DECUS
PROGRAM LIBRARY

DECUS NO.	8-660b
TITLE	STAT
AUTHOR	Lars Palmer
COMPANY	AB HASSLE Fack 431 20 MOLNDAL 1 Sweden
DATE	4 Aug. 1976
SOURCE LANGUAGE	PAL 8 FORTRAN IV, RALF

ATTENTION

This is a USER program. Other than requiring that it conform to submittal and review standards, no quality control has been imposed upon this program by DECUS.

The DECUS Program Library is a clearing house only; it does not generate or test programs. No warranty, express or implied, is made by the contributor, Digital Equipment Computer Users Society or Digital Equipment Corporation as to the accuracy or functioning of the program or related material, and no responsibility is assumed by these parties in connection therewith.

PROGRAM REVISION

A. GENERAL INFORMATION

Object Computer(s) PDP-8 Source Computer (if different) 8

Original File Name and Title STAT DECUS or DEC No.

New File Name and Title (if different)

Original Author Lars Palmer

Revising Author same

Affiliation AB HASSLE

Address Fack

431 20 MOLNDAL 1

Country SWEDEN

Category Source Language

Monitor/Operating System DEC No.

Core Storage Required 12K; Full Utilization (Batch + all files) Starting Address

Peripherals Required OS/8 FORTRAN IV

Other Software Required F IV DEC or DECUS No.

Restrictions, Deficiencies, Problems no

B. REASON(s) FOR REVISION

1. Debug, correct known problem ☐
2. Extend to handle new or different configurations ☐
3. Operate under different monitor or new system ☐
4. Increased operational efficiency ☒
5. Operate on different processor ☐ Specify
6. Other (please specify)

C. TAPES AVAILABLE

Paper Tapes Object Binary ☐ Binary Patch ☐ Object ASCII ☐ Source ☐ Other

DECtape ☒ LINCtape ☐ Format Magtape: 7 Track ☐ 9 Track ☐ BPI

Object Files ☒ Source Files ☒ Documentation Files ☒ Other

tape 1 tape 2 1/2

D. ABSTRACT

See page 1 of DECUS Program Library Write-up

CATALOG ABSTRACT

DECUS No. 8-660 b

STAT

Lars Palmer, AB Hässle, 431 20 MÖLNDAL , Sweden

A large FORTRAN IV package consisting of one large program for the statistical analyses and several assessory programs specifically for manipulation on the data matrix and for analyses of a more complex nature that cannot be covered by the main program.

Minimum hardware:	OS/8 FORTRAN IV compatible; Disk and FPP very useful
Other programs needed:	FORTRAN IV
Storage Requirement:	Storage will operate in 12K; Full utilization (BATCH + all files) requires 16K
Miscellaneous:	Tape 1 contains all the necessary programs to run the package. Tape 2 contains all the FORTRAN sources and batch files (Order either or both - specify)
Source Language:	FORTRAN IV, RALF

STAT with all its programs is on two tapes. Tape 1 contains the load modules to STAT and OUTLAY (the two main programs in the system) and the relocatable modules to KSORT and its function XFUNCT. The philosophy of the programs is described in HASSLE STAT PACK (file STAT00.WU) which you should have received as a write up to the program. Manuals to the running of the programs are in the files STAT.WU, OUTLAY.WU and KSORT.WU.

It is possible to run the programs with a normal Run Time System. However, they do call a subroutine which requires a small patch to FRTS. The patch can be affected by overlaying FRTS, version 3.05, with the binary FRTSOV.V3 which is on tape 1.

```
.LOAD SYS:FRTS.SV/I$FRTSOV.V3$
.SAVE SYS:FRTS
```

Sources and comments to this tape are on the DECUS submitted FORTRAN patch tape. DECUS No. was unknown when this was written.

The files STATST.DA and STATDA.DA are test files of running STAT. They are mentioned in the STAT write up.

On tape 2 there are all the sources necessary to compile the system including the sources to the FRTSX patches and the RALF subroutines which are called by the system. The Batch files, STATB and STATC, can be used to build the system but note that these files do not explicitly load the RALF modules as we have these included in our library. You will have to add the modules to your library or modify the Batch files. Several files from the SSP Library are also used by the STAT program. These are only included as binaries on the tape. Also included on the tape are some other FORTRAN statistical programs which will work on the same data files as STAT. These programs do not have proper write ups and are therefore not considered submitted to DECUS. You will have to look at the sources and the comments there if you want to use them.

The two files CONTEN.TX are commented directories of the respective tapes.

DECUS NO. 8-660b

Write up to STAT consists of the following page and write up No. 1.
If possible also 2, 3 and 4.

MARCH 15 1976

HASSLE STAT PACK

Lars Palmer
AB HASSLE
Fack
431 20 Molndal 1
Sweden

HASSLE STAT PACK
Generals

Hassle statistical packages consist of three distinct components:

A)2 simple statistical programs for basal statistical analyses written in BASIC (STAT1.BA and STAT2.BA).

B)A more advanced and complete statistical package written in FOCAL (not available under ETOS). (Not supported by Hassle any lonser.)

C)The most complete of the packages; a large FORTRAN IV package consisting of one large program for the statistical analyses and several assessory programs specifically for manipulation on the data matrix and for analyses of a more complex nature that can not be covered by the main program.

This introduction covers some general aspects which are relevant to all of the programs. Any diversences from this introduction will be specified under the different programs.

A.THE DATA FORMAT

As in all statistical analyses we start with a data matrix consisting of a number of rows and a number of columns. The rows are individual observations on the same or different individuals. The columns are different types of observations.

As a simple example: Label the columns length, height, weight and label the rows A,B,C etc for each individual. The number of columns can be as low as one in which case we can only calculate parameters on this variable. The maximum number is different in the different programs. If we have several columns we can also besides the parameters calculate statistical measures between the columns.

B.MISSING DATA

All routines (except the BASIC programs) handle missing data. As a convention in all the programs missing data is set to zero. As both FOCAL and FORTRAN are able to handle very small numbers there is no objection for using 10E-15 instead of a zero value. The FOCAL and FORTRAN programs in all routines compensate the missing values.

This compensation can be treated in three different ways:

1.On calculating parameters, all zero values in the column are skipped.

2.When comparing two columns, e.g. by T-tests, all observations in both columns are skipped if there is a missing data in one column.

3.In certain cases, e.g. Two-way Analysis of Variances, the whole row is skipped if there is a single missing data in any column.

