SCIENTIA
WNNR · CSIR

SPECIAL REPORT

PERS 100

C.H. MOSS, G.V. KASS and E.A. MAUGHAN-BROWN

SUBROUTINES FOR THE N.I.P.R. BATCH PROCESSING SYSTEM

REVISED EDITION

NATIONAL INSTITUTE FOR PERSONNEL RESEARCH
COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH

CSIR Special Report No. 100

UDC 681.3

Johannesburg, South Africa. February 1970

SYSTEM SUBROUTINES FOR THE N.I.P.R. BATCH PROCESSING PROGRAMS.

May, 1968.
Revised November, 1969.

INTRODUCTION

The first set of system subroutines was written in Assembly language for the IBM 704 computer. With the conversion to the IBM 360 machine, the set was first rewritten in Basic Fortran and when full Fortran IV was available the set was rewritten with many additicnal facilities. The final version was put into operation in August 1967.

The subroutines in section B are a selection of the more generally useful subroutines that have been written by various members of the division for different programs.

N.I.P.R. Package

The NIPR package is a set of subroutines for facilitating the writing of programs for the NIPR system. Its use ensures a uniform method of data storage so that the same data may be used without modification by many different programs.

The system also makes possible the inclusion of new facilities, for example in the future it will allow for the selection and rejection of data, without any modification of the computing programs.

There follows (I) a list of the present routines and their purpose

(II) a short guide to system conventions

(III) a description of the method for working out package storage requirements

(IV) a more detailed description of the subroutines and their parameters

(V) a list of error reports given by the package.

I. List of routines in the Basic Package

(those in parentheses () are specific to the package and are not used by programmers)

A. Basic Subroutines

Page		
A1	(CLSALL)	Close all currently open files.
A2	(CLSINP)	Close an input file.
A3	(CMSAVA)	A substitute for named common.
A4	CNV	Forms a double word containing 8 alphameric or special characters given integer representations of the characters required.
A5	DATE	Supplies the time and the date to the calling program.
A6	EBCDIC	Converts a Fortran integer to EBCDIC.
A7	ELT	Supplies elapsed time since the last call to STINT.
A8	FINISH	Normal terminating routine.
A9	(FREECM)	Frees an area in the common storage used by the package.
A10	(GETCM)	Obtains an area in the common storage used by the package.
A11	HEAD	Prints out a heading at top of a new page.
A12	IANAL	Tests if a double precision name refers to a data set.
A13	IN	Opens an NIPR formatted data file for input.
A13.1	INDSEQ	Compares two double precision alphanumeric names.
A14	INFCN	Supplies the column names of a given data set.
A15	INFCVN	Supplies a specified column name of a given data set.
A16	INFIP	Obtains the value of a specified parameter from the parameter record of a data set.
A17	INFJOB	Obtains the current jobstep title.
A18	INFNC	Supplies the no. of cases processed from a data set.
A19	INFNE	Supplies the no. of incorrect cases read from a data set.
A20	INFNV	Supplies the no. of variables for a data set.

B. Computational or Functional Subroutines

II. How to write a programme for the NIPR system

1. The purpose of using the package is:-

   (a) To ensure that programmes use consistent formats for input and output on magnetic tape or disc so that any programme can use any NIPR file.

   (b) To have some uniformity in control cards.

   (c) To standardise the housekeeping information (e.g. programme, parameter and data set identification, number of records read or written, error diagnostics etc.) printed out by each programme.

   (d) To simplify programme documentation by eliminating the need for describing data formats, error returns, etc.

   (e) Although the programme does not have to provide storage for the title, parameters and variable names of input files, the information can be obtained by using INFJOB for the title, INFNV or INFIP for the parameters, and INFRN, INFRVN, INFCN, INFCVN for variable names.

   The number of cases read in by any of the input subroutines is given by the function INFNC.

2. (a) The subroutine START is called to read in the job-step title and parameter cards, initialise storage used by the package, and print the parameters, the clock time and the date.
   The arguments of START are NP and KPARM and an array KPARM of dimension NP must be provided.
   The parameters are integers.

   (b) Data is read from magnetic tape or disc in binary mode in the NIPR standard format with subroutines RD (after opening the file with IN) or with RMB or RMBF (without IN). In special cases where the input is small, such as a supplementary matrix, card input may be provided as an alternative.

(c) Output is written on tape or disc in standard NIPR format with the subroutine WD (opening the file with OUT and closing it with WEF), or WMB or WMBF (without OUT or WEF), or printed.

(d) The NIPR standard layout for data on tape or disc is as follows:-

(i) The first record of a file contains the title card read in by the subroutine START for the step in which the file was generated.

(ii) The second record contains the file parameters, viz. the number of variables, the number of cases (or records), the form of the data matrix (0 = rectangular, 1 = symmetric, 2 = diagonal) and five others used for transmitting information between programmes.

(iii) The third record contains the names of the variables in double length words.

(iv) The remaining records have the following format:-

First word is a double length word containing the case number or case name in EBCDIC.

Subsequent words contain data in single precision hexadecimal floating point.

(v) The last record is a dummy record marking the end of file. It has same format as a data record, and case number OOABCDEF (in hexadecimal).

(e) All matrices, unless otherwise stated, are compacted into uni-dimensional arrays. (The elements of half-symmetric matrices are written in the order (1,1), (1,2), (2,2), etc.

(f) All programmes are terminated either with FINISH (normal ending) or, if errors have occurred, with UNCLE.

(g) UNCLE is available to programmers for reporting an error with an identifying number (>100) and terminating the job-step with a trace-back.

3. Variable names.

Storage for variable names (up to 270 names per file) is provided in the package for input files. The name of the Ith variable of the data set numbered NDS can be obtained by calling the double precision function INFRVN or the subroutine INFRN. Variable names must be provided for output files and this can be done in four ways. If the NAMES argument in OUT is zero, integer names are generated. If the argument is a positive integer this is interpreted as a data set number and the variable names of that data set are used. If it is a negative integer the absolute value is interpreted as a data set number, but in this case the column or case names are used provided the whole matrix (file) has been read in by RMB or RMBF. Anything else in the argument is interpreted as the name of a double precision array containing the required names.

If the matrix has been read in by RMB or RMBF, the column or case names can be accessed with the subroutines INFCN or INFCVN.

Modified output variable names that are related to an input set can be generated by using the subroutine TAG which shifts all characters of a name one position to the left and places a specified character in the last position.

III. Package Storage Requirements

In order to reduce the storage space required by NIPR programs, the package has been altered to reduce the amount of space available to it for storing information about the data sets being used from 2000 words to 900 words. This means that, in general, the maximum size of arrays of data read in by programs will be considerably diminished. However where the reduction in the number of variables causes inconvenience, e.g. with programs using several input data sets, the program should be compiled with a "large" version of subroutine START, requesting an appropriate amount of space determined as outlined below.

Calculation of storage requirements for a program

This can be done as follows:

1. A certain amount of space is reserved by package for general storage purposes. It includes:

Job Set Up Information	=	24 words
Vector of Data Sets	=	40 words
		64 words

For package with 900 words storage this leaves 900-64=836 words for storing information about data sets. This space is appropriated when asked for according to the rules stated below.

Note: that if an odd number of words of storage is requested package rounds this number up to the next multiple of 2.

2. Input Buffer:

If there are any input files an input buffer will be required with length in words equal to the largest number of variables in an input data set (rounded up, if necessary).

Note: in this connection that it is more efficient to open the data set with the largest number of variables first.

3. Each input file requires:

Compulsory

1. For file dope vector 20 words
2. For variable names (if NV=no. of variables in data set) (2*NV)+2 words

Optional

1. If RMB is used, space required for column names* (where NC=no. of column (case) names) (2*NC)+2 words
2. If a diagonal matrix is to be read by RMB additional vector space must be available when RMB is called, NV words (but this will be freed by RMB immediately (rounded up, after use)** if necessary)
   ([illegible]

*The column name space will be freed again if a diagonal matrix on disc is read into a full matrix, a half symmetric matrix or a diagonal matrix. However this space must be available to the program at the time that RMB is called.

