PROGRAM ABSTRACTS/ALGORITHMS

The last normal curve programs

D. LOUIS WOOD

University of Arkansas at Little Rock, Little Rock, Arkansas

The calculation of probabilities for parametric distributions such as chi square, z , t , and F has been a topic of enduring interest. Jaspens (1965) introduced techniques and a FORTRAN program that transform F , t , and chi square into approximations of the normal variate z . The probability associated with z is estimated using a transform and a correction formula introduced by Kelley (1947, 1948). Clearly, the accuracy of Jaspens's methods would be enhanced via use of techniques of finding normal curve probabilities that are exactly accurate to a large number of decimal places.

O'Grady (1981) outlined a FORTRAN IV program for the IBM 3032 that estimates probabilities and critical values for z , chi square, r , t , and F . The critical values are reportedly accurate to 14 decimal places, so the user having access to an IBM 3032 has everything necessary for a significance test at one of the specified α s. The computations of probabilities associated with specific values of these parametric statistics can in the worst case be accurate to only 3 places with error appearing in the 4th decimal place. O'Grady did not provide a program listing, which makes the routine impossible to evaluate; however, the occasional occurrence of 3-place accuracy and the requirement of a large computer limits its usefulness.

In 1983, von Collani published a general description of a Microsoft BASIC routine that estimates probabilities associated with F , t , chi square, and z , but, like O'Grady (1981), von Collani did not provide the routine for evaluation. The routine seems to be advanced over earlier versions, because it runs on a TRS-80 microcomputer (presumably a Model III), making it available and convenient to many, and it is fast. Unfortunately, Collani was vague in describing the accuracy of the routine, stating that the "accuracy is at least 5 decimal places for the normal distribution and about 4 to 6 decimal places for the other distributions" (p. 543). Scholars publishing probability figures would probably insist on exact-accuracy figures.

Recently Evans and Gilfillan (1986) published a probability-approximation routine written in APL*PLUS/PC. This program appears to be a regression in that it makes errors in the 3rd decimal place, fails

to directly accept ts , and computes binomial probabilities only when $p = .5$.

Two common flaws underlie the weaknesses of all these routines: parametric numerical integration is too large a task to undertake with a single program and a variety of approximation algorithms having limited accuracy have been repeatedly used. The techniques of Kelley (1947, 1948) were employed, for example, by Jaspens (1965) and in F probabilities by Evans and Gilfillan (1986). As von Collani (1983) pointed out, "the accuracy of these approximation formulas is rather poor in most cases" (p. 543).

Remedies for these two weaknesses or probability-finding programs began with the work of Brophy (1983), who recognized that the scientist seeking probabilities would be better served by specific routines than by a single master program and that more powerful mathematics was needed. He examined seven techniques for finding normal curve probabilities and discovered that the technique of Moran (1980) provides 6- to 7-decimal-place accuracy when single-precision arithmetic was used and 9- to 10-place accuracy when the 9-digit numerical precision of the Radio Shack Color Computer running Extended Color BASIC was employed.

Although these levels of accuracy are impressive, Brophy's (1983) focus on comparing the seven algorithms apparently led into brief descriptions of how each is applied and a program that is not as flexible as its users might wish. The routines will not, for example, directly provide the normal curve area between $Z = -1$ and $Z = 2$. Brophy pointed the way to accurate probability determination by focusing on a single statistic, z , and by employing more powerful techniques than the old standbys.

Wood and Wood (1984) applied an approach congruent to Brophy's in addressing t probabilities. Specifically, their routine only treats t and does so via the high precision of Simpson's rule, which finds the area under a function, here $f(t)$, between two end points. The curve over this range is divided into pairs of segments (Anton, 1981), and a parabola is fitted to the top of each pair. Because the tops curve, they fit many functions, such as $f(t)$, well enough to provide accuracy comparable with the trapezoidal rule, but Simpson's rule uses half as many segments and this greatly reduces computer running time.

Because Wood and Wood (1984) sought generality of application, their program was written in single numerical precision in Microsoft BASIC for computers such as the IBM-PC and the TRS-80 Models III and 4, and that restricted the precision of the probabilities calculated to 5-place accuracy. Wood (1985) brought Simpson's rule to bear on the chi-square distribution and attained probabilities accurate to 6 decimal places over a wide range of degrees of freedom.

