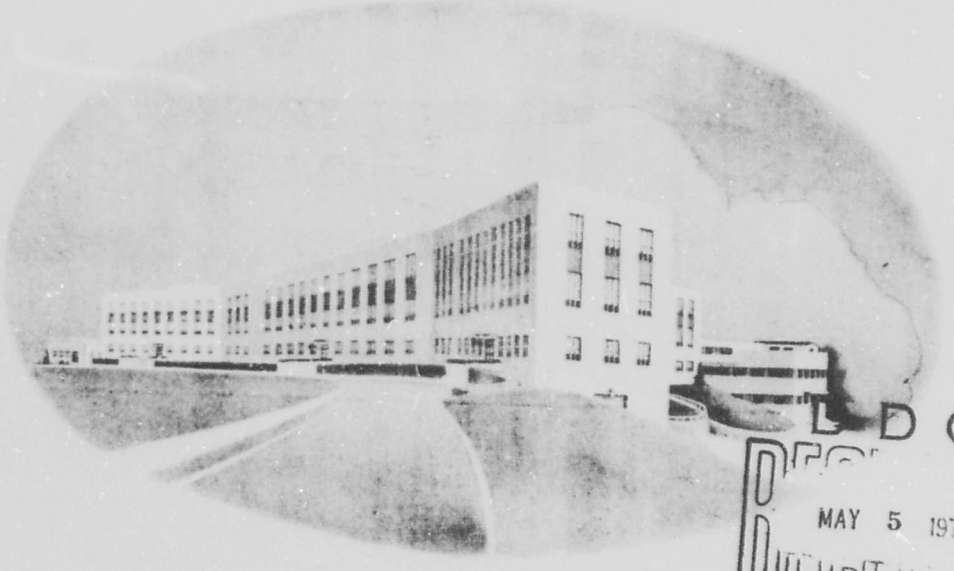
NAVAL MEDICAL RESEARCH INSTITUTE

NMRI



ANALYSIS OF BIOMEDICAL DATA BY TIME-SHARING COMPUTERS
I. NON-LINEAR REGRESSION ANALYSIS

RESEARCH REPORT

Project No. MR005.20-0287

Report No. 25

ANALYSIS OF BIOMEDICAL DATA BY TIME-SHARING COMPUTERS

I. NON-LINEAR REGRESSION ANALYSIS

D. L. Horwitz, M.D., Ph.D.
and
Louis D. Homer, M.D., Ph.D.

RESEARCH REPORT

Project No. MR005.20-0287 Report No. 25

Naval Medical Research Institute
National Naval Medical Center
Bethesda, Maryland 20014

February 26, 1970

In the analysis of biological and medical data, we frequently desire to obtain mathematical equations which describe numerical data and predict the results of future experiments. Because of the nature of biological systems, the equations we desire are often not linear. In the past, such non-linear systems were often avoided because mathematical methods for analyzing them were tedious and, occasionally, quite imprecise. However, with the increasing availability of time-sharing computers, we may now deal readily with non-linear biological systems. To facilitate this, a computer program has been devised which may readily be used by persons without statistical training or computer experience. It is written in the "BASIC" computer language, which was designed at Dartmouth College as a language which the non-programmer can learn in a few hours.

The program is a very general one for fitting a function of one independent variable and one dependent variable to any equation containing up to ten parameters. That is, we wish to find the parameters B_1, B_2,..., B_k (where k is no larger than ten) in the relation $Y = f(X,B_1,B_2,\ldots,B_k)$ where $\underline{f}$ is an arbitrary function. The program is based on an algorithm which is valid for functions which are non-linear in the B's and thus differs from the usual linear regression methods. We shall show later how the program may be modified to take into account more than one independent variable.

