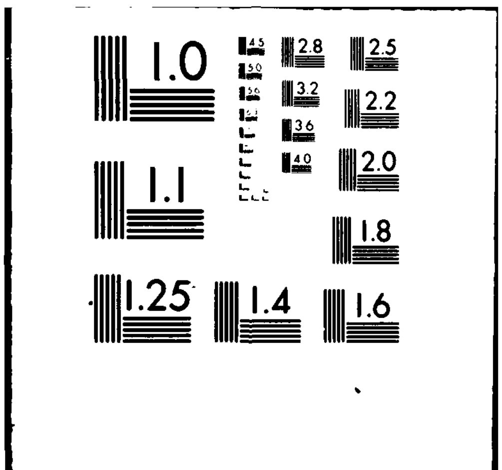
AD-A084 259 TEXAS A AND M UNIV COLLEGE STATION INST OF STATISTICS F/6 9/2
ONESAM, A COMPUTER PROGRAM FOR NONPARAMETRIC DATA ANALYSIS AND --ETC(U)
MAR 80 E PARZEN, S ANDERSON DAAG29-80-C-0070
UNCLASSIFIED TR-B-1 ARO-16992.1-M NL

1 OF 1
AD A
084259

END
DATE
FILMED
6-80
DTIC




MICROCOPY RESOLUTION TEST CHART

LEVEL II ARO 16992.1-M

# TEXAS A&M UNIVERSITY

COLLEGE STATION, TEXAS 77843

INSTITUTE OF STATISTICS
Phone 713 - 845-3141



(6) ONESAM, A COMPUTER PROGRAM FOR NONPARAMETRIC DATA ANALYSIS AND DENSITY QUANTILE ESTIMATION.

(10) by Emanuel/Parzen ~~and~~ Scott/Anderson

Institute of Statistics, Texas A&M University

(14) TR-B-1

12/79

(9) Technical Report No. B-1

(11) March 1980



Texas A&M Research Foundation
Project No. 4226

"Robust Statistical Data Analysis and Modeling"

Sponsored by the U.S. Army Research Office
Grant DAAG29-80-C-0070 (15)

Professor Emanuel Parzen, Principal Investigator

Approved for public release; distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER<br>Technical Report No. B-1 | 2. GOVT ACCESSION NO.<br>AD-A084 259 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>ONESAM, A Computer Program for Nonparametric Data Analysis and Density Quantile Estimation | | 5. TYPE OF REPORT & PERIOD COVERED<br>Technical |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>Emanuel Parzen and Scott Anderson | | 8. CONTRACT OR GRANT NUMBER(s)<br>DAAG29-80-C-0070 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Texas A&M University<br>Institute of Statistics<br>College Station, TX 77843 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>U. S. Army Research Office<br>P. O. Box 12211<br>Research Triangle Park, NC 27709 | | 12. REPORT DATE<br>March 1980 |
| | | 13. NUMBER OF PAGES<br>76 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The view, opinions, and/or findings contained in this report are those of the author(s) and should not be construed as an official Department of the Army position, policy, or decision, unless so designated by other documentation.

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Statistical data modeling, statistical computing, goodness of fit tests, nonparametric data analysis, density quantile function.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report is a User's Guide to ONESAM, a mainline Fortran computer program that implements the one-sample non-parametric data analysis and Goodness of Fit procedures introduced by Parzen.

# ONESAM, A COMPUTER PROGRAM FOR NONPARAMETRIC DATA ANALYSIS AND DENSITY QUANTILE ESTIMATION

by

Emanuel Parzen and Scott Anderson

## 1. Introduction

ONESAM (one sample) is a FORTRAN mainline program that implements the one sample non-parametric data analysis and density quantile estimators proposed by Parzen (1979). There are 4 stages to a quantile function and density-quantile function analysis of a single data sample. The program: Stage 1. Accepts grouped or individual (ungrouped) data, and forms non-parametric raw estimates of the quantile function; Stage 2. Forms a raw density-quantile function and sample spacings.

If the user specifies a null hypothesis probability law, the program checks the goodness of fit of the data to that probability law, and forms non-parametric smooth estimates of the density-quantile function in Stages 3 and 4.

Section 2 describes the stages of analysis in more detail. Section 3 is a listing of the general operating characteristics and requirements of the ONESAM program. Section 4 details the procedure to use ONESAM including the options available to users. Section 5 lists example IBM JCL setups.

## 2. Comments on the Stages of Analysis

Stage 1: User inputs data, outputs sample quantile function.

A. If the input is ungrouped data, the user must supply: the tape number (specified by DATA), upon which the data resides (if other than unit 5 which is the tape the data is usually on when read in from cards); the format of the data; and the data itself. The output of stage 1 consists of: various descriptive statistics; the order statistics by quartiles; the empirical quantile function, $\tilde{Q}(u)$, defined by $\tilde{Q}(u) = X_{j;n}$ for $u = \frac{j-.5}{n}$, $j = 1, \ldots, n$; and a quantile box plot.

Options:

1. Data can be standardized to take values in the unit (0,1) interval by specifying IOPT2 = 1. The output (which includes descriptive statistics, order statistics by quartiles, and $\tilde{Q}$) is based on the standardized data. When this option is used, future stages of analysis will use standardized data for analysis.

2. Compute $\tilde{Q}(u)$ using linear interpolation, for $u = jh$, $j = 1, \ldots, NQ$ where $NQ = [1/h]$ by specifying FINC = h. When this option is used, future stages will use $\tilde{Q}$ computed at points $h, 2h, \ldots, 1 - h$ .

B. If the user inputs grouped data by specifying IOP1 = 1, the user must also supply: the lower limit of the first interval (XMIN); the interval width (XINT); and, if desired, the total number of observations (NOBS). The user must input the numbers of observations in each interval in Format (20F4.0) and supply a value, h (FINC), to be used in computing $\tilde{Q}$. As with the raw data, the user must supply the number of the unit (DATA) that the data is to be read from (if other than unit 5).

Stage 2: Raw density-quantile function $\tilde{f}Q(u)$ and sample spacings.

In Stage 2, the function $\tilde{f}Q(u)$, defined by

$$\tilde{f}Q(jh) = \frac{2h}{\tilde{Q}((j+1)h)-\tilde{Q}((j-1)h}$$

for $u = jh$, $j = 1, \ldots, NQ$, is computed. Also computed in Stage 2 is $\tilde{q}(u) = 1/\tilde{f}Q(u)$ . Plots of these functions are given.

Note that if $\tilde{Q}$ is not computed via linear interpolation from grouped data, but rather is computed from ungrouped data, the definition of $\tilde{q}(u)$ is such that

$$\tilde{q}(u) = n\{\tilde{Q}((j+.5)/n) - \tilde{Q}((j-.5)/n)\} = n\{X_{(j+1);n} - X_{j;n}\}$$

for $\frac{j-.5}{n} \leq u < \frac{j+.5}{n}$, $j = 1, 2, \ldots, n-1$.

Stages 1 and 2 constitute the non-parametric raw analysis of the data. If the user supplies a null-hypothesis probability law, one can proceed to Stages 3 and 4 which constitute a non-parametric smooth analysis of the data and also a goodness of fit test for the null hypothesis.

Stage 3: Raw $\tilde{D}(u)$ and $\tilde{\phi}(v)$.

The user must specify a density-quantile function $f_0Q_0$ for the null hypothesis $H_0$: $Q = \mu + \sigma Q_0$. This is done using indicator variables. If one wishes to test against density "j", then set DQH(j) = 1. The possible densities one can use are supplied in Table I.

The program computes: (1) the raw transformation density,

$$\tilde{d}(u) = f_0Q_0(u)\tilde{q}(u)/\tilde{\sigma}_0 ,$$

also called the weighted spacings; and (2) the raw transformation distribution function, $\tilde{D}(u)$ called the cumulative weighted spacings. Tests for $H_0$ could be obtained by using the facts that under $H_0$ the $\tilde{d}$'s should be distributed as exponential with mean 1, and $\sqrt{n}\,\{\tilde{D}(u) - u\}$ is asymptotically a a Brownian Bridge stochastic process $B(u)$, or a linear transformation of $B(u)$. $\tilde{\sigma}_0$ is an estimate of $\sigma_0$ defined by

$$\sigma_0 = \int_0^1 f_0Q_0(u)q(u)du .$$

The program then computes the Fourier transform

$$\tilde{\phi}(v) = \int_0^1 \tilde{d}(u) \exp(2\pi ivu)du \text{ for } v = 0, 1, \ldots, m$$

by specifying ORDER = m (the program chooses order 5 if no other order is specified). By comparing the square modulus of the phi's, $|\tilde{\phi}(v)|^2$ , which are plotted, to a suitable threshold value (for example, 2*NOBS), one could use these values as evidence to accept or reject $H_0$.

The real part of the phi's are submitted as $\phi'(m)$ , $\phi'(m-1)$, ..., $\phi'(1)$, ..., $\phi'(m-1)$, $\phi'(m)$ to obtain the shifted S array. Two separate shifted S array's are computed; first where $\phi'(v) = \mathrm{Re}\tilde{\phi}(v)$ and second where $\phi'(v) = \mathrm{Re}(-1)^v\tilde{\phi}(v)$ for v = 1, 2, ..., m .

Stage 4: Smooth $\hat{D}(u)$ and $\hat{f}Q(u)$.

Autoregressive smoothed density estimators $\hat{f}Q$ and smoothed transformation distribution functions $\hat{D}(u)$ are computed for successive orders 1, 2, ..., m. Plots of $\tilde{D}$ vs $\hat{D}$ are given for all orders to help select the order that gives the best fit to the data. In addition, $\sigma_m^2$ , $\log_e(\sigma_m^2)$ , and the values of CAT and AIC criteria are printed by AR order. If CAT (or AIC) selects order 0, one considers this as evidence to not reject the null hypothesis.

Alternative analyses: Stages 3 and 4 can be repeated for several possible hypothesized densities simply by setting any other DQH(j) = 1.

TABLE I

List of Density-Quantile Functions

| j | $f_0Q_0$ |
|---|---|
| 1 | Normal |
| 2 | Exponential |
| 3 | Logistic |
| 4 | Double Exponential |
| 5 | Uniform Reciprocal |
| 6 | Cauchy |
| 7 | Extreme Value |
| 8 | Log Normal |
| 9 | Pareto |
| 10 | Weibull |
| 11 | Half Logistic |

Table II

A List

| Parameter | Value | Status | Default | Remarks |
|---|---|---|---|---|
| SIZE | Real $\geq 0$ | Optional | 0.0 | Scale factor for CALCOMP plots. If equal to 0.5 the plots are half sized. |
| PPLT | Integer | Optional | 0 | Plotting option -- equal to 0 if only printer plots are desired. Equal to 1 if only CALCOMP plots are desired. Not equal to 0 or 1 if both CALCOMP and printer plots are desired. |
| NBATCH | Integer $\geq 1$ | Required | 0 | Number of data sets to be processed. |

TABLE II I

Parameter List

| Parameter | Value | Status | Default | Remarks |
|---|---|---|---|---|
| N = | integer $\leq$ 510 | required | 0 | Sample size for raw data; the number of intervals for grouped data |
| DATA = | integer | optional | 5 | 1 = data on unit 1<br>(if Data = 2, card 14 must change accordingly) |
| TRANS = | real | optional | 1.0 | Value of exponent for power transformation if desired |
| IOPT1 = | 0, 1, 2 | optional | 0 | 1 for grouped data with equal interval widths<br>2 for grouped data in unequal intervals |
| XMIN = | real | optional<br>required if IOPT1 = 1 | first order statistic | Natural minimum of data. For grouped data, it is lower limit of first interval |
| FINC = | real | optional<br>required if IOPT1 = 1 | 0.0 | Value of h for computing equally spaced $\tilde{Q}$.<br>If used, FINC usually equals .01, .025, or .05 |
| NOBS = | integer | optional | N | Number of observations for grouped data.<br>It is used in computing CAT |
| IOPT2 = | 0, 1 | optional | 0 | 1 if user desires data to be standardized to (0, 1) before analysis. Sometimes useful for comparing several batches |
| ORDER = | integer | optional | 5 | Maximum order to be used for autoregressive estimator usually $\leq$ 10 |
| XINT = | real | required if IOPT1 = 1 | 0 | Interval width for grouped data |

TABLE III(continued)

| Parameter | Value | Status | Default | Remarks |
| --- | --- | --- | --- | --- |
| DQH(J) = J = 1, ... 12 | 0, 1 | optional | 0 | 1 if user wishes to test against "J"th hypothesized density. See table in Section 3 for allowable densities |
| WIDTH = | Integer | optional | 80 | Number of columns in printer plots |
| BATAP = | real > 0 | required if D(9) = 1 | -- | Shape parameter for Pareto density |
| BETAW = | real > 0 | required if D(10) = 1 | -- | Shape parameter for Weibull density |

## 3. Operating Characteristics

The ONESAM program is written completely in FORTRAN and requires approximately 192 K bytes of storage. The program reads on I/O unit 5 and writes on I/O unit 6. Additionally, I/O unit 9 is required for scratch file purposes and the user may request that the program fetch its data set off of another (not 5, 6, 9) I/O unit.

Initially, the ONESAM package is set up to require the California Computer Products Plotting System (CALCOMP). The program is easily converted to systems without the CALCOMP plotter by:

(1) deleting all sections of code enclosed by cards of the form C$$$$$$ followed by C - - - - - - -.

(2) removing the subroutines GPLOT, LABEL, MXMNSC, and QBXPLT.

ONESAM requires the inverse normal IMSL subroutine MDNRIS. Systems without the IMSL package may substitute an inverse normal subroutine of the form

```
subroutine MPNRIS (P, Y, IER)
```

where P is the input probability value, Y is the output inverse normal value, and IER is a dummy error indicator variable. A listing of the IMSL subroutine MDNRIS is included in this report on page 75.

## 4. Procedures

This section details the procedure one should use to complete a data analysis using the ONESAM package. The most critical feature of the ONESAM package is to understand completely the variables listed in Tables II and III. These variables contain almost all the information and options used in the analysis.

The procedures are most easily explained by use of the following typical JCL setup:

```
 1.    //XXXXX JOB (XXXX,XXXX,530,005,XX),'XXXXXX',MSGLEVEL=(1,0)
 2.    /*JOBPARM R=192,K=0
 3.    /*LEVEL        0
 4.    //JOBLIB  DD DSN=WYL.SW.ROD.JOBLIB,DISP=OLD
 5.    //STEP1   EXEC PGM=ONSPLP,REGION=192K
 6.    //PLOTLOG  DD SYSOUT=A
 7.    //VECTR1   DD DSN=&&VECTR1,UNIT=SYSDA,SPACE=(TRK,(5,1)),DISP=(,PASS)
 8.    //VECTR2   DD DSN=&&VECTR2,UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS)
 9.    //PLOTPARM DD *
10.     &PLOT MSGLVL=1,XMIN=-.5,YMIN=-.5,XMAX=810.0 &END
11.    //FT05F001 DD *
12.     &A NBATCH=III,PPLT=I,&END
13.    DATA SET NAME
14.     &PARMS N=III,&END
15.    FORMAT OF DATA EG (16F5.1)
16.    //FT06F001 DD SYSOUT=A
17.    //FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
18.    // DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
19.    //FT01F001 DD DSN=WYL.XX.XXX.XXXXX(XXXXX),
20.    // DISP=SHR,LABEL=(,,,IN)
21.    //STEP2  EXEC PGM=VTPLOT,REGION=192K,COND=(0,NE)
22.    //STEPLIB  DD DSN=SYS1.VTEC.PLOTLIB,DISP=SHR
23.    //PLOTLOG  DD SYSOUT=A
24.    //VECTR1   DD DSN=&&VECTR1,DISP=(OLD,DELETE)
25.    //VECTR2   DD DSN=&&VECTR2,DISP=(OLD,DELETE)
26.    //SYSVECTR DD SYSOUT=P,DCB=LRECL=126
27.    /*END
```

Cards 6 - 10 and 21 - 26 are only required if the CALCOMP plotting is requested, i.e., PPLT $\neq$ 0. Cards 12 and 14 utilize the namelist feature of the FORTRAN language and contain the user specified options of Tables II and III respectively. Both of these cards are required even if no options are requested. Card 12 must begin with b&Ab and end with ,&END. (Note: b denotes a blank space). Card 14 must begin with b&PARMSb and end with ,&END. User options are specified as illustrated. Card 15 is required only if the user inputs ungrouped data. Data format is assumed (20F4.0) if grouped data. Card 11 specifies the input unit. Card 16 specifies the output unit. Cards 17 and 18 specify the scratch unit. Cards 19 and 20 specify the optional unit on which the user has his data file located. If the user wishes to include his data set within his JCL deck he would list his data between cards 15 and 16.

