A Description of Program BELEM.BAS

by
James R Warren BSc MSc PhD PGCE

The program BELEM.BAS listed in Appendix One is designed to integrate basic statistical information with point features charted upon a map.

Each point feature is characterised by some scalar metric, a VALUE MAGNITUDE, which specifies its absolute importance in one or more groups (CLUSTERS). The use of the word "cluster" does not necessarily imply propinquity. In my application the magnitude is a production figure for a particular metal mine whose Main Engine Shaft or surface point feature of equivalent status is mapped on an Ordnance Survey projection. BELEM.BAS is, however, entirely general.

The program generates three kinds of statistical summary for each CLUSTER:-

(a) The Population Mass Centroid and Dynamic Radius

(b) A Nest of Momental Statistics upon Magnitude

(c) The Plotting Vertices of a quadrilateral cartographic device called a BELEMNOID which indicates the cluster Mass Centroid, Arithmetic Mean and Standard Deviation, and points the cluster Migration Direction

The key features of the algorithm are developed from the mathematical principles contained in my paper "Characterising the Belemnoid: A Graphical Device for Mapping Descriptive Statistics". Accordingly, I shall not dwell upon those fundamentals here, focusing instead upon extra new features and upon technical innovations.

The Computation of Descriptive Statistics

Momental descriptive statistics are computed for each cluster using the value magnitudes of each member. Subroutine MOMENTAL supplies the Arithmetic Mean, Population Standard Deviation, Sample Standard Deviation, Momental Skewness and Momental Kurtosis for a simple data distribution.

Arithmetic Mean and Population Standard Deviation are later used for sizing the belemnoid pointer (Subroutine DIATESSARON), whilst Sample Standard Deviation is employed in computing distributional confidence limits (Subroutine CONFIDENCE).

Momental Skewness and Kurtosis[1] should respectively be zero and three for a Gaussian Distribution and thus could form the basis of a test of Normality. Normality is indeed assumed for the application of Student's t to confidence testing though Student's t is robust under non-normality in many situations.

Readers of the source listing will discover a Subroutine GF designed to establish the five statistics for a grouped frequency distribution[2]. Both GF and MOMENTAL have extensively been tested both within and outwith BELEM.BAS and have agreed with each other to twelve-figure accuracy and with Casio fx-570C assisted hand-calculations to ten figures. GF will function upon a simple data distribution such as a BELEM cluster provided that each element of the frequency array XF is set to unity and the point magnitudes fed to array XM. Whilst GF is of high value to sedimentologists and other appropriate researchers it is an unwarrantably expensive procedure in our context and I have disengaged it in favor of MOMENTAL.

The Computation of Student's t

Cluster populations are frequently small. In any event my Victorian lead mines can been formed into groups of perhaps three or ten without even a theoretical possibility of supplementation.

If a cluster population is Normal it may be tested using either the Gaussian or the Student's t distribution function but if the population is less than thirty the latter is the only realistic recourse.

Confidence then depends upon knowledge of a quantity called "Student's" or Gosset's t which is function of the number of items in the cluster and the confidence probability deemed satisfactory.

There are several approaches to computing t. Tabular interpolation is crude, derivative and potentially inaccurate. At the opposite extreme, direct numerical integrations firstly of the required Gamma Functions and then of the associated confidence probability can be engineered. These, however, sound horribly expensive.

Hill's Routine[3]

As a happy medium I have chosen to apply Hill's Process to obtain this number. Firstly, if there are two magnitudes in the cluster then an exact t is readily available via Cauchy's Theorem. There is also an exact, closed-form solution available for three data. The fun begins when there is something between three and thirty data in the batch and a venerable natural law well known in this country guarantees that to be the case. In such a circumstance Hill deploys one or other of two series expansions for t depending upon a secondary criterion. He suggests that either will furnish five-figure accuracy. In one of the two options, however, the expansion elaborates about the appropriate normal deviate, an algorithm for which Hill does not supply. My tests quickly affirmed that the offensive alternative covered almost all of the field of interest as per that ancient law, which delicacy forbids me to name.

The Action of Segment REFINERY

Clearly the accuracy of t can only be as good as the value of the normal deviate upon which it depends. Using Equation 26.2.23 in Abramowitz and Stegun[4] in order to approximate the deviate to 0.00045 I discovered that the consequential t did not attain four-figure accuracy. Although even this level of accuracy is somewhat academic in the context I was not happy that this Achilles' Heel should vitiate the algorithm to the extent of making it inferior to table-based solutions. My slightly expensive solution to this problem involved a tedious, but ultimately deadly-accurate repetition.

I might analogise this process to the technique of kedging resorted to by the old time sailing masters if they fell becalmed in shallows. They would have the anchor rowed a few hundred meters ahead in a cutter and then dropped to the sea bed. A party aboard the ship would then haul on the anchor capstan till the craft drew up to and lifted dry the anchor. Repetition then secured another tedious increment of uncial progress. I think the mining students amongst you will prefer the illustration of the Victorian roller crusher whose orbiting raff wheel returned the coarse comminute to the mill for another pass between the cylinders until it was small enough to fall through the sieve.

In any event, the Hastings[5] polynomial approximation from Abramowitz and Stegun was used as a starting point to furnish an estimate of z which I then flanked with two further values at z-ε and z+ε where ε was initially the 0.00045 guaranteed by Hastings. I then employed a 64-interval Romberg integration of the Gaussian Distribution Function to secure probability values attaching to the three z's in addition to the known target probability. As a second stage in the iteration a Lagrangian interpolation utilised the three computed probabilities with known z values to estimate a z for the target probability. This equivalates fitting a quadratic curve to the points and reading the fourth point at the required abscissa. On the next pass of the iteration ε was set to the absolute difference between the current and previous deviate estimates.

The tolerance in ε was set for eight figure accuracy. On the first iteration true z was approximated to five-figure accuracy by comparison with fiducial values in Abramowitz and Stegun.

The second pass, necessary and sufficient in all cases, yielded twelve-figure accuracy.

When this highly-superior estimate of the normal deviate was fed to Hill's Routine (Function T) the latter yielded a value which agreed with Federighi's[6] widely-respected published table of t to its limits of accuracy which is at least four figures.

The Automatic and Manual Scaling Alternatives

The chief design purpose of BELEM.BAS is to compute definitive co-ordinates of dart-shaped graphical deltoids or belemnoids.

The size of these devices on the sheet of paper can either be plotted automatically or manually. The automatic function can give good balance both as regards the relative size of the belemnoids between clusters and the spread of the figure over any associated spatial field. It has several compensating drawbacks and in particular there can be no guarantee that the resulting shape will not transgress the map boundaries or obscure other important detail.

The automatic facility hinges upon the prior calculation of the mean dynamic radius for all the available data clusters. (By employing MOMENTAL within STANDARD). This mean is then embodied as the radius of a circle circumscribing a pentagram. Such a pentagram contains three interpenetrative golden triangles. The length of the axis of symmetry of any of these triangles is now identified with the mean of the cluster magnitude arithmetic means, and all belemnoids including the scale standard plotted commensurately.

The manual facility requires the user to specify the axial length of the scale standard belemnoid (in map or paper units) and the associated reference quantity (in value magnitude units) and the cluster belemnoids are then scaled by reference to these parameters. This permits the researcher total control of plotting.

Though prior classification of the cluster members is likely to reduce variation in momental statistics, at least as regards standard deviation, researchers who find that their belemnoids are nevertheless too disparate may consider logarithmic data transformations or the like.

The Required Format of the Input Data

BELEM.BAS is a pseudo-batch routine, meaning that once the user has named the file which is the source of the data the algorithm drives to conclusion without further human intervention.

Researchers should note that the logical pathname is fixed within the source with the line:-

SP = "C:\QBASIC\QBFILES\"

This should be varied if a different file directory is to be employed.

The batch-like nature of the procedure requires fastidious attention to the arrangement and quality of the pre-filed input if crash or worse is to be avoided.