METHOD

We base our program on the method of Marquardt (1), which uses a Taylor's series to give an improved estimation of the B's for any given set of B's. It can be shown (1) that if a parameter (called lambda) is chosen large enough, the B's will always converge to their best values (by a least squares criterion), but that this convergence will be faster for small values of lambda. Given an initial set of B's the program will, for a given lambda, compute a new set of B's. It compares

the sum of the squares of the "errors" (i.e., $Y-f(X,B_1,B_2,\ldots,B_k)$, or the difference between the observed and calculated values of Y), abbreviated SSE, for the two sets of B's and if the new B's reduce the SSE, proceeds to calculate still another set of B's after first reducing lambda by a factor of ten. If, on the other hand, the SSE has increased with the new B's, the old B's will be used with a larger (by a factor of ten) lambda to determine the next set of B's. In this way, the program corrects for "overshooting" in making its approximations.

The Program

The program is written in the BASIC language for use on Time-Sharing facilities. The General Electric Mark I System was used for all calculations in this study, but the program should be adjustable for other systems. A complete listing of the program is given in Appendix A.

Use of the Program

1. The function which is to be fitted to the data is entered in terms of the "error" between observed and calculated values of Y on line 1000 of the program as follows:

1000 LET E=Y(J)-F(X(J),B(1),B(2),...B(K)), where J is a subscript identifying particular sets of X and Y. If necessary, the function F can be continued on lines 1001 to 1009 noting that it appears after a "minus" sign. Thus an example could be:

1000 LET E=Y(J)-(B(1)*EXP(B(2)*X(J)+B(3)*EXP(B(4)*X(J))), or for the same example,

```
1000LET E=Y(J)-B(1)*EXP(B(2)*X(J))
1001 LET E=E-B(3)*EXP(B(4)*X(J))
```

Both of these examples represent the equation $F=B_1e^{B_2X}+B_3e^{B_4X}$.

BLANK PAGE

2. The next data to be entered are, in order,

1) The maximum number of iterations to be performed

2) The initial value of lambda

3) The number of data points

4) The number of parameters (B) to be estimated (i.e., k).

Thus, if we desire a maximum of ten iterations to be performed starting with a lambda of one, and there are 100 data points and four parameters to be estimated, we enter this data in line 1011 of the program as:

1011 DATA 10,1,100,4.

3. Next are entered the starting values of B(1), B(2),...B(k). These values should be reasonable estimates based on whatever is known about the system being studied, so that the estimation converges to the best value in a minimum amount of time. None of the starting values may be zero.

4. Finally are entered the data Y(J), X(J). The Y's must proceed the X's. Up to 200 data points may be used. The number of pairs of data entered must be the same as entered in step 2 above.

Running the Program

After entering the data, the program is started in the usual way by typing RUN on the teletype. Initial printout consists of listings of the assigned starting values of the B parameters and the data for observations of X(J) and Y(J). If these are found to be in error (due to incorrectly punched or omitted data) it is suggested that the run be terminated and corrections made, as errors are likely to be compounded in the iterative process. The teletype will then print the SSE for each set of B's, the lambda used in finding the next set of B's, the SSE for the new set of B's, and the new set of B's. If the SSE has increased instead of

decreased, new B's will be found using a larger lambda. This does not count as an iteration, and no provision has been made for printing the new lambda (it will be ten times the old lambda). If the SSE is smaller, we go on to the next iteration using a smaller lambda.

If the new B's do not differ from the old B's by more than 0.01%, the computer types CONVERGENCE and gives a final set of B's based on lambda=0. This convergence limit may be changed, if desired, by modification of line 345 in the program to indicate the desired value. If the maximum number of iterations (entered in step 2 above) has been exceeded, it prints YOU ARE OUT OF ITERATIONS and computes a final set of B's based on lambda=0. The computer will then ask if you wish a printout of:

(1) The variance-covariance matrix

(2) A calculated Y for each X entered in the data, based on the best values of the parameters B found by the iteration. This will be printed along with the "error" or residual, and the 95% confidence limits of the calculated values of Y (that is, Y±2 standard deviations). Note that these are only approximate confidence limits--the goodness of the approximation increasing as the function approaches linearity.[2] Other limits may be selected by modifying line 580 to indicate the number of standard deviations to be included in the range. The value of the standard deviation is listed in the printout as the "standard error of the estimated y" or "S.E. EST. Y."

