),JNAME2(J),J=1,N)                           TWI  446
1105 FORMAT((1X,7(I4,1X,A4,1X,A4,'|'),I4,1X,A4,1X,A4))                  TWI  447
     PRINT 1106                                                         TWI  448
1106 FORMAT('OSAMPLE NAMES')                                            TWI  449
     PRINT 1107,(II,INAME1(II),INAME2(II),II=1,MM)                      TWI  450
1107 FORMAT((1X,7(I4,1X,2A4,' |'),I4,1X,2A4))                           TWI  451
     WRITE(4,2100)                                                      TWI  452
     PRINT 2100                                                         TWI  453
2100 FORMAT(///1X,'DO YOU WISH TO OMIT SOME SAMPLES?')                  TWI  454
     CALL OMIT(INFLAG,MM)                                               TWI  455
     I=0                                                                TWI  456
     ID=0                                                               TWI  457
     DO 400 II=1,MM                                                     TWI  458
     IF(INFLAG(II).NE.1) GOTO 400                                       TWI  459
     I=I+1                                                              TWI  460
     INAME1(I)=INAME1(II)                                               TWI  461
     INAME2(I)=INAME2(II)                                               TWI  462
     INFLAG(I)=II                                                       TWI  463
     IID=IBEGIN(II)                                                     TWI  464
 350 ID=ID+1                                                            TWI  465
     IDAT(ID)=IDAT(IID)                                                 TWI  466
     IF(IDAT(IID).EQ.-1) GOTO 400                                       TWI  467
     IID=IID+1                                                          TWI  468
     GOTO 350                                                           TWI  469
 400 CONTINUE                                                           TWI  470
     MM=I                                                               TWI  471
     NID=ID                                                             TWI  472
     RETURN                                                             TWI  473
 994 PRINT 1994,N,NMAX                                                  TWI  474
     WRITE(4,1994) N,NMAX                                               TWI  475
1994 FORMAT(1X,'***ABORT*** LARGEST SPECIES NUMBER ',                   TWI  476
    1'IS',I4,'BUT SHOULD NOT EXCEED',I4)                                TWI  477
     STOP                                                               TWI  478
 995 PRINT 1995,M,MMAX                                                  TWI  479
     WRITE(4,1995) M,MMAX                                               TWI  480
1995 FORMAT(1X,'***ABORT*** LARGEST SAMPLE NUMBER ',                    TWI  481
    1'IS',I4,'BUT SHOULD NOT EXCEED',I4)                                TWI  482
     STOP                                                               TWI  483
 996 PRINT 1996,II                                                      TWI  484
     WRITE(4,1996) II                                                   TWI  485
1996 FORMAT(1X,'***ABORT*** NO MORE SPACE FOR DATA ',                   TWI  486
    1'MATRIX, STOPPED AT SAMPLE',I4)                                    TWI  487
     STOP                                                               TWI  488
 997 PRINT 1997,II                                                      TWI  489
     WRITE(4,1997) II                                                   TWI  490
1997 FORMAT(1X,'***ABORT*** NEGATIVE NUMBER FOUND IN SAMPLE',I4)        TWI  491
     STOP                                                               TWI  492
 999 PRINT 1999,III                                                     TWI  493
     WRITE (4,1999) III                                                 TWI  494
1999 FORMAT(1X,'***ABORT*** NON-SEQUENTIAL SAMPLE',                     TWI  495
```

```
     1' NUMBER FOUND AFTER SAMPLE',I4)                                  TWI  496
      STOP                                                              TWI  497
      END                                                               TWI  498
C                                                                       TWI  499
      SUBROUTINE ENTER(NSPEC,NL,LCUT,MMIN,MIND,JMAX,LEVMAX,             TWI  500
     1IDIAGR,IPUNCH,LWGT,LIND,INDPOT)                                   TWI  501
C THIS SUBROUTINE ENTERS PARAMETERS FROM DEVICE 3                       TWI  502
      INTEGER LCUT(9),LWGT(9),LIND(9),INDPOT(NSPEC)                     TWI  503
      REAL CUT(9)                                                       TWI  504
      WRITE(4,2000)                                                     TWI  505
      PRINT 2000                                                        TWI  506
 2000 FORMAT('1','NOW ENTER INPUT PARAMETERS'/1X)                       TWI  507
   10 PRINT 2005                                                        TWI  508
      WRITE(4,2005)                                                     TWI  509
 2005 FORMAT(1X,'ENTER NUMBER (NOT EXCEEDING 9) OF PSEUDOSPCIES CUT ',  TWI  510
     1'LEVELS'/1X,'OR TYPE -1 FOR DEFAULT CUT LEVELS',                  TWI  511
     2', WHICH ARE 0 2 5 10 20.')                                       TWI  512
      READ(3,*) NL                                                      TWI  513
      PRINT 2010,NL                                                     TWI  514
      WRITE(4,2010) NL                                                  TWI  515
 2010 FORMAT(1X,'ANSWER = ',I2)                                         TWI  516
      IF(NL.EQ.-1) GOTO 18                                              TWI  517
      IF(NL.GE.1.AND.NL.LE.9) GOTO 13                                   TWI  518
      WRITE(4,2013)                                                     TWI  519
      PRINT 2013                                                        TWI  520
 2013 FORMAT(1X,'*** BEWARE ***  VALUE LIES OUTSIDE LIMITS',            TWI  521
     1' 1,9  -  TRY AGAIN')                                             TWI  522
      GOTO 10                                                           TWI  523
   13 WRITE(4,2015) NL                                                  TWI  524
      PRINT 2015,NL                                                     TWI  525
 2015 FORMAT('ONOW ENTER',I4,'  CUT LEVELS')                            TWI  526
      READ(3,*) (CUT(IL),IL=1,NL)                                       TWI  527
      IF(NL.EQ.1) GOTO 20                                               TWI  528
      DO 14 IL=2,NL                                                     TWI  529
      IF(CUT(IL-1).GT.CUT(IL)) GOTO 16                                  TWI  530
   14 CONTINUE                                                          TWI  531
      GOTO 20                                                           TWI  532
   16 PRINT 2016                                                        TWI  533
      WRITE(4,2016)                                                     TWI  534
 2016 FORMAT(1X,'*** BEWARE ***   CUT LEVELS OUT OF ORDER - TRY',       TWI  535
     1' AGAIN')                                                         TWI  536
      GOTO 10                                                           TWI  537
   18 NL=5                                                              TWI  538
      CUT(1)=0.0                                                        TWI  539
      CUT(2)=2.0                                                        TWI  540
      CUT(3)=5.0                                                        TWI  541
      CUT(4)=10.0                                                       TWI  542
      CUT(5)=20.0                                                       TWI  543
   20 PRINT 2020,(CUT(IL),IL=1,NL)                                      TWI  544
      WRITE(4,2020) (CUT(IL),IL=1,NL)                                   TWI  545
 2020 FORMAT('0CUT LEVELS'/1X,9F8.2)                                    TWI  546
      DO 30 IL=1,NL                                                     TWI  547
   30 LCUT(IL)=IFIX(1000.0*CUT(IL)+0.5)                                 TWI  548
      WRITE(4,2030)                                                     TWI  549
      PRINT 2030                                                        TWI  550
```

```
2030 FORMAT('0ENTER MINIMUM GROUP SIZE FOR DIVISION')                    TWI  551
     CALL PARA(MMIN,0,10000,5)                                           TWI  552
     WRITE(4,2040)                                                       TWI  553
     PRINT 2040                                                          TWI  554
2040 FORMAT('0ENTER MAXIMUM NUMBER OF INDICATORS PER DIVISION')          TWI  555
     CALL PARA(MIND,0,15,7)                                              TWI  556
     WRITE(4,2050)                                                       TWI  557
     PRINT 2050                                                          TWI  558
2050 FORMAT('0ENTER MAXIMUM NUMBER OF SPECIES IN FINAL TABULATION')      TWI  559
     CALL PARA(JMAX,0,1000,100)                                          TWI  560
     WRITE(4,2060)                                                       TWI  561
     PRINT 2060                                                          TWI  562
2060 FORMAT('0ENTER MAXIMUM LEVEL OF DIVISIONS')                         TWI  563
     CALL PARA(LEVMAX,0,15,6)                                            TWI  564
     WRITE(4,2070)                                                       TWI  565
     PRINT 2070                                                          TWI  566
2070 FORMAT('0TYPE 1 IF DIAGRAMS OF DIVISIONS ARE WANTED.')              TWI  567
     CALL PARA(IDIAGR,0,1,0)                                             TWI  568
     WRITE(4,2075)                                                       TWI  569
     PRINT 2075                                                          TWI  570
2075 FORMAT('0TYPE 1 IF MACHINE-READABLE COPY OF SOLUTION ',             TWI  571
    1'TO BE WRITTEN TO DEVICE 7')                                        TWI  572
     CALL PARA(IPUNCH,0,1,0)                                             TWI  573
     WRITE(4,2080)                                                       TWI  574
     PRINT 2080                                                          TWI  575
2080 FORMAT('0ENTER WEIGHTS FOR LEVELS OF PSEUDOSPECIES.'/               TWI  576
    11X,'FOR EXAMPLE WEIGHTS  1 2 2 2  SIGNIFY THAT PSEUDOSPECIES'/      TWI  577
    21X,'CORRESPONDING TO 3 HIGHER CUT LEVELS ARE TO BE GIVEN TWICE'/    TWI  578
    31X,'THE WEIGHT OF PSEUDOSPECIES AT THE LOWEST LEVEL.')              TWI  579
     CALL VPARA(LWGT,NL,0,1000000)                                       TWI  580
     WRITE(4,2090)                                                       TWI  581
     PRINT 2090                                                          TWI  582
2090 FORMAT('0ENTER INDICATOR POTENTIALS FOR CUT LEVELS.'/               TWI  583
    11X,'FOR EXAMPLE INDICATOR POTENTIALS   1 0 0 1 0   SIGNIFY '/       TWI  584
    21X,'THAT PSEUDOSPECIES AT LEVELS 1 AND 4 CAN BE USED AS'/           TWI  585
    31X,'INDICATORS, BUT THAT THOSE AT OTHER LEVELS CANNOT.'/            TWI  586
    41X,'IN THE DEFAULT CASE, ALL PSEUDOSPECIES ARE AVAILABLE'/          TWI  587
    51X,'AS INDICATORS.')                                                TWI  588
     CALL VPARA(LIND,NL,0,1)                                             TWI  589
     WRITE(4,2100)                                                       TWI  590
     PRINT 2100                                                          TWI  591
2100 FORMAT('0DO YOU WISH TO OMIT SOME SPECIES FROM LIST OF'/            TWI  592
    11X,'POTENTIAL INDICATORS?  SPECIES OMITTED FROM THIS LIST'/         TWI  593
    21X,'ARE USED IN THE CALCULATION, BUT CANNOT APPEAR AS',             TWI  594
    3' INDICATORS.')                                                     TWI  595
     DO 50 J=1,NSPEC                                                     TWI  596
  50 INDPOT(J)=1                                                         TWI  597
     CALL OMIT(INDPOT,NSPEC)                                             TWI  598
     RETURN                                                              TWI  599
     END                                                                 TWI  600
C                                                                        TWI  601
     SUBROUTINE OMIT(INFLAG,MM)                                          TWI  602
C STARTING WITH A VECTOR INFLAG, SETS TO ZERO ANY ITEMS THAT ARE         TWI  603
C TYPED IN FROM TERMINAL.                                                TWI  604
     INTEGER INFLAG(MM)                                                  TWI  605
```

```
      WRITE(4,2100)                                                    TWI  606
      PRINT 2100                                                       TWI  607
 2100 FORMAT(1X,'ENTER NUMBERS (NOT NAMES) OF ITEMS TO BE ',           TWI  608
     1'OMITTED'/1X,'ONE PER CARD, ENDING LIST WITH A -1.'/             TWI  609
     21X,'OTHER NEGATIVE NUMBERS DENOTE SEQUENCES.  FOR EXAMPLE '/     TWI  610
     31X,'A 4 FOLLOWED BY A -8 OMITS ITEMS 4 THROUGH 8.')             TWI  611
      NCNONO=MM+1                                                      TWI  612
  200 READ(3,*) NONO                                                   TWI  613
      IF(NONO.EQ.-1) GOTO 300                                          TWI  614
      IF(NONO.GT.MM) GOTO 260                                          TWI  615
      IF(NONO.LT.1) GOTO 250                                           TWI  616
      PRINT 2101,NONO                                                  TWI  617
      WRITE(4,2101) NONO                                               TWI  618
 2101 FORMAT(1X,'OMIT ITEM',I6)                                        TWI  619
      INFLAG(NONO)=0                                                   TWI  620
      NCNONO=NONO                                                      TWI  621
      GOTO 200                                                         TWI  622
  250 IF(NONO.LT.-MM) GOTO 260                                         TWI  623
      IF(NONO.GT.-NONONO-1) GOTO 260                                   TWI  624
      NCNO=-NONO                                                       TWI  625
      DO 255 II=NONONO,NCNO                                            TWI  626
  255 INFLAG(II)=0                                                     TWI  627
      NONO=-NONO                                                       TWI  628
      WRITE(4,2104) NONONO,NONO                                        TWI  629
      PRINT 2105,NCNO                                                  TWI  630
 2104 FORMAT(1X,'OMIT ITEMS',2I5)                                      TWI  631
 2105 FORMAT('+',9X,'S',5X,I5)                                         TWI  632
      NONONO=MM+1                                                      TWI  633
      GOTO 200                                                         TWI  634
  260 PRINT 2103,NONO                                                  TWI  635
      WRITE (4,2103) NONO                                              TWI  636
 2103 FORMAT(1X,'*** BEWARE ***  INADMISSIBLE NUMBER',I6,' - IGNORED')  TWI  637
      GOTO 200                                                         TWI  638
  300 PRINT 2102,NONO                                                  TWI  639
      WRITE(4,2102) NONO                                               TWI  640
 2102 FORMAT(1X,9X,I6)                                                 TWI  641
      RETURN                                                           TWI  642
      END                                                              TWI  643
C                                                                      TWI  644
      SUBROUTINE PARA(K,MIN,MAX,KK)                                    TWI  645
C READS PARAMETER K, CHECKS BOUNDS MIN AND MAX, AND SETS THE VALUE OF  TWI  646
C K TO THE DEFAULT VALUE KK IF K.EQ.-1.                                TWI  647
   10 WRITE(4,1000) KK,MIN,MAX                                         TWI  648
      PRINT 1000,KK,MIN,MAX                                            TWI  649
 1000 FORMAT(1X,'TYPE -1 FOR DEFAULT VALUE ( =',I3,');  OTHERWISE',    TWI  650
     1' TYPE (INTEGER) VALUE REQUIRED,'/                               TWI  651
     21X,'WHICH MUST NOT LIE OUTSIDE LIMITS',2I6)                      TWI  652
      READ(3,*) K                                                      TWI  653
      PRINT 1001,K                                                     TWI  654
      WRITE(4,1001) K                                                  TWI  655
 1001 FORMAT(1X,'ANSWER=',I5)                                          TWI  656
      IF(K.GE.MIN.AND.K.LE.MAX) RETURN                                 TWI  657
      IF(K.EQ.-1) GOTO 20                                              TWI  658
      PRINT 1002,MIN,MAX                                               TWI  659
      WRITE(4,1002) MIN,MAX                                            TWI  660
```