The author's mailing address is: University of Arkansas at Little Rock, 2801 South University, Little Rock, AR 72204.

In developing an algorithm for the numerical integration of the family of F distributions, Wood and Wood (1986) partially abandoned the goals of general applicability and speed of computation and instead emphasized the degree of accuracy of the probabilities computed. Their Simpson's rule algorithm divided the various F functions into enough segments to provide results accurate from 5 to 9 places according to the value of F and the two degrees of freedom associated with F . They also developed the program specifically for the IBM PC and other computers offering the BASIC/D option, which carries square roots, logs, functions of e , and other functions that have been defined in double precision to a full 16 places of accuracy.

The purpose of the present work was to develop normal curve programs that run on unmodified IBM PC computers, that are always highly accurate, and that will replace techniques currently used. The second goal was to explicate the process of curve area finding in sufficient detail to enable the reader to abandon less accurate techniques.

The Algorithm. The present work extends the focus of Wood and Wood (1986) by using Simpson's rule to determine normal curve probabilities with very high accuracy. It is presented in two versions, one of which delivers 6- to 9-place accuracy and maximum running time of 55 sec on the unmodified IBM PC. This program, named NORMLCV.QIK, offers a maximum range of $Z = -6$ to $Z=6$. It runs in BASIC/D and is recommended for general use. The other program, NORMLCV.EXT, places an even greater emphasis on accuracy and accordingly runs more slowly. Running times vary from a few seconds to 3.5 min on the IBM PC using BASIC/D. The minimum accuracy of the probabilities delivered is 8 decimal places and the maximum is 12, making this version ideal for mathematical and theoretical applications and for all other work in which extremely accurate and certainly exact probabilities are required. It may be preferred for general use by those having computers that are markedly faster than a standard PC. Its maximum range is $Z=-7$ to $Z=7$.

The Density Function. The density function of the unit normal curve is usually cited in the following form (Loftus & Loftus, 1982, p. 162):

$$f(z) = \left(\frac{1}{\sqrt{2\pi}} \right) e^{-.5z^2}.$$

Substitution of the values of π and e and performing the indicated arithmetic give the following result, which is the version used in the algorithm:

$$f(z) = \frac{.3989422804014327}{1.648721270700128^2}.$$

When the program is run, the user directly or indirectly defines the range of numerical integration with two Z scores. In some cases, one Z score is input and the pro-

gram sets the other limit at $Z=0$. Techniques described in the next section determine the number of segments into which this range will be divided, and then Simpson's rule finds and sums those areas, yielding $P(z)$.

Accuracy and the Number of Segments Employed. The accuracy with which Simpson's rule calculates areas under curves depends on the number of segments into which the range of numerical integration is divided, the curvature of the function within that range as indexed by the fourth derivative, and the width of that range. The maximum amount of error that can occur is given by the following inequality (Anton, 1981, p. 534):

$$|\text{error}| \leq \frac{M(b-a)^5}{180N^4}. \quad (1)$$

M is defined in the next section, b is the upper limit of the range, a defines the lower limit, and N is the number of segments used.

With the algorithms presented, the range ($R=b-a$) is user supplied; if, for example, the user wishes to know the normal curve area between $Z=3.1$ and $Z=1.5$, then $b=3.1$, $a=1.5$, and the range (R) = $3.1-1.5 = 1.6$.

When the user enters the values of b and a , the program reads a table indicating the minimum error achievable in 55 sec or less on the PC with NORMLCV.QIK and in 3.5 min or less with NORMLCV.EXT. This minimum error is substituted into the arrangement of Inequality 1, and it indicates a value of N that mathematically guarantees at least the tabled accuracy:

$$N \leq \frac{\sqrt[4]{M} (b-a)^{1.25}}{\sqrt[4]{|\text{error}| 180}}. \quad (2)$$

The programs evaluate this during each run, under the restriction that N must be even, and assigns the number of segments into which the normal curve range specified by the user will be divided.