(3) A table of residuals, i.e., Y(J)-F(X(J), B(1),...B(k)) for each set of X,Y entered in the original data. This is printed in the form:

X(1),E(1) X(2),E(2) X(3),E(3) X(4),E(4)

X(5),E(5) X(6),E(6) .....

This option is given for instances when we do not require the entire printout of option (2) but need only to examine the residuals for reasons of deciding whether a systemic bias is present in the equation.

Finally, the correlation matrix of the B's will be printed.

Modifications

Modifications of the convergence limit and the tabulated range of the calculated Y have already been discussed. We can also modify the program for use when Y is a function of several variables, which we shall call X(1), X(2),...X(M). We add the program lines:

```
20 DIM X(N,M), Y(N)
76 READ Y(I)
77 PRINT Y(I),
78 FOR J=1 TO M
80 READ X(I,J)
81 PRINT X(I,J),
85 NEXT J
86 PRINT
```

In the above, numerical values of N (the number of data points) and M (the number of independent variables) must be inserted. If the product N*M is too large, an error message will appear. In the final printout, a zero will replace the given values of the X's unless line 595 is modified to indicate that all values of X(I,J) are to be printed, e.g. 595 PRINT X(J,1); X(J,2),.....

Examples

Appendix B gives a straightforward use of the program with one independent variable and two parameters. It fits the simple linear equation $Y=B_1+B_2X$ to the data. Note that this program is not actually recommended for linear regressions of this type, for which more efficient programs exist. Appendix C gives an example of modifying the program for two independent variables, and fits the equation $Y=B_1 (X_1^{B_2}X_2)$.

REFERENCES

1. Marquardt, D.W., An algorithm for least-squares estimation of non-linear parameters. J. Soc. Indust. Appl. Math., 11: 431-441 (1963).

2. Draper, N.R., and Smith, H., Applied Regression Analysis (New York: John Wiley & Sons, Inc., 1966).

```
NONLIN


2   REM THIS IS A PROGRAM FOR FITTING A GENERAL FUNCTION
3   REM TO NON-LINEAR PARAMETERS USING A LEAST SQUARES METHOD.
4   REM
10   DIM O(10,10)
15 DIMA(10,10),C(10,10),B(10),G(10),P(10)
20 DIMX(200), Y(200)
25 LETY1=0
30 LETY2=0.
35 READ T1,L,N,N1
40 PRINT"STARTING PARAMETERS"
45 FORI=1TON1
50 READB(I)
55 PRINTB(I),
60 NEXTI
65 PRINT
70 PRINT
71 PRINT "OBSERVATIONS     X,Y"
75 FOR I=1 TO N
80 READY(I),X(I)
85 PRINT X(I), Y(I)
90 LETY1=Y1+Y(I)
95 LETY2=Y2+Y(I)*Y(I)
100 NEXT I
105 LET T2=0
110 LET T2=T2+1
115 LET SO=0
120 FOR I=0TO N1
125 LET G(I)=0
130 NEXT I
135 MAT A=ZER(N1,N1)
140 MAT Q=ZER(N1,N1)
145 MAT C=ZER(N1,N1)
150 FOR J=1 TO N
155 GOSUB 1000
160 LET SO=SO+E*E
165 LET E1=E
170 FOR I=1 TO N1
175 LET B(I)=B(I)*1.001
180 GOSUB 1000
185 LET B(I)=B(I)/1.001
190 LET P(I)=(E1-E)/(.001*B(I))
195 NEXT I
200 FOR I=1 TO N1
205 LET G(I)=G(I)+E1*P(I)
210 FOR I1=1 TO N1
215 LET A(I,I1)=A(I,I1)+P(I)*P(I1)
220 NEXT I1
225 NEXT I
230 NEXT J
235 FOR I=1 TO N1
```