**The row name space will be freed if a lxN vector on disc is read as a diagonal matrix.

4. Each output file requires:

Compulsory

1. For file dope vector 4 words

Optional

1. If integer names are to be generated and input buffer is less than 2*NV 2*NV words (This space must be available when requested but will be freed immediately after use)
2. If a half-symmetric matrix is to be written to disk and the input buffer is less than NV (No. of vars. per row) NV words (Space must be available, but will be freed (rounded up, immediately) if necessary)

It will be seen that the effective size of package can be considerably increased by judicious sequencing of the instructions for opening input and output files. (Because input data sets usually retain the space acquired by them, while most of the space used by output data sets is freed once they have been opened). E.g. A program requires 2 input data sets and 1 output data set for which integer names are to be generated:

(i) If data sets are opened in the sequence Input (1), Input (2), Output, the maximum number of variables in each data set is 112

[836 (space available)=
246 (1st Input)+112 (Buffer)+246 (2nd Input)+228 (Output)+4 (Unused)]

(ii) Whereas, if data sets are opened in the sequence Input (1), Output, Input (2), the maximum number of variables in each data set is 157

[836 (space available)=
336 (1st Input)+158 (Buffer, rounded up)+318 (Output) + Unused as yet
= 336 (1st Input)+158 (Buffer)+4 (Output)+336 (2nd Input)+2 (Unused)
(After 2nd Input has been opened).]

Detailed Examples showing calculation of space requirements

E.g. 1) Data sets : 1 Input, opened with IN
1 Output, names not generated.

Subtract constant requirements for data sets to find space available for storage of variables

i.e.	836 words	(Total Space Available)
-	20	(Dope Vector for Input File)
-	2	(For row names for Input File)
-	4	(Output File Dope Vector)
	810	(Remaining Space)

To find NV (the maximum number of variables permitted in the input data set), divide the space remaining by 3 (NV words are required for the input buffer and 2*NV for storing variable names)

i.e. $3\underline{|810\quad}$

270.0 (This would be truncated at the decimal point, if necessary).

i.e. Maximum no. of variables in the input data set is 270.

Checking, we find that

20 (Input dope vector)
542 (i.e. 2x70+2) (Variable Names)
270 (Input buffer)
4 (Output buffer)
836 Just fits into the available space.

2) Data sets : 1 Input, opened with RMB (and assumed to be square)
1 Output, names not generated.

Subtraction of constant requirements yields

836 words (Total Space Available)
- 26 (20+2+4 as in e.g. 1)
- 2 (For column names for Input File)
808 (Space remaining)

To find NV divide space remaining by 5 (NV words required for input buffer, 2*NV for row name storage and 2*NV for column name storage)

$5\underline{|808\quad}$

161.6

i.e. Maximum number of input variables when RMB is used is <u>161</u>.

Table 1 indicates the maximum permissible number of variables for the more commonly used data set combinations for package of size 900. It assumes (1) equal numbers of variables in all data sets, (2) square matrices when RMB used, and (3) all input data sets opened before all output data sets.

Table 1

Total No. of Data Sets	No. of Input Data Sets	How opened	No. of Output Data Sets	Names gen. or not	Max.no. of vars.
1	1	IN	0	-	271
1	0	-	1	Not generated	Unlimited
2	1	IN	1	Not generated	270
2	1	RMB	1	Not gen.	161
2	1	IN	1	Gen.	161
2	1	RMB	1	Gen.	115
8	1	IN	7	Not gen.	260
8	1	IN	7	All gen.	150
3	2	IN,IN	1	Not gen.	157
3	2	RMB,RMB	1	Not gen.	87
3	2	IN,RMB	1	Not gen.	112

Table 2 indicates the amount of package storage space required to accommodate data sets larger than those indicated in Table 1. It assumes (1) 1 output data set, names not generated, (2) square matrices when RMB used, and (3) equal numbers of variables in all data sets where more than one considered.

Table 2

	Data Sets Read with RD		Data Sets Read with RMB.	
No. of Variables	No. of Data Sets Read	Space Required	No. of Data Sets Read	Space Required
200	1	690	1	1092
300	1	990	1	1592
500	1	1590	1	2592
700	1	2190	1	3592
1000	1	3090	1	5092
200	2	1112	2	1916
300	2	1612	2	2816

IV. Description of the Subroutines and their Parameters

Conventions used for Subroutine Arguments

Parameter	Type	Use or Meaning
A	Real array	Data, usually internal.
CASE	Double precision variable	Name of a record.
CH	Double precision	Column headings (case names).
I	Integer variable	Various uses.
IP	Integer array, dimension 8	List of 8 parameters for a file:-
	IP(1) =	No. of variables.
	IP(2) =	No. of cases.
	IP(3) =	Form of data matrix, 0 = Rectangular 1 = Half Symmetric 2 = Diagonal.
	IP(4) =	No. of cases on which the data is based.
	IP(5),IP(6) =	Undefined as yet.
	IP(7) =	Parameter identifying records for tabulation programmes.
	IP(8) =	Undefined as yet.
NAMES	Double precision array	Variable or case (record) names, or small integral value. (See (a) below).
NCA	Integer	No. of Columns in A.
NDS	Integer	No. of Data Set as used by Fortran programmes.
NRA	Integer	No. of Rows in A.
REAL*8	Fortran convention for defining a real, double precision array	Used to indicate when variables or arrays should be double precision.
RH	Double precision	Row headings (variable names).
RXY	Real array	Data, usually formed by cross products.
SC	Real array	Data, usually initial input.

(a) Variable names may be referenced through a data set which has already been opened, by putting NAMES (1) equal to that data set number. e.g. If NAMES (1) = 12.ODO, then the names are taken from data set No. 12 (which must have already been opened). The use of generated integers (i.e. 1, 2, 3, 4, ...) is provided by putting NAMES (1) = 0.ODO.

(b) Subroutines are written in Fortran except where otherwise stated.

(c) Data matrices consist of cases written column-wise with variables row-wise. On external storage data matrices are stored as rectangular matrices with each column forming a record. Internally they are held either as single records, or if the complete matrix is in core, as a unidimensional array for half-symmetric matrices (the upper triangle of a symmetric matrix), or as a unidimensional or two-dimensional array in other cases.

SUBROUTINE CLSALL

Purpose : Close all currently open files, and print their data set numbers.

Called by FINISH

Routines Called : UNCLE

WEF

SUBROUTINE CLSINP(NDS)

Purpose : Close an input file that is positioned after the final record. If any records remain to be read, UNCLE is called.

Called by RMB.

Argument : NDS = Integer, input = Data set number of the input file.

Routines Called : FREECM
RD
UNCLE

SUBROUTINE CMSAVA(I,N,V)

Purpose : This routine is used as a substitute for named common to provide storage for timing intervals.

Arguments : If $I \leq 0$, set internal array TIME(N+1) = V
If $I > 0$, " V = TIME(N+1)
N = Integer specifying a timing interval.
V = Real number = a time.
Called by the timing subroutines
(See ELT, STINT).

Routines Called : none.

REAL*8 FUNCTION CNV(N1, ... N8)

Purpose : Return the low order bytes of N1 to N8 concatenated together. An assembly language subroutine.

Usage : REAL*8 CNV,NAME
NAME = CNV(N1, ... N8)

Arguments : N1 to N8 : Integer, inputs.

Routines Called : none.

SUBROUTINE DATE (Y,MM,D,H,M,S)

Purpose : Return the time and the date, correctly treating Leap years and centuries.

