DECUS
PROGRAM LIBRARY

DECUS NO.	FOCAL8-176
TITLE	PROGRAM FOR PRODUCING HISTOGRAMS FROM CLINICAL DATA ON TELETYPE
AUTHOR	Eddy Emons
COMPANY	Royal Postgraduate Medical School Hammersmith Hospital London, England
DATE	May 1, 1971
SOURCE LANGUAGE	FOCAL '69

PROGRAM FOR PRODUCING HISTOGRAMS FROM CLINICAL DATA ON TELETYPE

DECUS Program Library Write-up

DECUS NO. FOCAL8-176

This program uses data from the Hypertension Clinic, which are blood pressure measurements taken from patients in the lying and upright positions respectively. Both the systolic (upper) and the diastolic (lower) pressures are recorded for each position.

FOCAL is used with all the extended functions erased. The data are recorded with the high speed reader and stored in a two dimensional array in field one via the interger overlay FNEW. For each pressure measurement, the mean and standard deviation are computed.

From the two dimensional array stored in field one another two dimensional array is computed and stored in field zero, representing the histogram data. FOCAL then scans through each array and types the histogram on the teletype.

Abstract

In this program several new features are brought together:

(a) FOCAL interpreter is normally used in field zero with all the extended functions erased. Field one is used as a linear integer array for storage of data and can take 3900 integers, using FNEW function.

(b) Two dimensional array techniques are used to store data in field one, via the function FNEW and also in field zero. (FOCAL uses single subscripts only.)

(c) There is no need to ask first for the number of data points. The data is counted as it is read in the high speed reader; a STOP must be put at the end of the data.

(d) FOCAL types a colon for each data word when the high speed reader is used for data input. With large amounts of data the line would soon overflow with colons. One solution would be to delete the colon type out by changing location 1217/4551 to 7600. However, if a mistake in the data terminators would occur (e.g. extra comma or space) one could not easily identify this. The other solution is to give a carriage return/line feed (CR/LF) after 72 colons are typed out for each line. The STOP is sensed as one extra colon.

(e) Multiple use of variables of the same character is made to avoid storage overflow.

Operating instructions

(1) Load FOCAL tape and FNEW integer array overlay tape with binary loader.

(2) Load address 200, start, for each tape respectively.

(3) The extended functions are deleted automatically when FNEW tape is loaded.

(4) Load program * , put data tape in high speed reader.

(5) Type GO

Detailed program description

Maximum number of data rows is 100, i.e. dimensions of array is 100. The number of columns do not need to be specified, only four columns are used. The ERASE at line 1.01 resets all variables to be used to zero. A CR/LF is given and P is incremented by one (line 1.10). P is for the automatic CR/LF after 72 colons/line are typed out (see line 1.50). The value of the integer S is asked for at line 1.20. It is compared with the value Ø STOP to sense the out of data condition. When negative, transfer is made to line 1.30 if zero, to line 1.60 where the reader is turned off. Before the actual computation is carried out, the linear subscript G is computed representing the two dimensional subscript (J, K), J being the row subscript, K being column subscript (line 1.30). The value of S is then stored via FNEW in field one as an integer. Four columns are used in this program. To detect if the last column is reached, comparison is made to check the value of K (from 0 to 3). When negative, K is incremented at line 1.40. Control goes back to line 1.20 where the next integer is asked for.

When zero, K is reset to zero (line 1.50) and J is incremented by one. After 18 increments of J, 72 colons will be typed out and CR/LF must follow. Control goes to line 1.10, P is incremented and CR/LF is given. If J is less than 18, control goes back to line 1.20. After all the data is read in, two CR/LF's are given (line

1.60). The value of J is stored as N. The computation goes like this:

Starting with the K FOR loop at line 1.60, the height of each histogram H(K) is initially zero and so is the highest subscript HS(K), but the lower subscript LS(K) is set to the highest pressure value possible but divided by 10 (25 + K x 25). These 3 values will be determined with the J FOR loop in groups 2, 5 and 6 respectively and for each column.

First, the value of G is computed. The data in each column is added and squared and added (line 2.10). The data itself is used to determine the subscript B in array Z(B), which adds up all the values of the same subscript (line 2.15) which are stored two dimensionally in field zero, and represents the histograms. The computed value for B rounds off pressures like 155, 135 etc. Added to B is K x 25. K is the correct column, 25 is the highest pressure possible (250) divided by 10.

