COMPUTER ROUTINES
FOR PROBABILITY DISTRIBUTIONS,
RANDOM NUMBERS, AND RELATED FUNCTIONS

By W.H. Kirby

1983

UNITED STATES DEPARTMENT OF THE INTERIOR

WILLIAM P. CLARK, Secretary

GEOLOGICAL SURVEY

Dallas L. Peck, Director

CONTENTS

CONTENTS (continued)

COMPUTER ROUTINES FOR PROBABILITY DISTRIBUTIONS, RANDOM NUMBERS, AND RELATED FUNCTIONS

By W. H. Kirby

ABSTRACT

Use of previously coded and tested subroutines simplifies and speeds up program development and testing. This report presents routines that can be used to calculate various probability distributions and other functions of importance in statistical hydrology.

INTRODUCTION

This report describes computer subprograms that compute several probability distributions of importance in statistics and hydrology. Several other routines performing closely related functions such as random number generation, sorting, and plotting also are described. Various versions of these routines have been compiled from various sources and have been used in a variety of programs over a period of several years. Use of these previously coded and tested routines helps to simplify and speed up program development and testing. For this reason the routines have been collected here in a common format to improve their usability and accessibility to others.

The routines are written in Fortran. Versions are operational on the U.S. Geological Survey (USGS) Prime[1] computer (Fortran 77) and on the Amdahl computer (Fortran IV) in Reston, Va. There are only trivial differences between the Prime and Amdahl versions, so it is expected that these routines could be transported easily to other machines as well. Instructions for obtaining source code listings and machine-readable copies of these routines are given in the next two sections of this report, and instructions for calling the routines from your own Fortran program are given in the section after that.

General information about the definition and applications of various probability distributions is given by Benjamin and Cornell (1970) and by Mood, Graybill, and Boes (1974). Computational formulas and approximations are summarized by Zelen and Severo (1964). The mathematical and computational details of all commonly used distributions are exhaustively treated by Johnson and Kotz (1970).

It is believed that the routines presented in this report perform with reasonable efficiency substantially as described. No guarantees can be given, however, regarding accuracy, reliability, or speed. Users performing critical computations, particularly at extreme values of the function parameters and arguments, are encouraged to obtain a copy of the latest version of the routine

---

[1]/The use of brand names in this report is for identification only and does not imply endorsement by the U.S. Geological Survey.

and perform their own tests. The author would be grateful to receive reports
of any such tests as well as reports of errors or deficiencies in any of the
routines.

SOURCE CODE AVAILABILITY

The Fortran source code for these routines is stored on the USGS Re-1
Amdahl computer in two online cataloged data sets named

VG48DST.XLIB.IBM.FORT   (IBM Fortran IV)

and

VG48DST.XLIB.PRIME.FORT   (Prime Fortran 77).

Each of these data sets contains about 3,500 80-column card images. The IBM
data set contains some object-code and assembler-language decks intermingled
with the Fortran code. In both data sets the routines are stored in an order
dictated by the Prime subroutine library conventions.

The data sets may be accessed by means of WYLBUR, TSO, or batch programs
such as the IBM utilities. The following batch job may be used to punch either
data set into cards or card images:

```
//...  JOB(...)   (job card)
/*JOBPARM  L=6,C=1000
//PROCLIB  DD   DISP=SHR,DSN=VG48DST.PROCLIB
//         EXEC CARDPCH,UDS='VG48DST.XLIB.xxx.FORT'
//
```

in which xxx must be replaced by IBM or PRIME. The resulting card images may
be received and stored on magnetic media at suitably equipped remote terminals.
A listing may be produced along with the punched cards by adding the parameter
LIST=1 to the EXEC card. Sequence numbers may be written in card columns 73-80
by adding the parameter SEQ=1 to the EXEC card. A listing without any punched
cards may be obtained by changing the procedure name from CARDPCH to CARDPRT
(and omitting the LIST and SEQ parameters) on the EXEC card.

LINKAGE INFORMATION

These routines have been compiled and stored in subroutine libraries on
the Prime computer in Reston and on the Re-1 Amdahl system. The user has
only to call the routines as explained in the next section, and to make the
appropriate subroutine library available at linkage-editing time, and the
necessary routines automatically will be included in the program. It is not
necessary to retrieve or recompile the source code in order to use these
routines.

