RESEARCH REPORT

SURVO 76 EDITOR, A NEW TOOL FOR INTERACTIVE STATISTICAL COMPUTING, TEXT AND DATA MANAGEMENT (RELEASE 2)

BY

Seppo Mustonen

No. 24

February 1981

DEPARTMENT OF STATISTICS
UNIVERSITY OF HELSINKI
SF 00100 HELSINKI 10 FINLAND

ISBN 951-45-2209-5
ISSN 0357-9778

CONTENTS

INDEX

1. Introduction

SURVO 76 is an interactive statistical system intended to cover a wide range of activities in computational statistics. It has been designed especially for the needs of statisticians in both teaching and research work and its aims are slightly different from those of conventional statistical packages generally available for data analysis. In a certain sense the scope of SURVO 76 is wider permitting extended possibilities for data and text management, simulation, matrix computations and graphical analysis in interactive mode (see Mustonen 1977, 1980a,b).

In the present form SURVO 76 has been implemented on the desk computer Wang 2200VP which provides suitable means for rapid interchange of information between the system and the user.

SURVO 76 is an interactive system and no special job describing language or code is needed. Using this system is like discussing with the computer; we speak about SURVO 76 conversations. The discussion is transmitted from the system to the user by a fast CRT display and from the user to the system via a keyboard having also 32 programmable soft keys for control tasks. For a more precise and detailed output a line printer, a graphic CRT and a plotter are available.

SURVO 76 consists at the moment of about 60 statistical subsystems and programs and the total volume is almost 1 million bytes of program text. Formally the whole system is a single program written in the extended BASIC language of Wang.

In this paper we shall describe a new subystem SURVO 76 EDITOR which is intended for various text and data editing activities in connection with statistical data analysis.

It is quite common that when writing a research report containing numerical tables the output from the computer cannot be used as such, but the results have to be retyped manually. This may happen even if the computer output is well designed, since the needs of the user may change during the reporting phase. In an interactive environment a good way of avoiding those editorial problems is to have text processing facilities in connection with the statistical system.

The main purpose of EDITOR is to lessen the burden of a statistician in data management and report writing. This editor can be used not only for standard text processing purposes, but also for various tasks encountered in statistical data processing like

1) input and editing unformatted data,
2) saving data in SURVO 76 files,
3) editing SURVO 76 files and results,
4) sorting and transforming data,
5) arithmetic and statistical computations,
6) manipulating lists and multiway tables,
7) data analysis using techniques suitable in editing mode.

The whole editing process is controlled with the normal keyboard keys and programmable 'soft keys' (F-keys) which are used for simple text editing. For more complicated tasks several editing operations are available.

All the information is represented in an edit field which consists, for example, of 100 columns and 250 rows. The field is always partially visible on the CRT which is like a window to the field. The user can easily scroll the text on the screen to any direction by pressing certain F-keys with arrows indicating the direction. The editing operations are also typed in this field and they can be treated as normal text. Any operation can be activated by moving the cursor to the corresponding line and by pressing key CONTINUE. Whenever needed the contents of the edit field (tables, text and operations) can be saved in an edit file.

The edit field is like a notebook for the user, but it is much more flexible, since text and data in that notebook can be worked upon by editing operations and the results of these operations can be directed to any part of the field. Since the editing operations themselves are typed among the text and data the user can place them near the object of operation.

If the user likes he can put the operations on adjacent lines and carry them out step by step as an editing program, but usually this is unnecessary. On the contrary, it is typical that during the editing process the edit field is filled by a mixture of text, data and operations, and the user scratches unessential ingredients when needed.

In the next chapters we try to illustrate with some practical examples how to use EDITOR for typical data and text processing activities. In chapter 2 we give an introductory example related to the analysis of contingency tables. The fundamentals of standard text editing are also briefly described. Chapter 3 introduces the tasks of the editing keys and chapters 4,5 the editing operations for normal text and data management. In chapters 6 and 7 some statistical operations are described.

Release 2 of SURVO 76 EDITOR

After the first report (Mustonen 1980c) EDITOR has now been extended in several ways. The most important novelty, to be introduced in chapter 8, is the possibility of performing arithmetic computations in the edit field. The computations may involve symbolic notation and various mathematical and statistical functions. This approach also permits construction of computation schemes which are like simple computer programs in an exceptionally free and natural form.

Release 2 of EDITOR includes some new operations (COMP, CUMUL) and several extensions and amendments in old ones. When displaying larger data sets in the field, it is possible to split the CRT into two parts and see, for instance, the top and the bottom of a long list simultaneously (p.14). Also new control characters are available in printing.

In chapter 9 some non-statistical special applications of the editorial approach are briefly described.

2. An introductory example

When SURVO 76 is in use EDITOR will be called by the name E in the same way as other SURVO 76 modules. At first a list of possible F-starts is displayed on the screen.

Disp.2.1

```
        SURVO 76 EDITOR  (C)1979 S.MUSTONEN, UNIVERSITY OF HELSINKI

F1: BASIC START
F5: SAVE TEMPORARY FILE
F6: LOAD TEMPORARY FILE
F8: LOAD 'INDEX' FILE FROM THE DATA DISK

EDIT FIELD: 100 ROWS, 100 COLUMNS    (MAX 262 x 100)
```

When starting with a new job the basic start F1 will usually be employed at first and then the upper left side corner of an empty edit field is displayed on the 24x80 screen:

Disp.2.2

```
      1  SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
  1  *_
  2  *
  3  *
  4  *
  5  *
  6  *
  7  *
  8  *
  9  *
 10  *
 11  *
 12  *
 13  *
 14  *
 15  *
 16  *
 17  *
 18  *
 19  *
 20  *
 21  *
 22  *
 23  *
```

The cursor is blinking in the first position of the edit field and the user can start typing text, data values and editing operations.

When the edit field is displayed the system can be operated like a normal typewriter and the text appears continuously on the screen.

Correspondingly the extra 'the' on line 10 is removed by moving the cursor to that line at the second 'the':

Disp.2.7

```
 5    *   Example 1: Log-Linear models for Contingency tables
 6    *
 7    *   The following data are taken from a survey of the
 8    *political attitudes of a sample of British electors
 9    *which is reported in Butler and Stokes (1974).
10    *We examine the the relationships among four variables:
11    *vote, sex, class and age.
12    *
```

and pressing key F9:(delete) four times:

Disp.2.8

```
10    *We examine the relationships among four variables:
```

For the correction and editing purposes EDITOR provides several means controlled by F-keys. A thorough description of them will be given in chapter 3.

Besides the use of these 'soft keys' EDITOR includes many editing operations for more complicated tasks.

For instance, if we like to make the text in the edit field wider so that the present lines 7-11 should be up to 65 characters in length we achieve this situation by typing operation TRIM 7,11,65 on some empty line in the field (line 13 in this case):

Disp.2.9

```
 5    *   Example 1: Log-Linear models for Contingency tables
 6    *
 7    *   The following data are taken from a survey of the
 8    *political attitudes of a sample of British electors
 9    *which is reported in Butler and Stokes (1974).
10    *We examine the relationships among four variables:
11    *vote, sex, class and age.
12    *
13    *TRIM 7,11,65_
14    *
```

and by now keeping the cursor on line 13 and pressing key CONTINUE we obtain the following result

Disp.2.10

```
 5    *   Example 1: Log-Linear models for Contingency tables
 6    *
 7    *   The following data are taken from a survey of the political
 8    *attitudes of a sample of British electors which is reported in
 9    *Butler and Stokes (1974). We examine the relationships among four
10    *variables: vote, sex, class and age.
11    *
12    *TRIM 7,11,65
13    *           _
```

Since the new widened text requires one line less than the original one our TRIM operation locates now on line 12.

We may continue this trimming process and make the right edge of the text on lines 7-10 even by inserting extra blanks between some words. This is accomplished most easily by editing the TRIM operation on line 12 in the following form (only '2' has to be inserted):

Disp.2.11

```
12    *TRIM2_7,11,65
```

and activating this operation again with CONTINUE:

Disp.2.12

```
 5   *   Example 1: Log-Linear models for Contingency tables
 6   *
 7   *   The following data are  taken from a  survey of the  political
 8   *attitudes of a sample  of British electors  which is reported  in
 9   *Butler and Stokes (1974). We examine the relationships among four
10   *variables: vote, sex, class and age.
11   *
12   *TRIM2_7,11,65
13   *
```

Observe that the editing operations are typed and edited like normal text, but each line containing text can be <u>activated</u> by moving the cursor to that line and by pressing CONTINUE. Then, if the line can be interpreted as a valid operation, this operation will be carried out, but if the line contains text which does not correspond to an editing operation, nothing will happen.

Activation can take place many times, since the operation text remains in the field until it is overwritten or removed by keys F25:(delete line), F8:(erase) or by editing operations like CLEAR and SCRATCH.

The position of the operation with respect to the object of the operation is immaterial in principle. In practice, however, it is typical to place the operations on lines close to the text and data to be handled. On the other hand it is sensible to collect some general operations like PRINT (printout of selected lines or chapters), SAVE (saving the edit field) and LOAD (loading some related fields) to the first lines of the field.

If we now like to save the current situation in the edit field on disk we can type the operation SAVE ELECTORS on the first line:

Disp.2.13

```
 1   *SAVE ELECTORS_
 2   *
 3   *
 4   *
 5   *   Example 1: Log-Linear models for Contingency tables
 6   *
```

and by activating this operation the whole field will be saved in file ELECTORS on the user's data disk. The SAVE operation does not affect the contents of the field and we can continue the job from the present situation, but if we make a serious error like an unfortunate SCRATCH or CLEAR operation which partially or completely destroys our text, the situation before saving will be restored by typing and activating a LOAD ELECTORS operation.

The same procedure applies when we like to break the job for a while; we save the field and recall it later by a LOAD operation.

In order to print the lines 5-10 on paper we type the operation PRINT 5,10 on some empty line:

Disp.2.14

```
       1  SURVO 76 EDITOR   (C)1979 S.Mustonen    (100x100)
  1    *SAVE ELECTORS
  2    *
  3    *PRINT 5,10_
  4    *
  5    *   Example 1: Log-Linear models for Contingency tables
  6    *
  7    *   The following data are  taken from a  survey of the  political
  8    *attitudes of a sample  of British electors  which is reported  in
  9    *Butler and Stokes (1974). We examine the relationships among four
 10    *variables: vote, sex, class and age.
 11    *
 12    *TRIM2 7,11,65
 13    *
```

and after activation obtain the following output on printer:

```
           Example 1: Log-Linear models for Contingency tables

              The following data are  taken from a  survey of the  political
           attitudes of a sample  of British electors  which is reported  in
           Butler and Stokes (1974). We examine the relationships among four
           variables: vote, sex, class and age.
```

In that output the line numbers and the control column filled with asterisks do not appear.

So far this has only been an introduction to the normal text editing facilities of SURVO 76 EDITOR. Now we try to illustrate the possibilities for handling data sets and statistical operations.

We continue our example on contingency tables by entering the four-way table to be analyzed:

Disp.2.15

```
       1  SURVO 76 EDITOR   (C)1979 S.Mustonen    (100x100)
  1    *SAVE ELECTORS
  2    *
  3    *PRINT 5,10
  4    *
  5    *   Example 1: Log-Linear models for Contingency tables
  6    *
  7    *   The following data are  taken from a  survey of the  political
  8    *attitudes of a sample  of British electors  which is reported  in
  9    *Butler and Stokes (1974). We examine the relationships among four
 10    *variables: vote, sex, class and age.
 11    *   The variables are defined and the observed frequencies are
 12    *given in the four-way table below.
 13    *
 14    *   OBSERVED FREQUENCIES FOR VOTE BY SEX BY CLASS BY AGE (N=1257)
 15    *
 16    *
 17    *               Sex    Male         Female
 18    *               Vote   Cons  Labour Cons Labour
 19    *Class   Age    ******
 20    *
 21    *upper   >73             4     0     10   0
 22    *        51-73          27     8     26   9
 23    *        41-50          27     4     25   9_
```

When entering the elements of the table we reach for the first time the last line of the screen, but we can continue typing normally, since the field is scrolling automatically according to our needs and after typing the last lines of the table we have the display:

Disp.2.16

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 16   *
 17   *                Sex   Male          Female
 18   *                Vote  Cons  Labour Cons Labour
 19   *Class   Age     ******
 20   *
 21   *upper   >73             4     0     10    0
 22   *        51-73          27     8     26    9
 23   *        41-50          27     4     25    9
 24   *        26-40          17    12     28    9
 25   *        <26             7     6      7    3
 26   *
 27   *lower   >73             8     4      9    2
 28   *        51-73          21    13     33    8
 29   *        41-50          27    12     29    4
 30   *        26-40          14    15     17   13
 31   *        <26             9     9     13    7
 32   *
 33   *work    >73             8    15     17    4
 34   *        51-73          35    62     52   53
 35   *        41-50          29    75     32   70
 36   *        26-40          32    66     36   67
 37   *        <26            14    34     18   33_
```

Observe that we can represent a multiway table in a normal fashion. The structure of the table is defined by the <u>classifiers</u> Age, Class, Vote and Sex which can be traced by the aid of the string ****** on line 19. The number of asterisks points out the number of the longest class name in use.

In order to manipulate this table using various editing and statistical operations we have to insert a TABLE specification on any empty line in the field (a natural place is in front of the table; here on line 16):

Disp.2.17

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 16   *TABLE PAYNE,17,37,F_
 17   *                Sex   Male          Female
 18   *                Vote  Cons  Labour Cons Labour
 19   *Class   Age     ******
 20   *
 21   *upper   >73             4     0     10    0
 22   *        51-73          27     8     26    9
 23   *        41-50          27     4     25    9
 24   *        26-40          17    12     28    9
 25   *        <26             7     6      7    3
 26   *
 27   *lower   >73             8     4      9    2
 28   *        51-73          21    13     33    8
 29   *        41-50          27    12     29    4
 30   *        26-40          14    15     17   13
 31   *        <26             9     9     13    7
 32   *
 33   *work    >73             8    15     17    4
 34   *        51-73          35    62     52   53
 35   *        41-50          29    75     32   70
 36   *        26-40          32    66     36   67
 37   *        <26            14    34     18   33
```

TABLE PAYNE,17,37,F on line 16 now defines a multiway table called PAYNE located on lines 17-37. The last parameter F in the TABLE definition declares that the elements of this table should be interpreted as frequencies. (Another common alternative is a data table containing values of a certain variate according to a multiway classification; then we have X instead of F.)

EDITOR provides various means for modification of multiway tables. We can combine classes of any classifier, change places of classifiers in the table, form marginal tables by collapsing over some classifiers and compute with these modified tables as well as with the original one. A set of TAB-operations enables a great variety of table representations and reductions.

Here we can, for instance, collapse over the Age variable by entering a TABD operation:

Disp.2.18

```
         1    SURVO 76 EDITOR     (C)1979 S.Mustonen      (100x100)
 31     *          <26              9     9      13    7
 32     *
 33     *work     >73               8    15      17    4
 34     *          51-73           35    62      52   53
 35     *          41-50           29    75      32   70
 36     *          26-40           32    66      36   67
 37     *          <26             14    34      18   33
 38     *
 39     *TABD PAYNE,Age,40_
 40     *
 41     *
 42     *
 43     *
 44     *
 45     *
```

This operation has been typed on line 39 and the last parameter 40 indicates the first line for the result. This operation leads to the display:

Disp.2.19

```
         1    SURVO 76 EDITOR     (C)1979 S.Mustonen      (100x100)
 31     *          <26              9     9      13    7
 32     *
 33     *work     >73               8    15      17    4
 34     *          51-73           35    62      52   53
 35     *          41-50           29    75      32   70
 36     *          26-40           32    66      36   67
 37     *          <26             14    34      18   33
 38     *
 39     *TABD PAYNE,Age,40_
 40     *TABLE PAYNED,41,46,F
 41     *          Sex      Male          Female
 42     *          Vote     Cons   Labour Cons   Labour
 43     *Class     ******
 44     *upper                  82     30     96     30
 45     *lower                  79     53    101     34
 46     *work                  118    252    155    227
 47     *
```

where we have the required three-way marginal table automatically labelled with an appropriate TABLE definition on line 40. The name of this new table is generated by adding the fourth letter from the TAB-operation used (here D) at the end of the original name PAYNE. Thus subsequent operations may refer to this new table by name PAYNED.

For the analysis of contingency tables SURVO 76 EDITOR includes the TABFIT operation, which can be used for estimation of log-linear models. (For these models see e.g. Payne 1977, Bishop,Fienberg,Holland 1975.)

If we like to estimate a log-linear model for the table PAYNED assuming that Sex, Class and Vote are independent we specify this model in the following way:

Disp.2.20

```
40   *TABLE PAYNED,41,46,F
41   *         Sex     Male          Female
42   *         Vote    Cons   Labour Cons   Labour
43   *Class    ******
44   *upper               82     30     96     30
45   *lower               79     53    101     34
46   *work               118    252    155    227
47   *
48   *TABFIT PAYNED,49,50
49   *LOGLIN C,V,S
50   *
```

Here the TABFIT operation specifies the table to be analyzed (PAYNED), the line defining the model (49), and the line for the results (50). The full independence model here implies that the one-dimensional marginals for Class (C), Vote (V) and Sex (S) must be fixed in the estimation of the expected frequencies. This specification corresponds to the common X↑2-test for independence and it is typed on the line 49 in the form LOGLIN C,V,S. Observe that the initials of each classifier should be used in this specification.

Activation of line 48 leads to the results:

Disp.2.21

```
40   *TABLE PAYNED,41,46,F
41   *         Sex     Male          Female
42   *         Vote    Cons   Labour Cons   Labour
43   *Class    ******
44   *upper               82     30     96     30
45   *lower               79     53    101     34
46   *work               118    252    155    227
47   *
48   *TABFIT PAYNED,49,50
49   *LOGLIN C,V,S
50   *G↑2= 165.04 DF=   7 P=0.000 FITTED: C,V,S   (PAYNED)
51   *TABLE PAYNEDF,52,57,F
52   *         Sex     Male          Female
53   *         Vote    Cons   Labour Cons   Labour
54   *Class    ******
55   *upper               58     58     61     61
56   *lower               65     65     69     68
57   *work               184    183    193    192
58   *
```

On line 50 we have the goodness-of-fit statistics for the model and on lines 51-57 the table of fitted frequencies. As expected this model is very poor and we have to continue by trying more complicated ones.

Here we make just one more effort and estimate a model where it is assumed that Class is independent of Sex given Vote. This implies fixing the two-dimensional marginals VS,VC.