Usage : INTEGER*4 Y, MM,D,H,M,S
CALL DATE (Y,MM,D,H,M,S)

All arguments are integer, output.

Y :	The current year
MM :	The current month
D :	The current day of the month
H :	The current hour
M :	The current minute
S :	The current second.

Routines Called : TIME.

DOUBLE PRECISION FUNCTION EBCDIC (I)

Purpose : Convert a Fortran integer to EBCDIC

Usage : REAL*8 A,EBCDIC

```
I = 137
A = EBCDIC(I)
```

A now contains the EBCDIC representation of I

i.e. in Hexadecimal :-

A becomes 4040404040F1F3F7

Argument : I = Integer, input.

Restrictions : $I > 0$

Routines Called : CNV

UNCLE

FUNCTION ELT (N)

Purpose :	Give the elapsed time in minutes since the last call to STINT with the same identifier N.
Usage :	X = ELT(N)
Argument :	N = Integer, input = 0, 1, 2, 3.
Restrictions :	(i) Only N = 1, 2 or 3 may be used as the package uses N = 0. (ii) CALL STINT(N) must precede the use of ELT(N).
Routines Called :	CMSAVA TTIMER

SUBROUTINE FINISH

Purpose : Terminate a programme, close any open files and print CPU time used, real time of day, and the step title.

Usage : CALL FINISH

Routines Called : CLSALL
ELT
PRDATE
TTIMER

SUBROUTINE FREECM(LENGTH, INDEX)

Purpose : Returns an area in common /CHMXYZ/ to the list of free areas.

Usage : CALL FREECM (LENGTH, INDEX)

Action : LENGTH is rounded up to the next highest multiple of 2, and an area of LENGTH words long, starting at CHMXYZ(INDEX) is returned to the free list used by the package.

Restrictions : Area to be freed must not overlap an area currently in the free list nor the end of common.

Arguments : LENGTH = Integer, input
INDEX = Integer, input.

Routines Called : UNCLE.

SUBROUTINE GETCM(LENGTH, INDEX)

Purpose : Obtain an area in common /CHMXYZ/
from the list of free areas.

Usage : CALL GETCM (LENGTH, INDEX)

Action : An area of length equal to LENGTH rounded
up to a multiple of 2 is removed from the
free chain used by the package.
The location of the area is returned in
INDEX.

Argument : LENGTH = Integer, input
INDEX = Integer, output.

Routines Called : UNCLE.

SUBROUTINE HEAD (N)

Purpose : Print out current step title, date, time and page number at the top of a new page.

Action : Takes a new page and prints out the standard information.
If N≤O No page number is given.
If N=+ve number, the first page is numbered N and N is incremented by 1 ready for subsequent calls to the routine.

Usage : CALL HEAD (N)

Argument : N = Integer, input and output.

Routines Called : DATE
INFJOB

FUNCTION IANAL (D)

Purpose : Determine whether D is the number of a data set.

Action : IANAL = Integer value of D if D has an integer value or zero.

= -999 if D is EBCDIC.

Usage : REAL*8 D
I = IANAL (D)

Argument : D = Double precision, input.

Restriction : (Tested for integers < 100)

Routines Called : none.

SUBROUTINE IN(NDS)

Purpose : Open an NIPR formatted data file for input.

Usage : CALL IN(NDS)

NDS is a Fortran data set number.

Action : This opens the data set NDS, and saves information about it in its file dope vector.

A message that the file has been opened and the data set number, the title of the job that created the data, the file parameters, and the creation date are printed.

Restrictions : NDS must not be equal to the installation defined data set numbers for SYSIN and SYSOUT, and must be less than the installation defined maximum.

The data set NDS must not be open. (An input data set is automatically closed when the end of data is read.)

Arguments : NDS = Integer, input.

Routines Called : FREECM
GETCM
INTERP
UNCLE.

FUNCTION INDSEQ (MAN1,MAN2)

Purpose : Indicate the numeric sequence of two numeric case names which have been read in with 'A' format. (Note: This subroutine should <u>always</u> be used when A-format numeric names are compared for sequence. It is designed to eliminate problems caused by the unexpected sign changes which can invalidate direct numeric comparisons of EBCDIC numbers. When making comparisons the function does not allow for shifted columns e.g. 19 will be higher than ' 2',but '2 'will be regarded as higher than 19).

Action : INDSEQ = Negative (i.e.-1) if MAN1< MAN2
= 0 if MAN1= MAN2
= Positive (i.e.+1) if MAN1> MAN2

Usage : REAL*8 MAN1,MAN2
IF (INDSEQ (MAN1,MAN2)) 10,20,30

Arguments : MAN1 = Double precision, input
MAN2 = Double precision, input

Routines Called : none.

SUBROUTINE INFCN (NDS,NAMES)

Purpose : Supply column names (case names) of the data matrix in the data set numbered NDS.

Usage :
```
REAL*8 NAMES (500)
CALL INFCN (NDS,NAMES)
```

Action : The column names (case names) are placed in the REAL*8 array NAMES.

Arguments : NDS = Integer, input.
NAMES = Double precision array, output.

Restrictions : NDS must refer to a matrix that has been read in by RMB or RMBF.

Routines Called : UNCLE.

DOUBLE PRECISION FUNCTION INFCVN(NDS,I)

Purpose : Supply a single column (case) name.

Usage : REAL*8 NAME,INFCVN

```
NAME =   INFCVN(NDS,I)
```

Action : NAME contains the Ith column or case name of the data set NDS.

Arguments : NDS = Integer, input.

I = Integer, input.

Restrictions : NDS must refer to a matrix that has been read in by RMB or RMBF.

Routines Called : UNCLE.

FUNCTION INFIP(NDS,I)

Purpose : Obtain the value of a given parameter of a data set.

Usage : J = INFIP(NDS,I)

Action : J is the value of the I^{th} parameter of data set NDS.

Arguments : NDS = Integer, input.
I = Integer, input.

Restrictions : $1 \leq I \leq 8$
NDS must have been opened for input.

Routines Called : UNCLE.

SUBROUTINE INFJOB(A)

Purpose : Obtain the current job step title.

Usage : DIMENSION A(18)
CALL INFJOB(A)

Action : The current job step title will be placed in A.

Argument : A = Real array, output.

Routines Called : none.

FUNCTION INFNC(NDS)

Purpose : To obtain the number of cases read correctly from or written to the data set numbered NDS.

Usage : K = INFNC(NDS)

Action : If NDS is an input file, K is the number of cases read and passed to the programme.
If NDS is an output file, K is the number of cases written.

Argument : NDS = Integer, input.

Restrictions : NDS must have been opened.

Routines Called : UNCLE.

FUNCTION INFNE(NDS)

Purpose : To obtain the number of records incorrectly read.

Usage : K = INFNE(NDS)

Action : K is set to the number of errors.

Argument : NDS = Integer, input.

Restriction : This is a dummy subroutine at present.

Routines Called : none.

FUNCTION INFNV(NDS)

Purpose : Obtain the number of variables in the data set numbered NDS.

Usage : K = INFNV(NDS)

Action : K is set to the number of variables in the data set NDS.

Argument : NDS = Integer, input.

Restrictions : NDS must have been opened.

Routines Called : UNCLE.

FUNCTION INFRN(NDS,NAMES)

Purpose	:	Supply row names for data set numbered NDS.
Usage	:	REAL*8 NAMES (500) CALL INFRN(NDS,NAMES)
Action	:	The row names of data set NDS will be placed in the array NAMES. Each name of 8 characters will occupy 1 double word. (Note : a row name is a variable name.)
Arguments	:	NDS = Integer, input. NAMES = Double precision array, output.
Restrictions	:	NDS must have been opened for input.
Routine Called	:	UNCLE.

DOUBLE PRECISION FUNCTION INFRVN(NDS,I)

Purpose : Supply the Ith row name (variable name) for data set numbered NDS.