The height of each histogram is determined with the IF loop at lines 2.15, 2.20.

At group 5 using the same method, the highest value of the pressure is determined HS(K), and at group 6 the lowest value of the pressure LS(K) for each column respectively.

The alternative here would have been to do group 5 and 6 separately with a FOR loop from 1 to 25, but this would call up values which might not be in the Z(B) array. Once called up they are stored in the symbol table as being zero and storage of variables would be greatly reduced. Unless stored with another FNEW (G, S), but this becomes cumbersome in keeping track of the

integer G as it is a single integer array only.

An additional advantage of this method is that the scanning of the horizontal axis is done much faster, i.e. as soon as the highest pressure is reached (B FOR loop at line 1.75) a CR/LF is typed, reducing teletype time. Return is made from the DO loop (group 7). The first results are printed for each column, the value of K being compared to select the correct title (line 3.20), and well mean pressures and standard deviations. Part of the type written section in group 3 is used again in the final result. Group 4 deals mainly with correct spacing of the results. Line 4.50 is used frequently later on in the program. After line 1.70, the computation for each histogram follows at line 1.75. The K FOR loop selects the column, types CR/LF, transfers control to group 8. Here the value of Z is set one higher than H(K). This is done because the value of Z is decremented by 1 at line 1.90 for each value in the G FOR loop. Returning to group 8 at line 8.10, K is compared for the zero value, control goes to 8.40, another CR/LF follows and return is made to line 1.75. This is done to overcome that the title is typed out before the histogram is computed and typed out. If the histogram is finished, the scale and the title are typed, because K is now set one larger than is required for the correct title (at 8.25). K is set one smaller before the rest of the title is determined at group 3. K must then be reset to the original value. Returning to line 1.75 at the G FOR loop, control goes to line 1.90. Z is typed out being the current height value for the Y axis. The correct spacing is determined for the horizontal axis before the X is typed. First, the B FOR loop (group 7) scans from the lower to higher subscript,

where the current value of Z is compared with the value in array Z(B), when negative, control goes to line 7.20. When zero or positive to line 7.30. Finally at line 1.80 the title is typed out for the last histogram and program halts.

Requirements

The program uses 515 locations leaving 565 locations for data in field 0, of which on average 80% is used.

Execution time

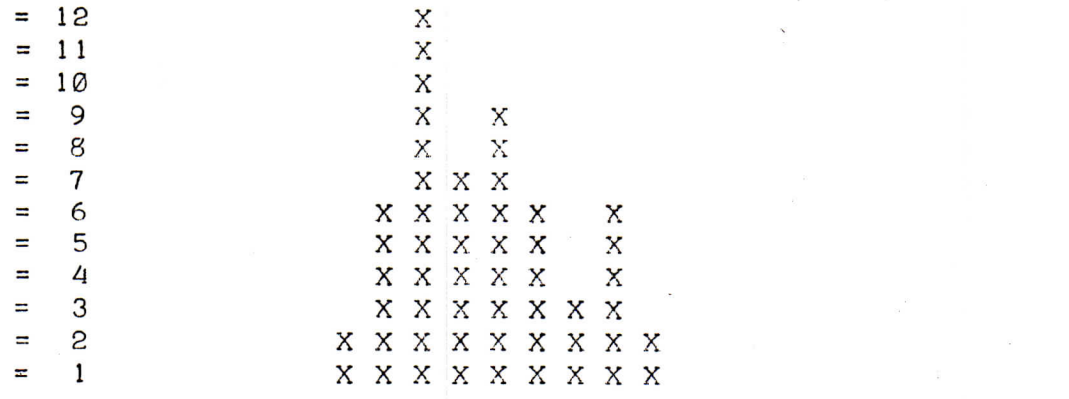
Computing time is approximately 150 msec per data word.