```
 1002 FORMAT(1X,'*** BEWARE ***   VALUE LIES OUTSIDE ALLOWED LIMITS',    TWI  661
     1216,'   - TRY AGAIN')                                              TWI  662
      GOTO 10                                                           TWI  663
   20 K=KK                                                              TWI  664
      PRINT 1003,K                                                      TWI  665
      WRITE(4,1003) K                                                   TWI  666
 1003 FORMAT(1X,'VALUE SET TO',I5,' BY DEFAULT')                        TWI  667
      RETURN                                                            TWI  668
      END                                                               TWI  669
C                                                                       TWI  670
      SUBROUTINE VPARA(IX,M,MIN,MAX)                                    TWI  671
C READS IN PARAMETERS IX AND CHECKS WHETHER IN RANGE MIN,MAX            TWI  672
      INTEGER IX(M)                                                     TWI  673
   50 PRINT 1000                                                        TWI  674
      WRITE(4,1000)                                                     TWI  675
 1000 FORMAT(1X,'TYPE -1 FOR DEFAULT VALUES (I.E. IF ALL VALUES TO BE', TWI  676
     1' SET TO 1)'/1X,'OR TYPE 0 IF NON-DEFAULT VALUES ARE TO BE',      TWI  677
     2' ENTERED')                                                       TWI  678
      READ (3,*) K                                                      TWI  679
      PRINT 1001,K                                                      TWI  680
      WRITE(4,1001) K                                                   TWI  681
 1001 FORMAT(1X,'ANSWER=',I5)                                           TWI  682
      IF(K.NE.0) GOTO 100                                               TWI  683
      PRINT  1002,M,MIN,MAX                                             TWI  684
      WRITE(4,1002) M,MIN,MAX                                           TWI  685
 1002 FORMAT(1X,'NOW',I5,'  INTEGER VALUES MUST BE SUPPLIED'/           TWI  686
     11X,'NONE OF WHICH MUST LIE OUTSIDE LIMITS',2I9)                   TWI  687
      READ(3,*) (IX(I),I=1,M)                                           TWI  688
      WRITE(4,1003)                                                     TWI  689
      PRINT 1003                                                        TWI  690
 1003 FORMAT(1X,'ANSWER = ')                                            TWI  691
      WRITE(4,1004) (IX(I),I=1,M)                                       TWI  692
      PRINT 1005,(IX(I),I=1,M)                                          TWI  693
 1004 FORMAT(1X,20I4)                                                   TWI  694
 1005 FORMAT(1X,20I6)                                                   TWI  695
      DO 60 I=1,M                                                       TWI  696
      K=IX(I)                                                           TWI  697
      IF(K.LT.MIN.OR.K.GT.MAX) GOTO 150                                 TWI  698
   60 CONTINUE                                                          TWI  699
      RETURN                                                            TWI  700
  100 IF(K.EQ.-1) GOTO 200                                              TWI  701
  150 WRITE(4,1010) K                                                   TWI  702
      PRINT 1010,K                                                      TWI  703
 1010 FORMAT(1X,'*** BEWARE ***  UNACCEPTABLE VALUE',                   TWI  704
     116,2X,' - TRY AGAIN')                                             TWI  705
      GOTO 50                                                           TWI  706
  200 DO 300 I=1,M                                                      TWI  707
  300 IX(I)=1                                                           TWI  708
      RETURN                                                            TWI  709
      END                                                               TWI  710
C                                                                       TWI  711
      SUBROUTINE ISORT(IX,N)                                            TWI  712
      INTEGER IX(N)                                                     TWI  713
C EFFICIENT (HEAP-SORT) IN SITU SORTING OF VECTOR IX(N)                 TWI  714
      DO 10 I=1,N                                                       TWI  715
```

```
      J=I                                                                TWI  716
      JX=IX(J)                                                           TWI  717
    5 IF(J.EQ.1) GOTO 8                                                  TWI  718
      JJ=J/2                                                             TWI  719
      JJX=IX(JJ)                                                         TWI  720
      IF(JJX.GE.JX) GOTO 8                                               TWI  721
      IX(J)=JJX                                                          TWI  722
      J=JJ                                                               TWI  723
      GOTO 5                                                             TWI  724
    8 IX(J)=JX                                                           TWI  725
   10 CONTINUE                                                           TWI  726
      I=N                                                                TWI  727
      GOTO 14                                                            TWI  728
   12 IX(J)=JX                                                           TWI  729
   14 IF(I.EQ.1) RETURN                                                  TWI  730
      JX=IX(I)                                                           TWI  731
      IX(I)=IX(1)                                                        TWI  732
      I=I-1                                                              TWI  733
      J=1                                                                TWI  734
      JJ=2                                                               TWI  735
   15 IF(I-JJ) 12,17,16                                                  TWI  736
   16 JJX=IX(JJ)                                                         TWI  737
      JJJ=JJ+1                                                           TWI  738
      JJJX=IX(JJJ)                                                       TWI  739
      IF(JJX.GE.JJJX) GOTO 18                                            TWI  740
      IF(JX.GE.JJJX) GOTO 12                                             TWI  741
      IX(J)=JJJX                                                         TWI  742
      J=JJJ                                                              TWI  743
      JJ=J*2                                                             TWI  744
      GOTO 15                                                            TWI  745
   17 JJX=IX(JJ)                                                         TWI  746
   18 IF(JX.GE.JJX) GOTO 12                                              TWI  747
      IX(J)=JJX                                                          TWI  748
      J=JJ                                                               TWI  749
      JJ=J*2                                                             TWI  750
      GOTO 15                                                            TWI  751
      END                                                                TWI  752
C                                                                        TWI  753
      SUBROUTINE CLORD(MM,LEVMAX,ICLASS,IX)                              TWI  754
C ORDERS SAMPLES IN ORDER OF CLASSIFICATION IN ICLASS, MAKING            TWI  755
C ALLOWANCES FOR DIFFERING LEVELS OF DIVISION                            TWI  756
      INTEGER ICLASS(MM),IX(MM)                                          TWI  757
      IEND=2**(LEVMAX+2)                                                 TWI  758
      IBIG=10000                                                         TWI  759
      DO 10 II=1,MM                                                      TWI  760
      IC=ICLASS(II)                                                      TWI  761
      IF(IC.EQ.0) GOTO 7                                                 TWI  762
    5 IC=IC*2                                                            TWI  763
      IF(IC.LT.IEND) GOTO 5                                              TWI  764
      IX(II)=IBIG*IC+II                                                  TWI  765
      GOTO 10                                                            TWI  766
    7 IX(II)=II                                                          TWI  767
   10 CONTINUE                                                           TWI  768
      CALL ISORT(IX,MM)                                                  TWI  769
      DO 12 II=1,MM                                                      TWI  770
```

```
      IC=IX(II)                                                          TWI  771
   12 IX(II)=IC-(IC/IBIG)*IBIG                                           TWI  772
      RETURN                                                             TWI  773
      END                                                                TWI  774
C                                                                        TWI  775
      SUBROUTINE JORDER(M,N,NDAT,IDAT,IADDR,                             TWI  776
     1JORD,IY,IIY)                                                       TWI  777
C ORDERS DATA MATRIX WITHIN SAMPLES SO AS TO ACCORD WITH A               TWI  778
C SPECIFIED ORDER JORD.  ADDRESSES OF BEGINNING OF EACH                  TWI  779
C SAMPLE ARE STORED IN IADDR(I).  NOTE THAT QUANTITATIVE                 TWI  780
C DATA IN EXCESS OF 10000 ARE DISALLOWED.                                TWI  781
      INTEGER IDAT(NDAT),IADDR(M),JORD(N),IY(N),IIY(N)                   TWI  782
      IBIG=10000                                                         TWI  783
      DO 10 J=1,N                                                        TWI  784
      JJ=JORD(J)                                                         TWI  785
   10 IY(JJ)=J                                                           TWI  786
      ID=1                                                               TWI  787
      DO 50 I=1,M                                                        TWI  788
      IJ=0                                                               TWI  789
      IADDR(I)=ID                                                        TWI  790
   24 J=IDAT(ID)                                                         TWI  791
      ID=ID+1                                                            TWI  792
      IF(J.EQ.-1) GOTO 26                                                TWI  793
      JJ=IY(J)*IBIG                                                      TWI  794
      JJQ=IDAT(ID)                                                       TWI  795
      ID=ID+1                                                            TWI  796
      JJ=JJ+JJQ                                                          TWI  797
      IJ=IJ+1                                                            TWI  798
      IIY(IJ)=JJ                                                         TWI  799
      GOTO 24                                                            TWI  800
   26 IF(IJ.EQ.0) GOTO 50                                                TWI  801
      CALL ISORT(IIY,IJ)                                                 TWI  802
      ID=IADDR(I)                                                        TWI  803
      DO 40 IIJ=1,IJ                                                     TWI  804
      JJ=IIY(IIJ)                                                        TWI  805
      J=JJ/IBIG                                                          TWI  806
      IDAT(ID)=JORD(J)                                                   TWI  807
      ID=ID+1                                                            TWI  808
      JJQ=JJ-J*IBIG                                                      TWI  809
      IDAT(ID)=JJQ                                                       TWI  810
      ID=ID+1                                                            TWI  811
   40 CONTINUE                                                           TWI  812
      ID=ID+1                                                            TWI  813
   50 CONTINUE                                                           TWI  814
      RETURN                                                             TWI  815
      END                                                                TWI  816
C                                                                        TWI  817
      SUBROUTINE TABOUT(M,N,IMIN,IMAX,JMIN,JMAX,NDAT,                    TWI  818
     1IDAT,IADDR,IORD,JORD,INFLAG,                                       TWI  819
     2JNFLAG,JNAME1,JNAME2,ICLASS,JCLASS)                                TWI  820
C PRINTS OUT ROWS AND COLUMNS OF MATRIX IN ARBITRARY                     TWI  821
C ORDERS IORD,JORD (BUT NOTE THAT ROWS ARE ASSUMED TO BE                 TWI  822
C ALREADY ORDERED INTERNALLY BY JORD).  THE MATRIX IS                    TWI  823
C TRANSPOSED FOR PRINTING OUT, AS IS CONVENTIONAL IN                     TWI  824
C PHYTOSOCIOLOGY.  THE LIMITS IMIN ETC ARE THE LIMITS                    TWI  825
```

```
C OUTSIDE WHICH MATRIX IS NOT PRINTED.                                  TWI  826
      INTEGER IDAT(NDAT),IADDR(M),IORD(M),JORD(N),                      TWI  827
     1ITEM(200),IOUT(10)                                                TWI  828
      INTEGER BLANK,STAR,DASH                                           TWI  829
      INTEGER INFLAG(M),JNFLAG(N),JNAME1(N),JNAME2(N)                   TWI  830
      INTEGER ICLASS(M),JCLASS(N)                                       TWI  831
      COMMON/WORK/ITEM                                                  TWI  832
      DATA BLANK/' '/,STAR/'*'/                                         TWI  833
      DATA DASH/'-'/                                                    TWI  834
      DATA IOUT/'0','1','2','3','4','5','6','7','8','9'/               TWI  835
      DO 10 I=1,200                                                     TWI  836
   10 ITEM(I)=BLANK                                                     TWI  837
      NNAME=BLANK                                                       TWI  838
      PRINT 1002                                                        TWI  839
 1002 FORMAT('1')                                                       TWI  840
      DO 30 IL=1,4                                                      TWI  841
      IDIV=10**(4-IL)                                                   TWI  842
      III=2                                                             TWI  843
      DO 20 I=IMIN,IMAX                                                 TWI  844
      III=III+1                                                         TWI  845
      II=IORD(I)                                                        TWI  846
      IORIG=INFLAG(II)                                                  TWI  847
      IR=IORIG/IDIV                                                     TWI  848
      IF(IR.GE.1) ITEM(III)=IOUT(IR-(IR/10)*10+1)                       TWI  849
   20 CONTINUE                                                          TWI  850
   30 PRINT 1000,(ITEM(II),II=1,III)                                    TWI  851
      PRINT 1001                                                        TWI  852
 1001 FORMAT(1X)                                                        TWI  853
      IF(JMAX.LT.JMIN) RETURN                                           TWI  854
      DO 110 J=JMIN,JMAX                                                TWI  855
      JJ=JORD(J)                                                        TWI  856
      CALL BIN(ITEM,JCLASS(JJ))                                         TWI  857
      DO 90 II=1,6                                                      TWI  858
   90 ITEM(150+II)=ITEM(1+II)                                           TWI  859
      ITEM(1)=JNFLAG(JJ)                                                TWI  860
      ITEM(2)=JNAME1(JJ)                                                TWI  861
      ITEM(3)=JNAME2(JJ)                                                TWI  862
      III=3                                                             TWI  863
      DO 100 I=IMIN,IMAX                                                TWI  864
      II=IORD(I)                                                        TWI  865
      ID=IADDR(II)                                                      TWI  866
   93 JJJ=IDAT(ID)                                                      TWI  867
      IF(JJJ.NE.JJ) GOTO 98                                             TWI  868
      ID=ID+1                                                           TWI  869
      JQ=IDAT(ID)                                                       TWI  870
      ID=ID+1                                                           TWI  871
      IADDR(II)=ID                                                      TWI  872
      III=III+1                                                         TWI  873
      IF(JQ.GE.10) GOTO 96                                              TWI  874
      ITEM(III)=IOUT(JQ+1)                                              TWI  875
      GOTO 100                                                          TWI  876
   96 ITEM(III)=STAR                                                    TWI  877
      GOTO 100                                                          TWI  878
   98 III=III+1                                                         TWI  879
      ITEM(III)=DASH                                                    TWI  880
```

```
  100 CONTINUE                                                          TWI  881
      III=III+1                                                         TWI  882
      ITEM(III)=BLANK                                                   TWI  883
      III=III+1                                                         TWI  884
      ITEM(III)=BLANK                                                   TWI  885
      DO 105 II=1,6                                                     TWI  886
      III=III+1                                                         TWI  887
  105 ITEM(III)=ITEM(150+II)                                            TWI  888
      PRINT 2000,(ITEM(II),II=1,III)                                    TWI  889
 1000 FORMAT(1X,5X,A4,1X,A4,2X,108A1)                                   TWI  890
 2000 FORMAT(1X,I4,1X,A4,1X,A4,2X,108A1)                                TWI  891
  110 CONTINUE                                                          TWI  892
      PRINT 1001                                                        TWI  893
      ITEM(49)=BLANK                                                    TWI  894
      ITEM(50)=BLANK                                                    TWI  895
      DO 220 LEVEL=1,6                                                  TWI  896
      III=50                                                            TWI  897
      DO 210 I=IMIN,IMAX                                                TWI  898
      II=IORD(I)                                                        TWI  899
      IC=ICLASS(II)                                                     TWI  900
      CALL BIN(ITEM,IC)                                                 TWI  901
      III=III+1                                                         TWI  902
  210 ITEM(III)=ITEM(1+LEVEL)                                           TWI  903
      PRINT 1000,(ITEM(II),II=49,III)                                   TWI  904
  220 CONTINUE                                                          TWI  905
      RETURN                                                            TWI  906
      END                                                               TWI  907
C                                                                       TWI  908
      SUBROUTINE PSEUDO(MM,NN,NMAX,NL,NDAT,NSPEC,IDAT,LCUT,             TWI  909
     1JNFLAG,JNAME1,JNAME2,JNAM,INDPOT,IIY)                             TWI  910
C SETS UP PSEUDOSPECIES DATA IN IDAT, AND ALSO VECTOR INDPOT,           TWI  911
C WHICH STORES THE REAL SPECIES NUMBERS CORRESPONDING TO THE            TWI  912
C PSEUDOSPECIES.  PSEUDOSPECIES CUT LEVELS ARE IN LCUT.                 TWI  913
C SPECIES NAMES IN JNAME1,2.  JNAM STORES LEVELS OF PSEUDOSPECIES.      TWI  914
      INTEGER JTOP(10),IDAT(NDAT),LCUT(NL),JNAME1(NMAX),JNAME2(NMAX),   TWI  915
     1JNFLAG(NMAX),JNAM(NMAX),INDPOT(NMAX),IIY(NMAX)                    TWI  916
      IBIG=50000                                                        TWI  917
      ID=0                                                              TWI  918
      IDMAX=0                                                           TWI  919
      DO 20 II=1,MM                                                     TWI  920
   15 ID=ID+1                                                           TWI  921
      IDMAX=IDMAX+1                                                     TWI  922
      J=IDAT(ID)                                                        TWI  923
      IF(J.EQ.-1) GOTO 20                                               TWI  924
      ID=ID+1                                                           TWI  925
      JQ=IDAT(ID)                                                       TWI  926
      JJQ=0                                                             TWI  927
      DO 16 IL=1,NL                                                     TWI  928
      IF(JQ.GE.LCUT(IL))JJQ=JJQ+1                                       TWI  929
   16 CONTINUE                                                          TWI  930
      IDAT(ID)=JJQ                                                      TWI  931
      IDMAX=IDMAX+JJQ-1                                                 TWI  932
      GOTO 15                                                           TWI  933
   20 CONTINUE                                                          TWI  934
      IF(IDMAX.LE.NDAT) GOTO 40                                         TWI  935
```

SEE ERRATA

