DECUS
PROGRAM LIBRARY

DECUS NO.	8-564
TITLE	A STATISTICAL SYSTEM IN PS/8
AUTHOR	Jens G. Rosenkrantz, M.D.
COMPANY	Childrens Hospital of Los Angeles Los Angeles, California
DATE	July 6, 1972
SOURCE LANGUAGE	PS/8 FORTRAN (FORTRAN II)

SUMMARY

This paper describes a system, built upon DEC's PS/8, which runs a number of programs to do statistical analyses of data. The programs are written in PS/8 Fortran (Fortran II) and require the following hardware: PDP 8 computer with at least 8 K of core and a disc or dectape. The following programs are provided: (i) Mean-Variance; (ii) Student's t test; (iii) Rank analysis; (iv) Analysis of Variance; (v) Correlation; and (vi) Chi Square. Additional programs can be easily added to the system.

Data may be given from a variety of input devices: (i) teletype; (ii) high speed paper tape reader; (iii) card reader; (iv) any dectape unit; or (v) the system device (disc or dectape). Answers may be written, at the user's option, on the teletype, the high speed punch, or on a line printer, and are formatted on 11 inch "pages". A large number of data files may be chained together by the user in order to permit batching of data, each file calling a particular statistical test. Thus the system can function as a desk calculator (with teletype input) or can process large batches of data unattended.

Supported by NIH Grant # HE 12973.

DEFINITIONS

1. Datum or Data Point = a single entry to be included in the statistical analysis.
2. Group = a collection of data in a single statistical sample.
3. Set = a collection of groups to be analyzed by a single statistical test.
4. File = a collection of sets each to be analyzed by a single statistical test.
5. Batch = a collection of files to be run "at one sitting" and employing a variety of statistical tests.

RUNNING THE SYSTEM

I. Preparation of a data file:

1. Prepare Data Tape for a single file as follows:

a. Using PS/8 Editor, for each set in the file, give the following:

# of groups in the set (integer number up to 2047) CARRIAGE RETURN; # of data points in the first group (integer number up to 2047) CARRIAGE RETURN; 1st Data Point of first group (-999.999 to +9999.999) CARRIAGE RETURN; 2nd Data Point of first group; etc.; # data points in 2nd group; etc., until finished with set. If you want a data point to represent a blank entry (and the statistical test you wish to use can accept blank entries) let it = 11111. Don't forget the decimal point in all data entries.

b. For a subsequent set in the same file, repeat the process above.

c. When you are finished with the last set in the file, let the next integer number = Ø.

d. If you wish to use the file for input from the low speed or high speed paper tape reader or the card reader, this is all that is necessary, and the file can be closed. If the file is to be read on a dectape transport or disc, you must add another number after the Ø indicated above. This is the program number which you desire to use for the file following the one you are preparing and corresponds to the program contents of the system: 1 = Mean-Variance; 2 = Student's t; 3 = Rank; 4 = Variance; 5 = Correlation; 6 = Rank Correlation; 7 = Chi Square. In this manner you can chain together various files using different statistical tests to form one batch. If you wish the present file to be the last one in the batch, put Ø in lieu of the program number.

e. Naming the file (dectape and disc files only). The file must be named according to the scheme: XNMM.DA, where XN is the identifier for the statistical test to be run on the file, MM is the number of the file among all files in the batch using that particular test, and

.DA is the extension to enable the PS/8 Fortran System to find the file on the specified device.

The identifier (XN) for the statistical test is coded as follows: A1 = Mean-Variance; A2 = Student's t; A3 = Rank Test; A4 = Variance; A5 = Correlation; A6 = Rank Correlation; A7 = Chi Square.

The number of the file (MM) is coded from Ø1 to 99. For different files in a given batch using the same identifier, these numbers must be consecutive, beginning (in each batch) with Ø1. The system will not skip over a number; so if you skip one, it will search in vain for the missing one and will exit with an error message.

As examples of the above:

A2Ø1.DA is the first file using Student's t in a given batch. A5Ø3.DA is the 3rd file using Correlation in the batch (the batch must include A5Ø1.DA & A5Ø2.DA).

2. Running a test:

   a. Set up your data input file(s) as above on the input device you select.

   b. Type "R STATIS".

   c. Answer the questions appropriately.

   d. If the program types "↑", answer by striking any teletype key.

   e. The system will run automatically from there.

EXAMPLES OF CONVERSATION:

1. WITH CONTENTS:

```
.R STATIS

STATIS: 10- 1-72   CONTENTS? (Y OR N)                (USER ANSWERS "Y")
1)MEAN-VARIANCE
2)STUDENTS T
3)RANK TEST
4)VARIANCE ANALYSIS
5)CORR COEFF
6)RANK CORRELATION
7)CHI SQUARE

# OF PRGM:    5                                      (USER ANSWERS "5")
OPTION TO PRGM #:
 5) IS STUDENTS T
 6) IS RANK TEST
OPTION? (Y OR N): Y                                  (USER ANSWERS "Y")

INPUT DEVICES:
1) TTY
2) HSR
3) CARD READER
4) SYS
5) DECTAPE

INPUT DEV #: 5                                       (USER ANSWERS "5")
DTA #:       3                                       (USER ANSWERS "3")

OUTPUT DEVICES:
1) TTY
2) HSP
3) LINE PRINTER

OUTPUT DEV #:1                                       (USER ANSWERS "1")


SUMMARIZE? (Y OR N)                                  (USER ANSWERS "Y")
PROGRAM:        CORRELATION: STUDENTS T: Y
INPUT DEV:      DECTAPE #:    3
OUTPUT DEV:     KEYBOARD
OK? (Y OR N)                                         (USER ANSWERS "Y")
```

```
CORRELATION:
10- 1-72
```

2. WITH NO CONTENTS:

```
.R STATIS

STATIS: 10- 1-72   CONTENTS? (Y OR N)          (USER ANSWERS "N")
# OF PRGM:    5                                (USER ANSWERS "5")
OPTION? (Y OR N): Y                            (USER ANSWERS "Y")
INPUT DEV #: 5                                 (USER ANSWERS "5")
DTA #:       3                                 (USER ANSWERS "3")
OUTPUT DEV #:1                                 (USER ANSWERS "1")


SUMMARIZE? (Y OR N)                            (USER ANSWERS "N")
```

CORRELATION:
10- 1-72

SAMPLE DATA TAPE:
(SUITABLE FOR MEAN-VARIANCE PRGM)

```
1                              (# GROUPS IN 1ST SET)
2                              (# DATA POINTS IN 1ST GROUP)
2.
4.
1                              (# GROUPS IN 2ND SET)
8                              (# DATA PTS IN 1ST GROUP)
18.
64.
92.
100.
16.24
18.
42.
1.247
0                              (END OF SET)
0                              (SIGN OFF AFTER THIS SET)

#
```

```
MEAN-VARIANCE:
10- 1-72

FILE #:   1      SET #:   1

    2.000      4.000

    MEAN=      3.000
    ST DEV=    1.414
    ST ERR=    1.000
    N=         2

FILE #:   1      SET #:   2

   18.000     64.000     92.000    100.000     16.240
   18.000     42.000      1.247

    MEAN=     43.936
    ST DEV=   37.406
    ST ERR=   13.225
    N=         8
```

DETAILED DESCRIPTION

(Note: Understanding this description and the source programs requires knowledge of DEC's PS/8 Fortran System and of the SABR language).

I. Controlling and Utility Programs: STATIS, STATØ, FRMAT, LINFED, TABLE.

These programs set up the parameters of the system, read data from the input device onto the system device, handle page formatting for the answers, and prepare statistical tables for use by the system.

A. STATIS:

This program, called by the command "R STATIS" to the PS/8 keyboard monitor, begins the operation of the system. It obtains the following information: 1) the date - from its storage in the PS/8 system; 2) the first statistical test to be run; 3) the input device to be used for the batch; and 4) the output device to be used for all the answers. It also asks whether any optional tests are to be run. Placing this information in common storage, it chains to the controlling section, STATØ.