On the Prime computer the routines were compiled with the Fortran-77
compiler F77. They are stored in a binary library named XLIB77. They may
be made available to your Fortran program by the following commands:

2

```
F77      your.program
SEG      -LOAD
LO       your.program
LI       XLIB77
LI
SA
Q
```

On the Amdahl Re-1 system, the routines were compiled with the IBM Fortran IV H-Extended compiler. They are stored in a cataloged load-module library named VG48DST.XLIBXX. They may be made available to your Fortran program by means of the ULIB parameter provided with the Fortran procedures described in chapter 7 of the USGS Computer Users' Manual (unpublished documentation available from the USGS Computer Center Division, User Services). Typical compile-load-and-go usage is as follows:

```
//  EXEC  FORTXCG,ULIB='VG48DST.XLIBXX'
    your  Fortran program

//GO.SYSIN  DD  *
    your   data
```

Any other parameters needed on the EXEC card may be placed before or after the ULIB parameter.

CALLING INSTRUCTIONS

The following paragraphs contain instructions for calling the routines from Fortran programs. Each paragraph contains the name of the routine, a typical argument list, and a brief description of the function performed. Familiarity with the general mathematical character of the functions is assumed, as is familiarity with the Fortran language. In particular, variable names beginning with the letters I-N refer to integer data and names beginning with A-H or O-Z refer to real-valued data. Detailed discussion of computational methods, limitations on arguments, handling of error conditions, and performance generally is not attempted. The references and source listings may be consulted for this information if necessary.

BESFI0(X) - Function returns the value of the modified Bessel function I_0 ("I-zero") of order zero at argument x. Source--IBM (1970) SSP routine I0.

BETAP(X,A,B) - Function returns the value of the incomplete beta function

$$I_X(a,b) = \frac{1}{B(a,b)} \int_0^X u^{a-1}(1-u)^{b-1} \, du.$$

This value also is the probability that the beta-distributed random variable with parameters a and b will be less than or equal to X. Restriction--The parameters a and b must be strictly positive; otherwise, BETAP will be set to a large negative number. Source--Stanford University Computation Center (unpublished library program C060, 1969).

3

CHISQP(X,N) - Function returns the probability that the chi-square random variable with n degrees of freedom will be less than or equal to x, $P\{\chi_n^2 \le x\}$. Reference--Hill and Pike (1967).

CHISQX(P,N) - Function returns the p-th quantile of the chi-square random variable with n degrees of freedom. That is, it returns the solution for x of the equation $P\{\chi_n^2 \le x\}$ = p for the given value of p. Reference--Goldstein (1973).

DATME(IALF) - Subroutine looks up the current date and time of day and stores them as alphanumeric characters in the 5-fullword integer array IALF in the format mm/_dd/_yy__hh:_mm__ (month/day/year hour: minute), which may be printed under the format (IX,5A3).

DGAMMA(X) - Double precision gamma function of double precision argument X. DGAMMA and X must be declared double precision in the calling program. See GAMMA.

DLGAMA(X) - Double precision function returns the natural logarithm of the gamma function $\Gamma(X) = \int_0^\infty t^{x-1}e^{-t}\, dt$. Restrictions--X must be greater than 10^{-9} and less than 10^{35}; otherwise, DLGAMA is set to 10^{38}. X and DLGAMA must be declared double precision in the calling program. Source-- IBM (1970) SSP routine DLGAM.

ERF(X) - Function returns the value of the error function $(2/\sqrt{\pi})\int_0^x e^{-u^2}\, du$. Source--IBM (1970) SSP routine NDTR.

EXPIF(X) - Function returns the value of the exponential integral $\int_x^\infty (e^{-t}/t)\, dt$, variously denoted as $Ei(x)$, $-Ei(-x)$, and $E_1(x)$. It is also known as the well function, W(u), for the Theis-Jacob-Wenzel nonequilibrium artesian well. The value of x may be positive or negative; EXPIF is set to a large positive number when x=0. Source--IBM (1970) SSP routine EXPI.

GAMMA(X) - Function returns the value of the gamma function, defined for x>0 by

$$\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\, dt$$

and extended to other x by the recursion formula $\Gamma(x+1) = x\,\Gamma(x)$. Restriction--x must be less than 34.0 and not within 10^{-6} of a negative integer or 0.0; otherwise, GAMMA is set to 10^{38}. The IBM FORTRAN IV library version of GAMMA requires positive values of x. Source--IBM (1970) SSP routine GMMMA.