Reference

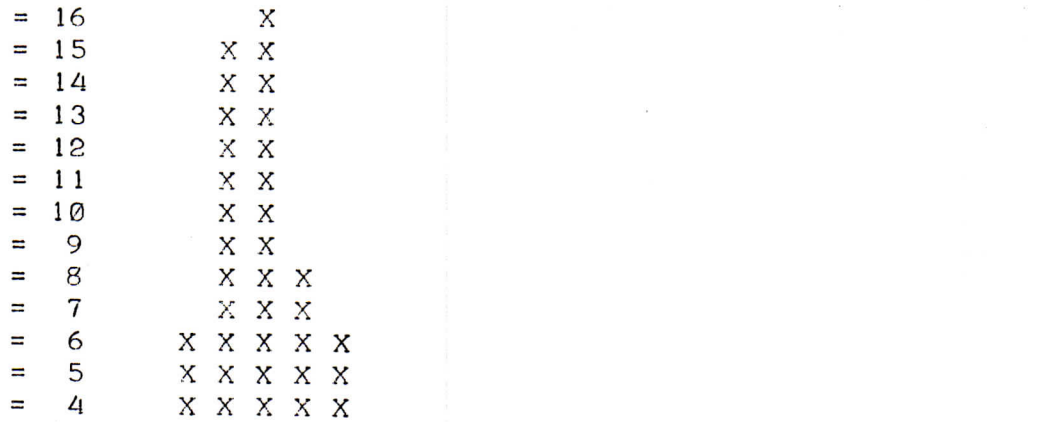
Advanced FOCAL technical specifications, DEC-08-AJBB-DL.

```
*?00.00
*E A
**
**********************************************
*W
C-FOCAL, 1969

01.01 E
01.05 S L=100;*
01.10 T !;S P=P+1
01.20 A S;I (S-0STOP)1.3,1.6
01.30 S G=J+L*K;S Z=FNEW(G,S);I (K-3)1.4,1.5
01.40 S K=K+1;G 1.2
01.50 S K=0;S J=J+1;I (J*4-P*72)1.2,1.1
01.60 *;T !!;S N=J;F K=0,3;D 1.95;F J=0,N-1;S G=J+L*K;D 2;D 5;D 6
01.65 F K=0,3;D 3
01.70 T !"NUMBER OF PATIENTS ATTENDING"%2,N,!
01.75 F K=0,3;T !;D 8;F G=1,H(K);D 1.9;F B=LS(K),HS(K);D 7
01.80 T !;D 8;Q
01.90 T !;S Z=Z-1;T Z;S SP=(LS(K)-K*25-5)*2+1;D 4.5
01.95 S LS(K)=25+K*25

02.10 S L1(K)=L1(K)+FNEW(G);S M(K)=M(K)+[FNEW(G)]↑2
02.15 S B=FITR(FNEW(G)/10+.5)+K*25;S Z(B)=Z(B)+1;I (H(K)-Z(B))2.2;R
02.20 S H(K)=Z(B)

03.10 S MB=L1(K)/N;S SD=FSQT[(M(K)-L1(K)↑2/N)/(N-1)]
03.20 I (K-1)3.4,3.5;I (K-3)3.6;T "STANDING DIASTOLIC
03.25 D 4;R
03.40 T "LYING SYSTOLIC
03.45 D 4;R
03.50 T "LYING DIASTOLIC
03.55 D 4;R
03.60 T "STANDING SYSTOLIC
03.65 D 4

04.10 S SP=10;D 4.5;T %1,K+1;S SP=7;D 4.5;T %5.02,MB
04.20 S SP=5;D 4.5;T %4.02,SD,!;R
04.50 F P=1,SP;T " "

05.10 I (HS(K)-B)5.2;R
05.20 S HS(K)=B

06.10 I (B-LS(K))6.2;R
06.20 S LS(K)=B

07.10 I (Z(B)-Z)7.2,7.3,7.3
07.20 T "  ";R
07.30 T "X "

08.10 S Z=H(K)+1;I (K),8.4;F P=1,55;T "-"
08.20 T !"      50          100         150         200         250"!
08.25 T "HISTOGRAM OF ";S K=K-1;D 3.2;T " B.P."!;S K=K+1
08.40 T !;R
*
```

```
*GO

::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::
::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::::

LYING SYSTOLIC                  = 1       = 158.11     = 26.08
LYING DIASTOLIC                 = 2       =  90.76     = 13.57
STANDING SYSTOLIC               = 3       = 146.42     = 21.40
STANDING DIASTOLIC              = 4       =  86.51     = 12.03

NUMBER OF PATIENTS ATTENDING= 53


= 12                      X
= 11                      X
= 10                      X
=  9                      X X
=  8                      X X
=  7                  X   X X
=  6                  X   X X X
=  5                  X X X X X         X
=  4                  X X X X X X       X
=  3                  X X X X X X       X
=  2              X X X X X X X X       X
=  1          X   X X X X X X X X X     X X
-------------------------------------------------------
     50         100         150         200         250
HISTOGRAM OF LYING SYSTOLIC     B.P.


= 24          X
= 23          X
= 22          X
= 21          X
= 20          X
= 19          X
= 18          X
= 17          X
= 16          X
= 15          X
= 14          X
= 13          X
= 12          X
= 11          X
= 10          X
=  9          X
=  8          X
=  7        X X
=  6        X X X X
=  5      X X X X X
=  4      X X X X X X
=  3      X X X X X X
=  2      X X X X X X
=  1    X X X X X X X
-------------------------------------------------------
     50         100         150         200         250
HISTOGRAM OF LYING DIASTOLIC    B.P.
```