The source is currently configured to utilise up to eleven clusters each containing up to a hundred data positions. Twenty-one statistics (including the Cluster Name) are returned for each cluster.

Data names initialed with S are strings and with L are long integers (absent in BELEM.BAS). Otherwise I through N are short integers. All other names are double-precision reals.

Researchers who compose data using spreadsheets or similar tools should regard names starting with S as denoting text and anything else as signifying plain numbers. BELEM.BAS is indifferent as to whether input matrices' empty elements are filled with nullities or zeros, but most spreadsheet-based CSV-generators will probably work better from zero-padded blocks.

The format determined by Subroutine DATAIN is:-

Group A

SW
CB,CC
EM,AO
D,AL
K,NM

Group B

K lots of
SK,NC
followed by NC lots of
IP(I,J)

Group C

NM lots of SN
NM lots of AE
NM lots of AN

Group D

K lots of
NM lots of PP(I,J)

Group A is a predicate of general controls. SW is the name of the magnitude units (e.g. metric tonnes, population of settlement). CB is the magnitude represented by the standard belemnoids symmetry axis. If this is zero or negative then automatic scaling ensues but if it is positive the quoted value is adopted as the actual scale parameter. CC is the standard belemnoid symmetry axis length in map units and a dummy value should be included even for automatically scaled jobs. EM and AO are respectively the highest easting and lowest northing to define the nidus for the standard belemnoid towards the bottom right corner of the map. D and AL are Confidence Parameters. D is the tolerable error in estimating arithmetic mean magnitude expressed in magnitude units (e.g. kilograms, tonnes, head of sheep). AL is the confidence interval expressed as a finite probability between zero and unity (e.g. 0.95 for 95% confidence). K and NM are Data Extent Parameters: K is the number of clusters defined for this job and NM the maximum number of data items in each cluster (including null data).

Group B is a block of Data Item Picking Controls that permits a specified selection of cluster members for the job in hand. There are K lines of data starting with the Cluster Name SK and then the Number of Cluster Members NC. (NC's name might vary in the published source). The rest of the line contains the NC subscripts of the Data Elements themselves. Clearly this scheme can be used to skip null or other undesired point magnitudes.

Group C is a block of Point Location Specifiers. It comprises a row of NM Map Feature Names SN (e.g. Lead Mine Names, Village Names, Station Names), matched by a row of NM Eastings beneath, and NM Northings for a third row.

Finally, Group D is a grand rectangular matrix of null and finite Magnitudes, as many columns as there are available data objects (NM) and as many rows as clusters (K). These magnitudes might be metric tonnes of lead metal production, head of sheep for a given station and year or whatever. Each matrix column associates with a co-ordinate Group C Map Feature.

All this data is carefully composed in a comma separated variable ASCII file to be entitled *.CSV where * is any lawful MicroSoft filename.

The Format of the Output Data

BELEM.BAS dumps the output comma separated ASCII table as *.BEL down the pathname SP.

The sole output datum of the first row echos the quoted filename. A blank record then precedes a row containing the Cluster Names since cluster statistics are to be developed columnwise. There is then a second blank row.

The central tabular body has Output Statistics Names as the first column. These statistics names define rows of result data and are the twenty statistics labels "ITEMS IN CLUSTER" through "BELEMNOID RIGHT FOOT Y" shown in Appendix Two on the BELEM.BAS (SuperCalc 5) Validation Document BELVALID.CAL. The second column is reserved for the vertex co-ordinates of the standard belemnoid. The subsequent K columns list the statistics for each cluster.

The third distinct output element is a statement at the start of a final record specifying the number of magnitude units represented by the standard belemnoid symmetry axis.

This is the mean of magnitude means for automatically scaled jobs or the given CB for manually scaled ones.

Examples for The Mines of Opprobrium

Appendix Three includes intercalated specimens of input and output data for the Mines of Opprobrium whose production figures were tabulated and mapped in the mother paper "Characterising the Belemnoid: A Graphical Device for Mapping Descriptive Statistics".

The input sheets are straight listings of the relevant ASCII *.CSV files. The output sheets are the *.BEL file contents dressed for presentation as SuperCalc 5 worksheets.

No results have been altered. *.BEL outputs were compared with BELVALID.CAL whose centroid and vertex data were computed by independent spreadsheet methods. For BELVALID.CAL centroid data were re-computed by hand and momental statistics computed by hand.

The First and Second Data Sets respectively present Opprobrium figures with automatic belemnoid scaling. The First Set utilises null elements in the Production Matrix: The Second Set zero infill. Readers may confirm the identity of the results.

The Third Data Set is for manual scaling with CB=800 (Metric Tonnes) and CC=20 (millimeters on paper). Readers are invited to check that only the belemnoid vertices are changed. Manual plotting and superposition confirmed the required contraction and the concordance of scaling.

Twelve-figure accuracy would appear to be general (only seven figures are shown on the worksheet printouts) except for the confidence limits where five figures seems realistic except for Cluster 1960 where two degrees of freedom allow exactitude. I ascribe the obvious discrepancies to inevitable imprecisions in Federighi's excellent four-figure t-table. Normative sample size is rounded to the next largest integer.

References

1 "Statistics in Geography"
Second Edition 1985
David Ebdon
Blackwell of Oxford
ISBN 0-631-13688-6

2 "Quantitative Geography"
Cole and King 1968
John Wiley and Sons Limited
SBN 471-16475-5

3 "Algorithm 396 Student's t-Quantiles [S14]"
G W Hill
Communications of the ACM
V13 N10 pp 619-620 October 1970

4 "Handbook of Mathematical Functions"

Milton Abramowitz and Irene A Stegun
Dover of New York 1965
ISBN 0-486-61272-4

5 "Approximations for Digital Computers"
C Hastings Jnr 1955
Princeton University Press of Princeton

6 "Extended Tables of the Percentage Points of Student's t Distribution"
Enrico T Federighi
Journal of the American Statistical Association
V54 pp 683-688 1959