C.END CODES

When inputting data to the programs, it is necessary to signal to the program that end of data is reached. This is done by giving an end code. The end code is in most cases a very large number, but other conventions may in some cases be used. Also in the data file extremely large numbers are sometimes used to signal end of data to the program when doing input from the file. This numbers are automatically inserted by the programs and can, therefore, appear on listing of a data file even though it was not input from the keyboard.

D.DATA FILES

The FOCAL and FORTRAN programs were written with the goal to provide programs that could be used for a large number of statistical analyses easily requested from the keyboard and at the same time able to preserve the data that once had been input. For this reason the programs create and read from data files. The creation of the files is described in the individual program manuals. The files are not compatible, i.e. it is not possible to create a program file by a FOCAL package and read it with a FORTRAN or vice versa. However, the structure of the files is reasonably compatible and it would not be very difficult to write a conversion program between the files. Such a program is, however, not included in the package.

E.HEADINGS

In the FORTRAN package, there is also a capability of including headings to the materials. The data file holds one heading (72 characters long) for the whole material and one heading (18 characters) for each of the sub groups. These headings are copied onto the output list by the statistical program. In some cases, e.g. in the ANOVA program, the headings are used for special purposes.

F.VARIOUS ANALYSES IN THE PACKAGE

All three program packages are based on a list of different statistical analyses which have been kept as far as possible the same in all the programs. This is commented on here and some general comments are made as to the formulas used in the values analyses.

NUMBER	B	F	F(BASIC,FOCAL,FORTRAN)
-1	-	x	x	Lists the indata. In some cases it also contains correction routines.
0	x	x	x	Always fetches new data, either from keybboard or from the input data file. Actual interpretation of zero varies slightly between the programs.
1	x	x	x	Calculation of means, standard deviation and standard error according to standard formulas. The devisor for standard deviation is in all cases N= -1.
2	x	x	x	T-tests. Three formulas for T-tests are implemented in the program. These are: A)Standard T-test comparing the means of the two columns calculating according to the formula (I). B)T-test with the presumption that the variances in the different groups are not equal (formula II).

NUMBER	B	F	F(BASIC,FOCAL,FORTRAN)
			C)T-test on paried values. If one value in a row is missing so is the row skipped in the calculation (formula III).
3	x	x	x	Regression lines calculated accordins to standard formulas (see ref. S&R page 7) from which test data for regression lines is taken.
4	-	x	x	The normal regression line presumes that X is measured without error. If this is not the case, i.e. if X and Y are random variables. There are some special ways of solvins the problem. The one used here is the so called Bartletts method of regression. See reference S&R page 7 from which also the test data is taken.
5	-	x	x	Correlations expressed as correlation matrix.
6	-	x	x	Analysis of Variance, in the text usually abbreviated to ANOVA.
7	-	-	x	Sheffe contrasts. The Sheffe contrasts in both the FORTRAN and FOCAL programs is an adoptation of the program DECUS FOCAL no 8-66. The Sheffe contrasts is calculated as a complete matrix on columns against each other for the siven levels of significants. For further details see the FOCAL and FORTRAN programs (see also analysis 36).
7	-	x	-	Cochrans & Bartletts tests.

NUMBER	B	F	F(BASIC,FOCAL,FORTRAN)
8	-	x	x	Analysis of Variances Two-sided, calculated according to normal formulas. The FOCAL program refuses to accept analysis 8 if missing data in the matrix. The FORTRAN program solves this problem by skipping every row in which there is a missing data. Unsymmetric Two-way Analysis of Variances can be solved by the program UNSYM which is included in the FORTRAN package.
13	-	x	x	Wilcoxon matched pairs rank sign test, programmed exactly from Siegel (see ref. S page 7) both for the large and the small cases and corrections for ties (Siegel test data pages 82 resp. 79).
14	-	x	x	Mann-Whitney U-test. The comments in 13 apply here too (Siegel test data page 122).
15	-	x	x	Kruskal-Wallis One-way Analysis of Variances by ranks, programmed from Siegel (test data page 187).
16	-	x	x	Spearman rank programmed direct from Siegel (test data page 206).
17	-	x	x	Friedmanan Two-way non-parametric Analysis of Variances (Siegel test data page 171).
18	-	-	x	Numerical integration performed for laboratory data. The numerical integration is performed both by a method of overlapping parabolas and by the Trapeze method.
19	-	-	x	Calculate medians and ranks of the given column.
20	-	-	x	Cross tabulation with one column as X axis and one as Y axis. Count of how many fall in the intervals.

HASSLE STAT PACK
Generals

NUMBER	B	F	F(BASIC,FOCAL,FORTRAN)	
21	-	-	x	Scatter plot. Actually the same information as in 20 given as a plot of X against Y.
21	-	x	-	Rearrangement of data matrix.
22	-	x	x	Various conversions such as converting one column to the logarithms of another given column.
36	-	x	-	Sheffe (see 7).
37	-	x	-	Chi-two.
47-49	-	x	-	File manipulation.
48	-	-	x	Comments.
49	-	-	x	Flag settings.
50	-	x	x	End of analysis exit the monitor.

G.REFERENCES

The following are referred to at various places, and data from these books are used as test data in the STATDA.DA file.

S.SIEGEL (S)	Nonparametric Statistics for the Behavioral Sciences 1956 (Analyses Nos. 11-17)
R SOKAL & F J ROHLF (S&R)	Biometry 1969 (Analyses Nos. 4,6,8)
W DIXON & F MASSEY (D)	Introduction to Statistical Analysis 1969 (Analyses Nos. 2,3,5)
P DAVIS & P RABINOWITZ (D&R)	Numerical Integration 1967 (Analysis No. 18).
DECUS FOCAL 8-66	(Analyses Nos.7 & 36).

H.TEST DATA

To the FOCAL and FORTRAN programs there exists a test data file which contains data to test almost all functions of programs. This test data is taken from various sources.

3,4	S&R
6,8	D
12-17	SIEGEL

I.MAIN DIFFERENCES

The principle difference between the three programs - the BASIC,FOCAL and FORTRAN programs - lies in the implementation of the handling of the data.

The BASIC program includes some very simple programs that read the data from the keyboard but do not save the data. Only sums and product sums are accumulated as required for the analyses.

The FOCAL program is more sophisticated. It fetches data from the keyboard, saves the data in core and on request analyzes on it and writes it to the output file. Due to this construction and due to the FOCAL program language, the FOCAL program is limited to some about 100 variables and is slow in execution.