Usage : REAL*8 INFRVN,A
A = INFRVN(NDS,I)

Action : The Ith variable name will be placed in A.

Arguments : NDS = Integer, input.
I = Integer, input.

Restrictions : NDS must have been opened for input.

Routines Called : UNCLE.

SUBROUTINE INICM (LENGTH,ND,LUSER)

Purpose : Initialise the COMMON/CHMXYZ/ for a particular version of the package.

Usage : CALL INICM(LENGTH,ND,LUSER)

LENGTH is the length of the common array (present version = 900)

ND is the maximum data set number (present version = 40)

LUSER is the length of the user area (0)

(This argument is left over from an earlier idea, and is not used.)

Arguments : LENGTH = Integer, input.

ND = Integer, input.

LUSER = Integer, input.

Restrictions : This subroutine is called from START.

It should not normally be used elsewhere.

Routines Called : none.

SUBROUTINE INOUT(NDS,NV,INK)

Purpose : Open an NIPR formatted data file (data set no.NDS), which has already been started and position it at a specified point in the data set, so that additions to the file can be made from that point.

Usage : CALL INOUT(NDS,NV,INK)

Action : This opens the data set NDS and checks that the number of variables in the first parameter of the IP array is equal to NV, the number of variables expected. If not, an error message is printed and there is a transfer to UNCLE.

A message that the file has been opened and the data set number, the title of the job that created the data, the file parameters, and the creation date are printed.

However information about the IP array and the Variable Names is not stored in the file dope vector for further use.

INK cases of the data set are then read, leaving the data set positioned so that additions to it will begin immediately after the INK'th case.

A message giving the name of the last case read, and indicating that additions to the data set will start at that point, is printed.

Restrictions : NDS must not be equal to the installation defined data set numbers for SYSIN and SYSOUT, and must be less than the installation defined maximum.

The data set NDS should not have been opened, and left open, by the calling program.

(An input data set is automatically closed when the end of data is read).

A23.1 (Continued)

Arguments : NDS = Integer, input
NV = Integer, input
INK = Integer, input

Routines Called : FREECM
GETCM
UNCLE.

SUBROUTINE INTERP(NDS)

Purpose : To interpret the variable and case selection information read in by RDFS and to write variable selection arrays for data set numbered NDS.

Usage : CALL INTERP(NDS)

Argument : NDS = Integer, input.

Note : At the moment this is a dummy subroutine.

Routines Called : none.

FUNCTION ISYM(I,J)

Purpose : Return the position in a half-symmetric array of the (I,J)th element of a matrix.

Usage : K = ISYM(I,J)
A = B(K)

Action : If B contains the half-symmetric array, then A will contain the (I,J)th element of that array.

Arguments : I = Integer, input.
J = Integer, input.

Routines Called : none.

FUNCTION ITCH(A,N)

Purpose : Obtain the Nth byte of A.
An Assembly language routine.

Usage : K = ITCH(A,N)

K has the value of the Nth byte of A.

Arguments : A = Real, input.
N = Integer, input.

Note : Applies to REAL*8 numbers and character strings.

Routines Called : none.

FUNCTION MISS(X)

Purpose : Test for missing information.

Usage : K = MISS(X)

Action : If X≠-1.E-50, K = 1
If X=-1.E-50, K = 0
Note: (-1.E-50 is the NIPR convention for missing information.)

Arguments : X = Symbolic address, input.

Routines Called : none.

SUBROUTINE OUT(NDS,IP,NAMES)

Purpose : Open data set numbered NDS for output, write and print the title, and parameter records from the array IP, and write the variable names record.

Usage : CALL OUT (NDS, IP,NAMES)

Action : The contents of the variable names record is determined by the value in NAMES.

If NAMES = 0.ODO then integer variable names are generated and written.

If NAMES = Double precision positive whole number between 1 and 40, the variable names are taken from the row names of the data set numbered NAMES after it has been converted to a single precision integer. The data set must have been opened.

If NAMES = Double precision negative whole number between 1 and 40, the variable names are taken from the column names of the data set numbered NAMES after it has been converted to a single precision positive integer. The data matrix must have been read by RMB.

If NAMES ≠ Double precision whole number then the variable names are taken from the array NAMES where NAMES(1) is the 1st variable name,
NAMES(2) " the 2nd " " etc.

Arguments : NDS = Integer, input.
IP = Integer array; dimension = 8, input.
NAMES = Double precision array, input.

Routines Called : DATE, EBCDIC, FREECM, GETCM, IANAL, UNCLE, INFRVN, INFCVN.

SUBROUTINE PRDATE

Purpose : Print the time and the date.

Usage : CALL PRDATE

Action : This prints the current time and date, without taking a new line, in the following format, starting in column 88:

```
DATE:bbdd/mm/yybbbbTIME:bbhh,mm.ss
     b = blank
```

Routines Called : DATE.

SUBROUTINE RD(NDS,CASE,SC)

Purpose : Read and select a record from an NIPR file.

Usage : REAL CASE*8,SC(100)
CALL RD(NDS,CASE,SC)

Action : Reads a record from data set numbered NDS, placing the variables in the SC array and the case (or record, or column) name in CASE.

If the end of file is reached, CASE is returned with a value of O.DO, and a message stating that the input file has been closed, and the number of records read is printed. After the end of file has been read, it may not be accessed again without re-opening with IN.

Arguments : NDS = Integer, input.
CASE = Double precision, output.
SC = Real array, output.

Restrictions : IN must have been called for the file NDS.

Routines Called : UNCLE.

ENTRY RDI(NDS,CASE,SC,IJK)

Purpose : To read and select a record from an NIPR data set No. NDS, returning the case name in CASE, and placing the variables into:
SC(1), SC(1+IJK), SC(1+2*IJK), SC(1+3*IJK) etc.

Usage : REAL CASE*8,SC(100)
CALL RD(NDS,CASE,SC,IJK)

Arguments : NDS = Integer, input.
CASE = Double precision, output.
SC = Real array, output.
IJK = Integer, input.

Restrictions : IN must have been called for the file NDS.

NOTE : (i) RDI is an entry point in RD.
(ii) If IJK = 1, then the result is the same as a call to RD.

Routines Called : UNCLE.

SUBROUTINE RDFS

Purpose : Read file selection cards.

Action : Called in START if cols. 1-4 of the parameter card are zero. This subroutine reads and stores the file selection information in uninterpreted form for use by the subroutine INTERP, called by IN the first time the file is opened.

Restriction : Do not use anywhere else.
It is presently a dummy routine.

Routines called : none.

SUBROUTINE RMB(NDS,A,LA,NRA,NCA)

Purpose : Read a matrix from disc or tape in NIPR format, and write it to array A.

Usage : CALL RMB(NDS,A,LA,NRA,NCA)

Action : A matrix of NRA rows and NCA columns in the data set numbered NDS is read into the array A. The matrix is always in rectangular form on disc or tape, but its form in core depends on the parameter LA.

If LA = 0,a rectangular matrix is requested.
If the matrix on disc is rectangular, it is copied into A.
If the matrix on disc is symmetric it is copied into A in full format.
If the matrix on disc is diagonal, it is written into A as a diagonal matrix with zeroes on the off diagonal elements.

If LA = 1, a half-symmetric matrix is requested.
If the matrix on disc is not square, an error message is printed and there is a transfer to UNCLE.
If the matrix on disc is rectangular, a warning message is printed and the upper triangle is copied into A.
If the matrix on disc is symmetric, the upper triangle is copied into A columnwise.
If the matrix on disc is diagonal, it is expanded into a half-symmetric matrix.

If LA = 2, a diagonal matrix is requested.
If the matrix on disc is rectangular
it must be square or Nxl or lxN otherwise
an error message is printed and there is
a transfer to UNCLE.
If the matrix on disc is rectangular or
half-symmetric, its diagonal is extracted;
or if it is an Nxl or lxN vector the
vector becomes the diagonal.
If the matrix on disc is diagonal, it
is copied into A.