HISTOGRAM OF STANDING SYSTOLIC B.P.



HISTOGRAM OF STANDING DIASTOLIC B.P.

*

```
            /22 MAY 1971
            /MODIFIED BY EDDY EMONS
            /REAL ARRAY OVERLAY FOR FOCAL 1969 WITHOUT EXTENDED
            /FUNCTIONS TO ALLOW STORAGE OF DATA ARRAYS
            /CALL: FNEW (J,Z)-STORES 3-WORD FLOATING POINT VARIABLE
            /Z IN ARRAY ELEMENT J. FNEW (J)-GETS ARRAYED DATA J
            /LIMITS: J MUST BE AN UNSIGNED INTEGER LESS THAN 1321
            START=5315
            *BOTTOM
0035  5314  FNEW-1
            *FNTABF+6     /REMOVE EXTENDED FUNCTIONS
0402  2725                ERROR5        /FATN
0403  2725                ERROR5        /FEXP
0404  2725                ERROR5        /FLOG
0405  2725                ERROR5        /FSIN
0406  2725                ERROR5        /FCOS
            *FNTABF+14
0410  5315  FNEW
            *START
5315  4453  FNEW,         JMS I INTEGE/FIX J VALUE
5316  1371                TAD M2451     /CHECK SIZE
5317  7500                SMA           /CORRECT SIZE
5320  4566                ERROR4        /J TOO LARGE!
5321  1372                TAD P2451     /RESTORE J VALUE
5322  3376                DCA TEMP
5323  1376                TAD TEMP      /MULTIPLY J BY 3
5324  7104                CLL RAL
5325  1376                TAD TEMP
5326  4542                PUSHA         /STORE ON AC STACK
5327  1066                TAD CHAR      /CHECK FOR SECOND ARG
5330  1374                TAD MCOMMA
5331  7640                SZA CLA
5332  5353                JMP XFCOMG    /NONE:DO GET
5333  4540                PUSHJ         /GET 2ND ARGUMENT INTO FLTG AC
5334  1612                EVAL-1        /(INTERNAL SUBROUTINES)
5335  7040                CMA           /GET BACK FIRST ARG.
5336  1413                POPA
5337  3011                DCA XRT       /SET UP POINTER TO STORE
5340  1375                TAD XFLAC     /SET POINTER TO FLTG ACC
5341  3012                DCA XRT2
5342  1373                TAD M3        /SET COUNT FOR 3 WORDS
5343  3377                DCA COUNT
5344  1412                TAD I XRT2    /MOVE FLTG ACC TO STORE
5345  6211                CDF T          /CHANGE TO DATA FIELD 1
5346  3411                DCA I XRT
5347  6201                CDF P          /START AT THIS ADDRESS
                          /THE ARRAY IN FIELD 0
5350  2377                ISZ COUNT
5351  5344                JMP .-5
```

```
5352  5536                JMP I EFUN3I/EXIT TO FOCAL FUNC PROGRAM
5353  1375  XFCOMG,       TAD XFLAC   /SET POINTER TO FLTG ACC
5354  3011                DCA XRT
5355  7040                CMA         /SET POINTER TO DATA
5356  1413                POPA
5357  3012                DCA XRT2
5360  1373                TAD M3      /SET 3 WORD COUNT
5361  3377                DCA COUNT
5362  6211                CDF T        /MOVE DATA TO FLOATING
                          /ACC FROM FIELD 1
5363  1412                TAD I XRT2
5364  6201                CDF P        /IN FIELD 0
5365  3411                DCA I XRT
5366  2377                ISZ COUNT
5367  5362                JMP .-5
5370  5536                JMP I EFUN3I/EXIT TO FOCAL FUNC PROGRAM
5371  5327  M2451,        -2451       /OCTAL FOR -1321
5372  2451  P2451,        2451        /OCTAL FOR 1321
5373  7775  M3,           -3
5374  7524  MCOMMA,-254
5375  0043  XFLAC,        44-1
5376  0000  TEMP,         0
5377  0000  COUNT,        0
            FNTABF=374
            INTEGER=53
            ERROR4=4566
            ERROR5=2725