Cards 12 - 15 are repeated NBATCH times.

## 5. Sample JCL

Sample JCL are given for the following types of jobs:

A. Ungrouped data on external file (WYLBUR) using I/O unit 1;

B. Ungrouped data on cards, with three sets of data;

C. Grouped data on cards and requesting CALCOMP plotting.

A

```
//SCOTT JOB (R579,002B,S30,005,SW),'ONSPLP',MSGLEVEL=(1,0)
/*JOBPARM R=192,K=0
/*LEVEL        0
//JOBLIB  DD DSN=WYL.SW.RQD.JOBLIB,DISP=OLD
//STEP1   EXEC PGM=ONSPLP,REGION=192K
//FT05F001 DD *
 &A &END
BUFFALO SNOWFALL DATA (1910-1972)
 &PARMS N=63,DQH(1)=1,DATA=1,FINC=0.05,&END
(5F10.0)
//FT06F001 DD SYSOUT=A
//FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//FT01F001 DD DSN=WYL.MW.ENP.QUANT(BFSNO),
// DISP=SHR,LABEL=(,,,IN)
/*END
```

B

```
//SCOTT JOB (R579,002B,S30,005,SU),'ONSPLP',MSGLEVEL=(1,0)
/*JOBPARM R=192,K=0
/*LEVEL        0
//JOBLIB  DD DSN=WYL.SU.RQD.JOBLIB,DISP=OLD
//STEP1   EXEC PGM=ONSPLP,REGION=192K
//FT05F001 DD *
 &A NBATCH=3,&END
TIPPETT'S WARP BREAKS  AL
 &PARMS N=9,DQH(1)=1,DQH(2)=1,&END
(10F5.0)
 26. 30. 54. 25. 70. 52. 51. 26. 67.
TIPPETT'S WARP BREAKS  AM
 &PARMS N=9,DQH(1)=1,DQH(2)=1,&END
(10F5.0)
 18. 21. 29. 17. 12. 18. 35. 30. 36.
TIPPETT'S WARP BREAKS  AH
 &PARMS N=9,DQH(1)=1,DQH(2)=1,&END
(10F5.0)
 36. 21. 24. 18. 10. 43. 28. 15. 26.
//FT06F001 DD SYSOUT=A
//FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
/*END
```