If LA = 3, the matrix is required in the same
form as it is on the disc.
After it has been written LA is changed
to correspond to its actual form.

Arguments : NDS = Integer, input.
A = Real array, output.
LA = Integer, input and output.
NRA = Integer, output.
NCA = Integer, output.

Routines Called : CLSINP
FREECM
GETCM
IN
RD
UNCLE.

Restriction : The IP(2) file parameter (no. of cols.) should not be smaller than the number of columns actually in the data set when RMB is used.

SUBROUTINE RMBF(NDS,A,NDIM, NRA,NCA)

Purpose : Read an NRA x NCA rectangular matrix from data set numbered NDS into the array A (which has rows of length NDIM) in the standard 2-dimensional Fortran form instead of the packed NIPR form.

Usage : CALL RMBF(NDS,A,NDIM,NRA,NCA)

Action : Calls RMB

Arguments : NDS = Integer, input.
A = Real array, output.
NDIM = Integer, input.
NRA = Integer, output.
NCA = Integer, output.

Note : For further description of the compacted and full matrix forms see the IBM Scientific Subroutine Package Programmer's Manual (Version III) "Overall Rules of Usage" Pages 3-6

Routines Called : RMB
UNCLE.

SUBROUTINE START(NP,KPARM)

Purpose : Initialise the common area used by the package, read the job-step title card and the parameter card, determine if variable selection is required, and read NP parameters from the parameter card into the array KPARM.

Usage : CALL START (NP,KPARM)

Action : The title card is read from the 1st 72 columns in SYSIN and printed. The variable selection parameter is read from columns 1-4 of the parameter card and is acted upon as follows:-

(i) If the variable selection parameter is equal to 1, then the NP programme parameters are read into the array KPARM from columns 5-8, 9-12, 13-16, etc. (i.e. in 4 column fields starting from field 5-8).

(ii) If the variable selection parameter is greater than 1, then (NP+1) programme parameters are read into the array KPARM, from c columns 1-4, 5-8, 9-12, ... etc. (i.e. in 4 column fields starting from field 1-4). This is known as the "Packed Option".

(iii) If the variable selection parameter is equal to 0; or (in the Packed Option) greater than 1 but less than 1000, (i.e. Column 1 is equal to 0), RDFS is called to read the data selection specifications. (See RDFS).

Arguments : NP = Integer, input.

KPARM = Integer array, output.

Note : (i) A parameter card is required for the variable selection parameter even if there are no other parameters.

(ii) Parameters may be packed into 1-column fields (i.e. variable selection option in Column 1, next parameter in column 2, next parameter in column 3, ... etc.) by using the Packed Option. Parameters are then passed "4-at-a-time" starting with column 1 and must be unpacked by the programme.

Routines Called : CMSAVA
INICM
PRDATE
RDFS
STIMER.

SUBROUTINE STIMER(I,TIM)

Purpose : Set up a timing interval of length TIM minutes.

Usage : CALL STIMER (I,TIM)

Action : If I $\leq$ 0, sets up a time interval (of length TIM) which is to be decremented only when the associated task is active (i.e. it measures task time).

If I = 1, sets up a time interval (of length TIM) which is to be decremented continuously (i.e. it measures real time).

If I = 2, program goes into a WAIT state for an interval of length TIM.

Arguments : I = Integer, input.
TIM = Integer, input.

Note : This routine is usually used in conjunction with SUBROUTINE TTIMER(I).

Restriction : The Package calls STIMER in SUBROUTINE START.
(Setting an interval of 600 Minutes).

Routines Called : DATE.

SUBROUTINE STINT(N)

Purpose	:	Start timer number N.
Usage	:	CALL STINT(N)
Argument	:	N = Integer, input, = 0, 1, 2, 3
Restrictions	:	Only N = 1, 2, 3 may be used as the package uses N=0
Routines Called	:	CMSAVA TTIMER.

SUBROUTINE TAG(NAMES,N,NCHAR)

```
Purpose          :  Tag the character NCHAR at the end of N
                    successive double precision words in the
                    array NAMES.

Usage            :  REAL*8  NAMES(500)
                    CALL TAG(NAMES, N, NCHAR)

Arguments        :  NAMES  =  Double precision array, input and output.
                      N    =  Integer, input.
                    NCHAR  =  Integer, input.

                    Note:     In tagging the character at the end
                              of a name in NAMES, the name is
                              shifted left and the 1st character
                              is lost.

Routines Called  :  CNV
                    ITCH.
```

SUBROUTINE TIME(NSECS, NDAY, NYEAR)

Assembler routine.

Purpose	:	Supply the current time as NSECS seconds, NDAY days and NYEAR years (mod 100).
Usage	:	CALL TIME (NSECS, NDAY, NYEAR)
Arguments	:	All the arguments are Integer, output. Note : It is usually more convenient to use SUBROUTINE DATE.
Routines Called	:	none.

FUNCTION TTIMER(I)

Purpose : Supply the remaining time in minutes in the interval I set up by STIMER.

Usage : TMLEFT = TTIMER(I)

Action : If I=0, supply remaining time.

If I=1, supply remaining time and cancel the interval.

Argument : I = Integer, input.

Note : CALL STIMER (I,TIM) must preceed the use of this function. (This is done in SUBROUTINE START, setting an interval of 600 minutes).

Routines Called : DATE.

SUBROUTINE UNCLE(X)

Purpose : Print the number X as an error message and close all open files.

Usage : CALL UNCLE(X)

Action : The error message number is printed in format F7.2. and a trace back to the main programme is provided.

Argument : X = Real, input.

Note : The package error message numbers range from 1. to 99.. The remaining numbers may be used by programmers.
A List of the Error Message numbers and their meanings appears in Section C.

Routines Called : CLSALL
ELT
PRDATE
TTIMER.

SUBROUTINE WD(NDS,CASE,SC)

Purpose : To write a record on data set numbered NDS with the case name from CASE, and data from the array SC.

Usage :
```
REAL  CASE*8,SC(100)
CALL  WD(NDS,CASE,SC)
```

Arguments :

NDS	=	Integer, input
CASE	=	Double precision, input
SC	=	Real array, input.

Restriction : OUT must have been previously called for file numbered NDS.

Routines Called : UNCLE.

SUBROUTINE WEF(NDS)

Purpose : Write the terminal record and end of file to disc or tape for the data set numbered NDS and print a message to that effect and the number of records written.

Usage : CALL WEF(NDS)

Arguments : NDS = Integer, input.

Restriction : The file numbered NDS must be open for output.

Routines Called : UNCLE.

SUBROUTINE WMB(NDS,A,LA,NRA,NCA,IPX,RH,CH)

Purpose : Write a matrix A of size NRAxNCA and form LA into the NIPR formatted data set numbered NDS.

Usage : CALL WMB(NDS,A,LA,NRA,NCA,IPX,RH,CH)

Action : NRA = No. of rows of A, is written on the file parameter IP(1)

NCA = No. of columns of A, is written on the file parameter IP(2)

LA = Form of the matrix A, is written on the file parameter IP(3)

0 = Rectangular
1 = Half-Symmetric
2 = Diagonal

Note that the form applies to the matrix in core, in external storage it is always rectangular.

IPX = Last 5 parameters of IP array

RH = Row Headings for matrix A

CH = Column Headings for matrix A.

Note : If RH = A small positive no. - Names taken from Row Names of that data set.

If RH = A small negative no. - Names taken from Column Names of that data set.

If CH = A small positive no. - Names taken from Column Names of that data set.

If CH = A small negative no. - Names taken from Row Names of that data set.

If RH or CH = 0.ODO - Integer names will be generated.

Otherwise - Names will be taken from the double precision arrays RH or CH.