```
      IDMAX=IDMAX+50                                                    TWI   936
      PRINT 1001,IDMAX                                                  TWI   937
      WRITE(4,1001) IDMAX                                               TWI   938
 1001 FORMAT(1X,'***ERROR***  DATA ARRAY NEEDS',I6,' ELEMENTS')         TWI   939
      STOP                                                              TWI   940
   40 IID=NDAT                                                          TWI   941
   80 IDAT(IID)=IDAT(ID)                                                TWI   942
      ID=ID-1                                                           TWI   943
      IID=IID-1                                                         TWI   944
      IF(ID.GT.0) GOTO 80                                               TWI   945
      DO 90 II=1,MM                                                     TWI   946
      JJJ=0                                                             TWI   947
   84 IID=IID+1                                                         TWI   948
      J=IDAT(IID)                                                       TWI   949
      IF(J.EQ.-1) GOTO 87                                               TWI   950
      IID=IID+1                                                         TWI   951
      JJQ=IDAT(IID)                                                     TWI   952
      IF(JJQ.EQ.0) GOTO 84                                              TWI   953
      DO 86 JQ=1,JJQ                                                    TWI   954
      JJJ=JJJ+1                                                         TWI   955
   86 IIY(JJJ)=J+IBIG*(JQ-1)                                            TWI   956
      GOTO 84                                                           TWI   957
   87 DO 88 JJ=1,JJJ                                                    TWI   958
      ID=ID+1                                                           TWI   959
   88 IDAT(ID)=IIY(JJ)                                                  TWI   960
      ID=ID+1                                                           TWI   961
      IDAT(ID)=-1                                                       TWI   962
      IF(ID.LE.IID) GOTO 90                                             TWI   963
      IDMAX=IDMAX+300                                                   TWI   964
      PRINT 1001,IDMAX                                                  TWI   965
      STOP                                                              TWI   966
   90 CONTINUE                                                          TWI   967
C WE NOW HAVE THE DESIRED INFORMATION IN IDAT.  ALL WE HAVE             TWI   968
C TO DO IS TO CONDENSE THE NUMBERING                                    TWI   969
      IP=0                                                              TWI   970
      DO 140 IL=1,NL                                                    TWI   971
      DO 100 JJJ=1,NSPEC                                                TWI   972
  100 IIY(JJJ)=0                                                        TWI   973
      ID=0                                                              TWI   974
      DO 110 II=1,MM                                                    TWI   975
  103 ID=ID+1                                                           TWI   976
      JJJ=IDAT(ID)                                                      TWI   977
      IF(JJJ.EQ.-1) GOTO 110                                            TWI   978
      IF(JJJ.GE.IBIG) GOTO 103                                          TWI   979
      IF(JJJ.LT.-1) GOTO 103                                            TWI   980
      IIY(JJJ)=IIY(JJJ)+1                                               TWI   981
      GOTO 103                                                          TWI   982
  110 CONTINUE                                                          TWI   983
      DO 120 JJJ=1,NSPEC                                                TWI   984
      IF(IIY(JJJ).EQ.0) GOTO 120                                        TWI   985
      IP=IP+1                                                           TWI   986
      INDPOT(IP)=JJJ                                                    TWI   987
      IIY(JJJ)=IP                                                       TWI   988
  120 CONTINUE                                                          TWI   989
      ID=0                                                              TWI   990
```

```
      DO 130 II=1,MM                                                    TWI  991
  123 ID=ID+1                                                           TWI  992
      JJJ=IDAT(ID)                                                      TWI  993
      IF(JJJ.EQ.-1) GOTO 130                                            TWI  994
      IF(JJJ.GE.IBIG) GOTO 127                                          TWI  995
      IF(JJJ.LT.-1) GOTO 123                                            TWI  996
      JJJ=IIY(JJJ)                                                      TWI  997
  127 ICAT(ID)=JJJ-IBIG                                                 TWI  998
      GOTO 123                                                          TWI  999
  130 CONTINUE                                                          TWI 1000
      JTOP(IL)=IP                                                       TWI 1001
  140 CONTINUE                                                          TWI 1002
      ID=0                                                              TWI 1003
      DO 150 II=1,MM                                                    TWI 1004
  145 ID=ID+1                                                           TWI 1005
      JJ=IDAT(ID)                                                       TWI 1006
      IF(JJ.EQ.-1) GOTO 150                                             TWI 1007
      IDAT(ID)=JJ+IBIG                                                  TWI 1008
      GOTO 145                                                          TWI 1009
  150 CONTINUE                                                          TWI 1010
      PRINT 1002,ID                                                     TWI 1011
      WRITE(4,1002) ID                                                  TWI 1012
 1002 FORMAT(//1X,'LENGTH OF DATA ARRAY AFTER DEFINING PSEUDOSPECIES',  TWI 1013
     1I7)                                                               TWI 1014
      IF(ID*2.GT.NDAT) PRINT 1003,NDAT                                  TWI 1015
      IF(ID*2.GT.NDAT) WRITE(4,1003) NDAT                               TWI 1016
 1003 FORMAT(1X,'*** BEWARE ***  THIS EXCEEDS ONE-HALF LENGTH OF',      TWI 1017
     1' DATA ARRAY (=',I6,')'/                                          TWI 1018
     21X,'NO SPECIES CLASSIFICATION CAN BE PERFORMED')                  TWI 1019
      NN=IP                                                             TWI 1020
      IF(NN.LE.NMAX) GOTO 155                                           TWI 1021
      PRINT 1000,NN                                                     TWI 1022
      WRITE(4,1000) NN                                                  TWI 1023
 1000 FORMAT(1X,'***ERROR*** - NEEDS DIMENSIONING FOR',I6,              TWI 1024
     1'  PSEUDOSPECIES')                                               TWI 1025
      STOP                                                              TWI 1026
  155 NSPEC=JTOP(1)                                                     TWI 1027
      PRINT 1004,NN                                                     TWI 1028
      WRITE(4,1004) NN                                                  TWI 1029
 1004 FORMAT(//1X,'TOTAL NUMBER OF SPECIES AND PSEUDOSPECIES',I7)       TWI 1030
      PRINT 1005,NSPEC                                                  TWI 1031
      WRITE(4,1005) NSPEC                                               TWI 1032
 1005 FORMAT(//1X,'NUMBER OF SPECIES, EXCLUDING PSEUDOSPECIES ',        TWI 1033
     1'AND ONES WITH NO OCCURRENCES',I5)                                TWI 1034
      DO 160 JJ=1,NSPEC                                                 TWI 1035
      J=INDPOT(JJ)                                                      TWI 1036
      IIY(J)=JJ                                                         TWI 1037
      JNFLAG(JJ)=JNFLAG(J)                                              TWI 1038
      JNAME1(JJ)=JNAME1(J)                                              TWI 1039
  160 JNAME2(JJ)=JNAME2(J)                                              TWI 1040
      DO 170 JJ=1,NN                                                    TWI 1041
      J=INDPOT(JJ)                                                      TWI 1042
  170 INDPOT(JJ)=IIY(J)                                                 TWI 1043
      JJJ=2                                                             TWI 1044
      IL=0                                                              TWI 1045
```

```
      DO 180 JJ=1,NN                                                    TWI 1046
      J=IABS(INDPOT(JJ))                                                TWI 1047
      JNFLAG(JJ)=JNFLAG(J)                                              TWI 1048
      JNAME1(JJ)=JNAME1(J)                                              TWI 1049
      JNAME2(JJ)=JNAME2(J)                                              TWI 1050
      IF(J.LE.JJJ) IL=IL+1                                              TWI 1051
      JNAM(JJ)=IL                                                       TWI 1052
  180 JJJ=J                                                             TWI 1053
      RETURN                                                            TWI 1054
      END                                                               TWI 1055
C                                                                       TWI 1056
      SUBROUTINE CLASS(MM,NN,NDAT,MIND,MMZ,MMS,IX,ICLASS,               TWI 1057
     1IIROW,IADDR,INDPOT,INDORD,IZONE,IY,JJCOL,IDAT,                    TWI 1058
     2INDSIG,IPICT,X,XX,RTOT,RRWT,ROWWGT,Y,YY,CTOT,CCWT,                TWI 1059
     3COLWGT,INAME1,INAME2,JNAME1,JNAME2,JNAM,X3,X4,X5,LIND,            TWI 1060
     4INFLAG,IPUNCH)                                                    TWI 1061
C THIS DOES THE CLASSIFICATION.  IF ISEC.EQ.2 IT DOES A SPECIES         TWI 1062
C CLASSIFICATION, PRINTING OUT LESS INFORMATION THAN IN THE CASE WHERE  TWI 1063
C ISEC.EQ.1.                                                            TWI 1064
      INTEGER PLUS,IX(MM),ICLASS(MM),IIROW(MM),IADDR(MM),INFLAG(MM),    TWI 1065
     1INDPOT(NN),INDORD(NN),IZONE(MM),IY(NN),                           TWI 1066
     2JJCOL(NN),IDAT(NDAT),INDSIG(MMS),                                 TWI 1067
     3IPICT(MMZ,MMS),ITEM(200)                                          TWI 1068
      INTEGER LIND(10)                                                  TWI 1069
      INTEGER INAME1(MM),INAME2(MM),JNAME1(NN),                         TWI 1070
     1JNAME2(NN),JNAM(NN)                                               TWI 1071
      REAL X(MM),XX(MM),RTOT(MM),RRWT(MM),ROWWGT(MM),Y(NN),YY(NN),      TWI 1072
     1CTOT(NN),CCWT(NN),COLWGT(NN)                                      TWI 1073
      REAL X3(MM),X4(MM),X5(MM)                                         TWI 1074
      DATA PLUS/'+'/,MINUS/'-'/                                         TWI 1075
      COMMON/WORK/ITEM                                                  TWI 1076
      COMMON/PICT/MZ,MS,IZD,IIZD,ISD,ISHIFT,MZIND,                      TWI 1077
     1MZCRIT,MZOUT,NIND                                                 TWI 1078
      COMMON/LIMS/RARE,FEEBLE,FRQLIM,TOL,RATLIM,REPLIM                  TWI 1079
      COMMON/SECOND/ISEC                                                TWI 1080
      COMMON/SWITCH/IFAIL,IDIAGR,ISTOP,IREWT                            TWI 1081
      COMMON/IARBS/ICWEXP,IEND,MMIN,IPREXP,LEVMAX                       TWI 1082
      COMMON/ARBS/CWTMIN,CRLONG,CRCUT                                   TWI 1083
      IC=1                                                              TWI 1084
      LEVEL=1                                                           TWI 1085
      ISTOP=0                                                           TWI 1086
      IEND=2**(LEVMAX+2)                                                TWI 1087
      ID=1                                                              TWI 1088
      DO 5 II=1,MM                                                      TWI 1089
      IADDR(II)=ID                                                      TWI 1090
    4 JJ=IDAT(ID)                                                       TWI 1091
      ID=ID+1                                                           TWI 1092
      IF(JJ.EQ.-1) GOTO 5                                               TWI 1093
      GOTO 4                                                            TWI 1094
    5 CONTINUE                                                          TWI 1095
      DO 10 II=1,MM                                                     TWI 1096
   10 ICLASS(II)=IC                                                     TWI 1097
      DO 15 II=1,MM                                                     TWI 1098
      ID=IADDR(II)                                                      TWI 1099
      IF(IDAT(ID).EQ.-1) ICLASS(II)=IEND                                TWI 1100
```

```
   15 CONTINUE                                                          TWI 1101
      GOTO 30                                                           TWI 1102
   20 CALL DECODE(M,MM,N,NN,NDAT,IIROW,IADDR,IDAT,JJCOL)                TWI 1103
      CALL UPDATE(M,MM,IC,LEVEL,ICLASS,IIROW,IX)                        TWI 1104
      IF(ISTOP.EQ.1) GOTO 500                                           TWI 1105
   30 CALL RECODE(M,MM,N,NN,NDAT,IC,ICLASS,IIROW,JJCOL,IADDR,IDAT)      TWI 1106
      CALL BIN(ITEM,IC)                                                 TWI 1107
      PRINT 1010                                                        TWI 1108
 1010 FORMAT(1X,30('****'))                                             TWI 1109
      PRINT 1000,IC,M,(ITEM(IT),IT=1,30)                                TWI 1110
      WRITE(4,1000) IC,M,(ITEM(IT),IT=1,30)                             TWI 1111
 1000 FORMAT('0DIVISION',I5,'  (N=',I4,')',                             TWI 1112
     19X,'I.E. GROUP ',30A1)                                            TWI 1113
      IF(M.GE.MMIN) GOTO 40                                             TWI 1114
      PRINT 1001                                                        TWI 1115
 1001 FORMAT(' DIVISION FAILS - THERE ARE TOO FEW ITEMS')               TWI 1116
      IFAIL=1                                                           TWI 1117
      GOTO 20                                                           TWI 1118
   40 CALL WEIGHT(M,MM,N,NN,NDAT,X,Y,IIROW,IADDR,                       TWI 1119
     1RRWT,ROWWGT,JJCOL,CCWT,COLWGT,IDAT)                               TWI 1120
      CALL RA(M,MM,N,NN,NDAT,EIG,X,XX,X3,X4,X5,Y,RTOT,                  TWI 1121
     1CTOT,IIROW,IADDR,RCWWGT,COLWGT,IDAT)                              TWI 1122
      IF (IFAIL.EQ.1) GOTO 20                                           TWI 1123
      CALL XMAXMI(X,AXMAX,AXMIN,M)                                      TWI 1124
      IF (AXMAX.GT.-AXMIN) GOTO 45                                      TWI 1125
      DO 44 I=1,M                                                       TWI 1126
   44 X(I)=-X(I)                                                        TWI 1127
      AX=AXMAX                                                          TWI 1128
      AXMAX=-AXMIN                                                      TWI 1129
      AXMIN=-AX                                                         TWI 1130
   45 CRMIN=AXMIN*CRLONG                                                TWI 1131
      CRMAX=AXMAX*CRLONG                                                TWI 1132
      CRMID=0.5*(CRMIN+CRMAX)                                           TWI 1133
      CRHALF=0.5*(CRMAX-CRMIN)                                          TWI 1134
      IF(IREWT.EQ.0) GOTO 48                                            TWI 1135
      DO 47 IRE=1,IREWT                                                 TWI 1136
      DO 46 I=1,M                                                       TWI 1137
      AX=(X(I)-CRMID)/CRHALF                                            TWI 1138
      IF(AX.GT.1.0) AX=1.0                                              TWI 1139
      IF(AX.LT.-1.0) AX=-1.0                                            TWI 1140
   46 XX(I)=AX                                                          TWI 1141
      CALL POLISH(M,MM,N,NN,NDAT,X,XX,Y,YY,IIROW,JJCOL,RTOT,            TWI 1142
     1ROWWGT,CCWT,COLWGT,IADDR,IDAT)                                    TWI 1143
      CALL XMAXMI(X,AXMAX,AXMIN,M)                                      TWI 1144
      CRMID=0.5*(AXMIN+AXMAX)                                           TWI 1145
      CRHALF=0.5*CRLONG*(AXMAX-AXMIN)                                   TWI 1146
      CRMIN=CRMID-CRHALF                                                TWI 1147
   47 CRMAX=CRMID+CRHALF                                                TWI 1148
   48 IF(IC.EQ.1) GOTO 63                                               TWI 1149
C WE NOW FIND OUT WHICH OF THE DERIVED GROUPS MORE CLOSELY RESEMBLES     TWI 1150
C THE BROTHER OF THE GROUP THAT HAS JUST BEEN DIVIDED                   TWI 1151
C TO DO THIS WE REQUIRE SPECIES TOTALS (IGNORING PSEUDOSPECIES WEIGHTS) TWI 1152
      SMALL=1.0E-7                                                      TWI 1153
      DO 49 JJ=1,NN                                                     TWI 1154
      Y(JJ)=SMALL                                                       TWI 1155
```