To estimate this model we edit the lines 48 and 49 to the form

Disp.2.22

```
48   *TABFIT PAYNED,49,51
49   *LOGLIN VS,VC
```

indicating that the results are to appear from line 51 onwards.
Activation of line 48 once again produces:

Disp.2.23
```
 40    *TABLE PAYNED,41,46,F
 41    *         Sex      Male          Female
 42    *         Vote     Cons   Labour Cons   Labour
 43    *Class    ******
 44    *upper                82     30     96     30
 45    *lower                79     53    101     34
 46    *work                118    252    155    227
 47    *
 48    *TABFIT PAYNED,49,51
 49    *LOGLIN VS,VC
 50    *G↑2= 165.04 DF=   7 P=0.000 FITTED: C,V,S    (PAYNED)
 51    *G↑2=   2.76 DF=   4 P=0.602 FITTED: VS,VC    (PAYNED)
 52    *TABLE PAYNEDF,53,58,F
 53    *         Sex      Male          Female
 54    *         Vote     Cons   Labour Cons   Labour
 55    *Class    ******
 56    *upper                79     32     99     28
 57    *lower                80     47    100     40
 58    *work                121    256    152    223
 59    *
```

The previous results on lines 51-57 are overwritten by the new ones. The fit described on line 51 is now good.

If we want the fitted frequencies more accurate we simply edit at least one element in the table PAYNED to a form corresponding to the accuracy desired and activate TABFIT again.

Thus by editing line 58 temporarily into the form
```
 46    *work                118    252    155    227.00
```
and by repeating TABFIT we get the results in the form:

Disp.2.24
```
 40    *TABLE PAYNED,41,46,F
 41    *         Sex      Male          Female
 42    *         Vote     Cons   Labour Cons   Labour
 43    *Class    ******
 44    *upper                82     30     96     30
 45    *lower                79     53    101     34
 46    *work                118    252    155    227.00
 47    *
 48    *TABFIT PAYNED,49,51_
 49    *LOGLIN VS,VC
 50    *G↑2= 165.04 DF=   7 P=0.000 FITTED: C,V,S    (PAYNED)
 51    *G↑2=   2.76 DF=   4 P=0.602 FITTED: VS,VC    (PAYNED)
 52    *TABLE PAYNEDF,53,58,F
 53    *         Sex      Male          Female
 54    *         Vote     Cons   Labour Cons   Labour
 55    *Class    ******
 56    *upper            78.70  32.11  99.30  27.89
 57    *lower            79.59  46.56 100.41  40.44
 58    *work            120.71 256.33 152.29 222.67
 59    *
```

Thereafter the residuals can be formed simply by a TAB- operation:
```
 60    *TAB- PAYNED,PAYNEDF,61_
```
and the result is:

Disp.2.25

```
 59    *
 60    *TAB- PAYNED,PAYNEDF,61_
 61    *TABLE PAYNEDF-,62,67,F
 62    *          Sex      Male          Female
 63    *          Vote     Cons   Labour Cons   Labour
 64    *Class     ******
 65    *upper                3.30  -2.11  -3.30   2.11
 66    *lower               -0.59   6.44   0.59  -6.44
 67    *work                -2.71  -4.33   2.71   4.33
 68    *
```

These 'raw' residuals could be normalized (for instance by dividing them by the square roots of the expected frequencies) in a few editing steps, but we stop our analysis here.

At this stage the lines 1-67 of the edit field are filled by a mixture of text, data, results and editing operations. We may save the present situation on disk by returning to the first line of the field. This is achieved by pressing key F22:(previous page) a couple of times and by activating the old SAVE ELECTORS operation on line 1:

Disp.2.26

```
       1  SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
  1    *SAVE ELECTORS          _
  2    *
  3    *PRINT 5,10
  4    *
  5    *  Example 1: Log-Linear models for Contingency tables
  6    *
  7    *  The following data are  taken from a  survey of the  political
  8    *attitudes of a sample  of British electors  which is reported  in
  9    *Butler and Stokes (1974). We examine the relationships among four
 10    *variables: vote, sex, class and age.
 11    *  The variables are defined and the observed frequencies are
 12    *given in the four-way table below.
 13    *
 14    *  OBSERVED FREQUENCIES FOR VOTE BY SEX BY CLASS BY AGE (N=1257)
 15    *
 16    *TABLE PAYNE,17,37,F
 17    *                  Sex    Male          Female
 18    *                  Vote   Cons  Labour Cons Labour
 19    *Class   Age       ******
 20    *
 21    *upper   >73               4      0     10    0
 22    *        51-73            27      8     26    9
 23    *        41-50            27      4     25    9
```

To take a new printout of the essential lines, the PRINT operation on line 3 should be edited to form PRINT 5,67 and activated.

In order to make this output clean it can be purged from editing operations and technical lines with F25:(delete line) and F8:(erase line) and all the relevant comments may be inserted among the text. When more lines are needed for such insertions key F25:(insert line) is to be employed.

In many cases it is worth while to preserve the editing operations during the analysis stage. If the editing operations are left in the edit field, it is usually easy to work out the same problem with another data set just by changing the data lines and by activating the old operation lines.

In teaching situations it may be valuable to demonstrate the effects of various changes in the data set to the results by repeating the steps of analysis with data edited in various ways.

3. The use of edit keys

When EDITOR has been called from SURVO 76 and the basic start F1 is taken the F-keys operate as follows:

F0: List of F-starts (as in other SURVO 76 modules)
F1: extra character 1, default is 'ä'
F2: extra character 2, default is 'ö'
F3: UNDERLINING ON/OFF: The next characters to be typed will be printed (printer 2231-W3) underlined. After pressing F3 again the next characters will be printed without underlining.
F4: (END) moves the cursor to the right end of the CRT.
F5: (arrow down) moves the cursor one line downwards.
F6: (arrow up) moves the cursor one line upwards.
F7: (BEGIN) moves the cursor to the left of end of the CRT.
F8: (ERASE) erases the line to the right from the cursor.
F9: (DELETE) deletes one character.
F10: (INSERT) inserts space for one character.
F11: (--->) moves the cursor 5 steps to the right
F12: (->) moves the cursor one step to the right.
F13: (<-) moves the cursor one step to the left.
F14: (<---) moves the cursor 5 steps to the left.
If the edge of the CRT is surpassed in these moves the visible part of the edit field will be moved correspondingly.
F15: copies one line to the current line starting from the current position of the cursor. After pressing F15 the question LINE (NO.) TO BE COPIED? will first be displayed.
F16: inserts missing words typed on the next line to a place pointed out by the cursor. Observe that a normal procedure for minor insertions is to use F26 (INSERT) repeatedly.
F17: extra character 3, default is 'Ä'
F18: extra character 4, default is 'ü'
F19: types characters which are missing on the keyboard. After F19 the question <---HEXCODE? will be displayded and the user has to enter the hexcode of the character. The hexcodes are listed in appendix F of the '2200 VP Basic-2 Language Reference Manual'.
F20: (END) moves the cursor to the bottom line of the CRT
F21: (arrow down) displays the next page on the CRT
F22: (arrow up) displays the previous page on the CRT
F23: (BEGIN) moves the cursor to the first line of the CRT
F24: (extra character) operates as F19 when used for the first time. Thereafter F24 types directly the selected character. To alter the extra character exit and select EDITOR again.
F25: (DELETE) deletes the current line,
F26: (INSERT) inserts a new empty line after the current line,
F27: (--->) displays the right side of the edit field,
F28: (->) displays the right side of the edit field,
F29: (<-) moves the cursor to the start of the current line,
F30: (<---) moves the cursor to the start of the current line,

F31: moves the cursor to the control column 0 indicated normally by an asterisk *. Exit from the control column takes place by using the arrow keys F11,F12. The characters in the control column have the following tasks:

char.	task
-	In the PRINT operation the line will be printed with elongated characters.
s	The next line feed (in printer 2231W-3) is 1/3 of a normal line feed. This feature can be used for typing exponents, indices etc.

76 * For example
77 s
78 s Variables $X_1^{(i)}, X_2^{(i)}, \dots, X_n^{(i)}$
79 *
80 * (next line)

char.	task
q	as s, but only a 1/12 of a normal line feed.
r	as s, but only a 1/6 of a normal line feed.
/	starts a new page when the extended form of PRINT operation is used.
?	The line will not be printed.
<	Extra empty lines. For example, <10 means that 10 empty lines is to be printed in PRINT operations.

'blank' splits the display on the CRT into two parts so that the upper part is from the current first line to the line with a blank in the control column and the lower part is a copy of the first part at first.
From now on the cursor can be moved in the lower part according to the common rules and edit field can be seen through this window, but the upper window displays the contents of the original lines constantly.
This procedure is useful when long lists and tables are to be manipulated in the edit field. Then it is profitable to keep the labels and eventually some of the first lines in the upper window and scroll the list in the lower one.
To restore the normal display, remove 'blank' from the control column by inserting a '*', for instance.

4. Text editing

In this chapter we shall consider basic operations needed in text and data management.

4.1 Clearing the edit field

Clearing of single characters and lines takes place by the keys:
F9: delete one character,
F25:delete one line,
F8: erase line to the right of the cursor

If the user likes to clear the field partially or completely during the work he can employ SCRATCH, CLEAR, ERASE and DELETE operations.

SCRATCH
: (without any parameters) clears the edit field from the SCRATCH line onwards.

CLEAR L1,L2
: where L1 and L2 are line numbers and L1<=L2, clears the lines L1-L2.

CLEAR xy
: clears a rectangular subfield indicated by the character x in the left upper corner and y in the right lower corner. The characters used for field indication (xy) must be uniquely determined in the edit field.

For instance, if we activate CLEAR *+ in the following situation:

Disp.4.1

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 38   *
 39   *
 40   *TABLE PAYNED,41,46,F
 41   *        Sex     Male          Female
 42   *        Vote    Cons   Labour Cons   Labour
 43   *Class   ******
 44   *upper             82     30     96     30
 45   *lower             79     53    101     34
 46   *work             118    252    155    227 +
 47   *
 48   *CLEAR *+_
 49   *
```

nothing happens, since * is ambiguous, but insertion of a $ after the asterisks on line 43

Disp.4.2

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 38   *
 39   *
 40   *TABLE PAYNED,41,46,F
 41   *        Sex     Male          Female
 42   *        Vote    Cons   Labour Cons   Labour
 43   *Class   ****** $
 44   *upper             82     30     96     30
 45   *lower             79     53    101     34
 46   *work             118    252    155    227 +
 47   *
 48   *CLEAR $+_
 49   *
```

and activation of CLEAR $+ on line 48 leads to

Disp.4.3

```
         1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 38     *
 39     *
 40     *TABLE PAYNED,41,46,F
 41     *         Sex      Male          Female
 42     *         Vote     Cons   Labour Cons   Labour
 43     *Class    ******
 44     *upper
 45     *lower
 46     *work
 47     *
 48     *CLEAR $+_
 49     *
```

A third form of the CLEAR operation is CLEAR L1,L2,K which clears the lines L1-L2 for those columns which are indicated by non-blank characters on an image line K. Consider the situation:

Disp.4.4

```
         1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 40     *TABLE PAYNED,41,46,F
 41     *         Sex      Male          Female
 42     *         Vote     Cons   Labour Cons   Labour
 43     *Class    ******
 44     *upper                 82     30     96     30
 45     *lower                 79     53    101     34
 46     *work                 118    252    155    227
 47     *                         XXXXXX        XXXXX
 48     *CLEAR 42,46,47_
 49     *
 50     *
```

Execution of CLEAR 42,46,47 gives:

Disp.4.5

```
         1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 40     *TABLE PAYNED,41,46,F
 41     *         Sex      Male          Female
 42     *         Vote     Cons          Cons
 43     *Class    ******
 44     *upper                 82            96
 45     *lower                 79           101
 46     *work                 118           155
 47     *                         XXXXXX        XXXXX
 48     *CLEAR 42,46,47_
 49     *
 50     *
```

In EDITOR the columns to be processed are never referred to by column indices, but image lines are used to point out the columns involved.

The ERASE operation is to be used for erasing selected characters in the field.

ERASE <string>

erases all characters occuring in <string> from the edit field.

For instance, to erase all numbers execute ERASE 1234567890.

The DELETE operation is a generalization of key F9:(delete).

DELETE L1,L2

deletes the column indicated by the cursor from the lines L1-L2 simultaneously. Thus the current column will be lost and the columns to the right from the cursor are moved one step to the left.

4.2. Moving parts of the edit field

For various insertions in the edit field we can use following operations.

INSERT L1,L2

inserts one empty column to the current position of the cursor for the lines L1-L2 simultaneously. Thus the columns to the right of the cursor are moved one step to the right.

Thus INSERT is a generalization of key F10:(insert). For example, by activating INSERT 41,46 on line 48 five times

Disp.4.6

```
 40    *TABLE PAYNED,41,46,F
 41    *        Sex      Male         Female
 42    *        Vote     Cons   Labour Cons   Labour
 43    *Class   ******
 44    *upper              82     30     96     30
 45    *lower              79     53    101     34
 46    *work              118    252    155    227
 47    *
 48    *INSERT 41,46                _
 49    *
```

we shall have:

Disp.4.7

```
 40    *TABLE PAYNED,41,46,F
 41    *        Sex      Male              Female
 42    *        Vote     Cons   Labour      Cons   Labour
 43    *Class   ******
 44    *upper              82     30          96     30
 45    *lower              79     53         101     34
 46    *work              118    252         155    227
 47    *
 48    *INSERT 41,46                _
 49    *
```

COPY L1,L2,L

moves lines L1-L2 to a new place starting from line L. The previous contents of lines L,L+1,...,L+L2-L1+1 will be destroyed.

Observe that for moving single lines it is simplest to use key F15.

The MOVE operation is a generalization of COPY permitting various rectangular subfields to be copied in the edit field. MOVE has two forms.

MOVE xyz

moves a subfield indicated by character x in the left upper corner and character y in the right lower corner to a place indicated by character z so that x will be in the position of z. The indicators x,y,z must be unambiguous; otherwise the the MOVE xyz operation is not carried out.

MOVE L1,L2,K,L

moves a subfield from the lines L1-L2 to lines starting from L. The columns which are to moved and to which columns are specified by an image line K consisting of at least two X's and one Y. The columns to be moved are determined by the first and last X and the first column of the subfield in the new place by the first Y.

In the next example both MOVE operations on lines 48,49 will do the same task:

Disp.4.8

```
40   *TABLE PAYNED,41,46,F
41   *          Sex     <Male             Female            ↑
42   *          Vote    Cons   Labour Cons    Labour
43   *Class     ******
44   *upper                 82      30     96      30
45   *lower                 79      53    101      34
46   *work                 118     252)   155     227
47   *                  X              X                   Y
48   *MOVE <>↑
49   *MOVE 41,46,47,41
```

and the result is

Disp.4.9

```
41   *          Sex     <Male             Female           <Male
42   *          Vote    Cons   Labour Cons    Labour     Cons    Labour
43   *Class     ******
44   *upper                 82      30     96      30         82      30
45   *lower                 79      53    101      34         79      53
46   *work                 118     252)   155     227        118     252)
47   *                  X              X                   Y
48   *MOVE <>↑
49   *MOVE 41,46,47,41
```

Note that after using MOVE <>↑ it is easy to cancel the auxiliary characters <>↑ from the edit field by an ERASE <>↑ operation.

4.3. Saving, loading and combining edit fields

The user can save edit fields on data disks and load them back for subsequent operations. Also parts of various edit files can be joined together in the present edit field. Observe, however, that in order to build up more comprehensive documents it is not necessary to reorganize the material in a right order before printout. The extended form of the PRINT operation enables the output to consist of any number of chapters located in various files in any order (see 4.4).

The present contents of the edit field is saved temporarily on the program disk by pressing F0 and thereafter F5. This temporary file is loaded into the edit field pressing F0 and thereafter F6.

During a long job it is recommended to employ this temporary saving procedure especially before activating operations which may harmfully spoil the results achieved.

Saving on the user's data disk takes place by a SAVE operation.

SAVE XYZ

saves the the field in file XYZ. If the file is not in the catalog a new file XYZ will automatically be created and space will be reserved according to the dimensions of the edit field. If the file XYZ is already in the catalog this file is used for saving.

LOAD XYZ

loads the edit file XYZ in the edit field and the previous contents of the field will be destroyed. The field will also be automatically redimensioned according to the file to be loaded.

During the job the user can alter the dimensions of the edit field with a REDIM operation. For instance, after execution of REDIM 124,80 the number of lines is 124 and the number of columns 80. The REDIM operation does not alter the contents of the edit field, unless the new dimensions are too small. Then the excessive lines and columns will disappear.

The JOIN operation can be used for connecting chapters and subfields from edit files to the present field. JOIN has two forms.

JOIN L1,L2,N,L

moves the lines L1-L2 from edit file N to the present edit field so that the the first line will be on line L in the present edit field. The previous contents of lines L,L+1,...,L+L2-L1+1 will be destroyed.

JOIN N,C1,C2,...,L

moves the subfields (chapters) C1,C2,... from edit file N to the present edit field so that the first line of the first subfield will be L. The chapters C1,C2,... have to be defined in file N by a DEF definition.

DEF <name of subfield>,L1,L2

defines a subfield consisting of lines L1-L2 in the present edit field. After saving the edit field in an edit file this subfield can be referred to in JOIN and PRINT operations.

To keep record of various edit files on disk the user can create an INDEX file which is like any other edit file, but INDEX can also be loaded simply by F8-start after EDITOR has been called or F0 has been pressed. In the INDEX file the user may have a list of various files on disk e.g. in the following form:

Disp.4.10

```
        1   SURVO 76 EDITOR     (C)1979 S.Mustonen     (40x60)
  1     *SAVE INDEX
  2     *
  3     *Index for disk 189/SM
  4     *
  5     *LOAD DATA1
  6     *LOAD DATA2
  7     *LOAD DATA3
  8     *           data sets...
  9     *LOAD LETTER1
 10     *LOAD LETTER2
 11     *           letters...
 12     *LOAD EX1
 13     *LOAD EX2
 14     *           examples...
 15     *
```

Thus any edit file may be loaded by moving the cursor to the corresponding LOAD line and by keying CONTINUE.

4.4. Printout of the edit field and edit files

The PRINT operation has three alternative forms.

PRINT L1,L2

prints the lines L1-L2 on paper without line numbers and the control column as seen in Disp.1.14. The control column can be used for elongating the text on certain lines (- in control column) or for creating non-standard line feeds for indices etc. (see F31 in chapter 3).

PRINT L1,L2,<separator character>

works as PRINT above, but starts a new line whenever the selected separator character is encountered.

```
PRINT R1,R2,<1st page>,<left edge>,<page length>,<extra lines>
        <file 1>,<chapter 1>
        <file 2>,<chapter 2>
        ------ -----
        END
```

prints a document consisting of several chapters in several edit files so that the text will automatically be divided into pages and the pages will be started by one or several header lines with appropriate page numbers.

When the general PRINT operation is activated, the chapters to be printed need not to be in the current edit field.

The header lines for the pages are to be described on edit lines R1-R2 and the line R1 must contain the image of the page number in the form ###.

The pages will be numbered starting from the value <1st page>. In front of each line, the number of columns pointed out by the parameter <left edge> will be left empty. The parameter <page length> specifies the maximum number of lines in one page (including the header lines). The parameter <extra lines> specifies the number of empty lines to be left between the pages.

The chapters to be printed must be defined by DEF's in their own edit files.

The page division can also be controlled by inserting / to the control column of those lines which are starting a new page. Each page will be filled up to the last line, unless otherwise stated by a / in the control column.