Appendix One

The Program BELEM.BAS

```
'       PROGRAM BELEM.BAS
'       A PROGRAM TO COMPUTE THE DESCRIPTIVE STATISTICS OF THE MAGNITUDES
'       IN CONTEMPORANEOUS CLUSTERED OR CLASSIFIED ENTITIES LOCATED IN A
'       PLANAR CARTESIAN GRID SYSTEM.
'       THE PROGRAM ALSO COMPUTES THE SPATIAL MASS CENTROIDS AND DYNAMIC
RADII
'       OF SUCCESSIVE CLUSTERS AND ERECTS PLOTTABLE BELEMNOIDS WHICH
SUMMARISE
'       MAGNITUDE MIGRATION DIRECTION AND MAGNITUDE MEAN AND STANDARD
DEVIATION
'       FOR EACH CLUSTER.
'       CONFIDENCE LIMITS ARE QUOTED FOR THE MEAN OF THE ENTITIES IN EACH
CLUSTER.
'       THE PROGRAM IS CONFIGURED FOR BATCH WORKING AFTER INTERACTIVE
FILENAME
'       PROMPTING.
'       CONSULT THE PAPER "A DESCRIPTION OF PROGRAM BELEM.BAS"
'       FOR DATA FORMATTING DETAILS.
'       THESE FILES BEAR THE EXTENSION SHOWN:-
'
'             THE DATA       INPUT  FILE      .CSV
'             THE STATISTICS OUTPUT FILE      .BEL
'
'       WRITTEN BY:-
'
'          JAMES R WARREN BSc MSc PhD PGCE
'          "SOUTHGATE"
'          31 VICTORIA AVENUE
'          BLOXWICH
'          WS3 3HS
'          UNITED KINGDOM
'
'          22 JULY 1996
'
'       THIS PROGRAM IS WRITTEN IN MICROSOFT QBASIC
'
' VARIABLE TYPE DEFAULTS
       DEFDBL A-H, O-R, T-Z
       DEFSTR S
       DEFINT I-K, M-N
       DEFLNG L
' SEGMENT DECLARATIONS
       DECLARE SUB CENTROID (J, N, P(), X(), Y(), CX, CY, DS)
       DECLARE SUB CONFIDENCE (J, N, D, AL, AM, VS, NS, DL, DU)
       DECLARE SUB DATAIN (SW, CB, CC, EM, AO, D, AL, K, NM, SK(), IP(),
SN(), AE(), AN(), PP(), V())
       DECLARE SUB DATAOUT (K, CB, AX, SW, SJ(), SK(), V())
       DECLARE SUB DIATESSARON (J, CK, PI, AM, VP, TX, CX, CY, V())
       DECLARE SUB FINALE ()
       DECLARE FUNCTION GAUSSIAN (Z)
       DECLARE SUB GETTER ()
       DECLARE SUB GF (NDC, N, XM(), XF(), AM, VP, VS, VQ, VK)
       DECLARE SUB LAGRANGIAN (N, XL(), FL(), XT, FT)
       DECLARE SUB MOMENTAL (N, P(), AM, VP, VS, VQ, VR)
       DECLARE SUB NOTE (S)
       DECLARE SUB PICK (KT, I, L, R, S, IY, IX, MK, SF)
       DECLARE SUB PLACE (KT, I, L, R, S, IY, IX, IU, MK, JF, SF, IB)
       DECLARE SUB REFINERY (PI, TL, UV, PV)
       DECLARE SUB RESULTHETA (K, PI, TA())
       DECLARE SUB ROMBERG (NT, CF, FL, FU, RI)
       DECLARE SUB ROTATE (TH, XO, YO, XI, YI, XR, YR)
       DECLARE SUB STANDARD (CB, CC, PI, PH, K, CK, DR(), AX, EM, AO, V())
       DECLARE SUB THETA (J, PI, TA(), V())
       DECLARE FUNCTION T (JD, AL)
       DECLARE FUNCTION TND (AL)
```

```
' COMMON VARIABLES
       COMMON SHARED IA, SA
       COMMON SHARED PI, HP
       COMMON SHARED SC, SM, SCR
       COMMON SHARED IU, IV, SP, SF, SXV, SXW
       COMMON SHARED SAP, SB, SE, SI, SL, SLF, SMP, SN, SO, SRF, ST, SU,
SV, SX, SY
' STATIC ARRAY DEFINITIONS
'      ( none )
' DYNAMIC ARRAY DEFINITIONS
'      CURRENT NUMBER OF CLUSTERS                           NCL=11
'      CURRENT NUMBER OF DATA PER CLUSTER                   NDC=100
'      CURRENT NUMBER OF OUTPUT STATISTICS PER CLUSTER      NOS=21
       NCL = 11: NDC = 100: NOS = 21
       REDIM SK(NCL), DR(NCL)
       REDIM TA(NCL)
       REDIM SN(NDC), AE(NDC), AN(NDC)
       REDIM P(NDC), X(NDC), Y(NDC)
       REDIM V(NDC, NOS)
       REDIM IP(NCL, NDC), PP(NCL, NDC)
       REDIM SJ(NOS)
       REDIM NR(NCL), CL(NCL), CU(NCL)
' DEVICE ATTRIBUTIONS
       SCREEN 12: WINDOW (1, 1)-(640, 480)
' LOGICAL UNIT, EXTENSION AND PATHNAME SETTINGS
       IU = 1: IV = 2
       SXV = ".CSV": SXW = ".BEL"
       SP = "C:\QBASIC\QBFILES\"
' FORMAT DEFINITIONS
'      ( none )
' NUMERICAL CONSTANT DEFINITIONS
       PH = 1.618033988749895#
       PI = 3.141592653589793#
       HP = PI / 2#
' STRING CONSTANT DEFINITIONS
       SC = ":": SM = ",": SCR = CHR$(13) + CHR$(10)
' TEXT VARIABLE DEFINITIONS
       SAP = "APEX       ": SB = "BELEMNOID ": SE = "CENTROID ": SI =
"ITEMS ": SL = "CLUSTER "
       SLF = "LEFT FOOT  ": SMP = "SAMPLE ": SN = "NAME ": SO = "LOWER ":
SRF = "RIGHT FOOT "
       ST = "CONFIDENCE LIMIT": SU = "UPPER ": SV = "VALUES ": SX = "X": SY
= "Y"
'
' ** THE ALGORITHM **
'
       SZ = "ENTER THE FILENAME ( WITHOUT EXTENSION ):"
       PLACE 4, 0, 0, 0, SZ, 15, 16, 1, 2, 1, SF, 0
       PICK 4, 0, 0, 0, SF, 15, 59, 3, SA8
       CLS
       PLACE 4, 0, 0, 0, "PLEASE WAIT", 15, 36, 1, 14, 1, SF, 0
       DATAIN SW, CB, CC, EM, AO, D, AL, K, NM, SK(), IP(), SN(), AE(),
AN(), PP(), V()
       FOR J = 1 TO K
          FOR I = 1 TO V(J, 2)
             P(I) = PP(J, IP(J, I)): X(I) = AE(IP(J, I)): Y(I) = AN(IP(J,
I))
          NEXT I
          CENTROID J, FIX(V(J, 2)), P(), X(), Y(), V(J, 3), V(J, 4), V(J,
5)
          DR(J) = V(J, 5)
       NEXT J
       FOR J = 1 TO K - 1
          THETA J, PI, TA(), V()
       NEXT J
```

```
       RESULTHETA K, PI, TA()
       AX = 0#
       FOR J = 1 TO K
          FOR I = 1 TO V(J, 2): P(I) = PP(J, IP(J, I)): NEXT I
          MOMENTAL FIX(V(J, 2)), P(), V(J, 6), V(J, 7), V(J, 8), V(J, 9),
V(J, 10)
          CONFIDENCE J, FIX(V(J, 2)), D, AL, V(J, 6), V(J, 8), NS, V(J,
12), V(J, 13)
          V(J, 11) = NS: AX = AX + V(J, 6)
       NEXT J
       AX = AX / K
       STANDARD CB, CC, PI, PH, K, CK, DR(), AX, EM, AO, V()
       FOR J = 1 TO K
          DIATESSARON J, CK, PI, V(J, 6), V(J, 7), TA(J), V(J, 3), V(J, 4),
V()
       NEXT J
       DATAOUT K, CB, AX, SW, SJ(), SK(), V()
       FINALE
       END

       SUB CENTROID (J, N, P(), X(), Y(), CX, CY, DS)
' A SUBROUTINE TO COMPUTE THE MASS CENTROID AND THE DYNAMIC RADIUS OF
' A SET OF VALUES IN A CARTESIAN CO-ORDINATE SYSTEM.
' THE CENTROID FORMULAE ARE TAKEN FROM PAGE 116 OF
' "GEOGRAPHICAL DATA:SOURCES, PRESENTATION AND ANALYSIS" BY HUGH MATTHEWS
' AND IAN FOSTER ( ISBN 0-19-913328-X ).
' THE FORMULA FOR DYNAMIC RADIUS IS TAKEN FROM PAGE 313 OF "LOCATIONAL
METHODS"
' BY HAGGETT, CLIFF AND FREY ( ISBN 0-7131-5956-1 )
'      ARGUMENTS:
'         J      THE CLUSTER NUMBER
'         N      THE NUMBER OF CLUSTER MAGNITUDES
'         P()    THE ARRAY OF VALUE MAGNITUDES
'         X()    THE ARRAY OF MAGNITUDE X CO-ORDINATES