STATIS, after initializing, queries the system for the date (which was written on the system device by the DATE command of the PS/8 system) and stores the day, month and year in the common storage location, ID, MO, and IY. The program then asks (on the teletype) if the user wants the

"contents" printed out on the teletype before each subsequent question. If so, it lists the alternative answers possible before asking each of the four questions to which it requires answers. If not, it skips this listing of alternatives and merely asks for the number of the answer. STATIS asks first for the number of the program to be run on the first data file (or tape). The user answers with the number of the statistical test desired. Currently the contents are: 1. Mean-Variance; 2. Student's t Test; 3. Rank Analysis; 4. Analysis of Variance; 5. Correlation; 6. Rank Correlation; and 7. Chi Square. The answer is stored in the common storage location, NPRG, having been given by the user as an integer number. Inappropriate answers cause the listing of the contents and repetition of the query. STATIS then asks: "OPTION (Y or N)." Two of the current tests have optional parts (Correlation with optimal Student's t and Rank Correlation with optional Rank Test). Answering "Y" to this question puts a Ø in the common storage location, IOPT, and causes the option to be used whenever Correlation. or Rank Correlation is later used. Any other answer then "Y" is equivalent to "No", and the option is never used in that batch.

Next, STATIS asks for the number of the input device for data transmission, according to the following: 1 = teletype; 2 = high speed reader; 3 = card reader; 4 = system device; 5 = dectape. In the event that the

dectape answer is chosen, the program asks for the particular dectape number. This information is stored in the common locations, INPUT and JDEV, and is used by the controlling program, STATØ, to make up its various READ statements and CALL IOPEN statements for input of data. STATIS then asks for the output device number: 1 = teletype; 2 = high speed punch; and 3 = line printer. This information is stored in common storage (IOUT) and used by all subsequent programs in WRITE statements, for the output of answers.

STATIS then asks "SUMMARIZE? (Y or N)". If the user answers "Y", the initial program selected, the option, if appropriate, and the input and output devices selected are summarized on the teletype and the question "OK? (Y or N)" is asked. The answer "Y" causes STATIS to chain to the program, STATØ. Any other answer restarts STATIS. Any answer other than "Y" to the "SUMMARIZE?" question causes STATIS to chain to STATØ.

Common storage locations used by STATIS, in addition to those already described, are LIN and NFIL(I). The former keeps track of the number of lines left on an 11 inch page of teletype output and the latter is the number of times each of the seven statistical programs have been used in the batch. Both are initialized by STATIS and kept current by subsequent programs in the system. The information in common storage, passed on to the controlling program, STATØ,

must not be lost by subsequent programs in the system. Thus all run-time programs must have, as a minimum, the common storage possessed by STATIS.

Additional statistical tests may be added to the system, up to a maximum of 99. STATIS must be altered to include the additional programs' names and numbers in its contents, to allow the additional answers (presently any answer over 7 to the question "# of program" causes repetition of the question), to dimension the location NFIL(I) appropriately and to set all NFIL(I) = Ø in the initialization of the system.

B. The Controlling Program, STATØ:

This program is the keystone of the entire system: it reads the data, writes it in a file on the system device, and chains to subsequent programs, handling the running of the various statistical tests selected by the user. It is entered by one of two routes: either from STATIS on the first statistical test selected by the user or at the completion of any of the statistical programs.

Input data format:

The system accepts for processing data in the range of -999.999 to +9999.999, up to three decimal places. Any number more negative than the lower limit causes an error message to be typed and leads to inaccuracies in the answers given by the system. Should the number be -10,000 or more negative, STATØ fails, with a series of error messages given

by the Floating Point Library on the teletype, and the eventual outcome of the run is unpredictable. A similar course of action follows input of data at +100,000 or greater. This occurs because STATØ reads the data in F9.3 format. Negative numbers between -1000 and -9999.999 cause the Statis System to print the error message "INPUT ERROR" on whatever output device is selected for the answers. This message appears during output of data and is a warning that answers related to that data point are inaccurate. However the system continues to run, so that later data can be accurately analyzed. Positive numbers in the range, +10,000 to 99,999.999 are read by STATØ, but interpreted as blank entries by certain statistical tests (such as Correlation) and cause the message "INPUT ERROR" on other tests where blank entries are not appropriate.

The system allows the inclusion of up to 2047 data points in a group. The statistical system actually consists of two parallel systems: one to process limited data and another (slower) to process large amounts of data. If the size of a group exceeds 50 data points, the slower large data system is chosen by STATØ. With 50 or less data points in a group, STATØ attempts to use the faster limited data system, subject to the constraints of the next paragraph.

The system allows the inclusion of up to 2047 groups in a single set. If the number of groups in a set is over 7, the large data system is selected by STATØ for subsequent processing of the data.

Should the input device be the teletype, the high speed reader or the card reader, only one file is permitted in a given batch. If dectape or disc is selected as the input device, then more than one file is permitted in a batch (up to 99 files calling each of the 7 statistical tests). When any of the statistical tests is called for the 100th time, the error message "TOO MANY DATA TAPES IN PRGM #X" is printed, the system calls the PS/8 monitor and refuses to process any more files in the batch. Files written on dectape or disc are numbered as follows: (i) a single letter - single decimal number combination corresponding to the statistical test to be run on the file (A1 = Mean-Variance; A2 = Student's t; A3 = Rank Analysis; A4 = Analysis of Variance; A5 = Correlation; A6 = Rank Correlation; A7 = Chi Square; with any additional statistical test identifiers following this scheme, as the system is enlarged); and (ii) a two-digit decimal number identifying the file number for a given statistical test. Thus, for example, the dectape or disc file A423.DA would be the 23rd file calling the Analysis of Variance program. The file extension .DA is required on all dectape or disc input files under the PS/8 Fortran System.

STATØ expects data input on the selected device to be in ASC II format according to the following scheme: An integer number (I4 format) giving the number of groups in the first set of the file (NG); a second integer number (I4 format)

giving the number of data points in the first group of the set (N); exactly N data points in floating format; an integer number giving the number of data points in the second group, and so forth until exactly NG groups have been coded for the first set. The coding is repeated for subsequent sets on a file. When all sets on a file have been coded, the user adds Ø as the next NG entry and STATØ recognizes that the file has been completely processed. With teletype, tape or card reader input devices, STATØ, on encountering this Ø as it reads the file, types "FINISHED" and control returns to the keyboard monitor of the PS/8 system. With dectape or disc input files, the system examines the next entry after the NG = Ø entry, picks up the integer number which the user had entered there (I4 format) and uses this as the next statistical test number to be run. For example, the entry 4 after NG = Ø on a file will cause the system, after giving the answers on the current file, to search for the next A4 file on the input device and to run Analysis of Variance on those data. In this manner a large number of files and statistical tests (up to 99 files for any given statistical test) can be chained together for a single batch run.

Operation of STATØ:

STATØ first types a series of dashes to mark the top of the first output page, sets the common storage, LIN, to 55 (to give 55 typed lines per teletype output page) and then

types the name of the statistical test to be run and the date (obtaining this information from common storage). It then branches according to the input device selected by STATIS. If INPUT = 1 (teletype input) STATØ waits for input from the teletype, and echoes the input data during transmission. If INPUT is 2 or 3 (high speed reader or card reader), STATØ types a "↑" and waits for any key of the teletype to be struck before reading data. If INPUT is 4 or 5 (system device or dectape) STATØ bootstraps information from the common storage locations JDEV (to get DTA number), INPUT (to get "SYS" or "DTA"), NPRG (program number to get letter-number statistical test identifier into file-name) and NFIL (NPRG) (to get appropriate file number into file name). These are bootstrapped into a CALL IOPEN ('ØØØØ', DØØØ') with STATØ writing 'SYS' or 'DTA#' in the first area and the file-name (e.g., 'A423') in the second. Thus the exact file wanted is opened on the device specified by STATIS. Reading of data is accomplished by setting the location INP to INPUT for INPUT = 1,2,3 or to 4 for INPUT = 4 or 5. STATIS interprets dectape #0 or #8 as "system" and sets INPUT = 4 for these answers. All READ calls in STATØ are then written as READ (INP, Format #), giving STATØ its versatility in input devices. Subsequent passes through STATØ, after a file has been processed, use the same input device; so all data in a given batch must be read on the same input device.

STATØ then reads data from a single file onto the system device into a file named DATA.DA. All data from the file are written, integer numbers in A2 format and floating numbers in A6 format. At the conclusion of this writing, after the NG = Ø entry, STATØ writes an additional Ø if device 1, 2, or 3 was selected or reads and writes the next integer number for disc or dectape. This number is used to chain to the next statistical test following the processing of the current test (see Reader section). It is set to Ø for non-file structured input devices in order to admit only one file in a single batch for these devices.