The remaining source of error stems from the round-off, which necessarily occurs during the $N+1$ times that $f(z)$ is evaluated during each run. These errors, of course, begin in the last decimal place of the constants and variables involved and move extremely slowly leftward as a function of the number of iterations performed. Round-off error and N are directly related. The two programs avoid round-off errors in two ways. First, all nonintegral numbers and all subroutines having nonintegral solutions are processed with 16-place precision via double-precision arithmetic and BASIC/D. Because round-off errors begin in the extreme decimal places, they should never reach the (maximum) 12-place probabilities of NORMLCV.EXT and the (maximum) 9-place probabilities of NORMLCV.QIK; in fact, they do not. As part of the testing process, NORMLCV.EXT was modified to provide 15-decimal-place probabilities across its range, and the results were compared with the 15-place probabilities published by Abramowitz and Stegun (1964). Full agreement occurred without exception throughout exten-

sive testing. The user is assured by these tests, as well as by the mathematics of Inequalities 1 and 2, that all digits displayed in probabilities are exactly accurate.

The Fourth Derivative of $f(z)$. As stated, the range is defined as $R=b-a$, where b and a are user-supplied Z scores. Inequality 1 contains a value M , which is a number very slightly larger than the maximum of the fourth derivative in that range. For example, where $b=1$ and $a=-.8$, the maximum value of $f^4(z)=1.1968$ makes $M=1.2$ a usable value. In fact, 1.1968 is the maximum of $f^4(z)$ in the range $-\infty < Z < \infty$ and occurs at $Z=0$.

Because the calculation of M for any specified range is time-consuming, the program assigns $M=1.2$ in each of the following conditions: $b > 0$ and $a < 0$, $b=0$, $a=0$, $|b| < .3$, and $|a| < .3$. In English, if the range of integration contains or is bounded by zero or either b or a has an absolute value within .3 of zero, then $M=1.2$ is used. When both b and a are on the same side of $Z=0$ and both have an absolute value of .3 or greater, M is less than 1 for all possible ranges and running speed is enhanced by substituting the exact value of M into Inequality 1. This is accomplished by evaluating the following formula at .10 intervals across R and assigning M the maximum computed $f^4(z)$:

$$f^4(z) = \left(\frac{1}{\sqrt{2\pi}} \right) e^{-z^2/2} (z^4 - 6z^2 + 3).$$

Computing M rather than always substituting $M=1.2$ reduces the number of segments employed, which reduces running times and helps eliminate round-off errors.

Ranges, Minimum Accuracies and Maximum Running Times. The accuracies with which probabilities are calculated vary as a function of the Z -score range input and with time constraints written into the program. Generally, the narrower the range, the briefer the running time and the greater the number of decimals to which the probability is calculated. The program utilizes these advantages of narrow ranges by responding to a user-entered Z -score range by selecting the narrowest of the four or five tabled range categories containing that R . It then assigns the computation the associated number of decimal places, maximizing the degree of accuracy for that R . The probabilities computed are always exactly accurate to the number of digits displayed on the screen.

Most computations are quicker than the maximum times listed in Tables 1 and 2, particularly far from $Z=0$. It is the small fourth derivatives that exist in the tails that result in dramatically faster times. For example, 00:23 sec are required to find the area between $Z=-.75$ and $Z=.75$ ($R=1.5$) using NORMLCV.QIK, whereas only 00:11 elapse in computing the area for that same range bounded by $Z=4.5$ and $Z=6$.

Using the Programs. The only substantive difference between the programs is the numbers of decimal places computed and, therefore, their running times. The longer running times of NORMLCV.EXT are accompanied by screen messages informing the user of the maximum running time for the range entered, as indicated in Table 2.

Table 1
Ranges of Numerical Integration, Degrees of Accuracy, and Maximum Running Times for NORMLCV.QIK

Ranges	Accuracy	Maximum Running Time
≤ 1.5	9 places	00:23
≤ 2.5	8 places	00:24
≤ 5.0	7 places	00:33
≤ 12.0	6 places	00:55

Both programs present a menu to the user offering four choices:

1. *A two-tailed Z test plus the μ to Z area.* The execution of this option begins by requesting that the user enter a Z score. In most cases, the area between $Z=0$ and the Z entered is computed and doubled, giving the area from $-Z$ to Z . This area is next subtracted from 1 for the two-tailed probability associated with the Z entered. The program also prints the area from 0 to Z . This computation is relatively fast because usually only the 0 to Z area is directly computed by Simpson's rule.