A43 (Continued)

Arguments : NDS = Integer, input
A = Real array, input
LA = Integer, input
NRA = Integer, input
NCA = Integer, input
IPX = Integer array, input
RH = Double precision array, input
CH = Double precision array, input.

Routines Called : EBCDIC
FREECM
GETCM
IANAL
INFCVN
INFRVN
ISYM
OUT
WD
WEF
UNCLE.

SUBROUTINE WMBF(NDS,A,LA,NDIM,NRA,NCA,IPX,RH,CH)

Purpose : Write a matrix of size NRA x NCA and form LA to the NIPR formatted data set numbered NDS from the array A which has rows of length NDIM (i.e. the matrix is in Fortran form instead of the packed NIPR form).

Usage : CALL WMBF(NDS,A,LA,NDIM,NRA,NCA,IPX,RH,CH)

Action : After compacting the matrix in A the subroutine WMB(NDS,A,LA,NRA,NCA,IPX,RH,CH) is called. The data is then restored to the original form.

The use of the arguments is the same as in WMB.

Arguments : NDS = Integer, input
A = Real array, input
LA = Integer, input
NDIM = Integer, input
NRA = Integer, input
NCA = Integer, input
IPX = Integer array, input
RH = Double precision array, input
CH = Double precision array, input.

Routines Called : WMB
UNCLE.

SUBROUTINE ASDLIM(AV,SD,AN,BCLA,UCLA,BCLS,UCLS)

Purpose : Calculate 95% confidence limits for the mean and for the standard deviation, for a sample of size AN, with mean AV and unbiassed standard deviation SD.

Usage : CALL ASDLIM(AV,SD,AN,BCLA,UCLA,BCLS,UCLS)

Action : If AN ≤ 1, BCLA=UCLA=BCLS=UCLS=0

If AN > 1, BCLA=Lower 95% confidence limit for mean
UCLA=Upper 95% confidence limit for mean
BCLS=Lower 95% confidence limit for S.D.
UCLS=Upper 95% confidence limit for S.D.

For AN ≤ 8

all confidence limits are found using table look-up techniques.

For AN > 8

confidence limits for the mean are calculated using an approximation for the "t" distribution

(Ref: Gardiner and Bombay, Technometrics, Vol.7, 1965, pp.71,72)

Confidence limits for the S.D. are calculated using an approximation to the X^2 distribution

(See Annals of Mathematical Statistics, Vol.17, 1946, p.220)

Arguments : AV=Real,input
SD=Real,input
AN=Real,input
BCLA=Real,output
UCLA=Real,output
BCLS=Real,output
UCLS=Real,output

Routines Called : SQRT.

SUBROUTINE AVLIM(AV,SD,BN,UCL,BCL)

Purpose : Calculate 95% confidence limits for a sample with mean AV, unbiassed standard deviation SD, based on BN observations, and using an approximation for "t" distribution.

(Ref: Gardiner and Bombay, Technometrics, Vol. 7, 1965, pp.71,72)

Usage : CALL AVLIM(AV,SD,BN,UCL,BCL)

Action : UCL = the upper confidence limit.
BCL = the lower confidence limit.

Arguments : AV = Real, input
SD = Real, input
BN = Real, input
UCL = Real, output
BCL = Real, output

Routines Called : none.

SUBROUTINE CHISQ (Y, NDFK, P)

Purpose : Calculation of inverse Chi Square. Given the probability P and degrees of freedom NDFK, the subroutine calculates the corresponding Chi Square deviate.

Usage : CALL CHISQ (Y, NDFK, P)

Arguments : NDFK = Integer, input
P = Real, input
Y = Real, output

Accuracy : For probabilities between .001 and .999 and any number of degrees of freedom, the subroutine is accurate to at least 2 decimal places.

Routines Called : FNORM
ALOG
SQRT

SUBROUTINE CONR(R,T,UCL,BCL)

Purpose : Compute 95% Confidence Limits for a correlation coefficient R, for a sample of size T.

Usage : CALL CONR(R,T,UCL,BCL)

Action : If $T \leq 1$ UCL = BCL = 0
If $T > 1$ UCL = the Upper 95% Confidence Limit
BCL = the Lower 95% Confidence Limit

Arguments : R = Real, input
T = Real, input
UCL = Real, output
BCL = Real, output

Routines Called : ALOG
EXP
SQRT

SUBROUTINE DBN (DB, X, F)

Purpose : To form a smoothed frequency distribution. Integer values are assumed evenly spread over a unit interval, and when the unit interval overlaps a category boundary the adjacent categories are credited with the corresponding fraction of the interval. This results in non-integral frequencies.

Usage : CALL DBN (DB, X, F)

DB = Array for the frequency distribution.
X = Value to be categorised.
F = ((Maximum value on New Scale +1)/
(Maximum value on Old Scale +1))

Arguments : DB = Real array, input and output
X = Real, input
F = Real, input

Restrictions : X must be positive and integral.
If F is greater than or equal to 1, X is left in its original grouping.

The DB Array in the calling program must have a dimension of at least 'Maximum value on New Scale +1'. This allows for zero scores.

Routines Called : None.

FUNCTION FNORM (P,A,S)

Purpose : Given the mean, A, the standard deviation, S, and the area under the Normal curve, P, solve for X, the Normal deviate,

the equation, $$P = \int_{-\infty}^{X} \frac{1}{\sqrt{2\pi S^2}} e^{-\frac{1}{2}\left[\left(\frac{x - A}{S}\right)^2\right]} dx$$

using the approximation given by Hastings in "Approximations for Digital Computers", on page 192. The maximum error is about .00045 standard deviations.

Usage : X = FNORM(P,A,S)

Arguments : P = Real, input
A = Real, input
S = Real, input

Routines Called : ALOG
SQRT

SUBROUTINE HIST (D, IE)

Purpose : To print a histogram

Usage : HIST (D, IE)
where D = a unidimensional array with values
to be plotted.
(the values need not be integers,
but must be in floating point).
IE = number of elements in D, (including 0)

Arguments : D = Real, input
IE = Integer, input

Routines Called : None.

SUBROUTINE INSLD (S, NORDS, R, NRANK, D)

Purpose : Invert a matrix S of order NORDS and, optionally, compute stepwise log determinants.

Action : R = Inverse of S unless S is singular, when R is the conditional inverse.
R (NORDS*(NORDS + 1)/2+1) onwards (i.e. directly after the half symmetric inverse), is used as working space for storing scale factors.

R = S is permissible.
D is a vector of NORDS stepwise log determinants.
If NRANK on entry
= -1, stepwise log determinants only are calculated
= -2, inverse only is calculated
= 0, both are calculated.
NRANK on exit is the rank of S. (>0)

Usage : CALL INSLD(S,NORDS,R,NRANK,D)

Arguments :

S	=	Real array, input
NORDS	=	Integer, input
R	=	Real array, output
NRANK	=	Integer, input and output
D	=	Real array, output.

Restrictions : R and S must be half symmetric in a unidimensional array. If the matrix is not positive definite or semidefinite the subroutine may break down. To allow for working space, the dimension of R must be at least NORDS x (NORDS +3)/2.

Routines Called : ISYM
SQRT.

SUBROUTINE MOUT(A,LA,NRA,NCA,RNAM,CNAM,NFORM)

Purpose : Print an NRA x NCA matrix A.

Usage : CALL MOUT(A,LA,NRA,NCA,RNAM,CNAM,NFORM)

Action : If LA = 1, A is stored in core as a half-symmetric matrix
≠ 1, A is a full matrix.

If NFORM = 0, output format is 8E14.7
= 1, output format is 8F14.4
RNAM = Row names
CNAM = Column names

The same rules as those applying to RH and CH in subroutine WMB are used.