GAMMAP(A,X) - Function returns the value of the incomplete gamma function
ratio

$$\gamma(a,x) = \int_0^x t^{a-1} e^{-t} \, dt / \Gamma(a), \qquad (a>0)$$

which also is the probability that the gamma-distributed random variable
with shape factor a is less than or equal to x. Note--The gamma is a
member of the Pearson Type III family of distributions. It has mean a,
standard deviation $\sqrt{a}$, skewness $2/\sqrt{a}$, and lower bound 0. See also
HARTIV and WILFRT. Restriction--The shape factor a must be greater than
0; otherwise, the distribution is concentrated at x=0. Source--IBM (1970)
SSP routine CDTR.

GAUSAB(P) - Function returns the p-th quantile (or Gaussian abscissa) of the
Gaussian or standardized normal distribution. This value also is known
as the standardized normal deviate with non-exceedance probability p. It
is the solution for x of the equation

$$P\{Z \le x\} = (1/\sqrt{2\pi}) \int_{-\infty}^x e^{-z^2/2} \, dz = p$$

for the given value of p, where Z is the standardized normal random
variate. Restriction--If p is less than or equal to 0, GAUSAB is set to
-10; if p is greater than or equal to 1, GAUSAB is set to +10. Reference--
Zelen and Severo (1964).

GAUSCF(X) - Function returns the value of the Gaussian or standard normal
cumulative probability function at x, as follows:

$$P\{Z \le x\} = (1/\sqrt{2\pi}) \int_{-\infty}^x e^{-z^2/2} \, dz$$

where Z is the standardized normal random variate. Reference--Zelen and
Severo (1964).

GAUSDY(X) - Function returns the value of the Gaussian or standard normal
density, $(1/\sqrt{2\pi}) e^{-x^2/2}$, at x.

GAUSEX(PEX) - Function returns the Gaussian or standard normal deviate with
exceedance probability (tail probability) PEX. It is the solution for x
of the equation

$$P\{Z > x\} = (1/\sqrt{2\pi}) \int_x^\infty e^{-z^2/2} \, dz = P_{ex}$$

for the given value of P_{ex}, where Z is the standardized normal random
variate. Restriction--If P_{ex} is greater than or equal to 1.0, GAUSEX
is set to -10. If P_{ex} is less than or equal to 0.0, GAUSEX is set to +10.
Reference--Zelen and Severo (1964).

5

GAUSSF(IRAN) - Function returns a quasi-random number drawn from the standard normal (Gaussian) distribution. IRAN is the "seed" of the random number generator; it must be initialized by the user but thereafter is updated automatically by the generator. The initial IRAN may be any value between 1 and 2147483646. References--Box and Muller (1958), Lewis and others (1969).

GAUSSV(IRAN,X,N) - Subroutine places a quasi-random sample of size N in vector X. The sample is drawn from the standard normal (Gaussian) distribution. IRAN is the "seed" of the random number generator; it must be initialized by the user but thereafter is updated automatically by the subroutine. The initial IRAN may be anywhere between 1 and 2147483646. References--Box and Muller (1958), Lewis and others (1969).

GUMBEL(P) - Function returns the value of the standardized deviate with non-exceedance probability P from the Gumbel Type I (double-exponential) extreme-value distributions. For a general Gumbel variate with mean m and standard deviation s, the p-th quantile is

$$X_p = m + s*GUMBEL(p).$$

Reference--Benjamin and Cornell (1970).

GUMBEP(X) - Function returns the value of the cumulative probability distribution function of the standardized Gumbel Type I (double-exponential) extreme-value random variate. If Y is a general Gumbel variate with a mean m and standard deviation s, then

$$P\{Y \le x\} = GUMBEP ((x-m)/s).$$

Reference--Benjamin and Cornell (1970).

HARTIV(SKEW,RIPS) - Subroutine looks up percentage points of the standardized Pearson Type III distribution with skew coefficient SKEW in Harter's (1969, 1971) tables. Interpolation is done on the skew coefficient at all 31 tabular probabilities considered in Harter's two papers. The resultant interpolated percentage points are placed in the 31-word vector RIPS in increasing order. Restriction--The skew must not exceed 9.0 in absolute value; otherwise, RIPS is filled with zeros. References--Harter (1969, 1971), U.S. Water Resources Council (1977).