The printout can be checked in advance by using this PRINT operation with a - sign in the place of <left edge>. The the text will be displayed on the CRT and the output will be stopped when the page should be changed. After a stop of this kind the user can move the cursor in the control column by the keys F6 (upwards) and F5 (downwards) and set a new page change by pressing / or remove such a change by pressing *. If the user is satisfied with the present suggestion, RETURN(EXEC) is pressed and the next possible point of page change will be displayed.

This checking can be repeated indefinitely, and the true output will take place by changing <left edge> to its final value and by activating PRINT once again.

For example the original copy of this paper has been printed with a single PRINT operation of the form:

Disp.4.11

```
 14   *S.Mustonen: SURVO 76 EDITOR       10.8.1980                                   ##
 15   *--------------------------------------------------------------------------------
 16   *
 17   *PRINT 14,16,1,10,66,6
 18   *EDI0,A
 19   *EDI1,A
 20   *EDI2,A
 21   *EDI3,A
 22   *EDI4,A
 23   *EDI5,A
 24   *EDI6,A
 25   *END
```

4.5. Making up and adjusting the text lines

Three TRIM operations are available for changing and adjusting the line length of the text in the edit field.

```
TRIM L1,L2,C
```

makes up the lines L1-L2 using the line width C without splitting words, numbers and other contiguous character strings. If new lines are needed they will automatically be inserted and,

unnecessary lines will be deleted.
After TRIM only one space will be left between the 'words' and the right edge will not necessarily be even. To make it straight use TRIM2 immediately after TRIM.
The lines after an empty line and the lines starting with at least one space will stay as new lines.

TRIM2 L1,L2,C
makes the right edge of the lines L1-L2 straight in the column position C by inserting spaces evenly between the 'words' without splitting or connecting the lines. If the line is short for more than 8 characters nothing will be changed on that line.

TRIM3 L1,L2,C
operates as TRIM, but splits the words when needed according to the rules of Finnish. TRIM3 operates rather satisfactorily also in other languages. Incorrect splittings are easy to modify afterwards by using standard editing keys.

An example of TRIM operations is already presented in chapter 2 (Displays 2.9-12).

4.6. Searching for strings and replacing them

FIND <string>
finds the first occurence of <string> in the edit field. The cursor will then point at the first character of <string>. By pressing CONTINUE immediately again the next occurence will be found etc.

FIND L1,L2,K
works as FIND above but scans only lines L1-L2. The string to be found is the contents of image line K.

REPLACE L1,L2,K1,K2
searches for the string described on line K1 from lines L1-L2. If CONTINUE is pressed thereafter the string K1 will be replaced by the string on line K2 and the next occurence of K1 will be searched for. If CLEAR is pressed K1 is not replaced, but the next K1 will be searched for.

5. Editing operations for data matrices

The SURVO 76 EDITOR can be used in many ways for data input and management. Edited data sets may be saved in SURVO 76 data files. On the other side, SURVO 76 data files and various special files, like correlation files and MATRI files can be loaded in the edit field and modified there for subsequent operations in SURVO 76 or for final printout with text and comments.

For illustration, we consider the following small data set of 12 observations:

Yearly consumption of certain beverages per inhabitant
in some European countries in 1972-1976
(Source: Statistical Abstract of Sweden 1979)

	Coffee (kg)	Tea (kg)	Beer (l)	Wine (l)	Spirits (l)
Sweden	12.9	0.30	58.3	7.9	2.9
Denmark	11.8	0.41	113.9	10.4	1.7
Finland	12.5	0.15	54.7	7.6	2.7
Norway	9.4	0.19	43.5	3.1	1.8
France	5.2	0.10	44.5	104.3	2.5
Ireland	0.2	3.73	124.5	3.8	1.9
Italy	3.6	0.06	13.6	106.6	2.0
Holland	9.2	0.58	75.5	9.7	2.7
Portugal	2.2	0.03	27.5	89.3	0.9
Switzerland	9.1	0.25	73.5	44.9	2.1
Spain	2.5	0.03	43.6	73.2	2.7
England	1.8	3.49	113.7	5.1	1.4

This data set can be typed in the edit field as it is, but if the user likes to save time and effort it is not necessary to do it exactly in straight columns, since this can easily be achieved afterwards.

Thus after the input of data we may have the following display:

Disp.5.1

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
  8   *           Coffee Tea  Beer  Wine Spirits
  9   *            (kg)  (kg)  (l)   (l)  (l)
 10   *
 11   *Sweden 12.9 0.3 58.3 7.9 2.9
 12   *Denmark 11.8 0.41 113.9 10.4 1.7
 13   *Finland 12.5 0.15 54.7 7.6 2.7
 14   *Norway 9.4 0.19 43.5 3.1 1.8
 15   *France  5.2 0.10 44.5 104.3 2.5
 16   *Ireland  0.2 3.73 124.5 3.8 1.9
 17   *Italy 3.6 0.06 13.6 106.6 2
 18   *Holland 9.2 0.58 75.5 9.7 2.7
 19   *Portugal 2.2 0.03 27.5 89.3 0.9
 20   *Switzerland  9.1 0.25 73.5 44.9 2.1
 21   *Spain 2.5 0.03 43.6 73.2 2.7
 22   *England 1.8 3.49 113.7 5.1 1.4_
 23   *
 24   *
 25   *
```

5.1. Formatting

In order to obtain a neat tabular form for this data matrix we use a FORM operation:

Disp.5.2

```
20    *Switzerland  9.1 0.25 73.5 44.9 2.1
21    *Spain 2.5 0.03 43.6 73.2 2.7
22    *England 1.8 3.49 113.7 5.1 1.4
23    *
24    * XXXXXXXXXXX 12.1 1.12 123.1 123.1 1.1
25    *FORM 11,22,24_
```

FORM operates on lines 11-22 and the new format of these lines is specified on image line 24. The numeric fields are defined by numbers in the desired format and the alpha fields by letters. In many cases one of the operand rows edited to the final form can be used as an image line. Now activation of line 25 leads to the result:

Disp.5.3

```
       1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 8    *            Coffee Tea  Beer  Wine Spirits
 9    *             (kg)  (kg)  (l)   (l)  (l)
10    *
11    * Sweden       12.9 0.30  58.3   7.9 2.9
12    * Denmark      11.8 0.41 113.9  10.4 1.7
13    * Finland      12.5 0.15  54.7   7.6 2.7
14    * Norway        9.4 0.19  43.5   3.1 1.8
15    * France        5.2 0.10  44.5 104.3 2.5
16    * Ireland       0.2 3.73 124.5   3.8 1.9
17    * Italy         3.6 0.06  13.6 106.6 2.0
18    * Holland       9.2 0.58  75.5   9.7 2.7
19    * Portugal      2.2 0.03  27.5  89.3 0.9
20    * Switzerland   9.1 0.25  73.5  44.9 2.1
21    * Spain         2.5 0.03  43.6  73.2 2.7
22    * England       1.8 3.49 113.7   5.1 1.4
23    *
24    * XXXXXXXXXXX 12.1 1.12 123.1 123.1 1.1
25    *FORM 11,22,24_
```

5.2. Sorting

Formatted data matrices can be sorted in numeric or alphanumeric order using any column as a sort key.

SORT L1,L2,K
: sorts the lines L1-L2 in the order determined by image line K. If K has the form " XXXXXX ", the lines will be sorted in alphanumeric order using the XXXXXX column as a sort key.
If K has the form "   111111   ", the lines will be sorted in ascending numeric order using 111111 column as a sort key.

-SORT L1,L2,K
: sorts the lines L1-L2 in descending order in case of numeric sorting.

For example, to sort the countries in alphanumeric order we type the SORT operation:

Disp.5.4
```
 8    *             Coffee Tea  Beer  Wine Spirits
 9    *              (kg)  (kg)  (l)   (l)  (l)
10    *
11    * Sweden        12.9 0.30  58.3   7.9 2.9
12    * Denmark       11.8 0.41 113.9  10.4 1.7
13    * Finland       12.5 0.15  54.7   7.6 2.7
14    * Norway         9.4 0.19  43.5   3.1 1.8
15    * France         5.2 0.10  44.5 104.3 2.5
16    * Ireland        0.2 3.73 124.5   3.8 1.9
17    * Italy          3.6 0.06  13.6 106.6 2.0
18    * Holland        9.2 0.58  75.5   9.7 2.7
19    * Portugal       2.2 0.03  27.5  89.3 0.9
20    * Switzerland  9.1 0.25  73.5  44.9 2.1
21    * Spain          2.5 0.03  43.6  73.2 2.7
22    * England        1.8 3.49 113.7   5.1 1.4
23    *
24    * XXXXXXXXXXX
25    *SORT 11,22,24_
```

and after the execution the display is:

Disp.5.5
```
 8    *             Coffee Tea  Beer  Wine Spirits
 9    *              (kg)  (kg)  (l)   (l)  (l)
10    *
11    * Denmark       11.8 0.41 113.9  10.4 1.7
12    * England        1.8 3.49 113.7   5.1 1.4
13    * Finland       12.5 0.15  54.7   7.6 2.7
14    * France         5.2 0.10  44.5 104.3 2.5
15    * Holland        9.2 0.58  75.5   9.7 2.7
16    * Ireland        0.2 3.73 124.5   3.8 1.9
17    * Italy          3.6 0.06  13.6 106.6 2.0
18    * Norway         9.4 0.19  43.5   3.1 1.8
19    * Portugal       2.2 0.03  27.5  89.3 0.9
20    * Spain          2.5 0.03  43.6  73.2 2.7
21    * Sweden        12.9 0.30  58.3   7.9 2.9
22    * Switzerland  9.1 0.25  73.5  44.9 2.1
23    *
24    * XXXXXXXXXXX
25    *SORT 11,22,24_
```

Similarly, to obtain the order according to wine consumption we edit the lines 24,25 to form
```
24    *                                 11111
25    *-SORT 11,22,24_
```
and by activating the line 25 again we obtain the result:

Disp.5.6

```
 8   *             Coffee Tea  Beer  Wine Spirits
 9   *              (kg)  (kg)  (l)   (l)  (l)
10   *
11   * Italy         3.6 0.06  13.6 106.6 2.0
12   * France        5.2 0.10  44.5 104.3 2.5
13   * Portugal      2.2 0.03  27.5  89.3 0.9
14   * Spain         2.5 0.03  43.6  73.2 2.7
15   * Switzerland   9.1 0.25  73.5  44.9 2.1
16   * Denmark      11.8 0.41 113.9  10.4 1.7
17   * Holland       9.2 0.58  75.5   9.7 2.7
18   * Sweden       12.9 0.30  58.3   7.9 2.9
19   * Finland      12.5 0.15  54.7   7.6 2.7
20   * England       1.8 3.49 113.7   5.1 1.4
21   * Ireland       0.2 3.73 124.5   3.8 1.9
22   * Norway        9.4 0.19  43.5   3.1 1.8
23   *
24   *                               11111
25   *-SORT 11,22,24_
```

The columns of tables can also be rearranged using the PERM operation.

PERM L1,L2,L

> permutes the fields (columns) on lines L1-L2 according to the fields on the lines L,L+1,L+2,...
> The numbers of fields on line L+i and L1+i must be the same and the fields on line L1+i will be permuted in a new order corresponding to the alphabetic order of the fields on line L+i.
> If some of the lines L,L+1,... is empty the remaining lines will be permuted according to the last non-empty line. For example, when changing the order of columns in a table the line L should point out the new order and the line L+1 should be empty.

An example of PERM will be found in Display 6.18.

5.3. Numerical operations

Data matrices and other numeric tables can be transformed and edited using several numeric operations. At first the auxiliary operations SET and COUNT are introduced.

To create constant columns (numeric or alpha) the SET operation is to be employed.

SET L1,L2,K

> inserts the non-blank characters of image line K to corresponding columns on lines L1-L2.

For creating line numbers or other index columns with a constant increment or cyclic behaviour the COUNT operation is available. It has three forms:

COUNT L1,L2,K,C

> inserts numbers C,C+1,C+2,... on lines L1-L2 in the format indicated by an image line K of the form " -123.1 ".

COUNT L1,L2,K,C,D

> works similarly, but inserts the numbers C,C+D,C+2D,...

COUNT L1,L2,K,C,D,E

> with E>0 inserts numbers
> C,C+D,C+2D,...,C+(E-1)D,C,C+D,C+2D,...,C+(E-1)D,C,C+D,C+2D,...
> on lines L1-L2 in the format indicated by image line K thus generating a cyclical count.
> If E<0, the numbers inserted will be
> C,C,...,C,C+D,C+D,...,C+D,C+2D,C+2D,...,C+2D,....
> where each of values C,C+D,C+2D is repeated -E times.

For instance, annotations for a monthly time series in the form

```
year:month
1980: 1
1980: 2
1980: 3
----
----
1980:11
1980:12
1981: 1
1981: 2
etc.
```

are easily generated by COUNT and SET operations. Assume that the edit field is empty. We start by entering the years with COUNT.

Disp.5.7

```
     1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 1  *
 2  *
 3  *
 4  *COUNT 7,23,5,1980,1,-12_
 5  *   1234
 6  *
 7  *   1980
 8  *   1980
 9  *   1980
10  *   1980
11  *   1980
12  *   1980
13  *   1980
14  *   1980
15  *   1980
16  *   1980
17  *   1980
18  *   1980
19  *   1981
20  *   1981
21  *   1981
22  *   1981
23  *   1981
```

Thereafter

```
4   *COUNT 7,23,5,1,1,12_
5   *      12
```

creates the month indices:

Disp.5.8

```
          1   SURVO 76 EDITOR   (C)1979 S.Mustonen    (100x100)
  1    *
  2    *
  3    *
  4    *COUNT 7,23,5,1,1,12_
  5    *        12
  6    *
  7    *   1980  1
  8    *   1980  2
  9    *   1980  3
 10    *   1980  4
 11    *   1980  5
 12    *   1980  6
 13    *   1980  7
 14    *   1980  8
 15    *   1980  9
 16    *   1980 10
 17    *   1980 11
 18    *   1980 12
 19    *   1981  1
 20    *   1981  2
 21    *   1981  3
 22    *   1981  4
 23    *   1981  5
```

Finally, to place the missing ':' between year and month we perform a SET operation:

```
  4    *SET 7,23,5_
  5    *        :
```

The operations C+,C-,C* and C/ are to be used for making new columns by addition, subtraction, multiplication and division, respectively.

```
C+ L1,L2,K
     where image line K="   XXXX  XXXX   XXXX   -123.12   "
     computes the sum of XXXX-columns to the column indicated
     by -123.12 for each of the lines L1-L2.
C* L1,L2,K
     works as C+, but the product of XXXX-columns is computed.
C- L1,L2,K
     where K="   XXXX  YYYY   -123.12   "
     subtracts the YYYY-column from the XXXX-column.
C/ L1,L2,K
     divides the XXXX-column by the YYYY-column.
```

For example, let us create a column for the total alcohol consumption in the following display:

Disp.5.9

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
   1    *
   2    *
   3    *
   4    * Alcoholic beverages consumed in Finland 1969-1978
   5    * (litres per inhabitant, 100 % alcohol)
   6    *
   7    *         strong  wines   long     beer
   8    *         liquors         drinks
   9    * 1969     1.58    0.48    0.04    2.11
  10    * 1970     1.75    0.53    0.04    1.98
  11    * 1971     2.07    0.56    0.04    2.05
  12    * 1972     2.19    0.57    0.06    2.28
  13    * 1973     2.46    0.69    0.10    2.35
  14    * 1974     2.90    0.89    0.18    2.48
  15    * 1975     2.81    0.77    0.20    2.41
  16    * 1976     2.95    0.72    0.20    2.44
  17    * 1977     2.99    0.68    0.22    2.49
  18    * 1978     2.82    0.72    0.20    2.48
```

By using the C+ operation

```
   2    *C+ 9,18,3_
   3    *          XXXX    XXXX    XXXX    XXXX    1.12
```

we obtain:

Disp.5.10

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
   1    *
   2    *C+ 9,18,3_
   3    *          XXXX    XXXX    XXXX    XXXX    1.12
   4    * Alcoholic beverages consumed in Finland 1969-1978
   5    * (litres per inhabitant, 100 % alcohol)
   6    *
   7    *         strong  wines   long     beer
   8    *         liquors         drinks
   9    * 1969     1.58    0.48    0.04    2.11    4.21
  10    * 1970     1.75    0.53    0.04    1.98    4.30
  11    * 1971     2.07    0.56    0.04    2.05    4.72
  12    * 1972     2.19    0.57    0.06    2.28    5.10
  13    * 1973     2.46    0.69    0.10    2.35    5.60
  14    * 1974     2.90    0.89    0.18    2.48    6.45
  15    * 1975     2.81    0.77    0.20    2.41    6.19
  16    * 1976     2.95    0.72    0.20    2.44    6.31
  17    * 1977     2.99    0.68    0.22    2.49    6.38
  18    * 1978     2.82    0.72    0.20    2.48    6.22
```

The user can also define more complicated column transformations by means of a FUNC operation, which has three forms.

FUNC L1,L2,K

where image line K has a form " XXXX XXXX XXXX -123.12 " computes a numerical function of XXXX-columns as a new column indicated by -123.12 on the image line for the lines L1-L2. The function must be defined on program lines 100-119 as a subroutine in the form

100 Y=function(X(1),X(2),...,X(M)):RETURN

where M is the number of XXXX-columns.

If M=1 (function of single variable), the subroutine may have the simple form 100 Y=function(X):RETURN

If the EDITOR program is protected (↑ERR A06), the general form of FUNC described below must be employed.

FUNC N,L1,L2,K
where K=" XXXX XXXX XXXX -123.12 "
works as FUNC above, but employs a function subroutine N which is saved on the data disk under name N in the form:

```
 99 GOTO 1415
100 Y=function(X(1),X(2),...,X(M)):RETURN
```

FUNC N,L1,L2,K,C(1),C(2),....
works as FUNC N,L1,L2,K above, but uses constants C(1),C(2),... as auxiliary parameters. Now the subroutine must be saved on the data disk under name N and in the form

```
 99 GOTO 1415
100 Y=function(X(1),X(2),...,X(M);C(1),C(2),...):RETURN
```

For example, the following subroutine computes the sum of squares of any columns in a table (e.g. the communalities of a factor matrix):

```
 99 GOTO 1415:REM "COMM"
100 Y=0:FOR I=1 TO C(1):Y=Y+X(I)↑2:NEXT I:RETURN
```

If this subroutine is saved on user's data disk under name "COMM", it can be called in EDITOR by the FUNC operation of the form FUNC COMM,L1,L2,K,M where K is the image line specifying M columns filled with X's.

The following displays show how our FUNC COMM operates for computation of communalities. Activation of FUNC COMM

Disp.5.11

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
 20    *Factor matrix on 48 athletes in Decathlon:
 20    *FUNC COMM,22,34,21,4_
 21    *            XXXXXXX  XXXXXXX  XXXXXXX  XXXXXXX  1.1234
 22    *POINTS       0.3914   0.6913   0.4496   0.1532
 23    *100M        -0.0561   0.7453  -0.3773  -0.2139
 24    *BR.JUMP      0.0797   0.5958   0.4356   0.0017
 25    *SHOTPUT      0.8521  -0.0128  -0.0906  -0.0605
 26    *HEIGHT       0.3345  -0.3018   0.5465   0.0725
 27    *400M        -0.4020   0.6098  -0.1976   0.4181
 28    *HURDLES      0.2286   0.6324  -0.1086   0.1471
 29    *DISCUS       0.8616   0.0124   0.0428  -0.1672
 30    *POLEVLT     -0.3184   0.1773   0.2089  -0.6447
 31    *JAVELIN      0.1203  -0.0381   0.7079   0.2428
 32    *1500M       -0.6032  -0.2191  -0.0245   0.5554
 33    *LENGTH       0.8026  -0.0491  -0.2760   0.3152
 34    *WEIGHT       0.8629  -0.1230  -0.2792   0.0942
```

leads to the result:

Disp.5.12

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
 20    *Factor matrix on 48 athletes in Decathlon:
 20    *FUNC COMM,22,34,21,4_
 21    *            XXXXXXX  XXXXXXX  XXXXXXX  XXXXXXX    1.1234
 22    *POINTS       0.3914   0.6913   0.4496   0.1532    0.8567
 23    *100M        -0.0561   0.7453  -0.3773  -0.2139    0.7467
 24    *BR.JUMP      0.0797   0.5958   0.4356   0.0017    0.5510
 25    *SHOTPUT      0.8521  -0.0128  -0.0906  -0.0605    0.7381
 26    *HEIGHT       0.3345  -0.3018   0.5465   0.0725    0.5068
 27    *400M        -0.4020   0.6098  -0.1976   0.4181    0.7473
 28    *HURDLES      0.2286   0.6324  -0.1086   0.1471    0.4856
 29    *DISCUS       0.8616   0.0124   0.0428  -0.1672    0.7722
 30    *POLEVLT     -0.3184   0.1773   0.2089  -0.6447    0.5920
 31    *JAVELIN      0.1203  -0.0381   0.7079   0.2428    0.5759
 32    *1500M       -0.6032  -0.2191  -0.0245   0.5554    0.7209
 33    *LENGTH       0.8026  -0.0491  -0.2760   0.3152    0.8221
 34    *WEIGHT       0.8629  -0.1230  -0.2792   0.0942    0.8465
```

For moving one or more columns downwards with respect to the other columns a LAG operation is available.