When $Z > 2.5$ in NORMLCV.QIK or $Z > 3$ in NORMLCV.EXT, Z tests are computed differently. The area in these cases is found between the user-entered Z and $Z=7$ or $Z=8$, respectively, and then it is doubled for the two-tailed test. This is much more rapid than the 0 to Z calculation. The $Z=7$ and $Z=8$ upper limits are used because the person selecting this option desires the area between Z and infinity. Infinity is operationally defined as a point under the normal curve sufficiently far from $Z=0$ that two times the area beyond it is not detectable within 9 decimal places with NORMLCV.QIK or within 12 places with NORMLCV.EXT, and this point exists at $Z=7$ and $Z=8$, respectively.

Persons wanting the area from Z to infinity are advised to use the Z test option.

2. *An area over a range specified by you.* This option requests the user to enter the algebraic values of two Z scores, and the program finds the normal curve area they subtend. The maximum usable Z s are $Z=6$ and $Z=7$, respectively, in NORMLCV.QIK and NORMLCV.EXT. Remember that the area between any specified Z and these upper limits is *not* that between the Z and infinity; that area is determined using the Z test option.

3. *A percentile rank.* The area is found between a user-entered Z and $Z=0$. The area is then either added to .5 if $Z > 0$ or subtracted from .5 when $Z < 0$, and the percentile rank is displayed as a percentage rather than as a fraction.

Table 2
Ranges of Numerical Integration, Degrees of Accuracy, and Maximum Running Times for NORMLCV.EXT

Ranges	Accuracy	Maximum Running Time
≤ 1.5	12 places	02:08
≤ 3	11 places	02:51
≤ 4	10 places	02:18
≤ 8	9 places	03:04
≤ 14	8 places	03:30

4. A two-tailed test of Pearson's r . Fisher (1950) introduced a very useful transform (Z') of the r distribution for significance testing that is quite accurate as long as the degrees of freedom are not small. Z' is very nearly normally distributed with the following standard deviation, where N is the number of score pairs:

$$\sigma Z' = \frac{1}{\sqrt{N-3}}.$$

Because this test uses normal curve areas, it has been included as an option in both programs. A user-entered r is immediately transformed into Z' by the following formula:

$$Z' = .5 \ln \left(\frac{1+r}{1-r} \right).$$

This is next transformed into a Z score as follows:

$$Z = Z' \sqrt{N-3}.$$

A two-tailed test is then performed, and the results are displayed on the screen.

Computer Specifics. The IBM-PC user can respond to the DOS prompt by typing BASIC/D <ENTER> in all versions of DOS from 2.0 forward. Run in this mode, all functions of e , logs, square roots, and other functions processed in double-precision arithmetic are carried to 16-place accuracy. Both programs are written for use with BASIC/D, and only with BASIC/D do they deliver the advertised accuracy. The probabilities displayed on the computer screen are always exactly accurate to the number of decimal places shown. Other systems that use Microsoft BASIC, such as the TRS-80 Model 4, will run the program with only one modification; however, the absence of BASIC/D limits the accuracy of the result.

The computer on which the programs were developed and tested was a stock IBM-PC with 64K RAM. No tests have been made using microcomputers modified by turboboards or special number-processing boards. Users running these routines on a computer other than IBM are advised that line 15 should be changed to the following:

15 PRINT " (1) A TWO-TAILED Z TEST AND THE AREA FROM MU TO Z "

Discussion. Despite the fact that curve-area finding techniques have long been known, probability-finding algorithms of mediocre or lower quality continue to be published. The present article attempts to illuminate and improve that situation by presenting two programs that find normal curve probabilities to high and mathematically guaranteed degrees of accuracy. Full explanations are offered of the process involved, which it is hoped serve as a guide for those planning to publish stochastic quadrature.

The quality of modern personal computers and the

probability-finding algorithms published by Brophy (1983), Moran (1980), Wood (1985), and Wood and Wood (1984, 1986) have eliminated the usefulness of the techniques of Kelley (1947-1948) and others having limited accuracy. Because most programmers writing in FORTRAN or COBOL are also fluent in BASIC, translations can readily be made when programs must be written for a special computer or in a language other than BASIC, so that programs introducing error in the 3rd or 4th decimal place should disappear from use.