HARTK(P,RIPS) - Function looks up the p-th quantile (standardized deviate with non-exceedance probability p) of the standardized Pearson Type III distribution tabulated in the vector RIPS. The vector RIPS must have been filled by a prior call to HARTIV. Restriction--The value of p must be between 0.0001 and 0.9999; otherwise, HARTK will be set to a large number (positive if p>0.9999, negative if p<0.0001). Reference--see HARTIV.

HARTKK(Z,RIPS) - Function looks up that quantile of the standardized Pearson Type III distribution tabulated in the vector RIPS that has the same non-exceedance probability as the given standard normal deviate z. That is, HARTKK solves for x in the equation

$$P\{ R \leq x\} = P\{ \underline{Z} \leq z\}$$

where $\underline{Z}$ is the standard normal random variable and R is the standard Pearson Type III random variable tabulated in RIPS. The vector RIPS must have been filled by a call to HARTIV. Restriction--The value of z must not exceed 3.719 in absolute value; otherwise, HARTKK will be set to a large value of the same sign as z. Reference--see HARTIV.

HARTP(Z,RIPS) - Function looks up the cumulative (non-exceedance) probability of the given value Z in the standardized Pearson Type III distribution tabulated in the vector RIPS. The vector RIPS must have been filled by a prior call to HARTIV. If z is outside the range of the table, HARTP is set to 0 (if z<0) or 1 (if z>0). Reference--see HARTIV.

HARTRG(QR) - Function returns the value of the skew coefficient corresponding to the given value of the 2-10-100-quantile ratio, $QR = (X_{100}-X_{10})/(X_{10}-X_2)$, in which X_T is the "T-year" quantile of a Pearson Type III distribution (having exceedance probability 1/T). The results are obtained from empirical polynomial and semilog formulas that correlate skew-values and quantile-ratios from Harter's (1969, 1971) tables.

HARTTP(IORDER,PROBV) - Subroutine copies the 31 standard tabular probabilities from Harter's (1969, 1971) tables into the user's 31-word Vector PROBV. If IORDER is negative, the probabilities are stored in decreasing order as exceedance probabilities; otherwise, in increasing order. Reference--see HARTIV.

MINV (A,N,DET,IWORK,JWORK) - Subroutine inverts the NxN matrix A and returns the value of the determinant in DET. The original contents of A are destroyed and replaced by the inverse. A singular A-matrix yields a DET value of zero. IWORK and JWORK are integer work vectors of dimension N. Source--IBM (1970) SSP routine MINV.

OUTKGB(SIG,N) - Function returns the value of the Grubbs-Beck (1972) one-sided single-outlier criterion for normal samples of size N at significance level SIG. That is, OUTKGB is the solution for x of the equation

$$P\{ (X_{max} - \overline{x})/s > x\} = SIG$$

where X_{max} is the maximum of a sample of N normal random variates, $\overline{x}$ and s are the sample mean and standard deviation $(s=\{\Sigma(x-\overline{x})^2/(N-1)\}^{1/2})$, and SIG is the given significance level, expressed as either a percentage or a decimal fraction. This is the high-outlier criterion; the low-outlier criterion is just the negative of this: that is, -OUTKGB. Restrictions-- The sample size N must be at least 3. The available significance levels are 1., 2.5, 5., and 10. percent (or 0.01, 0.025, 0.05, and 0.10). Otherwise, OUTKGB will be set to a large negative number. Note--Piecewise linear approximation is used for N above 100 and OUTKGB is constant for N above 180. Reference--Grubbs and Beck (1972).

PRPLOT - Subroutine produces "printer-plotted" graphs on the line printer.
Graphs are produced in three phases by calling three entry points, as
follows:

PLOT2(DUMMY,XMAX,XMIN,YMAX,YMIN) - Blanks out an internal plot-
image area and then fills it with a rectangular coordinate grid.
The standard grid has a vertical (Y) axis 51 print lines high and
a horizontal (X) axis 101 print columns wide. Five horizontal
grid lines and 10 vertical ones are drawn at 10-space intervals
in each direction. The axis annotations at the grid lines are
printed with three decimal places. Nonstandard grids can be
defined by PLOT1: see the reference for details. Subroutine
PRPSCL, described below, can be used to make the axis annotations
come out as "even" numbers. The argument DUMMY must be supplied
for compatibility with earlier versions of the PRPLOT package,
but it is not used in any way by the subroutine.