C

```
//SCOTT JOB (R579,002B,530,005,SU),'ONSPLP',MSGLEVEL=(1,0)
/*JOBPARM R=196,K=0
/*LEVEL        0
//JOBLIB  DD DSN=UYL.SU.RQD.JOBLIB,DISP=OLD
//STEP1   EXEC PGM=ONSPLP,REGION=196K
//PLOTLOG  DD SYSOUT=A
//VECTR1   DD DSN=&&VECTR1,UNIT=SYSDA,SPACE=(TRK,(5,1)),DISP=(,PASS)
//VECTR2   DD DSN=&&VECTR2,UNIT=SYSDA,SPACE=(CYL,(1,1)),DISP=(,PASS)
//PLOTPARM DD *
 &PLOT MSGLVL=1,XMIN=-.5,YMIN=-.5,XMAX=810.0 &END
//FT05F001 DD *
 &A PPLT=1,&END
BLOOD PRESSURE DATA--NON-USERS--AGE 17-24
 &PARMS N=16,DGH(1)=1,IOPT1=1,XMIN=85.,XINT=5.,FINC=.05,&END
   0   1   3  10  11  15  20  13  10   8   4   3   2   0   0   0
//FT06F001 DD SYSOUT=A
//FT09F001 DD UNIT=SYSDA,SPACE=(TRK,50),
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=3120)
//STEP2  EXEC PGM=VTPLOT,REGION=196K,COND=(0,NE)
//STEPLIB  DD DSN=SYS1.VTEC.PLOTLIB,DISP=SHR
//PLOTLOG  DD SYSOUT=A
//VECTR1   DD DSN=&&VECTR1,DISP=(OLD,DELETE)
//VECTR2   DD DSN=&&VECTR2,DISP=(OLD,DELETE)
//SYSVECTR DD SYSOUT=P,DCB=LRECL=126
/*END
```

```
C***************************************************************
C    DRIVER PROGRAM FOR ONE SAMPLE ANALYSIS. SEE DR. PARZEN
C    FOR COPY OF USAGE INSTRUCTIONS.
C***************************************************************
      COMMON BETAP,BETAU
      DIMENSION NAME(20),IFORM(20),LICASE(22)
      COMMON/SAMPLE/X(552),XS(552),WXS(552),QN(552),U(552),
     &              DHAT(552),CWXS(552),F(552),FQ(552)
      DATA LICASE/4HNORM,4HAL  ,4HEXPO,4HNTAL,4HLOGI,4HSTIC,
     &            4HDBL ,4HEXP ,4HUN R,4HECIP,4HCAUC,4HHY  ,
     &            4HEX V,4HALUE,4HLOG ,4HNORM,4HPARE,4HTO  ,
     &            4HWEIB,4HULL ,4HHALF,4H LOG/
      INTEGER DATA,WIDTH
      INTEGER DQH(30),ORDER      ,PPLT
      DATA XONE/.0000000001/
      NAMELIST/PARMS/PPLT,SIZE,DATA,WIDTH,ORDER,N,TRANS,XMIN,BETAP,
     +BETAU,DQH,XINT,IOPT1,IOPT2,NOBS,FINC/A/NBATCH,PPLT
      NBATCH=1
      PPLT=0
      READ(5,A)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL PLOTS(0,0,0)
C-------------------------------------------------------------------
      DO 999 IBATCH=1,NBATCH
   10 CONTINUE
      READ(5,1001) NAME
 1001 FORMAT(20A4)
      DO 15 I = 1,30
   15 DQH(I) = 0
      ORDER = 5
      WIDTH = 0
      DATA = 5
      SIZE = 0.0
      N = 0
      TRANS = 1.0
      XMIN = XONE
      XINT=0.
      IOPT1=0
      IOPT2=0
      NOBS=0
      FINC=0.
      BETAP=0.0
      BETAU=0.0
      READ(5,PARMS)
      IF(NOBS.EQ.0)NOBS=N
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
```

```
      IF((SIZE.NE.0.0).AND.PPLT.NE.0) CALL FACTOR(SIZE)
C-------------------------------------------------------------
      WIDTH=70
C     IF(WIDTH .EQ. 0) WIDTH = (10*N)/(8*((N-1)/71+1))
      ORDER = ORDER + 1
      IF(N .LE. 511) GO TO  19
      WRITE(6,901) N
      GO TO 999
   19 CONTINUE
      IF(IOPT1.GT.0)GO TO 155
C   READ UNGROUPED DATA FOR IOPT1=0
      READ(5,1001) IFORM
      READ(DATA,IFORM) (X(I),I=1,N)
      Z = SIGN(1.0,TRANS)
      IF(TRANS .EQ. 1.0) GO TO 55
C   TRANSFORM UNGROUPED DATA IF DESIRED
      IF(TRANS .EQ. 0.0) GO TO 52
      DO 51 I = 1,N
   51 X(I) = Z*X(I)**TRANS
      WRITE(6,905)
  905 FORMAT(1H1/////)
      GO TO 55
   52 DO 53 I = 1,N
   53 X(I) = ALOG(X(I))
      WRITE(6,905)
   55 CONTINUE
      IFIRST = 1
      X0 = XMIN
      GO TO 57
C   READ GROUPED DATA IN EQUAL-SIZED INTERVALS FOR IOPT1=2
  155 READ(DATA,903)(X(I),I=1,N)
      X0=XMIN
      IF(IOPT1.EQ.1)GO TO 56
      READ(5,1001)IFORM
      READ(DATA,IFORM)(DN(I),I=1,N)
   56 CONTINUE
      IFIRST=1
   57 CONTINUE
      IF(BETAP.GT.0)GO TO 22
      IF(DQN(9).EQ.0)GO TO 22
      DQN(9)=0
      WRITE(6,902)BETAP
 22   CONTINUE
      IF(BETAW.GT.0)GO TO 23
      IF(DQN(10).EQ.0)GO TO 23
```

```
      DQH(10)=0
      WRITE(6,902)BETAW
 23   CONTINUE
      DO 26 II=1,11
      IF(DQH(II).EQ.0) GO TO 26
      JJ=1+(II-1)*2
      CALL ONESAM(N,XO,6,2,NAME,WIDTH,1,IFIRST,
     +II,ORDER,LICASE(JJ),IOPT1,IOPT2,NOBS,FINC,XINT,NQ,PPLT)
      IFIRST=0
   26 CONTINUE
  999 CONTINUE
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL PLOT(0.,0.,999)
C---------------------------------------------------------------
      STOP
  901 FORMAT(//,5X,'SAMPLE SIZE OF ',I5,' G.T. MAX OF 511. RUN ABORTED')
  902 FORMAT(//,5X,F10.4,' IS ILLEGAL VALUE FOR BETA. DATA SET SKIPPED')
  903 FORMAT(20F4.0)
      END
```

```
      SUBROUTINE ACODEF(NTAPE,XNAME,IFORM,X)
C**************************************************
C   SUBROUTINE TO CONVERT 8 CHARACTER ALPHANUMERIC ARRAY
C   XNAME WHICH IS IN A-FORMAT TO THE REAL VARIABLE
C   X WHICH HAS THE 8 CHARACTER F-FORMAT IFORM
C   INPUT  : NTAPE : SCRATCH TAPE NUMBER
C                XNAME, IFORM
C   OUTPUT : X
C****************************************************
      DIMENSION XNAME(2),IFORM(2)
      REWIND NTAPE
      WRITE(NTAPE,10)XNAME
      REWIND NTAPE
      READ(NTAPE,IFORM)X
   10 FORMAT(2A4)
      RETURN
      END
```

```
      FUNCTION AREST(X,L,OPTKHN,OPTCOE)
C************************************************************************
C   FUNCTION TO COMPUTE AUTOREGRESSIVE ESTIMATOR EVALUATED AT X
C   METHOD: ARTWO = OPTKHN / ABS(1 + Y)**2
C           WHERE Y = OPTCOE(J)*EXP(I*J*2*PI*X) SUMMED OVER J = 1, L
C   INPUT:
C          X: SCALAR AT WHICH AUTOREGRESSIVE ESTIMATE IS EVALUATED.
C          L: ORDER. MUST BE LESS THAN 11. SEE METHOD.
C          OPTKHN: SEE METHOD.
C          OPTCOE: AUTOREGRESSIVE COEFFICIENTS OF ORDER L. SEE METHOD.
C                  OPTCOE IS A COMPLEX 10-VECTOR.
C   OUTPUT: FUNCTION RETURNS VALUE OF AUTOREGRESSIVE ESTIMATOR EVALUATED
C           AT X.
C   SUBROUTINES CALLED: NONE.
C************************************************************************
      COMPLEX OPTCOE(L)
      COMPLEX G
      PI=4.*ATAN(1.0)
      G=CMPLX(1.,0.)
      DO 1 J=1,L
      FJ=J
      G=G+OPTCOE(J)*CEXP(CMPLX(0.,X*2.*PI*FJ))
    1 CONTINUE
      AREST=OPTKHN/REAL(G*CONJG(G))
      RETURN
      END
```

```
      SUBROUTINE AUTORG(A,LSP1,M,ALPHA,PHI,FKH)
C************************************************************
C     COMPUTES THE COEFFICIENTS ALPHA(.) AND FKH OF THE
C     AUTOREGRESSIVE ESTIMATOR ACCORDING TO A RECURSIVE
C     ALGORITHM
C     INPUT :
C          A : VECTOR OF COMPLEX FOURIER TRANSFORM,
C              OF DIMENSION AT LEAST M
C          M : (M-1) IS THE MAXIMUM ORDER OF SCHEME
C              TO BE COMPUTED
C          LSP1 : ORDER OF SCHEME BEING COMPUTED PLUS 1. LSP1.GE.2
C     OUTPUT :
C          ALPHA : VECTOR OF COEFFICIENTS DEFINING THE
C              APPROXIMATING FUNCTION. HAS TO BE DIMEN-
C              SIONED AT LEAST M AND DECLARED COMPLEX
C          FKH : SCALES THE AUTOREGRESSIVE ESTIMATOR TO
C              INTEGRATE TO A(1). DECLARED REAL
C     ALPHA, PHI AND FKH ARE USED RECURSIVELY,IE. THEIR
C     VALUES AT OUTPUT FOR ORDER J ARE USED AS INPUT
C     FOR ORDER (J+1)
C************************************************************
      COMPLEX A(LSP1),ALPHA(LSP1),PHI(LSP1),G,FJH
      COMPLEX FKH
      LS=LSP1-1
      TWOPI=8.*ATAN(1.0)
      FJH=CMPLX(0.,0.)
      PHI(LS) = CMPLX(1.,0.)
      IF(LS .EQ. 1) FKH = CONJG(A(1))
      DO 4 I = 1,LS
    4 FJH=FJH+CONJG(A(I+1)*PHI(I))
      G=-FJH/FKH
      ALPHA(LS) = G
      IF(LS .EQ. 1) GO TO 5
      K = LS -1
      DO 2 I = 1,K
    2 ALPHA(I) = ALPHA(I) + G*PHI(I)
    5 CONTINUE
      DO 3 I=1,LS
    3 PHI(I) = CONJG(ALPHA(LS+1-I))
      FKH=FKH-FJH*CONJG(FJH)/CONJG(FKH)
      RETURN
      END
```

```
      SUBROUTINE CLPLT1(X,N,INIT,NAME,MM,IOPT)
C***********************************************************
C   SUBROUTINE TO PRINT AND PRINTER PLOT THE N-VECTOR X.
C   INPUT :
C           N,X
C           INIT : PRINTED INDEX OF FIRST PRINTED X
C           NAME : 4 CHARACTER LITERAL CONSTANT GIVING
C                  LABEL FOR X
C           MM : NUMBER OF COLUMNS IN PLOT (LE.101)
C           IOPT : 1,2 (POINT OR BAR PLOT)
C   SUBROUTINES CALLED : MAX,MIN
C***********************************************************
      DIMENSION X(N),AL(101)
      DATA NOUT/6/
      DATA BLANK,DOT,Z/1H ,1H.,1H*/
      IOPTX=0
      IF(N.GT.1) GO TO 10
      WRITE(NOUT,11) NAME,X(1)
   11 FORMAT(10X,A4,'(1) = ',F15.8)
      GO TO 99
   10 CONTINUE
C   INITIALIZE AL :
      ON=(MM-1)/2
      DO 20 J=1,MM
   20 AL(J)=DOT
      WRITE(NOUT,25) NAME,(AL(J),J=1,MM)
   25 FORMAT(/,14X,1HI,6X,A4/10X,15(1H-),2X,101A1)
      DO 30 J=1,MM
   30 AL(J)=BLANK
C   FIND MAX AND MIN :
      CALL MAX(X,N,XMAX,IND)
      CALL MIN(X,N,XMIN,IND)
      RX=XMAX-XMIN
      IF(RX.LT.1.E-20) IOPTX=1
C   PLOT :
      JJ=INIT
      DO 40 J=1,N
      IF(IOPTX.EQ.1) GO TO 36
      C1=(X(J)-XMIN)/RX
      C1=2.*(C1-.5)
      GO TO 37
   36 C1=0.
   37 K=ON*(C1+1.)+1.5
      AL(K)=Z
      IF(IOPT.EQ.1) GO TO 35
```

```
      DO 39 I=1,K
39    AL(I)=Z
35    CONTINUE
      WRITE(NOUT,38) JJ,X(J),(AL(I),I=1,MM)
38    FORMAT(10X,I5,F10.4,2X,101A1)
      JJ=JJ+1
      AL(K)=BLANK
      IF(IOPT.EQ.1) GO TO 40
      DO 41 I=1,K
41    AL(I)=BLANK
40    CONTINUE
99    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE COLPLT(X,N,INIT,NAME)
C*************************************************
C   SUBROUTINE TO PRINTER PLOT AND PRINT THE VECTOR X OF LENGTH N
C   THE VALUES PLOTTED ARE WRITTEN TO THE LEFT OF THE PLOT.
C   INPUT :
C           N,X(1),....,X(N)
C           NAME : A 4 CHARACTER ALPHANUMERIC VARIABLE
C           USED TO LABEL PRINTING.
C           INIT : THE FIRST VALUE PRINTED IS OF THE FORM
C              NAME (INIT) =
C   SUBROUTINES CALLED :
C           MAX,MIN
C*************************************************
      DIMENSION X(N),AL(61)
      DATA NOUT,BLANK,DOT,Z/6,1H ,1H.,1H*/
      IF(N.GT.1) GO TO 10
      WRITE(NOUT,11) NAME,X(1)
   11 FORMAT(10X,A4,'(1) = ',F15.8)
      RETURN
   10 CONTINUE
      DO 4 J=1,61
    4 AL(J)=DOT
      WRITE(NOUT,5) AL
    5 FORMAT(45X,61A1)
      DOT=BLANK
      DO 6 J=1,61
    6 AL(J)=BLANK
      AL(31)=DOT
      CALL MAX(X,N,XMAX,IND)
      CALL MIN(X,N,XMIN,IND)
      C=XMAX-XMIN
      IF(C.LT.1.E-20) GO TO 99
      JJ=INIT
      DO 7 J=1,N
      C1=(X(J)-XMIN)/C
      C1=2.*(C1-.5)
      K=30.*(C1+1.)+1.5
      AL(K)=Z
      WRITE(NOUT,8) NAME,JJ,X(J),AL
    8 FORMAT(11X,A4,1H(,I3,6H) =   ,F15.8,5X,61A1)
      JJ=JJ+1
      AL(K)=BLANK
      AL(31)=DOT
    7 CONTINUE
      RETURN
```

```
 99  WRITE(NOUT,100) XMAX
100  FORMAT(10X,'RANGE.LT.1.E-20 IN COLPLT, XMAX = ',E10.3)
     RETURN
     END
```

```
      SUBROUTINE DATSTD(X,XO,N,QN)
C**************************************************
C   SUBROUTINE TO STANDARDIZE VECTOR X BY (X(I)-XO)/(X(N)-XO)
C   AND RETURN THE STANDARDIZED DATA IN THE VECTOR QN
C   INPUT :
C          X,N : VECTOR OF ORDERED DATA OF LENGTH N
C          XO : MINIMUM VALUE TO BE USED IN STANDARDIZATION
C   OUTPUT :
C          QN : VECTOR OF STANDARDIZED DATA OF LENGTH N+1
C   SUBROUTINES CALLED : NONE
C**************************************************
      DIMENSION X(N),QN(N)
      N=N-1
      FACT = 1.0/(X(N) - XO)
      QN(1) = 0.0
      DO 10 I = 1,N
   10 QN(I+1)=(X(I) - XO)*FACT
      RETURN
      END
```

```
      SUBROUTINE DESTAT(X,N,NAME,IUNIT,IHEAD,IPPLT,WK1,WK2,NN3,FAC)
C************************************************************************
C   SUBROUTINE TO PRINT ORDERED ARRAY BY QUANTILES AND COMPUTE
C   DESCRIPTIVE STATISTICS.
C   INPUT:
C         X: ARRAY OF ORDER STATISTICS
C         N: DIMENSION OF ARRAY X
C         NAME: NAME OF DATA SET.MUST BE ARRAY OF DIMENSION 20 IN
C               CALLING PROGRAM.