Arguments :		
A	=	Real array, input
LA	=	Integer, input
NRA	=	Integer, input
NCA	=	Integer, input
RNAM	=	Double precision array, input
CNAM	=	Double precision array, input
NFORM	=	Integer, input.

Routines Called : INFRVN
INFCVN

FUNCTION MRSIG (R,N,IND)

Purpose : Test whether a multiple correlation coefficient,R, based on N cases and IND independent variables is significantly different from 0 at the 95% level, print a message and leave MRSIG = 1 or 0, depending on whether R is significant or not.

Usage : ISSIG = MRSIG(R,N,IND)

Arguments : R = Real, input
N = Integer, input
IND = Integer, input.

Routines Called : TANH
SQRT

3) ENTRY DRAW (X,Y,N,K,J,L)

Action : Plots a series of points whose co-ordinates are taken from the X and Y arrays. This can be done either by placing a specified symbol at each of the points required, or by joining successive points from the arrays by a line, or by combining the two methods i.e. using a line and symbols.

Usage : CALL DRAW (X,Y,N,K,J,L)

Arguments : X - Array of X-values
Y - Array of Y-values
N - No. of points to be plotted
K - Increment for choosing successive points from arrays
J - Indicates method of plotting required
J = 0, points joined by a line
J = 1, points joined by a line, and a symbol at all data points
J = 2, points joined by a line, and a symbol at every second data point.
ETC.
J =-N, a symbol produced at every Nth data point, lines between them suppressed.
L - Code for symbol to be used. (See Plotter codes on page B52.A. One of the first thirteen symbols may be used.)

All arguments input
Real arrays - X,Y
Integer - N,K,J,L

4) ENTRY LABEL (T,NT)

Action : Labels output with the title in array T. The label is written below the graph, parallel to the X-axis, and the printing is lined up to start where the X-axis starts.

Usage : CALL LABEL (T,NT)

Arguments : T-Array of characters for label (Real array,input)
NT- No. of characters in array T (Integer,input)

---

5) ENTRY NEXT

Action : Feeds paper to the end of the current plot and resets the reference point in preparation for the next graph.

Usage : CALL NEXT

---

6) ENTRY CLOSE(I)

Action : Feeds paper to the end of the current plot and writes the time, date and jobname parallel to the y-axis.

Usage : CALL CLOSE(I)

Argument : I - Indicates mode of call (Integer, input)
I = 0, Normal mode of call
I = 1, 'NEXT' has been called just prior to this call and so feeding of paper is to be suppressed.

Routines Called : The following installation supplied routines are called:
PLOTS, PLOT, SCALE, AXIS, LINE, LABEL, SYMBOL, PLOTT.

SUBROUTINE PRNTR (ARRAY, NVARS, NAR, NRODIM, AA)

Purpose : Prints several arrays in succeeding lines.

Usage : CALL PRNTR (ARRAY, NVARS, NAR, NRODIM, AA)

where ARRAY = two-dimensional array in which each column represents one of the arrays being printed.

NVARS = the number of variables or items in each single array.

NAR = the number of single arrays.

NRODIM = the size of a single array as given in the dimension statement for the calling programme.

AA = an array for storing the headers for the single arrays.

The output format is (A8, X, 12F9.3)

Note : The calling programme will require an equivalence statement making the single arrays columns of the complete array.

e.g. EQUIVALENCE (AVE(1), ARRAY(1,1)),
(SDS(1), ARRAY(1,2))

Arguments :

ARRAY	=	Real 2-dimensional array, input
NVARS	=	Integer, input
NAR	=	Integer, input
NRODIM	=	Integer, input
AA	=	REAL*8 array, input

Routines Called : None.

Note : A new entry, PRNTRK, allowing a choice of the source from which variable names are to be taken, has been added to the subroutine. See Page B70.A

SUBROUTINE PSAM(NDS,RXY,J)

Purpose : Print a half-symmetric matrix RXY with format options, taking variable names from the data set numbered NDS.

Usage : CALL PSAM(NDS,RXY,J)

Action : The half-symmetric matrix RXY is printed out.

If J = 0, the output format is 8G14.6
= 1, the output format is 19F6.2
= 2, the output format is 19(A2,4X)

Arguments : NDS = Integer, input
RXY = Real array, input
J = Integer, input

Restrictions : The data set NDS must have been opened.

Routines Called : None.

Note : A new entry, PSAMK, allowing a choice of the source from which variable names are to be taken, has been added.

See Page B70.A

SUBROUTINE PSYM(NDS,RXY)

Purpose : Print a half-symmetric matrix RXY in fixed format F6.2 taking row names from the data set numbered NDS.

Usage : CALL PSYM(NDS,RXY)

Arguments : NDS = Integer, input
RXY = Real array, input

Restrictions : Data set numbered NDS must have been opened.

Routines Called : None.

Note : A new entry, PSYMK, allowing a choice of the source from which variable names are to be taken, has been added.

See Page B70.A

Modifications to

SUBROUTINES PRNTR,PSAM,PSYM

These three subroutines have been modified to include additional entry points

PRNTRK(ARRAY,NV,NAR,NRODIM,AA,NAMES),
PSAMK(NV,RXY,J,NAMES),
PSYMK(NV,RXY,NAMES), respectively.

Purpose : Allows for greater flexibility with regard to the source from which the variable names printed for the arrays are taken.

```
Usage          :  REAL*8  NAMES(500)
                  CALL  PRNTRK(ARRAY,NV,NAR,NRODIM,AA,NAMES)
               or CALL  PSAMK(NV,RXY,J,NAMES)
               or CALL  PSYMK(NV,RXY,NAMES)
```

Action : The action when these entries are called is exactly the same as for the original subroutines. However there is now a choice of the source from which the variable names to be printed are taken.

For all three entries,

```
If NAMES =+N - Take names from rows of data
               set N (where N is a small posi-
               tive no.)
          -N - Take names from columns of data
               set N
        0.0D0 - Generate integer variable names
    otherwise - NV variable names should be
                  taken from the array NAMES
```

Note : that for PSAMK and PSYMK, NV is the number of variables to be printed and not a data set no. as NDS is in PSAM and PSYM.

Arguments : NAMES - Double precision array, input
NV - Integer, input
All other arguments are the same as for the original subroutines.

Routines Called : Addition of the entries means that the following routines are now called by the original subroutines, as well as by the entries:

```
IANAL  )
INFCVN )   all three subroutines
INFRVN )

EBCDIC )   PSAM,PSYM, but not PRNTR
INFNV  )
```

SUBROUTINE **RAPS** (R,NORDR,NFAC,TOL,MAXIT,B,RTS)

Purpose : Obtain NFAC latent roots and principal components of half-symmetric matrix R of size NORDR to a level of accuracy specified by TOL. No more than MAXIT iterations should be carried out for any root, and the roots should be placed in the array RTS and the vectors in the array B.

Usage : CALL RAPS (R, NORDR, NFAC, TOL, MAXIT, B, RTS)

Action : Hotelling's method is used.

The vector of latent roots is arranged in descending order of magnitude.

The latent column vectors are of length equal to the absolute value of their corresponding latent roots (principle components).

Information on number of iterations etc. is printed by the subroutine. To suppress this precede NORDR with a minus sign.

If an initial approximation to the component matrix is in B precede NFAC with a minus sign.

On exit R contains the residual matrix.

TOL is the value within which successive approximations to the roots must agree.

Arguments :

R	=	Real array, input and output
NORDR	=	Integer, input
NFAC	=	Integer, input
TOL	=	Real, input
MAXIT	=	Integer, input
B	=	Real array, output
RTS	=	Real array, output.

Restrictions : To allow for working space, B must be of dimension at least NORDR*(NFAC +1).

Routines Called : None.