PLOT3(SYMBOL,X,Y,N) - Plots the character SYMBOL at the N locations
defined by the vectors X and Y. The SYMBOLS are plotted in the
internal plot-image area initialized by PLOT2. SYMBOL may be
either a literal constant or a CHARACTER*1(Prime) or LOGICAL*1(IBM)
variable or array. If SYMBOL contains more than one character,
only the leftmost one is plotted. X-Y coordinates outside the
limits of the grid are ignored. Several curves can be drawn on
one graph by repeated calls to PLOT3. When several points occupy
the same grid position, the first-plotted points are obliterated;
only the last-plotted point shows.

PLOT4(N,LABEL) - Prints the internal plot-image area and scale anno-
tations for the X and Y axes. The N-character string LABEL is
printed as a label for the vertical (Y) axis. The standard grid
uses 53 print lines, including the X-axis annotation. The user
is responsible for ejecting the page and printing page headings,
captions, and the X-axis label. PLOT4 does not destroy the plot
area, so additional curves may be plotted on the same graph, if
desired, by additional calls to PLOT3 and then printed by PLOT4.
Calling PLOT2 reinitializes the plotting grid.

Probability plots (normal probability grid) may be made by calling PLOT2P
and PLOT4P instead of PLOT2 and PLOT4. The horizontal (X) axis is used for
cumulative probabilities running from 0.1 percent at the left to 99.9 percent
at the right. Probabilities outside this range are ignored. The user must
convert the cumulative probabilities to the corresponding standard normal
deviates; X = GAUSAB(PCUM) may be used for this purpose. Then PLOT3 is used
in the usual way to plot the transformed cumulative probabilities/deviates on
the X scale and the data observations on the Y scale. The necessary subroutine
calls are as follows:

PLOT2P(DUMMY,YMAX,YMIN) - Initializes the PRPLOT internal plot area
with a standard normal probability grid. The argument DUMMY must be
supplied for compatibility with previous versions of the program, but
is not used in any way by the program.

8

PLOT4P(N,LABEL) - Prints the plot area and scale annotations. The N-character string LABEL is printed on the vertical axis and a cumulative probability label is printed under the X axis. Fifty-four lines are printed. The user is responsible for page ejects, titles, etc.

A similar pair of routines, PLOT2G and PLOT4G, is available for plotting on Gumbel extreme-value probability coordinates. Cumulative probabilities must be transformed to standardized Gumbel deviates by the formula X = GUMBEL(PCUM). Then PLOT3 is used to plot the transformed cumulative probabilities on the X (horizontal) scale and the data observations on the Y (vertical) scale. Plotting the logarithms of the data gives a straight-line fit for a Weibull population. The necessary subroutine calls are as follows:

PLOT2G(DUMMY,YMAX,YMIN) - Sets up the standard Gumbel probability grid in the internal plot-image area. The argument DUMMY is required for compatibility with an earlier version of the subroutine but is not used in any way by the subroutine.

PLOT4G(N,YLABEL) - Prints the plot area and scale annotations. The character string YLABEL, of length N, is printed on the Y (vertical) axis and a cumulative probability label is printed under the X (horizontal) axis. Fifty-four lines are printed. The user is respon-sible for printing captions, starting on a fresh page, etc.

Finally, routines PLOT2L and PLOT4L provide a logarithmic probability grid for plotting exponentially distributed data or the logarithms of Pareto-distributed data. Before plotting, cumulative probabilities must be trans-formed to exponential variates by the formula X = -ALOG(1.-PCUM). Then PLOT3 is used to plot the transformed probabilities on the X (horizontal) scale and the data on the Y (vertical) scale. As above, PLOT2L sets up the grid, PLOT3 plots the data, and PLOT4L prints the graph and scale annotations. The calling sequences are:

PLOT2L(DUMMY,YMAX,YMIN)

PLOT4L(N,YLABEL)

in which all arguments are as above. PLOT4L prints 54 lines, including a probability label under the horizontal axis. The user is responsible for page ejects, titles, etc.