The FORTRAN package is implemented slightly differently. It consists of three programs:

1)OUTLAY which reads data from the keyboard and creates the data file.

2)KSORT which is available for various reorganizations of the data file and performs conversions of the file (which is more complex than those that are in the FOCAL package) and

3)STAT, the statistical package.

This arrangement is on the whole not suitable for data input from keyboard but specifically designed for file input. This means that at least two and sometimes three programs have to be invoked to do a statistical analysis. The advantage of the FORTRAN program is a much larger repertoire of statistical analyses available, a much larger area (at the expense of utilizing 12 K of core) and a much faster execution. The latter is true in a non-FPP configuration and of course much more so in a system having a floating point processor.

MARCH 15 1976

HASSLE STAT PACK
FORTRAN IV - STAT

Lars Palmer
AB HASSLE
Fack
431 20 Molndal 1
Sweden

Applies to STAT version 5.00.

HASSLE STAT PACK
F IV - Stat

A)GENERAL

The components of the total package are:

1)STAT	Described here.
2)OUTLAY	Creates an input file to STAT.
3)KSORT	Rearranges rows and columns in the file.
4)OUTCHK	Checks on unformatted file for errors.
5)BIGSTA	Handles >10 columns.

Operating on OUTLAY files are also several specific programs e.g. LESQ that does a non linear least squares curve fit.

B)IMPLEMENTATION

As of todays date the following routines are implemented:

NUMBER	PERFORMS
-1	LIST
0	NEW DATA FROM SAME FILE
1	MEANS ETC
2	T-TESTS
3	REGRESSION LINES
4	TYPE 2 REGRESSION
5	CORRELATION MATRIX
6	ANOVA ONE SIDED
7	SHEFFE CONTRASTS
8	ANOVA TWO SIDED
13	WILCOXON
14	MANN-WHITNEY
15	KRUSKAL-WALLIS
16	SPEARMAN RANK
17	FRIEDMAN
18	NUMERIC INTEGRATION ON X/Y
19	MEDIANS AND RANK OF COLUMN
20	CROSSTABULATION
21	SCATTER PLOT
22	VARIOUS CONVERSIONS
48	OUTPUT A COMMENT
49	FLAG-SETTING ROUTINE
50	EXIT TO MONITOR

C)INDATA AND FILE UNIT NUMBERS

The following indata files are used:

UNIT	CONTAINS
6	The data - in OUTLAY format - to be analyzed.
4(TTY)	A parameter file containing the request for the analyses to be performed. Format see below (D).

OUTFILES

UNIT	CONTAINS
LDEVE	Error information and trace of program flow.
LDEVL	Normal program output unit.

Assignment of these devices is as follows:

	/4=FILE	/4=TERMINAL
LDEVE	3	0
LDEVL	3	0

The assignment of LDEVL can be reversed by stating /L on the last line to FRTS (the same as the altmode). This makes it possible to run from terminal with the line printer as output unit for the results.

LDEVE and LDEVL are further specified under card 49.

D)THE PARAMETER FILE

Note that at program start the program reads the parameter file as if it just had seen a 0 request card.

The request cards have the following format:
CT(CARD TYPE),ARG1(ARGUMENT 1),ARG2,....

Some of the request cards must be followed by parameter cards.
The most commonly used format is:

CT, KOL1, KOL2, IOPT, with KOL1, KOL2 beins 2 cols to be compared and IOPT an analysis specific parameter.

CARD TYPE	ARGUMENTS	PARAMETER CARDS
-1	0-1	1
0	0	1(-4) GROUPS
		+1 COLUMNS
1	0-2	
2	3	
3	2-3	
4	2-3	0-1
5	0	
6	0	
7	0-1	0-1
8	0	
13	2	
14	2	
15	0	
16	2	
17	0-1	
18	2-3	
19	0-1	
20	2-3	
21	2-3	0-1
22	1-3	
48	0-1	AS SPECIFIED BY ARGUMENT.
49	0-4	
50	0	

****COMMENTS***
<is optional argument>

CARD TYPE

-1 The request card -1 asks for a listing of the input data.

If ARG1=0 it must be followed by a card indicating the format for the listing. This must be a standard FORTRAN format notation with as many format descriptions as there are columns in the file. It should contain 1X , as its first element.

If ARG1 <> 0 one of three internal formats is used:

-1,1 F10.1
-1,2 F10.4
-1,3 G12.4

****MISSING DATA****
All routines in the program treat a 0-value as missing data (nonmissing data = 0 can be set as e.g. 1E-25). Format -1,3 is not suitable for missing data. If there are any, a specification error will be generated.

0 Reads new data. The parameter file always begins with the parameter card following the 0 card.

****PARAMETER CARD***
1(-4) Number of groups to be read into core and their numbers. The group numbers are not preserved in core. They are treated as one group.
Two special forms of this input are available:

1)If only the number of groups is specified, it is taken to mean the first N sequential groups
i.e.
5
is interpreted equivalent to
5,1,2,3,4,5

Note: IF N>24 use several cards see *

* These cards are in 25I3 formats and must if necessary be carried over to another card.

CARD TYPE

2)If the first card in the input file is negative, it implies that the analyses from here to the next 0 card, or to the 50 card if no 0 card intervenes, are to be done for all groups in the input file
i.e.
-1
5
1
5
0
implies 'On all groups in input file generate mean and correlation matrix for the first 5 columns'. If the requests are given from the console this option should not be used.

NEXT Number of columns to be used and their numbers. The columns can be reorganized in core by placing the request in a non-sequential manner. Special form 1) as described under above also applies here.

If number of col is stated > than number of columns in file, it is adjusted to the latter. If a totally empty column is read, the column count is reduced by analysis no 1. Therefore, it is possible to routinely specify e.g. 10 as number of columns if all existing columns are wanted.

1 The treatment of the arguments is:
If ARG1 and ARG2 both = 0 calculate means and standard errors.

If ARG1=15 calculate means only on rows that are complete i.e. contain no missing data, see 7.

If ARG1<=NK and ARG2=0 print sums of squares of col(ARG1).

If ARG1 and ARG2 both <= NK print sums of squares and crossums for all complete pairs. If one data is missing so are both skipped.

2 3 types of T- tests are in the package.

Calling: 2,K1,K2,N
N is type of test and K1 K2 the 2 columns to compare.

N=1 Standard formula. Sigma presumed equal.
N=2 Sigma not presumed equal.
N=3 Paired values.