```
   49 YY(JJ)=SMALL                                                        TWI 1156
      TOTO=0.0                                                            TWI 1157
      TOT1=0.0                                                            TWI 1158
      DO 52 I=1,M                                                         TWI 1159
      II=IIROW(I)                                                         TWI 1160
      ID=IADDR(II)                                                        TWI 1161
      IF(X(I).GE.CRMID) TOT1=TOT1+RRWT(II)                                TWI 1162
      IF(X(I).LE.CRMID) TOTO=TOTO+RRWT(II)                                TWI 1163
   51 J=IDAT(ID)                                                          TWI 1164
      ID=ID+1                                                             TWI 1165
      IF(J.EQ.-1) GOTO 52                                                 TWI 1166
      JJ=JJCOL(J)                                                         TWI 1167
      JJJ=IABS(INDPOT(JJ))                                                TWI 1168
      IF(X(I).GE.CRMID) YY(JJJ)=YY(JJJ)+RRWT(II)*CCWT(JJ)                 TWI 1169
      IF(X(I).LE.CRMID) Y(JJJ)=Y(JJJ)+RRWT(II)*CCWT(JJ)                   TWI 1170
      GOTO 51                                                             TWI 1171
   52 CONTINUE                                                            TWI 1172
      IIC=IC+1                                                            TWI 1173
      IF(IIC.EQ.(IIC/2)*2)  IIC=IIC-2                                     TWI 1174
      CALL CLOSER(YSCOR1,IIC,MM,N,NN,NDAT,IEND,TOTO,TOT1,                 TWI 1175
     1IDAT,ICLASS,IADDR,INDPOT,JJCOL,RRWT,CCWT,COLWGT,Y,YY)               TWI 1176
      IIIIC=IC/2+1                                                        TWI 1177
      IF(IIIIC.EQ.(IIIIC/2)*2) IIIIC=IIIIC-2                              TWI 1178
      YSCOR2=0.0                                                          TWI 1179
      IF(IC.LE.3) GOTO 54                                                 TWI 1180
      CALL CLOSER(YSCOR2,IIIIC,MM,N,NN,NDAT,IEND,TOTO,TOT1,               TWI 1181
     1IDAT,ICLASS,IADDR,INDPOT,JJCOL,RRWT,CCWT,COLWGT,Y,YY)               TWI 1182
   54 IW=IC-(IC/4)*4                                                      TWI 1183
      W=0.5                                                               TWI 1184
      IF(IW.EQ.1.OR.IW.EQ.2) W=-W                                         TWI 1185
      YSCORE=YSCOR1+W*YSCOR2                                              TWI 1186
      IF(IIC.NE.(IIC/2)*2) GOTO 559                                       TWI 1187
C  IIC IS EVEN, SO WE WANT THE NEGATIVE GROUP TO BE CLOSER TO IT          TWI 1188
      IF (YSCORE.LE.0) GOTO 63                                            TWI 1189
      GOTO 60                                                             TWI 1190
C  IN THIS CASE IIC IS ODD                                                TWI 1191
  559 IF (YSCORE.GE.0) GOTO 63                                            TWI 1192
   60 DO 61 I=1,M                                                         TWI 1193
   61 X(I)=-X(I)                                                          TWI 1194
      AX=AXMAX                                                            TWI 1195
      AXMAX=-AXMIN                                                        TWI 1196
      AXMIN=-AX                                                           TWI 1197
      AX=CRMAX                                                            TWI 1198
      CRMAX=-CRMIN                                                        TWI 1199
      CRMIN=-AX                                                           TWI 1200
      CRMID=-CRMID                                                        TWI 1201
   63 CALL ZONEUP(X,AXMIN,CRMIN,CRMAX,AXMAX,M,MZ,MZOUT,MZCRIT,IZONE)      TWI 1202
      IF (MIND.EQ.0) GOTO 100                                             TWI 1203
      CUT1=CRMID-0.5*(CRMAX-CRMIN)*FLOAT(MZIND)/(FLOAT(MZCRIT)+0.001)     TWI 1204
      CUT2=2.0*CRMID-CUT1                                                 TWI 1205
      CALL INDSCO(M,MM,N,NN,NDAT,X,AXPOS,AXNEG,                           TWI 1206
     1CUT1,CUT2,Y,YY,IIROW,ROWWGT,IADDR,IDAT)                             TWI 1207
      DO 65 J=1,N                                                         TWI 1208
   65 Y(J)=YY(J)/AXPOS-Y(J)/AXNEG                                         TWI 1209
      CALL TOPIND(N,NN,MIND,FEEBLE,Y,INDORD,INDSIG,                       TWI 1210
```

SEE ERRATA

```
     1JJCOL,INDPCT,JNAM,NIND,LIND)                                          TWI 1211
      IF(NIND.EQ.0) GOTO 100                                                TWI 1212
      MIS=10000                                                             TWI 1213
      CALL CODESC(M,MM,N,NN,NDAT,NIND,INDORD,IX,IY,IIROW,IADDR,IDAT)        TWI 1214
C NOW FIND BEST NUMBER OF INDICATORS AND BEST DIVISION POINT                TWI 1215
      DO 70 IIND=1,NIND                                                     TWI 1216
      CALL TABLE(M,MMZ,MMS,MIND,IX,IZONE,INDSIG,IPICT,0,IIND)               TWI 1217
      CALL FIND(MMZ,MMS,IPICT,IZD1,ISD1,MISCL)                              TWI 1218
      IF(MISCL.GE.MIS)GOTO 70                                               TWI 1219
      MIS=MISCL                                                             TWI 1220
      ISD=ISD1                                                              TWI 1221
      IZD=IZD1                                                              TWI 1222
      IIZD=IZD-MZIND                                                        TWI 1223
      IND=IIND                                                              TWI 1224
   70 CONTINUE                                                              TWI 1225
      CALL TABLE(M,MMZ,MMS,MIND,IX,IZONE,INDSIG,IPICT,1,IND)                TWI 1226
      PRINT 1003                                                            TWI 1227
 1003 FORMAT(1X,'INDICATORS, TOGETHER WITH THEIR SIGN')                     TWI 1228
      IT=0                                                                  TWI 1229
      DO 80 IIND=1,IND                                                      TWI 1230
      J=INDORD(IIND)                                                        TWI 1231
      JJ=JJCOL(J)                                                           TWI 1232
      IT=IT+1                                                               TWI 1233
      ITEM(IT)=JNAME1(JJ)                                                   TWI 1234
      IT=IT+1                                                               TWI 1235
      ITEM(IT)=JNAME2(JJ)                                                   TWI 1236
      IT=IT+1                                                               TWI 1237
      ITEM(IT)=JNAM(JJ)                                                     TWI 1238
      IT=IT+1                                                               TWI 1239
      IF(INDSIG(IIND).GT.0) GOTO 75                                         TWI 1240
      ITEM(IT)=MINUS                                                        TWI 1241
      GOTO 80                                                               TWI 1242
   75 ITEM(IT)=PLUS                                                         TWI 1243
   80 CONTINUE                                                              TWI 1244
      PRINT 1004,(ITEM(IIT),IIT=1,IT)                                       TWI 1245
 1004 FORMAT(1X,7(A4,1X,A4,I1,'(',A1,')',3X))                               TWI 1246
      GOTO 120                                                              TWI 1247
  100 DO 110 I=1,M                                                          TWI 1248
  110 IX(I)=0                                                               TWI 1249
      MS=1                                                                  TWI 1250
      IZD=(1+MZ)/2                                                          TWI 1251
      IIZD=IZD                                                              TWI 1252
      ISD=0                                                                 TWI 1253
      IF(MIND.EQ.0) NIND=0                                                  TWI 1254
      IF(MIND.NE.0)PRINT 1006                                               TWI 1255
 1006 FORMAT(1X,'(NO INDICATORS)')                                          TWI 1256
  120 CALL REPORT(M,MM,N,NN,NDAT,MMZ,MMS,IC,X,XX,ROWWGT,                    TWI 1257
     1Y,YY,COLWGT,IX,IZONE,IIROW,IY,JJCOL,IADDR,IDAT,IPICT,                 TWI 1258
     2INAME1,INAME2,JNAME1,JNAME2,JNAM)                                     TWI 1259
      DO 130 I=1,M                                                          TWI 1260
      AX=X(I)                                                               TWI 1261
      IF(AX.GT.3.5) IX(I)=1                                                 TWI 1262
      IF(AX.LT.3.5) IX(I)=0                                                 TWI 1263
  130 CONTINUE                                                              TWI 1264
      GOTO 20                                                               TWI 1265
```

```
  500 PRINT 1010                                                        TWI 1266
      IF(IPUNCH.EQ.0) RETURN                                            TWI 1267
      NUMBER=0                                                          TWI 1268
      DO 510 II=1,MM                                                    TWI 1269
      IF(ICLASS(II).NE.0) NUMBER=NUMBER+1                               TWI 1270
  510 CONTINUE                                                          TWI 1271
      WRITE(7,1020) NUMBER                                              TWI 1272
 1020 FORMAT(1X,'THERE ARE',I6,'  ITEMS')                               TWI 1273
      DO 550 II=1,MM                                                    TWI 1274
      IC=ICLASS(II)                                                     TWI 1275
      IF(IC.EQ.0) GOTO 550                                              TWI 1276
      CALL BIN(ITEM,IC)                                                 TWI 1277
      IF(ISEC.EQ.2) GOTO 530                                            TWI 1278
      WRITE(7,1030) INFLAG(II),INAME1(II),INAME2(II),IC,                TWI 1279
     1(ITEM(IT),IT=1,30)                                                TWI 1280
      GOTO 550                                                          TWI 1281
  530 WRITE(7,1040) INFLAG(II),INAME1(II),INAME2(II),IC,                TWI 1282
     1(ITEM(IT),IT=1,30)                                                TWI 1283
 1030 FORMAT(1X,I4,1X,2A4,I7,3X,30A1)                                   TWI 1284
 1040 FORMAT(1X,I4,1X,A4,1X,A4,I6,3X,30A1)                              TWI 1285
  550 CONTINUE                                                          TWI 1286
      RETURN                                                            TWI 1287
      END                                                               TWI 1288
C                                                                       TWI 1289
      SUBROUTINE CLOSER(YSCORE,IIC,MM,N,NN,NDAT,IEND,TOTO,TOT1,         TWI 1290
     1IDAT,ICLASS,IADDR,INDPOT,JJCOL,RRWT,CCWT,Y,YO,Y1)                 TWI 1291
C OBTAINS A QUANTITY YSCORE, TO DETERMINE WHICH ORDER THE TWO HALVES    TWI 1292
C OF DIVISION SHOULD BE PUT IN.  IF YSCORE IS NEGATIVE, THEN GROUP      TWI 1293
C IIC MORE CLOSELY RESEMBLES THE NEGATIVE SIDE OF DIVISION, AND         TWI 1294
C VICE VERSA IF YSCORE IS POSITIVE.                                     TWI 1295
      INTEGER IDAT(NDAT),IADDR(MM),ICLASS(MM),INDPOT(NN),JJCOL(NN)      TWI 1296
      REAL RRWT(MM),CCWT(NN),Y(NN),YO(NN),Y1(NN)                        TWI 1297
      TOT=0.0                                                           TWI 1298
      DO 50 JJ=1,NN                                                     TWI 1299
   50 Y(JJ)=1.0E-7                                                      TWI 1300
      DO 54 II=1,MM                                                     TWI 1301
      IIIC=ICLASS(II)                                                   TWI 1302
      IF(IIIC.GT.IEND) IIIC=IIIC-IEND                                   TWI 1303
      IF(IIIC.NE.IIC.AND.IIIC/2.NE.IIC.AND.IIIC/4.NE.IIC) GOTO 54       TWI 1304
      ID=IADDR(II)                                                      TWI 1305
      TOT=TOT+RRWT(II)                                                  TWI 1306
   53 JJ=IDAT(ID)                                                       TWI 1307
      ID=ID+1                                                           TWI 1308
      IF(JJ.EQ.-1) GOTO 54                                              TWI 1309
      JJJ=IABS(INDPOT(JJ))                                              TWI 1310
      Y(JJJ)=Y(JJJ)+RRWT(II)*CCWT(JJ)                                   TWI 1311
      GOTO 53                                                           TWI 1312
   54 CONTINUE                                                          TWI 1313
      YNEG=0.0                                                          TWI 1314
      YIND=0.0                                                          TWI 1315
      YPOS=0.0                                                          TWI 1316
      X1NEG=0.0                                                         TWI 1317
      X1IND=0.0                                                         TWI 1318
      X1POS=0.0                                                         TWI 1319
      X0NEG=0.0                                                         TWI 1320
```

```
      XOIND=0.0                                                          TWI 1321
      XOPOS=0.0                                                          TWI 1322
      DO 57 J=1,N                                                        TWI 1323
      JJ=JJCOL(J)                                                        TWI 1324
      JJJ=IABS(INDPOT(JJ))                                               TWI 1325
      IF(JJ.NE.JJJ) GOTO 57                                              TWI 1326
      AY=Y(JJ)/TOT                                                       TWI 1327
      AY0=Y0(JJJ)/TOT0                                                   TWI 1328
      AY1=Y1(JJJ)/TOT1                                                   TWI 1329
      PREF=ABS(AY0-AY1)/(AY0+AY1)                                        TWI 1330
      PREF=PREF/0.3                                                      TWI 1331
      IF(PREF.GT.1.0) PREF=1.0                                           TWI 1332
      PREF=PREF*PREF*PREF*PREF                                           TWI 1333
      IF(AY1.GT.AY0) GOTO 55                                             TWI 1334
      PNEG=PREF                                                          TWI 1335
      PIND=(1.0-PREF)                                                    TWI 1336
      PPOS=0.0                                                           TWI 1337
      GOTO 56                                                            TWI 1338
   55 PNEG=0.0                                                           TWI 1339
      PIND=(1.0-PREF)                                                    TWI 1340
      PPOS=PREF                                                          TWI 1341
   56 YNEG=YNEG+PNEG*AY                                                  TWI 1342
      YIND=YIND+PIND*AY                                                  TWI 1343
      YPOS=YPOS+PPOS*AY                                                  TWI 1344
      X0NEG=X0NEG+PNEG*AY0                                               TWI 1345
      X1NEG=X1NEG+PNEG*AY1                                               TWI 1346
      X0IND=X0IND+PIND*AY0                                               TWI 1347
      X1IND=X1IND+PIND*AY1                                               TWI 1348
      X0POS=X0POS+PPOS*AY0                                               TWI 1349
      X1POS=X1POS+PPOS*AY1                                               TWI 1350
   57 CONTINUE                                                           TWI 1351
      XNEG=X0NEG+X1NEG                                                   TWI 1352
      XIND=X0IND+X1IND                                                   TWI 1353
      XPOS=X0POS+X1POS                                                   TWI 1354
      IF(XPOS.GT.XNEG)GOTO 58                                            TWI 1355
      IF(XIND.LE.(XNEG-XPOS)) GOTO 59                                    TWI 1356
      YIND=YIND*(XNEG-XPOS)/XIND                                         TWI 1357
      GOTO 59                                                            TWI 1358
   58 YIND=-YIND                                                         TWI 1359
      XIND=-XIND                                                         TWI 1360
      IF(-XIND.LE.(XPOS-XNEG)) GOTO 59                                   TWI 1361
      YIND=YIND*(XPOS-XNEG)/(-XIND)                                      TWI 1362
   59 YSCORE=YPOS-YNEG+YIND                                              TWI 1363
      YSCORE=YSCORE*TOT                                                  TWI 1364
      RETURN                                                             TWI 1365
      END                                                                TWI 1366
C                                                                        TWI 1367
      SUBROUTINE DECODE(M,MM,N,NN,NDAT,IIROW,IADDR,IDAT,JJCOL)           TWI 1368
C GOES THROUGH ROWS OF SUBMATRIX, BACKTRANSLATING J TO JJ, I.E.          TWI 1369
C RETURNING TO COLUMN NUMBERING IN FULL MATRIX.                          TWI 1370
      INTEGER IIROW(M),IADDR(MM),IDAT(NDAT),JJCOL(N)                     TWI 1371
      DO 10 I=1,M                                                        TWI 1372
      II=IIROW(I)                                                        TWI 1373
      ID=IADDR(II)                                                       TWI 1374
    5 J=IDAT(ID)                                                         TWI 1375
```