The scale annotations may be forced to come out as "even" values by the following subroutine call:

PRPSCL(U1,U2,NGRID,P1,P2) - Subroutine computes "pretty" endpoints P1, P2 for a scale with NGRID grid marks for plotting data between the "ugly" values U1 and U2.

Reference--unpublished documentation for USGS program number B524 (PRPLOT), available from USGS Computer Center Division, User Services. PLOT1 - PLOT4 are part of program B524 (PRPLOT); the probability-plot routines make use of PRPLOT but are not part of program B524; PRPSCL is an independent subroutine. The present version of PRPLOT has been revised to use an internal plot-image

9

area of size 61 x 121 (7,381) characters, the size of a full printed page, rather than an area defined in the calling program. The revised PRPLOT nonetheless is compatible with the original one, so existing programs need not be changed, with the following exceptions:

1. Multi-page plots, with image areas larger than 7,381 characters, are not supported.

2. The calling program cannot inspect or modify the plot image area (except by calling PRPLOT).

3. Plot-image definitions can be removed from the calling program to save space, but do not have to be removed.

RANDUB(IRAN) - Function returns a quasi-random number drawn from the uniform distribution on the unit interval. IRAN is the "seed" of the random number generator; it must be initialized by the user but thereafter is updated automatically by the generator. The initial value of IRAN may be between 1 and 2147483646. Note--This generator is the Lewis-Goodman-Miller (1969) generator designed for the IBM/360 computer.

RANDUV(IRAN,X,N) - Subroutine places a quasi-random sample of size N from the unit uniform (0,1) distribution into the vector X. IRAN is the "seed" of the random number generator; it must be initialized by the user but there-after is updated automatically by the generator. The initial value of IRAN may be between 1 and 2147483646. Note--This generator is the Lewis-Goodman-Miller (1969) generator designed for the IBM/360 computer.

SMIRP(X) - Function returns the large-sample limiting probability distribution of the Kolmogorov and Smirnov statistics, as follows:

$$P\{\sqrt{n}D_n \leq x\} \qquad \text{(Kolmogorov 1-sample)}$$

$$P\{\sqrt{(n+m)/nm}\ D_{n,m} \leq x\} \qquad \text{(Smirnov 2-sample)}$$

where

$$D_n = \max_x |F_n(x) - F(x)|$$

$$D_{n,m} = \max_x |F_n(x) - F_m(x)|$$

and F_n, F_m are empirical (sample) distribution functions of sizes n and m and F is the corresponding hypothetical distribution. Source--IBM (1970) subroutine SMIRN.

SNCTPA(X,N,D) - Function returns an approximate value of the Student non-central t probability $P\{t_{N,D} \leq X\}$ with N degrees of freedom and non-centrality parameter D. When D=0, the noncentral t becomes the ordinary Student t. The approximation is intended for N greater than say, 20, but retains some usefulness down to N equal 5 or 10. Reference--Zelen and Severo (1964, eqn. 26.7.10).

10

SNCTXA(P,N,D) - Function returns an approximate value of the Student noncentral t quantile with N degrees of freedom, noncentrality parameter D, and non-exceedance probability P. That is, it is the solution for x of the equation $P\{t_{N,D} \leq x\} = P$. The approximation is intended for large N but retains some value for N as small as 5 or 10. Reference--Zelen and Severo (1964, eqn. 26.7.10).

SNEFPB(X,N1,N2) - Function returns the probability that the Snedecor F (or variance-ratio) random variable is less than or equal to the given value of x. That is,

$$\text{SNEFPB} = P\{F_{N1,N2} \leq x\}$$

where $F_{N1,N2} = S_1^2/S_2^2$ is the F or variance-ratio statistic with N_1 degrees of freedom in the numerator and N_2 in the denominator. SNEFPB is an entry point of subprogram BETAP (which see).

SORT(X,N) - Subroutine rearranges the N real numbers in the vector X into increasing order. If the value of N is negative (SORT(X,-N)), the numbers are sorted in decreasing order. This routine uses the N-log-N Quicksort algorithm. It is about 3 times as fast as a simple bubble sort for N=25 and about 15 times as fast for N=250. Source--Stanford University Computation Center (unpublished program C063, 1971).