CARD TYPE

3 Regression line - Calling 3,K1,K2<IOPT>.
If IOPT > 1 the calculated Y values for all X values are also tabulated.

4 Type II Regression (Bartletts 3 group) ref S&R page 480.

Calling 4,KOLX,KOLY,IOPT OPTIONAL
The line (Y=A+BX) is calculated and confidence limits to B are printed.

If IOPT = 0 the values P=0.05,0.01,0.001 are used.
If IOPT>0 - IOPT(<4) P values must be given. These are in the form 0<P<0.5.

5 Correlation. No arguments.
Numbers of pairs in each combination is printed if they are unequal.

6 Oneway Analysis of variance (Anova).

8 Twoway d:o (see also under 7).

7 Sheffes contrasts. To perform this analysis, it is necessary to have some special information:

1)Means must be calculated.
2)The mean square must be calculated.
3)The program must be supplied with the relevant P-values.

Therefore, the following is required when doing Sheffe:

1)Analysis 1 must be performed. If the Anova to be used is a twoway Anova and there is missing data, the means must only be calculated on full rows (any row with missing data is excluded). This is signalled by requesting analysis 1,15.

2)The relevant Anova must be requested. If there is no missing data, analysis 1 or 1,15 may be used with either Anova, but if there is missing data, it is imperative to use the pairs STAT6-STAT1 or STAT8-STAT1,15.

3)Finally, analysis 7 is requested and the P-values have to be supplied to the program in one of two ways:

CARD TYPE

A)If ARG is non-zero: The next card should contain as many P-values as ARG (max=3) separated by commas with the largest first. The P-values should be in the form 0<P<0.5 or a specification error will be generated.

B)If ARG is zero: The program uses the internal default values of 0.05,0.01,0.001.

The output is in the form of two matrices - one showing those contrasts exceeding the given limits and one showing the exact probabilities for the contrasts.

13 Wilcoxon Calling 13,K1,K2

14 Mann Whitney Calling 14,K1,K2

15 Kruskall Wallis. If there is no missing data or if 15,1 is requested, the table of rankdifferences is also output (in the latter case by ignoring any row in which any data is missing). See Colquhoun.

16 Spearman rank Calling 16,K1,K2

17 Friedman analysis. The rank differences are calculated.

18 Numerical integration by overlapping parabolas.
K1 is used as X and K2 as Y.
If ARG1=0 only final area.
If ARG1=1 also tabulates all partial areas.
The areas are normally calculated by a method of overlapping parabolas. Although a very good method, it fails miserably on some kinds of experimental data. A check on this has been incorporated. Areas are also calculated by a Trapeze formula and the two areas are compared.

19 Medians and rank of column
IOPT=0

Medians are calculated once on each column considered separately and if missing data also once on each column skipping any data in a row where there is missing data. This is equivalent to analysis 1,15.

CARD TYPE

0<IOPT<=NK

Column IOPT is ranked and the ranks are printed for each observation. Also printed are the two tiesums (used for nonparametric analyses).

TIESUM1=SUM(CT**3-CT)/12
TIESUM2=SUM(CT*(CT-11)/2)

For usage check e.g. Siegel.

20 Crosstabulation of 2 columns or tabulation ("Numeric Histogram") of one. The latter specified by KOL1=KOL2. The number of intervals can be 3 to 10 with 5 as the default.

21 Scatter plot. KOL1 is X-axis and KOL2 Y-axis. X-axis is always divided into 50 steps. Y-axis can be divided into 10 to 50 steps specified by IOPT. Default is 25 steps.

As logarithmic plots are common, a special call has been incorporated to generate E-logs of the X-axis. This is done by adding 100 to the IOPT argument i.e. 21,1,2,110 request a plot with 10 steps on the vertical axis and logs on the horizontal. Also note that logs can be generated by option 22.

This plot is normally scalled so that the largest X Y pair is placed in the upper right hand corner and the smallest in the lower left hand corner. It is possible to use another scaling of the plot. By negating ARG1 (e.g. stating -25), it is possible to request the co-ordinates of the lower left hand and the upper right hand corner to be taken from the input file. They must follow the 21 card on the next line.

If working from keyboard they are specifically requested. They are given in the order X0,Y0,X1,Y1 as the co-ordinates of the two points. Note that it is impossible (due to the format used in reading the request cards) to input a four figure option. It is, therefore, impossible to request manual input and logarithmic scaling at the same time. In this case you must first convert the data to logarithms and then do the plot.

22 Transformations on the data in core. The original data file is not changed. When so is required use KSORT. Analysis 22 only handles transformations where one input column is involved. If more are involved, e.g. KOL1 in % of KOL2, the transformations have to be done via KSORT.

CARD TYPE

The calls are:

22,KOL1 Sorts the data matrix by KOL1. Missing (zero) values are always placed last.

22,-KOL1 Sorts by KOL1 in descending order.

22,NK,KOL1,-1 Reduces data matrix from NK columns (must be given) to KOL1 columns. Only valid if the ATT switch (see the 49 card) is set. Can be used instead of re-reading the data with fewer columns when the data in core is too big for a required analysis.

22,KOL1,KOL2,CODE Sets KOL2 to code (KOL1).

The codes are:

1 SQUARE ROOT
2 EXP(E**X)
3 NATURAL LOGS
4 LOGS
5 SQUARES
6 EQUAL i.e. copies KOL1 into KOL2.

48 Output a comment. Sometimes it is useful to be able to pass a comment to the output file on unit 3, e.g. when a standard parameter file often is used on different data. Stating 48, IOPT will prompt the program to ask for IOPT cards from the input file (unit 4). If the ATT flag is set (see 49), the program prompts the operator to give the comment line by outputting a number to the TTY (unit 0) so as to minimize the risk of setting a letter in a numeric field on manual input.

49 Flag setting routine. The program uses 3 flags and 2 device numbers to schedule certain parts of the information. These are:

LIST If true - print the trace information. If the program is doing a long unattended run with the printer output directed to a file for future listing, it is useful to be able to see where the program crashed if it has done so. Requesting the trace information makes it possible to see where and when such a program crash has occurred.

CARD TYPE

ATT — If false - the program assumes that it is running unattended.

The following are the actions taken by the settings of the ATT switch:

TRUE	FALSE
KEYBOARD INPUT ALLOWED	NOT ALLOWED
PROMPTS "ANALYSIS NO"	NO PROMPTING GIVEN

LIST — Is false at program start.