The techniques explicated in the two routines are highly flexible and can be modified to give 4-decimal-place probabilities that always compute in 11 sec or less and operate over the range, $Z = -4$ to $Z = 4$. In order to do this, it is necessary to redefine ER and P\$ in line 62 of NORMLCV.QIK to ER=1E-5 and "#####", respectively, and to set the upper limit of integration at $Z=5$ in Z tests. Similar modification will adapt the program to give 15-place accuracy and a range of $Z = -12$ to $Z = 12$.

Availability. A listing of either or both programs can be obtained without charge from the author. A disk copy may be obtained by sending a formatted IBM disk (DOS 2.0 or later) in a returnable box.

REFERENCES

- ABRAMOWITZ, M., & STEGUN, I. A. (Eds.) (1964). *Handbook of mathematical functions*. Washington, DC: U.S. Government Printing Office.
- ANTON, H. (1981). *Calculus*. New York: Wiley.
- BROPHY, A. L. (1983). Accuracy and speed of seven approximations of the normal distribution function. *Behavior Research Methods & Instrumentation*, 15, 604-605.
- EVANS, S., & GILFILLAN, L. (1986). APL approximations for common statistical tables. *Behavior Research Methods, Instruments, & Computers*, 18, 337-338.
- FISHER, R. A. (1950). *Contributions to mathematical statistics*. New York: Wiley.
- JASPEN, N. (1965). The calculation of probabilities corresponding to values of z , t , F and chi square. *Educational and Psychological Measurement*, 25, 877-880.
- KELLEY, T. L. (1947). *Fundamentals of statistics*. Cambridge: Harvard University Press.
- KELLEY, T. L. (1948). *The Kelley statistical tables* (rev. ed.). Cambridge: Harvard University Press.
- LOFTUS, G. R., & LOFTUS, E. F. (1982). *Essence of statistics*. Monterey, CA: Brooks/Cole.
- MORAN, P. A. P. (1980). Calculation of the normal distribution function. *Biometrika*, 67, 675-676.
- O'GRADY, K. E. (1981). Probabilities and critical values for z , chi-square, t , and F . *Behavior Research Methods & Instrumentation*, 13, 55-56.
- VON COLLANI, G. (1983). Computing probabilities for F , t , chi square, and z in BASIC. *Behavior Research Methods & Instrumentation*, 15, 543-544.
- WOOD, D. L. (1985). High precision chi-square and normal curve integration. *Behavior Research Methods, Instruments, & Computers*, 17, 429-431.
- WOOD, D. L., & WOOD, D. (1984). TINT: A Microsoft BASIC t integration program. *Behavior Research Methods, Instruments, & Computers*, 16, 479-480.
- WOOD, D. L., & WOOD, D. (1986). Precise F integration. *Behavior Research Methods, Instruments, & Computers*, 18, 405-406.