'         Y()    THE ARRAY OF MAGNITUDE Y CO-ORDINATES
'         CX     THE ARRAY OF CENTROID  X CO-ORDINATES
'         CY     THE ARRAY OF CENTROID  Y CO-ORDINATES
'         DS     THE ARRAY OF DYNAMIC RADII
'
       PS = 0#: XS = 0#: YS = 0#: DS = 0#
       FOR I = 1 TO N
          PS = PS + P(I): XS = XS + P(I) * X(I): YS = YS + P(I) * Y(I)
       NEXT I
       CX = XS / PS: CY = YS / PS
       FOR I = 1 TO N
          DS = DS + P(I) * ((X(I) - CX) ^ 2 + (Y(I) - CY) ^ 2)
       NEXT I
       DS = SQR(DS / PS)
       END SUB

       SUB CONFIDENCE (J, N, D, AL, AM, VS, NS, DL, DU)
' A SUBROUTINE TO COMPUTE FOR A DATA CLUSTER THE NORMATIVE SAMPLE SIZE
NR(J) AND
' THE UPPER AND LOWER CONFIDENCE LIMITS CL(J) AND CU(J)
'      ARGUMENTS:
'         J      THE CLUSTER NUMBER
'         N      THE NUMBER OF CLUSTER MAGNITUDES
'         D      THE DESIRED ACCURACY ( UNITS )
'         AL     THE CONFIDENCE INTERVAL ( PROBABILITY )
'         AM     THE ARITHMETIC MEAN           OF THE CLUSTER MAGNITUDES
'         VS     THE SAMPLE STANDARD DEVIATION OF THE CLUSTER MAGNITUDES
'         NS     THE NORMATIVE SAMPLE SIZE
'         DL     THE LOWER CONFIDENCE LIMIT
'         DU     THE UPPER CONFIDENCE LIMIT
'
```

```
       JD = N - 1: TT = VS * T(JD, (1 - AL) / 2)
       NS = INT((TT / D) ^ 2) + 1
       A = TT / SQR(N): DL = AM - A: DU = AM + A
       END SUB

       SUB DATAIN (SW, CB, CC, EM, AO, D, AL, K, NM, SK(), IP(), SN(),
AE(), AN(), PP(), V())
' A SUBROUTINE TO LOAD THE SCALER AND LOCATIONAL DATA FOR A MAGNITUDE
CLUSTER
' MIGRATION ANALYSIS
'      ARGUMENTS:
'         SW     THE MAGNITUDE UNIT NAME
'         CB     THE REPRESENTED MAGNITUDE OF THE STANDARD BELEMNOID AXIS (
MAGNITUDE UNITS )
'         CC     THE STANDARD BELEMNOID AXIS LENGTH ( MAP UNITS )
'         EM     THE HIGHEST EASTING
'         AO     THE LOWEST NORTHING
'         D      THE DESIRED ACCURACY ( UNITS )
'         AL     THE CONFIDENCE INTERVAL ( PROBABILITY )
'         K      THE NUMBER OF CLUSTERS
'         NM     THE NUMBER OF DATA OBJECTS
'         SK()   THE ARRAY OF CLUSTER NAMES
'         IP()   THE ARRAY OF PICKING LABELS
'         SN()   THE ARRAY OF DATA OBJECT NAMES
'         AE()   THE ARRAY OF DATA OBJECT EASTINGS
'         AN()   THE ARRAY OF DATA OBJECT NORTHINGS
'         PP()   THE ARRAY OF MAGNITUDES
'         V()    THE ARRAY OF OUTPUT STATISTICS
' ( IU, SP, SF AND SXV ARE COMMON SHARED )
'
       OPEN "I", IU, SP + SF + SXV
       INPUT #IU, SW
       INPUT #IU, CB, CC
       INPUT #IU, EM, AO
       INPUT #IU, D, AL
       INPUT #IU, K, NM
       FOR I = 1 TO K
          INPUT #IU, SK(I), V(I, 2)
          FOR J = 1 TO V(I, 2): INPUT #IU, IP(I, J): NEXT J
       NEXT I
       FOR I = 1 TO NM: INPUT #IU, SN(I): NEXT I
       FOR I = 1 TO NM: INPUT #IU, AE(I): NEXT I
       FOR I = 1 TO NM: INPUT #IU, AN(I): NEXT I
       FOR I = 1 TO K: FOR J = 1 TO NM: INPUT #IU, PP(I, J): NEXT J: NEXT I
       CLOSE IU
       END SUB

       SUB DATAOUT (K, CB, AX, SW, SJ(), SK(), V())
' A SUBROUTINE TO SAVE THE SCALER AND LOCATIONAL SUMMARY STATISTICS FOR
' A MAGNITUDE CLUSTER MIGRATION ANALYSIS
'      ARGUMENTS:
'         K      THE NUMBER OF CLUSTERS
'         CB     THE REPRESENTED MAGNITUDE OF THE STANDARD BELEMNOID AXIS (
MAGNITUDE UNITS )
'         AX     THE MEAN OF THE MAGNITUDE MEANS
'         SW     THE MAGNITUDE UNIT NAME
'         SJ()   THE ARRAY OF DEFINITIONAL CAPTIONS
'         SK()   THE ARRAY OF CLUSTER NAMES
'         V()    THE ARRAY OF OUTPUT STATISTICS
' ( IV, SP, SF, SM, SCR, SXW AND ALL CLICHES ARE COMMON SHARED )
'
' CAPTION DEFINITION PHASE
       S2 = "  "
       SJ(1) = SL + SN: SJ(2) = SI + "IN " + SL: SJ(3) = SL + SE +
"EASTING": SJ(4) = SL + SE + "NORTHING"
```

```
       SJ(5) = SL + "DYNAMIC RADIUS": SJ(6) = SV + "ARITHMETIC MEAN": SJ(7)
= SV + "POPULATION SD"
       SJ(8) = SV + SMP + "SD": SJ(9) = SV + "SKEWNESS": SJ(10) = SV +
"KURTOSIS": SJ(11) = "NORMATIVE " + SMP + "SIZE"
       SJ(12) = SO + ST: SJ(13) = SU + ST: SJ(14) = SB + SE + S2 + SX:
SJ(15) = SB + SE + S2 + SY: SJ(16) = SB + SLF + SX
       SJ(17) = SB + SLF + SY: SJ(18) = SB + SAP + SX: SJ(19) = SB + SAP +
SY: SJ(20) = SB + SRF + SX: SJ(21) = SB + SRF + SY
' DATA RECORDING PHASE
       OPEN "O", IV, SP + SF + SXW
       PRINT #IV, SF
       PRINT #IV,
       PRINT #IV, SJ(1) + SM + "STANDARD";
       FOR J = 1 TO K: PRINT #IV, SM + SK(J); : NEXT J
       PRINT #IV, SCR
       FOR I = 2 TO 21
          PRINT #IV, SJ(I);
          FOR J = 0 TO K: PRINT #IV, SM; V(J, I); : NEXT J
          PRINT #IV,
       NEXT I
       PRINT #IV, "Standard Belemnoid Mu = ";
       IF CB <= 0# THEN
             PRINT #IV, AX;
          ELSE
             PRINT #IV, CB;
       END IF
       PRINT #IV, SW
       CLOSE IV
       END SUB

       SUB DIATESSARON (J, CK, PI, AM, VP, TX, CX, CY, V())
' A SUBROUTINE TO COMPUTE THE VERTEX CO-ORDINATES OF A CLUSTER BELEMNOID
'      ARGUMENTS:
'         J      THE CLUSTER NUMBER
'         CK     THE SCALING CONSTANT
'         PI     THE LUDOLPHINE CONSTANT
'         AM     THE ARITHMETIC MEAN OF THE CLUSTER MAGNITUDES
'         VP     THE POPULATION STANDARD DEVIATION OF THE CLUSTER
MAGNITUDES
'         TX     THE CLUSTER DIRECTIONAL ANGLE
'         CX     THE CLUSTER CENTROID X CO-ORDINATE
'         CY     THE CLUSTER CENTROID Y CO-ORDINATE
'         V()    THE ARRAY OF OUTPUT STATISTICS
'
       AU = AM / CK: AS2 = VP / (2 * CK): H = AS2 * TAN(PI / 5)
       V(J, 14) = CX: V(J, 15) = CY
       V(J, 16) = CX - AS2: V(J, 17) = CY - H
       V(J, 18) = CX: V(J, 19) = CY + AU
       V(J, 20) = CX + AS2: V(J, 21) = CY - H
       FOR I = 16 TO 20 STEP 2
          II = I + 1
          ROTATE TX, CX, CY, V(J, I), V(J, II), XR, YR
          V(J, I) = XR: V(J, II) = YR
       NEXT I
       END SUB

       SUB FINALE
' A SUBROUTINE TO SOUND THE PROGRAM TERMINATION SIGNAL
'
       NOTE "3C0506"
       NOTE "4D0509"
       NOTE "5E0512"
       NOTE "6F0515"
       END SUB
```

```
        FUNCTION GAUSSIAN (Z)