```
LAG L1,L2,K,C
     where image line K=" XXXX XXXX XXX "
     moves the columns indicated by X's on the image line C steps
     downwards. The C first lines on the corresponding columns will
     be cleared. Missing C implies C=1.
```

The DIFF and %DIFF operations can be used for computing differences in time series.

```
DIFF L1,L2,K,C
     where K=" XXXX -123.12 "
     computes on lines L1-L2 the C step differences of X-column ac-
     cording to the image -123.12. Missing C implies C=1.
%DIFF L1,L2,K,C
     works as DIFF, but computes the differences in percentages.
```

For example, to compute the yearly proportional change of total consumption in Display 5.9 we enter the %DIFF operation

```
  2    *%DIFF 9,18,3_
  3    *                                          XXXX  +12.12
```

and after execution obtain the result:

Disp.5.13

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
  1    *
  2    *%DIFF 9,18,3_
  3    *                                          XXXX   +1.12
  4    * Alcoholic beverages consumed in Finland 1969-1978
  5    * (litres per inhabitant, 100 % alcohol)
  6    *
  7    *        strong  wines   long    beer   total    %
  8    *        liquors         drinks
  9    * 1969    1.58   0.48    0.04    2.11    4.21
 10    * 1970    1.75   0.53    0.04    1.98    4.30   +2.13
 11    * 1971    2.07   0.56    0.04    2.05    4.72   +9.76
 12    * 1972    2.19   0.57    0.06    2.28    5.10   +8.05
 13    * 1973    2.46   0.69    0.10    2.35    5.60   +9.80
 14    * 1974    2.90   0.89    0.18    2.48    6.45  +15.17
 15    * 1975    2.81   0.77    0.20    2.41    6.19   -4.03
 16    * 1976    2.95   0.72    0.20    2.44    6.31   +1.93
 17    * 1977    2.99   0.68    0.22    2.49    6.38   +1.10
 18    * 1978    2.82   0.72    0.20    2.48    6.22   -2.50
```

(The labels 'total' and '%' on line 7 are inserted afterwards.)

The operations L+,L-,L* and L/ compute sums, differences, products and ratios of numbers located on consecutive lines in the same columns.

```
L+ L1,L2,K,L
     where image line K="    -123.12   -123.12    -123.12      "
     computes the sum of lines L1-L2 for the columns indicated by
     the image line and inserts the sums to line L according to the
     formats on image line.
L* L1,L2,K,L
     works as L+, but the product is computed.
L- L1,L2,K,L
     computes the difference of lines L1 and L2 to line L.
L/ L1,L2,K,L
     computes the ratio of lines L1 and L2 to line L.
```

For instance, if we like to compute the sums of the frequencies over 'Class' in the following display:

Disp.5.14
```
40    *TABLE PAYNED,41,46,F
41    *         Sex      Male          Female
42    *         Vote     Cons   Labour Cons   Labour
43    *Class    ******
44    *upper               82     30     96     30
45    *lower               79     53    101     34
46    *work               118    252    155    227
47    *
48    *
49    *
```

the operation
```
49    *                   111    111    111    111
50    *L+ 44,46,49,47_
```
will do that for us and the result is:

Disp.5.15
```
40    *TABLE PAYNED,41,46,F
41    *         Sex      Male          Female
42    *         Vote     Cons   Labour Cons   Labour
43    *Class    ******
44    *upper               82     30     96     30
45    *lower               79     53    101     34
46    *work               118    252    155    227
47    *                   279    335    352    291
48    *
49    *                   111    111    111    111
50    *L+ 44,46,49,47_
```

5.4. Loading and saving SURVO 76 files

The data matrices typed in the edit field can be saved in SURVO 76 data files for subsequent analysis with various SURVO 76 modules.

On the other hand, parts of SURVO 76 data files, correlation files, and various matrix files can be loaded in the current edit field for editing and output or for statistical operations which are controlled from EDITOR.

Before saving a new data set we have to create a suitable file on the user's data disk. Thus, if the data is already located in the edit field, save it temporarily, exit from EDITOR (press F0 and F16), and select the DATA module for creating a data file. At this stage it is not necessary to define all variables needed; it is only important that the file created is large enough for the data set in question. Thereafter return to EDITOR and load the edit field to be used.

The data set to be saved in a SURVO 76 file must be in matrix form, the columns corresponding to variables and the rows to observations. Each observation line has to start with a name (a contiguous string; only 8 first characters will be saved) and somewhere in the field (typically just above the data set) a header line declaring the names of the variables (columns) must be typed. Also the names of the variables have to be contiguous strings and the 8 first characters of each are used as names for variables in the SURVO 76 file.

For instance, among the preceding displays the data sets in displays 5.3 and 5.9 have the desired structure. In display 5.3 we have 5 variables (Coffee, Tea, Beer, Wine, Spirits) and 12 observations with names Sweden, Denmark, Finland etc. When saving this data matrix in a SURVO 76 file the line 8 could be used as a header line which defines the names of the variables also in the SURVO 76 data file. In display 5.9 we have 10 observations of 4 variables and line 7 will serve as a header line.

To save the data in display 5.3 in an empty SURVO 76 file DRINKS the following SAVE operation should be activated:

Disp.5.16

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
  8   *             Coffee Tea  Beer  Wine Spirits
  9   *              (kg)  (kg)  (l)   (l)  (l)
 10   *
 11   * Sweden        12.9 0.30  58.3   7.9 2.9
 12   * Denmark       11.8 0.41 113.9  10.4 1.7
 13   * Finland       12.5 0.15  54.7   7.6 2.7
 14   * Norway         9.4 0.19  43.5   3.1 1.8
 15   * France         5.2 0.10  44.5 104.3 2.5
 16   * Ireland        0.2 3.73 124.5   3.8 1.9
 17   * Italy          3.6 0.06  13.6 106.6 2.0
 18   * Holland        9.2 0.58  75.5   9.7 2.7
 19   * Portugal       2.2 0.03  27.5  89.3 0.9
 20   * Switzerland  9.1 0.25  73.5  44.9 2.1
 21   * Spain          2.5 0.03  43.6  73.2 2.7
 22   * England        1.8 3.49 113.7   5.1 1.4
 23   *
 24   *SAVE 11,22,8,DRINKS_
 25   *
```

The general structure of this SAVE operation is:

SAVE L1,L2,K,N

where N is the name of a SURVO 76 data file, L1-L2 are the observations to be saved and K is the line specifying the names of the variables.

The SAVE operation can be used for saving new data matrices as well as for enlarging and editing SURVO 76 data files. Eventual new variables and observations will automatically be defined in the SURVO 76 file.

For instance, to alter some data values in the SURVO 76 file DRINKS after the initial saving in Display 5.16 it is sufficient to make the alterations in the edit field and thereafter reactivate the LOAD operation. Of course in many cases it is more natural to make corrections directly in the data file DRINKS by means of the SURVO 76 module DATA. If the corrected data file is needed later in the edit field it is easy to load it back by using a LOAD operation.

A LOAD operation without any parameters can be used for loading parts of SURVO 76 data files and results of SURVO 76 conversations saved on disk in a matrix form.

When LOAD has been activated the edit field disappears temporarily and a typical SURVO 76 conversation follows. At first the user has to select the type of the file to be loaded into the edit field:

Disp.5.17

```
LOADING VARIOUS SURVO 76 FILES:
    1: DATA FILE
    2: CORRELATION FILE
    3: MATRIX FILE (I.E. CORRELATION FILE ETC.)
    4: 'MATRI' FILE
    5: TABLE FILE (MULTIWAY TABLE)
SELECT 1,2,3,4,5? _
```

If alternative 1 (DATA FILE) is selected, the user has to enter the name of the file and select the variables and observations to be loaded. Also the first line in the edit field for use of the selected data set is to be specified. Default is always the first empty line in the field. Moreover the user can determine the format for each variable separately in the form -###.###

For example, if we load the 5 first observations and variables Wine, Beer from the file DRINKS using format ###.## for data values we shall have a following display:

Disp.5.18
```
 1   *
 2   *LOAD_
 3   *DATA DRINKS,6,10,4
 4   *          Beer    Wine
 5   *
 6   * Sweden     58.30    7.90
 7   * Denmark   113.90   10.40
 8   * Finland    54.70    7.60
 9   * Norway     43.50    3.10
10   * France     44.50  104.30
11   *
```

Observe that EDITOR specifies this data set with a DATA line which is placed automatically above the data matrix. In statistical operations working on data matrices a DATA specification will be used in the same way as TABLE specification was used already in the introductory example (see also chapter 7).

The other types of SURVO 76 files can be loaded in a similar way. During a short SURVO 76 conversation the dimensions and the format of the output will be defined and thereafter the matrix selected will appear with proper labels in the edit field.

6. Multiway tables

The most important statistical operations working directly in the edit field are dealing with multiway tables. EDITOR provides operations for the analysis of variance (ANOVA) and log-linear models for contingency tables (TABFIT). Several technical operations are also available for table editing and modification. By means of these operations it is possible to obtain derived tables needed in the analysis.

Before presenting the statistical operations we introduce the general structure of a multiway table in the edit field and various auxiliary operations to be used for table management.

6.1. Definition of a multiway table

A multiway table is an n-dimensional array consisting either of data values or frequencies. The n dimensions governing this array are usually variables on a nominal scale. They are often called factors. We here use a general term classifier.

When n=1 or 2 the table can readily be presented in a simple tabular form, but when n>2 difficulties may arise, since no multidimensional representation is possible directly in two dimensions.

Usually three- and more dimensional tables are represented by entering some of the classifiers in a nested form like in Display 6.1 where we have Class and Age as row classifiers and Sex and Vote as column classifiers. (This display is same as Disp.2.17.)

Disp.6.1

```
      1  SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
 16  *TABLE PAYNE,17,37,F_
 17  *                Sex    Male          Female
 18  *                Vote   Cons  Labour Cons Labour
 19  *Class   Age     ******
 20  *
 21  *upper   >73              4     0     10    0
 22  *        51-73           27     8     26    9
 23  *        41-50           27     4     25    9
 24  *        26-40           17    12     28    9
 25  *        <26              7     6      7    3
 26  *
 27  *lower   >73              8     4      9    2
 28  *        51-73           21    13     33    8
 29  *        41-50           27    12     29    4
 30  *        26-40           14    15     17   13
 31  *        <26              9     9     13    7
 32  *
 33  *work    >73              8    15     17    4
 34  *        51-73           35    62     52   53
 35  *        41-50           29    75     32   70
 36  *        26-40           32    66     36   67
 37  *        <26             14    34     18   33
```

In principle, we can have any number of row and column classifiers the total number of them being the dimension n.

The one- and two-dimensional subtables can be treated simply by standard operations introduced before. For handling of a multiway table as an integrated entity we must provide a TABLE specification which is on line 16 in the previous example.

The general form of a TABLE specification is

TABLE <name of table>,L1,L2,<type>

where L1-L2 are lines for labels and elements in the table,
and <type> is either X (=elements are data values) or
F (=elements are frequencies).

In the sequel we shall speak about X- and F-tables. Most of the auxiliary operations treat all tables similarly, but in certain cases the treatment varies depending on the type.

Observe that the TABLE specification tells nothing about the dimension and the classifiers. EDITOR can "see" this information directly from the layout by the aid of the auxiliary line (line 19 in the example) containing one or several asterisks (*) as guidance.

The names of the row classifiers (Class, Age) are typed on the same line to the left from *'s and the names of the column classifiers are above the *'s in the same column. The number of *'s defines the maximum length of labels.

The class names used as labels for columns and rows must be contiguos strings. They can be typed hierarchically as above, or completely. The contents of the table must always obey the hierarchy specified by the order of classifiers and the number of distinct classes (class names) in label rows and columns. In case of ambiguous labels the general structure interpreted by EDITOR defines the position of each element in the n-dimensional table.

To make the table structure clearer for the reader extra empty lines (as lines 26 and 32 in the example) may be inserted between the rows.

6.2. Table management and editing

The following TAB-operations are available for table modifications and they insert the modified table from line L onwards and label it by name <old name><4th letter of operator>.

TABC <table>,L
    copies a table.

TABS <table>,S,L
    displays the table with S row classifers (S=0,1,2,...,N).

TABM <table>,A,B,L
    changes the positions of the classifiers A and B.

TABI <table>,C,C1,C2,L
    changes the positions of classes C1 and C2 for classifier C.

TABD <table>,C,L
    deletes classifier C.
    In a <u>frequency table</u> the elements of the modified table are the marginal sums of C-classification; In this case TABD means collapsing over classifier C and the result is an (n-1)-dimensional table.
    In a <u>data table</u> the C-classification will be changed to a N-classification where the C-class names are replaced by indices 1,2,... The modified table has still n-dimensional and has the old structure and contents; only the name C is replaced by N.
    If TABD is applied to a data table which already includes an N-classification, the C-classification will be united to the N-classification and now the dimension of the table will be decreased by 1 but the number of elements remains the same.

The rationale behind TABD is obvious in case of F-tables, but in case of X-tables some comments are needed. The classifier N with class names 1,2,3,... corresponds to pure ordering of replications in experimental designs. When deleting a classifier the elements are still preserved, but they are arranged within an augmented N-classification. The result of using TABD successively for all classifiers in a X-table will be a one-dimensional table with a single vector of all the original elements along an N-classification.

TABJ <F-table>,C,C1,C2,Cn,L
    combines (in a frequency table) the classes C1 and C2 in the C-classification and calls the united class by Cn. The class Cn takes the position of C1 and C2-class will be cancelled.

TAB+ <F-table 1>,<F-table 2>,L
    forms the sum of tables 1 and 2.

TAB- <F-table 1>,<F-table 2>,L
forms the difference of tables 1 and 2.

As an example on, how these operations work in practice we shall make certain modifications to the following, artificial 4-dimensional table:

Disp.6.2
```
       1  SURVO 76 EDITOR   (C)1979 S.Mustonen   (100x100)
  1    *
  2    *
  3    *
  4    *TABLE TEST,5,11,X
  5    *           A  A1      A2      A3
  6    *           B  B1 B2 B3 B1 B2 B3 B1 B2 B3
  7    * C   D     **
  8    * C1  D1        1  5  9 13 17 21 25 29 33
  9    *     D2        2  6 10 14 18 22 26 30 34
 10    * C2  D1        3  7 11 15 19 23 27 31 35
 11    *     D2        4  8 12 16 20 24 28 32 36
 12    *
```

In this representation A and B are column classifiers with 3 classes each while C and D are dichotomous row classifiers.
If we like to have also C as a column classifier this situation will be reached by a TABS operation:

```
 13    *TABS TEST,1,14_
```

which gives to us:

Disp.6.3
```
 12    *
 13    *TABS TEST,1,14_
 14    *TABLE TESTS,15,20,X
 15    *   A   A1                A2                A3
 16    *   B   B1    B2    B3    B1    B2    B3    B1    B2    B3
 17    *   C   C1 C2 C1 C2 C1 C2 C1 C2 C1 C2 C1 C2 C1 C2 C1 C2 C1 C2
 18    *D  **
 19    *D1      1  3  5  7  9 11 13 15 17 19 21 23 25 27 29 31 33 35
 20    *D2      2  4  6  8 10 12 14 16 18 20 22 24 26 28 30 32 34 36
 21    *
```

or when we like to have all classifiers on the row side it is enough to change the parameter 1 to 4 in the TABS operation and after the execution we obtain the result:

Disp.6.4
```
 12    *
 13    *TABS TEST,4,14
 14    *TABLE TESTS,15,51,X
 15    *A  B  C  D  **
 16    *A1 B1 C1 D1       1
 17    *         D2       2
 18    *      C2 D1       3
 19    *         D2       4
 20    *   B2 C1 D1       5
 21    *         D2       6
 22    *      C2 D1       7
 23    *         D2       8
```

In this form the table has 36 rows. Only 8 first of them are now visible, but the user can scan the table easily by scrolling the field with edit keys.
Observe that this representation is closely related to the normal data matrix. By means of standard editing operations it is easy to modify this table into the form:

Disp.6.5
```
 12   *
 13   *TABS TEST,4,14
 14   *
 15   *      A   B   C   D         X
 16   * 1    1   1   1   1         1
 17   * 2    1   1   1   2         2
 18   * 3    1   1   2   1         3
 19   * 4    1   1   2   2         4
 20   * 5    1   2   1   1         5
 21   * 6    1   2   1   2         6
 22   * 7    1   2   2   1         7
 23   * 8    1   2   2   2         8
```

and then we have a standard data matrix of 36 observations and 5 variables. On the other hand, if we load in the edit field a SURVO 76 data file having this structured representation it is possible to reach other representations by the TABS operation.

In order to change the positions of two classifiers the TABM oper-operation is to be employed. For example,
```
 13   *TABM TEST,A,D,14_
```
changes the places of A- and D-classification giving:

Disp.6.6
```
 12   *
 13   *TABM TEST,A,D,14_
 14   *TABLE TESTM,15,23,X
 15   *         D    D1       D2
 16   *         B    B1 B2 B3 B1 B2 B3
 17   *C   A    **
 18   *C1  A1         1  5  9  2  6 10
 19   *    A2        13 17 21 14 18 22
 20   *    A3        25 29 33 26 30 34
 21   *C2  A1         3  7 11  4  8 12
 22   *    A2        15 19 23 16 20 24
 23   *    A3        27 31 35 28 32 36
 24   *
```

Moreover, if some classes ought to be interchanged, the TABI will do that. For instance, in the modified table TESTM above the the classes D1 and D2 are interchanged (in situ) with
```
 13   *TABI TESTM,D,D1,D2,14_
```
and we obtain

Disp.6.7
```
 12   *
 13   *TABI TESTM,D,D1,D2,14
 14   *TABLE TESTMI,15,23,X
 15   *         D    D2       D1
 16   *         B    B1 B2 B3 B1 B2 B3
 17   *C   A    **
 18   *C1  A1         2  6 10  1  5  9
 19   *    A2        14 18 22 13 17 21
 20   *    A3        26 30 34 25 29 33
 21   *C2  A1         4  8 12  3  7 11
 22   *    A2        16 20 24 15 19 23
 23   *    A3        28 32 36 27 31 35
 24   *
```

Observe that table TESTMI now overwrites TESTM.

The preceding operations work similarly on X- and F-tables, but in TABD the result is different depending on whether an X or F is appearing as the 4th parameter in TABLE specification.

An example of TABD was already presented in chapter 2 (Displays 2.18-19). Now let us consider X-table TEST in Disp.6.2 again.

Assume that we like to delete both the row classifiers C,D in order to have the data set structured only by classifiers A,B. This will be achieved in two steps. At first perform a TABD operation