C         IUNIT: NUMBER OF UNIT OUTPUT IS DESIRED ON.
C         IPPLT : SAME AS PPLT IN SUBROUTINE ONESAM
C         WK1,WK2 : WORK VECTORS OF LENGTH NN1=N*2+3
C   OUTPUT: PRINTED OUTPUT IS ON IUNIT.
C   NO SUBROUTINES CALLED.
C************************************************************************
      DIMENSION X(N),NAME(20),SUM(4),SUMSQ(4),WK1(NN1),WK2(NN1)
      DIMENSION ALF(3),FAC(3)
      DIMENSION BOX(5),QUART(5),ITITL(20),L(4)
      DIMENSION IHEAD(20),NAMEY(2),NAMEB(2)
      DATA ITITL/4HBOX ,4HPLOT,18*4H    /
      DATA NAMEY/4HQUAR,4HTILE/
      DATA NAMEB/4H    ,4H  ON/
      DATA ALF,BLANK/.05,.10,.25,1H /
      NN1=NN3-2
C   COMPUTE L,THE ARRAY OF QUARTILE SIZES
      LL = N/4
      IBRAN = N - 4*LL + 1
      IF(IBRAN.EQ.1)GO TO 10
      IF(IBRAN.EQ.2)GO TO 11
      IF(IBRAN.EQ.3)GO TO 12
      IF(IBRAN.EQ.4)GO TO 13
      GO TO 999
   10 CONTINUE
      L1 = LL
      L2 = LL
      L3 = LL
      L4 = LL
      GO TO 20
   11 CONTINUE
      L1 = LL
      L2 = LL
      L3 = LL
      L4 = LL + 1
      GO TO 20
   12 CONTINUE
```

```
      L1 = LL + 1
      L2 = LL
      L3 = LL
      L4 = LL + 1
      GO TO 20
   13 CONTINUE
      L1 = LL
      L2 = LL + 1
      L3 = LL + 1
      L4 = LL + 1
   20 CONTINUE
      L(1)=L1
      L(2)=L2
      L(3)=L3
      L(4)=L4
C    PRINT DATA ARRAY - ONE COLUMN FOR EACH QUANTILE.
      WRITE(IUNIT,1001) NAME
      WRITE(IUNIT,1020) IHEAD
      WRITE(IUNIT,1002)
      WRITE(IUNIT,1003)
      DO 30 I = 1,LL
   30 WRITE(IUNIT,1004) I,X(I),X(L1 + I),X(L1+L2+I),
     *                  X(L1+L2+L3+I)
      WRITE(IUNIT,1005)
      IF(L1 .GT. LL) WRITE(IUNIT,1006) X(L1)
      IF(L2 .GT. LL) WRITE(IUNIT,1007) X(L2 + L1 )
      IF(L3 .GT. LL) WRITE(IUNIT,1008) X(L3 + L2 + L1 )
      IF(L4 .GT. LL) WRITE(IUNIT,1009) X(L4 + L3 + L2 + L1 )
      IF(L4 .GT. LL) WRITE(IUNIT,1015) L4
C    COMPUTE AND PRINT DESCRIPTIVE STATISTICS.
      K = 1
      KK = 0
      DO 50 I = 1,4
      KK = KK + L(I)
      SUM(I) = 0.0
      SUMSQ(I) = 0.0
      DO 40 J = K,KK
      SUM(I) = SUM(I) + X(J)
      SUMSQ(I) = SUMSQ(I) + X(J)*X(J)
   40 CONTINUE
      K = K + L(I)
   50 CONTINUE
      WRITE(IUNIT,1010) (SUM   (I),I=1,4)
      WRITE(IUNIT,1011) (SUMSQ (I),I=1,4)
      S = SUM(1) + SUM(2) + SUM(3) + SUM(4)
```

```
      XBAR = S/FLOAT(N)
      XBI4 = (SUM(2) + SUM(3))/FLOAT(L2 + L3)
      XB04 = (SUM(1) + SUM(4))/FLOAT(L1 + L4)
      SSQ = SUMSQ(1) + SUMSQ(2) + SUMSQ(3) + SUMSQ(4)
      VAR = (SSQ -S*S/FLOAT(N))/FLOAT(N-1)
      SD = SQRT(VAR)
      R = X(L1+L2+L3+1) - X(L1)
      XMED = (X(N/2 + MOD(N,2)) + X(N/2+1) )/2.
      WRITE(IUNIT,1014) N,XMED                ,XBAR,XBI4,XB04,VAR,SD,R
      TRIMN = (X(L1)+ 2.*XMED + X(L1+L2+L3+1))/4.
      GASTY = .3*X(N/3+1) + .4*XMED +.3*X(N-N/3)
      WRITE(IUNIT,1019) TRIMN
      WRITE(IUNIT,1016) GASTY
      DO 60 I = 1,3
      IG = INT(ALF(I)*FLOAT(N))
      H = N -2*IG
      NMGM1 = N - IG - 1
      IGP2 = IG + 2
      TRM = X(IG+1)+X(N-IG)
      WMN = X(IG+1)*FLOAT(IG) + X(N-IG)*FLOAT(IG)
      DO 70 J = IGP2,NMGM1
      TRM = TRM + X(J)
      WMN = WMN + X(J)
   70 CONTINUE
      TRM = TRM/H
      WMN = WMN/FLOAT(N)
      WRITE(IUNIT,1017) ALF(I),WMN
      WRITE(IUNIT,1018) ALF(I),TRM
   60 CONTINUE
      BOX(1) = X(1)
      BOX(2) = X(L1)
      BOX(3) = XMED
      BOX(4) = X(L1 + L2 + L3 + 1)
      BOX(5) = X(N)
      DO 90 I = 1,5
   90 QUART(I) = FLOAT(I-1)*25.
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(IPPLT.EQ.0)GO TO 91
      CALL QBXPLT(X,N,UK1,UK2,NN3,NAME,3,FAC,BO)
      GO TO 999
 91   CONTINUE
C---------------------------------------------------------------
      DO 92 I=1,NN1
      UK2(I)=FLOAT(I)/(1.0+FLOAT(NN1))
 92   CONTINUE
```

```
      CALL EXPAND(N,X,NN1,UK1)
      CALL PPLOT(UK2,UK1,NN1,NAMEY,NAMEB,NAME,UK2,1,0,BLANK,IHEAD)
  999 CONTINUE
      RETURN
 1001 FORMAT(1H1,'DATA SET: ',20A4)
 1002 FORMAT(1H0,20X,'ORDER STATISTICS IN QUARTERS',/1H+,20X,
     +       '____________________________')
 1003 FORMAT(' SEQUENCE'/'   WITHIN'/' QUARTILE   ',
     +'FIRST QUARTER  SECOND QUARTER   THIRD QUARTER  FOURTH QUARTER'/
     +'+________   -------------  --------------   -------------  ------
     +________')
 1004 FORMAT(1X,I8 , 4(1X,F15.4))
 1005 FORMAT(1X)
 1006 FORMAT(1H+, 9X,F15.4)
 1007 FORMAT(1H+,25X,F15.4)
 1008 FORMAT(1H+,41X,F15.4)
 1009 FORMAT(1H+,57X,F15.4)
 1010 FORMAT(1H0,'SUM',5X,4(1X,F15.4))
 1011 FORMAT(1H0,'SUM OF'/' SQUARES',1X,4(1X,F15.4))
 1014 FORMAT(1H0,'SAMPLE SIZE = ',I6/' MEDIAN = ',F10.3/
     +' MEAN = ',F10.3/' INNER FOURTHS MEAN = ',F10.3/
     +' OUTER FOURTHS MEAN = ',F10.3/' VARIANCE = ',F10.3/
     +' STANDARD DEVIATION = ',F10.3/' INTERQUARTILE RANGE = ',F10.3)
 1015 FORMAT(1H+,I8)
 1019 FORMAT(1X ,'TRIMEAN = ',F10.3)
 1016 FORMAT(1X ,'GASTWIRTHS ESTIMATE = ',F10.3)
 1017 FORMAT(1X ,F3.2,'-WINSORIZED MEAN = ',F10.3)
 1018 FORMAT(1X ,F3.2,'-TRIMMED MEAN = ',F10.3)
 1020 FORMAT(1X,20A4)
      END
```

```
      SUBROUTINE EXPAND(N,X,NN1,Y)
C************************************************************
C     ROUTINE TO EXPAND A DATA SET BY COMPUTING AVERAGES OF
C     CONSECUTIVE DATA POINTS.
C     INPUT:
C           X : VECTOR OF SIZE N
C           NN1=2*N+1
C     OUTPUT:
C           Y : VECTOR OF EXPANDED DATA OF SIZE 2*N+1
C***********************************************************
      DIMENSION X(N),Y(NN1)
      NN=2*N
      NM1=N-1
      DO 5 I=1,N
      Y(I)=X(I)
 5    CONTINUE
      DO 10 J=1,N
      JM1=J-1
      JNE=2*(N-JM1)
      JNO=2*N-2*JM1-1
      NJ=N-JM1
      NJ1=N-JM1-1
      IF(NJ1.EQ.0)NJ1=1
      Y(JNE)=Y(NJ)
      Y(JNO)=(Y(NJ)+Y(NJ1))/2.0
 10   CONTINUE
      Y(NN1)=Y(NN)
      RETURN
      END
```

```
      SUBROUTINE FCODEA(NTAPE,X,IFORM,NAME)
C*****************************************************
C   SUBROUTINE TO CONVERT REAL VARIABLE X
C   WHICH HAS 4 CHARACTER F-FORMAT IFORM
C   TO 8 CHARACTER ALPHAMERIC ARRAY NAME WHICH IS
C   IN A FORMAT.
C   INPUT  : NTAPE : SCRATCH TAPE NUMBER
C              X, IFORM
C   OUTPUT : NAME(1),NAME(2) : 4 CHARACTERS EACH
C*******************************************************
      DIMENSION NAME(2),IFORM(2)
      REWIND NTAPE
      WRITE(NTAPE,IFORM)X
      REWIND NTAPE
      READ(NTAPE,10)NAME
   10 FORMAT(2A4)
      RETURN
      END
```

```
      SUBROUTINE FORIER(F,U,N,A,NA)
C*******************************************************************
C   SUBROUTINE TO COMPUTE THE FOURIER TRANSFORM
C   PHI(V) OF A DENSITY DEFINED ON (0,1) FOR V=0,1,....,N
C   INPUT :
C            F,U,N : VECTORS OF LENGTH N CONTAINING F(U),U
C            NA : MAXIMUM VALUE OF V FOR WHICH PHI(V) IS COMPUTED
C   OUTPUT : A : COMPLEX-VALUED VECTOR CONTAINING THE PHI'S
C   SUBROUTINES CALLED : NONE
C*******************************************************************
      DIMENSION F(N),U(N)
      COMPLEX A(NA),Z
      TWOPI=8.*ATAN(1.)
      FN=FLOAT(N)
      DO 20 IM=1,NA
      FIM=IM-1
      A(IM)=CMPLX(0.,0.)
      DO 10 I=1,N
         Z=CMPLX(0.,TWOPI*FIM*U(I))
   10    A(IM)=A(IM)+F(I)*CEXP(Z)
      A(IM)=A(IM)/FLOAT(N)
   20 CONTINUE
      A(1)=CMPLX(1.,0.)
      RETURN
      END
```

```
      FUNCTION FQFNC(X,IDQH)
C*********************************************************
C  ROUTINE TO COMPUTE THE VARIOUS DENSITY-QUANTILE FUNCTIONS
C       INPUT:
C              X - VALUE AT WHICH THE FUNCTION IS TO BE COMPUTED
C              IDQH - INDICATOR FOR THE DESIRED FUNCTION.
C                     MUST BE IN THE EXCLUSIVE RANGE 1-11
C*********************************************************
      COMMON BETAP,BETAU
      GO TO(1,2,3,4,5,6,7,8,9,10,11),IDQH
C  COMPUTE THE NORMAL
 1    CONTINUE
      IF( X .LT. .001) GO TO 101
      IF( X .GT. .999) GO TO 101
      PI=4.*ATAN(1.0)
      CALL MDNRIS(X,T,IER)
      FOQO=EXP(-0.5*T**2)/(SQRT(2.*PI))
      FQFNC = FOQO
      GO TO 99
  101 FOQO=0.
      FQFNC = FOQO
      GO TO 99
C  COMPUTE THE EXPONENTIAL
 2    CONTINUE
      FOQO=1.-X
      FQFNC = FOQO
      GO TO 99
C  COMPUTE THE LOGISTIC
 3    CONTINUE
      FQFNC = X*(1. - X)
      GO TO 99
C  COMPUTE THE DOUBLE EXPONENTIAL
 4    CONTINUE
      IF(X .LE. .5) FQFNC = X
      IF(X .GT. .5) FQFNC = 1. - X
      GO TO 99
C  COMPUTE THE UNIFORM RECIPROCAL
 5    CONTINUE
      FQFNC = (1. - X)**2
      GO TO 99
C  COMPUTE THE CAUCHY
 6    CONTINUE
      PI = 4.*ATAN(1.0)
      FQFNC = SIN(PI*X)**2/PI
      GO TO 99
```

```
C  COMPUTE THE EXTREME VALUE
 7    CONTINUE
      IF(X .EQ. 1.) GO TO 102
      FQFNC = (X - 1.)*ALOG(1. - X)
      GO TO 99
  102 CONTINUE
      FQFNC = 0.
      GO TO 99
C  COMPUTE THE LOG NORMAL
 8    CONTINUE
      IF(X .EQ. 1.) GO TO 103
      IF(X .EQ. 0.) GO TO 103
      CALL MDNRIS(X,T,IER)
      PI = 4.*ATAN(1.0)
      FQFNC = (-0.5*T**2-T)/SQRT(2.*PI)
      FQFNC = EXP(FQFNC)
      GO TO 99
  103 FQFNC = 0.0
      GO TO 99
C  COMPUTE THE PARETO
 9    CONTINUE
      FQFNC = (1.-X)**(1.+BETAP)/BETAP
      GO TO 99
C  COMPUTE THE WEIBULL
 10   CONTINUE
      IF(X .EQ. 1.) GO TO 104
      FQFNC = (1.-X)*(-ALOG(1.-X))**(1.-BETAW)/BETAW
      GO TO 99
  104 CONTINUE
      FQFNC = 0.
      GO TO 99
C  COMPUTE THE HALF LOGISTIC
 11   CONTINUE
      FQFNC=1.-X**2
 99   RETURN
      END
```

```
      SUBROUTINE GPLOT(X,Y,N,KODE,IOPT,LABEL,NCHAR,FAC,NP2)
C*********************************************************
C   SUBROUTINE TO PERFORM GENERAL PLOTTING OF AN ABSCISSA
C   VECTOR X(1),....,X(N) VERSUS AN ORDINATE VECTOR Y(1),....,Y(N)
C*********************************************************
C   GPLOT HAS THE FOLLOWING FEATURES :
C*********************************************************
C      1) POINT PLOTS, LINE PLOT, OR LINE PLOT WITH SYMBOLS
C         ARE AVAILABLE VIA IOPT.
C       2) THE SIZE OF THE AXIS CAN BE VARIED VIA FAC
C       3) LABELS FOR X-AXIS, Y-AXIS, AND CAPTION ABOVE PLOT
C        (UP TO 80 CHARACTERS EACH) CAN BE DETERMINED VIA
C        LABEL AND NCHAR.
C       4) SCALING CAN BE DONE OUTSIDE OR INSIDE GPLOT VIA IOPT.
C       5) A FAMILY OF PLOTS ON THE SAME AXIS CAN BE OBTAINED
C        BY FOLLOWING THE INITIAL CALL TO GPLOT WITH SUCCESSIVE
C        CALLS OF THE FORM
C          CALL GPLOT(X,Y,N,ISYMB+100,IOPT)
C        WHERE X,Y,N,IOPT ARE DESCRIBED BELOW AND ISYMB
C        IS AN INTEGER REPRESENTING THE INDEX OF THE DESIRED
C        SYMBOL TO BE PLOTTED AT EACH POINT (UNLESS IOPT
C        SPECIFIES A LINE PLOT)
C       6) GPLOT FRAMES EACH PLOT WITH A BORDER-BOX.
C        THE DIMENSIONS OF THE BOX ARE DETERMINED BY
C          HORIZ DIMENSION=1.2*X-AXIS LENGTH+1.0
C          VERT DIMENSION=1.2*Y-AXIS LENGTH +1.0
C        (THUS FOR 8.5BY 11 INCH BOX, X-AXIS LENGTH=8.33,
C        Y-AXIS LENGTH=6.25)
C        7) SUCCESSIVE BOXES ARE STACKED. IF THE BOX TO BE
C        DRAWN WOULD EXCEED THE LIMIT OF THE PAPER IN THE
C        Y-DIRECTION THE BOX (AND THE PLOT) ARE DRAWN AT THE
C        BASE OF THE NEXT COLUMN, ALIGNING THE ORIGIN WITH
C        THE FURTHEST EXTENDING BOX.
C        8) THE PEN IS AT THE UPPER LEFT CORNER OF THE BOX
C        DRAWN AFTER THE PLOT IS DRAWN. THUS THE PEN CAN BE
C        MOVED AT PUT BACK BETWEEN VALLS OF GPLOT.
C*********************************************************
C*********************************************************
C   INPUT :
C*********************************************************
C        X(1).....X(N),Y(1),....Y(N),N : THE ARRAYS TO BE PLOTTED
C             AND THE NUMBER OF POINTS.
C      NOTE : SCALING FACTORS ARE PUT IN THE N+1 AND N+2