During reading of data, STATØ checks the various NG and N integral entries. If on a given file a single NG entry (number of groups in a set) or N entry (number of data points in a group) exceeds 7 or 50 respectively, then the common storage JDAT is set to 2; otherwise it is set to 1. JDAT = 2 indicates that the large data system is to be used for that file; JDAT = 1 is a switch to use the limited data system for processing that file. The reason for the two systems is simply this: the various programs can run relatively swiftly by reading and processing data into a dimensioned buffer in common storage. However, the size of the buffer is constrained by the core available to the user. As a reasonable compromise, with 8K of core,

a 50 x 7 data matrix was chosen as the maximum permitted in the limited data system. This allows most data files to be processed as rapidly as the PS/8 Fortran system and output device allows. Also several programs type back data in a matrix in which there are as many columns as groups in a set. Because of the limitation of teletype output, 7 groups seemed the maximum which could be so typed by the teletype and remain intelligible to the user. However, to enable the user to process the occasional file which exceeds these limitations, a second (large data) system was written parallel to the limited data system. This large data system can process files with up to 2047 groups in a set or up to 2047 data points in a group, but sacrifices speed for this increased capability. The difference in time is considerable for large collections of data in a given group or set. STATØ selects the system to be used for a given file and is capable of changing this system on subsequent files.

Once the data has been written on the system, STATØ bootstraps CALL CHAIN ('RDRØ') to one of three programs: (i) RDR1 for all programs of the limited data system; (ii) RDR2 for large data system and those statistical tests which type data in separate groups (Mean-Variance, Student's t, Rank Test, and Variance Analysis), or (iii) RDM2 for large data system and those tests which type data as a matrix (Correlation, Rank Correlation, and

and Chi Square). Please see the Reader section for further description of this output formatting.

Finally STATØ sets a common storage switch (MULT) to Ø. This switch can be used for multisectioned programs to control the entry point of the initial section of the program, or can be employed as a switch during program execution.

Alterations:

(i) To add new statistical tests: Increase the dimension of NFIL(I) appropriately. Add format statements giving the name(s) of the additional statistical tests and adjust the conditional GOTO statement before statement #120 and add the appropriate WRITE (IOUT,#) instruction(s).

(ii) To disable the large data system: The user with a dectape PS/8 system may well not want this capability. Change the IF (JDAT) instruction preceding statement #420 appropriately to give an error message and return to the Monitor (GOTO 1000). Or leave it as is, omit RDR2.SV and RDM2.SV from the system and accept the "CHER" message which the PS/8 system types out. Or, if you wish the system to to continue, change the IF (JDAT) instruction, so that STATØ types an appropriate error message and then picks up (from location NUM) the next test

a 50 x 7 data matrix was chosen as the maximum permitted in the limited data system. This allows most data files to be processed as rapidly as the PS/8 Fortran system and output device allows. Also several programs type back data in a matrix in which there are as many columns as groups in a set. Because of the limitation of teletype output, 7 groups seemed the maximum which could be so typed by the teletype and remain intelligible to the user. However, to enable the user to process the occasional file which exceeds these limitations, a second (large data) system was written parallel to the limited data system. This large data system can process files with up to 2047 groups in a set or up to 2047 data points in a group, but sacrifices speed for this increased capability. The difference in time is considerable for large collections of data in a given group or set. STATØ selects the system to be used for a given file and is capable of changing this system on subsequent files.

Once the data has been written on the system, STATØ bootstraps CALL CHAIN ('RDRØ') to one of three programs: (i) RDR1 for all programs of the limited data system; (ii) RDR2 for large data system and those statistical tests which type data in separate groups (Mean-Variance, Student's t, Rank Test, and Variance Analysis), or (iii) RDM2 for large data system and those tests which type data as a matrix (Correlation, Rank Correlation, and

and Chi Square). Please see the Reader section for further description of this output formatting.

Finally STATØ sets a common storage switch (MULT) to Ø. This switch can be used for multisectioned programs to control the entry point of the initial section of the program, or can be employed as a switch during program execution.

Alterations:

(i) To add new statistical tests: Increase the dimension of NFIL(I) appropriately. Add format statements giving the name(s) of the additional statistical tests and adjust the conditional GOTO statement before statement #120 and add the appropriate WRITE (IOUT,#) instruction(s).

(ii) To disable the large data system: The user with a dectape PS/8 system may well not want this capability. Change the IF (JDAT) instruction preceding statement #420 appropriately to give an error message and return to the Monitor (GOTO 1000). Or leave it as is, omit RDR2.SV and RDM2.SV from the system and accept the "CHER" message which the PS/8 system types out. Or, if you wish the system to to continue, change the IF (JDAT) instruction, so that STATØ types an appropriate error message and then picks up (from location NUM) the next test

number, places it in NPRG and recycles to read the next file.

C. Page Formatter: FRMAT:

Output from the teletype or line printer is formatted into 11 inch pages, typed between dotted lines. This is accomplished by keeping track of the number of lines left on a page in the common register, LIN. Whenever one or more lines are typed, the subroutine FRMAT is called. This subroutine updates LIN and, if LIN = Ø or less, gives sufficient line-feed instructions to finish off the page. It then types a dotted line and enough line-feeds to margin the top of the next page and resets LIN to 55. In addition, FRMAT must be called prior to each CALL CHAIN (or the equivalent instruction, "LIN = LIN-1"), since the PS/8 system with teletype output gives a single line-feed to the teletype each time CALL CHAIN is used.

Likewise, each statistical test program, when finished, executes a CALL FRMAT (55) to "turn the page" and resets LIN = LIN+54 to reset the line number. Thus a subsequent file is typed out, beginning on a new page. STATØ accomplishes the same task directly and gives a blank output page one line longer than the 11 inch pages to act as a separator between files.

Alterations:

If a different length page is desired on output, change all the 55's in the entire system (using Editor and the buffer

character-string search command) to the appropriate number.

D. Page Format Compensator: LINFED:

If the output device is other than the teletype, the automatic line-feed occurring with CALL CHAIN does not occur. To compensate for this, each program prior to chaining to another calls the subroutine, LINFED. This subroutine inserts a line-feed instruction to the output device if it is other than teletype, and does nothing in the event that the teletype was selected as the output device.

E. TABLE:

This program is used in setting up the system, but is not used at run-time.

Six of the seven programs furnished have the capability of looking up their answers in a statistical table. For example, the program, Student's t, calculates the T value and degrees of freedom corresponding to the difference in means between two groups in a set. It then looks up the corresponding probability that this T value or larger values may be due to chance and types out whether this probability (P) is greater than or equal to .Ø5, between .Ø1 and .Ø5, or less than or equal to .Ø1. The program accomplishes this by reading its table from the system device. The program, TABLE, is employed in writing

the table on the system device. Corresponding to each of the appropriate statistical tests, paper tape is provided which summarizes a statistical probability table in ASC II format. The entries on each tape provide the following information: 1) the name of the table; 2) the number of entries on the table; 3) the values of the statistical parameter corresponding to P = .Ø5 and P = .Ø1 for each of the various degrees of freedom. For example, with program #2 (student's t), the tape TTAB.DA is provided. This tape has on its first line the phrase, "TTAB", on its second line the number 42 (42 subsequent entries) and then contains on subsequent lines the numbers 12.7Ø6 - 3 spaces - 63.657, (carriage return) Ø4.3Ø3 - 3 spaces - Ø9.925, etc. On each line the first number represents the value of T corresponding to P = .Ø5 for that degree of freedom and the second number the value of T corresponding to P = .Ø1. The program, TABLE, is used to read this tape or a dectape, disc or card-reader copy of it, and write it in Fortran, A2 and A6 format on a system file called TTABLE.DA. This file is read by program #2 whenever a Student's t test is run.

The program asks the reader which input device he wishes for preparation of the table: (teletype, high speed paper tape reader, card reader, system device, or a given dectape transport unit). In a manner similar to STATØ it then reads the table into a file on the system device. The table's name

is made up of the first four letters of the first line of the table plus the extension "LE.DA" to make up the word "XTABLE.DA" (for the tape, the first line of which reads XTAB). If the input (ASCii) table is on the system device or dectape, it must be named XTAB.DA or RTAB.DA, etc., so that it can be found.

Alterations:

When adding a new table, a rigid format must be followed. The first line gives a 4 letter identifier to the table, which must be unique and must be used (along with the ...LE.DA extension) in the statistical program reading the table. The second line (in I5 integer format) gives the number of entry lines in the table. Each entry line consists of two F6.3 numbers corresponding to the value for P = .05 and P = .01, the 5% value being given first. All zeros must be typed for these numbers (including leading zeros) and the two numbers must be separated by exactly 3 blank spaces. Use of the TAB key is not permitted. In the event that all numbers in a given table do not fit into the format, F6.3, multiples or fractions of the true entry can be substituted provided the statistical program reading the table knows which entries are thus altered and can compensate for the alteration. The calculation of which line to read (corresponding to the proper degree of freedom or entry number) is the responsibility of the statistical program reading the table.