ATT — Is true if unit 4 is keyboard, false if unit 4 is assigned to a file.

LDEVE — Is the device for error texts . It is defaulted to 3 when input requests from file and to 4 when keyboard input.

LDEVL — Is the device for the normal program output. It is defaulted as specified earlier.

NOPAGE — The program normally gives a new page for each request for analysis 0 and plot. If a large number of groups are to be repeated with e.g. only means, this takes a lot of paper. Setting NOPAGE to true, inhibits the pagination.

These 5 bits of information can be changed by the 49 card.

Note: The 49 card is read as an analysis request and must, therefore, not come before the initial requests for groups and columns. The format is:

49,ATT,LDEVL,LDEVE,NOPAGE,LIST.
ATT,LIST and NOPAGE are set to true if they are equal to 1 on the 49 card, else they are set to false. The 49 card can be repeated at any time.

LDEVL, LDEVE can be changed by this card. They can, however, not be changed to anything but 3 and 4. If they are zero on the card they are ignored. If anything but 3, 4 and 0 is input, they are defaulted to 4 so as to avoid an unintentional request for line printer usage, which might hang up the program.

E)RUNNING STAT

STAT is available as loader image and run by:

```
.EX STAT
*INDATAFILE/6
*PARAMETER FILE/4
*LPT:/3              OPTIONAL
$
```

Some timing considerations: Stat is runnable in any 12K OS/8 configuration but it is slow if no FPP and disk is available. Due to the heavy overlaying, it is very slow in a Dectape system. If possible, try to rearrange the modules for faster calling if you have such a system (i.e. utilize more than 12K of core).

F)CAPACITIES

In 12K core maxxl=1000.
Columns in the file max 10.
Total data in core (columns * rows) < maxxl.

For non parametric analyses:

Given that
NK=NO of columns in core
NR=NO of rows in core

So must for routine		This product be less than maxxl
4		(NK+2)*NR
13		(NK+5)*NR
14		(NK+4)*NR
15	MISSING.AND.15,1	(NK+2*NK)*NR
	ELSE	(NK+NK)*NR
16		(NK+5)*NR
17	MISSING	(NK+2*NK)*NR
	NO MISSING	(NK+NK)*NR
18		(NK+4)*NR
19		(NK+1)*NR
20 21		(NK+2)*NR

G)ERROR HANDLING

ERROR CONDITION	HANDLED	MESSAGE
A) On reading input data parameters:		
COLUMN OR GROUP REQUEST OUT OF RANGE	JOB ABORTED$	YES
TOTAL DATA>MAXXL	JOB ABORTED$	YES
B)On reading the analysis requests:		
NON-EXISTING ANALYSIS REQUESTED	IGNORED	YES
ANALYSIS REQUIRES COL NR THESE ARE WRONG <TOO LARGE OR .LE.0>	IGNORED	#
ANALYSES REQUIRES A 3RD ARGUMENT BUT THIS IS MISSING	IGNORED	#
C)In specific analyses:		
Beside these there are some selfexplanatory messages.		
NR		
-1 ARGUMENT OUT OF RANGE	IGNORED	#
BAD FORMAT ON 0	FRTS ABORTS PROGRAM	
1 ARG OUT OF RANGE	IGNORED	#
2 CODE WRONG	IGNORED	YES
2 TOO FEW DATA	IGNORED	YES
3 TOO FEW DATA	IGNORED	YES
4 BAD P VALUE	SKIP 4	#
7 "	SKIP 7	#
5,6,8 EMPTY COL	IGNORED	YES
13 N<6	SKIPPED	#
15 NK<4	SKIPPED	#
16 N<4	SKIPPED	#
17 NK<3	SKIPPED	#
18 N<3	SKIPPED	#
22 ON BAD ARGUMENTS(<0)TO SQUARE ROOTS OR LOGS THE VALUE IS SET TO 0.		
MANY ROUTINES ON EXCEEDING CAPACITY	IGNORED	#

#All these are signalled by the text:
Specification error in stat## where ## is the module number.

All these texts are sent to device LDEVS.

In some places, a divide by zero might be produced when it is not guarded for.

Note that line numbers are irrelevant (all modules compiled with /N).

$If the Job is aborted, the program searches the parameter file for a 50 card or a 0 card so as to <if possible> continue with the next indata request.

H)LOADING

The loading information is in the file STATB.BI for use by batch. The file STATC.BI is a batch file for compiling the necessary routines (note: this file is OS/9 V 3 format).

Note that this file also presumes that the routines CMDOPT and FILSIZ are available in FORLIB. If they are not, you must re-edit the batch files so as to include them. They must be included at level 0.

Note that all modules are compiled without line numbers. The program requires 12K. But in 12K maximal usage is impossible (running under Batch Plus several files goes over 12K). If the program is loaded without overlays, it takes about 35K. In a 16K Dectape system you are adviced to move some of the routines now at level 2 to level main. This saves a lot of overlaying core.

If you want to remove routines from the program <if you feel it is unnecessarily large>, study the transfer table in SORT. You can inhibit calls to the various modules by changing the values in the data list in SORT. Note that the overlays are very compactly packed. To increase the package a good deal of thought would be required.

I)TESTDATA

Testdata to the program is in STATDA.DA(/6).
Using STATST.DA as unit 4 will test most of the routines in the package.

J)REFERENCES:

Besides those mentioned in the general introduction the following books have been referred to in this package:

D.COLQUHOUN (C) — Lectures on Biostatistics 1971

M ABRAMOWITZ & I STEGUN (A) — Handbook of Mathematical Functions 1964

K)SUBROUTINES

The statistical package uses a number of subroutines that are of a more general nature and are as such not considered part of the package. These are:

1)A number of routines from the SSP library (Scientific Subroutines). See the DECUS library for further information. These routines are:

UTEST (performs Mann Whitney Utest)
MPAIR (performs Wilcoxons Matched Pairs rank sign test)
SRANK (performs Spearman rank)
TWOAV (performs Friedman Analysis)
and the general routines TIE, NDIR and RANK.

2)Several generalized subroutines programmed from various sources:

NUMINT	Numerical Integration (D&R)
PVALUE	Probability Function (adapted from DEC-10 library.
FINV,TINV,ZINV	Probability Functions (A)
FILSIZ	A RALF function for finding the size of input files.
CMDOPT	A RALF function to enable a FORTRAN program to see a switch given to the Run Time System. This subroutine requires a small patch to the Run Time System. This patch together with the error trap patch from the OS/8 Newsletter are available in the file FRTSOV.V3 which should be overlaid the FORTRAN Run Time System.