APPENDIX A Listing of Program NORMLCV.QIK

```

10 CLS:PRINT"The range of this program is Z=-6 to Z=6.More extreme values are re
jected.":PRINT
11 PRINT"To find the area between a Z and infinity,please select option 1 from t
he menu."
12 DEFDBL A-Z:DEFINT C,J,V,W:DEFSNG D,E,I,M
13 PRINT:S=0:T=0:PR=0:CC=0:PRINT
14 PRINT"Enter your choice by NUMBER:":PRINT
15 PRINT"(1) A TWO-TAILED Z TEST,THE AREA FROM "CHR$(230)" TO Z AND THE AREA FRO
M Z TO INFINITY"
16 PRINT"(2) AN AREA OVER A RANGE SPECIFIED BY YOU"
17 PRINT"(3) A PERCENTILE RANK"
18 PRINT"(4) A TWO-TAILED SIGNIFICANCE TEST FOR PEARSON'S r"
19 PRINT"(5) END WORK":PRINT
20 Z$=INKEY$:IF Z$=""THEN 20 ELSE IF VAL(Z$)<>5 THEN CLS
21 IF VAL(Z$)<1 OR VAL(Z$)>5 THEN CLS:PRINT"PLEASE SELECT 1,2,3,4 OR 5":GOTO 13
22 ON VAL(Z$) GOTO 23,30,37,42,55
23 INPUT"Enter the absolute value of the Z score: ",Z
24 IF ABS(Z)>6 THEN PRINT:GOSUB 56:GOTO 13
25 IF ABS(Z)<=2.5 THEN L=0:U=ABS(Z):D=U ELSE L=ABS(Z):U=7:D=U-L:CC=1
26 GOSUB 60:GOSUB 49
27 PRINT"THE PROBABILITY"SPC(5)"THE AREA FROM 0 TO Z"SPC(5)"THE AREA FROM Z TO I
NFINITY"
28 IF CC<>1 THEN PRINT USING P$;A3:PRINT TAB(25) USING P$;A:PRINT TAB(52) USIN
G P$;.5-A ELSE PRINT USING P$;A*2:PRINT TAB(25)USING P$;.5-A:PRINT TAB(52) USI
NG P$;A
29 PRINT:GOTO 13
30 CLS:INPUT"ENTER THE SMALLER Z SCORE ALGEBRAICALLY: ",L
31 INPUT"ENTER THE LARGER Z SCORE ALGEBRAICALLY: ",U
32 IF ABS(L)>6 OR ABS(U)>6 THEN GOSUB 56:GOTO 13
33 PRINT:D=U-L:IF U<L THEN GOSUB 90:GOTO 13
34 GOSUB 60
35 GOSUB 49:PRINT"NO. SEGMENTS"," AREA WITHIN THE RANGE SPECIFIED"TAB(50)"AREA B
EYOND THE RANGE SPECIFIED"
36 PRINT SPC(3)W SPC(14) USING P$;A:PRINT SPC(23) USING P$;A2:PRINT:GOTO 13
37 PR=1:INPUT"Enter the algebraic value of the Z score: ",Z
38 IF ABS(Z)>6 THEN GOSUB 56:GOTO 13
39 U=ABS(Z):L=0:D=U:PRINT:GOSUB 60:GOSUB 49
40 IF ZZ<0 THEN PR=(.5-H*S/3)*100 ELSE PR=(H*S/3+.5)*100
41 PRINT"The Percentile Rank Of Z =":PRINT USING"###.####";PR:PRINT"%":GOTO 13

42 PRINT"Fisher's Z ' Transformation Requires 4 Or More Score Pairs For r.":PRIN
T
43 INPUT"Enter The Algebraic Value Of r : ",R:PRINT:GOSUB 95
44 INPUT"Enter The Number Of Score Pairs : ",V:PRINT
45 PR=1:ZP=.5*LOG((1+R)/(1-R)):Z=ZP*SQR(V-3)
46 PRINT"Z ' ="ZP TAB(27)"Z(r) ="Z:PRINT
47 U=ABS(Z):L=0:D=U:PRINT:GOSUB 60:GOSUB 49:IF R=0 THEN P$="#.#####"
48 PRINT"The Two Tailed P(r) =" USING P$;A3:PRINT:GOTO 13
49 ZZ=Z:H=(U-L)/W:Z=L:GOSUB 54:S=S+Y:Z=U:GOSUB 54:S=S+Y:V=W/2
50 FOR J=1 TO V:Z=L+(2*J-1)*H:GOSUB 54:T=T+Y:NEXT:S=S+4*T:T=0:V=V-1
51 PRINT"H A L F D O N E !":PRINT
52 FOR J=1 TO V:Z=L+2*J*H:GOSUB 54:T=T+Y:NEXT:S=S+2*T
53 A=ABS(H*S/3):A2=ABS(1-A):A3=ABS(1-2*A):RETURN
54 Y=.3989422804014327#/1.648721270700128#^(Z*Z):RETURN
55 END
56 PRINT"The range of application is Z=-6 to Z=6":RETURN
60 PRINT"C O M P U T I N G !",
61 IF L<=0 AND U>0 OR ABS(L)<.299 OR ABS(U)<.299 THEN M=1.2 ELSE GOSUB 80
62 IF ABS(D)<=1.5 THEN ER=1E-10:P$="#####":GOSUB 70:RETURN
63 IF ABS(D)<=2.5 THEN ER=1E-09:P$="#####":GOSUB 70:RETURN
64 IF ABS(D)<=5 THEN ER=1E-08:P$="#####":GOSUB 70:RETURN
65 IF ABS(D)<=12 THEN ER=.0000001:P$="#####":GOSUB 70:RETURN
70 D2=(M^.25*D^1.25)/(180*ER)^.25
71 D2=D2+1:D2=INT(D2)
72 IF D2/2>INT(D2/2) THEN W=D2+1 ELSE W=D2
73 IF W<6 THEN W=6
74 RETURN
80 E=L:EU=U