II. Data Reading Section: RDR1, RDR2, RDM2

These programs read data from the file, DATA.DA, prepared by STATØ, and write the data and statistical parameters of each group on the output device (see under Mean-Variance for further details). RDR1 performs this task for the limited data system; for large data system, RDR2 and RDM2 are used.

RDR1 increments the set number (NSET) and then reads data into the matrix, A(I,J) from the proper set of DATA.DA. As it does this, it calculates mean and standard deviation of each group. Output is in one of two formats, depending on the statistical test chosen. For those tests in which the data are not considered as ordered pairs, data are put out by groups (column format). These tests include Mean-Variance, Student's t, Rank Test and Variance Analysis. For those tests calculating from ordered pairs of entries from two groups (and allowing the possibility of blank entries), the output is in matrix format, with the groups of data arranged as columns of the matrix. To acomplish this, it is necessary for RDR1 to calculate the maximum number of entries in the groups of the set. It does this by reading DATA.DA through the proper set to compute the maximum group size (MAX). Then on a second pass reading DATA.DA it accepts data into buffer, filling out each group with blank entries when the actual number of entries is less

than maximum (a blank entry is one over +1Ø,ØØØ). This process of calculating MAX and writing blanks is not done for those statistical tests written in column format. For those programs, the reader gives the message, "INPUT ERROR", for entries over 1Ø,ØØØ. A similar message is given for numbers more negative than -1000.

For a given set the first entry (NG) represents the number of groups in the set. If NG = Ø, RDR1 interprets the next integer in DATA.DA as the number of the next test to be run, storing it in the common location NPRG and chaining back to STATØ.

RDR2 and RDM2 perform the same tasks in the large data system. RDM2 reads DATA.DA for programs in matrix output, calculating NG (the number of groups in the set) and MAX (the size of the largest group) on the first pass, and reading the floating numbers of the set into a NG x MAX matrix, called MATRIX.DA, on the system device. RDM2 then transposes this matrix and chains to RDR2. The transposition of the matrix can be quite time-consuming for sets where MAX is over 300. RDR2 is used to read data for those tests in which output is in column format and writes output for both types of tests.

A. Individual Programs for Statistical Analysis:

1. Mean-Variance; Program #1; incorporated in the reading and typing section:

This program accepts data only if there are no blanks, with the first entry of the data tape (NG) equal to 1. As many sets as desired may be put on a single data tape. In the large data system, the maximum number of entries in the group is 2047. The output of data, mean, standard deviation, standard error and number of entries is bunched by group.

The parameters are calculated as follows:

$$\bar{X} = \sum_{i=1}^{N} X_i / N \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (1)$$

$$SD = \left[\frac{1}{N-1} \cdot \sum_{i=1}^{N} (X_i - \bar{X})^2 \right]^{1/2} \quad \ldots\ldots\ldots \quad (2)$$

$$SE = SD / \sqrt{N} \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad (3)$$

where:

i) N = number of data in group

ii) X_i = individual datum

iii) $\bar{X}$ = mean

iv) SD = standard deviation

v) SE = standard error

2. Student's t; Program #2; STUD1; STUD2; TTABLE:

This program is a Student's t test[1], which processes data only if there are no blanks and writes data and parameters for each set in column format. As many sets as desired may be put on a single data tape. In the large data system, the maximum number of entries in a single group is each 2Ø47. Output is as follows: 1) the data, means, standard deviations, standard errors, and numbers of entries for each group in the set-output by the Reader section of the system; 2) for each pair of groups in the set, the T-value by Student's t test and the degrees of freedom for the pair, together with the T-value for that degree of freedom which corresponds to P = .Ø5 and P = .Ø1 and a statement about the relationship of the probability to these limits. This information relative to probability is obtained from a table of probabilities corresponding to values of T.[2]

The formulae for calculating T and degrees of freedom are as follows:

$$T = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{n_1 s_1^2 + n_2 s_2^2}} \cdot \left[\frac{n_1 n_2 (n_1 + n_2) - 2}{n_1 + n_2} \right]^{1/2} \quad \ldots \quad (4)$$

$$DF = n_1 + n_2 - 2 \quad \ldots \quad (5)$$

where:

i) $\bar{X}_1$ and $\bar{X}_2$ = means of group 1 and group 2 respectively.

ii) n_1 and n_2 = numbers of entries in groups 1 and 2 respectively.

iii) s_1 and s_2 = standard deviations of group 1 and group 2 respectively.

iv) T = student's t value

v) DF = degrees of freedom,

where these parameters were calculated by Eqs. (1) and (2).

The limited data system (STUD1 and TTABLE) are called by RDR1 and calculate the T value and degrees of freedom from information left in core by RDR1. STUD1 does the calculation and types, for each pair of groups, the group numbers, the T value and the degrees of freedom for the pair. It then looks up on the system device, file TTABLE.DA, the critical T values corresponding to P = .Ø5 and P = .Ø1 for that particular degree of freedom and types these values. Finally STUD1 compares the T value with those critical T values and types out "P > .Ø5", "P = .05", "Ø1 < P < .Ø5", "P = .01", or "P < .Ø1" according to this comparison. When finished with this pair, STUD1 continues with subsequent pairs. The order of processing pairs of groups in a set is as follows: Group 1 vs 2; Group 1 vs 3.... Group 1 vs Last; Group 2 vs 3; Group 2 vs 4, etc.... When finished with this process, STUD1 "turns the page" (i.e., provides sufficient

line-feeds to finish the page) and chains to RDR1 for reading and processing of the next set.

STUD1 has a section which, examining the switch MULT, allows it to chain to any program which calls it. In the present system the two programs which call STUD1 in addition to RDR1 are STUD2 and CORR1, and the methods for operating the switch MULT are indicated in the description of each such calling program. When STUD1 is called by RDR1, MULT is set at +1 which causes STUD1 to loop back and compare all pairs of groups in a set and finally to chain to RDR1.

The large data program (STUD2) reads the data from the systems device file DATA.DA and calculates T and the degrees of freedom for a pair of groups, storing these in the registers R and NØ. The group numbers are stored in registers J1 and J2. It then sets MULT = Ø and chains to the output section of STUD1. STUD1 runs through its output section, typing the group numbers with the pair, the T value and degrees of freedom and the probability data as in the paragraph above. With MULT set to Ø, STUD1 then bootstraps in the name of STUD2 (using information stored in the registers JDAT and NPRG) and chains back to STUD2. With MULT still set to Ø, STUD2 increments the appropriate group numbers, tests to see if it is finished with the set and, if not, reads the data in for the appropriate group(s) and again calculates J1, J2, R and NØ for this pair. The process of calling STUD1 is repeated until all pairs have been examined. STUD2 then calls RDR2, after "turning the page".

3. Rank analysis; Program #3, RANK 11, RANK 12, RANK 2, RTAB.DA:

These programs give a Wilcoxon-White Two Sample Rank Test, with ties considered.[3] In the large data system the maximum number of groups accepted is 2Ø47, and the maximum number of entries per group 3ØØ. The test arranges each of the test groups in ascending order and then compares entries in the lower numbered group with entries in the group compared with it, keeping cumulative count of the sum of the ranks in the lower numbered group. In the case of ties, the ranks are averaged. The sum of ranks of the group with the lower number of entries is then calculated and called T1. T2 is then calculated according to the formula:

$$T2 = N2\ (N1 + N2 + 1) - T1 \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad (6)$$

where:

N1 is number in group with smaller number of entries,

N2 is number in group with larger number of entries.

R is then calculated as the smaller of T1 and T2 and the probabilities for rejection of the null hypotheses looked up on a table.[4] For values of N1 and N2 exceeding the table,

the approximate normal deviate is calculated by the formulae:

$$\left.\begin{aligned} Z &= (|M-R| - 0.5)/S \\ M &= N1\ (N1 + N2 + 1)/2 \\ S &= \sqrt{N2 \cdot M/6} \end{aligned}\right\} \quad . \ . \ . \ . \ . \quad (7)$$

where Z is the approximate normal deviate. The corresponding probabilities are calculated from a normalized probability table.