The other subroutines are specific to the package. All routines specific to the package have names in the form STAT??.FT.

MARCH 15 1976

HASSLE STAT PACK
FORTRAN IV - OUTLAY

Lars Palmer
AB HASSLE
Fack
431 20 Molndal 1
Sweden

A)GENERAL

OUTLAY is used whenever a data file to be used by the Statistical Packase is to be created from the keyboard.

B)INDATA AND FILES

OUTLAY accepts as indata a keyboard input and converts these into a formatted file usable by FORTRAN programs. The standard mode of OUTLAY will only accept the keyboard input. However, a special option exists which allows it to also accept a previously created file and correct data in this file or add more data to it.

File format:

The file consists of 2 parts:

1)A number of headings for the material.
2)The actual data.

The data can be organized in up to 10 columns (e.g. time and measurements). The data can also be divided into a number of groups (up to 100). The programs will use these group numbers when reading the data file. Associated with each group number is a heading describing the group. The program also provides one main heading for the whole material.

The file units are:
Number 5 <must always be defined !!!> output file for data. If the extended mode of OUTLAY is requested by option so can numbers 6-9 be used for any of the possible types of input:
Headings, formatted data <including headings>, or unformatted data.

C)THE DATA FILE FORMAT

The file created with OUTLAY has the following format:

CARD NO	CONTAINS
1	NR GROUPS NR COLUMNS (2I3)
2	MAIN HEADING(3A6)
3-NK+2	1 HEADING FOR EACH OF NK GROUPS
NK+3..	DATA IN FORMAT I3,7(E11.4)
LAST 2 CARDS	BLANK.

D)RUNNING OUTLAY

D1)Simple input case:

```
.EX OUTLAY
*FILE</5  See files above.
$ALTMODE  To start program.

Input data is in the format:
G,X1,X2,X3 With G being the group number and X1..XNT the data
           or
X1,X2,X3   (see below).
```

The input to the program is interactive. You are asked for the input data and you supply them at the request of the program. The first information asked for is the number of groups and the number of columns that are to be used in the new file. The answer is e.g. 3,5 meaning 3 groups and 5 columns. You are then asked for a main heading which can be up to 72 characters long. Once this is input the program goes to the data input routine. The data is input sequentially for each group with, in the example case, five data on each line separated by commas. Each group is terminated by striking ctrl/Z on the keyboard and the machine will then ask for next data group.

After the final data group, the program enters the error correction routine.

Error Correction Routine

Errors can be corrected in several ways. The positions on the error correction lines are:

ROW NUMBER, COLUMN NUMBER, DATA.

If col number = 0 the row is given group number = data.
If only row number is given, and the rest of the line is blank, the whole line is set to zero (which in the Statistical Package is treated as a missing observation).
If col number > 0 this column (and the following) are given the value = data.

End of corrections is signalled by a blank line
e.g.

```
1,0,5              change row 1 to group 5
1,0                remove row 1 (all data set to 0)
1,4,15             change row 1 col 4 to 15
3,4,1,2,3          change in row 3 col 4-6 to 1,2,3
```

LISTING

After end of corrections, the program asks for listing. Listing is requested by the line:
L,U,D
If U=0 no list done.
If L>1 the list begins at line number L.
If U=0 the listing ends with last line of the buffer.
If U>0 the listing ends with line number U.
If D>0 listing is to line printer else to console.

The program iterates the questions:

"LIST","ANY ERRORS" until the last is answered with no (0!). It then writes the data to the output file and the program exits to the OS/8 monitor.

D2)Full Syntax

.EX OUTLAY
*FILE</5
*FILES/N
*/A$

Full syntax is requested by the /A option on the last line to the command decoder. With this option a large number of various ways of inputting data to the OUTLAY program are implemented. The following is a description of the various inputs that the program expects.

1)The program first asks for the possibility of taking the headings and group numbers from an existing file. If you have many groups, you are adviced to put the headings in a file by using an editor before running OUTLAY. If you do not have a file, the program requests numbers of groups and columns directly followed by NK+1 headings on separate lines. Optionally group numbers can be given negative (-NG,NK). In this case main heading is given from console and subheadings are generated by OUTLAY in the form "Group nr N" (this latter form of internally generated group headings is always used in the reduced option case). On "Wait" the program does either directory or file work. Wait until it requests more input. The program asks for a possible input file (this input file refers to a previously by OUTLAY created file, or an equivalent file created by another FORTRAN program. You may optionally put the unformatted input to OUTLAY in a file, but this should then be read under the options in paragraph D2:3). If you have none, so to next paragraph. The input file must be a properly formatted OUTLAY file.

2)After reading the file (or on full buffer, see errors), the program asks for any error corrections (see error correction routine). After error corrections the programs asks for a possible listing, (see listing) and then writes the data to the file. The program then repeats the question "Input file" until this question is answered by no (0).

3)No input file (or all input files read). The program now asks for new unformatted input. This can be from the console or from a file (e.g. an editor prepared data file). In the first case, the answer must be 4 (=TTY) and in the latter case the file should be stated. Input to the program can take two main forms: either the input can be available in such a way that the groups are separated e.g. we may have one sheet of paper for each group that is to be used in the further analysis, or the groups may be intermixed e.g. the first line on each paper may be group no. 1. In the first case use the input format described under 4 and in the second case the input format described under 5.

4)Input in sequential groups with all material in one group collected together.

Answer the question "More input" with either -unit or -unit,group. Unit is 4 for teletype or 6 -9 if you have defined an input file. The data is then in the format described under the reduced option (D1) except that ctrl/z cannot be put into a file; instead you may use 1E10 or a larger number as the X value in the first column. If the group number is not stated, the program assumes the input data to contain as many groups as you have stated that the total input is to contain. It will read data from the input file or from keyboard until the total number of groups have been satisfied or until the current core buffer contains 100 rows. In either case it will go to the error and list routines and depending on if all input was given or not, it will continue to accept data where it left off or it will ask you for how to proceed. The -unit option may not be used if any previous input data has been fed to the program i.e. if it has read data from a file or if data has been fed from the keyboard.

The -unit,group can be used at any time. It signals that the data which now is to be fed in from unit is to go into the stated group number until it detects an end of group signal. In this case the "More input" question will come when the end of input is signalled.