' A FUNCTION TO SUPPLY THE GAUSSIAN PROBABILITY DENSITY FUNCTION
' OF THE NORMAL DEVIATE Z.
' ( NOTE: THE SCALING CONSTANT IS SUPPLIED INDEPENDENTLY AS CF )
'       ARGUMENT:
'          Z      THE NORMAL DEVIATE
'
        GAUSSIAN = EXP(-.5 * Z ^ 2)
        END FUNCTION

        SUB GETTER
' A SUBROUTINE TO ACCEPT A KEYSTROKE AS SA AND TO YIELD ITS ASCII CODE AS
IA
' ( SA AND IA ARE COMMON SHARED )
        DO
           SA = INKEY$
        LOOP UNTIL SA <> ""
        IA = ASC(SA)
        END SUB

        SUB GF (NDC, N, XM(), XF(), AM, VP, VS, VQ, VK)
' A SUBROUTINE TO COMPUTE THE ARITHMETIC MEAN, POPULATION STANDARD
DEVIATION, SAMPLE STANDARD DEVIATION,
' SKEWNESS AND KURTOSIS OF A GROUPED FREQUENCY DISTRIBUTION OF CLASS-
INTERVAL MID-POINTS XM(N) AND
' CLASS INTERVAL FREQUENCIES XF(N).
' THIS SUBROUTINE WILL ALSO COMPUTE THOSE DESCRIPTIVE STATISTICS FOR A
SIMPLE DATA SET IF THE VALUES
' ARE ASSIGNED TO XM(N) AND THE PAIRED XF(N) EACH SET TO UNITY.
' THIS METHOD IS BASED UPON THAT GIVEN IN "QUANTITATIVE GEOGRAPHY" BY COLE
AND KING ( SBN 471-16475-5 ).
' THE FORMULA FOR THE SAMPLE STANDARD DEVIATION OF A GROUPED FREQUENCY
DISTRIBUTION IS TAKEN FROM
' PAGE 70 OF "STATISTICAL ANALYSIS" BY YA-LUN CHOW ( ISBN 0-03-089422-0 ).
'       ARGUMENTS:
'          NDC    THE MAXIMUM NUMBER OF DATA PER CLUSTER
'          N      THE NUMBER OF CLASS INTERVALS
'          XM()   THE ARRAY OF CLASS INTERVAL MID-POINTS
'          XF()   THE ARRAY OF CLASS INTERVAL FREQUENCIES
'          AM     THE ARITHMETIC MEAN
'          VP     THE POPULATION STANDARD DEVIATION
'          VS     THE SAMPLE     STANDARD DEVIATION
'          VQ     THE SKEWNESS
'          VK     THE KURTOSIS
'
        REDIM W(NDC + 3, 4)
        N1 = N + 1: N2 = N + 2: N3 = N + 3
        FS = 0#: FOR I = 1 TO N: FS = FS + XF(I): NEXT I
        FOR I = 1 TO N
           FOR J = 1 TO 4
              W(I, J) = XF(I) * XM(I) ^ J
              W(N1, J) = W(N1, J) + W(I, J)
           NEXT J
        NEXT I
        FOR J = 1 TO 4: W(N2, J) = W(N1, J) / FS: NEXT J
        W(N3, 2) = W(N2, 2) - W(N2, 1) ^ 2
        W(N3, 3) = W(N2, 3) - 3 * W(N2, 1) * W(N2, 2) + 2 * W(N2, 1) ^ 3
        W(N3, 4) = W(N2, 4) - 4 * W(N2, 1) * W(N2, 3) + 6 * W(N2, 1) ^ 2 *
W(N2, 2) - 3 * W(N2, 1) ^ 4
        AM = W(N1, 1) / FS
        VP = SQR(W(N3, 2))
        DS = 0#: FOR I = 1 TO N: DS = DS + XF(I) * (XM(I) - AM) ^ 2: NEXT I
        VS = SQR((FS * W(N1, 2) - W(N1, 1) ^ 2) / (FS * (FS - 1)))
        VQ = W(N3, 3) / VP ^ 3
        VK = W(N3, 4) / VP ^ 4
```

```
        ERASE W
        END SUB

        SUB LAGRANGIAN (N, XL(), FL(), XT, FT)
' A SUBROUTINE TO COMPUTE A LAGRANGIAN INTERPOLATE FT BEING A FUNCTIONAL
VALUE
' AT ABSCISSOR XT FROM KNOWLEDGE OF THE N POLYNOMIAL CO-ORDINATES
(XL(),FL())
'       ARGUMENTS:
'          N       THE NUMBER OF KNOWN CO-ORDINATES
'          XL()    THE ARRAY  OF KNOWN X CO-ORDINATES
'          FL()    THE ARRAY  OF KNOWN Y CO-ORDINATES
'          XT      THE ABSCISSAL VALUE AT WHICH FT IS TO BE ESTIMATED
'          FT      THE INTERPOLATED FUNCTION OF XT
'
        FT = 0#
        FOR I = 1 TO N
           A = 1#
           FOR J = 1 TO I - 1: A = A * (XT - XL(J)) / (XL(I) - XL(J)): NEXT
J
           FOR J = I + 1 TO N: A = A * (XT - XL(J)) / (XL(I) - XL(J)): NEXT
J
           FT = FT + FL(I) * A
        NEXT I
        END SUB

        SUB MOMENTAL (N, P(), AM, VP, VS, VQ, VR)
' A SUBROUTINE TO COMPUTE THE MOMENTAL DESCRIPTIVE STATISTICS OF
' THE SIMPLE DATA DISTRIBUTION XM().
' THE METHOD IS TAKEN FROM PAGE 30 OF "STATISTICS IN GEOGRAPHY"
' ( SECOND EDITION ) BY DAVID EBDON ( ISBN 0-631-13688-6 ).
'       ARGUMENTS:
'          N       THE NUMBER OF DATA
'          P()     THE ARRAY OF DATA VALUES
'          AM      THE ARITHMETIC MEAN
'          VP      THE POPULATION STANDARD DEVIATION
'          VS      THE SAMPLE     STANDARD DEVIATION
'          VQ      THE MOMENTAL SKEWNESS
'          VR      THE MOMENTAL KURTOSIS
'
        AM = 0#: FOR I = 1 TO N: AM = AM + P(I): NEXT I: AM = AM / N
        VP = 0#: VS = 0#: VQ = 0#: VR = 0#
        FOR I = 1 TO N
           XD = P(I) - AM
           XX = XD * XD: VP = VP + XX
           XX = XX * XD: VQ = VQ + XX
           XX = XX * XD: VR = VR + XX
        NEXT I
        VS = SQR(VP / (N - 1)): VP = SQR(VP / N)
        VQ = VQ / (N * VP ^ 3): VR = VR / (N * VP ^ 4)
        END SUB

        SUB NOTE (S)
' A SUBROUTINE TO SOUND A NOTE UPON THE COMPUTER SPEAKER
'       ARGUMENT:
'          S     THE NOTE SPECIFIER STRING "IN$NL" e.g. "2B0506"
'                   I   THE OCTAVE NUMBER        ( 0-6 )
'                   N$  THE NOTE LETTER          ( ABCDEFG )
'                   N   THE NOTE NUMBER          ( 0-84 )
'                   L   THE LENGTH OF THE NOTE   ( 1-64 )
'
        I = VAL(MID$(S, 1, 1)): N$ = MID$(S, 2, 1)
        N = VAL(MID$(S, 3, 2)): L = VAL(MID$(S, 5, 2))
        PLAY "O" + STR$(I) + "N" + STR$(N) + "L" + STR$(L) + "X" +
VARPTR$(N$)
```

```
       END SUB

       SUB PICK (KT, I, L, R, S, IY, IX, MK, SF)
' A SUBROUTINE TO OBTAIN A VARIABLE OF TYPE KT AT SCREEN POSITION IY,IX
'      ARGUMENTS:
'         KT     THE DATUM TYPE CHOICE
'                   1   SHORT INTEGER
'                   2   LONG  INTEGER
'                   3   DOUBLE PRECISION REAL
'                   4   STRING
'         I      THE SHORT INTEGER
'         L      THE LONG INTEGER TO BE OBTAINED  ( OPTION )
'         R      THE REAL         TO BE OBTAINED  ( OPTION )
'         S      THE STRING       TO BE OBTAINED  ( OPTION )
'         IY     THE STARTING SCREEN ROW
'         IX     THE STARTING SCREEN COLUMN
'         MK     THE PRINTING COLOR
'         SF     THE PROPER ( OR DEFAULT ) PRINTING FORMAT
'                ( "SAnn" TRUNCATES A STRING TO nn CHARACTERS )
'
       SDEL = CHR$(0) + CHR$(83)
       COLOR MK: LOCATE IY, IX: IO = I: LO = L: RO = R: SOL = S: L1 =
LEN(SF)
       SELECT CASE KT
          CASE 1
             IF LEN(STR$(I)) > L1 THEN L1 = LEN(STR$(I))
          CASE 2
             IF LEN(STR$(L)) > L1 THEN L1 = LEN(STR$(L))