The limited data system consists of RANK 11, RANK 2, and RTABLE. RANK 11 arranges each of the NG groups in ascending order and then compares pairs of groups, calculating sum of ranks as indicated above. It then chains to RANK 2 which completes the calculation, types the answers and looks up the probabilities on RTABLE.DA. Finally RANK 2 bootstraps to chain to the calling program.

In the large data system RANK 12 reads data from a pair of groups, computes the sum of ranks and chains to RANK 2. RANK 2, finishing its tasks, returns to RANK 12 by a bootstrap. RANK 12 then increments the pair of groups to be tested.

Information is transferred between programs in the registers R (sum of ranks), NØ (number of entries in smaller group) and N1 (number of entries in larger group).

RANK 12 and RANK 2 are also used by RCORR 1 and RCORR 2 for optional rank test after the rank correlation test. This use is explained in detail under Rank Correlation. In order to allow the use of RANK12 by RCORR 1 and 2, RANK 12 checks for, and ignores, blanks.

4. Analysis of Variance; Program #4, VAR1, VARSUB, VAR2, FTAB.DA:

These programs perform analysis of Variance[5] accepting data in column format. In the large data system the limits on group and entry size are 2047.

The entities, "group sum square" and "group degrees of freedom", are calculated according to the following:

Given K groups of n_j entries each (j = 1,k) where x_{ij} is a typical entry (i = 1,n_j), one calculates a typical class total and the grand total as:

$$\left.\begin{aligned} G &= \sum_{j=1}^{K} T_j = \sum_{j=1}^{K} \left(\sum_{i=1}^{n_j} X_{ij} \right) \\ N &= \sum_{j=1}^{K} n_j \end{aligned}\right\} \quad \ldots\ldots\ldots \quad (8)$$

where:

G = grand total (sum)

T_j = typical class total and

N = total number of entries

then:

$$\left.\begin{aligned} &\text{Group DF} = K-1 \\ &\text{Group Sum Square} = \sum_{j=1}^{K} \left(T_j^2/n_j - G^2/N \right) \end{aligned}\right\} \quad \ldots \quad (9)$$

total sum square and degrees of freedom are calculated by:

$$\text{Total DF} = N-1$$

$$\text{Total Sum Square} = \left\{ \sum_{j=1}^{K} \left[\sum_{i=1}^{n_j} X_{ij}^2 \right] - G^2/N \right\} \quad \ldots \quad (10)$$

The "error" sum square and degrees of freedom are then calculated by subtraction of the group entities from the total entities. Group and error mean squares are calculated by division of the sum square by the corresponding degrees of freedom; and Fisher's F value by division of the group mean square by the error mean square. Probability for rejection of the null hypothesis is looked up in a table[6], according to the F value and the group and error degrees of freedom.

The limited data system (VAR1, VARSUB, and FTABLE) works as follows:

VAR1 calculates the sums, sum of squares and class totals from data in core. Mean squares, degrees of freedom and the F value are calculated and typed. VARSUB is used to adapt the two degrees of freedom to FTABLE so that VAR1 can read the table and print appropriate answers. If P > .Ø5, VAR2 then types the overall mean, standard deviation and number of the entire set. Control is then returned to RDR1 by a bootstrap process.

In the large data system, VAR2 cumulates the sums, sum of squares, and class totals, and then chains to VAR1 for further calculation and typing. By using the information in the register JDAT, VAR1 is able to return to RDR2 in the large data system.

5. Correlation; Program #5: CORR1, PROB, CORR2, CTABLE:

These programs calculate and type the following information on all pairs of groups: 1) the sample correlation coefficient and variance by the method of least squares[7]; 2) Fisher's Z value corresponding to the correlation coefficient[8]; 3) the probabilities corresponding to the correlation coefficient; and 4) the two regression lines.[9] It then can calculate a student's t test on the unpaired data of the two groups if that option is selected (IOPT = Ø). In the large data system up to 2Ø47 groups and entries may be used.

The formulae for calculation are as follows:

$$R = \sum_{i=1}^{N_o} (x_i - \bar{x}) \cdot (y_i - \bar{y}) \Big/ \left[\sum_{i=1}^{N_o} (x_i - \bar{x})^2 \cdot \sum_{i=1}^{N_o} (y_i - \bar{y})^2 \right] \quad (11)$$

$$V = \sum_{i=1}^{N_o} (y_i - \hat{y}_i)^2 / (N_o - 2) \quad (12)$$

$$Z = \log_e \left[\frac{1 + R}{1 - R} \right]^{1/2} \quad (13)$$

$$SZ = 1/\sqrt{N_o - 3} \quad (14)$$

$$\left.\begin{aligned} \hat{y} &= (\bar{y} - b\bar{x}) + b\hat{x} \\ b &= \sum_{i=1}^{N_o} (x_i - \bar{x})\,(y_i - \bar{y}) \Big/ \sum_{i=1}^{N_o} (x_i - \bar{x})^2 \end{aligned}\right\} \quad (15)$$

where:

i) R is the correlation coefficient

ii) N_o the number of paired data points in the two groups

iii) x_i, y_i the data points

iv) $\bar{x}$, $\bar{y}$ the means calculated according to the formula:

$$\bar{x} = \sum_{i=1}^{N_o} x_i/N_o; \quad \bar{y} = \sum_{i=1}^{N_o} y_i/N_o$$

v) V is the variance

vi) $\hat{y}_i$ is the value of y corresponding to x_i calculated from the regression line in Equation (15)

vii) Z is Fisher's Z value

viii) SZ or Sigma Z is the standard error of the distribution of Z

ix) $\hat{y}$ and $\hat{x}$ are theoretic values of x and y in the regression line obtained by least squares fit

x) b is the slope of the regression line.

CORR1, PROB and CTABLE comprise the limited data version of this test. CORR1 and PROB are also used by CORR2 for large data. CORR1, entered from RDR2 with MULT = +1, cumulates the data and products of data from the data buffer. It then calculates the entities above from this information and types out the correlation coefficient, Fisher's Z value, sigma Z, the variance and the degree of freedom corresponding to the correlation coefficient. CORR1 chains to PROB which looks up and types the probabilities corresponding to the correlation coefficient and the degrees of freedom. These are looked up on CTAB.DA, taken from Snedecor

and Cochran.[10] PROB sets the switch MULT = Ø and returns to CORR1: This switch causes entry to CORR1 at a location such that the regression line of the first group on the second is typed. CORR1 then swaps independent and dependent variables, calculates and prints out the regression line of the second group on the first. Two paths are then possible, depending on the contents of common storage, IOPT. This register, set to 0 by the user's answer "yes" to the question "option" in STATIS, causes CORR1 to set MULT = -1 and bootstraps so that it chains to STUD1 or STUD2, according to whether the limited data or large data system is being used. A student's t test is then run on the full data (not just the paired data) from the two groups. The entry point into STUD1 is controlled by setting MULT = -1. This same switch causes STUD1 to bootstrap back to CORR1 at any entry point which increments the pair of groups to be tested, tests them as above; and so forth.

If, however, IOPT is non-zero (caused by any answer save "y" to the question, "option ?" in STATIS, then CORR1 skips student's t test and increments group numbers and retests for the limited data system, or reloads CORR2 for large data.

With large data CORR2 is employed to cumulate the appropriate sums and products of data, chaining to CORR1 and PROB for output. When this typing is done, CORR1 examines IOPT and JDAT, and if IOPT = Ø, bootstraps in the program STUD2 which calculates and prints a student's t test on the unpaired data of the two groups, using STUD1 for output. STUD2 bootstraps back to CORR2 by examining JDAT. With MULT = 1 at entry, CORR2 then increments and checks subsequent pairs of groups, finally chaining to RDM2 when finished. If IOPT is non-zero, CORR1 chains back to CORR2 for subsequent groups.

6. Rank Correlation; Program #6: RCORR1, RCORR2:

Rank correlation calculates Spearman's rank correlation coefficient[9-11] according to the formula:

$$R = 1 - 6 \sum_{i=1}^{N\emptyset} (x_i - y_i)^2 / N\emptyset(N\emptyset^2 - 1) \quad \ldots\ldots\ldots \quad (16)$$

where $N\emptyset$ = number of paired data in the two test groups; x_i, y_i = relative ranks within group 1 and 2 respectively. In the large data system, the maximum group size permitted is 5ØØ; 3ØØ for optional rank test.