Appendix A-2

NØNLIN CØNTINUED

```
240 FØR J=1 TØ N1
245 LET Q(I,J)=A(I,J)/(SQR((A(I,I))*(A(J,J))))
250 NEXT J
255 LET G(I)=G(I)/(SQR(A(I,I)))
260 NEXT I
265 PRINT"SSE=";S0;"ITERATIØN NØ.";T2;"LAMBDA=";L
270 PRINT"R SQUARE=";1.-(S0/(Y2-(Y1*Y1/N)))
275 FØR I=1 TØ N1
280  LET Q(I,I)=Q(I,I)*(1+L)
285 NEXT I
290 MAT C=INV(Q)
295 FØR I =1TØ N1
300 LET P(I)=0
305 FØR J=1TØ N1
310 LET P(I)=P(I)+C(I,J)*G(J)
315 NEXT J
320 LET P(I)=P(I)/(SQR(A(I,I)))
325 NEXT I
330 IF T1<0 THEN500
335 IF T2>=T1 THEN 365
340 FØR I=1 TØ N1
345 IF ABS(P(I)/B(I))>=.0001 THEN375
350 NEXT I
355 PRINT "CØNVERGENCE"
360 GØT0485
365 PRINT"YØU ARE ØUT ØF ITERATIØNS"
370 GØ TØ 485
375 FØR I=1 TØ N1
380 LET B(I)=B(I)+P(I)
385 NEXT I
390 LET S1=0
395 FØR J=1 TØ N
400 GØSUB 1000
405 LET S1 =S1+E*E
410 NEXT J
415 PRINT"SSE FØR NEXT B=";S1
420 PRINT"PARAMETERS"
425 FØR I=1 TØ N1
430 PRINT B(I),
435 NEXT I
440 PRINT
445 IF S1>S0 THEN 460
450 LET L=L/10
455 GØ TØ 110
460 LET L=L*10
465 FØR I=1 TØ N1
470 LET B(I)=B(I)-P(I)
475 NEXT I
480 GØ TØ 275
485 LETT1=-1
```

NONLIN CONTINUED

```
490 LETL=0
495 GØTØ110
500 PRINT
505 LETV=SO/(N-N1)
510 LETV1=SQR(V)
515 PRINT"VARIANCE=";V;"STD. DEV.=";V1;"SSE=";SO
516 PRINT
518 MAT C=INV(A)
520 MAT A=(V)*C
522: FINAL PARAMETERS   STD. ERRØR ØF PARAM.     CØEFF. ØF VAR.
523: ----------------   -------------------     -------------
524:     ###.####            ###.####             ###.####
525 PRINT USING 522
526 PRINT USING 523
530 FØRI=1TØN1
535 LET D=SQR(A(I,I))
540 PRINT USING 524,B(I),D,D/B(I)
541 NEXT I
542 PRINT  "PRINT VARIANCE-CØVARIANCE MATRIX(0=NØ,1-YES)";
543 INPUT H3
544 IF H3=0 THEN 550
545 MAT PRINT A
548 PRINT
549 PRINT
550 PRINT "DØ YØU WISH PRINTØUT ØF ESTIMATED Y FØR EACH X"
555 PRINT "(0=NØ,1=YES)";
560 INPUT H1
565 IF H1=0 THEN 690
570 PRINT"X","Y","ESTIMATED Y","ERRØR"
575 PRINT" "," ","LØWER LIMIT","UPPER LIMIT","S.E. EST. Y"
580 LETT=2.
585 FØRJ=1TØN
590 GØ SUB1000
595 PRINTX(J),Y(J),Y(J)-E,E
600 LETE1=E
605 FØRI=1TØN1
610 LETB(I)=B(I)*1.001
615 GØSUB1000
620 LETB(I)=B(I)/1.001
625 LETP(I)=(E1-E)/(.001*B(I))
630 NEXTI
635 LETV1=0.
640 FØRK=1TØN1
645 FØRK1=1TØN1
650 LETV1=V1+A(K1,K)*P(K1)*P(K)
655 NEXTK1
660 NEXTK
665 LETV2=SQR(V1)
670 LETL1=Y(J)-E1-T*V2
675 LETL2=L1+2.*T*V2
```