          CASE 3
             IF LEN(STR$(R)) > L1 THEN L1 = LEN(STR$(R))
          CASE 4
             L1 = LEN(S)
       END SELECT
       PRINT SPACE$(L1)
       DO
          LOCATE IY, IX: PRINT "."
          FOR II = 1 TO 10: NEXT II
          LOCATE IY, IX: PRINT " "
          SA = INKEY$
       LOOP UNTIL SA <> ""
       IA = ASC(SA)
       IF IA <> 13 THEN
          SCON = "": IT = IX: LOCATE IY, IX
             DO
                IF SA = SDEL THEN
                      IT = IT - 1
                      LOCATE IY, IT: PRINT SPACE$(1)
                      LOCATE IY, IT: SCON = LEFT$(SCON, LEN(SCON) - 1)
                   ELSE
                      SELECT CASE KT
                         CASE 1 TO 3
                            IF IA > 47 AND IA < 58 OR IA = 46 OR IA = 45
THEN
                                  SCON = SCON + SA
                                  PRINT SA;
                                  IT = IT + 1
                               ELSE
                                  IF IA <> 13 THEN NOTE "2B0506"
                            END IF
                         CASE 4
                            IF IA > 31 AND IA < 127 THEN
                                  SCON = SCON + SA
                                  PRINT SA;
                                  IT = IT + 1
                               ELSE
```

```
                              IF IA <> 13 THEN NOTE "2B0506"
                        END IF
                  END SELECT
            END IF
            GETTER
         LOOP UNTIL IA = 13
         SELECT CASE KT
            CASE 1
               I = INT(VAL(SCON) + .5)
            CASE 2
               L = INT(VAL(SCON) + .5)
            CASE 3
               R = VAL(SCON)
         END SELECT
      ELSE
         I = IO: L = LO: R = RO
   END IF
   L1 = LEN(SF): IF LEN(SCON) > L1 THEN L1 = LEN(SCON)
   LOCATE IY, IX: PRINT SPACE$(L1): LOCATE IY, IX
   SELECT CASE KT
      CASE 1
         PRINT USING SF; I
      CASE 2
         PRINT USING SF; L
      CASE 3
         PRINT USING SF; R
      CASE 4
         LOCATE IY, IX: PRINT SPACE$(LEN(SCON)): LOCATE IY, IX
         IF SOL <> "" AND SCON = "" THEN SCON = SOL
         IF LEFT$(SF, 2) = "SA" THEN
               S = LEFT$(SCON, VAL(MID$(SF, 3)))
            ELSE
               S = SCON
         END IF
         PRINT S
   END SELECT
   END SUB

   SUB PLACE (KT, I, L, R, S, IY, IX, IU, MK, JF, SF, IB)
' A SUBROUTINE TO PLACE A VARIABLE OF TYPE KT AT SCREEN POSITION IY, IX
' OR ALTERNATIVELY PLACE THE VARIABLE WITHIN A PADDED REPORT FILE
'      ARGUMENTS:
'         KT     THE DATUM TYPE CHOICE
'                   1   SHORT INTEGER
'                   2   LONG INTEGER
'                   3   DOUBLE PRECISION REAL
'                   4   STRING
'         I      THE SHORT INTEGER TO BE PRINTED  ( OPTION )
'         L      THE LONG  INTEGER TO BE PRINTED  ( OPTION )
'         R      THE REAL          TO BE PRINTED  ( OPTION )
'         S      THE STRING        TO BE PRINTED  ( OPTION )
'         IY     THE STARTING SCREEN ROW
'         IX     THE STARTING SCREEN COLUMN
'         IU     THE LOGICAL UNIT NUMBER
'                   1   PRINT TO THE SCREEN
'                   2   PRINT TO A REPORT FILE
'         MK     THE NOMINAL PRINTING COLOR
'         JF     THE LINE FEED SUPPRESSOR SWITCH
'                   0   FOLLOW WITH A LINE FEED
'                   1   DO NOT FOLLOW WITH A LINE FEED
'         SF     THE REQUIRED PRINTING FORMAT
'         IB     THE NUMBER OF FORWARD PADDING SPACES
'
   IF SF = "" THEN SF = "#########"
   COLOR MK
```

```
       SELECT CASE IU
          CASE 1
             LOCATE IY, IX
                SELECT CASE KT
                   CASE 1
                      PRINT USING SF; I
                   CASE 2
                      PRINT USING SF; L
                   CASE 3
                      PRINT USING SF; R
                   CASE 4
                      PRINT S
                END SELECT
          CASE 2
             IF IB > 0 THEN PRINT #IU, SPACE$(IB);
             IF JF = 1 THEN SCC = ";" ELSE SCC = CHR$(13) + CHR$(10)
                SELECT CASE KT
                   CASE 1
                      PRINT #IU, USING SF; I; SCC
                   CASE 2
                      PRINT #IU, USING SF; L; SCC
                   CASE 3
                      PRINT #IU, USING SF; R; SCC
                   CASE 4
                      PRINT #IU, S
                END SELECT
       IF JF = 0 THEN PRINT #LU,
       END SELECT
       END SUB

       SUB REFINERY (PI, TL, UV, PV)
' A SUBROUTINE TO IMPROVE AN APPROXIMATE NORMAL DEVIATE
'      ARGUMENTS:
'         PI     THE LUDOLPHINE CONSTANT
'         TL     THE REQUIRED ( PLUS OR MINUS ) TOLERANCE
'         UV     THE ESTIMATED NORMAL DEVIATE
'         PV     THE PROBABILITY ATTACHING TO THE TRUE DEVIATE
'
       DIM XL(6), FL(6)
       PV = .5# - PV
       M = 3: N = 7: TM = .00045#: CF = 1 / SQR(2 * PI)
       DO
          XL(1) = UV - TM: XL(2) = UV: XL(3) = UV + TM
          FOR I = 1 TO M
             ROMBERG N, CF, 0, XL(I), FL(I)
          NEXT I
          UW = UV
          LAGRANGIAN M, FL(), XL(), PV, UV
          TM = ABS(UV - UW)
       LOOP UNTIL TM < TL
       PV = .5# - PV
       END SUB

       SUB RESULTHETA (K, PI, TA())
' A SUBROUTINE TO COMPUTE THE CENTRAL TENDENCY OF
' THE K-1 CLUSTER DIRECTIONAL ANGLES TA().
' THE VECTORIAL RESULTANT MEAN DIRECTIONAL ANGLE IS ASSIGNED TO TA(K)
'      ARGUMENTS:
'         K      THE NUMBER OF CLUSTERS
'         PI     THE LUDOLPHINE CONSTANT
'         TA()   THE ARRAY OF DIRECTIONAL ANGLES
'
       J = 1: TC = 0#: TS = 0#
       FOR I = 1 TO K - 1
          TC = TC + COS(TA(I)): TS = TS + SIN(TA(I))
```

```
        NEXT I
        IF TC >= 0# THEN
           IF TS >= 0# THEN
                 J = 0
              ELSE
                 J = 2
           END IF
        END IF
        TA(K) = J * PI + ATN(TS / TC)
        END SUB

        SUB ROMBERG (NT, CF, FL, FU, RI)
' A SUBROUTINE TO PERFORM A ROMBURG INTEGRATION OF A USER-SPECIFIED