SORTP(X,IPOINT,N) - Subroutine sorts the N elements of the array X into increasing numerical order while maintaining pointers to the original positions of the sorted data items. If N is specified as a negative number, SORTP(X,IPOINT,-N), then X is sorted into decreasing order. In either case, IPOINT is an integer vector computed such that its i-th element, IPOINT(i), equals the original (input) index in X of the data element that sorts into the i-th location on output. The IPOINT values thus can be used to refer to the values of variables associated with sorted values of X, as follows: If before sorting X(i) and Y(i) are correlated, then after sorting X(i) and Y(IPOINT(i)) are correlated. The following example further illustrates the operation of SORTP:

X(input)	4.2	1.3	8.7	9.2	2.1
X(output)	1.3	2.1	4.2	8.7	9.2
IPOINT	2	5	1	3	4

STAT2(X,N,XBAR,STDDEV) - Subroutine computes the sample mean and standard deviation of the N observations stored in the vector X. STDDEV is formed by taking the square root of the sum of squares of deviations from XBAR divided by N-1. (See STAT3 for formulas.) If N is less than 2, STDDEV is set equal to zero; if N is less than 1, XBAR also is set to zero.

STAT3(X,N,A,S,G) - Subroutine computes the sample mean, standard deviation, and skew coefficient (A, S, and G) of the N real numbers in the vector X. The sample statistics are defined as follows:

$$A = \Sigma X/N \qquad S = \left\{ \Sigma (X-A)^2/(N-1) \right\}^{1/2}$$

$$G = \left\{ N\Sigma (X-A)^3 \right\} / \left\{ (N-1)(N-2)S^3 \right\}$$

The sample size N must be at least 3 to define the skew, 2 to define the standard deviation, and 1 to define the mean. If the sample is too small to define any of these statistics (or if N is negative), the undefined statistics are set equal to zero; no error messages or warnings are issued.

STUTP(X,N) - Function returns the probability that Student's t with n degrees of freedom is less than or equal to x, $P\{ t_n \leq x\}$. Note--Tail probabilities for two-sided t tests can be computed as follows:

$$P\{ |t_n| > x\} = 2.*STUTP(-x,n)$$

for $x \geq 0$. Reference--Hill (1970a).

STUTPB(X,N) - Function returns the probability that Student's t with n degrees of freedom is less than or equal to x. STUTPB performs the same function as STUTP, but is provided automatically as part of the BETAP routine for the beta distribution (which see).

STUTX(P,N) - Function returns the p-th quantile of Student's t with n degrees of freedom. That is, it solves for x the equation $P\{ t_n \leq x\} = p$ for the given value of p. Note--Two-sided t-criteria can be computed as follows: $P\{ |t_n| > x\} = a$ has the solution x = STUTX(1.-a/2.,n). Reference--Hill (1970b).

WCFGSM(FLAT,FLON) - Function returns the value of the generalized skew coefficient shown on Plate 1 of the U.S. Water Resources Council's "Guidelines for Determining Flood Flow Frequency" (Hydrology Committee Bulletin 17, 1976, or 17A, 1977) at latitude FLAT and longitude FLON. FLAT and FLON must be expressed in decimal degrees so that, for example, 45°30' would have to be expressed as 45.5°. For points outside the conterminous United States, Alaska, and Hawaii, the function returns a large negative number.

WEIBUL(SKU,P) - Function returns the p-th quantile of a standardized Weibull variate with skew SKU. Thus, it returns the value $(W_p-EW)/s_w$, in which EW and s_w are the mean and standard deviation of the Weibull variate W and W_p is the solution for x of the equation

$$P\{ W \leq x\} = 1 - \exp(-x^c) = p$$

for any probability p. In this equation, c is the Weibull shape factor, which depends on the skew; the program automatically calculates c, EW, and s_w each time the skew coefficient is changed. Restrictions--The skew coefficient must be less than 100 in absolute value; otherwise, 100 is used. The skew decreases to zero as the shape factor c increases to

about 3.6. Negatively skewed distributions are obtained by reflecting the positively skewed distribution, not by increasing c beyond 3.6. Reference—Benjamin and Cornell (1970).