NONLIN CONTINUED

```
680 PRINT"                                    ",L1,L2,V2
685 NEXT J
690 PRINT "DØ YOU WISH TABLE ØF RESIDUALS(0=NØ, 1=YES)";
695 INPUT H2
700 IF H2>0 THEN 718
705 PRINT"CØRRELATIØN MATRIX"
710 MAT PRINT Q
715 STØP
718 PRINT
719 PRINT
720 PRINT "                RESIDUALS"
725 PRINT
730 FØR J=1 TØ N
735 GØ SUB 1000
739: ##.##,###.##      ##.##,###.##      ##.##,###.##      ##.##,###.##
740 PRINT USING 739, X(J),E;
745 NEXT J
746 PRINT
747 PRINT
750 GØ TØ 705
755 STØP
1000 REM  ØN THIS LINE IS ENTERED THE FUNCTIØN TØ WHICH WE WISH
1001 REM  TØ FIT ØUR DATA.  THE UNKNOWN PARAMETERS ARE B(1),
1002 REM  B(2),...B(K) WHERE K<10.  IF Y(J) IS THE J-TH
1003 REM  ØBSERVATIØN ØF THE DEPENDENT VARIABLE, AND X(J) IS
1004 REM J-TH ØBSERVATIØN ØF THE INDEPENDENT VARIABLE, THEN
1005 REM  LINE 1000 IS WRITTEN E=Y(J)-F(X(J),B(1),...B(K)).
1006 REM THE FUNCTIØN MAY BE CØNTINUED UP TØ LINE 1009 IF ATTENTIØN
1007 REM IS PAID TØ SIGN.
1009 RETURN
1010 REM  ENTER, IN ØRDER, THE NUMBER ØF ITERATIØNS TØ BE PERFØRMED,
1011 REM  THE STARTING VALUE ØF LAMBDA (USUALLY 10),THE NUMBER ØF
1012 REM  DATA PØINTS, THE VALUE ØF K, THE STARTING VALUES ØF
1013 REM  B(1),...B(K), AND THE PAIRS ØF DATA IN THE ØRDER
1014 REM  Y(I), X(I).
9999 END
```