FUNCTION
' BETWEEN FL AND FU.
' CF IS THE INTEGRAL SCALING CONSTANT ( UNITY IF ABSENT ).
' ( NOTE: SEGMENT ROMBERG IS CURRENTLY CONFIGURED FOR THE GAUSSIAN
PROBABILITY INTEGRAL )
'       ARGUMENTS:
'          NT     THE NUMBER OF TRAPEZOIDAL INTEGRATIONS TO BE PERFORMED (
NT < 10 )
'          CF     THE INTEGRAL SCALING CONSTANT
'          FL     THE INTEGRAL LOWER BOUND
'          FU     THE INTEGRAL UPPER BOUND
'          RI     THE ROMBERG INTEGRAL
'
        DIM WA(257), WB(8, 9)
        NS = NT - 1: MI = 2 ^ NS: MJ = MI + 1: DS = (FU - FL) / MI
        WA(1) = GAUSSIAN(FL): WA(MJ) = GAUSSIAN(FU)
        FOR I = 2 TO MI: WA(I) = 2 * GAUSSIAN(FL + (I - 1) * DS): NEXT I
        FOR J = 0 TO NS
           II = 2 ^ J: TT = 0!: FOR I = 1 TO MJ STEP II: TT = TT + WA(I):
NEXT I
           WB(0, NT - J) = .5# * II * DS * TT
        NEXT J
        FOR J = 1 TO NS
           JJ = 2 ^ (2 * J)
           FOR I = J + 1 TO NT: WB(J, I) = (JJ * WB(J - 1, I) - WB(J - 1, I
- 1)) / (JJ - 1): NEXT I
        NEXT J
        RI = CF * WB(NS, NT)
        END SUB

        SUB ROTATE (TH, XO, YO, XI, YI, XR, YR)
' A SUBROUTINE TO ROTATE A PAIR OF CARTESIAN CO-ORDINATES (X,Y)
' DEXTRALLY ABOUT XO,YO THROUGH ANGLE TH
'       ARGUMENTS:
'          TH     THE ANGLE OF ROTATION ( RADIANS )
'          XO     THE ABSCISSAL CENTER OF ROTATION
'          YO     THE ORDINAL   CENTER OF ROTATION
'          XI     THE UNROTATED X CO-ORDINATE
'          YI     THE UNROTATED Y CO-ORDINATE
'          XR     THE ROTATED   X CO-ORDINATE
'          YR     THE ROTATED   Y CO-ORDINATE
'
        AC = COS(TH): AI = SIN(TH): X = XI - XO: Y = YI - YO
        XR = XO + X * AC + Y * AI
        YR = YO + Y * AC - X * AI
        END SUB

        SUB STANDARD (CB, CC, PI, PH, K, CK, DR(), AX, EM, AO, V())
' A SUBROUTINE TO COMPUTE THE VERTEX CO-ORDINATES OF THE STANDARD BELEMNOID
'       ARGUMENTS:
'          CB     THE REPRESENTED MAGNITUDE OF THE STANDARD BELEMNOID AXIS (
MAGNITUDE UNITS )
```

```
'          CC      THE STANDARD BELEMNOID AXIS LENGTH ( MAP UNITS )
'          PI      THE LUDOLPHINE CONSTANT
'          PH      THE RATIO OF PHIDIAS
'          K       THE NUMBER OF CLUSTERS
'          CK      THE SCALING CONSTANT
'          DR()    THE ARRAY OF DYNAMIC RADII
'          AX      THE MEAN OF MAGNITUDE MEANS
'          EM      THE HIGHEST EASTING
'          AO      THE LOWEST NORTHING
'          DT()    THE ARRAY OF BELEMNOID VERTEX CO-ORDINATES
'          V()     THE ARRAY OF OUTPUT STATISTICS
'
       IF CB <= 0# THEN
             MOMENTAL K, DR(), AM, VP, VS, VQ, VR
             AU = AM * (1 + 1 / PH ^ 2): CK = AX / AU
          ELSE
             AU = CC: AM = CC / (1 + 1 / PH ^ 2): CK = CB / CC
       END IF
       AG = SQR(2 * AM ^ 2 * (1 - COS(2 * PI / 5)))
       HS = (AG / 2) * TAN(PI / 5)
       V(0, 14) = EM - AG / 2: V(0, 15) = AO + HS
       V(0, 16) = EM - AG: V(0, 17) = AO
       V(0, 18) = EM - AG / 2: V(0, 19) = AO + HS + AU
       V(0, 20) = EM: V(0, 21) = AO
       END SUB

       FUNCTION T (JD, AL)
' A FUNCTION TO SUPPLY STUDENT'S t.
' THIS FUNCTION IS ADAPTED FROM BY JAMES R WARREN FROM
' THE ORIGINAL ALGOL SOURCE OF SEGMENT STUDENT'S t-QUANTILE
' BY GW HILL ( CACM V13 N10 PAGE 620 OCTOBER 1970 )
'      ARGUMENTS:
'          JD      THE DEGREES OF FREEDOM
'          AL      THE INTEGRAL PROBABILITY
' ( HP IS COMMON SHARED )
'
       AL1 = 2 * AL
       SELECT CASE JD
          CASE 1
             P = AL1 * HP: T = COS(P) / SIN(P)
          CASE 2
             T = SQR(2 / (AL1 * (2 - AL1)) - 2)
          CASE ELSE
             A = 1 / (JD - .5)
             B = 48 / A ^ 2
             C = ((20700 * A / B - 98) * A - 16) * A + 96.36
             D = ((94.5 / (B + C) - 3) / B + 1) * SQR(A * HP) * JD
             X = D * AL1: Y = X ^ (2 / JD)
             IF Y > .05 + A THEN
                   X = TND(.5 * AL1): Y = X ^ 2
                   IF JD < 5 THEN C = C + .3 * (JD - 4.5) * (X + .6)
                   C = (((.05 * D * X - 5) * X - 7) * X - 2) * X + B + C
                   Y = (((((.4 * Y + 6.3) * Y + 36) * Y + 94.5) / C - Y -
3) / B + 1) * X
                   Y = A * Y ^ 2
                      IF Y > .002 THEN
                            Y = EXP(Y) - 1
                         ELSE
                            Y = .5 * Y ^ 2 + Y
                      END IF
                ELSE
                   Y = ((1 / (((JD + 6) / (JD * Y) - .089 * D - .822) * (JD
+ 2) * 3) + .5 / (JD + 4)) * Y - 1) * (JD + 1) / (JD + 2) + 1 / Y
             END IF
             T = SQR(JD * Y)
```

```
       END SELECT
       END FUNCTION

       SUB THETA (J, PI, TA(), V())
' A SUBROUTINE TO COMPUTE A DIRECTIONAL ANGLE TA(J)
'      ARGUMENTS:
'         J         THE CLUSTER NUMBER
'         PI        THE LUDOLPHINE CONSTANT
'         TA()      THE ARRAY OF DIRECTIONAL ANGLES
'         V(J,3)    THE ROW OF CENTROID X CO-ORDINATES
'         V(J,4)    THE ROW OF CENTROID Y CO-ORDINATES
'
       TX = 0#: IF V(J, 4) >= V(J + 1, 4) THEN TX = PI
       TA(J) = TX + ATN((V(J + 1, 3) - V(J, 3)) / (V(J + 1, 4) - V(J, 4)))
       END SUB

       FUNCTION TND (AL)
' A FUNCTION TO APPROXIMATE A NEGATIVE NORMAL DEVIATE.
' THIS ROUTINE EMPLOYS A POLYNOMIAL APPROXIMATOR DUE TO HASTINGS
' TAKEN FROM 26.2.23 ON PAGE 933 OF "HANDBOOK OF MATHEMATICAL FUNCTIONS"
' BY ABRAMOWITZ AND STEGUN ( ISBN 0-486-61272-4 ).
' SEGMENT REFINERY DUE TO JAMES R WARREN IS AN ITERATED COUPLE OF
' ROMBERG INTEGRATION AND FOUR-POINT LAGRANGIAN INTERPOLATION
' WHICH TAKES THE APPROXIMATION TO TWELVE-FIGURE ACCURACY
'      ARGUMENT:
'         AL     THE INTEGRAL PROBABILITY
'
       DIM C(6)
       PI = 3.141592653589793#
       TL = .00000001#
       C(1) = 2.515517: C(2) = .802853: C(3) = .010328
       C(4) = 1.432788: C(5) = .189269: C(6) = .001308
       U = SQR(LOG(1 / AL ^ 2))
       U1 = C(1) + C(2) * U + C(3) * U ^ 2
       U2 = 1 + C(4) * U + C(5) * U ^ 2 + C(6) * U ^ 3
       UV = U - U1 / U2
       REFINERY PI, TL, UV, AL
       TND = -UV
       END FUNCTION
```