```
      IF(J.EQ.-1) GOTO 10                                              TWI 1376
      IDAT(ID)=JJCOL(J)                                                TWI 1377
      ID=ID+1                                                          TWI 1378
      GOTO 5                                                           TWI 1379
   10 CONTINUE                                                         TWI 1380
      RETURN                                                           TWI 1381
      END                                                              TWI 1382
C                                                                      TWI 1383
      SUBROUTINE UPDATE(M,MM,IC,LEVEL,ICLASS,IIROW,IX)                 TWI 1384
C UPDATES CLASSIFICATION;  IF DIVISION HAS FAILED, ADDS IEND TO        TWI 1385
C CLASS NUMBER IC;  FINDS NEW CLASS IC TO BE DIVIDED.                  TWI 1386
      INTEGER ICLASS(MM),IIROW(M),IX(M)                                TWI 1387
      COMMON/IARBS/ICWEXP,IEND,MMIN,IPREXP,LEVMAX                      TWI 1388
      COMMON/SWITCH/IFAIL,IDIAGR,ISTOP,IREWT                           TWI 1389
      IF(IFAIL.EQ.0) GOTO 20                                           TWI 1390
      DO 10 I=1,M                                                      TWI 1391
      II=IIROW(I)                                                      TWI 1392
   10 ICLASS(II)=ICLASS(II)+IEND                                       TWI 1393
      GOTO 40                                                          TWI 1394
C WE NOW APPLY THE DIVISION HELD IN IX(I) (=0 OR 1), AND UPDATE        TWI 1395
C THE CLASS REGISTER FOR THE ROWS                                      TWI 1396
   20 DO 30 I=1,M                                                      TWI 1397
      II=IIROW(I)                                                      TWI 1398
   30 ICLASS(II)=ICLASS(II)*2+IX(I)                                    TWI 1399
C WE NOW FIND THE NEXT CLASS TO BE DIVIDED AND SET ISTOP=1 IF          TWI 1400
C WE HAVE COME TO THE END OF THE DIVISIONS                             TWI 1401
   40 IC=IEND                                                          TWI 1402
      DO 50 II=1,MM                                                    TWI 1403
      IIC=ICLASS(II)                                                   TWI 1404
      IF(IIC.LT.IC)IC=IIC                                              TWI 1405
   50 CONTINUE                                                         TWI 1406
      IF(IC.GE.IEND) GOTO 60                                           TWI 1407
      IF(2**LEVEL.GT.IC)GOTO 80                                        TWI 1408
      PRINT 1000,LEVEL                                                 TWI 1409
 1000 FORMAT('0   END OF LEVEL',I4,//)                                 TWI 1410
      LEVEL=LEVEL+1                                                    TWI 1411
      IF (LEVEL.LE.LEVMAX) GOTO 80                                     TWI 1412
   60 ISTOP=1                                                          TWI 1413
      PRINT 1001                                                       TWI 1414
 1001 FORMAT('0THIS IS THE END OF THE DIVISIONS REQUESTED')            TWI 1415
      DO 70 II=1,MM                                                    TWI 1416
      IC=ICLASS(II)                                                    TWI 1417
      IF(IC.GE.IEND)ICLASS(II)=IC-IEND                                 TWI 1418
   70 CONTINUE                                                         TWI 1419
   80 RETURN                                                           TWI 1420
      END                                                              TWI 1421
C                                                                      TWI 1422
      SUBROUTINE RECODE(M,MM,N,NN,NDAT,IC,ICLASS,IIROW,                TWI 1423
     1JJCOL,IADDR,IDAT)                                                TWI 1424
C GIVEN CLASS IC TO BE DIVIDED, CALCULATES M,N, THE NUMBERS OF         TWI 1425
C ROWS AND COLUMNS IN SUBMATRIX DEFINED BY IC.  RENUMBERS THESE        TWI 1426
C WITH CONSECUTIVE NUMBERING, SETTING UP TRANSLATION VECTORS           TWI 1427
C IIROW,JJCOL, TO GET BACK TO OLD NUMBERING WHEN NECESSARY.            TWI 1428
      INTEGER ICLASS(MM),IIROW(MM),JJCOL(NN),IADDR(MM),IDAT(NDAT)      TWI 1429
      I=0                                                              TWI 1430
```

```
      DO 10 II=1,MM                                                    TWI 1431
      IF (ICLASS(II).NE.IC) GOTO 10                                    TWI 1432
      I=I+1                                                            TWI 1433
      IIROW(I)=II                                                      TWI 1434
   10 CONTINUE                                                         TWI 1435
      M=I                                                              TWI 1436
      DO 20 JJ=1,NN                                                    TWI 1437
   20 JJCOL(JJ)=0                                                      TWI 1438
C NOW ACCUMULATE COLUMN TOTALS IN JJCOL                                TWI 1439
      DO 30 I=1,M                                                      TWI 1440
      II=IIROW(I)                                                      TWI 1441
      ID=IADDR(II)                                                     TWI 1442
   25 JJ=IDAT(ID)                                                      TWI 1443
      IF(JJ.EQ.-1) GOTO 30                                             TWI 1444
      JJCOL(JJ)=JJCOL(JJ)+1                                            TWI 1445
      ID=ID+1                                                          TWI 1446
      GOTO 25                                                          TWI 1447
   30 CONTINUE                                                         TWI 1448
      J=0                                                              TWI 1449
      DO 40 JJ=1,NN                                                    TWI 1450
      IF(JJCOL(JJ).EQ.0) GOTO 40                                       TWI 1451
      J=J+1                                                            TWI 1452
      JJCOL(JJ)=J                                                      TWI 1453
   40 CONTINUE                                                         TWI 1454
      N=J                                                              TWI 1455
C NOW GO THROUGH THE MATRIX RECODING THE COLUMNS                       TWI 1456
      DO 50 I=1,M                                                      TWI 1457
      II=IIROW(I)                                                      TWI 1458
      ID=IADDR(II)                                                     TWI 1459
   45 JJ=IDAT(ID)                                                      TWI 1460
      IF(JJ.EQ.-1) GOTO 50                                             TWI 1461
      IDAT(ID)=JJCOL(JJ)                                               TWI 1462
      ID=ID+1                                                          TWI 1463
      GOTO 45                                                          TWI 1464
   50 CONTINUE                                                         TWI 1465
C NOW THE REVERSE CODING J TO JJ IS PUT IN JJCOL                       TWI 1466
      J=0                                                              TWI 1467
      DO 60 JJ=1,NN                                                    TWI 1468
      IF(JJCOL(JJ).EQ.0) GOTO 60                                       TWI 1469
      J=J+1                                                            TWI 1470
      JJCOL(J)=JJ                                                      TWI 1471
   60 CONTINUE                                                         TWI 1472
      RETURN                                                           TWI 1473
      END                                                              TWI 1474
C                                                                      TWI 1475
      SUBROUTINE BIN(ITEM,IC)                                          TWI 1476
C CONVERTS CLASS NUMBER IC TO BINARY DIGITS, READY FOR PRINTING OUT    TWI 1477
      INTEGER ITEM(200), BLANK,ZERO,ONE,STAR                           TWI 1478
      DATA BLANK/' '/,ZERO/'0'/,ONE/'1'/,STAR/'*'/                     TWI 1479
      LEVEL=0                                                          TWI 1480
      IIC=1                                                            TWI 1481
   10 IF(IIC.GT.IC) GOTO 20                                            TWI 1482
      IIC=IIC*2                                                        TWI 1483
      LEVEL=LEVEL+1                                                    TWI 1484
      GOTO 10                                                          TWI 1485
```

```
   20 DO 30 IT=1,30                                                    TWI 1486
   30 ITEM(IT)=BLANK                                                   TWI 1487
      IT=LEVEL                                                         TWI 1488
      IF(LEVEL.GT.30) IT=30                                            TWI 1489
      IIC=IC                                                           TWI 1490
   40 IIIC=IIC/2                                                       TWI 1491
      IIC=IIC-2*IIIC                                                   TWI 1492
      IF(IIC.EQ.0)ITEM(IT)=ZERO                                        TWI 1493
      IF(IIC.EQ.1)ITEM(IT)=ONE                                         TWI 1494
      IIC=IIIC                                                         TWI 1495
      IT=IT-1                                                          TWI 1496
      IF(IT.GT.0) GOTO 40                                              TWI 1497
      ITEM(1)=STAR                                                     TWI 1498
      RETURN                                                           TWI 1499
      END                                                              TWI 1500
C                                                                      TWI 1501
      SUBROUTINE WEIGHT(M,MM,N,NN,NDAT,X,Y,IIROW,IADDR,RRWT,ROWWGT,    TWI 1502
     1JJCOL,CCWT,COLWGT,IDAT)                                          TWI 1503
C WEIGHTS COLUMNS ACCORDING TO HOW RARE THEY ARE.  THOSE RARER THAN    TWI 1504
C A PROPORTION FRQLIM ARE DOWNWEIGHTED IN PROPORTION TO THEIR          TWI 1505
C SHORTFALL FROM FRQLIM, RAISED TO THE POWER ICWEXP.  ROW-WEIGHTS      TWI 1506
C FOR SUBMATRIX ARE PUT IN ROWWGT.                                     TWI 1507
      REAL X(M),Y(N),RRWT(MM),ROWWGT(M),CCWT(NN),COLWGT(N)             TWI 1508
      INTEGER IIROW(M),IADDR(MM),JJCOL(NN),IDAT(NDAT)                  TWI 1509
      COMMON/LIMS/RARE,FEEBLE,FRQLIM,TOL,RATLIM,REPLIM                 TWI 1510
      COMMON/IARBS/ICWEXP,IEND,MMIN,IPREXP,LEVMAX                      TWI 1511
      COMMON/ARBS/CWTMIN,CRLONG,CRCUT                                  TWI 1512
      CWT=1.0-CWTMIN                                                   TWI 1513
      IF(FRQLIM.LT.1.0E-10)FRQLIM=1.0E-10                              TWI 1514
      DO 10 J=1,N                                                      TWI 1515
   10 COLWGT(J)=1.0                                                    TWI 1516
      TOT=0.0                                                          TWI 1517
      DO 20 I=1,M                                                      TWI 1518
      II=IIROW(I)                                                      TWI 1519
      RR=RRWT(II)                                                      TWI 1520
      ROWWGT(I)=RR                                                     TWI 1521
      TOT=TOT+RR                                                       TWI 1522
   20 X(I)=1.0                                                         TWI 1523
      CALL XYMULT(M,MM,N,NN,NDAT,X,Y,IIROW,IADDR,ROWWGT,               TWI 1524
     1COLWGT,IDAT)                                                     TWI 1525
      DO 30 J=1,N                                                      TWI 1526
      JJ=JJCOL(J)                                                      TWI 1527
      AY=Y(J)/TOT                                                      TWI 1528
      IF(AY.GE.FRQLIM)AY=FRQLIM                                        TWI 1529
      COLWGT(J)=CCWT(JJ)*((AY/FRQLIM)**ICWEXP*CWT+CWTMIN)              TWI 1530
   30 CONTINUE                                                         TWI 1531
      RETURN                                                           TWI 1532
      END                                                              TWI 1533
C                                                                      TWI 1534
      SUBROUTINE RA(M,MM,N,NN,NDAT,EIG,X,XX,X3,X4,X5,Y,                TWI 1535
     1RTOT,CTOT,IIROW,IADDR,ROWWGT,COLWGT,IDAT)                        TWI 1536
C DOES RECIPROCAL AVERAGING, DERIVING EIGENVECTOR X AND EIGENVALUE     TWI 1537
C EIG.  ROWWGT, COLWGT ARE ROW AND COLUMN WEIGHTS.  XX,X3,X4,X5,Y ARE  TWI 1538
C WORKSPACE.                                                           TWI 1539
      REAL X(M),XX(M),Y(N),RTOT(M),CTOT(N),ROWWGT(M),COLWGT(N)         TWI 1540
```

```
      REAL X3(M),X4(M),X5(M)                                          TWI 1541
      INTEGER IIROW(M),IADDR(MM),IDAT(NDAT)                           TWI 1542
      COMMON/LIMS/RARE,FEEBLE,FRQLIM,TOL,RATLIM,REPLIM                TWI 1543
      COMMON/SWITCH/IFAIL,IDIAGR,ISTOP,IREWT                          TWI 1544
      IFAIL=0                                                         TWI 1545
      DO 10 I=1,M                                                     TWI 1546
   10 XX(I)=1.0                                                       TWI 1547
      DO 20 J=1,N                                                     TWI 1548
   20 Y(J)=1.0                                                        TWI 1549
      CALL XYMULT(M,MM,N,NN,NDAT,XX,CTOT,IIROW,IADDR,ROWWGT,          TWI 1550
     1COLWGT,IDAT)                                                    TWI 1551
      CALL YXMULT(M,MM,N,NN,NDAT,Y,RTOT,IIROW,IADDR,ROWWGT,           TWI 1552
     1COLWGT,IDAT)                                                    TWI 1553
      TOT=0.0                                                         TWI 1554
      DO 30 J=1,N                                                     TWI 1555
   30 TOT=TOT+CTOT(J)                                                 TWI 1556
      DO 40 I=1,M                                                     TWI 1557
   40 X(I)=FLOAT(I)                                                   TWI 1558
      X(1)=1.1                                                        TWI 1559
C WE NOW HAVE TRIAL VECTOR X, DELIBERATELY MADE IRREGULAR TO AVOID    TWI 1560
C NASTY COINCIDENCES                                                  TWI 1561
      TTOL=1.0E-5                                                     TWI 1562
      ICOUNT=0                                                        TWI 1563
      IITIM=0                                                         TWI 1564
   50 A=0.0                                                           TWI 1565
C FIRST CENTER TO ZERO MEAN AND UNIT LENGTH                           TWI 1566
      DO 60 I=1,M                                                     TWI 1567
   60 A=A+X(I)*RTOT(I)                                                TWI 1568
      A=A/TOT                                                         TWI 1569
      EX=0.0                                                          TWI 1570
      DO 70 I=1,M                                                     TWI 1571
      AX=X(I)-A                                                       TWI 1572
      EX=EX+AX*AX*RTOT(I)                                             TWI 1573
   70 X(I)=AX                                                         TWI 1574
      EX=SQRT(EX)                                                     TWI 1575
      DO 80 I=1,M                                                     TWI 1576
   80 X(I)=X(I)/EX                                                    TWI 1577
      A11=0.0                                                         TWI 1578
      A12=0.0                                                         TWI 1579
      A22=0.0                                                         TWI 1580
      A23=0.0                                                         TWI 1581
      A33=0.0                                                         TWI 1582
      A34=0.0                                                         TWI 1583
      A44=0.0                                                         TWI 1584
      B13=0.0                                                         TWI 1585
      B14=0.0                                                         TWI 1586
      B24=0.0                                                         TWI 1587
      CALL XYMULT(M,MM,N,NN,NDAT,X,Y,IIROW,                           TWI 1588
     1IADDR,ROWWGT,COLWGT,IDAT)                                       TWI 1589
      DO 81 J=1,N                                                     TWI 1590
   81 Y(J)=Y(J)/CTOT(J)                                               TWI 1591
      CALL YXMULT(M,MM,N,NN,NDAT,Y,XX,IIROW,                          TWI 1592
     1IADDR,ROWWGT,COLWGT,IDAT)                                       TWI 1593
      A=0.0                                                           TWI 1594
      DO 82 I=1,M                                                     TWI 1595
```

```
      AX=XX(I)                                                          TWI 1596
      XX(I)=AX/RTOT(I)                                                  TWI 1597
      A=A+AX                                                            TWI 1598
   82 A11=A11+AX*X(I)                                                   TWI 1599
      IF(A11.GT.TTOL) GOTO 83                                           TWI 1600
      IFAIL=1                                                           TWI 1601
      RETURN                                                            TWI 1602
   83 A=A/TOT                                                           TWI 1603
      DO 84 I=1,M                                                       TWI 1604
      AX=XX(I)-A-A11*X(I)                                               TWI 1605
      A12=A12+AX*AX*RTOT(I)                                             TWI 1606
   84 XX(I)=AX                                                          TWI 1607
      A12=SQRT(A12)                                                     TWI 1608
      DO 86 I=1,M                                                       TWI 1609
   86 XX(I)=XX(I)/A12                                                   TWI 1610
      IF(A12.LT.TOL) GOTO 200                                           TWI 1611
      IF(ICOUNT.GT.5) GOTO 200                                          TWI 1612
      ICOUNT=ICOUNT+1                                                   TWI 1613
      CALL XYMULT(M,MM,N,NN,NDAT,XX,Y,IIROW,                            TWI 1614
     1IADDR,ROWWGT,COLWGT,IDAT)                                         TWI 1615
      DO 91 J=1,N                                                       TWI 1616
   91 Y(J)=Y(J)/CTOT(J)                                                 TWI 1617
      CALL YXMULT(M,MM,N,NN,NDAT,Y,X3,IIROW,                            TWI 1618
     1IADDR,ROWWGT,COLWGT,IDAT)                                         TWI 1619
      A=0.0                                                             TWI 1620
      DO 92 I=1,M                                                       TWI 1621
      AX=X3(I)                                                          TWI 1622
      X3(I)=AX/RTOT(I)                                                  TWI 1623
      A=A+AX                                                            TWI 1624
      A22=A22+AX*XX(I)                                                  TWI 1625
   92 B13=B13+AX*X(I)                                                   TWI 1626
      IF(A22.LT.TTOL) GOTO 120                                          TWI 1627
      A=A/TOT                                                           TWI 1628
      DO 94 I=1,M                                                       TWI 1629
      AX=X3(I)-A-A22*XX(I)-B13*X(I)                                     TWI 1630
      A23=A23+AX*AX*RTOT(I)                                             TWI 1631
   94 X3(I)=AX                                                          TWI 1632
      A23=SQRT(A23)                                                     TWI 1633
      DO 96 I=1,M                                                       TWI 1634
   96 X3(I)=X3(I)/A23                                                   TWI 1635
      CALL XYMULT(M,MM,N,NN,NDAT,X3,Y,IIROW,                            TWI 1636
     1IADDR,ROWWGT,COLWGT,IDAT)                                         TWI 1637
      DO 101 J=1,N                                                      TWI 1638
  101 Y(J)=Y(J)/CTOT(J)                                                 TWI 1639
      CALL YXMULT(M,MM,N,NN,NDAT,Y,X4,IIROW,                            TWI 1640
     1IADDR,ROWWGT,COLWGT,IDAT)                                         TWI 1641
      A=0.0                                                             TWI 1642
      DO 102 I=1,M                                                      TWI 1643
      AX=X4(I)                                                          TWI 1644
      X4(I)=AX/RTOT(I)                                                  TWI 1645
      A=A+AX                                                            TWI 1646
      A33=A33+AX*X3(I)                                                  TWI 1647
      B14=B14+AX*X(I)                                                   TWI 1648
  102 B24=B24+AX*XX(I)                                                  TWI 1649
      IF(A33.LT.TTOL) GOTO 120                                          TWI 1650
```