WILFRT(SKU,ZETA) - Function returns an (approximately) Pearson Type III standardized variate with skew SKU and probability corresponding to that of the given standard normal deviate ZETA. That is, the value returned is an approximate solution for x of the equation

$$P\{W \leq x\} = P\{Z \leq ZETA\}$$

where Z and W are respectively the standard normal and Pearson Type III variates. Specifically, WILFRT(SKU,GAUSAB(P)) is an approximation to the p-th quantile (x such that $P\{W \leq x\}$ = p) of the Pearson Type III distribution with skew SKU. Restriction—The skew must not exceed 9.75 in absolute value; otherwise, $\pm$9.75 will be used. Note—The approximation is an improved Wilson-Hilferty (cube-root-normal) transformation which matches the mean, standard deviation, skew, and lower bound of the Pearson Type III distribution. The approximation is excellent for skews below 2 in absolute value and is useful throughout the defined range of skews. Reference—Kirby (1972).

WILFRV(SKU,ZETAV,N) - Subroutine transforms the N-element vector ZETAV from standard normal deviates to (approximate) Pearson Type III standardized deviates with skew SKU. The vector ZETAV must be filled with standard normal deviates before calling WILFRV. GAUSAB or GAUSSV may be used for this purpose. WILFRV is a vectorial version of WILFRT (which see).

XETIME(0.) - Function returns the total amount of execution time (CPU time) used by the program in seconds. The timer is started by the first call to XETIME (which returns a value of 0.0), and all subsequent times are measured relative to this origin. To ensure proper operation of the function, its argument always should be specified as 0.0.

REFERENCES

Benjamin, J. R., and Cornell, C. A., 1970, Probability, statistics, and decision for civil engineers: New York; McGraw-Hill, 684 p.

Box, G. E. P., and Muller, M. E., 1958, A note on the generation of random normal deviates: Annals of Mathematical Statistics, v. 29, no. 2, p. 610-611.

Goldstein, R. B., 1973, Algorithm 451, Chi-square quantiles: Communications of the Association for Computing Machinery, v. 16, no. 8, p. 483-485.

Grubbs, F. E., and Beck, G., 1972, Extension of sample sizes and percentage points of outlying observations: Technometrics, v. 14, no. 4, p. 847-854.

Harter, H. L., 1969, A new table of percentage points of the Pearson Type III distribution: Technometrics, v. 11, no. 1, p. 177-187.

Harter, H. L., 1971, More percentage points of the Pearson Type III
    distribution: Technometrics, v. 13, no. 1, p. 203-204.

Hill, G. W., 1970a, Algorithm 395, Student's t-distribution: Communications
    of the Association for Computing Machinery, v. 13, no. 10, p. 617-619.

_____ 1970b, Algorithm 396, Student's t-quantiles: Communications of the
    Association for Computing Machinery, v. 13, no. 10, p. 619-620.

Hill, I. D., and Pike, M. C., 1967, Algorithm 299, Chi-squared integral:
    Communications of the Association for Computing Machinery, v. 10,
    no. 4, p. 243-244.

International Business Machines, 1970, System/360 scientific subroutine package
    version III programmer's manual: IBM Order No. GH20-0205.

Johnson, N. L., and Kotz, S., 1970, Continuous univariate distributions, 2 vols:
    New York, John Wiley, 300 p., 306 p.

Kirby, W., 1972, Computer-oriented Wilson-Hilferty transformation that
    preserves the first three moments and the lower bound of the Pearson
    Type 3 distribution: Water Resources Research, v. 8, no. 5, p. 1251-1254.

Lewis, P. A. W., Goodman, A. S., and Miller, J. M., 1969, A pseudorandom
    number generator for the System/360: IBM Systems Journal, v. 8, no. 2, p.
    136-146.

Mood, A. M., Graybill, F. A., and Boes, D. C., 1974, Introduction to the
    theory of statistics, 3d ed: New York, McGraw-Hill, 564 p.

U.S. Water Resources Council, Hydrology Subcommittee, 1977, Guidelines for
    determining flood flow frequency, Bulletin 17A: Washington, D.C.

Zelen, N., and Severo, N. C., 1964, Probability functions, Chap. 26 in
    M. Abramowitz and I. A. Stegun, eds., Handbook of Mathematical Functions:
    U.S. National Bureau of Standards, Applied Mathematics Series No. 55,
    1045 p.