```
 13    *TABD TEST,D,14_
```

and obtain:

Disp.6.8

```
 12    *
 13    *TABD TEST,D,14_
 14    *TABLE TESTD,15,21,X
 15    *          A    A1         A2         A3
 16    *          B    B1 B2 B3 B1 B2 B3 B1 B2 B3
 17    *C   N    **
 18    *C1   1          1  5  9 13 17 21 25 29 33
 19    *     2          2  6 10 14 18 22 26 30 34
 20    *C2   1          3  7 11 15 19 23 27 31 35
 21    *     2          4  8 12 16 20 24 28 32 36
 22    *
```

Observe now that the effect was simply changing the D-classification into N-classification with classes 1,2. Now to delete also C-classification, edit line 13 into form

```
 13    *TABD TESTD,C,14_
```

and activation of this operation gives finally:

Disp.6.9

```
 12    *
 13    *TABD TESTD,C,14_
 14    *TABLE TESTDD,15,21,X
 15    *    A    A1         A2         A3
 16    *    B    B1 B2 B3 B1 B2 B3 B1 B2 B3
 17    *N   **
 18    * 1          1  5  9 13 17 21 25 29 33
 19    * 2          2  6 10 14 18 22 26 30 34
 20    * 3          3  7 11 15 19 23 27 31 35
 21    * 4          4  8 12 16 20 24 28 32 36
 22    *
```

and hence we have a dummy N-classification left in order to keep record of similar observations with respect to the remaining A- and B-B-classifications.

Of course, the same result could have been reached in this case by simple editing.

The TABJ operation works only on F-tables. If we like to combine classes 'upper' and 'lower' in 'Class'-classification, the TABJ operation does that as follows:

Disp.6.10

```
38  *
39  *TABJ PAYNE,Class,upper,lower,middle,40_
40  *TABLE PAYNEJ,41,53,F
41  *                  Sex     Male           Female
42  *                  Vote    Cons   Labour Cons   Labour
43  *Class   Age       ******
44  *middle  >73                 12      4     19      2
45  *        51-73               48     21     59     17
46  *        41-50               54     16     54     13
47  *        26-40               31     27     45     22
48  *        <26                 16     15     20     10
49  *work    >73                  8     15     17      4
50  *        51-73               35     62     52     53
51  *        41-50               29     75     32     70
52  *        26-40               32     66     36     67
53  *        <26                 14     34     18     33
54  *
```

6.3. Analysis of variance

The ANOVA operation computes the analysis of variance table for any X-table according to the standard n-factorial layout. Thus analysis of variance for one-way, two-way, three-way etc. classification with equal number of observations in each combination of factors can be computed. Also Latin squares and some other balanced experiments may be handled by pooling results from one or more ANOVA operations (see e.g. Afifi,Azen, 1979).
In SURVO 76 EDITOR analysis of variance tables for those more special experiments may be formed by numerical editing operations after some ANOVA operations.

The ANOVA operation has a simple structure ANOVA <X-table>,L where L is the first line for the ANOVA table.

For example, to compute the results for the following 3x2x2x2 factorial experiment:

Disp.6.11

```
     1  SURVO 76 EDITOR   (C)1979 S.Mustonen    (100x100)
 1  *
 2  *
 3  *TABLE EXP,4,11,X
 4  *           A   1         2         3
 5  *           B   1   2     1   2     1   2
 6  * C   D     *
 7  * 1   1       1.3 1.2   1.8 1.5   2.3 2.1
 8  *     2       1.2 1.0   1.7 1.5   2.4 2.0
 9  *
10  * 2   1       1.5 1.4   2.2 1.9   2.6 2.4
11  *     2       1.5 1.3   2.3 2.0   2.9 2.7
12  *
```

we activate the ANOVA operation

```
13  *ANOVA EXP,14_
```

and obtain the results:

Disp.6.12

```
13   *ANOVA EXP,14_
14   *ANALYSIS OF VARIANCE ON TABLE EXP
15   *SOURCE OF          SUM OF    DEGREES OF   MEAN
16   *VARIATION          SQUARES   FREEDOM      SQUARE
17   * A                 5.0625       2         2.5313
18   * B                 0.3038       1         0.3038
19   * AB                0.0175       2         0.0088
20   * C                 0.9204       1         0.9204
21   * AC                0.0608       2         0.0304
22   * BC                0.0004       1         0.0004
23   * ABC               0.0058       2         0.0029
24   * D                 0.0038       1         0.0038
25   * AD                0.0625       2         0.0313
26   * BD                0.0038       1         0.0038
27   * ABD               0.0075       2         0.0038
28   * CD                0.0504       1         0.0504
29   * ACD               0.0108       2         0.0054
30   * BCD               0.0004       1         0.0004
31   * ABCD              0.0058       2         0.0029
32   * TOTAL             6.5163      23         0.2833
33   *
```

Next we consider the following example taken from Hald (1952),p.444.

Disp.6.13

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 1   *
 2   *
 3   *
 4   *   The quality of paper from a paper mill was tested at regular inter-
 5   *vals by taking samples and determining their breaking strength.
 6   *We test the hypothesis that the variation of breaking strength is ran-
 7   *dom by using the following data set consisting of 11 sets of samples
 8   *and 5 determinations of breaking strength made in each set.
 9   *
10   *TABLE PAPER,11,23,X
11   *       N   1   2   3   4   5
12   * Set   ***
13   *  1         9.1 8.5 8.6 7.2 7.6
14   *  2         8.1 8.2 7.6 7.7 8.0
15   *  3         8.5 9.0 8.1 8.5 8.0
16   *  4         7.4 7.2 8.3 8.5 8.5
17   *  5         9.0 7.9 8.2 7.6 8.7
18   *  6         8.6 8.8 9.2 8.8 9.8
19   *  7         8.5 8.1 8.8 8.6 8.5
20   *  8         7.8 8.8 7.6 8.9 8.7
21   *  9         8.7 8.2 8.7 9.2 8.7
22   * 10         8.5 8.9 7.7 8.1 9.0
23   * 11         7.3 7.9 7.9 8.5 9.2_
```

At first we execute an ANOVA operation and obtain the following results:

Disp.6.14

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
 21   * 9         8.7 8.2 8.7 9.2 8.7
 22   * 10        8.5 8.9 7.7 8.1 9.0
 23   * 11        7.3 7.9 7.9 8.5 9.2
 24   *
 25   *ANOVA PAPER,26_
 26   *ANALYSIS OF VARIANCE ON TABLE PAPER
 27   *SOURCE OF          SUM OF     DEGREES OF   MEAN
 28   *VARIATION          SQUARES    FREEDOM      SQUARE
 29   * N                 0.8764        4         0.2191
 30   * S                 5.0873       10         0.5087
 31   * NS               11.7636       40         0.2941
 32   * TOTAL            17.7273       54         0.3283
 33   *
```

At this stage we have the results in the form of two-factor (N,S) layout, but since the variation within sets is assumed to be random the sums of squares and degrees of freedom for N and NS variation must be pooled by simple addition.

This is done by means of a L+ operation (see Disp.5.15) as follows:

Disp.6.15

```
 24   *
 25   *ANOVA PAPER,26
 26   *ANALYSIS OF VARIANCE ON TABLE PAPER
 27   *SOURCE OF          SUM OF     DEGREES OF   MEAN
 28   *VARIATION          SQUARES    FREEDOM      SQUARE
 29   * N                 0.8764        4         0.2191
 30   * S                 5.0873       10         0.5087
 31   * NS               11.7636       40         0.2941
 32   * TOTAL            17.7273       54         0.3283
 33   *
 34   * N                 0.8764        4         0.2191
 35   * NS               11.7636       40         0.2941
 36   *                  12.6400       44
 37   *                  11.1111       11
 38   *L+ 34,35,37,36_
 39   *
 40   *
```

We have copied (by F15) the lines 29 and 31 to lines 34-35 and computed the required sums by the L+ operation. The result corresponding to the variation within sets is then on line 36.

By inserting new labels and rearranging the lines in the ANOVA table the result may be displayed in the traditional form:

Disp.6.16

```
 26   *ANALYSIS OF VARIANCE ON TABLE PAPER
 27   *SOURCE OF          SUM OF     DEGREES OF   MEAN
 28   *VARIATION          SQUARES    FREEDOM      SQUARE
 29   * Between sets      5.0873       10         0.5087
 30   * Within sets      12.6400       44         0.2873
 31   * Total            17.7273       54
 33   *
```

As a third example we consider a Latin square with 5 treatments. (Also this one is taken from Hald 1952, p.508.)

Disp. 6.17

```
         1  SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 11     *TABLE HALD,12,18,X
 12     *  B    1    2    3    4    5
 13     *A  *
 14     *1      7.6 11.8 17.6  8.8 17.9
 15     *2     21.4 12.9 12.4 15.0 20.6
 16     *3     16.0  9.7  7.4 18.4 16.6
 17     *4     16.0 18.3 23.6 27.4 25.2
 18     *5     23.3 30.5 25.8 24.5 26.6
 19     *
 20     *1        C    E    A    D    B
 21     *2        B    D    E    C    A
 22     *3        A    C    D    B    E
 23     *4        E    B    C    A    D
 24     *5        D    A    B    E    C
 25     *
```

There results of an experiment have been entered as an X-table HALD according to two criteria A,B. The 5 replications of the 5 treatments are distributed over this square according to the Latin square represented on lines 14-18; we call the corresponding factor C.

To create the ANOVA table for this Latin Square we perform two ANOVA operations; the first of them directly on table HALD and the second on a modified table (HALDC) where the rows of table HALD have been permuted according to the lines 14-18 by a PERM operation. Thus we shall have two ANOVA tables; the first on factors A,B and the second on factors A,C and the final results for the Latin square experiment will be obtained as a simple combination.

The whole story with comments inserted to the right side can be summarized as follows:

Disp.6.18

```
     Display on the CRT                                Comments
     (lines entered by the user
      are indicated by -)
-11  *TABLE HALD,12,18,X                               Lines 11-18:
-12  *  B    1    2    3    4    5                     Entering the observations
-13  *A  *                                             according to the levels of
-14  *1      7.6 11.8 17.6  8.8 17.9                   factors A and B as a table HALD
-15  *2     21.4 12.9 12.4 15.0 20.6
-16  *3     16.0  9.7  7.4 18.4 16.6
-17  *4     16.0 18.3 23.6 27.4 25.2
-18  *5     23.3 30.5 25.8 24.5 26.6
 19  *
-20  *1        C    E    A    D    B                   Lines 20-24:
-21  *2        B    D    E    C    A                   Entering the third factor C
-22  *3        A    C    D    B    E                   as a Latin square
-23  *4        E    B    C    A    D
-24  *5        D    A    B    E    C
 25  *
-26  *TABC HALD,28                                     Copy HALDC of HALD on 28-35
-27  *PERM 31,35,20                                    PERM permutes the copy HALDC
 28  *TABLE HALDC,29,35,X                              according to the Latin square.
 29  *  C    1    2    3    4    5                     B is replaced by C.
 30  *A  *
 31  *1     17.6 17.9  7.6  8.8 11.8
 32  *2     20.6 21.4 15.0 12.9 12.4
 33  *3     16.0 18.4  9.7  7.4 16.6
 34  *4     27.4 18.3 23.6 25.2 16.0
 35  *5     30.5 25.8 26.6 23.3 24.5
 36  *
```

```
-37  *ANOVA HALD,38                                  ANOVA computes the ANOVA table
 38  *ANALYSIS OF VARIANCE ON TABLE HALD             for HALD.
 39  *SOURCE OF         SUM OF     DEGREES OF
 40  *VARIATION         SQUARES    FREEDOM
 41  * B                77.154      4
 42  * A               660.342      4
 43  * BA              301.890     16
 44  * TOTAL          1039.386     24
 45  *
-46  *ANOVA HALDC,47                                 ANOVA computes the ANOVA table
 47  *ANALYSIS OF VARIANCE ON TABLE HALDC            for HALDC.
 48  *SOURCE OF         SUM OF     DEGREES OF
 49  *VARIATION         SQUARES    FREEDOM
 50  * C               181.546      4
 51  * A               660.342      4
 52  * CA              197.498     16
 53  * TOTAL          1039.386     24
 54  *                                               Copying by F15:
 55  *ANALYSIS OF VARIANCE ON LATIN SQUARE HALD      copy of line 38 (modified)
 56  * A               660.342      4                copy of line 42
 57  * B                77.154      4                copy of line 41
 58  * C               181.546      4                copy of line 50
 59  * RESIDUAL        120.344     12
 60  * TOTAL          1039.386     24                copy of line 44
 61  *                 919.042     12
-62  *                1234.123     12                image for L-operations below
-63  *L+ 56,58,62,61                                 forming 61 as a sum of 56-58
-64  *L- 60,61,62,59                                 forming residual line 59
```

6.4. Log-linear models for contingency tables

For the estimation of log-linear models for F-tables the TABFIT operation is available. We have used this operation already in the introductory example (see chapter 2, especially displays 2.20-25).

To characterize the structure of the TABFIT operation assume that we have a 4-way F-table as the TABLE PAYNE in displays 2.17 and 6.1. Here the classifiers are Age, Class, Vote and Sex. When specifying a log-linear model these variables are referred to by the initials as A,C,V and S in this case.

The general form of the TABFIT operation is

TABFIT <F-table>,K,L

where K is an image line specifying the model and L is the first line for the results.

Model specification on line K has the structure:

LOGLIN <list of marginals to be fitted>

Only hierarchical hypotheses may be tested (see e.g. Payne 1977, Bishop, Fienberg, Holland 1975), and hence it is enough to list the sufficient marginals in the model specification. For instance, if the three-dimensional marginal ACV will be fitted there is no need to mention the marginals AC,AV,CV,A,C and V in the specification.

For example, to estimate log-linear model for table PAYNE with marginals VSC,VA,SA,CA fitted we enter the TABFIT operation:

```
 39    *TABFIT PAYNE,40,41
 40    *LOGLIN VSC,VA,SA,CA
```

and activate it. During the estimation the edit field disappears from the screen and we will have a display telling us how the iterative proportional fitting (IPS) procedure is acting:

Disp.6.19

```
FITTING MARGINALS BY ITERATION (IPS METHOD):
# OF ITERATION       FIT
 1                7.85676411443
 2                .4350143992
 3                .0542655821
 4                .0074416167
 5                .0010219528
 6                .0001402384
```

and after the desired accuracy is reached the edit field will appear again with results:

Disp.6.21

```
       1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
 39    *TABFIT PAYNE,40,41 _
 40    *LOGLIN VSC,VA,SA,CA
 41    *G↑2=  37.62 DF=  28 P=0.106 FITTED: VSC,VA,SA,CA    (PAYNE)
 42    *TABLE PAYNEF,43,60,F
 43    *                   Sex      Male            Female
 44    *                   Vote     Cons   Labour Cons   Labour
 45    *Class    Age       ******
 46    *upper    >73                   6       1      6       1
 47    *         51-73                25       7     30       7
 48    *         41-50                23       9     25       8
 49    *         26-40                21      10     26      10
 50    *         <26                   7       4      9       4
 51    *lower    >73                   9       3     10       2
 52    *         51-73                23      13     31       8
 53    *         41-50                22      15     26       9
 54    *         26-40                15      13     21       9
 55    *         <26                  10       9     13       6
 56    *work     >73                  11      10     14       9
 57    *         51-73                36      61     48      56
 58    *         41-50                33      72     40      61
 59    *         26-40                26      71     36      68
 60    *         <26                  13      37     17      33
 61    *
```

On the first line for the results (41) the goodness-of-fit statistics are displayed. On the next lines is the table of fitted frequencies. These results could be presented more accurately by editing at least one of the original frequencies to a form corresponding to the desired accuracy and by activating TABFIT again.

The TABFIT operation applies also to incomplete multiway tables containing structural zeros. For the structural zeros the common notation "-" should be used.

The following example is taken from Bishop, Fienberg, Holland (1975), p.224.

Disp.6.21

```
     1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 40   *
 41   *   Meadow Spittlebug Color Forms by Sex and Location
 42   *   (Color forms MAR,LAT and FLA are absent in the male sex.)
 43   *
 44   *
 45   *TABLE BUG,46,53,F
 46   *           COL  POP TYP TRI MAR LAT FLA ALB
 47   *SEX    LOC  *****
 48   *M      high      23  55   2   -   -   -   1
 49   *       low       67 186  30   -   -   -   0
 50   *       grass     24  60  12   -   -   -   0
 51   *F      high      28  61  12  11   3   1   3
 52   *       low       63 148  17   9   6   1   3
 53   *       grass     40  75  15  11   4   0   0.00
 54   *
```

To fit a model with Sex-Locale two-factor effect we enter the TABFIT operation

```
 55   *TABFIT BUG,56,57
 56   *LOGLIN SL,C
```

and obtain the result:

Disp.6.22