```
      A=A/TOT                                                          TWI 1651
      DO 104 I=1,M                                                     TWI 1652
      AX=X4(I)-(A+A33*X3(I)+B14*X(I)+B24*XX(I))                        TWI 1653
      A34=A34+AX*AX*RTOT(I)                                            TWI 1654
  104 X4(I)=AX                                                         TWI 1655
      A34=SQRT(A34)                                                    TWI 1656
      DO 106 I=1,M                                                     TWI 1657
  106 X4(I)=X4(I)/A34                                                  TWI 1658
      CALL XYMULT(M,MM,N,NN,NDAT,X4,Y,IIROW,                           TWI 1659
     1IADDR,ROWWGT,COLWGT,IDAT)                                        TWI 1660
      DO 111 J=1,N                                                     TWI 1661
  111 Y(J)=Y(J)/CTOT(J)                                                TWI 1662
      CALL YXMULT(M,MM,N,NN,NDAT,Y,X5,IIROW,                           TWI 1663
     1IADDR,ROWWGT,COLWGT,IDAT)                                        TWI 1664
      DO 112 I=1,M                                                     TWI 1665
  112 A44=A44+X4(I)*X5(I)                                              TWI 1666
  120 AX1=1.0                                                          TWI 1667
      AX2=0.1                                                          TWI 1668
      AX3=0.01                                                         TWI 1669
      AX4=0.001                                                        TWI 1670
      DO 130 ITIMES=1,100                                              TWI 1671
      AXX1=A11*AX1+A12*AX2                                             TWI 1672
      AXX2=A12*AX1+A22*AX2+A23*AX3                                     TWI 1673
      AXX3=A23*AX2+A33*AX3+A34*AX4                                     TWI 1674
      AXX4=A34*AX3+A44*AX4                                             TWI 1675
      AX1=A11*AXX1+A12*AXX2                                            TWI 1676
      AX2=A12*AXX1+A22*AXX2+A23*AXX3                                   TWI 1677
      AX3=A23*AXX2+A33*AXX3+A34*AXX4                                   TWI 1678
      AX4=A34*AXX3+A44*AXX4                                            TWI 1679
      EX=SQRT(AX1**2+AX2**2+AX3**2+AX4**2)                             TWI 1680
      AX1=AX1/EX                                                       TWI 1681
      AX2=AX2/EX                                                       TWI 1682
      AX3=AX3/EX                                                       TWI 1683
      AX4=AX4/EX                                                       TWI 1684
      IF(ITIMES.NE.(ITIMES/5)*5) GOTO 130                              TWI 1685
      EXX=SQRT(EX)                                                     TWI 1686
      RESI=SQRT((AX1-AXX1/EXX)**2+(AX2-AXX2/EXX)**2+                   TWI 1687
     1(AX3-AXX3/EXX)**2+(AX4-AXX4/EXX)**2)                             TWI 1688
      IF(RESI.LT.TOL*0.05) GOTO 135                                    TWI 1689
  130 CONTINUE                                                         TWI 1690
  135 IITIM=IITIM+ITIMES                                               TWI 1691
      DO 140 I=1,M                                                     TWI 1692
  140 X(I)=AX1*X(I)+AX2*XX(I)+AX3*X3(I)+AX4*X4(I)                      TWI 1693
      GOTO 50                                                          TWI 1694
  200 PRINT 1000,A11,ICOUNT                                            TWI 1695
 1000 FORMAT(1X,'EIGENVALUE',F6.3,'  AT ITERATION',I4)                 TWI 1696
      IF(A12.GT.TOL) PRINT 1010,ICOUNT,A12,TOL                         TWI 1697
 1010 FORMAT(1X,'RA TROUBLE',I4,' ITERATIONS, AND RESIDUAL IS STILL ', TWI 1698
     1F6.3,'  INSTEAD OF ',F6.3,' (THE TOLERANCE)')                     TWI 1699
      EIG=A11                                                          TWI 1700
      RETURN                                                           TWI 1701
      END                                                              TWI 1702
C                                                                      TWI 1703
      SUBROUTINE XMAXMI(X,AXMAX,AXMIN,M)                               TWI 1704
C CALCULATES MAXIMUM AND MINIMUM OF VECTOR X(M).                       TWI 1705
```

```
      REAL X(M)                                                          TWI 1706
      AXMAX=-1.0E10                                                      TWI 1707
      AXMIN=-AXMAX                                                       TWI 1708
      DO 10 I=1,M                                                        TWI 1709
      AX=X(I)                                                            TWI 1710
      IF(AX.GT.AXMAX)AXMAX=AX                                            TWI 1711
      IF(AX.LT.AXMIN)AXMIN=AX                                            TWI 1712
   10 CONTINUE                                                           TWI 1713
      RETURN                                                             TWI 1714
      END                                                                TWI 1715
C                                                                        TWI 1716
      SUBROUTINE POLISH(M,MM,N,NN,NDAT,X,XX,Y,YY,                        TWI 1717
     1IIROW,JJCOL,RTOT,ROWWGT,CCWT,COLWGT,IADDR,IDAT)                    TWI 1718
C SUBROUTINE POLARIZES ORDINATION X SO THAT ORDINATION APPROXIMATES      TWI 1719
C BETTER TO A CLASSIFICATION.  TO DO THIS, DERIVES TWO NEW               TWI 1720
C ORDINATIONS, TAKES THEIR SUM, AND PUTS BACK IN X.                      TWI 1721
C ORDINATION 1 IS ADDITIVE, DERIVED BY ADDITNG PREFERENCE SCORES IN      TWI 1722
C A SUMMATION IN WHICH ONLY GOOD (I.E. REASONABLY FREQUENT AND           TWI 1723
C REASONABLY PREFERENTIAL) SPECIES ARE GIVEN APPRECIABLE WEIGHT.         TWI 1724
C ORDINATION 2 TAKES THE MEAN PREFERENCE SCORE OF THE SPECIES IN         TWI 1725
C EACH STAND, UNWEIGHTED EXCEPT FOR BASIC WEIGHTS CCWT.                  TWI 1726
      INTEGER IIROW(M),JJCOL(N),IADDR(MM),IDAT(NDAT)                     TWI 1727
      REAL X(M),XX(M),Y(N),YY(N),RTOT(M),ROWWGT(M),CCWT(NN),COLWGT(N)    TWI 1728
      COMMON/LIMS/RARE,FEEBLE,FRQLIM,TOL,RATLIM,REPLIM                   TWI 1729
      COMMON/IARBS/ICWEXP,IEND,MMIN,IPREXP,LEVMAX                        TWI 1730
      COMMON/ARBS/CWTMIN,CRLONG,CRCUT                                    TWI 1731
      CALL XMAXMI(X,AXMAX,AXMIN,M)                                       TWI 1732
      CUTMID=(AXMAX+AXMIN)*0.5                                           TWI 1733
      CUTHLF=(AXMAX-AXMIN)*0.5*CRCUT                                     TWI 1734
      CUT1=CUTMID-CUTHLF                                                 TWI 1735
      CUT2=CUTMID+CUTHLF                                                 TWI 1736
      CALL INDSCO(M,MM,N,NN,NDAT,X,AXPOS,AXNEG,CUT1,CUT2,                TWI 1737
     1Y,YY,IIROW,ROWWGT,IADDR,IDAT)                                      TWI 1738
C PUT PREFERENCE SCORE IN YY AND CALCULATE APPROPRIATE COLUMN WEIGHTS     TWI 1739
      PRLIM=(RATLIM-1.0)/(RATLIM+1.0)                                    TWI 1740
      DO 10 J=1,N                                                        TWI 1741
      JJ=JJCOL(J)                                                        TWI 1742
      AY=Y(J)/AXNEG                                                      TWI 1743
      AYY=YY(J)/AXPOS                                                    TWI 1744
      PREF=(AYY-AY)/(AYY+AY)                                             TWI 1745
      FREQ=AYY+AY                                                        TWI 1746
      IF(FREQ.GT.FRQLIM) FREQ=FRQLIM                                     TWI 1747
      IF(PREF.GT.PRLIM)PREF=PRLIM                                        TWI 1748
      IF(PREF.LT.-PRLIM)PREF=-PRLIM                                      TWI 1749
      IF(ABS(PREF).LT.0.001)PREF=0.001                                   TWI 1750
      COLWGT(J)=CCWT(JJ)*(FREQ/FRQLIM)**ICWEXP*(ABS(PREF)/PRLIM)**IPREXPTWI 1751
      Y(J)=PREF/PRLIM                                                    TWI 1752
   10 CONTINUE                                                           TWI 1753
      CALL YXMULT(M,MM,N,NN,NDAT,Y,X,IIROW,IADDR,ROWWGT,COLWGT,IDAT)     TWI 1754
      DO 20 I=1,M                                                        TWI 1755
   20 X(I)=X(I)/ROWWGT(I)                                                TWI 1756
      CALL XMAXMI(X,AXMAX,AXMIN,M)                                       TWI 1757
      IF(ABS(AXMIN).GT.AXMAX)AXMAX=ABS(AXMIN)                            TWI 1758
      DO 30 I=1,M                                                        TWI 1759
   30 X(I)=X(I)/AXMAX                                                    TWI 1760
```

```
C WE NOW ABOLISH COLUMN WEIGHTS                                          TWI 1761
      DO 40 J=1,N                                                        TWI 1762
      YY(J)=1.0                                                          TWI 1763
      JJ=JJCOL(J)                                                        TWI 1764
   40 COLWGT(J)=CCWT(JJ)                                                 TWI 1765
      CALL YXMULT(M,MM,N,NN,NDAT,YY,RTOT,IIROW,IADDR,                    TWI 1766
     1ROWWGT,COLWGT,IDAT)                                                TWI 1767
      CALL YXMULT(M,MM,N,NN,NDAT,Y,XX,IIRCW,IADDR,ROWWGT,COLWGT,IDAT)    TWI 1768
      DO 50 I=1,M                                                        TWI 1769
   50 X(I)=X(I)+XX(I)/RTOT(I)                                            TWI 1770
      RETURN                                                             TWI 1771
      END                                                                TWI 1772
C                                                                        TWI 1773
      SUBROUTINE ZONEUP(X,AXMIN,CRMIN,CRMAX,AXMAX,M,MZ,MZOUT,            TWI 1774
     1MZCRIT,IZONE)                                                      TWI 1775
C INPUT IS ORDINATION IN VECTOR X, WITH MAXIMUM AXMAX, MINIMUM           TWI 1776
C AXMIN.  SETS UP ZONED ORDINATION, STORED IN VECTOR IZONE.              TWI 1777
C CRITICAL ZONE LIES BERTWEEN CRMIN,CRMAX.  TOTAL NUMBER OF              TWI 1778
C ZONES IS MZ.                                                           TWI 1779
      REAL X(M)                                                          TWI 1780
      INTEGER IZONE(M)                                                   TWI 1781
      MZCRIT=MZ-2*MZOUT                                                  TWI 1782
      SMALL=1.0E-10                                                      TWI 1783
      SEG1=(CRMIN-AXMIN)/(FLOAT(MZOUT)+SMALL)+SMALL                      TWI 1784
      SEG2=(CRMAX-CRMIN)/(FLOAT(MZCRIT)+SMALL)+SMALL                     TWI 1785
      SEG3=(AXMAX-CRMAX)/(FLOAT(MZOUT)+SMALL)+SMALL                      TWI 1786
      DO 10 I=1,M                                                        TWI 1787
      AX=X(I)                                                            TWI 1788
      IF(AX.GE.CRMIN) GOTO 3                                             TWI 1789
      IZ=IFIX((AX-AXMIN)/SEG1)+1                                         TWI 1790
      IF(IZ.GT.MZOUT) IZ=MZOUT                                           TWI 1791
      IF(IZ.LT.1)IZ=1                                                    TWI 1792
      GOTO 8                                                             TWI 1793
    3 IF(AX.LE.CRMAX) GOTO 6                                             TWI 1794
      IZ=IFIX((AX-CRMAX)/SEG3)+1                                         TWI 1795
      IF(IZ.LT.1) IZ=1                                                   TWI 1796
      IF(IZ.GT.MZOUT) IZ=MZOUT                                           TWI 1797
      IZ=IZ+MZ-MZOUT                                                     TWI 1798
      GOTO 8                                                             TWI 1799
    6 IZ=IFIX((AX-CRMIN)/SEG2)+1                                         TWI 1800
      IF(IZ.LT.1)IZ=1                                                    TWI 1801
      IF(IZ.GT.MZCRIT)IZ=MZCRIT                                          TWI 1802
      IZ=IZ+MZOUT                                                        TWI 1803
    8 IZONE(I)=IZ                                                        TWI 1804
   10 CCNTINUE                                                           TWI 1805
      RETURN                                                             TWI 1806
      END                                                                TWI 1807
C                                                                        TWI 1808
      SUBROUTINE TOPIND(N,NN,MIND,FEEBLE,Y,INDORD,INDSIG,                TWI 1809
     1JJCOL,INDPOT,JNAM,NIND,LIND)                                       TWI 1810
C INPUT IS INDICATOR VALUES STORED IN Y.  SETS UP NIND BEST              TWI 1811
C INDICATORS IN INDORD, WITH THEIR SIGN IN INDSIG.  IY IS WORKSPACE.     TWI 1812
C NIND IS SET TO MIND UNLESS THERE IS SHORTAGE OF GOOD INDICATORS.       TWI 1813
      REAL Y(N)                                                          TWI 1814
      INTEGER INDORD(N),INDSIG(MIND),JJCOL(N),INDPOT(NN)                 TWI 1815
```