The program works as follows: The data in each group are paired, i.e., data are considered only when there are corresponding entries in both groups. This is accomplished by reading corresponding entries in each group. If either entry is blank (exceeds 1Ø,ØØØ), then the other is "crossed off" (11,111 is added to both). Since the program reads as non-blank numbers only those under 1Ø,ØØØ, this prevents the program from accepting these numbers on subsequent passes. NØ is used to count the number of non-blank pairs accepted. RCORR1 does this process in the data buffer array. The program then searches a given group for the lowest entry, assigns this entry a rank 1 in the array IB (I,J), crosses the entry off, loops back to find the rank 2 entry, and continues until all data are crossed off. RCORR1, operating upon the only copy of the data, must then restore the buffer by subtracting 11,111 from each entry in both groups. The differences in the ranks of corresponding entries of both groups are then squared and cumulated in the register R. Note that no provisions are made for a tie in ranks within a group.

RCORR1 calculates and types out the rank correlation coefficient. Since, for numbers of pairs over 1Ø, the probability table for correlation coefficient applies, RCORR1 calls upon PROB (from the correlation program) to look up and print out the probabilities corresponding to R. PROB determines whether it is used by Correlation or Rank Correlation. If by the latter, it uses data at the end of the table for NØ under 1Ø (the entries being those worked out by Kendall.[12] PROB then returns to the original calling program (either RCORR1 or RCORR2) for examination of subsequent pairs of groups, unless an optional rank test has been chosen. If the option for rank test was chosen by the user initially, then the register IOPT = Ø, and PROB calls and runs the system RANK12 - RANK 2 to run this test. After RANK 2 outputs the answers for the rank test, it chains back to RCORR2 for rank correlation of subsequent pairs of groups in the given set. This use of the large data system for all subsequent rank correlation is necessary, even though RCORR1 may have been used originally, since original paired data is lost by the rank test, and must be re-read into core.

RCORR2 calculates the cumulated squares of the differences in rank for the two test groups in the large data system, using the file MATRIX.DA on the system device for input. It then calls RCORR1, with the number of paired entries, NØ, and the cumulated square of the difference in ranks, R, in common storage. RCORR2 then calculates and types out the rank correlation coefficient, as above.

7. Chi Square; Program #7: CHSQ1, CHSQ2, XTABLE:

This test computes the value for X^2 with Yates' continuity correction, for an r x k contingency table,[13] according to the formula:

$$X^2 = \sum_{i=1}^{r} \sum_{j=1}^{K} \left[(|O_{ij} - E_{ij}| - 0.5)^2 / E_{ij} \right]$$

$$DF = (r - 1) \cdot (K - 1) \qquad \text{. . . . . (17)}$$

where:

O_{ij} = observed number in the 6th row and jth column of the table

E_{ij} = corresponding expected number under the null hypothesis, and

DF = degrees of freedom

This last entry, E_{ij}, is calculated from the row and column sums by the formula:

$$E_{ij} = S_r(i) \cdot S_c(j)/T \qquad \text{. . . . . (18)}$$

where:

$S_r(i)$ = the sum of the entries in the ith row

$S_c(j)$ = the sum of the entries in the jth column, and

T = the sum of all entries in the table

CHSQ1 cumulates the partial sums and calculates Chi squared and the degrees of freedom, storing them in the registers R and NØ. CHSQ2 accomplishes the same task by reading data from the two system files, DATA.DA and MATRIX.DA and writes the row and column sums on the system file CHISQ.DA. It then reads this file and MATRIX.DA in calculating and cumulating for X^2. The purpose of this is to avoid an array which would sharply limit the size of the contingency table which could be read in the large data system. CHSQ2 puts the value of X^2 in the register R and places the degrees of freedom in the Register NØ and then chains to CHSQ1.

CHSQ1 types out the answers and looks up the corresponding probabilities on the table, XTABLE,[14] returning finally to RDR1 for limited data or RDM2 for large data.

ADDITION OF NEW PROGRAMS IN SYSTEM

The following rules should be followed when adding new programs to the system, which can contain a total of 99 statistical tests.

1. The limited data version of the test will run on sets which have 7 groups or less and 50 entries or less in the largest group. If these limits might be exceeded by the user, a large data program (or dummy program) must be written.

2. The new test must be numbered and saved with the next number after the last test in the current system. For limited data system, this is S1Ø8.SV, and for large data system, S2Ø8.SV. It is then necessary to adjust the dimension of NFIL to 8 in all programs of the system and to write the title messages for the new test in STATIS and STATØ. The logic of STATIS which prevents tests above 7 from being recorded in the current system must be adjusted. In STATIS, STATØ, RDR1, RDR2 there are various conditional GOTO phrases according to the test number selected. These must be adjusted to include the new test. Depending on the nature of the new test (whether it works on ordered pairs, or not) the GOTO phrases of RDR1 and RDR2 and the conditional GOTO in STATØ which chains to RDR2 or RDM2 must be altered.

3. Common storage must be the same as RDR1 for the limited data test and identical to RDR2 as far as the entry, MAX, for large data. There are already 6 common registers (R, S1, S2, SS1, SS2, S3) for the transfer of information between sections of the new test in the limited data system and 4 registers for the transfer of integers. If more information must be transferred, the common registers must appear after entry A(I,J) in the limited data system and MAX in the large data system.

4. Dimensioned registers in common storage must include the same dimensions as RDR1 for limited data, and for large data, must include NFIL(X) for a new test, where X is the total number of tests on the new system.

5. The register MULT can be used to control entry points of a multiple-sectioned statistical test and can be altered at will. It will be set to +1 by the reading sections of the system. The other entries of common storage must not be disturbed by the new programs.

6. Any bootstrapping of the CALL CHAIN in a new program can be accomplished as follows. Write the new program with the bootstrapping routine as follows:

```
1000    Continue
  S     CLA
  S     TAD \INFO1
  S     DCA ]1ØØ
  S     TAD \INFO2
  S     DCA ]1ØØ#
        CALL CHAIN ('XXXX')
```

where INFO1 and INFO2 are registers which contain information used to alter the name of the program called by CALL CHAIN. Use the Fortran compiler and get the SABR assembly listing. In the SABR listing, following the location of the Fortran variables not in common storage, is a list and the location numbers of the dummies used by the various calls. The title, 'XXXX', will appear as, for example:

```
        ]22, 3Ø3Ø
(       3Ø3Ø
```

Then the Fortran program can be written in its final form, with]22 and]22# substituted for]1ØØ and]1ØØ#. One note of caution: the dummy file name, 'XXXX' must be unique in the program.

7. Any final output by the new program should use the call, WRITE (IOUT, Format #), so that output is consistently on the device chosen by STATIS.

8. Finally, the subroutine FRMAT should be used before each WRITE (IOUT,) statement to keep output pages at the same length. This subroutine, or the phrase LIN=LIN-1, should be used whenever a CALL CHAIN is to be accomplished. The subroutine LINFED is used before each CALL CHAIN to compensate for the fact that CALL CHAIN gives one line-feed to the teletype, but to no other output device.

Execution Times

In the limited data system, the execution times are essentially those of the output device when a disk is used as the system device. In a dectape system, however, considerable time is spent finding and loading programs, and this can be quite annoying if the user wants a fast answer to a simple statistical problem. In the large data system, the readers (particularly RDM2) are slow, even on a disk. In short, this system, limited in time by the many programs needed and the slowness of Fortran II is designed for batching large amounts of data, when the user's equipment is otherwise unused. The user desirous of repeated rapid runs of a simple statistical program will be quite frustrated, particularly if he does not possess a disc. In a dectape system, however, some time can be save by the apposition, at building time, of frequently called programs near the beginning of the system dectape.

BUILDING THE SYSTEM

The system may be built from the paper tape provided (ASCii source tapes) as follows: i) Programs: use the PS/8 Fortran compiler, SABR assembler and Linking Loader as indicated in the table below. Then use the SAVE command of PS/8, naming the program according to the table below; ii) Tables: type .R TABLE and answer the questions to read in paper tape; the rest is automatic.