Appendix B-1

```
ØLD
ØLD FILE NAME--NØNLIN


READY.

1000 LET E=Y(J)-(B(1)+B(2)*X(J))
1010 DATA 10, 0.001, 6, 2
1011 DATA 0.01, 0.5
1101 DATA 0, 1.0
1102 DATA 2.0, 4.7
1103 DATA 4.1, 8.0
1104 DATA -1.00, -1.9
1105 DATA -2.9, -6.0
1106 DATA 0.01, 0.0
RUN



NØNLIN     19:10    04 MØN 02/16/70

STARTING PARAMETERS
 .01              .5

ØBSERVATIØNS     X,Y
 1                0
 4.7              2
 8                4.1
-1.9             -1
-6               -2.9
 0                .01
SSE= .4095    ITERATIØN NØ. 1 LAMBDA= .001
R SQUARE= .986074
SSE FØR NEXT B= .304193
PARAMETERS
-.105457          .490235
SSE= .304193 ITERATIØN NØ. 2 LAMBDA= .0001
R SQUARE= .989655
SSE FØR NEXT B= .304193
PARAMETERS
-.105557          .490231
SSE= .304193 ITERATIØN NØ. 3 LAMBDA= .00001
R SQUARE= .989655
CØNVERGENCE
SSE= .304193 ITERATIØN NØ. 4 LAMBDA= 0
R SQUARE= .989655

VARIANCE= 7.60482E-02    STD. DEV.= .275768 SSE= .304193
```

```
 FINAL PARAMETERS     STD. ERRØR ØF PARAM.     CØEFF. ØF VAR.
 ----------------     --------------------     --------------
      -.1056                .1152                -1.0910
       .4902                .0251                 .0511
PRINT VARIANCE-CØVARIANCE MATRIX(0=NØ,1-YES)? 1
 1.32617E-02   -6.07083E-04

-6.07083E-04    6.28016E-04


DØ YØU WISH PRINTØUT ØF ESTIMATED Y FØR EACH X
(0=NØ,1=YES)? 1
X              Y          ESTIMATED Y      ERRØR
                          LØWER LIMIT      UPPER LIMIT     S.E. EST. Y
 1              0           .384674         -.384674
                            .159502          .609845        .112586
 4.7            2          2.19853          -.198527
                           1.90576          2.49129         .146382
 8              4.1        3.81629           .283711
                           3.398            4.23458         .209144
-1.9           -1         -1.037            3.69952E-02
                          -1.30409          -.769896        .13355
-6             -2.9       -3.04694           .146941
                          -3.46242         -2.63147         .207738
 0              .01        -.105557          .115557
                          -.335875           .124762        .115159
DØ YØU WISH TABLE ØF RESIDUALS(0=NØ,1=YES)? 1

            RESIDUALS

  1.00,  -.38       4.70,  -.20       8.00,   .28      -1.90,   .04
-6.00,   .15        .00,   .12

CØRRELATIØN MATRIX
 1                 .21036

 .21036            1
```

Appendix C-1

This example indicates how the program may be modified to accommodate functions of more than one independent variable. It finds the best values of B_1 and B_2 to fit the equation

$$Y = B_1(X_1^{B_2 X_2}).$$

Note that, unless the program is further modified to account for the additional independent variables, values of $\underline{x}$ in the "residuals" section will all be listed as zero.

```
ØLD
ØLD FILE NAME--NØNLIN

READY.

20 DIM X(6,2), Y(6)
76 READ Y(I)
77 PRINT Y(I),
78 FØR J=1 TØ 2
80 READ X(I,J)
81 PRINT X(I,J),
85 NEXT J
86 PRINT
595 PRINT X(J,1); X(J,2), Y(J), Y(J)-E, E
1000 L3
1000 LET E=Y(J)-(B(1)*X(J,1)↑(B(2)*X(J,2)))
1010 DATA 10, 10, 6, 2
1011 DATA 2,2
1101 DATA 1,0,0
1102 DATA 4,1,1
1103 DATA 2,1,2
1104 DATA 8,2,1
1105 DATA 0,0,2
1106 DATA 2,2,0
RUN
```

```
NONLIN     19:23    04 MON 02/16/70

STARTING PARAMETERS
 2                 2

OBSERVATIONS      X,Y
 1                 0                 0
 4                 1                 1
 2                 1                 2
 8                 2                 1
 0                 0                 2
 2                 2                 0
SSE= 5 ITERATION NO. 1 LAMBDA= 10
R SQUARE= .877551
SSE FOR NEXT B= 4.99121
PARAMETERS
 2.00458           1.9997
SSE= 4.99121 ITERATION NO. 2 LAMBDA= 1
R SQUARE= .377766
SSE FOR NEXT B= 4.94072
PARAMETERS
 2.03422           1.98754
SSE= 4.94072 ITERATION NO. 3 LAMBDA= .1
R SQUARE= .879003
SSE FOR NEXT B= 4.79322
PARAMETERS
 2.14756           1.90364
SSE= 4.79322 ITERATION NO. 4 LAMBDA= .01
R SQUARE= .882615
SSE FOR NEXT B= 4.75026
PARAMETERS
 2.24214           1.83452
SSE= 4.75026 ITERATION NO. 5 LAMBDA= .001
R SQUARE= .883667
SSE FOR NEXT B= 4.75
PARAMETERS
 2.24995           1.83011
SSE= 4.75 ITERATION NO. 6 LAMBDA= .0001
R SQUARE= .883673
CONVERGENCE
SSE= 4.75 ITERATION NO. 7 LAMBDA= 0
R SQUARE= .883673

VARIANCE= 1.1875  STD. DEV.= 1.08972 SSE= 4.75

 FINAL PARAMETERS    STD. ERROR OF PARAM.     COEFF. OF VAR.
 ----------------    --------------------     --------------
      2.2499                .5449                  .2422
      1.8301                .4006                  .2189
PRINT VARIANCE-COVARIANCE MATRIX(0=NO,1-YES)? 1
 .296875         -.19024

-.19024           .160478
```

```
DØ YØU WISH PRINTØUT ØF ESTIMATED Y FØR EACH X
(0=NØ,1=YES)? 1
```