C             ELEMENTS OF X AND Y. THUS X AND Y MUST BE
C             DIMENSIONED AT LEAST N+2.
```

```
C        KODE : LE.99 MEANS A NEW PLOT
C             GT.100 MEANS PLOT ON PREVIOUSLY DRAWN AXIS
C             AND KODE-100 IS PLOTTED SYMBOL IF ONE REQUESTED.
C        IOPT : -1,0,1 : POINT PLOT,LINE PLOT, LINE PLOT WITH
C             SYMBOLS.SCALING DETERMINED BY GPLOT
C               -101,100,101 : SAME AS -1,0,1 EXCEPT SCALING
C             FACTORS INPUTTED IN X(N+1),X(N+2),Y(N+1),Y(N+2)
C             (SEE SCALE WRITEUP)
C        LABEL : 20 BY 3 ARRAY WHOSE COLUMNS CONTAIN LABEL
C             FOR X-AXIS, LABEL FOR Y-AXIS, AND CAPTION ABOVE
C             PLOT RESPECTIVELY (EACH ELEMENT CONTAINS 4 CHARS)
C        NCHAR(1),....,NCHAR(3) : NUMBER OF CHARS IN X-AXIS
C             LABEL, NUMBER OF CHARS IN Y-AXIS LABEL, NUMBER
C             OF CHARS IN CAPTION.
C        FAC(1),....,FAC(3) : LENGTH OF X-AXIS, LENGTH OF
C             Y-AXIS, SCALE FACTOR FOR LABEL SYMBOL SIZE
C             WHERE SIZE=.07*FAC(3).
C*****************************************************************
C      NOTE :
C*****************************************************************
C   GPLOT USES CALIFORNIA COMPUTER PRODUCTS ROUTINES :
C      CALL PLOT(X,Y,IPEN)
C      CALL SCALE(X,AXLEN,NPTS,ICYCLE,TPER10)
C      CALL LINE(X,Y,NPTS,ICYCLE,ITYPE,ISYMB)
C      CALL AXIS1(X,Y,LAB,NC,AXLEN,ROTAT,XMIN,DEL,TPER10)
C      CALL SYMBOL(XCOORD,YCOORD,HEIGHT,LAB,ROTAT,NC)
C      CALL WHERE(XLOC,YLOC,FCT)
C   THE PLOTTING PAPER IS ASSUMED PAPERY INCHES IN HEIGHT.
C   THIS VALUE IS SET IN THE DATA STATEMENT.
C*********************************************************************
      DIMENSION X(NP2),Y(NP2),LABEL(20,3),NCHAR(3),FAC(3)
C   SET MACHINE PAPER LIMITS AND OUTPUT DEVICE NUMBER :
      DATA PAPERY,NOUT,IFST/11.0,6,0/
C   ADD NEW PLOT TO PREVIOUS AXIS (IF KODE.GE.100) :
      IFST=IFST+1
      IF(KODE.LT.100) GO TO 10
      TX=.2*SFAC1
      TY=1.0+SFAC2
      CALL PLOT(TX,-TY,-3)
      IPT=IOPT
      IIPT=IABS(IOPT)
      IF(IIPT.GT.1) GO TO 5
      CALL SCALE(X,SFAC1,N,1,10.0)
      CALL SCALE(Y,SFAC2,N,1,10.0)
    5 CONTINUE
```

```
      IF(IIPT.GT.1) IPT=ISIGN(IIPT-100,IOPT)
      L=KODE-100
      IF(IPT.EQ.0) L=0
      CALL LINE(X,Y,N,1,IPT,L)
      CALL PLOT(-TX,TY,-3)
      GO TO 99
   10 CONTINUE
      SFAC1=FAC(1)
      SFAC2=FAC(2)
C    NEW AXIS (IF KODE.LT.100) :
C    CHECK IF BOX IS WITHIN PAPER LIMITS IN Y-DIRECTION :
      CALL WHERE(XX,YY,FCT1)
      XBOX=(1.2*FAC(1)+1.0)*FCT1
      YBOX=(1.2*FAC(2)+1.0)*FCT1
      IF(YBOX.LE.PAPERY) GO TO 30
      WRITE(NOUT,20)
   20 FORMAT(10X,35HBOX TOO BIG IN Y-DIRECTION IN GPLOT)
      GO TO 99
   30 CONTINUE
C    CHECK IF BOX WILL STACK :
      IF(IFST.EQ.1) TOTX=0.0
      IF(IFST.EQ.1) TOTY=0.0
      IF(YBOX+TOTY.LE.PAPERY) GO TO 40
C    BOX WONT FIT, MOVE PEN TO LOWER LEFT OF BOX IN NEXT COLUMN :
      TOTX1=TOTX/FCT1
      TOTY1=TOTY/FCT1
      CALL PLOT(TOTX1,-TOTY1,-3)
      TOTY=0.0
      TOTX=0.0
   40 CONTINUE
C    DO SCALING (IF IABS(IOPT).LE.1) :
      IPT=IOPT
      IIPT=IABS(IOPT)
      IF(IIPT.GT.1) GO TO 50
      CALL SCALE(X,FAC(1),N,1,10.0)
      CALL SCALE(Y,FAC(2),N,1,10.0)
   50 CONTINUE
      IF(IIPT.GT.1) IPT=ISIGN(IIPT-100,IOPT)
C    DRAW AXIS :
      TX=.2*FAC(1)
      TY=.2*FAC(2)
      CALL PLOT(TX,TY,-3)
      CALL AXIS1(0.0,0.0,LABEL(1,1),-NCHAR(1),FAC(1),0.0,
     1X(N+1),X(N+2),10.0)
      CALL AXIS1(0.0,0.0,LABEL(1,2),NCHAR(2),FAC(2),90.0,
```

```
     1Y(N+1),Y(N+2),10.0)
C   DRAW PLOT :
      L=11
      IF(IPT.EQ.0) L=0
      CALL LINE(X,Y,N,1,IPT,L)
C   RETURN PEN TO LOWER LEFT CORNER, ADD CAPTION,AND DRAW BOX :
      CALL PLOT(-TX,-TY,-3)
      HGT=.07*FAC(3)
      CALL SYMBOL(.25*FAC(1),1.05*FAC(2),HGT,LABEL(1,3),
     10.0,NCHAR(3))
      BOXX=1.2*FAC(1)+1.0
      BOXY=1.2*FAC(2)+1.0
      CALL PLOT(0.0,BOXY,3)
      CALL PLOT(0.0,0.0,2)
      CALL PLOT(BOXX,0.0,2)
      CALL PLOT(BOXX,BOXY,2)
      CALL PLOT(0.0,BOXY,-2)
      TOTY=TOTY+FCT1*BOXY
      IF(FCT1*BOXX.GT.TOTX) TOTX=FCT1*BOXX
  99  CONTINUE
C   RETURN :
C     WRITE(NOUT,157) TOTX,TOTY,IFST
C157  FORMAT(10X,'TOTX=',F10.4,2X,'TOTY=',F10.4,2X,'IFST=',I3)
      RETURN
      END
```

```
      SUBROUTINE ICODEA(NTAPE,K,IFORM,NAME)
C***************************************************
C   SUBROUTINE TO CONVERT INTEGER VARIABLE K
C   WHICH HAS 8 CHARACTER I-FORMAT IFORM
C   TO 8 CHARACTER ALPHAMERIC ARRAY NAME WHICH IS
C   IN A-FORMAT.
C   INPUT  : NTAPE : SCRATCH TAPE NUMBER
C            K, IFORM
C   OUTPUT : NAME(1),NAME(2) : 4 CHARACTERS EACH
C****************************************************
      DIMENSION NAME(2),IFORM(2)
      REWIND NTAPE
      WRITE(NTAPE,IFORM)K
      REWIND NTAPE
      READ(NTAPE,10)NAME
   10 FORMAT(2A4)
      RETURN
      END
```

```
      SUBROUTINE KSD(D,U,N,DM,UM,DP,UP)
C*****************************************************
C   SUBROUTINE TO COMPUTE KOMOLGOROV-SMIRNOFF STATISTIC FOR
C   THE DEVIATIONS D(U)-U.  UPPER AND LOWER BOUNDS ARE GIVEN.
C   INPUT : D,U,N
C   OUTPUT :
C          DP,UP : MAX (+) DEVIATION, DP, WHICH IS AT U=UP
C          DM,UM : MAX (-) DEVIATION, DM, WHICH IS AT U=UM
C   SUBROUTINES CALLED : NONE
C*****************************************************
      DIMENSION D(N),U(N)
      SQN = SQRT(FLOAT(N))
      DP = SQN*(D(1) - U(1))
      UP = U(1)
      DM = DP
      UM = UP
      DO 10   I = 2,N
      DT = SQN*(D(I) - U(I) )
      IF(DT .GT. DP) UP = U(I)
      IF(DT .GT. DP) DP = DT
      IF(DT .LT. DM) UM = U(I)
      IF(DT .LT. DM) DM = DT
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  LABL(NAMEY,NAME,ITITLE,LABEL,Y,X,N,FAC)
C*************************************************************************
C  SUBROUTINE TO PERFORM A UTILITY, I.E. TO BUILD LABELS USED IN GPLOT FROM
C                                  LABELS IN PPLOT AND ONESAM AND TO SET SCALES
C  INPUT:   NAMEY: <=80 CHARACTERS FOR LABEL OF Y-AXIS
C           NAME:  <=80 CHAR. FOR LABEL OF X-AXIS
C           ITITLE:<=80 CHAR. FOR CAPTION OF THE PLOT
C           Y:     Y-AXIS PTS. (DIM >= N+2)
C           X:     X-AXIS PTS. (DIM >= N+2) N PTS TO BE PLOTTED
C           N:     NO. OF DATA PTS. TO BE PLOTTED BY GPLOT.
C           FAC:   CONTAINS FACTORS USED BY GPLOT IN AXIS FORMATION.
C  OUTPUT:  LABEL: COL1 CONTAINS NAME,COL2 CONTAINS NAMEY,COL3 CONTAINS ITITLE
C           X(N+1),X(N+2): SCALES FOR X-AXIS. (0,1) FOR U
C           Y(N+1),Y(N+2): SCALES FOR Y-AXIS USED IN GPLOT
C  SUBROUTINES CALLED:  MXMNSC (FROM CAL. COMP. PROD.)
C****************************************************************************
      DIMENSION  NAMEY(2),NAME(20),ITITLE(20),LABEL(20,3) ,X(552),Y(552)
     1,FAC(3)
      DATA IBL/4H    /
      DO 1 I=1,2
      LABEL(I,1)=NAME(I)
      LABEL(I,2)=NAMEY(I)
    1 LABEL(I,3)=ITITLE(I)
      DO 2 I=3,20
      LABEL(I,1)=NAME(I)
      LABEL(I,2)=IBL
    2 LABEL(I,3)=ITITLE(I)
      X(N+1)=0.0
      X(N+2)=0.125
      CALL SCALE(Y,FAC(2),N,1,10.0)
      RETURN
      END
```

```
      SUBROUTINE MAX(X,N,XMAX,IND)
C*******************************************************
C   SUBROUTINE TO FIND THE MAXIMUM VALUE (XMAX) AND THE
C   INDEX OF THE MAXIMUM VALUE (IND) OF A VECTOR X OF
C   LENGTH N.
C   INPUT :
C           N,X(1),....,X(N)
C   OUTPUT :
C           XMAX,IND
C   SUBROUTINES CALLED : NONE
C*******************************************************
      DIMENSION X(N)
      XMAX=X(1)
      IND=1
      IF(N.EQ.1) RETURN
      DO 1 I=2,N
      IF(XMAX.LT.X(I)) IND=I
    1 XMAX=X(IND)
      RETURN
      END
```

```
      SUBROUTINE MIN(X,N,XMIN,IND)
C*****************************************************
C   SUBROUTINE TO FIND THE MINIMUM VALUE (XMIN) AND THE
C   INDEX OF THE MINIMUM VALUE (IND) OF A VECTOR X OF
C   LENGTH N.
C   INPUT :
C           N,X(1),.....X(N)
C   OUTPUT :
C           XMIN,IND
C   SUBROUTINES CALLED : NONE
C*****************************************************
      DIMENSION X(N)
      XMIN=X(1)
      IND=1
      IF(N.EQ.1) RETURN
      DO 1 I=2,N
      IF(XMIN.GT.X(I)) IND=I
    1 XMIN=X(IND)
      RETURN
      END
```

```
      SUBROUTINE MUHAT(P,X,N,UMEAN,VMEAN,XMEAN)
C**********************************************************
C   SUBROUTINE COMPUTES MUHAT BASED ON NORMAL FOQO
C   INPUT:  P : TRUNCATION PT
C           X,N : DATA VECTOR OF LENGTH N
C           UMEAN,VMEAN : WEIGHT AND VARIANCE AS DETERMINED BY STEP1
C   OUTPUT: XMEAN
C   SUBROUTINES CALLED : NONE
C**********************************************************
      DIMENSION X(N)
      Q=0.
      KP=INT(N*P)
      J=KP+2
      L=N-KP-1
      DO 10 K=J,L
   10 Q=Q+X(K)/FLOAT(N)
      WT1=FLOAT(KP+1)/FLOAT(N)-P
      WT2=1.-P-FLOAT(N-KP-1)/FLOAT(N)
      Q=Q+WT1*X(KP+1)+WT2*X(N-KP)
      XMEAN=(Q+(X(KP+1)+X(N-KP))*UMEAN)*VMEAN
      RETURN
      END
```

```
      SUBROUTINE MXMNSC(XMIN,XMAX,AXLEN,FIRSTV,DELX)
C************************************************************
C   SUBROUTINE TO FIND SCALING INFORMATION FIRSTV, DELX FOR
C   A VECTOR X HAVING MINIMUM VALUE XMIN AND MAXIMUM VALUE
C   XMAX AND AN AXIS LENGTH AXLEN.
C   INPUT :
C           XMIN,XMAX,AXLEN
C   OUTPUT :
C           FIRSTV : LABEL OF FIRST TIC MARK ON AXIS
C           DELX : INCREMENT FOR LABELS OF SUCCESSIVE MARKS
C   SUBROUTINES CALLED : SCALE
C************************************************************
      DIMENSION XX(4)
      XX(1)=XMIN
      XX(2)=XMAX
      CALL SCALE(XX,AXLEN,2,1,10.)
      FIRSTV=XX(3)
      DELX=XX(4)
      RETURN
      END
```

```
      SUBROUTINE ONESAM(N,XO,IUNIT,IOPT,NAME,MM,JOPT,IFIRST,
     +IDQH,M,ICASE,IOPT1,IOPT2,NOBS,FINC,XINT,NQ,PPLT)
C*************************************************************
C   DRIVER FOR ONE SAMPLE ANALYSIS
C   INPUT  :
C            X : VECTOR OF SAMPLE OBSERVATIONS
C            N : DIMENSION OF X, XS, WXS, CWXS, U
C            XO : NATURAL MINIMUM VALUE FOR DATA INPUTTED FROM
C                 MAINLINE IN A-FORMAT
C            MM : NUMBER OF COLUMNS IN PRINTER PLOTS
C            IOPT : 1 OR 2 (POINT OR BAR PLOTS)
C            NAME : NAME OF DATA SET, MUST BE AN ARRAY OF DIMENSION
C                   20 IN CALLING PROGRAM.
C            IUNIT : NUMBER OF UNIT OUTPUT IS DESIRED ON
C            JOPT : IF 0, SCALINGS ARE PLOTTED; IF 1, LOG OF THE
C                   SCALINGS ARE PLOTTED; IF 2, BOTH ARE PLOTTED
C           IFIRST : IS 1 FOR THE FIRST CALL TO ONESAM TO OBTAIN
C                     SAMPLE DIAGNOSTICS AND RAW SPACINGS
C            IDQH : NUMBER CORRESPONDING TO HYPOTHESIZED FQ
C            M : MAXIMUM AR ORDER TO BE COMPUTED
C            ICASE : NAME OF HYPOTHESIZED DENSITY IN A-FORMAT
C            IOPT1 : EQUALS 1 FOR GROUPED DATA
C            IOPT2 : EQUALS 1 IF DATA IS TO BE STANDARDIZED TO (0,1)
C            FINC : IS INCREMENT FOR EQUALLY SPACED Q(U)
C            XINT : INTERVAL WIDTH FOR GROUPED DATA
C            PPLT : PLOTTING OPTION -
C                             EQUAL 0 : PRINTER PLOT
C                             EQUAL 1 : CALCOMP PLOT
C                             OTHERWISE : BOTH
C   OUTPUT :
C            X : VECTOR OF ORDER STATISTICS (IF IFIRST=1)
C            QN : VECTOR OF STANDARDIZED DATA (IF IFIRST=1)
C            XS : VECTOR OF SPACINGS (IF IFIRST=1)
C            WXS : VECTOR OF WEIGHTED SPACINGS
C            CWXS : VECTOR OF CUMULATIVE WEIGHTED SPACINGS
C            DHAT : VECTOR OF SMOOTHED D'S FOR CASE ICASE
C            FQ : VECTOR OF FQ'S FOR CASE ICASE