```

APPENDIX A (Continued)

```

81 D4=.3989423/(1.6487213#^(E*E))*(E^4-6*E*E+3)
82 M=ABS(D4)
83 FOR I=E+.05 TO EU STEP .1
84 D4=.3989423/(1.6487213#^(I*I))*(I^4-6*I*I+3)
85 IF ABS(D4)>M THEN M=ABS(D4)
86 NEXT
87 RETURN
90 PRINT
91 PRINT"The program requires that the first Z score entered be SMALLER than the
  second one.Please try again."
92 RETURN
95 IF ABS(R)>1 THEN CLS:PRINT"r CANNOT EXCEED 1.":GOTO 13
99 RETURN

```

APPENDIX B
Listing of Program NORMLCV.EXT

```

10 CLS:PRINT"The range of this program is Z=-7 to Z=7.More extreme values are re
  jected.":PRINT
11 PRINT"To find the area between a Z and infinity,please select option 1 from t
  he menu."
12 DEFDBL A-Z:DEFINT C,J,V,W:DEFSNG D,E,I,M:N$="MAXIMUM RUNNING TIME ="
13 PRINT:S=0:T=0:PR=0:CC=0:PRINT
14 PRINT"Enter your choice by NUMBER:":PRINT
15 PRINT"(1) A TWO-TAILED Z TEST,THE AREA FROM "CHR$(230)" TO Z AND THE AREA FRO
  M Z TO INFINITY"
16 PRINT"(2) AN AREA ABOVE A RANGE SPECIFIED BY YOU"
17 PRINT"(3) A PERCENTILE RANK"
18 PRINT"(4) A TWO-TAILED SIGNIFICANCE TEST FOR PEARSON'S r"
19 PRINT"(5) END WORK":PRINT
20 Z$=INKEY$:IF Z$=""THEN 20 ELSE IF VAL(Z$)<>5 THEN CLS
21 IF VAL(Z$)<1 OR VAL(Z$)>5 THEN CLS:PRINT"PLEASE SELECT 1,2,3,4 OR 5":GOTO 13
22 ON VAL(Z$) GOTO 23,30,37,42,55
23 INPUT"Enter the absolute value of the Z score: ",Z
24 IF ABS(Z)>7 THEN PRINT:GOSUB 56:GOTO 13
25 IF ABS(Z)=>3 THEN U=8:L=ABS(Z):D=U-L:CC=1 ELSE L=0:U=ABS(Z):D=U
26 GOSUB 60:GOSUB 49
27 PRINT"THE PROBABILITY"SPC(5)"THE AREA FROM 0 TO Z"SPC(5)"THE AREA FROM Z TO I
  NFINITY"
28 IF CC=1 THEN PRINT USING P$;A*2:PRINT TAB(24) USING P$;.5-A:PRINT TAB(51) U
  SING P$;A ELSE PRINT USING P$;A3:PRINT TAB(24) USING P$;A:PRINT TAB(51) USING
  P$;.5-A
29 PRINT:GOTO 13
30 CLS:INPUT"ENTER THE SMALLER Z SCORE ALGEBRAICALLY: ",L
31 INPUT"ENTER THE LARGER Z SCORE ALGEBRAICALLY: ",U
32 IF U<L THEN GOSUB 98:GOTO 13
33 IF ABS(L)>7 OR ABS(U)>7 THEN GOSUB 56:GOTO 13
34 PRINT:D=U-L:GOSUB 60:GOSUB 49
35 PRINT"NO. SEGMENTS"," AREA WITHIN THE RANGE SPECIFIED"TAB(50)"AREA BEYOND THE
  RANGE SPECIFIED"
36 PRINT SPC(3)W SPC(14) USING P$;A:PRINT SPC(23) USING P$;A2:PRINT:GOTO 13
37 PR=1:INPUT"Enter the algebraic value of the Z score: ",Z
38 IF ABS(Z)>7 THEN GOSUB 56:GOTO 13
39 U=ABS(Z):L=0:D=U:PRINT:GOSUB 60:GOSUB 49
40 IF ZZ<0 THEN PR=(.5-H*S/3)*100 ELSE PR=(H*S/3+.5)*100
41 PRINT"The Percentile Rank Of Z =":PRINT USING"###.#####";PR:PRINT"%":GOT
  O 13
42 PRINT"Fisher's Z ' Transformation Requires 4 Or More Score Pairs For r.":PRIN
  T
43 INPUT"Enter The Algebraic Value Of r : ",R:PRINT:GOSUB 110
44 INPUT"Enter The Number Of Score Pairs : ",V:PRINT
45 PR=1:ZP=.5*LOG((1+R)/(1-R)):Z=ZP*SQR(V-3):IF ABS(Z)>7 THEN PRINT"The r (Z="US
  ING"###.#####";Z:PRINT") is beyond the range of the program.":PRINT:GOSUB 56:GOT
  O 13
46 PRINT"Z '="ZP TAB(27)"Z(r)="Z:PRINT
47 U=ABS(Z):L=0:D=U:PRINT:GOSUB 60:GOSUB 49:IF R=0 THEN P$="#.#####
48 PRINT"The Two Tailed P(r)=" USING P$;A3:PRINT:GOTO 13
49 ZZ=Z:H=(U-L)/W:Z=L:GOSUB 54:S=S+Y:Z=U:GOSUB 54:S=S+Y:V=W/2
50 FOR J=1 TO V:Z=L+(2*J-1)*H:GOSUB 54:T=T+Y:NEXT:S=S+4*T:T=0:V=V-1