X		Y	ESTIMATED Y LØWER LIMIT	ERRØR UPPER LIMIT	S.E. EST. Y
0	0	1	2.24995 -1.24995		
1	1	4	1.16022 3.33967	2.24995 1.75005	.544862
1	2	2	1.16022 3.33967	2.24995 -.249946	.544862
2	1	8	1.16022 3.33967	7.99999 8.09133E-06	.544862
0	2	0	5.82054 10.1794	0 0	1.08972
2	0	2	0 0	0 0	0
			2.24995 -.249946	1.16022 3.33967	.544862

```
DØ YØU WISH TABLE ØF RESIDUALS(0=NØ,1=YES)? 1
```

RESIDUALS

.00, -1.25	.00, 1.75	.00, -.25	.00, .00
.00, .00	.00, -.25		

CØRRELATIØN MATRIX

1	.87158
.87158	1

UNCLASSIFIED
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY *(Corporate author)*
NAVAL MEDICAL RESEARCH INSTITUTE
NATIONAL NAVAL MEDICAL CENTER
BETHESDA, MARYLAND 20014

2a. REPORT SECURITY CLASSIFICATION
UNCLASSIFIED

2b. GROUP

3. REPORT TITLE
ANALYSIS OF BIOMEDICAL DATA BY TIME-SHARING COMPUTERS
I. NON-LINEAR REGRESSION ANALYSIS

4. DESCRIPTIVE NOTES *(Type of report and inclusive dates)*
Medical research interim report

5. AUTHOR(S) *(First name, middle initial, last name)*
David L. HORWITZ and Louis D. HOMER

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
26 February 1970	16	2

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	MR005.20-0287, Report No. 25
c.	9b. OTHER REPORT NO(S) *(Any other numbers that may be assigned this report)*
d.	

10. DISTRIBUTION STATEMENT
THIS DOCUMENT HAS BEEN APPROVED FOR PUBLIC RELEASE AND SALE: ITS DISTRIBUTION IS UNLIMITED.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Bureau of Medicine and Surgery (Navy) Washington, D. C.

13. ABSTRACT

Analysis of biomedical data frequently involves obtaining a mathematical equation to describe numerical data. Because of the nature of biological systems, the desired equations are often not linear. In the past, such non-linear systems were often avoided because mathematical methods for analyzing them were tedious and, occasionally, quite imprecise. Modern computers have made such systems far easier to work with. This publication describes a program for obtaining these non-linear "regression" equations. The program described has a number of advantageous properties: (1) it can be used for almost any type of equation, including exponential, logarithmic, and power functions; (2) it is written in the BASIC program language, which can be learned by a person without computer knowledge in only a few hours; (3) it is designed for time-sharing computers, so that the user may readily interact with the computer and modify the equations being used at will; and (4) the output of the program includes sufficient statistical parameters to enable further evaluation of the results. In addition to giving the program, this publication gives full instructions on its use, suggestions for modifying the program for additional applications, and several illustrative examples.

DD FORM 1 NOV 65 1473 (PAGE 1)
S/N 0101-807-6801

UNCLASSIFIED
Security Classification