```
      INTEGER JNAM(NN)                                                  TWI 1816
      INTEGER LIND(10)                                                  TWI 1817
      IBIG=10000                                                        TWI 1818
      DO 10 J=1,N                                                       TWI 1819
   10 INDORD(J)=-IBIG*IFIX(ABS(Y(J)*500.0))-IBIG+J                      TWI 1820
      CALL ISORT(INDORD,N)                                              TWI 1821
      DO 20 J=1,N                                                       TWI 1822
      IV=-INDORD(J)                                                     TWI 1823
   20 INDORD(J)=(IV/IBIG)*IBIG-IV+IBIG                                  TWI 1824
C INDORD NOW CONTAINS ORDERING BUT MUST BE DOCTORED TO TRUNCATE OFF     TWI 1825
C POOR AND DISALLOWED INDICATORS                                        TWI 1826
      NIND=0                                                            TWI 1827
      DO 60 IND=1,N                                                     TWI 1828
      IF(NIND.EQ.MIND) GOTO 60                                          TWI 1829
      J=INDORD(IND)                                                     TWI 1830
      JJ=JJCOL(J)                                                       TWI 1831
      JJP=INDPOT(JJ)                                                    TWI 1832
      IF(JJP.LT.1) GOTO 60                                              TWI 1833
      IP=INDPOT(JJP)                                                    TWI 1834
      IF(IP.LT.1) GOTO 60                                               TWI 1835
      IF(ABS(Y(J)).LT.FEEBLE) GOTO 60                                   TWI 1836
      IL=JNAM(JJ)                                                       TWI 1837
      IF(LIND(IL).EQ.0) GOTO 60                                         TWI 1838
      NIND=NIND+1                                                       TWI 1839
      INDORD(NIND)=J                                                    TWI 1840
      INDPOT(JJP)=-INDPOT(JJP)                                          TWI 1841
      INDSIG(NIND)=1                                                    TWI 1842
      IF(Y(J).LT.0) INDSIG(NIND)=-1                                     TWI 1843
   60 CONTINUE                                                          TWI 1844
      IF(NIND.EQ.0) GOTO 80                                             TWI 1845
      DO 70 IND=1,NIND                                                  TWI 1846
      J=INDORD(IND)                                                     TWI 1847
      JJ=JJCOL(J)                                                       TWI 1848
      JJP=IABS(INDPOT(JJ))                                              TWI 1849
      INDPOT(JJP)=JJP                                                   TWI 1850
   70 CONTINUE                                                          TWI 1851
   80 RETURN                                                            TWI 1852
      END                                                               TWI 1853
C                                                                       TWI 1854
      SUBROUTINE CODESC(M,MM,N,NN,NDAT,NIND,INDORD,IX,IY,               TWI 1855
     1IIROW,IADDR,IDAT)                                                 TWI 1856
C STARTING WITH NIND BEST INDICATORS, STORES INDICATOR-COMPOSITION      TWI 1857
C OF SAMPLES IN CODED FORM IN IX.  THE FIRST (BEST) INDICATOR IS GIVEN  TWI 1858
C A CODED SCORE OF 1, THE NEXT A SCORE OF 2, THE NTH A SCORE 2**(N-1).  TWI 1859
C CODE NUMBERS ARE ADDED TO OBTAIN CODED INDICATOR-COMPOSITION.         TWI 1860
      INTEGER INDORD(N),IX(M),IY(N),IIROW(M),IADDR(MM),IDAT(NDAT)       TWI 1861
      DO 10 J=1,N                                                       TWI 1862
   10 IY(J)=0                                                           TWI 1863
      IIY=1                                                             TWI 1864
      DO 20 IND=1,NIND                                                  TWI 1865
      J=INDORD(IND)                                                     TWI 1866
      IY(J)=IIY                                                         TWI 1867
   20 IIY=IIY*2                                                         TWI 1868
      DO 30 I=1,M                                                       TWI 1869
      II=IIROW(I)                                                       TWI 1870
```

```
      IIX=0                                                              TWI 1871
      ID=IADDR(II)                                                       TWI 1872
   25 J=IDAT(ID)                                                         TWI 1873
      IF(J.EQ.-1) GOTO 26                                                TWI 1874
      IIX=IIX+IY(J)                                                      TWI 1875
      ID=ID+1                                                            TWI 1876
      GOTO 25                                                            TWI 1877
   26 IX(I)=IIX                                                          TWI 1878
   30 CONTINUE                                                           TWI 1879
      RETURN                                                             TWI 1880
      END                                                                TWI 1881
C                                                                        TWI 1882
      SUBROUTINE TABLE(M,MMZ,MMS,MIND,IX,IZONE,INDSIG,IPICT,             TWI 1883
     1IINDTRU,IIND)                                                      TWI 1884
C CODED INDICATOR-COMPOSITION IS IN IX.  CALCULATES MATRIX IPICT,        TWI 1885
C RELATING ZONES TO INDICATOR-SCORES.  ONLY IIND TOP INDICATORS ARE      TWI 1886
C USED.  ISHIFT IS ADDED TO INDICATOR-SCORES TO KEEP THEM POSITIVE.      TWI 1887
C IF THE SWITCH INDTRU=1, SUBROUTINE ALSO REPLACES CODED INDICATOR-      TWI 1888
C COMPOSITION BY THE TRUE INDICATOR-SCORES.                              TWI 1889
      INTEGER IX(M),IZONE(M),INDSIG(MIND),IPICT(MMZ,MMS)                 TWI 1890
      COMMON/PICT/MZ,MS,IZD,IIZD,ISD,ISHIFT,MZIND,MZCRIT,MZOUT,NIND      TWI 1891
      MS=IIND+1                                                          TWI 1892
      ISHIFT=1                                                           TWI 1893
      DO 10 IND=1,IIND                                                   TWI 1894
      IF(INDSIG(IND).LT.0) ISHIFT=ISHIFT+1                               TWI 1895
   10 CONTINUE                                                           TWI 1896
      DO 20 IZ=1,MZ                                                      TWI 1897
      DO 20 IS=1,MS                                                      TWI 1898
   20 IPICT(IZ,IS)=0                                                     TWI 1899
      DO 30 I=1,M                                                        TWI 1900
      IZ=IZONE(I)                                                        TWI 1901
      IS=ISHIFT                                                          TWI 1902
      IIX=IX(I)                                                          TWI 1903
      DO 25 IND=1,IIND                                                   TWI 1904
      IIIX=IIX/2                                                         TWI 1905
      IS=IS+INDSIG(IND)*(IIX-IIIX*2)                                     TWI 1906
   25 IIX=IIIX                                                           TWI 1907
      IF(INDTRU.EQ.1)IX(I)=IS                                            TWI 1908
   30 IPICT(IZ,IS)=IPICT(IZ,IS)+1                                        TWI 1909
      RETURN                                                             TWI 1910
      END                                                                TWI 1911
C                                                                        TWI 1912
      SUBROUTINE FIND(MMZ,MMS,IPICT,IZD1,ISD1,MISCL)                     TWI 1913
C INPUT IS MATRIX IPICT, RELATING INDICATOR SCORES TO ZONES.             TWI 1914
C SUBROUTINE FINDS CLASSIFICATION CENTER IZD1,ISD1 THAT MINIMIZES        TWI 1915
C NUMBER OF MISCLASSIFICATIONS MISCL.                                    TWI 1916
      INTEGER IPICT(MMZ,MMS)                                             TWI 1917
      COMMON/PICT/MZ,MS,IZD,IIZD,ISD,ISHIFT,MZIND,MZCRIT,MZOUT,NIND      TWI 1918
      MINZ=MZOUT+MZIND                                                   TWI 1919
      MAXZ=MZOUT+MZCRIT                                                  TWI 1920
C WE FORM THE TOTALS IN IPICT OF THE ELEMENTS THAT LIE BELOW AND TO      TWI 1921
C THE LEFT OF A GIVEN ONE. I.E. THE NEW VALUE OF IPICT(IZ,IS) IS THE     TWI 1922
C SUM OF ALL THE OLD VALUES IF IPICT(IIZ,IIS) FOR WHICH IIZ IS LESS      TWI 1923
C THAN OR EQUAL TO IZ AND IIS IS LESS THAN OR EQUAL TO IS.               TWI 1924
      DO 10 IS=2,MS                                                      TWI 1925
```

```
   10 IPICT(1,IS)=IPICT(1,IS-1)+IPICT(1,IS)                              TWI 1926
      DO 20 IZ=2,MZ                                                      TWI 1927
   20 IPICT(IZ,1)=IPICT(IZ-1,1)+IPICT(IZ,1)                              TWI 1928
      DO 30 IS=2,MS                                                      TWI 1929
      DO 30 IZ=2,MZ                                                      TWI 1930
      IIS=IS-1                                                           TWI 1931
      IIZ=IZ-1                                                           TWI 1932
   30 IPICT(IZ,IS)=IPICT(IIZ,IS)+IPICT(IZ,IIS)-                          TWI 1933
     1IPICT(IIZ,IIS)+IPICT(IZ,IS)                                        TWI 1934
      MISCL=10000                                                        TWI 1935
      DO 40 IZ=MINZ,MAXZ                                                 TWI 1936
      DO 40 IS=1,MS                                                      TWI 1937
      IIZ=IZ-MZIND                                                       TWI 1938
      MISS=IPICT(IIZ,MS)-IPICT(IIZ,IS)+IPICT(MZ,IS)-IPICT(IZ,IS)        TWI 1939
      A=FLOAT(IPICT(IIZ,MS))                                             TWI 1940
      B=FLOAT(IPICT(MZ,MS)-IPICT(IZ,MS))                                 TWI 1941
      C=ABS((A-B)/(A+B))                                                 TWI 1942
      IA=IABS(IS-ISHIFT)                                                 TWI 1943
      IB=IABS(1+MZ-IZ-IIZ)                                               TWI 1944
      IF(MISS-MISCL) 35,36,40                                            TWI 1945
   35 MISCL=MISS                                                         TWI 1946
      IZD1=IZ                                                            TWI 1947
      ISD1=IS                                                            TWI 1948
      IIA=IA                                                             TWI 1949
      IIB=IB                                                             TWI 1950
      CC=C                                                               TWI 1951
      GOTO 40                                                            TWI 1952
   36 IF(C-CC) 35,38,40                                                  TWI 1953
   38 IF(IA.LT.IIA) GOTO 35                                              TWI 1954
      IF(IB.LT.IIB) GOTO 35                                              TWI 1955
   40 CONTINUE                                                           TWI 1956
      RETURN                                                             TWI 1957
      END                                                                TWI 1958
C                                                                        TWI 1959
      SUBROUTINE REPORT(M,MM,N,NN,NDAT,MMZ,MMS,IC,X,XX,ROWWGT,           TWI 1960
     1Y,YY,COLWGT,IX,IZONE,IIROW,IY,JJCOL,IADDR,IDAT,IPICT,              TWI 1961
     2INAME1,INAME2,JNAME1,JNAME2,JNAM)                                  TWI 1962
C REPORTS ON DICHOTOMY.  TRUE INDICATOR SCORES ARE NOW IN IX.            TWI 1963
C MATRIX IPICT CONTAINS DIAGRAM OF DICHOTOMY, RELATING ZONES             TWI 1964
C TO INDICATOR SCORES.                                                   TWI 1965
      REAL X(M),XX(M),ROWWGT(M),Y(N),YY(N),COLWGT(N)                     TWI 1966
      INTEGER ITEM(200)                                                  TWI 1967
      INTEGER INAME1(MM),INAME2(MM),JNAME1(NN),JNAME2(NN),JNAM(NN)       TWI 1968
      INTEGER IX(M),IZONE(M),IIROW(M),IY(N),JJCOL(N),                    TWI 1969
     1IADDR(MM),IDAT(NDAT),IPICT(MMZ,MMS),NCAT(6)                        TWI 1970
      COMMON/PICT/MZ,MS,IZD,IIZD,ISD,ISHIFT,MZIND,MZCRIT,MZOUT,NIND      TWI 1971
      COMMON/LIMS/RARE,FEEBLE,FRQLIM,TOL,RATLIM,REPLIM                   TWI 1972
      COMMON/SWITCH/IFAIL,IDIAGR,ISTOP,IREWT                             TWI 1973
      COMMON/IARBS/ICWEXP,IEND,MMIN,IPREXP,LEVMAX                        TWI 1974
      COMMON/WORK/ITEM                                                   TWI 1975
      COMMON/SECOND/ISEC                                                 TWI 1976
      DO 10 ICAT=1,6                                                     TWI 1977
   10 NCAT(ICAT)=0                                                       TWI 1978
      DO 20 I=1,M                                                        TWI 1979
      IZ=IZONE(I)                                                        TWI 1980
```

```
      IS=IX(I)                                                          TWI 1981
      IF(IZ.GT.IIZD) GOTO 13                                            TWI 1982
      IF(IS.GT.ISD) GOTO 12                                             TWI 1983
      ICAT=1                                                            TWI 1984
      GOTO 18                                                           TWI 1985
   12 ICAT=3                                                            TWI 1986
      GOTO 18                                                           TWI 1987
   13 IF(IZ.GT.IZD) GOTO 16                                             TWI 1988
      IF(IS.GT.ISD) GOTO 15                                             TWI 1989
      ICAT=2                                                            TWI 1990
      GOTO 18                                                           TWI 1991
   15 ICAT=5                                                            TWI 1992
      GOTO 18                                                           TWI 1993
   16 IF(IS.GT.ISD) GOTO 17                                             TWI 1994
      ICAT=6                                                            TWI 1995
      GOTO 18                                                           TWI 1996
   17 ICAT=4                                                            TWI 1997
   18 X(I)=FLOAT(ICAT)                                                  TWI 1998
      NCAT(ICAT)=NCAT(ICAT)+1                                           TWI 1999
   20 CONTINUE                                                          TWI 2000
      NNEG=NCAT(1)+NCAT(2)+NCAT(3)                                      TWI 2001
      NPOS=NCAT(4)+NCAT(5)+NCAT(6)                                      TWI 2002
      CALL INDSCG(M,MM,N,NN,NDAT,X,AXPOS,AXNEG,3.5,3.5,                 TWI 2003
     1Y,YY,IIROW,ROWWGT,IADDR,IDAT)                                     TWI 2004
      PRLIM=(REPLIM-1.0)/(REPLIM+1.0)-0.0001                            TWI 2005
      DO 30 J=1,N                                                       TWI 2006
      AY=Y(J)/AXNEG                                                     TWI 2007
      AYY=YY(J)/AXPOS                                                   TWI 2008
      IIY=0                                                             TWI 2009
      IF(AY.GE.RARE) IIY=1                                              TWI 2010
      IF(AYY.GE.RARE) IIY=1                                             TWI 2011
      PREF=(AYY-AY)/(AYY+AY)                                            TWI 2012
      IF(PREF.GT.PRLIM) GOTO 25                                         TWI 2013
      IF(PREF.LT.-PRLIM) GOTO 23                                        TWI 2014
      IY(J)=IIY                                                         TWI 2015
      GOTO 30                                                           TWI 2016
   23 IY(J)=2*IIY                                                       TWI 2017
      GOTO 30                                                           TWI 2018
   25 IY(J)=3*IIY                                                       TWI 2019
   30 CONTINUE                                                          TWI 2020
C WE NOW HAVE A VARIABLE IY THAT IS 0 FOR RARITIES, 3 FOR POSITIVE      TWI 2021
C PREFERENTIALS, 2 FOR NEGATIVE PREFERENTIALSS, 1 FOR INDIFFERENTS.     TWI 2022
      IF(NIND.EQ.0) GOTO 60                                             TWI 2023
      ISD1=ISD-ISHIFT                                                   TWI 2024
      ISD2=ISD1+1                                                       TWI 2025
      PRINT 1050,ISD1,ISD2                                              TWI 2026
 1050 FORMAT(1X,'MAXIMUM INDICATOR SCORE FOR NEGATIVE GROUP ',I4,       TWI 2027
     1'      MINIMUM INDICATOR SCORE FOR POSITIVE GROUP ',I4)           TWI 2028
      IF(IDIAGR.EQ.1) CALL DIAGR(IPICT,MMZ,MMS)                         TWI 2029
   60 IIC=IC*2                                                          TWI 2030
      CALL BIN(ITEM,IIC)                                                TWI 2031
      PRINT 1002,IIC,NNEG,(ITEM(IT),IT=1,30)                            TWI 2032
 1002 FORMAT('0ITEMS IN NEGATIVE GROUP',I4,'  (N=',                     TWI 2033
     1I5,')',9X,'I.E. GROUP ',30A1)                                     TWI 2034
      CALL XLIST(M,MM,X,IIROW,1,2,3,INAME1,INAME2)                      TWI 2035
```