5)In the other case when the input data from the different groups is mixed, it is simplier to input the data with the group number signal associated with each input line. This input is requested by unit number positive. Unit can be 4 for teletype input or 6 - 9 for file input. In either case the input format is the same. On each line first the group number followed by the data values for that line. End of input in this case is signalled by group number 0 (equivalent to simply striking a blank line). In this case also the program goes to the "More input" after having processed the errors and list routines.

ERRORS IN INPUT

All input lines where input can come from the terminal are safe guarded by calls to the error checker routines. So as not to allow this to be nullified by an error in unformatted input from another unit, the program will not allow reading from another unit once teletype unit has been used, i.e. You cannot use the sequence -4 followed by some more data from e.g. unit 6. If you want to take an unformatted file from unit 6 and add to it from the keyboard, you must read the data file first and then input the data from the keyboard. In this way you will not come to the situation where you have fed in a large number of data from the keyboard and the program crashes due to a numeric error in an input file.

E)CAPACITIES

Columns max 10.
Groups max 100.
Rows in core max 100.

The current buffer must be edited and listed and written out before more can be entered.

F)ERROR HANDLING

1)Too much data - "Buffer full" and edit this data.
2)Input group NR>NG - "Wrong".
3)Non-integer group number - (normally = missing GR NR) "Wrong". If reading from a file the erroneous line is printed.

4)Input file contains more groups or col than stated. Excessive ignored.

5)Input requested from file after keyboard input the error message "Only TTY allowed" will appear. If you come to this situation you will have to write the present data out and restart the program reading in the file data after the 4 you do keyboard input.

6)As mentioned above all possible ways of keyboard are guarded by calls to the error checker routines. This leads to several error messages which can appear if there are numeric errors in the input i.e. two decimal points, a mistake typing a letter instead of a number or anything of this kind.

G)LOADING

Compile standard F4, requires 12 K.
Requires the CMDOPT function and its FRTS patch and the error checking routines from Jim Crapuchettes as published in the OS/8 Newsletter. Both routines are on the source tape for the Stat Package.

HASSLE STAT PACK
FORTRAN IV-KSORT

Lars Palmer
AB HASSLE
Fack
431 20 Molndal 1
Sweden

HASSLE STAT PACK
F IV - Ksort

A major revision of KSORT was done in October 74.

Functions of KSORT.

KSORT will do the following:

1)Rearrange the material in an OUTLAY file to allow for the comparisons in Stat of e.g. column 1 group 1 against column 1 group 2.
2)Convert material in a given column to e.g. roots.
3)Sort the file according to any given column.
4)When interfaced with a user written subroutine splitting or combining groups or excluding certain data according to (more or less) any criteria.

A)INDATA AND FILE UNIT NUMBERS

UNIT	CONTAINS
6	INDATA FILE OUTLAY FORMAT
5	PARAMETER FILE SEE B)
3	SUMMARY OF THE NEW FILE
7	THE OUTPUT FILE AS REQUESTED
0	ERROR MESSAGES

B)THE PARAMETER FILE

The parameter file (unit 5) has the following format:

CARD NO CONTAINS:

1 NG,NK,FILL,SORT,EXTERN,LIST,MISS.
Written in inteser format separated by commas.

NG — No of groups in new file can be in the forms:

- >0 Used as stated. NG subheadings must follow.
- =0 Use the number of groups and headings in unit 6.
- <0 Use absolute (NG) groups, generate headings in the program.

NK — Number of columns in new file. If NG is set to 0 or < 0, NK can be set to 0 in which case columns are taken from unit 6.

FILL — If set to non-zero, any group-col not specifically requested by parameter cards is to be copied over from input. If set to zero, such columns will be set to "Missing".

CRD NO	CONTAINS	
	SORT	If >0, the material is sorted accordins to col "Sort". N.B. under Extern.
	EXTERN	If set to non-zero, EXTERN is to be called. See this.
	LIST	If set to non-zero, the effects of the operations of KSORT are summarized on the printer (unit 3). See also under Extern.
	MISS	If non-zero, any row with a missins observation is deleted.
2		Main headins new file.
3 TO NK+3		One headins for each sroup in new file. Headinss are conditional on the value of NG. Headins cards must not be included if NG is set to 0 or <0.
NK+4 TO L-1		Request for make-up of new file. See below.
L		Last card must be blank.

Request cards are made up as follows:

POSITION NR	CONTAINS
1	NEW GROUP NR **
2	NEW COL NR **
3	OLD GROUP NR
4	OLD COL NR
	<i.e. 1,3,2,4 means place old group 2 col 4 in new group 1 col 3>
5	CODE &
6	OLD COL 2 &
7	CONSTANT &
	i.e. 1,3,2,4,6,3,1.2 is: use old group 2 cols 4 & 3 combining by operation code 6 and constant 1.2

& See XFUNC below.
** Either or both of the new group number and new col number can be replaced by -1. This leads to the corresponding information being matched by all in the input file. The corresponding "New" is ignored and can be set to 0.

C)XFUNC

The subroutine XFUNC is called for every data with code <>0. It implements the conversions detailed in the source code for the subroutine. See appendix 1.

D)EXTERN

The subroutine EXTERN must be added by the user if EXTERN is non-zero. It is called just before writing the row to the output file. It can be used to implement any kind of logic not possible by XFUNC.

```
SUBROUTINE EXTERN (NG,X,NKO,SW,TWICE,COUNT,SKIP)
LOGICAL SW,TWICE,SKIP
INTEGER COUNT
DIMENSION X(1)
```

The parameters are:

NG	Current group number.
X	The vector of X-values to be output.
NKO	Number of values in X **Must not be changed**
SW	Set to false if the data is not to be written.
TWICE	Set to true if the subroutine wants to be re-called for this data e.g. if it is to be duplicated.
COUNT	Is a counter of the number of times the routine has been called for this data.
SKIP	If set to true inhibits further calls to Extern in this data group.

N.B. If the routine changes the size of NG, it is impossible to do a sort. In this case also the summary produced by the list request is in error. For these reasons sort and list are negated if NG is changed. If NKO is changed the program stops.

See appendix 2 for an example of the usage of the routine.

E)RUNNING THE PROGRAM

KSORT is available as loader image and run by:

```
EX KSORT.LD
*PARAMETERFILE/5
*LPT:/3            Optional if lineprinter is available (only
                   used if list is true).
*OLDDATA/6
*NEWDATA</7
*$
```

Note that if EXTERN is to be used you must:

1)COMPILE EXTERN
2).EX TEMP<DTA0:KSORT.RL,XFUNC,EXTERN

EXTERN and XFUNC may overlay each other but will probably not have to do so.