```
 54   *
 55   *TABFIT BUG,56,57
 56   *LOGLIN SL,C
 57   *G↑2=  32.17 DF= 21 P=0.056 FITTED: SL,C   (BUG)
 58   *TABLE BUGF,59,66,F
 59   *            COL        POP    TYP    TRI    MAR    LAT    FLA    ALB
 60   *SEX    LOC   *****
 61   *M      high            21.45  51.23   7.71   0.00   0.00   0.00   0.61
 62   *       low             74.96 178.98  26.92   0.00   0.00   0.00   2.14
 63   *       grass           25.43  60.71   9.13   0.00   0.00   0.00   0.73
 64   *F      high            28.68  68.48  10.30   7.22   3.03   0.47   0.82
 65   *       low             59.53 142.15  21.38  14.98   6.28   0.97   1.70
 66   *       grass           34.95  83.45  12.55   8.80   3.69   0.57   1.00
```

Observe that the fitted frequencies are now displayed with 2 decimals, since one element in table BUG was entered in the form 0.00.

When estimating several alternative models on the same data it is useful to place the results on consecutive lines for comparison as we alredy have done in the introductory example (see display 2.24). Then also the conditional G↑2 statistics can be computed by means of an L-operation.

7. Correlations and linear regression analysis

SURVO 76 includes several modules for regression analysis. There is no urgent need to perform computations of this type directly using EDITOR, since the results from various regression programs can be easily taken into the edit field for final reporting.

It might to some extent, however, be useful to have correlation and regression analysis in editor mode. This is especially so when using small data sets in a teaching situation.

Therefore EDITOR provides some operations from this area, too. These operations (CORR and REGRAN) assume the data set to be in the form of a data matrix and defined with a DATA specification. Each data set which is loaded from a SURVO 76 data file to the edit field automatically has this specification (see display 5.18).

The DATA specification is:

DATA <name of data set>,L1,L2,K

where L1-L2 are the lines for the data values in a matrix form
and K is a line for the names of variables (columns).
The data set is to be represented in the form described in 5.4.

Observe that (typically for EDITOR) the DATA specification may be located on any line outside the data itself. Thus various subsets of the same data may be defined simultaneously by different DATA specifications. Normally, of course, DATA is typed above the data matrix.

The data appearing in display 5.16 is now defined with a following DATA specification:

Disp.7.1

```
          1  SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
  6     *
  7     *DATA DRINKS,11,22,8_
  8     *           Coffee Tea  Beer  Wine Spirits
  9     *             (kg)  (kg)  (l)   (l)  (l)
 10     *
 11     * Sweden       12.9 0.30  58.3   7.9 2.9
 12     * Denmark      11.8 0.41 113.9  10.4 1.7
 13     * Finland      12.5 0.15  54.7   7.6 2.7
 14     * Norway        9.4 0.19  43.5   3.1 1.8
 15     * France        5.2 0.10  44.5 104.3 2.5
 16     * Ireland       0.2 3.73 124.5   3.8 1.9
 17     * Italy         3.6 0.06  13.6 106.6 2.0
 18     * Holland       9.2 0.58  75.5   9.7 2.7
 19     * Portugal      2.2 0.03  27.5  89.3 0.9
 20     * Switzerland  9.1 0.25  73.5  44.9 2.1
 21     * Spain         2.5 0.03  43.6  73.2 2.7
 22     * England       1.8 3.49 113.7   5.1 1.4
 23     *
```

The CORR operation for computing means, standard deviations and correlations can be used as follows. To have the results for all variables in DRINKS we enter:

Disp.7.2

```
 20     * Switzerland  9.1 0.25  73.5  44.9 2.1
 21     * Spain         2.5 0.03  43.6  73.2 2.7
 22     * England       1.8 3.49 113.7   5.1 1.4
 23     *
 24     *              XXXX XXXX XXXXX XXXXX XXX
 25     *CORR DRINKS,24,26_
```

where 24 is the line specifying the columns (variables) and 26 is the line for the results. Thus by activating CORR we shall have:

Disp.7.3

```
24   *             XXXX XXXX XXXXX XXXXX XXX
25   *CORR DRINKS,24,26_
26   *                 MEAN             STD.DEV.
27   *  Coffee           6.700000         4.606319
28   *  Tea              0.776666         1.334352
29   *  Beer            65.566666        35.702618
30   *  Wine            38.825000        42.506365
31   *  Spirits          2.108333         0.611196
32   *
33   *CORR:     Coffee Tea    Beer   Wine   Spirit
34   *Coffee     1.000 -0.511 -0.022 -0.445  0.465
35   *Tea       -0.511  1.000  0.764 -0.453 -0.311
36   *Beer      -0.022  0.764  1.000 -0.690 -0.147
37   *Wine      -0.445 -0.453 -0.690  1.000 -0.100
38   *Spirits    0.465 -0.311 -0.147 -0.100  1.000
39   *
```

The REGRAN operation computes the parameters and the residuals of a linear regression model and has the general form:

REGRAN <name of data set>,K,L

where image line K specifies the model and L is the first line for the results.
K has the form " YYYY   XXXX   XXXX   XXXX   -123.12   ", where the Y's indicate the column of the dependent variable, X's indicate the columns of independent variables (regressors) and -123.12 indicates the place and the format for the residuals which will be displayed as a new column in the data matrix.
To have a constant term in the model a column of 1's has to be inserted as a regressor in the data set.

For example, if we like to estimate a model for the dependence of beer consumption on the consumption of other beverages in DRINKS, we shall proceed as follows:

At first a column of 1's is generated by a SET operation:

Disp.7.4

```
      1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (100x100)
 1   *
 2   *
 3   *
 4   *
 5   *SET 11,22,6_
 6   *                                        1
 7   *DATA DRINKS,11,22,8
 8   *             Coffee Tea  Beer  Wine Spirits Constant
 9   *              (kg)  (kg)  (l)   (l)  (l)
10   *
11   * Sweden        12.9 0.30  58.3   7.9 2.9      1
12   * Denmark       11.8 0.41 113.9  10.4 1.7      1
13   * Finland       12.5 0.15  54.7   7.6 2.7      1
14   * Norway         9.4 0.19  43.5   3.1 1.8      1
15   * France         5.2 0.10  44.5 104.3 2.5      1
16   * Ireland        0.2 3.73 124.5   3.8 1.9      1
17   * Italy          3.6 0.06  13.6 106.6 2.0      1
18   * Holland        9.2 0.58  75.5   9.7 2.7      1
19   * Portugal       2.2 0.03  27.5  89.3 0.9      1
20   * Switzerland    9.1 0.25  73.5  44.9 2.1      1
21   * Spain          2.5 0.03  43.6  73.2 2.7      1
22   * England        1.8 3.49 113.7   5.1 1.4      1
23   *
```

Also the label 'Constant' has been typed on line 8.
To perform the regression analysis we enter the REGRAN operation in the form:

Disp.7.5
```
 7    *DATA DRINKS,11,22,8
 8    *            Coffee Tea  Beer  Wine Spirits Constant
 9    *             (kg)  (kg)  (l)   (l)  (l)
10    *
11    * Sweden      12.9 0.30  58.3   7.9 2.9     1
12    * Denmark     11.8 0.41 113.9  10.4 1.7     1
13    * Finland     12.5 0.15  54.7   7.6 2.7     1
14    * Norway       9.4 0.19  43.5   3.1 1.8     1
15    * France       5.2 0.10  44.5 104.3 2.5     1
16    * Ireland      0.2 3.73 124.5   3.8 1.9     1
17    * Italy        3.6 0.06  13.6 106.6 2.0     1
18    * Holland      9.2 0.58  75.5   9.7 2.7     1
19    * Portugal     2.2 0.03  27.5  89.3 0.9     1
20    * Switzerland  9.1 0.25  73.5  44.9 2.1     1
21    * Spain        2.5 0.03  43.6  73.2 2.7     1
22    * England      1.8 3.49 113.7   5.1 1.4     1
23    *             XXXX XXXX YYYYY XXXXX XXX     X           -123.1
24    *REGRAN DRINKS,23,25_
```

and after the execution the display is:

Disp.7.6
```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (100x100)
 7    *DATA DRINKS,11,22,8
 8    *            Coffee Tea  Beer  Wine Spirits Constant
 9    *             (kg)  (kg)  (l)   (l)  (l)
10    *
11    * Sweden      12.9 0.30  58.3   7.9 2.9     1            -16.1
12    * Denmark     11.8 0.41 113.9  10.4 1.7     1             35.4
13    * Finland     12.5 0.15  54.7   7.6 2.7     1            -15.0
14    * Norway       9.4 0.19  43.5   3.1 1.8     1            -19.2
15    * France       5.2 0.10  44.5 104.3 2.5     1              5.7
16    * Ireland      0.2 3.73 124.5   3.8 1.9     1              4.7
17    * Italy        3.6 0.06  13.6 106.6 2.0     1            -19.9
18    * Holland      9.2 0.58  75.5   9.7 2.7     1              7.4
19    * Portugal     2.2 0.03  27.5  89.3 0.9     1             -5.0
20    * Switzerland  9.1 0.25  73.5  44.9 2.1     1             12.4
21    * Spain        2.5 0.03  43.6  73.2 2.7     1             17.8
22    * England      1.8 3.49 113.7   5.1 1.4     1             -8.2
23    *             XXXX XXXX YYYYY XXXXX XXX     X           -123.1
24    *REGRAN DRINKS,23,25_
25    *REGRESSION ANALYSIS: REGRESSAND:'Beer'  DATA:'DRINKS'
26    *   TOTAL VARIANCE=     1274.676 DF= 11
27    *   RESIDUAL VARIANCE=   456.039 DF=  7  R↑2=0.7723
28    * VARIABLE       REGR.COEFF.  STD.DEVIATION      T
29    * Coffee            3.983421       3.61848      1.100
30    * Tea              26.583812      12.08591      2.199
31    * Wine             -0.015917       0.36583     -0.043
32    * Spirits          -4.596942      12.02483     -0.382
33    * Constant         28.540865      48.56771      0.587
34    *
```

From this point on it is simple to continue the analysis by means of various editing and numerical operations. For example, the predicted values for 'Beer' can be computed as a difference between 'Beer' and the residuals by a C- operation.

In this environment it is simple to consider the data set ordered in different ways. For instance, to see the order of countries according to the "overconsumption" of beer the data set can be sorted using the residual column as a sort key and the result after a SORT operation is

Disp.7.7

```
        1   SURVO 76 EDITOR     (C)1979 S.Mustonen      (100x100)
  5    *-SORT 11,22,6_
  6    *                                                     11111
  7    *DATA DRINKS,11,22,8
  8    *            Coffee Tea  Beer  Wine Spirits Constant Residual
  9    *             (kg)  (kg)  (l)   (l)   (l)
 10    *
 11    * Denmark      11.8 0.41 113.9  10.4 1.7     1           35.4
 12    * Spain         2.5 0.03  43.6  73.2 2.7     1           17.8
 13    * Switzerland   9.1 0.25  73.5  44.9 2.1     1           12.4
 14    * Holland       9.2 0.58  75.5   9.7 2.7     1            7.4
 15    * France        5.2 0.10  44.5 104.3 2.5     1            5.7
 16    * Ireland       0.2 3.73 124.5   3.8 1.9     1            4.7
 17    * Portugal      2.2 0.03  27.5  89.3 0.9     1           -5.0
 18    * England       1.8 3.49 113.7   5.1 1.4     1           -8.2
 19    * Finland      12.5 0.15  54.7   7.6 2.7     1          -15.0
 20    * Sweden       12.9 0.30  58.3   7.9 2.9     1          -16.1
 21    * Norway        9.4 0.19  43.5   3.1 1.8     1          -19.2
 22    * Italy         3.6 0.06  13.6 106.6 2.0     1          -19.9
```

Observe also how simple it is to make alterations in the data set and in the model definition, and see their their effects after reactivation of REGRAN.

8. Numerical computations in the edit field

Standard arithmetic calculations enriched with various elementary, statistical and user-defined functions can easily be carried out in the edit field.

8.1. Simple arithmetics

The arithmetic expressions to be evaluated are written according to the rules of BASIC. For instance, to compute the mean of the numbers 12, 17 and 25, type

Disp.8.1
```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *
  3    *          (12+17+25)/3=_
  4    *
  5    *
```

Now, this expression will be evaluated by pressing the key CONTINUE when the cursor is positioned after the "=" sign and we shall have the display

Disp.8.2
```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *
  3    *          (12+17+25)/3=18
  4    *
  5    *
```

Similarly, to compute the geometric mean we enter

Disp.8.3
```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *
  3    *          (12+17+25)/3=18
  4    *          (12*17*25)↑(1/3)=_
  5    *
```

and activation of this expression by pressing CONTINUE leads to

Disp.8.4
```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *
  3    *          (12+17+25)/3=18
  4    *          (12*17*25)↑(1/3)=17.2130062073
```

Since we are working in the edit field, it is always possible to modify the display by standard editing procedures. In this case the result could simply be rounded to three decimal places by moving the cursor to 17.2130062073 and pressing the 'erase' key F10 giving

Disp.8.5
```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *
  3    *          (12+17+25)/3=18
  4    *          (12*17*25)↑(1/3)=17.213_
```

In order to avoid repeating various constants and variables appearing in the expressions it is possible to use symbolic notation. For instance, the preceding example may be written in a more general form:

Disp.8.6
```
 6   *
 7   *            X=12, Y=17, Z=25
 8   * arithmetic mean    (X+Y+Z)/3=_
 9   * geometric  mean    (X*Y*Z)↑(1/3)=
10   *
```

If the expression on line 8 is activated by pressing CONTINUE, EDITOR is capable of finding the definitions of X,Y and Z and inserting their values in the expression considered. Thus, activation of the arithmetic mean leads to

Disp.8.7
```
 6   *
 7   *            X=12, Y=17, Z=25
 8   * arithmetic mean    (X+Y+Z)/3=18
 9   * geometric  mean    (X*Y*Z)↑(1/3)=
10   *
```

and the geometric mean can be activated too by moving the cursor to line 9 after the "-" sign and by pressing CONTINUE, thus giving

Disp.8.8
```
 6   *
 7   *            X=12, Y=17, Z=25
 8   * arithmetic mean    (X+Y+Z)/3=18
 9   * geometric  mean    (X*Y*Z)↑(1/3)=17.2130062073
10   *
```

The computations may be repeated with other initial values of the variables X,Y,Z immediately by editing the values needed and reactivating the expressions.

8.2. Computation schemes

Since the definitions of variables can be nested, i.e. a variable may be defined as a function of other variables, this approach enables the construction of general arithmetic computation schemes.

For instance, to compute values of the standard normal distribution function according to a well-known polynomial approximation, the following scheme typed in the edit field will be sufficient:

Disp.8.9

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 10   *
 11   *......................................................................
 12   *
 13   *   For X≥0, the standard normal distribution function F(X)
 14   *   is approximated by
 15   *   F=1-f*(b1*t+b2*t↑2+b3*t↑3+b4*t↑4+b5*t↑5) ,
 16   *   where
 17   *   f=(1/sqr(2*pi))*exp(-X↑2/2) (density function) and
 18   *   t=1/(1+r*X) , r=0.2316419
 19   *   b1=.31938153 ,      b2=-.356563782
 20   *   b3=1.781477937 ,    b4=-1.821255978
 21   *   b5=1.330274429 ,
 22   *   pi=3.14159265359 .
 23   *
 24   *   To compute F(X) enter X on line 26 and activate F
 25   *
 26   *   X=_   F=
 27   *
 28   *......................................................................
 29   *
```

The scheme needs no further explanations. The reader immediately sees the essential details and EDITOR can interpret all the components provided that everything is unambiguously defined.

Thus, inserting X=3 on line 26 and activating F on the same line gives the result

Disp.8.10

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 10   *
 11   *......................................................................
 12   *
 13   *   For X≥0, the standard normal distribution function F(X)
 14   *   is approximated by
 15   *   F=1-f*(b1*t+b2*t↑2+b3*t↑3+b4*t↑4+b5*t↑5) ,
 16   *   where
 17   *   f=(1/sqr(2*pi))*exp(-X↑2/2) (density function) and
 18   *   t=1/(1+r*X) , r=0.2316419
 19   *   b1=.31938153 ,      b2=-.356563782
 20   *   b3=1.781477937 ,    b4=-1.821255978
 21   *   b5=1.330274429 ,
 22   *   pi=3.14159265359 .
 23   *
 24   *   To compute F(X) enter X on line 26 and activate F
 25   *
 26   *   X=3   F=0.9986500327
 27   *
 28   *......................................................................
 29   *
```

Observe that this computation scheme includes nested definitions as F depends on t which depends on r, etc. EDITOR can compute expressions with up to 10 nested levels.

The order of the definitions in the edit field is immaterial. If multiple definitions occur for the same variable, EDITOR always employs the first one.

If a variable appearing in the scheme is not defined, an error message will be displayed when this variable is encountered during the computation process.

The search for definitions can be limited to a restricted area (<u>subfield</u>) in the edit field by inserting border lines of the form *.......... (asterisk and at least 10 .'s). Examples of them are lines 11 and 28 in the displays 8.9-10. Thus, the user may place several computation schemes in the same edit field and save them simultaneously in one edit file.

The SURVO 76 EDITOR computation schemes can be considered simple programs involving arithmetic operations in a free form. Since the schemes and the results may easily be edited by normal editing operations, the results can be printed on paper precisely in the way the user likes.

Combination of computation schemes with other activities provided by EDITOR increases the versatility of the SURVO 76 system.

8.3. Multiple activations

It may be often useful to obtain simultaneously several related results. This can be achieved by multiple activations as shown in the following computation scheme.

Disp.8.11

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 43    *
 44    *.........................................................................
 45    *
 46    *   If the radius of the sphere is r=10, then
 47    *   the volume is                4/3*pi*r↑3:=_
 48    *   the area of the surface is   4*pi*r↑2:=
 49    *   ( pi=3.14159265359 )
 50    *
 51    *.........................................................................
 52    *
```

If any of the expressions tailed with the characters := is activated by pressing CONTINUE when the cursor is placed just after =, <u>all</u> the expressions having the same tail and located in the same subfield will be evaluated.