TABLE FOR OPTIONS AND NAMES IN BUILDING SYSTEM

	NAME PRG	LINKING LOADER: SUBROUTINES IN ADDTN TO MAIN PRGM	LINKING LOADER: OPTION	NAME FOR SAVE
1.	STATIS	-	/I;/0	STATIS
	FRMAT	Not loaded, but assembled as FRMAT.RL		
	LINFED	Not loaded, but assembled as LINFED.RL		
	TABLE	-	/I;/0	TABLE
	STATØ	LINFED	/I;/0	STATØ
2.	RDR2	FRMAT, LINFED	/I	RDR1
	RDR2	FRMAT, LINFED	/I	RDR2
	RDM2	LINFED	/I;/0	RDM2
3.	STUD1	FRMAT, LINFED	/I	S1Ø2
	STUD2	LINFED	/I	S2Ø2
4.	RANK11	LINFED	-	S1Ø3
	RANK12	LINFED	/I	S2Ø3
	RANK2	FRMAT, LINFED	/I	RANK2
5.	VARSUB	Not loaded, but assembled as VARSUB.RL		
	VAR1	VARSUB, FRMAT, LINFED	/I	S1Ø4
	VAR2	LINFED	/I	S2Ø4
6.	CORR1	FRMAT, LINFED	-	S1Ø5
	CORR2	LINFED	/I	S2Ø5
	PROB	FRMAT, LINFED	/I	PROB
7.	RCORR1	FRMAT, LINFED	-	S1Ø6
	RCORR2	LINFED	/I	S2Ø6
8.	CHSQ1	FRMAT, LINFED	/I	S1Ø7
	CHSQ2	LINFED	/I;/0	S2Ø7

REFERENCES

1. Snedecor, G. W., and Cochran, W. G., Statistical Methods, 6th ed., Ames, Iowa: Iowa State Univ. Press, 1967, p. 100.
2. Ibid., Table A4, p. 549.
3. Ibid., pp. 130-132.
4. Ibid., Table A10, pp. 555-6.
5. Ibid., Chap. X.
6. Ibid., Table A14, pp. 560-563.
7. Ibid., p. 147; pp. 172-5.
8. Fisher, R. A., Metron 1:3, 1921.
9. Snedecor and Cochran, op. cit., pp. 135-144.
10. Ibid., Table A11, p. 557.
11. Spearman, C., Am. J. Psych. 15:88, 1904.
12. David, S. T., Kendall, M. G., and Stuart, A., Biometrika 38:131, 1951.
13. Snedecor and Cochran, op. cit., pp. 250-253.
14. Siegel, S., Appendix Table C, Nonparametric Statistics for the Behavioral Sciences, New York: McGraw-Hill Book Co., 1956, p. 249.

APPENDIX:

STATISTICAL TABLES USED IN SYSTEM

1. BY STUDENT'S T (TTAB.DA):

```
TTAB@@@@STUDNT
42
12.706    63.657
04.303    09.925
03.182    05.841
02.776    04.604
02.571    04.032
02.447    03.707
02.365    03.499
02.306    03.355
02.262    03.250
02.228    03.169
02.201    03.106
02.179    03.055
02.160    03.012
02.145    02.977
02.131    02.947
02.120    02.921
02.110    02.898
02.101    02.878
02.093    02.861
02.086    02.845
02.080    02.831
02.074    02.819
02.069    02.807
02.064    02.797
02.060    02.787
02.056    02.779
02.052    02.771
02.048    02.763
02.045    02.756
02.042    02.750
02.030    02.724
02.021    02.704
02.014    02.690
02.008    02.678
02.004    02.669
02.000    02.660
01.994    02.648
01.989    02.638
01.986    02.631
01.982    02.625
01.980    02.617
01.960    02.579
```

2. BY RANK TEST (RTAB.DA):

```
RTAB@@@@RANKS@@#'S=0.1*ACTUAL #!S
350
-1.000    -1.000
-1.000    -1.000
-1.000    -1.000
-1.000    -1.000
00.300    -1.000
00.300    -1.000
00.300    -1.000
00.400    -1.000
00.400    -1.000
00.400    -1.000
00.400    -1.000
00.400    -1.000
00.400    -1.000
00.500    -1.000
00.500    -1.000
00.500    00.300
00.500    00.300
00.600    00.300
00.600    00.300
00.600    00.300
00.600    00.300
00.600    00.300
00.700    00.300
00.700    00.400
00.700    00.400
-1.000    -1.000
00.600    -1.000
00.700    -1.000
00.700    -1.000
00.800    -1.000
00.800    00.600
00.900    00.600
00.900    00.600
01.000    00.700
01.000    00.700
01.100    00.700
01.100    00.800
01.200    00.800
01.200    00.800
01.300    00.800
01.300    00.900
01.400    00.900
01.400    00.900
01.500    01.000
01.500    01.000
01.600    01.000
01.600    01.100
01.700    01.100
01.700    01.100
00.000    00.000
01.000    -1.000
01.100    -1.000
01.200    01.000
```

01.300	01.000
01.400	01.100
01.500	01.100
01.500	01.200
01.600	01.200
01.700	01.300
01.800	01.400
01.900	01.400
02.000	01.500
02.100	01.500
02.100	01.600
02.200	01.600
02.300	01.700
02.400	01.800
02.500	01.800
02.600	01.900
02.700	01.900
02.800	02.000
02.800	02.000
02.900	02.100
00.000	00.000
00.000	00.000
00.000	00.000
01.700	01.500
01.800	01.600
02.000	01.700
02.100	01.700
02.200	01.800
02.300	01.900
02.400	02.000
02.600	02.100
02.700	02.200
02.800	02.200
02.900	02.300
03.100	02.400
03.200	02.500
03.300	02.600
03.400	02.700
03.500	02.800
03.700	02.900
03.800	02.900
03.900	03.000
04.000	03.100
04.200	03.200
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
02.600	02.300
02.700	02.400
02.900	02.500
03.100	02.600
03.200	02.700
03.400	02.800

03.500	03.000
03.700	03.100
03.800	03.200
04.000	03.300
04.200	03.400
04.300	03.600
04.500	03.700
04.600	03.800
04.800	03.900
05.000	04.000
05.100	04.200
05.300	04.300
05.500	04.400
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
03.600	03.200
03.800	03.400
04.000	03.500
04.200	03.700
04.400	03.800
04.600	04.000
04.800	04.100
05.000	04.300
05.200	04.400
05.400	04.600
05.600	04.700
05.800	04.900
06.000	05.000
06.200	05.200
06.400	05.300
06.600	05.500
06.800	05.700
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
04.900	04.300
05.100	04.500
05.300	04.700
05.500	04.900
05.800	05.100
06.000	05.300
06.300	05.400
06.500	05.600
06.700	05.800

07.000	06.000
07.200	06.200
07.400	06.400
07.700	06.600
07.900	06.800
08.200	07.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
06.300	05.600
06.500	05.800
06.800	06.100
07.100	06.300
07.300	06.500
07.600	06.700
07.900	07.000
08.200	07.200
08.400	07.400
08.700	07.600
09.000	07.800
09.300	08.100
09.500	08.300
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
07.800	07.100
08.100	07.400
08.500	07.600
08.800	07.900
09.100	08.100
09.400	08.400
09.700	08.600
10.000	08.900
10.300	09.200
10.700	09.400
11.000	09.700
00.000	00.000

00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
09.600	08.700
09.900	09.000
10.300	09.300
10.600	09.600
11.000	09.900
11.400	10.200
11.700	10.500
12.100	10.800
12.400	11.100
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
11.500	10.600
11.900	10.900
12.300	11.200
12.700	11.500
13.100	11.900
13.500	12.200
13.900	12.500
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000
00.000	00.000

```
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
13.700    12.500
14.100    12.900
14.500    13.300
15.000    13.700
15.400    14.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
16.000    14.700
16.400    15.100
16.900    15.500
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
```

```
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
18.500    17.100
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
00.000    00.000
```

3. ANALYSIS OF VARIANCE (FTAB.DA):

```
FTAB@@@ANAVAR@@#'S FOR IV2=1 ARE .01*ACTUAL #'S
361
01.610    40.520
18.510    98.490
10.130    34.120
07.710    21.200
06.610    16.260
05.990    13.740
05.590    12.250
05.320    11.260
05.120    10.560
04.960    10.040
04.840    09.650
04.750    09.330
04.600    08.860
04.490    08.530
04.350    08.100
04.260    07.820
04.170    07.560
04.080    07.310
04.030    07.170
02.000    49.990
19.000    99.000
09.550    30.820
06.940    18.000
05.790    13.270
05.140    10.920
```