SUBROUTINE SDLIM(VAR,TN,UCL,BCL)

Purpose	:	Compute 95% Confidence Limits for the standard deviation, given the variance VAR, from a sample of size TN using an approximation to the χ^2 distribution. (See Annals of Mathematical Statistics, vol. 17, 1946, p.220)
Usage	:	CALL SDLIM(VAR,TN,UCL,BCL)
Action	:	If TN $<$ 1, UCL = BCL = 0 If TN $\geq$ 1, UCL = the Upper 95% Confidence Limit BCL = the Lower 95% Confidence Limit

Arguments	:	VAR = Real, input
		TN = Real, input
		UCL = Real, output
		BCL = Real, output

Routines Called	:	SQRT.

SUBROUTINE SELECT(KSLECT,I,NSLECT, NAMES,NN)

Purpose : Used primarily when selection of variables for an output data set is required, the subroutine determines the identities of the variables to be selected.
It builds up an array, KSLECT, of I items (Variable Numbers), derived from parameter cards listing the variables required (by name or by position number).

Usage : REAL*8 NAMES(500)
CALL SELECT(KSLECT,I,NSLECT,NAMES,NN)

Action : KSLECT = An array containing the position nos. of the variables to be selected. It is built up by the subroutine on the basis of information read by the subroutine from variable specification cards.

I = Number of items required in KSLECT.

NSLECT = A parameter indicating whether variables are specified by position no. (NSLECT=1) or by name (NSLECT=2) on the variable specification cards.

NAMES = A double precision array of variable names used when required variables are specified by name. In order to determine the position nos. of the variables to be selected their names are compared with the names in this array. When a match is found the position no. of the name in array NAMES is entered as the position no. of the relevant variable in array KSLECT.

NN = Number of names in array NAMES.

The variable specification cards have the following format:

1. If variables are specified by position no. (i.e. NSELECT=1)

   COLS 1-3 1st field - Identity no. of 1st var. to be selected

   4-6 2nd field - Identity no. of 2nd var. to be selected

   etc.

   The end of the selection cards is indicated by a card with 999 in Cols. 1-3.

   If a set of consecutive variables is to be selected, give the first, followed by 0 in the next field, then the last.

2. If variables are specified by name (i.e. NSLECT=2)

   COLS 1 - May be used for card number

   2-9 - Name of 1st variable to be selected

   10 - Blank (or Minus, see below)

   11-18 - Name of 2nd variable to be selected

   etc.

   End of selection cards indicated by a card with ENDSELEC in Cols. 2-9.

   If a set of consecutive variables is to be selected, give the first and then the last, and put a minus in the column separating them (which would otherwise be blank).

Note : (1) When consecutive variables are specified the first and last variables must be on the same card.

(2) Variable specification cards need not be filled. The subroutine will take a new card if two consecutive 0's (for NSLECT=1)

or a blank space (for NSLECT=2) are encountered.

Arguments : KSLECT - Integer array, output
I - Integer, input
NSLECT - Integer, input
NAMES - Double precision array, input
NN - Integer, input

Routines Called : UNCLE.

SUBROUTINE SIGLV(X,SIG)

Purpose : Compute the One-Tailed Significance level
of a Normal deviate using
Hasting's Approximation. (See Approximations
for Digital Computers : Hastings, 2nd Printing,
1957, p. 186).

Usage : CALL SIGLV(X,SIG)

Action : Uses X = The Normal deviate.
Supplies SIG = Area under Normal curve from X
to infinity.

Arguments : X = Real, input
SIG = Real, output.

Routines Called: None.

SUBROUTINE SIGR(NDS,RXY,NC)

Purpose : Mark the correlations significant at the 5% level in a correlation matrix.

Usage : CALL SIGR(NDS,RXY,NC)

Action : Uses:

RXY = Correlation matrix in half-symmetric form.

NC = Number of cases used in calculating the correlation matrix.

NDS = Number of the data set from which row names(variable names) are to be taken.

The matrix RXY is printed with the correlations replaced by '+', '-' or '0' to show positive or negative significant or non-significant correlation.

Arguments : NDS = Integer, input
RXY = Real array, input
NC = Integer, input.

Routines Called : PSAM.

SUBROUTINE STACHI(C,D,P)

Purpose : Compute the Significance of a X^2 (Chi-Square), using an approximate formula from N.C. Severo and M. Zelen, Biometrika, 47,(1960),pages 411-416.

Usage : CALL STACHI (C,D,P).

Action : Uses:

C = Value of X^2 (Chi-Square).
D = Number of degrees of freedom

Gives:
If $D \leq 0$, P = 1.0
> 0, P = Significance of X^2 (Chi-Square)

Arguments : C = Real, input
D = Real, input
P = Real, output

Routines Called : SIGLV.

SUBROUTINE WILH (F, D1, D2, SIG)

Purpose : To compute the significance of an F-ratio, F, with numerator degrees of freedom D1 and denominator degrees of freedom D2. Significance level is returned in SIG.

Usage : CALL WILH (F, D1, D2, SIG)

Action : Subroutine normalises the F-ratio, and passes the result to SIGLV.

Arguments : F = Real, input
D1 = Real, input
D2 = Real, input
SIG = Real, output.

Routines Called : SIGLV.

V. Revised List of Package UNCLE error messages:-

No.	Routines	Meaning
1.00	CLSALL	
4.00	EBCDIC	
9.00	INFCN	
9.01		NDS out of range
9.02		Column names not available
10.00	INFCVN	
10.01		Column names not available
10.02		NDS out of range
12.00	IN	
12.01		NDS out of range
12.02		NDS already open
12.03		IP differs on re-reading a data set
12.04		End-of-file or error in reading headers
12.05		File identification incorrect
13.00	INFIP	
13.01		NDS is output file
13.02		NDS is not opened
13.03		NDS is out of range
15.00	INFNV	
15.01		File not opened
15.02		NDS out of range
16.00	INFRN	
16.01		File not opened
16.02		NDS out of range
17.00	INFRVN	
17.01		NDS less than 0
17.02		NDS out of range
30.00	OUT	
30.01		NDS out of range
30.03		Negative or zero no. of variables in IP array
30.04		Row names to be copied but information not available
30.05		NDS already open.

List of Package UNCLE error messages (continued):-

32.00	RD	
32.01		File closed
32.02		File not opened
32.03		File open for output
32.05		Premature end-of-file
32.06		Permanent read error
32.07		NDS out of range
33.00	RMB	
33.01		LA too large
33.02		Premature end-of-file in reading user specified H.S. matrix
33.03		Premature end-of-file in reading diagonal to half-symmetric conversion
33.04		IP(2) Parameter gives incorrect no. of cases. (No. is too small)
33.06		Attempt to form diagonal matrix from rectangular matrix of unacceptable dimensions
34.00	RMBF	
39.00	TAG	
42.00	WD,WDI	
42.01		NDS out of range
42.02		NDS opened for input
42.03		File not opened
43.00	WEF	
43.00		NDS negative
43.01		NDS too large
43.02		File not opened, or file opened for input
44.00	WMB	
44.00		NDS negative
45.00	WMBF	
46.00	CLSINP	
46.01		Data set was not completely read
48.00	INFNC	
48.01		File not opened
48.02		NDS out of range
51.00	MISS	
53.00	RDFS	

List of Package UNCLE error messages (continued):-

54.00	INTERP	
61.00	INICM	
62.00	GETCM	
62.01		Illegal branches
62.02		No more space in COMMON
63.00	FREECM	
63.01		Area to be freed overlaps area already free, or extends beyond COMMON
63.02		Impossible branches
63.03		Area to be freed is below dynamic area, or negative length of area
67.00	INFJOB	
72.00	INOUT	
72.01		NDS out of range
72.02		NDS already opened
72.03		IP differs on re-reading a data set
72.04		End-of-file or error in reading headers
72.05		File identification incorrect
72.06		IP(1) not equal to no. of variables expected
72.07		End-of-file or error in reading preliminary cases.