C   PRINTED OUTPUT IS ON IUNIT.  IT CONSISTS OF ORDER
C   STATISTICS BY QUARTILE, VARIOUS DESCRIPTIVE STATISTICS, AND
C   PRINTER PLOTS OF THE ORDER STATISTICS AND SPACINGS.
C   SUBROUTINES CALLED : AUTORG, DATSTD, DESTAT, FORIER, KSD, PARZ,
C            PRINTA, QHLIN, QTOFQ, QUICK, WSPACE, CLPLT1, FCODEA, ICODEA,
C          PPLOT
```

```
C   FUNCTIONS CALLED : AREST
C*****************************************************************
      COMMON/SAMPLE/X(552),XS(552),WXS(552),QN(552),U(552),
     &              DHAT(552),CWXS(552),F(552),FQ(552)
      DIMENSION CAT(50),RESVEC(50),RA(50),WK1(1104),WK2(1104)
      DIMENSION NAME(20),LAB1(20),          LAB3(20),
     +          LAB5(20),LAB6(20),LAB7(20),LAB8(20),LAB9(20)
      DIMENSION LAB10(20),LAB11(20),LAB12(20)
      DIMENSION LAB13(20),LAB14(20),LAB15(20),LAB16(20)
      DIMENSION LAB17(20),LAB18(20)
      DIMENSION NAMDH(2),NAMQN(2),NAMU(2),NAMXS(2),NAMWXS(2),
     +          NAMCWS(2),NAMFQ(2),ICASE(2)
      DIMENSION LABEL(20,3),NCHAR(3),FAC(3),IBLK(20,3)
      COMPLEX A(50),ALPHA(50),PHI(50),RESVAR
      INTEGER PPLT
      DATA IBLK/60*4H    /
      DATA ICAT,BLK,ASTER/4HCAT ,1H ,1H*/
      DATA NCHAR,FAC/80,8,80,8,33,6,25,2,0/
      DATA LAB1/4HQUAN,4HTILE,4H.  N,4HATUR,4HAL M,4HINIM,4HUM: ,
     +4HNONE,12*4H    /
      DATA LAB3/4HNORM,4HALIZ,4HED D,4HATA ,16*4H    /
      DATA LAB5/4HPLOT,4H OF ,4HSTAN,4HDARD,4HIZED,4HDATA,
     +14*4H    /
      DATA LAB6/4HRAW ,4HQUAN,4HTILE,4H DEN,4HSITY,15*4H    /
      DATA LAB7/4HSIGM,4HAO =,4H AVE,4HRAGE,4H WEI,4HGHTE,
     +4HD SP,4HACIN,4HG = ,4*4H    ,4HRANG,4HE = ,5*4H    /
      DATA LAB8/4HPLOT,4H OF ,4HRAW ,4HDENS,4HITY ,2*4H    ,
     +4H CAS,4HE   ,11*4H    /
      DATA LAB9/4HD+ =,2*4H    ,4H AT ,4HU = ,2*4H    ,4H.  D,
     +4H- = ,2*4H    ,4H AT ,4HU = ,7*4H    /
      DATA LAB10/4HRAW ,4HDIST,4HTRIB,4HUTIO,4HN D(,4HU)  ,
     +       3*4H    ,4H CAS,4HE   ,9*4H    /
      DATA LAB11/4HORDE,4HR = ,18*4H    /
      DATA LAB12/4HDHAT        ,4H PLO,4HTTED,4H AGA,4HINST,
     +4H RAW,4H D (,4H*)  ,3*4H    ,4H CAS,4HE   ,7*4H    /
      DATA LAB13/              4HDENS,4HITY-,4HQUAN,4HTILE,
     +4H FUN,4HCTIO,4HN   ,2*4H    ,4H CAS,4HE   ,9*4H    /
      DATA LAB14/4H    ,4H    ,4HQ(U),4H USI,4HNG L,4HINEA,
     +4HR IN,4HTERP,4HOLAT,4HION ,10*4H    /
      DATA LAB15/              4HRAW ,4HFQ  ,  18*4H    /
      DATA LAB16/              4HQ(U),4H USI,4HNG L,4HINEA,
     +4HR IN,4HTERP,4HOLAT,4HION ,4HFOR ,4HGROU,4HPED ,4HDATA,
     +8*4H    /
      DATA LAB17/4H   P,4HOSIT,4HIVE ,4HFOUR,4HIER ,
     +4HTRAN,4HSFOR,4HM OF,4H WEI,4HGHTE,4HD SP,4HACIN,4HGS  ,
```

```
      +7*4H    /
       DATA LAB1B/4H   A,4HLTER,4HNATI,4HNG S,4HIGNS,
      +4H OF ,4HFOUR,4HIER ,4HTRAN,4HSFOR,4HN OF ,4H WEI,4HGHTE,
      +4HD SP,4HACIN,4HGS  ,4*4H    /
       DATA XONE/.0000000001/
       DATA NTAP9/9/
       DATA NAMQN,NAMU/4H   Q,4HN(U),4H    ,4H   U/
       DATA NAMXS/4H  GA,4HP(U)/
       DATA NAMUXS,NAMCUS/4HFOQO,4HQ(U),4H    ,4HD(U)/
       DATA NAMDH,NAMFQ/4H DHA,4HT(U),4H   F,4HQ(U)/
       NSIGN=N
       N=IABS(N)
       NN1=N*2+1
       NN3=NN1+2
       NP1=N+1
       NP2=N+2
       IF(IFIRST.NE.1)GO TO 20
       NQ=N
       IF(IOPT1.GE.1)GO TO 120
C   OBTAIN DESCRIPTIVE STATISTICS FOR UNGROUPED DATA
  444 IF(NOBS.EQ.0)NOBS=N
       U(1)=0.
       DO 10 J=1,N
       U(J+1)=(FLOAT(J)-.5)/FLOAT(N)
   10 CONTINUE
       CALL QUICK(N,X)
       IF(X0.NE.XONE)CALL FCODEA(NTAP9,X0,8H(F8.0)  ,LAB1(8))
       IF(NSIGN.GT.0)GO TO 14
       LAB1(12)=ICASE(1)
       LAB1(13)=ICASE(2)
   14 CONTINUE
       CALL DESTAT(X,N,NAME,IUNIT,LAB1,PPLT,WK1,WK2,NN3,FAC)
       XM=X(1)
       IF(X0.NE.XONE)XM=X0
       RANGE=X(N)-XM
       IF(IOPT2.NE.1)GO TO 116
C   IF IOPT2=1, STANDARDIZE DATA TO (0,1), OBTAIN DESCRIPTIVE
C                STATISTICS AND, IF FINC.GT.0., COMPUTE Q VIA
C                LINEAR INTERPOLATION
       CALL DATSTD(X,XM,NP1,QN)
       CALL DESTAT(QN(2),N,NAME,IUNIT,LAB3,PPLT,WK1,WK2,NN3,FAC)
       IF(FINC.EQ.0.)GO TO 16
       NOFNOF=1./FINC
       NOFNOF=MAX0(N,NOFNOF)+1
       CALL QHLIN(QN(2),N,FINC,U,FQ,IOPT1,NQ,QN(1),XINT,NOFNOF)
```

```
      NQP1=NQ+1
      DO 114 I=1.NQP1
  114 QN(I)=FQ(I)
      DO 15 I=1.4
   15 LAB14(I+10)=LAB3(I)
      DO 115 I=1.20
  115 LAB5(I)=LAB14(I)
   16 NQP1=NQ+1
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMQN.NAME.LAB5.LABEL.QN.U.NQP1.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U.QN.NQP1.11.100.LABEL.NCHAR.FAC.NP2)
C-----------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U.QN.NQP1.NAMU.NAMQN.NAME.WK1.0.0.BLK.LAB5)
      GO TO 19
  116 IF(FINC.EQ.0.)GO TO 18
      MOFNOF=1./FINC
      MOFNOF=MAX0(N.MOFNOF)+1
C   IF FINC.GT.0.. COMPUTE Q VIA LINEAR INTERPOLATION
      CALL QHLIN(X.N.FINC.U.QN.IPT1.NQ.XM.XINT.MOFNOF)
      NQP1=NQ+1
      IF(PPLT.NE.1)
     +CALL PPLOT(U.QN.NQP1.NAMU.NAMQN.NAME.WK1.0.0.BLK.LAB14)
      GO TO 19
C   IF FINC.EQ.0.. COMPUTE Q
   18 DO 119 I=1.N
      I2=N-I+2
      I1=N-I+1
  119 QN(I2)=X(I1)
      QN(1)=XM
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMQN.NAME.LAB1.LABEL.QN.U.NP1.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U.QN.NP1.11.100.LABEL.NCHAR.FAC.NP2)
C-----------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U.QN.NP1.NAMU.NAMQN.NAME.WK1.0.0.BLK.LAB1)
   19 CONTINUE
      GO TO 20
C   COMPUTE Q FOR GROUPED DATA
  120 CONTINUE
      MOFNOF=1./FINC
      MOFNOF=MAX0(N.MOFNOF)+1
      RANGE=FLOAT(N)*XINT
      CALL QHLIN(X.N.FINC.U.QN.IOPT1.NQ.XO.XINT.MOFNOF)
      NQP1=NQ+1
```

```
      LAB16(14)=ICASE(1)
      LAB16(15)=ICASE(2)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMQN.NAME.LAB16.LABEL.QN.U.NQP1.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U.QN.NQP1.11.100.LABEL.NCHAR.FAC.NP2)
C------------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U.QN.NQP1.NAMU.NAMQN.NAME.WK1.0.0.BLK.LAB16)
      NTEMP=NQP1*2+3
      CALL DESTAT(QN.NQP1.NAME.IUNIT.LAB1.PPLT.WK1.WK2.NTEMP.FAC)
   20 CONTINUE
      IF(NSIGN.LE.0)GO TO 999
C   COMPUTE AND PLOT RAW SPACINGS (=Q) AND RAW FQ (=1./Q)
      NQP1=NQ+1
      CALL QTOFQ(QN.U.NQP1.XS.FQ)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMXS.NAME.LAB6.LABEL.XS.U(2).NQ.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U(2).XS.NQ.11.100.LABEL.NCHAR.FAC.NP2)
C------------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U(2).XS.NQ.NAMU.NAMXS.NAME.WK1.0.0.BLK.LAB6)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMFQ.NAME.LAB15.LABEL.FQ.U(2).NQ.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U(2).FQ.NQ.11.100.LABEL.NCHAR.FAC.NP2)
C------------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U(2).FQ.NQ.NAMU.NAMFQ.NAME.WK1.0.0.BLK.LAB15)
C   COMPUTE AND PLOT WEIGHTED SPACINGS FOR CASE ICASE
      NQP1=NQ+1
      CALL WSPACE(WXS.CWXS.NQP1.FQ.IDQH.U.SW)
      CALL FCODEA(NTAP9.SW.8H(F10.4) .LAB7(10))
      CALL FCODEA(NTAP9.RANGE.8H(F10.4) .LAB7(16))
      LAB8(6)=ICASE(1)
      LAB8(7)=ICASE(2)
      IF(PPLT.NE.1)
     +CALL PPLOT(U(2).WXS.NQ.NAMU.NAMWXS.LAB8.WK1.0.0.BLK.LAB7)
C   PLOT CUMULATIVE WEIGHTED SPACINGS WITH D+ AND D-
      NQP1=NQ+1
      CALL KSD(CWXS.U.NQ.DM.UM.DP.UP)
      CALL FCODEA(NTAP9.DP.8H(F7.4)  .LAB9(2))
      CALL FCODEA(NTAP9.UP.8H(F7.4)  .LAB9(6))
      CALL FCODEA(NTAP9.DM.8H(F7.4)  .LAB9(10))
      CALL FCODEA(NTAP9.UM.8H(F7.4)  .LAB9(14))
      LAB10(8)=ICASE(1)
      LAB10(9 )=ICASE(2)
```

```
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMCUS.NAME.LAB10.LABEL.CUXS.U.NQP1.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U.CUXS.NQP1.11.100 .LABEL.NCHAR.FAC.NP2)
C-----------------------------------------------------------------
      UK1(1)=U(1)
      UK2(1)=CUXS(1)
      DO 43 I=2.4
      II=NQP1+(I-2)
      UK1(I)=U(II)
      UK2(I)=CUXS(II)
 43   CONTINUE
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL GPLOT(UK1.UK2.2.100.100)
C-----------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U.CUXS.NQP1.NAMU.NAMCUS.LAB10.UK1.0.0.BLK.LAB9)
C   COMPUTE FOURIER TRANSFORM OF WEIGHTED SPACINGS
      CALL FORIER(UXS.U(2).NQ.A.M)
      CALL PRINTA(A.M.IUNIT)
      RA(M)=REAL(A(1))
      MM1=M-1
      DO 40 I=1.MM1
      II=M-I
      RA(II)=REAL(A(I+1))
      RA(M+I)=RA(II)
 40   CONTINUE
      MM2=M-3
      NDIM=2*MM2+9
      CALL SPLAY(RA.MM2.2.N.LAB17.UK1.UK2.NDIM)
      RA(M)=REAL(A(1))
      DO 41 I=1.MM1
      II=M-I
      RA(II)=((-1.0)**I)*REAL(A(I+1))
      RA(M+I)=RA(II)
 41   CONTINUE
      CALL SPLAY(RA.MM2.2.N.LAB18.UK1.UK2.NDIM)
C   OBTAIN AUTOREGRESSIVE ESTIMATOR FOR ORDERS 1 TO M-1
      DO 30 K=1.MM1
      IORD=K
      CALL ICODEA(NTAP9.IORD.8H(I2)    .LAB11(3))
      KP1=K+1
      CALL AUTORG(A.KP1.M.ALPHA.PHI.RESVAR)
      RESVEC(K)=RESVAR
      NFAC=1
      IF(NQ.LE.50)NFAC=4
```

```
      NIT=NFAC*NQ
      NITP1=NIT+1
      OPTKHN=REAL(RESVAR)
      DO 22 I=1.NITP1
         XI=FLOAT(I-1)/FLOAT(NIT)
         F(I)=AREST(XI.K.OPTKHN.ALPHA)
         IF(MOD(I-1.NFAC).NE.0)GO TO 22
         J=(I-1)/NFAC+1
         FQ(J)=FQFNC(XI.IDQH)/(F(I)*SW)
   22    CONTINUE
      DHAT(1)=0.
      DO 24 I=2.NQP1
         DHAT(I)=DHAT(I-1)+F(I)/FLOAT(NQ)
         IF(NQ.GT.50)GO TO 24
         J=NFAC*(I-2)+1
         FWIN=F(J)+4.*F(J+1)+2.*F(J+2)+4.*F(J+3)+F(J+4)
         DHAT(I)=DHAT(I-1)+FWIN/(12.*FLOAT(NQ))
   24    CONTINUE
      LAB12(9 )=ICASE(1)
      LAB12(10)=ICASE(2)
      DO 200 I=13.15
  200 LAB12(I+1)=LAB11(I-12)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMDH.NAME.LAB12.LABEL.DHAT.U.NQP1.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U.DHAT.NQP1.11.100.LABEL.NCHAR.FAC.NP2)
C----------------------------------------------------------------
      CWXS(NQP1+1)=DHAT(NQP1+1)
      CWXS(NQP1+2)=DHAT(NQP1+2)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL GPLOT(U.CWXS.NQP1.104.-101.IBLK.NCHAR.FAC.NP2)
C----------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U.DHAT.NQP1.NAMU.NAMDH.NAME.CWXS.0.1.ASTER.LAB12)
      LAB13(8 )=ICASE(1)
      LAB13(9 )=ICASE(2)
      DO 201 I=12.14
  201 LAB13(I)=LAB11(I-11)
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$-
      IF(PPLT.NE.0)CALL LABL(NAMFQ.NAME.LAB13.LABEL.FQ.U(2).NQ.FAC)
      IF(PPLT.NE.0)CALL GPLOT(U(2).FQ.NQ.11.100.LABEL.NCHAR.FAC.NP2)
C----------------------------------------------------------------
      IF(PPLT.NE.1)
     +CALL PPLOT(U(2).FQ.NQ.NAMU.NAMFQ.NAME.WK1.0.0.BLK.LAB13)
   30 CONTINUE
C   COMPUTE 'BEST' ORDER VIA CAT CRITERION AND PRINT
```

```
      WRITE(IUNIT.901)ICASE
      WRITE(IUNIT.904)
      DO 42 I=1.MM1
      TRVAR=ALOG(RESVEC(I))
      AIC=TRVAR+((2.0*FLOAT(I))/FLOAT(N))
      WRITE(IUNIT.905)I.RESVEC(I).TRVAR.AIC
42    CONTINUE
      CALL PARZ(RESVEC.M-1.NOBS.CAT.IOPTOR)
      WRITE(IUNIT.902)ICASE
      CALL CLPLT1(CAT.M-1.1.ICAT.81.1)
      WRITE(IUNIT.903)IOPTOR
 999  CONTINUE
 901  FORMAT(///.5X.'RESIDUAL VARIANCES BY AR ORDER'.
     +10X.2A4.' CASE'///)
 902  FORMAT(///.5X.'CAT CRITERION'.10X.2A4.' CASE'///)
 903  FORMAT(///.5X.'OPTIMAL ORDER = '.I2)
 904  FORMAT(//.14X.'I'.6X.'RVAR'.3X.'LN(RVAR)'.3X.
     +'LN(RVAR)+(2I/T)=AIC'./)
 905  FORMAT(14X.I1.F10.4.1X.F10.4.8X.F10.4)
      RETURN
      END
```

```
      SUBROUTINE PARZ(RVAR.M.N.CAT.NORD)
C*****************************************************
C   SUBROUTINE TO DETERMINE THE ORDER OF AN AUTOREGRESSIVE
C   PROCESS BY PARZENS CAT CRITERIA
C   INPUT :
C           M.RVAR(1).....RVAR(M) : STANDARDIZED RES VAR
C           FOR ORDERS 1 THRU M.
C           N : SAMPLE SIZE
C   OUTPUT :