04.740	09.550
04.460	08.650
04.260	08.020
04.100	07.560
03.980	07.200
03.880	06.930
03.740	06.510
03.630	06.230
03.490	05.850
03.400	05.610
03.320	05.390
03.230	05.180
03.180	05.060
02.160	54.030
19.160	99.170
09.280	29.460
06.590	16.690
05.410	12.060
04.760	09.780
04.350	08.450
04.070	07.590
03.860	06.990
03.710	06.550
03.590	06.220
03.490	05.950
03.340	05.560
03.240	05.290
03.100	04.940
03.010	04.720
02.920	04.510
02.840	04.310
02.790	04.200
02.250	56.250
19.250	99.250
09.120	28.710
06.390	15.980
05.190	11.390
04.530	09.150
04.120	07.850
03.840	07.010
03.630	06.420
03.480	05.990
03.360	05.670
03.260	05.410
03.110	05.030
03.010	04.770
02.870	04.430
02.780	04.220
02.690	04.020
02.610	03.830
02.560	03.720
02.300	57.640
19.300	99.300
09.010	28.240
06.260	15.520

05.050	10.970
04.390	08.750
03.970	07.460
03.690	06.630
03.480	06.060
03.330	05.640
03.200	05.320
03.110	05.060
02.960	04.690
02.850	04.440
02.710	04.100
02.620	03.900
02.530	03.700
02.450	03.510
02.400	03.410
02.340	58.590
19.330	99.330
08.940	27.910
06.160	15.210
04.950	10.670
04.280	08.470
03.870	07.190
03.580	06.370
03.370	05.800
03.220	05.390
03.090	05.070
03.000	04.820
02.850	04.460
02.740	04.200
02.600	03.870
02.510	03.670
02.420	03.470
02.340	03.290
02.290	03.180
02.370	59.280
19.360	99.360
08.880	27.670
06.090	14.980
04.880	10.450
04.210	08.260
03.790	07.000
03.500	06.190
03.290	05.620
03.140	05.210
03.010	04.880
02.920	04.650
02.770	04.280
02.660	04.030
02.520	03.710
02.430	03.500
02.340	03.300
02.250	03.120
02.200	03.020
02.390	59.810
19.370	99.370

08.840	27.490
06.040	14.800
04.820	10.290
04.150	08.100
03.730	06.840
03.440	06.030
03.230	05.470
03.070	05.060
02.950	04.740
02.850	04.500
02.700	04.140
02.590	03.890
02.450	03.560
02.360	03.360
02.270	03.170
02.180	02.990
02.130	02.880
02.410	60.220
19.380	99.390
08.810	27.340
06.000	14.660
04.780	10.150
04.100	07.980
03.680	06.710
03.390	05.910
03.180	05.350
03.020	04.950
02.900	04.630
02.800	04.390
02.650	04.030
02.540	03.780
02.400	03.450
02.300	03.250
02.210	03.060
02.120	02.880
02.070	02.780
02.420	60.560
19.390	99.400
08.780	27.230
05.960	14.540
04.740	10.050
04.060	07.870
03.630	06.620
03.340	05.820
03.130	05.260
02.970	04.850
02.860	04.540
02.760	04.300
02.600	03.940
02.490	03.690
02.350	03.370
02.260	03.170
02.160	02.980
02.070	02.800
02.020	02.700
02.430	60.820

19.400	99.410
08.760	27.130
05.930	14.450
04.700	09.960
04.030	07.790
03.600	06.540
03.310	05.740
03.100	05.180
02.940	04.780
02.820	04.460
02.720	04.220
02.560	03.860
02.450	03.610
02.310	03.300
02.220	03.090
02.120	02.900
02.040	02.730
01.980	02.620
02.440	61.060
19.410	99.420
08.740	27.050
05.910	14.370
04.680	09.890
04.000	07.720
03.570	06.470
03.280	05.670
03.070	05.110
02.910	04.710
02.790	04.400
02.690	04.160
02.530	03.800
02.420	03.550
02.280	03.230
02.180	03.030
02.090	02.840
02.000	02.660
01.950	02.560
02.450	61.420
19.420	99.430
08.710	26.920
05.870	14.240
04.640	09.770
03.960	07.600
03.520	06.350
03.230	05.560
03.020	05.000
02.860	04.600
02.740	04.290
02.640	04.050
02.480	03.700
02.370	03.450
02.230	03.130
02.130	02.930
02.040	02.740
01.950	02.560
01.900	02.460
02.460	61.690

19.430	99.440
08.690	26.830
05.840	14.150
04.600	09.680
03.920	07.520
03.490	06.270
03.200	05.480
02.980	04.920
02.820	04.520
02.700	04.210
02.600	03.980
02.440	03.620
02.330	03.370
02.180	03.050
02.090	02.850
01.990	02.660
01.900	02.490
01.850	02.390
02.480	62.080
19.440	99.450
08.660	26.690
05.800	14.020
04.560	09.550
03.870	07.390
03.440	06.150
03.150	05.360
02.930	04.800
02.770	04.410
02.650	04.100
02.540	03.860
02.390	03.510
02.280	03.250
02.120	02.940
02.020	02.740
01.930	02.550
01.840	02.370
01.780	02.260
02.490	62.340
19.450	99.460
08.640	26.600
05.770	13.930
04.530	09.470
03.840	07.310
03.410	06.070
03.120	05.280
02.900	04.730
02.740	04.330
02.610	04.020
02.500	03.780
02.350	03.430
02.240	03.180
02.080	02.860
01.980	02.660
01.890	02.470
01.790	02.290
01.740	02.180
02.500	62.610

19.460	99.470
08.620	26.500
05.740	13.830
04.500	09.380
03.810	07.230
03.380	05.980
03.080	05.200
02.860	04.640
02.700	04.250
02.570	03.940
02.460	03.700
02.310	03.340
02.200	03.100
02.040	02.770
01.940	02.580
01.840	02.380
01.740	02.200
01.690	02.100
02.510	62.860
19.470	99.480
08.600	26.410
05.710	13.740
04.460	09.290
03.770	07.140
03.340	05.900
03.050	05.110
02.820	04.560
02.670	04.170
02.530	03.860
02.420	03.610
02.270	03.260
02.160	03.010
01.990	02.690
01.890	02.490
01.790	02.290
01.690	02.110
01.630	02.000
02.520	63.020
19.470	99.480
08.580	26.350
05.700	13.690
04.440	09.240
03.750	07.090
03.320	05.850
03.030	05.060
02.800	04.510
02.640	04.120
02.500	03.800
02.400	03.560
02.240	03.210
02.130	02.960
01.960	02.630
01.860	02.440
01.760	02.240
01.660	02.050
01.600	01.940

4. BY CORRELATION AND RANK CORRELATION (CTAB.DA):

```
CTAB@@@@@TABLE FOR CORR AND RCORR
56
00.997    01.000
00.950    00.990
00.878    00.959
00.811    00.917
00.754    00.874
00.707    00.834
00.666    00.798
00.632    00.765
00.602    00.735
00.576    00.708
00.553    00.684
00.532    00.661
00.514    00.641
00.497    00.623
00.482    00.606
00.468    00.590
00.456    00.575
00.444    00.561
00.433    00.549
00.423    00.537
00.413    00.526
00.404    00.515
00.396    00.505
00.388    00.496
00.381    00.487
00.374    00.478
00.367    00.470
00.361    00.463
00.355    00.456
00.349    00.449
00.325    00.418
00.304    00.393
00.288    00.372
00.273    00.354
00.250    00.325
00.232    00.302
00.217    00.283
00.205    00.267
00.195    00.254
00.174    00.228
00.159    00.208
00.138    00.181
00.113    00.148
00.098    00.128
00.088    00.115
00.062    00.081
99.999    99.999
99.999    99.999
99.999    99.999
99.999    99.999
```

```
01.000    99.999
00.886    01.000
00.750    00.893
00.714    00.857
00.683    00.833
00.648    00.794
```

5. BY CHI SQUARE (XTAB.DA):

```
XTAB@@@@@@@CHI SQUARE@@LAST 7 ENTRIES=CHISQ/10
37
03.840    06.630
05.990    09.210
07.810    11.340
09.490    13.280
11.070    15.090
12.590    16.810
14.070    18.480
15.510    20.090
16.920    21.670
18.310    23.210
19.680    24.720
21.030    26.220
22.360    27.690
23.680    29.140
25.000    30.580
26.300    32.000
27.590    33.410
28.870    34.810
30.140    36.190
31.410    37.570
32.670    38.930
33.920    40.290
35.170    41.640
36.420    42.980
37.650    44.310
38.890    45.640
40.110    46.960
41.340    48.280
42.560    49.590
43.770    50.890
05.576    06.369
06.750    07.615
07.908    08.838
09.053    10.042
10.188    11.233
11.314    12.412
12.434    13.581
```