Thus pressing CONTINUE when we have the display 8.11 leads to

Disp.8.12

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 43    *
 44    *.........................................................................
 45    *
 46    *   If the radius of the sphere is r=10, then
 47    *   the volume is                4/3*pi*r↑3:=4188.790204786
 48    *   the area of the surface is   4*pi*r↑2:=1256.637061436
 49    *   ( pi=3.14159265359 )
 50    *
 51    *.........................................................................
 52    *
```

It is thereafter possible to alter the value of r and reactivate one of the := expressions, thus obtaining new values for all of them simultaneously.

8.4. Conditional definitions

EDITOR also provides a simple <u>if</u>-<u>then</u>-<u>endif</u>-statement for conditional definitions in computation schemes when <u>multiple</u> activation is used. A scheme may include 20 parallel conditions. Nested conditions are not permitted.

The <u>if</u>-<u>then</u>-<u>endif</u>-statement has the structure

<u>if</u> <condition> <u>then</u> <conditional assignments and definitions> <u>endif</u>

where <condition> is an equality or inequality written according to the rules of BASIC. The conditional assignments and definitions will be valid only if <condition> is true.

For instance, the roots for an algebraic equation of second degree may be computed in the following scheme:

Disp.8.13

```
      1  SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 50   *
 51   *.................................................................
 52   *
 53   *   The roots of the equation a*z↑2+b*z+c=0 are computed as follows:
 54   *
 55   *   Let the roots be z1=x1+i*y1 and z2=x2+i*y2
 56   *   and the discriminant=b↑2-4*a*c.
 57   *
 58   *   if discriminant>=0 then
 59   *
 60   *   x1=(-b+sqr(discriminant))/(2*a),  y1=0
 61   *   x2=(-b-sqr(discriminant))/(2*a),  y2=0   endif
 62   *
 63   *   if discriminant<0  then
 64   *
 65   *   x1=-b/(2*a),   y1=sqr(-discriminant)/(2*a)
 66   *   x2=x1,         y2=-y1                    endif
 67   *
 68   *   Let a=1, b=-9 and c=14.
 69   *   Then
 70   *       x1:=7                y1:=0
 71   *       x2:=2                y2:=0
 72   *       discriminant:=25
 73   *.................................................................
 74   *
```

In this scheme x1 on line 70 has been activated and since := implies multiple activation, also values for y1, x2, y2 and discriminant are obtained as results on lines 70-71. In this case discriminant>0 and the formulas on lines 60,61 for real roots were employed.

If the line 68 is now edited into the form

```
 68   *   Let a=1, b=-4 and c=13.
```

the roots will be complex and reactivation of x1 on line 70 leads to

Disp.8.14

```
 67   *
 68   *   Let a=1, b=-4 and c=13.
 69   *   Then
 70   *       x1:=2                y1:=3
 71   *       x2:=2                y2:=-3
 72   *       discriminant:=-36
 73   *.................................................................
```

8.5. Functions

As shown in the previous examples, many standard functions of one or several variables may be used in arithmetic expressions. In addition to functions in BASIC various statistical functions for computing characteristics of standard distributions are readily available. Moreover the user may devise and program his own functions and save them on disk to be employed in the arithmetic expressions (see 8.8).

For instance, the computation scheme presented in display 8.9 can be simplified by using the function N.F(m,s↑2,x) which directly computes the normal distribution function with mean=m and variance=s↑2.
Thus, for instance

Disp.8.13

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1   *
  2   *
  3   *   1-N.F(0,1,3)=0.0013499673
  4   *
  5   *
  6   *
```

Similarly, probabilities for the binomial distribution Bin(n,p) may be computed either in the form

```
C(n,x)*p↑x*(1-p)↑(n-x),
```

where C(n,x) is the binomial coefficient (function) or more simply by the function

```
Bin.f(n,p,x).
```

The general structure of a function call in EDITOR is

```
<function designator>(<argument 1>,<argument 2>,...)
```

where <function designator> is a string which cannot be interpreted as a number and arguments are numeric expressions. The arguments may thus depend on functions. For example,

```
C(C(4,2),sqr(4)+1)=C(6,3)=20.
```

8.6. Standard functions

The standard functions are summarized in the following table.

Function	Meaning	Examples
INT(x) or int(x)	greatest integer value	int(5.35)=5 int(-5.1)=-6
ABS(x) or abs(x)	absolute value	abs(-3.3)=3.3
SGN(x) or sgn(x)	1, if x>0 0, if x=0 -1, if x<0	sgn(-3.3)=-1
IND(x) or ind(x)	1, if x≥0 0, if x<0	
MOD(x,y) or mod(x,y)	remainder of x/y	mod(10,3)=1 mod(12.5,4)=0.5

ROUND(x,n) or round(x,n)	rounding to nth decimal place	round(0.123,2)=0.12 round(-1.5,0)=-2 round(1234,-2)=1200
RND(x) or rnd(x)	random number on (0,1)	rnd(1)=0.5439562111
SQR(x) or sqr(x)	square root	sqr(3↑2+4↑2)=5
MAX(x,y,...,z) MIN(x,y,...,z)	maximum value minimum value	max(2,5,1)=5 min(2,5,1)=1
LGT(x)	common logarithm (10)	lgt(1000)=3
LOG(x) or LN(x)	natural logarithm	ln(2)=0.6931471805
EXP(x)	exponential function	exp(ln(2))=2
SIN(x)	sine	pi=3.14159265359 sin(pi/4)*sqr(2)=1
COS(x)	cosine	cos(pi/6)=0.8660254038
TAN(x)	tangent	tan(pi/4)=1
ARCSIN(X) ARCCOS(X) ARCTAN(X)	arcsine arccosine arctangent	arcsin(.01)=0.0100001667 arccos(.5)/pi=0.3333333333 arctan(.999)/pi=0.2498407655

8.7. Combinatorial and statistical functions (Status in January 1981)

Function	Meaning	Examples
FACT(n) or fact(n)	n! (n factorial)	fact(5)=120
LFACT(n) or lfact(n)	log(n!)	lfact(1000)=5912.128178496
C(n,m)	binomial coefficient	C(5,2)=10 C(50,10)=10272278170

Binomial distribution Bin(n,p):

Bin.f(n,p,x)	probability	Bin.f(4,1/2,0)=0.0625 Bin.f(4,1/2,1)=0.25
Bin.F(n,p,x)	distribution function	Bin.F(4,1/2,1)=0.3125
Bin.G(n,p,y)	inverse distribution function	Bin.G(4,1/2,0.3125)=1 Bin.G(20,0.1,0.99)=6

Poisson distribution Poisson(a):

Poisson.f(a,x)	probability	Poisson.f(10,5)=0.0378332748
Poisson.F(a,x)	distribution function	Poisson.F(10,5)=0.0670859629
Poisson.G(a,y)	inverse distribution function	Poisson.G(10,0.067)=5

Normal distribution N(m,s↑2):

N.f(m,s↑2,x)	density function	N.f(0,1,0)=0.3989422804
N.F(m,s↑2,x)	distribution function	N.F(0,1,2)=0.9772499379
N.G(m,s↑2,y)	inverse distribution function	N.G(0,1,0.995)=2.576236081

t distribution with n degrees of freedom:

t.F(n,x)	distribution function	t.F(10,2)=0.9633059826

8.8. User-defined functions

Any user can enlarge the set of functions available in EDITOR. These new functions have to be programmed in BASIC according to certain conventions and saved either on the system disk or on the user's data disk.

For example, to define the function Fisher(r)=1/2*log((1+r)/(1-r)) the following BASIC statements saved under the name "Fisher" on the system disk are sufficient:

```
3000 REM "Fisher": GOTO 2800
3010 Y=0.5*LOG((1+P(1))/(1-P(1))): GOTO 2900
```

This new function can thereafter be used like any standard function in EDITOR as follows

Disp. 8.14

```
       1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 27    *
 28    *.........................................................................
 29    *
 30    *   The sample correlation coefficient is r=0.85 and
 31    *   the sample size n=25.
 32    *
 33    *   To test the hypothesis that in the population rho=0.6
 34    *   we form the test statistic
 35    *
 36    *   U=sqr(n-3)*(Fisher(r)-Fisher(rho)),
 37    *
 38    *   which is approximately N(0,1), if our hypothesis is true.
 39    *
 40    *   In this case U:=2.640730486 and N.F(0,1,U):=0.9958635844
 41    *
 42    *.........................................................................
 43    *
```

Here U on line 40 has been activated and since := causes a multiple activation also the probability N.F(0,1,U) has been evaluated.

If the function "Fisher" were saved on the data disk, it could be called by name "#Fisher" and the line 36 should read

```
 36    *   U=sqr(n-3)*(#Fisher(r)-#Fisher(rho))
```

Several related functions may be saved in one program file by using a file name ending with "." (full stop). For instance, the functions Bin.f, Bin.F and Bin.G related to the binomial distribution are saved in file "Bin.". The part of the function designator after "." is used for the final identification. Thus the file "Bin." has the following structure:

```
3000 REM "Bin. 30.11.1980/SM":GOTO 2800
3010 IF F1$="f" THEN 3020: IF F1$="F" THEN 3030: IF F1$="G" THEN 3060
3020 REM COMPUTE Bin.f(P(1),P(2),P(3))
     --------
     : GOTO 2900
3030 REM COMPUTE Bin.F(P(1),P(2),P(3))
     --------
     : GOTO 2900
3060 REM COMPUTE Bin.G(P(1),P(2),P(3))
     --------
     : GOTO 2900
```

Observe that F1$ is the variable carrying the final identification.

The general rules in writing user-defined function routines for EDITOR are:

1) The BASIC statements are on lines 3000-4999.

2) Line 3000 has the form
   3000 REM "file name (function designator)": GOTO 2800.

3) Actual computation starts on line 3010.

4) The number of arguments in the function call is P and the values of the arguments are P(1),P(2),...,P(P).

5) BASIC variables starting with letter F (as F,F5,F3$,F(7) etc.) can be used as auxiliary variables.

6) Exit from the routine takes place with GOTO 2900.

7) Exit in error condition (mathematical errors etc.) is GOTO 2910 leading to an error message.

8.9. Tables of functions

As an alternative to the FUNC operation, tables of functions may also be formed in EDITOR by a COMP operation in the following way. Assume that we like to have the probabilities of the binomial distribution Bin(20,0.3) i.e. the values of the function Bin.f(20,0.3,x) for x=0,1,2,...,20. To begin with a column of x values is created by a COUNT operation:

Disp.8.17

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 75    *
 76    *
 77    *COUNT 80,100,78,0_
 78    *  12
 79    *
 80    *   0
 81    *   1
 82    *   2
 83    *   3
 ---   -  ---
 98    *  18
 99    *  19
100    *  20
101    *
```

Then a COMP operation using two auxiliary lines (here lines 78,79) is entered on line 77:

Disp.8.18

```
        1  SURVO 76 EDITOR   (C)1979 S.Mustonen   (124x 80)
 75    *
 76    *
 77    *COMP 80,100,79,78_
 78    *  XX        0.12345678
 79    *   x       Bin.f(20,0.3,x)
 80    *   0
 81    *   1
 82    *   2
 83    *   3
 --    -  --
 98    *  18
 99    *  19
100    *  20
101    *
```

The third parameter (here line 79) defines the arguments and the function and the fourth parameter (here line 78) the columns of the arguments (indicated with X's) and the place and format for the function as a numerical image.
Thus activation of COMP on line 77 leads to the result:

Disp.8.19

```
        1  SURVO 76 EDITOR   (C)1979 S.Mustonen   (124x 80)
 75    *
 76    *
 77    *COMP 80,100,79,78
 78    *  XX        0.12345678
 79    *   x       Bin.f(20,0.3,x)
 80    *   0        0.00079792
 81    *   1        0.00683934
 82    *   2        0.02784587
 83    *   3        0.07160367
 84    *   4        0.13042097
 85    *   5        0.17886305
 86    *   6        0.19163898
 87    *   7        0.16426199
 --    -  --        ----------
 96    *  16        0.00000501
 97    *  17        0.00000050
 98    *  18        0.00000004
 99    *  19        0.00000000
100    *  20        0.00000000
101    *
```

The cumulative probabilities could either be tabulated as the values of the function Bin.F(20,0.3,x) or after the preceding results more quickly by employing the CUMUL operation:

Disp.8.20

```
       1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
 75    *
 76    *
 77    *COMP 80,100,79,78
 78    *  XX          0.12345678
 79    *   x        Bin.f(20,0.3,x)
 80    *   0          0.00079792          0.00079792
 81    *   1          0.00683934          0.00763726
 82    *   2          0.02784587          0.03548313
 83    *   3          0.07160367          0.10708680
 84    *   4          0.13042097          0.23750777
 85    *   5          0.17886305          0.41637082
 86    *   6          0.19163898          0.60800980
 87    *   7          0.16426199          0.77227179
 88    *   8          0.11439674          0.88666853
 89    *   9          0.06536957          0.95203810
 90    *  10          0.03081708          0.98285518
 91    *  11          0.01200665          0.99486183
 92    *  12          0.00385928          0.99872111
 93    *  13          0.00101783          0.99973894
 94    *  14          0.00021811          0.99995705
 95    *  15          0.00003739          0.99999444
 96    *  16          0.00000501          0.99999945
 97    *  17          0.00000050          0.99999995
 98    *  18          0.00000004          0.99999999
 99    *  19          0.00000000          0.99999999
100    *  20          0.00000000          0.99999999
101    *             XXXXXXXXXX           0.12345678
102    *CUMUL 80,100,101_
103    *
104    *
105    *
```

Also samples from various distributions may be generated with the COMP operation by using the fact that when G(y) is the inverse distribution function of distribution D and y is uniformly distributed on (0,1) then G(y) is a random variate from the distribution D.

For instance, to form a sample of 100 observations from a standard normal distribution we make a table of the function N.G(0,1,rnd(1)) by a COMP operation as follows:

Disp.8.21

```
       1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *
  3    *COMP 7,106,5,4_
  4    *  -1.12345
  5    *  N.G(0,1,rnd(1))
  6    *
  7    *  -0.35449
  8    *  -0.49315
  9    *   0.48293
 10    *  -0.55747
 11    *   1.94759
 12    *   1.21945
 --    -  -------
```

Here the values of the function on line 5 have been placed on lines 7-106 by using the format given on line 4.

Similarly, if we like to generate 12 values of variable Y according to the model Y=a0+a1*X1+a2*X2+eps where X1 and X2 are given and eps is an error term distributed according to N(0,s↑2), the COMP operation in the following computation scheme will do that for us:

Disp.8.22

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *      Generating data according to a given regression model:
  3    *
  4    *      Y=a0+a1*X1+a2*X2+eps,
  5    *      where eps=N.G(0,s↑2,rnd(1)), s=3,
  6    *      a0=100, a1=10 and a2=1.
  7    *
  8    *COMP 11,22,10,9_
  9    *     XXXXXXXX       XXX        123.12
 10    *        X1           X2          Y
 11    * 1  -0.35449        68
 12    * 2  -0.49315       110
 13    * 3   0.48293        33
 14    * 4  -0.55747        86
 15    * 5   1.94759       136
 16    * 6   1.21945        30
 17    * 7  -0.07194        94
 18    * 8   1.54428       144
 19    * 9   0.07833       150
 20    *10  -1.06143        89
 21    *11  -0.84473       130
 22    *12  -0.52181        84
 23    *
```

When COMP on line 8 is activated the Y values will be computed for the X1 and X2 values on lines 11-22. The auxiliary lines 9 and 10 indicate that Y is the function to be computed. EDITOR finds the definition of Y on line 4 and further explanations (definition of eps and values of s,a0,a1 and a2) on lines 5 and 6. On the basis of this information the Y column is created and displayed according to the format on image line 9:

Disp.8.23

```
        1   SURVO 76 EDITOR    (C)1979 S.Mustonen    (124x 80)
  1    *
  2    *      Generating data according to a given regression model:
  3    *
  4    *      Y=a0+a1*X1+a2*X2+eps,
  5    *      where eps=N.G(0,s↑2,rnd(1)), s=3,
  6    *      a0=100, a1=10 and a2=1.
  7    *
  8    *COMP 11,22,10,9_
  9    *     XXXXXXXX       XXX        123.12
 10    *        X1           X2          Y
 11    * 1  -0.35449        68        165.65
 12    * 2  -0.49315       110        195.23
 13    * 3   0.48293        33        137.54
 14    * 4  -0.55747        86        179.11
 15    * 5   1.94759       136        252.33
 16    * 6   1.21945        30        141.12
 17    * 7  -0.07194        94        200.35
 18    * 8   1.54428       144        263.96
 19    * 9   0.07833       150        252.53
 20    *10  -1.06143        89        172.77
 21    *11  -0.84473       130        218.60
 22    *12  -0.52181        84        177.53
 23    *
```

To "check" this result we can easily perform a REGRAN operation (after inserting a column C of 1's as a constant term):

Disp.8.24

```
         1   SURVO 76 EDITOR    (C)1979 S.Mustonen     (124x 80)
  1    *
  2    *      Generating data according to a given regression model:
  3    *
  4    *      Y=a0+a1*X1+a2*X2+eps,
  5    *      where eps=N.G(0,s↑2,rnd(1)), s=3,
  6    *      a0=100, a1=10 and a2=1.
  7    *
  8    *COMP 11,22,10,9
  9    *     XXXXXXXX       XXX        123.12
 10    *        X1          X2           Y        C
 11    * 1  -0.35449        68        165.65      1        2.76
 12    * 2  -0.49315       110        195.23      1       -8.09
 13    * 3   0.48293        33        137.54      1        0.13
 14    * 4  -0.55747        86        179.11      1        0.52
 15    * 5   1.94759       136        252.33      1       -4.77
 16    * 6   1.21945        30        141.12      1       -1.66
 17    * 7  -0.07194        94        200.35      1        8.23
 18    * 8   1.54428       144        263.96      1        3.44
 19    * 9   0.07833       150        252.53      1        2.69
 20    *10  -1.06143        89        172.77      1       -3.08
 21    *11  -0.84473       130        218.60      1       -0.72
 22    *12  -0.52181        84        177.53      1        0.54
 23    *     XXXXXXXX       XXX       YYYYYY      X      -12.12
 24    *DATA TEST,11,22,10
 25    *REGRAN TEST,23,26_
 26    *REGRESSION ANALYSIS: REGRESSAND:'Y'  DATA:'TEST'
 27    *    TOTAL VARIANCE=     1821.113 DF= 11
 28    *    RESIDUAL VARIANCE=     21.808 DF=  9  R↑2=0.9902
 29    * VARIABLE         REGR.COEFF.  STD.DEVIATION      T
 30    * X1                11.378313       1.44394      7.880
 31    * X2                 1.000256       0.03536     28.281
 32    * C                 98.905259       3.64851     27.108
 33    *
```

9. Special applications

SURVO 76 EDITOR can be used for data handling in various special applications. In addition to normal text we can use extra symbols in the edit files for output control. Special programs capable of interpreting the control code and of plotting or printing the text are needed.

So far two plotting programs using EDITOR as their host have been implemented. The first of them is a general text plotting program and the second one is a music plotting system. These programs have been made in collaboration with Elina and Olli Mustonen.

9.1. Text plotting program TEXTPLOT

TEXTPLOT is an extension of SURVO 76 EDITOR for plotting text in various formats via the graphic CRT or the plotter.

TEXTPLOT is loaded as a SURVO 76 module T from disk SURVO 76/S3B and it offers two alternative methods of use:

F1: PLOTTING SINGLE CHARACTERS (TYPEWRITER MODE)
F2: PLOTTING SURVO 76 EDIT FILES

In F1 the characters typed will immediately be plotted, but in F2 before plotting a text file must be created using SURVO 76 EDITOR. In most cases F2 is more suitable than F1, because the text to be plotted can easily be edited and some control commands are available only in F2.

The characters to be plotted are based on a standard letter set (see APPENDIX 1), but the size and the form of the characters may be modified continuously.

The form and the shape are controlled by three parameters:

Parameter 1 specifies the size. Default is 1 and corresponds to the height of 8 mm for the letter A on the graphic CRT (hardcopy) and to the height of 2.3 mm for the letter A on the plotter.

Parameter 2 is an elongating factor which changes the height of each character with respect of the width (size) of the characters. Default is 1.

Parameter 3 is the angle of slant in degrees. Default is 0 and also negative values can be employed.

For all the parameters also non-integer values are permitted. The size and shape parameters are entered after the F1 and F2 starts, but their values can be changed in F1 by typing # and in F2 by inserting commands of the type #(2,0.5,15) within the text to be plotted.

Also the starting point on the graphic screen or on the paper must be given with respect of the present home of the pen. This takes place by entering the coordinates X,Y of the starting point. X and Y have to be given in plotting units. Thus on the screen 0≤X≤800, 0≤Y≤512 and default is X=0,Y=450. On the plotter the plotting unit is 0.1 mm.

In F2 also the name of the text file and the first line to be plotted have to be entered. Before selecting the initial line the file can be scanned on the screen by pressing RETURN(EXEC).