F)CAPACITIES(16K)

Columns in the new file max 10.
Groups in the new file max 100.
Each group in the new file max 2000 observations.
Reducing data area to 1000 allows operation in 12K.

G)ERRORS
The followings errors are detected and handled as noted:

ERROR	ACTION	MESSAGE
GROUP>100	STOP	SPECIFICATION ERROR
COL>10	STOP	SPECIFICATION ERROR
GROUP OR COL REQUESTED OUT OF RANGE	IGNORED	NO
TOTAL DATA IN GROUP>1000	GROUP IGNORED	YES
SORT<0 OR SORT>NK	STOP	SPECIFICATION ERROR
NK CHANGED BY EXTERN	STOP	NO
NG CHANGED BY EXTERN: LIST DISABLED IF REQUESTED		YES
SORT DISABLED IF REQUESTED		YES

Appendix 1
Examples

A)Use old groups and generate 2 columns (4 & 5):

Col 4 = The difference between col 1 and 2.
Col 5 = The difference between col 1 and 3.

This is to be done for all groups:

```
0,5,1,0,0,1
-1,4,0,1,1,2,-1 /col 4=col 1+col 2*-1
-1,5,0,1,1,3,-1
00
```

B)Sort each group in a file according to col 3.

```
0,0,1,3
000
```

Appendix 2
Having a data file with 8 groups and 3 col we want to:

A)Make a group 9 = groups 1+2+3.
B)Make a group 10 = all with COL 1=col 2.
C)Unconditionally exclude all rows with any data >160.
D)For the so generated data:
  COL 4=LOG COL 2
  COL 5=SQRT(COL 3)
  and the data finally sorted according to COL 1.

This is done in 2 passes:
A)Load KSORT,XFUNC and the following subroutine:

```
            FUNCTION EXTERN(NG,X,NKO,SW,TWICE,N,SKIP)
            LOGICAL SW,TWICE
            DIMENSION X(1)
            GOTO(10,20,30),N
10          SW=.FALSE.
            IF(AMAX1(X(1),X(2),X(3)).GT.160)RETURN
            SW=.TRUE.
            TWICE=.TRUE.
            RETURN
20          IF(NG.GT.3)GOTO 30
            NG=9
            TWICE=.TRUE.
            RETURN
30          NG=10
            IF(X(1).NE.X(2))SW=.FALSE.
            RETURN
            END
```

B)
This is then run with the unit 5 file:

```
-10,0,1,0,1         /10 COL AUTO HEADINGS FILL OUT. NO SORT.
                    /CALL EXTERN
TEST                /MAIN HEADING
0000                /END
TEST A
```

C)The so produced file is then run with the following unit 5 file:

```
0,5,1,1,0,1         /HEADINGS & NG FROM UNIT 6. 5 COL.
                    /FILL OUT. SORT ACC COL 1. NO EXTERN .LIST.

TEST B              /MAIN HEADING
-1,4,0,2,7          /ALL GROUPS COL 4 FROM COL 2 CODE 7=LN
-1,5,0,3,9,,0.5     /ALL GROUPS COL 5 FROM COL 3 CODE 9
                    /CONST:0.5   CODE:9 X=X**CONST
000
```

This gives the desired result.

Appendix 3
Listing of XFUNC follows:

```
  OS/8  FORTRAN IV  3.02                    OCT 18  1974

          C       XFUNC
          C      73-10-24
0002              SUBROUTINE XFUNC (X,XIN,IARG,K)
          C       TO IMPLEMENT THE CONVERSION CODES LISTED
          C       CODE    RESULTS IN
          C       0       NO CHANGE
          C       1       X=Y+K*Z<Y=XIN(IARG(2)),Z=XIN(IARG(4))>
          C       2       =0
          C       3       X=Y*Z**K
          C       4       =0
          C       5       X=Y/Z*100
          C       6       X=((Y-Z)/Z)*100
          C       7       X=LN(Y)
          C       8       X=LOG(Y)
          C       9       X=Y**K
          C       10      X=Y*K
          C       11      X=Y+K
          C       12      =0
          C       13      =0
          C       14      =0
          C       15      =0
          C
          C       ANY CODE OUTSIDE 1-15 IS TAKEN AS 0
0003              DIMENSION XIN(1),IARG(1)
0004              REAL K
0005              COMMON NKO
0006              DIMENSION ITYPE(15)
0007              DATA ITYPE/4,0,4,0,2*2,1,1,3*3,4*0/
          C       ITYPE CONTAINS:
          C       A 0 IF CODE IS UNDEFINED DEFAULTS TO 0
          C       A 1 IF CODE IS A FUNCTION OF X
          C       A 2 IF CODE IS A FUNCTION OF Y Z
          C       A 3 IF CODE IS A FUNCTION OF Y K
          C       A 4 IF CODE IS A FUNCTION OF Y,Z,K
0010              IY=IARG(2)
0011              IP=IARG(3)
0012              IF(IP.LE.0.OR.IP.GT.15)GOTO 300
0013              IT=ITYPE(IP)
0014              Y=XIN(IY)
0015              IF(IT.EQ.0.OR.Y.EQ.0)GOTO 300
0016              IF(IT.EQ.1.OR.IT.EQ.3)GOTO 100
0017              IZ=IARG(4)
0020              IF(IZ.GT.NKO.OR.IZ.LE.0)GOTO 300
0021              Z=XIN(IZ)
0022              IF(Z.EQ.0)GOTO 310
0023      100     CONTINUE
0024              GOTO(1,300,3,300,5,6,7,8,9,10,11,
                  300,300,300,300),IP
0025      1       X=Y+K*Z
0026              GOTO 200
```

```
0027        3       X=Y*Z**K
0030                GOTO 200
0031        5       X=Y/Z*100.
0032                GOTO 200
0033        6       X=((Y-Z)/Z)*100.
0034                GOTO 200

0035        7       IF(Y.LT.0)GOTO 300
0036                X=ALOG(Y)
0037                GOTO 300
0040        8       IF(X.LT.0)GOTO 300
0041                X=ALOG10(Y)
0042                GOTO 300
0043        9       X=Y**K
0044                GOTO 300
0045        10      X=Y*K
0046                GOTO 300
0047        11      X=Y+K
0050                GOTO 200
0051        200     IF(X.EQ.0)X=1.E-25
0052        300     RETURN
0053        310     X=0
0054                END
```