C           NORD : DETERMINED ORDER
C           CAT(1).....CAT(M)
C   SUBROUTINES CALLED : MIN
C*****************************************************
      DIMENSION RVAR(M).CAT(M)
      ON=FLOAT(N)
      DO 1 I=1.M
         C=0.
         DO 2 J=1.I
    2    C=C+((1.-(FLOAT(J)/ON))/RVAR(J))
         C=C/ON
    1 CAT(I)=C-((1.-(FLOAT(I)/ON))/RVAR(I))
      CALL MIN(CAT.M.CATMI.NORD)
      IF(CATMI.GT.-1.) NORD=0
      RETURN
      END
```

```
      SUBROUTINE PPLOT(X,Y,NN1,LX,LY,LC,Z,IBOX,IZ,ZCHAR,LI)
C************************************************************
C     ROUTINE TO DISPLAY PRINTER QUANTILE-BOX PLOT
C     INPUT :
C           X - VECTOR CONTAINING THE VALUES J/(NN1+1) WHERE
C               J=1,2,3,....,NN1
C           Y - VECTOR OF SIZE NN1 TO BE PLOTTED
C           LX,LY - VECTORS OF SIZE 2 CONTAINING THE LABELS
C                   FOR X AND Y RESPECTIVELY
C           LC - VECTOR OF SIZE 20 CONTAINING THE CAPTION FOR
C                THE PLOT
C           Z - OPTIONAL VECTOR OF SIZE NN1 TO BE PLOTTED
C           IBOX - EQUAL ZERO IF BOX PLOTS ARE NOT WANTED
C           IZ - EQUAL 1 IF VECTOR Z IS TO BE PLOTTED
C           ZCHAR - PLOTTING CHARACTER TO BE USED FOR THE
C                   Z VECTOR. MUST BE DIFFERENT FROM 0.
C           LI - OPTIONAL LABEL TO FOLLOW CAPTION
C***********************************************************
      DIMENSION PARRAY(57,61),CHAR(10),X(NN1),Y(NN1),LX(2),LY(2),
     &          LC(20),XJ(11),Z(NN1),LI(20)
      DATA BLANK,DASH,PLUS,CAPI/1H ,1H-,1H+,1HI/
      DATA CHAR/1H0,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HN/
      CALL MIN(Y,NN1,VMIN,IER)
      CALL MAX(Y,NN1,VMAX,IER)
      IF(IZ.NE.1)GO TO 1
      CALL MIN(Z,NN1,ZMIN,IER)
      CALL MAX(Z,NN1,ZMAX,IER)
      IF(ZMIN.LT.VMIN)VMIN=ZMIN
      IF(ZMAX.GT.VMAX)VMAX=ZMAX
 1    CONTINUE
      VINC=(VMAX-VMIN)/56.0
      IF(VINC.EQ.0.0)GO TO 99
      DO 10 I=1,57
      PARRAY(I,1)=CAPI
      DO 5 J=2,61
      PARRAY(I,J)=BLANK
 5    CONTINUE
 10   CONTINUE
      DO 20 J=1,61
      PARRAY(57,J)=DASH
 20   CONTINUE
      J=0
      XJ(1)=0.0
      XJ(11)=1.0
      DO 25 I=1,57,6
```

```
      J=J+1
      XJ(J)=0.1*FLOAT(J-1)
      PARRAY(58-I,1)=PLUS
      PARRAY(57,I)=PLUS
25    CONTINUE
      PARRAY(57,61)=PLUS
      YMED=(Y(NN1/2+MOD(NN1,2))+Y(NN1/2+1))/2.0
      IYMED=(YMED-VMIN)/VINC+1.0
      IF(IBOX.EQ.0)GO TO 60
      XL=1.0/32.0
      DO 40 K=2,4
      XL=XL*2.0
      XU=1.0-XL
      CALL QFIND(X,Y,NN1,XL,YL)
      CALL QFIND(X,Y,NN1,XU,YU)
      IEND=(YU-VMIN)/VINC+1.0
      ISTART=(YL-VMIN)/VINC+1.0
      JSTART=XL*60.0+1.0
      JEND=XU*60.0+1.0
      DO 30 I=ISTART,IEND
      PARRAY(58-I,JSTART)=CAPI
      PARRAY(58-I,JEND)=CAPI
30    CONTINUE
      DO 35 J=JSTART,JEND
      PARRAY(58-ISTART,J)=DASH
      PARRAY(58-IEND,J)=DASH
35    CONTINUE
40    CONTINUE
      JS=JSTART+1
      JE=JEND-1
      DO 41 J=JS,JE
      PARRAY(57-IYMED,J)=DASH
41    CONTINUE
60    CONTINUE
      DO 45 K=1,NN1
      INDX=X(K)*60.0+1.0
      INDY=(Y(K)-VMIN)/VINC+1.0
      IF(IZ.NE.1)GO TO 70
      INDZ=(Z(K)-VMIN)/VINC+1.0
      PARRAY(58-INDY,INDX)=CHAR(1)
      PARRAY(58-INDZ,INDX)=ZCHAR
      IF(INDZ.EQ.INDY)PARRAY(58-INDY,INDX)=CHAR(10)
      GO TO 45
70    CONTINUE
      DO 42 I=1,10
```

```
      IF(PARRAY(58-INDY,INDX).EQ.CHAR(I))GO TO 43
   42 CONTINUE
      PARRAY(58-INDY,INDX)=CHAR(1)
      GO TO 45
   43 IF(I.EQ.10)I=I-1
      PARRAY(58-INDY,INDX)=CHAR(I+1)
   45 CONTINUE
      WRITE(6,100)LC,LI
      DO 55 I=1,56
      IF(PARRAY(I,1).EQ.PLUS)GO TO 50
      WRITE(6,105)(PARRAY(I,J),J=1,61)
      GO TO 55
   50 YXALUE=(1.0-FLOAT(I)/57.0)*(VMAX-VMIN)+VMIN
      WRITE(6,110)YXALUE,(PARRAY(I,J),J=1,61)
   55 CONTINUE
      WRITE(6,110)VMIN,(PARRAY(57,J),J=1,61)
      WRITE(6,115)XJ
      WRITE(6,120)LX,LY
      GO TO 999
   99 CONTINUE
      WRITE(6,199)LC
      WRITE(6,200)
  200 FORMAT(' X VALUES')
      WRITE(6,205)(X(I),I=1,NN1)
  205 FORMAT(1X,10F10.5)
      WRITE(6,210)
  210 FORMAT(' Y VALUES')
      WRITE(6,205)(Y(I),I=1,NN1)
  199 FORMAT(' ERROR IN PPLOT FOR THE PLOT OF ',20A4)
  999 RETURN
  100 FORMAT(1H1,20A4,/1X,20A4)
  105 FORMAT(13X,61A1)
  110 FORMAT(1X,F10.3,2X,61A1)
  115 FORMAT(12X,11(F3.1,3X))
  120 FORMAT(15X,'ABSCISSA IS ',2A4,4X,' , ORDINATE IS ',2A4)
      END
```

```
      SUBROUTINE PRINTA(A,N,IUNIT)
C****************************************************
C   SUBROUTINE TO COMPUTE AND PRINT THE SQUARE MODULUS OF THE
C   COMPLEX-VALUED FOURIER TRANSFORMS A(1).....A(N)
C   INPUT :
C        A,N
C        IUNIT : UNIT ON WHICH OUTPUT IS DESIRED
C   SUBROUTINES CALLED : NONE
C****************************************************
      DIMENSION AM(50)
      COMPLEX A(N)
      WRITE(IUNIT,1000)
      DO 10 I = 1,N
   10 AM(I) = REAL(CONJG(A(I))*A(I))
      CALL COLPLT(AM,N,1,4HPHI2)
      RETURN
 1000 FORMAT(///,5X,'MODULUS SQUARED OF FOURIER COEFFICIENTS'///)
      END
```

```
      SUBROUTINE SIGHAT(P,Y,N,USIGMA,VSIGMA,SIGMA)
C************************************************************
C   SUBROUTINE COMPUTES SIGMAHAT BASED ON NORMAL FOQO
C   INPUT:  P : TRUNCATION PT
C           Y,N : DATA VECTOR OF LENGTH N
C           USIGMA,VSIGMA : WEIGHT AND VARIANCE AS COMPUTED BT STEP1
C   OUTPUT: SIGMA
C   SUBROUTINES CALLED : FQFNC
C************************************************************
      DIMENSION Y(N)
      QPHI=0.
      KP=INT(N*P)
      J=KP+2
      L=N-KP-1
      DO 10 K=J,L
      X=FLOAT(K-1)/FLOAT(N)
      Y2=FLOAT(K)/FLOAT(N)
      X1=FQFNC(X,1)
      Y1=FQFNC(Y2,1)
   10 QPHI=Y(K)*(X1-Y1)+QPHI
      Z1=FLOAT(KP+1)/FLOAT(N)
      WT1=FQFNC(P,1)-FQFNC(Z1,1)
      Z2=FLOAT(N-KP-1)/FLOAT(N)
      WT2=FQFNC(Z2,1)-FQFNC(1.-P,1)
      QPHI=QPHI+WT1*Y(KP+1)+WT2*Y(N-KP)
      SIGMA=(2.*QPHI+(Y(KP+1)-Y(N-KP))*USIGMA)*VSIGMA
      RETURN
      END
```

```
      SUBROUTINE SPLAY(R.MAXP.MAXQ.NOBS.LAB.S.G.ND)
C*************************************************************
C   SUBROUTINE TO CALCULATE AND DISPLAY THE SHIFTED S ARRAY OF
C   GRAY ET. AL. FOR MAXIMUM AUTOREGRESSIVE ORDER MAXP AND
C   MAXIMUM MOVING AVERAGE ORDER MAXQ (LE.7).
C   INPUT :
C            R : ARRAY CONTAINING F(-MAXP-MAXQ).....F(0).....
C                F(MAXP+MAXQ)  (USUALLY F(J)=((-1)**J)*CORR(J)
C                OR F(J)=CORR(J))
C            MAXP : MAXIMUM AUTOREGRESSIVE ORDER
C            MAXQ : MAXIMUM MOVING AVERAGE ORDER
C            NOBS : SAMPLE SIZE
C            LAB(1).....LAB(20) : 80 CHARACTER HOLLERITH ARRAY
C                   CONTAINING LABEL FOR PRINTED OUTPUT (EACH
C                   ELEMENT OF LAB HAS 4 CHARACTERS)
C            ND : DIMENSION OF R.S.G
C   OUTPUT :
C            R : CONTAINS R(-MAXP-MAXQ).....R(MAXQ) FROM
C                (MAXP) TH ITERATION
C            S : CONTAINS S(-MAXP-MAXQ).....S(MAXP) FROM
C                (MAXP) TH ITERATION
C           G : CONTAINS G FOR MAXP TH ITERATION
C   SUBROUTINES CALLED : NONE
C*************************************************************
      DIMENSION R(ND).S(ND).LAB(20).SP(100).L1(10).L2(10).L3(50).G(ND)
      DIMENSION PIK(50)
      DATA NOUT.ONE.EPS/6.1.0.1.E-15/
      DATA IDASH/4H----/
      DATA BIGNUM/1000./
      INIT=MAXO(0.MAXQ-7)
C   PRINT HEADING :
      WRITE(NOUT.1) NOBS.(LAB(I).I=1.16)
    1 FORMAT(20X.11HS-PLAY T = .I4.2X.16A4)
      L2(1)=INIT
      MAXQMI=MAXQ-INIT
      DO 2 I=1.MAXQMI
      L1(I)=-(INIT+I)
    2 L2(I+1)=INIT+I
      NDASH=2*(2*MAXQMI+1)
      DO 6 I=1.NDASH
      L3(I)=IDASH
    6 CONTINUE
      WRITE(NOUT.7) (L3(I).I=1.NDASH)
    7 FORMAT(4X.3H---.30A4)
      WRITE(NOUT.3) (L1(I).I=1.MAXQMI)
```

```
    3  FORMAT(3X,1HP,1X,3HP2D,2X,4HCATA,4X,4HGSP1,3X,9X,7(I3,11X))
       MAXQP1=MAXQ+1
       NQMI=MAXQMI+1
       WRITE(NOUT,4) (L2(I),I=1,NQMI)
    4  FORMAT(5X,1HV,2X,2HPL,1X,1HM,3X,4HSPEC,2X,4HGSP2,3X,7(I3,11X),I3)
       WRITE(NOUT,7) (L3(I),I=1,NDASH)
C   INITIALIZATIONS :
       ON=FLOAT(NOBS)
       MPPQ=MAXP+MAXQ
       NN=2*MAXQ+2*MAXP+1
       MAXQM2=MAXQ-2
       DO 10 I=1,NN
       S(I)=ONE
   10  CONTINUE
       V=ONE
       DO 12 I=1,MPPQ
   12  G(I)=ONE
       MXP1=MPPQ+1
       G(MXP1)=R(MXP1)
       DO 13 I=1,MPPQ
       MXP1PI=MXP1+I
   13  G(MXP1PI)=G(MXP1PI-1)+R(MXP1PI)+R(MXP1-I)
C   FIX K (AUTOREGRESSIVE ORDER) :
       DO 100 K=1,MAXP
C   FORM NEW S AND G (NOTE IF ABS(R(J)).LT.EPS, G(J)=G(J))
       NNMK=NN-K
       DO 30 J=1,NNMK
       IF(ABS(R(J)).LT.EPS) GO TO 20
       FAC=(R(J+1)/R(J))-ONE
       S(J)=S(J+1)*FAC
       IF(ABS(FAC).LT.EPS) GO TO 15
       G(J)=G(J)-((G(J+1)-G(J))/FAC)
       GO TO 30
   15  CONTINUE
       WRITE(NOUT,16) FAC
   16  FORMAT(10X,29HEQUAL SUCCESSIVE R, FACTOR = ,E15.7)
       G(J)=G(J)-((G(J+1)-G(J))*BIGNUM)
       GO TO 30
   20  CONTINUE
       S(J)=S(J+1)
   30  CONTINUE
C   PRINT DESIRED VALUES OF S (NOTE : SKSTAR(J)
C   =SK(J-K+1)=S(MAXP+MAXQ+J-K+2))
       DO 40 J=1,MAXQ
   40  SP(J)=S(MPPQ-J-K+2)
```

```
      N1=NPPQ-K+2
      M1=N1+INIT
      M2=M1+MINO(7.MAXQ)
C    FIND P2D AND M (PIK(1)=TRUE PIK(0).
C    SKSTAR(J)=S(N1+J),J=0.....MAXQ, SKSTAR(-J)=
C    SP(J),J=1,....,MAXQ)
      IF(MAXQ.GE.3) GO TO 41
      P2D=.99
      M10=-1
      GO TO 44
   41 CONTINUE
      DO 42 J=1,MAXQ
   42 PIK(J)=S(N1+J-1)/SP(J)
      P2D=100.
      M10=-2
      DO 43 J=1,MAXQM2
      C=ABS(PIK(J)-2.0*PIK(J+1)+PIK(J+2))
      IF(C.GT.P2D) GO TO 43
      P2D=C
      M10=J-1
   43 CONTINUE
   44 CONTINUE
      PARL=ABS(S(N1)/SP(1))
      V=V*(ONE-PARL*PARL)
      CATA=ABS(V)
      CATA=ALOG(CATA)+(2.0*FLOAT(K)/ON)
      SPEC=V/(S(N1)*S(N1))
      GSP1=G(MXP1+1+INIT)
      GSP2=G(MXP1+2+INIT)
      JJ=INIT+1
      WRITE(NOUT,50) K,P2D,CATA,GSP1,(SP(J),J=JJ,MAXQ)
   50 FORMAT(2X,I2,1X,F3.2,F6.2,2X,F6.2,10X,6(F7.2,7X),F7.2)
      WRITE(NOUT,60) V,PARL,M10,SPEC,GSP2,(S(J),J=M1,M2)
   60 FORMAT(2X,F4.3,1X,F3.2,I2,1X,F6.2,F6.2,7(F7.2,7X),F7.2)
C    FIND NEW R (IF.K.LT.MAXP) :
      IF(K.EQ.MAXP) GO TO 100
      NNMKM1=NN-K-1
      DO 80 J=1,NNMKM1
      IF(ABS(S(J)).LT.EPS) GO TO 70
      R(J)=R(J+1)*((S(J+1)/S(J))-ONE)
      GO TO 80
   70 CONTINUE
      R(J)=R(J+1)
   80 CONTINUE
  100 CONTINUE
```

```
      SUBROUTINE STEP1(P,UMEAN,USIGMA,VMEAN,VSIGMA)
C***********************************************************
C   SUBROUTINE COMPUTES WEIGHTS AND VARIANCE FOR TRUNCATION PT P
C   TO BE USED IN FINDING MUHAT AND SIGHAT BASED ON NORMAL FOQO
C   INPUT : P
C   OUTPUT: UMEAN,USIGMA,VMEAN,VSIGMA
C   SUBROUTINES CALLED : MDNRIS(IMSL),FQFNC
C**********************************************************
      IF(P.EQ.0.)GO TO 20
      CALL MDNRIS(P,T,IER)
      FOQO=FQFNC(P,1)
      UMEAN=FOQO*(T+FOQO/P)
      USIGMA=T*UMEAN-FOQO
      VMEAN=1./(1.-2.*P+2.*UMEAN)
      VSIGMA=1./(2.*(1.-2.*P+2.*T*FOQO+T*USIGMA))
      GO TO 30
   20 CONTINUE
      UMEAN=0.
      USIGMA=0.
      VMEAN=1.
      VSIGMA=.5
   30 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QBXPLT(X.N.EX.EY.NN3.LC.IOPT.FAC.NC)
C************************************************************
C   SUBROUTINE OBTAINS CALCOMP PLOT OF QUANTILE BOX PLOT, WITH
C   IOPT BOXES DRAWN AND A CONFIDENCE INTERVAL FOR THE MEDIAN DRAWN
C   INPUT:  X.N   : DATA VECTOR OF LENGTH N
C           EX.EY : DUMMY VECTORS OF LENGTH NN3=N*2+3
C           IOPT : = NUMBER OF BOXES DESIRED DRAWN AT H,E,D,ETC.
C           FAC : AS IN SPLOT
C           LC : CAPTION LABEL FOR THE PLOT
C           NC : NUMBER OF CHARACTERS IN LC
C************************************************************
      DIMENSION X(N).EX(NN3).EY(NN3).LC(20).BX(7).BY(7).