```

APPENDIX B (Continued)

```

51 PRINT "H A L F   D O N E !":PRINT
52 FOR J=1 TO V:Z=L+2*J*H:GOSUB 54:T=T+Y:NEXT:S=S+2*T
53 A=ABS(H*S/3):A2=ABS(1-A):A3=ABS(1-2*A):RETURN
54 Y=.3989422804014327#/EXP(Z*Z/2):RETURN
55 END
56 PRINT "The range of application is Z=-7 to Z=7":RETURN
60 PRINT "C O M P U T I N G !" SPC(4)
61 IF L<=0 AND U=>0 OR ABS(L)<.299 OR ABS(U)<.299 THEN M=1.2 ELSE GOSUB 90
62 IF ABS(D)<=1.5 THEN C=1:ER=1E-13:P$=".:#####":GOSUB 80:GOSUB 70:RETURN
63 IF ABS(D)<=3 THEN C=2:ER=1E-12:P$=".:#####":GOSUB 80:GOSUB 70:RETURN
64 IF ABS(D)<=4 THEN C=3:ER=1E-11:P$=".:#####":GOSUB 80:GOSUB 70:RETURN
65 IF ABS(D)<=8 THEN C=4:ER=1E-10:P$=".:#####":GOSUB 80:GOSUB 70:RETURN
66 IF ABS(D)<=14 THEN C=5:ER=1E-09:P$=".:#####":GOSUB 80:GOSUB 70:RETURN
70 ON C GOTO 71,72,73,74,75
71 PRINT N$"02:08 "SPC(4):RETURN
72 PRINT N$"02:51 "SPC(4):RETURN
73 PRINT N$"2:18 "SPC(4):RETURN
74 PRINT N$"3:04 "SPC(4):RETURN
75 PRINT N$"03:30 "SPC(4):RETURN
80 D2=(M^.25*D^1.25)/(180*ER)^.25
81 D2=D2+1:D2=INT(D2)
82 IF D2/2>INT(D2/2) THEN W=D2+1 ELSE W=D2
83 IF W<6 THEN W=6
84 RETURN
90 E=L:EU=U
91 D4=.3989423/(1.6487213#^(E*E))*(E^4-6*E*E+3)
92 M=ABS(D4)
93 FOR I=E+.05 TO EU STEP .1
94 D4=.3989423/(1.6487213#^(I*I))*(I^4-6*I*I+3)
95 IF ABS(D4)>M THEN M=ABS(D4)
96 NEXT
97 RETURN
98 PRINT
99 PRINT "The program requires that the first Z score entered be SMALLER than the
    second. Please check the Z's."
100 RETURN
110 IF ABS(R)>1 THEN CLS:PRINT "r CANNOT EXCEED 1.":GOTO 13
115 RETURN

```

(Revision accepted for publication January 30, 1987.)