            PUSHA=4542
            CHAR=66
            PUSHJ=4540
            EVAL=1613
            POPA=1413
            XRT=11
            XRT2=12
            EFUN3I=136
            T=10
            P=0
            BOTTOM=35

BOTTOM  0035
CHAR    0066
COUNT   5377
EFUN3I  0136
ERROR4  4566
ERROR5  2725
EVAL    1613
FNEW    5315
FNTABF  0374
INTEGE  0053
MCOMMA  5374
```

```
M2451    5371
M3       5373
P        0000
POPA     1413
PUSHA    4542
PUSHJ    4540
P2451    5372
START    5315
T        0010
TEMP     5376
XFCOMG   5353
XFLAC    5375
XRT      0011
XRT2     0012
```

```
                /ARRAY OVERLAY FOR FOCAL 1969 WITHOUT EXTENDED FUNCTIONS
                /TO ALLOW STORAGE OF INTEGER ARRAYS
                /CALL AS ABOVE
                /LIMITS :J MUST BE UNSIGNED INTEGER LESS THAN 3965
                /MODIFIED BY EDDY EMONS, 27 4 1971
                START=5335
                *BOTTOM
0035    5334    FNEW-1
                *FNTABF+6       /REMOVE EXTENDED FUNCTIONS
0402    2725                    ERROR5          /FATN
0403    2725                    ERROR5          /FEXP
0404    2725                    ERROR5          /FLOG
0405    2725                    ERROR5          /FSIN
0406    2725                    ERROR5          /FCOS
                *FNTABF+14
0410    5335    FNEW
                *START
5335    4453    FNEW,           JMS I INTEGE/COMMENTS AS ABOVE
5336    7100                    CLL
5337    1373                    TAD M7576
5340    7430                    SZL
5341    4566                    ERROR4
5342    1374                    TAD P7576       /USE J DIRECTLY
5343    4542                    PUSHA
5344    1066                    TAD CHAR
5345    1376                    TAD MCOMMA
5346    7640                    SZA CLA
5347    5361                    JMP XFCOMG
5350    4540                    PUSHJ
5351    1612                    EVAL-1
5352    1413                    POPA
5353    3377                    DCA TEMP
5354    5453                    JMP I INTEGE/GET INTEGER Z
                                        /STORE INTEGER ONLY
5355    6211                    CDF T
5356    3777                    DCA I TEMP
5357    6201                    CDF P
5360    5536                    JMP I EFUN3I
5361    1413    XFCOMG,         POPA
5362    3377                    DCA TEMP
5363    6211                    CDF T
5364    1777                    TAD I TEMP
5365    6201                    CDF P
5366    3045                    DCA FLAC+1
5367    1375                    TAD P13
5370    3044                    DCA FLAC
5371    3046                    DCA FLAC+2
5372    5536                    JMP I EFUN3I
5373    0202    M7576,          -7576
```

```
5374   7576   P7576,        7576
              P=0
              T=10
5375   0013   P13,          13
              FLAC=44
              INTEGER=53
              CHAR=66
              ERROR4=4566
              ERROR5=2725
              PUSHA=4542
5376   7524   MCOMMA,       -254
              PUSHJ=4540
              EVAL=1613
              POPA=1413
              EFUN3I=136
              BOTTOM=35
              FNTABF=374
5377   0000   TEMP,         0

BOTTOM   0035
CHAR     0066
EFUN3I   0136
ERROR4   4566
ERROR5   2725
EVAL     1613
FLAC     0044
FNEW     5335
FNTABF   0374
INTEGE   0053
MCOMMA   5376
M7576    5373
P        0000
POPA     1413
PUSHA    4542
PUSHJ    4540
P13      5375
P7576    5374
START    5335
T        0010
TEMP     5377
XFCOMG   5361
```