In F2 the following control codes are available:

#(p1,p2,p3)	redefines the size and the shape parameters.
P(X)	selects pen X (X=1,2 or 3).
P(X,Y)	specifies new coordinates X,Y for the next character.
D(X,Y)	specifies new coordinates for the next character relative to the present position of the (invisible) cursor.
C	clears the screen.
R	restarts the plotting from the initial line.
W	causes a pause of ca.3 seconds in plotting. (Codes C,R and W can be used especially in making live displays on the graphic CRT in non-stop style.)
@	terminates plotting.
S(X,Y)	defines natural horizontal spacing between adjacent characters. Here 'natural' means that the horizontal gap between consecutive characters is X and the vertical spacing is Y. Narrow characters will take less space than broad ones in the X direction. Default is X=3,Y=0. The true spacing is always proportional to the size parameter.
T(X,Y)	defines strict spacing between adjacent characters. 'strict' means that the characters will be plotted with constant spacings X,Y. The true spacing is also proportional to the size parameter.

9.2. Music plotting

The music plotting program based on SURVO 76 EDITOR produces a copy of a music manuscript in a form correspoding to high standard music printing. The score to be plotted has to be transformed into a special code planned for this application and saved in an edit file. The code may be checked rapidly with the graphic CRT, but the final output is obtained using the plotter.

Since this application of EDITOR has no relation to statistical data processing, all the details are omitted here. However, to give an idea of how EDITOR may be used for special purposes we have two displays in APPENDIX 2 representing a portion of an edit file containing a musical composition and the final output of the same file.

REFERENCES

Afifi,A.A., Azen,S.P. (1979), Statistical Analysis, a Computer Oriented Approach, (2nd edition), Academic Press.

Bishop,Yvonne M.M., Fienberg,S.E., and Holland,P.W. (1975), Discrete Multivariate Analysis, Cambridge, Massachusetts: MIT Press.

Butler,D.E., and Stokes,D. (1974), Political Change in Britain, (2nd edition), London: Macmillan.

Hald,A. (1952), Statistical Theory with Engineering Applications, Wiley.

Mustonen,S. (1977), SURVO 76, a statistical data processing system, Research Report No.6, Dept.of Statistics, University of Helsinki

Mustonen,S. (1980), On interactive statistical data processing, 8th Nordic Conference on Mathematical Statistics.

Mustonen,S. (1980), Interactive analysis in SURVO 76, Proceedings in in computational statistics, ed. by M.M.Barritt and D.Wishart, 253-259, Physica-Verlag, Wien.

Mustonen,S. (1980), SURVO 76 EDITOR, a new tool for interactive statistical computing, text and data management, Research Report No. 19, Dept.of Statistics, University of Helsinki.

Payne,C. (1977), The log-linear model for contingency tables, in The Analysis of Survey Data, ed. by C.A.O'Muircheartaigh and C.Payne, Volume 2, pp.105-144, Wiley.

The standard letter set #(1,1,0) plotted on the graphic CRT:

ABCDEFGHIJKLMNOP
QRSTUVWXYZ
abcdefghijklmnop
qrstuvwxyzüäö
1234567890+-=()?%$/.,:

Example:

a) Text saved in an edit file:

```
      1   SURVO 76 EDITOR     (C)1979 S.Mustonen     (100x100)
41    *
42    *#(2,1,0)Textplot #(1,1.2,15)
43    *a text plotting program
44    *based on
45    *#(1.5,.5,0)SURVO 76 EDITOR.@
46    *
```

b) The same text plotted by TEXTPLOT:

Textplot

a text plotting program

based on

SURVO 76 EDITOR.

MUSIC PLOTTING

a) Composition coded and saved in an edit file:

```
   1   SURVO 76 EDITOR    (C)1979 S.Mustonen    ( 70x 75)
*SAVE QUAD
*
*
*"#(4/1/0)%(800/400)Quatrille"
*"#(2/1/0)%(0/200)Moderato con anima"
*"#(2/1/0)%(1670/200)Olli Mustonen (1979)"
*$G,T(3/4),V90,Q.4,L.75,F/4.,
*YC0,X"#(1.8/1.2/15)p",-Hb1/8,Q.9,XC1,<Eb2,H1>,XC1.*
*+F/4.,Q0.8,YA3,X(1/4),
*-<Hb1/16,XC1,YA3,X"#(1.5/1/0)     = ca.108",Eb2>,<Ab,Db3,A2,E>,XC1.*Q.4,
*V80,-<Hb1/8,XC1,C2>,XC1.,+<Ht1,XC1,A>,XC1.,<Hb,Ab>*
*G/4.,YG0,X+1,-C#2/8,XC2,<Ct,F#>,X+1.,XC2.,R*
*-<A2/8,XC2,YG0,X-1,G#2>,XC2.,
*<F,Eb/16,XC2,Db>,XC2.,+<H1,XC2,A,XC2.,G,F>,X-1.*
*V80,<E/8,XC1,F#>,XC1.,<Ft,XC1,Eb>,XC1.,<Et,D>*
*<C#/8,Z.,G>,Z.,<F#,Z.,Ct>,Z.,<H0,Z.,Ft1>,Z.*
*V100,K.2,<E1/8:A0,C#2>,YC0,X"#(1.8/1.2/15)mp",
*K9,-<Eb2/16,D,F#/8>,<Ft/16,XC1,Eb,XC1.,Ab1,Hb>,R*
*V80,+<F#/8,F/16,XC1,G#>,XC1.,
*<Ft/8,F/16,XC1,Gt>,XC1.,<E/8,D#/16,YC,XC1,C#>,YH0,XC1.*
*V90,+<Ct1/8:G,E2>,YC0,X"mf",
*K9,L.75,-<F#2/16,Ft,A/8>,<G#/16,XC1,H,XC1.,Gt,F>*
*-<E2,YF,XC1,G#,YA,XC1.,Eb,D>,<C#,D#,C,H1>,V80,+<Ct2,H1,F#,G#>,R*
```

b) The same composition plotted on the drum plotter (2272):

EDITING OPERATIONS (status in February 1981)

Each operation has to be typed in the beginning of an empty line and it will be activated by moving the cursor to the line of the operation to be used and by pressing the key CONTINUE once.

When the operation is active, RUN is displayed in the left upper corner of the CRT.

Notations:

L1,L2= line numbers of the first and last line to be operated,
K = image line number
x,y,z= arbitrary characters

1. CONTROL AND BASIC OPERATIONS

REDIM <# of lines>,<# of columns>
redimensions the edit field without changing its contents.

SCRATCH
clears the edit field from the SCRATCH line onwards.

CLEAR L1,L2
clears the lines L1-L2 to the right from the current position of the cursor.

CLEAR L1,L2,K
clears the lines L1-L2 for those columns indicated in the image line K by non-space characters.

CLEAR xy
clears a rectangular field indicated by an x in the left upper corner and y in the right lower corner.

ERASE <string>
erases all characters occuring in the string from the edit field.

DELETE L1,L2
deletes the column indicated by the cursor from the lines L1-L2.

INSERT L1,L2
inserts one empty column to the current cursor position for the lines L1-L2.

COPY L1,L2,L
copies lines L1-L2 to a new place starting from line L. The former contents of lines L,L+1,...,L+L2-L1+1 will be destroyed.

MOVE L1,L2,K,L3
```
K="       X     X    Y     "
```
copies a subfield consisting of columns indicated by X's and lines L1-L2 to a position starting from line L3 and column indicated by Y.

MOVE xyz
moves a rectangular field indicated by x in the left upper corner and y in the right lower corner to a place indicated by z so that x will be in the position of z.

JOIN L1,L2,N,L3
moves the lines L1-L2 from the edit file N to the present edit field so that the first line will be L3 in the edit field.

JOIN <name of edit file>,A1,A2,A3,...,L3
moves the subfields (chapters) A1,A2,A3,... defined by DEF operations to the present edit field so that the first line of the first chapter will be L3.

DEF <name of a subfield (chapter)>,L1,L2
defines the lines L1-L2 as a chapter to be called by its name. (DEF is a specification and needs no activation.)

FIND <string>
    finds the first occurence of the string in the edit field. The cursor will then point the first character of the string. By pressing CONTINUE immediately again the next occurence will be found etc.

FIND L1,L2,K
    finds the occurences of the string on the image line K from lines L1-L2.

REPLACE L1,L2,K1,K2
    looks for the string described on line K1 from lines L1-L2. If CONTINUE is pressed thereafter the K1 string will be replaced by the string on line K2 and the next occurence of K1 will be searched for. If CLEAR is pressed K1 is not replaced, but the next K1 will be searched for.

SET L1,L2,K
    inserts the non-space characters in K to the lines L1-L2.

2. SAVING AND LOADING

SAVE L1,L2,L3,<name of a SURVO 76 data file>
    moves a data matrix on lines L1-L2 to a SURVO 76 data file. The names of the variables are listed on the line L3. An obs.name must be at the beginning of each observation line. This operation can be used for input of new data as well as for enlarging and editing of a numeric SURVO 76 data file. Eventual new variables and observations will be automatically defined in the SURVO 76 data file.

SAVE <name of edit file>
    saves the current edit field in an edit field on the R disk.

LOAD <name of edit file>
    loads an edit file saved by SAVE to the edit field.

LOAD
    loads various SURVO 76 files to the edit field.

3. PRINTING

PRINT L1,L2
    prints the lines L1-L2 on the printer.

PRINT L1,L2,<separator character>
    works as PRINT above, but starts a new line whenever the separator character is encountered.

PRINT R1,R2,<1st page>,<left edge>,<page length>,<extra lines>
```
      <file 1>,<chapter 1>
      <file 2>,<chapter 2>
      ------   -------
      END
```
    prints a document consisting of several chapters in several edit files so that the text will be automatically divided into pages and the pages will be started by one or several header lines with appropriate page numbers.
    When the PRINT operation is activated, none of chapters to be printed needs to be in the current edit field.
    The header lines for the pages are to be described on edit lines R1-R2 and the line R1 must contain the image of the page number in the form ###.

4. FORMATTING

TRIM L1,L2,C

makes up the lines L1-L2 using the line width C without breaking words, numbers and other continuous character strings.
If new lines are needed, they will be automatically inserted and, on the other hand, unnecessary lines will be deleted.
After TRIM only one space will be left between the 'words' and the right edge will not necessarily be even. To make it straight use TRIM2 immediately after TRIM.
The lines after an empty line or starting with at least one space will be preserved as new lines.

TRIM2 L1,L2,C

makes the right edge of the lines L1-L2 straight in the column position C without breaking or connecting the lines by inserting spaces evenly between the 'words'.
If the line is short for more than 8 characters nothing will be changed.

TRIM3 L1,L2,C

operates as TRIM, but breaks the words when needed according to the rules of Finnish. TRIM3 operates rather satisfactorily also in other languages.

FORM L1,L2,K

```
K="      AAAA   -123.12   AAAA  -12.123   etc.   "
```

redisplays a table on lines L1-L2 and consisting of alpha and numeric columns in the format indicated by the image line K.

5. SORTING

SORT L1,L2,K

```
K="     XXXX    "
```

sorts the lines L1-L2 in alphabetic order according to the sort key defined by XXXX's on the image line K.
If the sort key is of the form 1111, sorting will be numeric in ascending order.

-SORT L1,L2,K

sorts in descending order in case of numeric sorting.

PERM L1,L2,L

permutes the fields (columns) on lines L1-L2 according to the fields on the lines L,L+1,L+2,...
The numbers of fields on line L+i and L1+i must be the same and the fields on line L1+i will be permuted in a new order corresponding to the alphabetic order of the L+i field.
If some of the lines L,L+1,... is empty the remaining lines will be permuted according to the last non-empty line. For example, when changing the order of columns in a table the line L should point out the new order and the line L+1 should be empty.

6. NUMERIC OPERATIONS FOR TABLES

COUNT L1,L2,K,C

inserts numbers C,C+1,C+2,... on lines L1-L2 in the format indicated by an image line K of the form " -123.1 ".

COUNT L1,L2,K,C,D

works similarly, but inserts the numbers C,C+D,C+2D,...

COUNT L1,L2,K,C,D,E

with E>0 inserts numbers
C,C+D,C+2D,...,C+(E-1)D,C,C+D,C+2D,...,C+(E-1)D,C,C+D,C+2D,...
on lines L1-L2 in the format indicated by image line K thus generating a cyclical count.
If E<0, the numbers inserted will be
C,C,...,C,C+D,C+D,...,C+D,C+2D,C+2D,...,C+2D,....
where each of values C,C+D,C+2D is repeated -E times.

```
CUMUL L1,L2,K
      where K="   XXXXXXX     -1.12345   "
      computes a new column as a cumulative sum of the values in the
      XXXXXXX column.

LAG L1,L2,K,C
      K="   XXXX   XXXX  XXX   "
      moves the columns indicated by X's on the image line
      C steps downwards. The C first lines on the corresponding
      columns will be cleared.
DIFF L1,L2,K,C
      K="   XXXX   -123.12  "
      computes on lines L1-L2 the C step differences of X-column
      according to the image -123.12. Missing C implies C=1.
%DIFF L1,L2,K,C
      works as DIFF, but computes the differences in percentages.

C+ L1,L2,K
      K="   XXXX  XXXX   XXXX   -123.12   "
      computes the sum of XXXX-columns to the column indicated
      by -123.12 for the lines L1-L2.
C* L1,L2,K
      works as C+, but the product of XXXX-columns is computed.
C- L1,L2,K
      K="  XXXX  YYYY   -123.12   "
      subtracts the YYYY-column from the XXXX-column.
C/ L1,L2,K
      divides the XXXX-field by the YYYY-field.
L+ L1,L2,K,L3
      K="    -123.12   -123.12    -123.12        "
      computes the sum of lines L1-L2 for the columns indicated by
      the image line and inserts the sums on the line L3 according
      to the images on the image line.
L* L1,L2,K,L3
      works as L+, but the product is computed.
L- L1,L2,K,L3
      computes the difference of lines L1 and L2 to line L3.
L/ L1,L2,K,L3
      computes the ratio of lines L1 and L2 to line L3.

FUNC L1,L2,K
      K="   XXXX XXXX XXXX  -123.12    "
      computes a numeric function of XXXX-columns as a new
      column indicated by -123.12 on the image line for the
      lines L1-L2. The function must be defined on program
      lines 100-119 as a subroutine in the form
      100 Y=function(X(1),X(2),...,X(M)):RETURN
      where M is the number of XXXX-columns.
      If M=1 (function of single variable), the subroutine
      may have the simple form  100 Y=function(X):RETURN
      If the EDITOR program is protected (↑ERR A06), the general
      form of FUNC described below or COMP must be employed.
FUNC N,L1,L2,K
      K="   XXXX XXXX XXXX  -123.12    "
      works as FUNC above, but employs a function subroutine N
      which has to be saved on the data disk under name N in the
      form:
        99 GOTO 1415
       100 Y=function(X(1),X(2),...,X(M)):RETURN
```

COMP L1,L2,F,K

where F and K are auxiliary lines and, for example,

```
F="      X          Y          sqr(X↑2+Y↑2)      ",
K="    XXXX       XXXX          -123.12          ",
```

computes values of the function (sqr(X↑2+Y↑2)) for the values (X,Y) of the XXXX columns on lines L1-L2. The new column will be displayed according to the numeric format (-123.12) defined on the image line K.
The function may be also defined by using various variables and constants specified outside the line F.

7. STATISTICAL OPERATIONS

DATA <name of data set>,L1,L2,L3

defines a data set on lines L1-L2. Each line includes one observation vector in the form
<name of observation> X(1) X(2) ... X(M).
Line L3 contains the names of the variables X(1),...,X(M).
(DATA is a specification and needs no activation.)

REGRAN <name of data set>,K,L

computes the parameters of a linear regression model from a data set defined by a DATA line. The image line K specifies the columns of the dependent variable by Y's and the columns of the regressors by X's (also a constant term has to be defined in this way). The results will appear from line L onwards.
If the image K includes a field of the form -123.12, the residuals of the model will be inserted to this column.

CORR <name of data set>,K,L

computes means, standard deviations and correlations from a data set defined by a DATA line and for the variables specified by the XXX-fields of the image line K. The results will appear from the line L onwards.

MULTIWAY TABLES:

TABLE <name of table>,L1,L2,<type>

defines a multidimensional frequency table (type F) or a multidimensional data table (type X). The table should be described from the line L onwards in a following form:

```
37   *TABLE TEST,38,46,X
38   *        A    A1        A2        A3        A4
39   *        B    B1   B2   B1   B2   B1   B2   B1   B2
40   *C   D   **
41   *C1  D1       1111 1211 2111 2211 3111 3211 4111 4211
42   *    D2       1112 1212 2112 2212 3112 3212 4112 4212
43   *    D3       1113 1213 2113 2213 3113 3213 4113 4213
44   *C2  D1       1121 1221 2121 2221 3121 3221 4121 4221
45   *    D2       1122 1222 2122 2222 3122 3222 4122 4222
46   *    D3       1123 1223 2123 2223 3123 3223 4123 4223
```

The variables (factors) A and B are here column classifiers and the variables C and D are row classifiers.
In this case ** on the line 40 points out the column classifiers (above) and the row classifiers (to the left). Number of *'s indicates the maximum length of the class names.
The class names can be freely selected and they can be typed hierarchically as above, or completely. The structure of the table must be hierarchical, however.

The following TAB-operations are available for table modifications and they put the modified table from line L onwards and call it with the name <old name><4th letter of operation word>.

TABC <table>,L
copies a table.

TABS <table>,S,L
displays the table with S row classifers (S=0,1,2,...,N).

TABM <table>,A,B,L
changes the positions of the classifiers A and B.

TABI <table>,C,C1,C2,L
changes the positions of classes C1 and C2 for C-classifier.

TABD <table>,C,L
deletes classifier C.
In a <u>frequency table</u> the elements of the modified table are the marginal sums of C-classification; In this case TABD means collapsing over classifier C and the result is an (n-1)-dimensional table.
In a <u>data table</u> the C-classification will be changed to a N-classification where the C-class names are replaced by indices 1,2,... The modified table has still n-dimensional and has the old structure and contents; only the name C is replaced by N.
If TABD is applied to a data table which already includes an N-classification, the C-classification will be united to the N-classification and now the dimension of the table will be decreased by 1 but the number of elements remains the same.

TABJ <F-table>,C,C1,C2,Cn,L
combines (in a frequency table) the classes C1 and C2 in the C-classification and calls the united class by Cn. The class Cn takes the position of C1 and C2-class will be cancelled.

TAB+ <table 1>,<table 2>,L
forms the sum of tables 1 and 2.

TAB- <table 1>,<table 2>,L
forms the difference of tables 1 and 2.

ANOVA <X-table>,L
computes from a m-dimensional data table the ANOVA table according to m factor experiment and positions that table from line L onwards.

TABFIT <F-table>,K,L
estimates from a F-table a loglinear model defined on line K.
The specification line K has the structure:
LOGLIN AB,AC,D
where AB,AC and D are the configurations (marginals) to be fitted in the ML-estimation.
Structural zeros are denoted by -.

2. muuttamaton painos

Helsingin yliopiston monistuspalvelu
Painatusjaos Helsinki 1981