```
      IF(NIND.EQ.0) GOTO 70                                              TWI 2036
      NOUT=NCAT(2)                                                       TWI 2037
      IF(NOUT.EQ.0) GOTO 65                                              TWI 2038
      PRINT 1003,NOUT                                                    TWI 2039
 1003 FORMAT('0BORDERLINE NEGATIVES    (N=',I5,')')                      TWI 2040
      CALL XLIST(M,MM,X,IIROW,2,2,2,INAME1,INAME2)                       TWI 2041
   65 NCUT=NCAT(3)                                                       TWI 2042
      IF(NOUT.EQ.0) GOTO 70                                              TWI 2043
      PRINT 1004,NOUT                                                    TWI 2044
 1004 FORMAT('0MISCLASSIFIED NEGATIVES (N=',I5,')')                      TWI 2045
      CALL XLIST(M,MM,X,IIROW,3,3,3,INAME1,INAME2)                       TWI 2046
   70 IIC=IC*2+1                                                         TWI 2047
      CALL BIN(ITEM,IIC)                                                 TWI 2048
      PRINT 1005,IIC,NPOS,(ITEM(IT),IT=1,30)                             TWI 2049
 1005 FORMAT('0ITEMS IN POSITIVE GROUP',I4,'  (N=',                      TWI 2050
     1I5,')',9X,'I.E. GROUP ',30A1)                                      TWI 2051
      CALL XLIST(M,MM,X,IIROW,4,5,6,INAME1,INAME2)                       TWI 2052
      IF(NIND.EQ.0) GOTO 80                                              TWI 2053
      NOUT=NCAT(5)                                                       TWI 2054
      IF(NOUT.EQ.0) GOTO 75                                              TWI 2055
      PRINT 1006,NOUT                                                    TWI 2056
 1006 FORMAT('0BORDERLINE POSITIVES    (N=',I5,')')                      TWI 2057
      CALL XLIST(M,MM,X,IIROW,5,5,5,INAME1,INAME2)                       TWI 2058
   75 NOUT=NCAT(6)                                                       TWI 2059
      IF(NOUT.EQ.0) GOTO 80                                              TWI 2060
      PRINT 1007,NOUT                                                    TWI 2061
 1007 FORMAT('0MISCLASSIFIED POSITIVES (N=',I5,')')                      TWI 2062
      CALL XLIST(M,MM,X,IIROW,6,6,6,INAME1,INAME2)                       TWI 2063
C WE NOW LIST OUT THE COMMONER SPECIES ACCORDING AS THEY ARE              TWI 2064
C PREFERENTIAL OR NEUTRAL                                                 TWI 2065
   80 IF(ISEC.EQ.2) GOTO 100                                             TWI 2066
      CALL INDSCO(M,MM,N,NN,NDAT,X,AXPOS,AXNEG,3.5,3.5,                  TWI 2067
     1Y,YY,IIROW,ROWWGT,IADDR,IDAT)                                      TWI 2068
      PRINT 1010                                                         TWI 2069
 1010 FORMAT('0NEGATIVE PREFERENTIALS')                                  TWI 2070
      CALL YLIST(N,NN,Y,YY,IY,JJCOL,2,JNAME1,JNAME2,JNAM)                TWI 2071
      PRINT 1011                                                         TWI 2072
 1011 FORMAT('0POSITIVE PREFERENTIALS')                                  TWI 2073
      CALL YLIST(N,NN,Y,YY,IY,JJCOL,3,JNAME1,JNAME2,JNAM)                TWI 2074
      PRINT 1012                                                         TWI 2075
 1012 FORMAT('0NON-PREFERENTIALS')                                       TWI 2076
      CALL YLIST(N,NN,Y,YY,IY,JJCOL,1,JNAME1,JNAME2,JNAM)                TWI 2077
  100 RETURN                                                             TWI 2078
      END                                                                TWI 2079
C                                                                        TWI 2080
      SUBROUTINE XYMULT(M,MM,N,NN,NDAT,X,Y,IIROW,IADDR,ROWWGT,           TWI 2081
     1COLWGT,IDAT)                                                       TWI 2082
C FORMS MATRIX PRODUCT Y=AX, APPLYING ROW AND COLUMN WEIGHTS             TWI 2083
C ROWWGT,COLWGT.                                                         TWI 2084
      REAL X(M),Y(N),ROWWGT(M),COLWGT(N)                                 TWI 2085
      INTEGER IIROW(M),IADDR(MM),IDAT(NDAT)                              TWI 2086
      DO 10 J=1,N                                                        TWI 2087
   10 Y(J)=0.0                                                           TWI 2088
      DO 20 I=1,M                                                        TWI 2089
      II=IIROW(I)                                                        TWI 2090
```

```
      AX=X(I)*ROWWGT(I)                                                 TWI 2091
      ID=IADDR(II)                                                      TWI 2092
   15 J=IDAT(ID)                                                        TWI 2093
      IF (J.EQ.-1) GOTO 20                                              TWI 2094
      Y(J)=Y(J)+AX                                                      TWI 2095
      ID=ID+1                                                           TWI 2096
      GOTO 15                                                           TWI 2097
   20 CONTINUE                                                          TWI 2098
      DO 30 J=1,N                                                       TWI 2099
   30 Y(J)=Y(J)*COLWGT(J)                                               TWI 2100
      RETURN                                                            TWI 2101
      END                                                               TWI 2102
C                                                                       TWI 2103
      SUBROUTINE YXMULT(M,MM,N,NN,NDAT,Y,X,IIROW,IADDR,                 TWI 2104
     1ROWWGT,COLWGT,IDAT)                                               TWI 2105
C FORMS MATRIX PRODUCT X=AY, WEIGHTED AS SUBROUTINE XYMULT.             TWI 2106
      REAL Y(N),X(M),ROWWGT(M),COLWGT(N)                                TWI 2107
      INTEGER IIROW(M),IADDR(MM),IDAT(NDAT)                             TWI 2108
      DO 10 J=1,N                                                       TWI 2109
   10 Y(J)=Y(J)*COLWGT(J)                                               TWI 2110
      DO 20 I=1,M                                                       TWI 2111
      II=IIROW(I)                                                       TWI 2112
      AX=0.0                                                            TWI 2113
      ID=IADDR(II)                                                      TWI 2114
   15 J=IDAT(ID)                                                        TWI 2115
      IF(J.EQ.-1) GOTO 16                                               TWI 2116
      AX=AX+Y(J)                                                        TWI 2117
      ID=ID+1                                                           TWI 2118
      GOTO 15                                                           TWI 2119
   16 X(I)=AX*ROWWGT(I)                                                 TWI 2120
   20 CONTINUE                                                          TWI 2121
      DO 30 J=1,N                                                       TWI 2122
   30 Y(J)=Y(J)/COLWGT(J)                                               TWI 2123
      RETURN                                                            TWI 2124
      END                                                               TWI 2125
C                                                                       TWI 2126
      SUBROUTINE INDSCO(M,MM,N,NN,NDAT,X,AXPOS,AXNEG,                   TWI 2127
     1CUT1,CUT2,Y,YY,IIROW,ROWWGT,IADDR,IDAT)                           TWI 2128
C INPUT IS ORDINATION IN X.  OUTPUT IS NEGATIVE INDICATOR SCORES IN Y,  TWI 2129
C POSITIVE INDICATOR SCORES IN YY.  AXPOS,AXNEG ARE ROW-WEIGHTED SUMS   TWI 2130
C OF POSITIVE AND NEGATIVE ELEMENTS.  CUT1,CUT2 ARE CUT-POINTS FOR      TWI 2131
C ZONE OF INTERMEDIATENESS.                                             TWI 2132
      INTEGER IIROW(M),IADDR(MM),IDAT(NDAT)                             TWI 2133
      REAL X(M),Y(N),YY(N),ROWWGT(M)                                    TWI 2134
      CUTMID=(CUT1+CUT2)/2.0                                            TWI 2135
      CUTHLF=(CUT2-CUT1)/2.0+1.0E-10                                    TWI 2136
      AXPOS=0.0                                                         TWI 2137
      AXNEG=0.0                                                         TWI 2138
      DO 10 J=1,N                                                       TWI 2139
      Y(J)=0.0                                                          TWI 2140
   10 YY(J)=0.0                                                         TWI 2141
      DO 30 I=1,M                                                       TWI 2142
      II=IIROW(I)                                                       TWI 2143
      ID=IADDR(II)                                                      TWI 2144
      AX=(X(I)-CUTMID)/CUTHLF                                           TWI 2145
```

```
      IF(AX.GT.1.0)AX=1.0                                                TWI 2146
      IF(AX.LT.-1.0)AX=-1.0                                              TWI 2147
      IF(AX.GT.0.0) GOTO 26                                              TWI 2148
      AX=AX*ROWWGT(I)                                                    TWI 2149
      AXNEG=AXNEG-AX                                                     TWI 2150
   25 J=IDAT(ID)                                                         TWI 2151
      IF(J.EQ.-1) GOTO 30                                                TWI 2152
      Y(J)=Y(J)-AX                                                       TWI 2153
      ID=ID+1                                                            TWI 2154
      GOTO 25                                                            TWI 2155
   26 AX=AX*ROWWGT(I)                                                    TWI 2156
      AXPOS=AXPOS+AX                                                     TWI 2157
   27 J=IDAT(ID)                                                         TWI 2158
      IF(J.EQ.-1) GOTO 30                                                TWI 2159
      YY(J)=YY(J)+AX                                                     TWI 2160
      ID=ID+1                                                            TWI 2161
      GOTO 27                                                            TWI 2162
   30 CONTINUE                                                           TWI 2163
      RETURN                                                             TWI 2164
      END                                                                TWI 2165
C                                                                        TWI 2166
      SUBROUTINE DIAGR(IPICT,MMZ,MMS)                                    TWI 2167
C PRINTS OUT DIAGRAM OF RELATION BETWEEN ZONES AND INDICATOR-SCORES      TWI 2168
C STORED IN MATRIX IPICT.  *'S ARE USED AS PUNCTUATION.                  TWI 2169
      INTEGER IPICT(MMZ,MMS)                                             TWI 2170
      COMMON/PICT/MZ,MS,IZD,IIZD,ISD,ISHIFT,MZIND,MZCRIT,MZOUT,NIND      TWI 2171
      IBIG=1000000                                                       TWI 2172
      MZZ=MZ+2                                                           TWI 2173
      DO 10 IZ=1,MZZ                                                     TWI 2174
      IPICT(IZ,MS+1)=IBIG                                                TWI 2175
   10 IPICT(IZ,MS+2)=IZ                                                  TWI 2176
      MSS=MS+2                                                           TWI 2177
      DO 20 IS=1,MSS                                                     TWI 2178
      IPICT(MZ+1,IS)=IBIG                                                TWI 2179
   20 IPICT(MZ+2,IS)=IS-ISHIFT                                           TWI 2180
      IPICT(MZZ,MSS)=IBIG                                                TWI 2181
      IPICT(MZ+2,MS+1)=IBIG                                              TWI 2182
      DO 30 IS=1,MSS                                                     TWI 2183
      IIZ=MZ-IZD+2                                                       TWI 2184
      IZ=MZ+4                                                            TWI 2185
      DO 21 IIIZ=1,IIZ                                                   TWI 2186
      IPICT(IZ,IS)=IPICT(IZ-2,IS)                                        TWI 2187
   21 IZ=IZ-1                                                            TWI 2188
      IPICT(IZ,IS)=IBIG                                                  TWI 2189
      IZ=IZ-1                                                            TWI 2190
      IIZ=IZD-IIZD                                                       TWI 2191
      DO 22 IIIZ=1,IIZ                                                   TWI 2192
      IPICT(IZ,IS)=IPICT(IZ-1,IS)                                        TWI 2193
   22 IZ=IZ-1                                                            TWI 2194
      IPICT(IZ,IS)=IBIG                                                  TWI 2195
   30 CONTINUE                                                           TWI 2196
      MZZ=MZ+4                                                           TWI 2197
      DO 40 IZ=1,MZZ                                                     TWI 2198
      IS=MS+3                                                            TWI 2199
      IIS=MS-ISD+2                                                       TWI 2200
```

```
      DO 31 IIIS=1,IIS                                                  TWI 2201
      IPICT(IZ,IS)=IPICT(IZ,IS-1)                                       TWI 2202
   31 IS=IS-1                                                           TWI 2203
      IPICT(IZ,IS)=IBIG                                                 TWI 2204
   40 CONTINUE                                                          TWI 2205
C WE NOW HAVE THE DESIRED INFORMATION IN IPICT;  ALL WE NOW NEED TO     TWI 2206
C DO IS TO PRINT IT OUT                                                 TWI 2207
      MSS=MS+3                                                          TWI 2208
      IS=MSS                                                            TWI 2209
      DO 50 IIIS=1,MSS                                                  TWI 2210
      PRINT 1000,(IPICT(IZ,IS),IZ=1,MZZ)                                TWI 2211
 1000 FORMAT(1X,60I2)                                                   TWI 2212
      IS=IS-1                                                           TWI 2213
   50 CONTINUE                                                          TWI 2214
      RETURN                                                            TWI 2215
      END                                                               TWI 2216
C                                                                       TWI 2217
      SUBROUTINE XLIST(M,MM,X,IIROW,ICAT1,ICAT2,                        TWI 2218
     1ICAT3,INAME1,INAME2)                                              TWI 2219
C PRINTS OUT LIST OF INDIVIDUALS (ROWS) FOR WHICH X=ICAT1,2,3.          TWI 2220
      REAL X(M)                                                         TWI 2221
      INTEGER IIROW(M),ITEM(200)                                        TWI 2222
      INTEGER INAME1(MM),INAME2(MM)                                     TWI 2223
      COMMON/WORK/ITEM                                                  TWI 2224
      COMMON/SECOND/ISEC                                                TWI 2225
      NITEM=12                                                          TWI 2226
      IF(ISEC.EQ.2) NITEM=11                                            TWI 2227
      ITEND=2*(NITEM-1)                                                 TWI 2228
      IT=0                                                              TWI 2229
      I=0                                                               TWI 2230
    3 I=I+1                                                             TWI 2231
      IIX=IFIX(X(I)+0.01)                                               TWI 2232
      IF((IIX-ICAT1)*(IIX-ICAT2)*(IIX-ICAT3).NE.0) GOTO 5               TWI 2233
      II=IIROW(I)                                                       TWI 2234
      IT=IT+1                                                           TWI 2235
      ITEM(IT)=INAME1(II)                                               TWI 2236
      IT=IT+1                                                           TWI 2237
      ITEM(IT)=INAME2(II)                                               TWI 2238
    5 IF(I.EQ.M) GOTO 6                                                 TWI 2239
      IF(IT.LE.ITEND) GOTO 3                                            TWI 2240
      IF(ISEC.EQ.2) GOTO 10                                             TWI 2241
      PRINT 1000,(ITEM(IIT),IIT=1,IT)                                   TWI 2242
      IT=0                                                              TWI 2243
      GOTO 3                                                            TWI 2244
    6 IF(ISEC.EQ.2) GOTO 15                                             TWI 2245
      IF(IT.NE.0) PRINT 1000,(ITEM(IIT),IIT=1,IT)                       TWI 2246
 1000 FORMAT(12(1X,2A4,1X))                                             TWI 2247
      RETURN                                                            TWI 2248
   10 PRINT 1001,(ITEM(IIT),IIT=1,IT)                                   TWI 2249
      IT=0                                                              TWI 2250
      GOTO 3                                                            TWI 2251
   15 IF(IT.NE.0) PRINT 1001,(ITEM(IIT),IIT=1,IT)                       TWI 2252
 1001 FORMAT(11(2X,A4,1X,A4))                                           TWI 2253
      RETURN                                                            TWI 2254
      END                                                               TWI 2255
```

```
C                                                                       TWI 2256
      SUBROUTINE YLIST(N,NN,Y1,Y2,IY,JJCOL,IIY,JNAME1,JNAME2,JNAM)      TWI 2257
C PRINTS OUT LIST OF ATTRIBUTES (COLUMNS) FOR WHICH IY(J)=IIY,          TWI 2258
C TOGETHER WITH NUMBERS OF OCCURRENCES ON 2 SIDES OF DICHOTOMY.         TWI 2259
      REAL Y1(N),Y2(N)                                                  TWI 2260
      INTEGER IY(N),JJCOL(N),ITEM(200)                                  TWI 2261
      INTEGER JNAME1(NN),JNAME2(NN),JNAM(NN)                            TWI 2262
      COMMON/WORK/ITEM                                                  TWI 2263
      NITEM=6                                                           TWI 2264
      ITEND=5*(NITEM-1)                                                 TWI 2265
      IT=0                                                              TWI 2266
      J=0                                                               TWI 2267
    3 J=J+1                                                             TWI 2268
      IF(IY(J).NE.IIY) GOTO 5                                           TWI 2269
      JJ=JJCOL(J)                                                       TWI 2270
      IT=IT+1                                                           TWI 2271
      ITEM(IT)=JNAME1(JJ)                                               TWI 2272
      IT=IT+1                                                           TWI 2273
      ITEM(IT)=JNAME2(JJ)                                               TWI 2274
      IT=IT+1                                                           TWI 2275
      ITEM(IT)=JNAM(JJ)                                                 TWI 2276
      IT=IT+1                                                           TWI 2277
      ITEM(IT)=IFIX(Y1(J)+0.001)                                        TWI 2278
      IT=IT+1                                                           TWI 2279
      ITEM(IT)=IFIX(Y2(J)+0.001)                                        TWI 2280
    5 IF(J.EQ.N) GOTO 6                                                 TWI 2281
      IF(IT.LE.ITEND) GOTO 3                                            TWI 2282
      PRINT 1000,(ITEM(IIT),IIT=1,IT)                                   TWI 2283
      IT=0                                                              TWI 2284
      GOTO 3                                                            TWI 2285
    6 IF(IT.NE.0) PRINT 1000,(ITEM(IIT),IIT=1,IT)                       TWI 2286
 1000 FORMAT(6(1X,A4,1X,A4,I1,'(',I3,',',I3,')'))                       TWI 2287
      RETURN                                                            TWI 2288
      END                                                               TWI 2289
```