     &          FAC(3).LA(20).LO(20).LABEL(20.3).NCHAR(3)
      DATA LA/4H    .4H    .4HQ(U).4H USI.4HNG L.4HINEA.
     +4HR IN.4HTERP.4HOLAT.4HION .10*4H    /
      DATA LO/4H QU .19*4H    /
      DATA NCHAR/80.4.80/
      NN1=NN3-2
      CALL EXPAND(N.X.NN1.EY)
      DO 5 I=1.NN1
      EX(I)=FLOAT(I)/FLOAT(NN1+1)
 5    CONTINUE
      DO 6 I=1.20
      LABEL(I.1)=LC(I)
      LABEL(I.2)=LO(I)
      LABEL(I.3)=LA(I)
 6    CONTINUE
      CALL SCALE(EY.FAC(2).NN1.1.10.0)
      EX(NN1+1)=0.00
      EX(NN3)=0.125
      NN1P2=NN1+2
      CALL GPLOT(EX.EY.NN1.4.100.LABEL.NCHAR.FAC.NN1P2)
      IF(IOPT.EQ.0)GO TO 99
      CALL GPLOT(EX.EY.NN1.104.102.LABEL.NCHAR.FAC.NN1P2)
      BY(6)=EY(NN1+1)
      BX(6)=EX(NN1+1)
      BY(7)=EY(NN3)
      BX(7)=EX(NN3)
      YMED=(EY(NN1/2+MOD(NN1.2))+EY(NN1/2+1))/2.0
      CALL QFIND(EX.EY.NN1.0.25.BY(1))
      BX(1)=0.25
      BY(2)=BY(1)
      CALL QFIND(EX.EY.NN1.0.75.BY(3))
      BX(2)=0.75
      BX(3)=0.75
```

```
      BX(4)=0.25
      BX(5)=0.25
      BY(4)=BY(3)
      BY(5)=BY(1)
      CALL GPLOT(BX,BY,5,101,100,LABEL,NCHAR,FAC,NN1P2)
      HH=(BY(3)-BY(1))/SQRT(FLOAT(N))
      BY(1)=YMED
      BY(2)=YMED
      BX(3)=0.5
      BX(4)=0.5
      BX(5)=0.5
      BY(3)=YMED
      BY(4)=YMED+HH
      BY(5)=YMED-HH
      CALL GPLOT(BX,BY,5,101,100,LABEL,NCHAR,FAC,NN1P2)
      NUM=IOPT-1
      DO 10 I=1,NUM
      QL=1.0/(2.0*(I+2))
      QU=1.0-QL
      CALL QFIND(EX,EY,NN1,QL,BY(1))
      BX(1)=QL
      BY(2)=BY(1)
      CALL QFIND(EX,EY,NN1,QU,BY(3))
      BX(2)=QU
      BX(3)=QU
      BX(4)=QL
      BX(5)=QL
      BY(4)=BY(3)
      BY(5)=BY(1)
      CALL GPLOT(BX,BY,5,101,100,LABEL,NCHAR,FAC,NN1P2)
10    CONTINUE
99    RETURN
      END
```

```
      SUBROUTINE QFIND(X,Y,N,QUANT,QVALUE)
C************************************************************
C     SUBROUTINE QFIND
C            PURPOSE:  TO FIND THE PERCENTILE VALUE OF Y
C                      AT QUANT.
C            INPUT:
C                  X - VECTOR OF SIZE N CONTAINING THE VALUES
C                      J/(N+1) WHERE J=1,2,3,....,N
C                  Y - VECTOR OF SIZE N
C                  N - NUMBER OF VALUES IN Y
C                  QUANT - QUANTILE VALUE
C                  QVALUE - OUTPUT THE VALUE OF Y AT QUANT
C************************************************************
      DIMENSION X(N),Y(N)
      DO 30 I=1,N
      IF(X(I).GE.QUANT)GO TO 31
 30   CONTINUE
      IF(X(I).LE.QUANT)GO TO 35
 31   IUQ=I
      IF(I.EQ.1)GO TO 35
      ILQ=IUQ-1
      QVALUE=(Y(IUQ)-Y(ILQ))*(QUANT-X(ILQ))
      QVALUE=QVALUE/(X(IUQ)-X(ILQ))
      QVALUE=QVALUE+Y(ILQ)
      GO TO 36
 35   CONTINUE
      QVALUE=Y(I)
 36   RETURN
      END
```

```
      SUBROUTINE QHLIN(X,N,H,U,Q,IOPT,NQ,XMIN,XINT,NOFNOF)
C*****************************************************************
C   SUBROUTINE TO COMPUTE Q(U) AT U = H,2H,3H,....,1-H WHERE
C   H IS > OR = .002 BY LINEAR INTERPOLATION FROM ORDER
C   STATISTICS IF DATA IS UNGROUPED (IOPT=0), OR FROM TALLIES
C   IF DATA IS GROUPED (IOPT=1).
C   INPUT :
C            X : VECTOR OF ORDER STATISTICS IF DATA IS UNGROUPED
C                VECTOR OF TALLIES IF DATA IS GROUPED
C            N : DIMENSION OF X
C            XMIN : NATURAL MINIMUM (IF DATA IS UNGROUPED)
C                   LOWER LIMIT OF 1ST GROUP (FOR GROUPED DATA)
C            XINT : GROUP WIDTH (IF DATA IS GROUPED)
C            IOPT : =0 IF DATA IS UNGROUPED
C                   =1 IF DATA IS GROUPED
C            H : FIRST VALUE (AND INCREMENT) FOR COMPUTING Q
C                 USUALLY H=.01,.025, OR .05
C            NOFNOF : MAXIMUM OF (N,1/FINC) PLUS 1
C   OUTPUT :
C            U : VECTOR OF U VALUES (MAX 500)
C            Q : VECTOR CONTAINING Q(U) (MAX 500)
C   SUBROUTINES CALLED : NONE
C******************************************************************
      DIMENSION X(NOFNOF),U(NOFNOF),Q(NOFNOF)
      NQ=1./H
      IF(NQ.GT.500)GO TO 900
      IF((FLOAT(NQ)*H).EQ.1.) NQ=NQ-1
      IF(IOPT.EQ.1)GO TO 50
C   COMPUTE Q FOR UNGROUPED DATA
      X(N+1)=X(N)
      DO 10 I=1,NQ
      J=FLOAT(N)*FLOAT(I)*H+.5
      U(I)=FLOAT(I)*H
      A=U(I)*FLOAT(N)-FLOAT(J)+.5
      Q(I)=A*X(J+1)+(1.-A)*X(J)
   10 CONTINUE
      GO TO 80
   50 CONTINUE
C   COMPUTE Q FOR GROUPED DATA
      XDEN=0.
      DO 60 I=1,N
      XDEN=XDEN+X(I)
   60 CONTINUE
      UL=0.
      XNUM=0.
```

```
      IU=1
      IV=1
      U(1)=H
   65 XNUM=XNUM+X(IU)
      UU=XNUM/XDEN
   70 IF(UL.LE.U(IV).AND.U(IV).LT.UU)GO TO 75
      IU=IU+1
      UL=UU
      GO TO 65
   75 CONTINUE
      A=(U(IV)-UL)/(UU-UL)
      XL=XMIN+XINT*FLOAT(IU-1)
      XU=XL+XINT
      Q(IV)=A*XU+(1.-A)*XL
      IV=IV+1
      IF(IV.GT.NQ)GO TO 80
      U(IV)=FLOAT(IV)*H
      GO TO 70
   80 CONTINUE
      DO 90 I=1,NQ
      I2=NQ-I+2
      I1=NQ-I+1
      Q(I2)=Q(I1)
   90 U(I2)=U(I1)
      Q(1)=XMIN
      U(1)=0.
      GO TO 999
  900 WRITE(6,901)H
  901 FORMAT(////,5X,'INCREMENT OF',F5.4,' IS TOO SMALL TO COMPUTE',
     +' Q',/'INCREMENT MUST BE AT LEAST .002')
  999 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QTOFQ(Q,U,NQP1,XS,FQ)
C*********************************************************
C   SUBROUTINE TO COMPUTE Q(U) AND FQ(U)=1./Q(U) FROM
C   THE EMPIRICAL QUANTILE FN CAP Q(U) AND THE U VALUES.
C   INPUT : Q,U,NQ
C   OUTPUT : XS : SPACINGS=Q(U)
C            FQ
C************************************************************
      DIMENSION Q(NQP1),U(NQP1),XS(NQP1),FQ(NQP1)
      NQ=NQP1-1
      XS(NQ)=(Q(NQ+1)-Q(NQ))/(U(NQ+1)-U(NQ))
      NQM1=NQ-1
      DO 10 I=1,NQM1
      XS(I)=(Q(I+2)-Q(I))/(U(I+2)-U(I))
   10 CONTINUE
      DO 20 I=1,NQ
      IF(XS(I).EQ.0.0)XS(I)=0.0001
      FQ(I)=1./XS(I)
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUICK(N,T)
C***********************************************************************
C   QUICK SORT.THIS ALGORITH IS ALSO REFERRED TO AS A PARTITIONED
C   EXCHANGE SORT.EXPECTED RUNTIME IS PROPORTIONAL TO N*LOG2(N)
C   ALTHOUGH THE WORST CASE IS PROPORTIONAL TO N**2.
C   REFERENCE: DONALD E. KNUTH- THE ART OF COMPUTER PROGRAMMING VOL 3.
C   INPUT :
C          X,N : VECTOR TO BE SORTED OF LENGTH N
C   OUTPUT :
C          X : SORTED VECTOR
C   SUBROUTINES CALLED : NONE
C***********************************************************************
      REAL T(N),Y
      INTEGER IP,LV(16),IV(16),LP,IUP
      LV(1)=1
      IV(1)=N
      IP=1
   10 IF(IP.LT.1) GO TO 75
   15 IF((IV(IP)-LV(IP)).LT.1) GO TO 20
      GO TO 25
   20 IP=IP-1
      GO TO 10
   25 LP=LV(IP)-1
      IUP=IV(IP)
      Y=T(IUP)
   30 IF((IUP-LP).LT.2) GO TO 45
      LP=LP+1
      IF(T(LP).LE.Y) GO TO 30
      T(IUP)=T(LP)
   35 IF((IUP-LP).LT.2) GO TO 40
      IUP=IUP-1
      IF(T(IUP).GE.Y) GO TO 35
      T(LP)=T(IUP)
      GO TO 30
   40 IUP=IUP-1
   45 T(IUP)=Y
      IF((IUP-LV(IP)).LT.(IV(IP)-IUP)) GO TO 55
      GO TO 60
   55 LV(IP+1)=LV(IP)
      IV(IP+1)=IUP-1
      LV(IP)=IUP+1
      GO TO 70
   60 LV(IP+1)=IUP+1
      IV(IP+1)=IV(IP)
      IV(IP)=IUP-1
```

```
70 IP=IP+1
   GO TO 15
75 RETURN
   END
```

```
      SUBROUTINE USMNMX (X,N,INC,XMIN,XMAX)
C   LOCATE MIN AND MAX OF A VECTOR
      DIMENSION X(N)
      XMIN = X(1)
      XMAX = X(1)
      DO 10 I = 1,N,INC
      IF(X(I) .GE. XMIN) GO TO 5
      XMIN = X(I)
      GO TO 10
    5 IF(X(I) .GT. XMAX) XMAX = X(I)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE WSPACE(WXS,CWXS,NQP1,FQ,IDQH,U,SIG0)
C*******************************************************
C   SUBROUTINE TO COMPUTE D(U), CUMULATIVE D'S, AND SIGMAO
C   FOR THE MODEL Q(U)=MU+SIGMA*Q0(U)
C   INPUT :
C           FQ,NQ : VECTOR OF LENGTH NQ CONTAINING FQ(U)
C           FOQ0 : HYPOTHESIZED DENSITY QUANTILE FN--EXTERNAL
C                  FUNCTION
C           U : VECTOR OF LENGTH NQ CONTAINING U VALUES
C   OUTPUT :
C           WXS : VECTOR OF LENGTH NQ CONTAINING D(U)
C           CWXS : VECTOR OF LENGTH NQ CONTAINING THE
C                  CUMULATIVE D'S
C           SIG0 : COMPUTED VALUE OF SIGMAO = CWXS(NQ)
C   SUBROUTINES CALLED : NONE
C*********************************************************
      DIMENSION FQ(NQP1),U(NQP1),WXS(NQP1),CWXS(NQP1)
      NQ=NQP1-1
      CWXS(1)=0.
      DO 10 I=2,NQP1
      FAC=FQFNC(U(I),IDQH)
      WXS(I-1)=FAC/FQ(I-1)
      CWXS(I)=CWXS(I-1)+WXS(I-1)
   10 CONTINUE
      SIG00=CWXS(NQP1)
      SIG0=SIG00/FLOAT(NQ)
      DO 20 I=1,NQ
      WXS(I)=WXS(I)/SIG0
      CWXS(I)=CWXS(I)/SIG00
   20 CONTINUE
      CWXS(NQP1)=1.
      RETURN
      END
```

```
      SUBROUTINE XBARS2(X.N,XBAR,S2)
      DIMENSION X(N)
      ON=N
      ON1=ON-1.
      C1=0.
      C2=0.
      DO 1 I=1,N
      C1=C1+X(I)
    1 C2=C2+X(I)*X(I)
      XBAR=C1/ON
      S2=(C2-((C1*C1)/ON))/ON1
      RETURN
      END
```

```
      SUBROUTINE MDNRIS (P,Y,IER)
C   QUANTILE FUNCTION FOR N(0,1)
      DIMENSION A(65),C(17),D(25),E(23)
      EQUIVALENCE (A(24),C(1)),(A(41),D(1)),  (A(1),E(1) )
      DATA E/.992885376618940,.120467516143104,
     +     .016078199342100,.002686704437162,
     +     .000499634730235,.000098898218599,
     +     .000020391812764,.000004327271618,
     +     .000000938081413,.000000206734721,
     +     .000000046159699,.000000010416680,
     +     .000000002371501,.000000000543928,
     +     .000000000125549,.000000000029138,
     +     .000000000006795,.000000000001591,
     +     .000000000000374,.000000000000088,
     +     .000000000000021,.000000000000005,
     +     .000000000000001/
      DATA C/.912158803417554,-.016266281867664,
     +     .000433556472949,.000214438570074,
     +     .000002625751076,-.000003021091050,
     +     -.000000012406062,.000000062406609,
     +     -.000000000540125,-.000000001423208,
     +     .000000000034384,.000000000033585,
     +     -.000000000001458,-.000000000000810,
     +     .000000000000053,.000000000000020,
     +     -.000000000000002/
      DATA D/.956679709020493,-.023107004309065,
     +     -.004374236097508,-.000576503422651,
     +     -.000010961022307,.000025108547025,
     +     .000010562336068,.000002754412330,
     +     .000000432484498,-.000000020530337,
     +     -.000000043891537,-.000000017684010,
     +     -.000000003991289,-.000000000186932,
     +     .000000000272923,.000000000132817,
     +     .000000000031834,.000000000001670,
     +     -.000000000002036,-.000000000000965,
     +     -.000000000000220,-.000000000000010,
     +     .000000000000013,.000000000000006,
     +     .000000000000001/
      DATA H1,H2,H3,H4,R2/-1.5488130423733,
     +     2.5654901231478,-.55945763132983,
     +     2.2879157162634,-1.4142135623731/
      XINF=2**31-1
      X = 1.-P-P
      IER = 0
      SIGMA = SIGN(1.,X)
```

```
      IF( .NOT.(X.GT.-1..AND.X.LT.1.)) GO TO 45
      Z = ABS(X)
      IF(Z .GT. .8) GO TO 25
      W = Z*Z/.32-1.
      N = 22
      IPP = 1
      L = 1
   10 LB2 = 1
      X3 = 1.
      X4 = W
      X6 = A(IPP)
   15 X6 = X6 + A(IPP+LB2) * X4
      X5 = X4 * W * 2.-X3
      X3 = X4
      X4 = X5
      LB2 = LB2 + 1
      IF(LB2 .LE. N) GO TO 15
      GO TO (20,35),L
   20 Y = Z * X6 * SIGMA
      GO TO 40
   25 B = SQRT(-ALOG(1.-Z*Z))
      IF(X .GT. .9975) GO TO 30
      W = H1*B+H2
      IPP = 24
      L = 2
      N = 16
      GO TO 10
   30 W = H3 * B + H4
      IPP = 41
      N = 24
      L = 2
      GO TO 10
   35 Y = B * X6 * SIGMA
   40 Y = R2*Y
      RETURN
   45 Y = SIGMA*XINF
      IER = 129
      WRITE(6,101)
  101 FORMAT(/////////1X.35H*****ERROR CONDITION IN MDNRIS*****  ///)
      RETURN
      END
```