Appendix Two

The Program Validation Data BELVALID.CAL

BELTEST

CLUSTER NAME	STANDARD	1960	1970	1980	1990
ITEMS IN CLUSTER		3	5	5	6
CLUSTER CENTROID EASTING		44.212	95.5483	77.5896	36.0021
CLUSTER CENTROID NORTHING		20.9695	51.3765	133.0116	91.7965
CLUSTER DYNAMIC RADIUS		12.6943	29.1896	23.977	10.4295
VALUES ARITHMETIC MEAN		404	685.8	1104.4	321
VALUES POPULATION SD		53.93206	268.8832	1818.1	218.2376
VALUES SAMPLE SD		66.05301	300.6205	2032.698	239.0674
VALUES SKEWNESS		-0.525032	-0.73474	1.478166	-0.226761
VALUES KURTOSIS		1.5	2.055144	3.221091	1.592853
NORMATIVE SAMPLE SIZE		33	279	12737	152
LOWER CONFIDENCE LIMIT		239.902	312.5902	-1419.123	70.07339
UPPER CONFIDENCE LIMIT		568.098	1059.01	3627.923	571.9266
BELEMNOID CENTROID X	138.7894	44.212	95.5483	77.5896	36.0021
BELEMNOID CENTROID Y	18.145	20.9695	51.3765	133.0116	91.7965
BELEMNOID LEFT FOOT X	127.5788	42.9294	90.9241	124.0766	31.7177
LELEMNOID LEFT FOOT Y	10	21.5235	46.1669	125.4343	88.1074
BELEMNOID APEX X	138.7894	58.7826	89.372	44.7082	34.8926
BELEMNOID APEX Y	44.5027	29.5998	79.4522	100.4246	105.2062
BELEMNOID RIGHT FOOT X	150	44.0815	101.9318	70.4308	40.8345
BELEMNOID RIGHT FOOT Y	10	19.5784	48.5885	179.5649	88.8618

Standard Belemnoid Mu = 628.8 Metric Tonnes

**** BELEM.BAS Validation Document BELVALID.CAL

Appendix Three

The Input and Output Data for
Test Files BELTEST, BELTEST0 and BELTEST1

```
Metric Tonnes
0,25
150,10
50,0.95
4,16
1960,3,1,2,3
1970,5,2,4,5,6,7
1980,5,5,6,8,9,10
1990,6,11,12,13,14,15,16
ABLE,BAKER,CARTER,DEACON,EDWARDS,FRANCIS,GRANVILLE,HARWELL,ISMAY,JARROLD,KELD,LATIMER,MORRIS,NUGENT,OSWALD,PYE
30,50,55,105,105,110,115,80,75,70,50,30,40,30,40,50
20,30,15,50,65,60,55,135,140,135,100,100,95,90,85,80
457,330,425,,,,,,,,,,,,,
,842,,223,551,846,967,,,,,,,,,
,,,,94,374,,1,4730,323,,,,,,
,,,,,,,,,,14,516,380,610,64,342
```

```
Metric Tonnes
0,25
150,10
50,0.95
4,16
1960,3,1,2,3
1970,5,2,4,5,6,7
1980,5,5,6,8,9,10
1990,6,11,12,13,14,15,16
ABLE,BAKER,CARTER,DEACON,EDWARDS,FRANCIS,GRANVILLE,HARWELL,ISMAY,JARROLD,KELD,LATIMER,MORRIS,NUGENT,OSWALD,PYE
30,50,55,105,105,110,115,80,75,70,50,30,40,30,40,50
20,30,15,50,65,60,55,135,140,135,100,100,95,90,85,80
457,330,425,0,0,0,0,0,0,0,0,0,0,0,0,0
0,842,0,223,551,846,967,0,0,0,0,0,0,0,0,0
0,0,0,0,94,374,0,1,4730,323,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,14,516,380,610,64,342
```

```
Metric Tonnes
800,20
150,10
50,0.95
4,16
1960,3,1,2,3
1970,5,2,4,5,6,7
1980,5,5,6,8,9,10
1990,6,11,12,13,14,15,16
ABLE,BAKER,CARTER,DEACON,EDWARDS,FRANCIS,GRANVILLE,HARWELL,ISMAY,JARROLD,KELD,LATIMER,MORRIS,NUGENT,OSWALD,PYE
30,50,55,105,105,110,115,80,75,70,50,30,40,30,40,50
20,30,15,50,65,60,55,135,140,135,100,100,95,90,85,80
457,330,425,0,0,0,0,0,0,0,0,0,0,0,0,0
0,842,0,223,551,846,967,0,0,0,0,0,0,0,0,0
0,0,0,0,94,374,0,1,4730,323,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,14,516,380,610,64,342
```

BELTEST

CLUSTER NAME	STANDARD	1960	1970	1980	1990
ITEMS IN CLUSTER	0	3	5	5	6
CLUSTER CENTROID EASTING	0	44.21205	95.54826	77.58964	36.00208
CLUSTER CENTROID NORTHING	0	20.96947	51.37649	133.0116	91.79647
CLUSTER DYNAMIC RADIUS	0	12.69431	29.18961	23.97703	10.42951
VALUES ARITHMETIC MEAN	0	404	685.8	1104.4	321
VALUES POPULATION SD	0	53.93206	268.8832	1818.1	218.2376
VALUES SAMPLE SD	0	66.05301	300.6205	2032.698	239.0674
VALUES SKEWNESS	0	-0.525032	-0.73474	1.478166	-0.226761
VALUES KURTOSIS	0	1.5	2.055144	3.221091	1.592853
NORMATIVE SAMPLE SIZE	0	33	279	12741	152
LOWER CONFIDENCE LIMIT	0	239.9152	312.5303	-1419.528	70.11421
UPPER CONFIDENCE LIMIT	0	568.0848	1059.07	3628.328	571.8858
BELEMNOID CENTROID X	138.7893982	44.21205	95.54826	77.58964	36.00208
BELEMNOID CENTROID Y	18.144979	20.96947	51.37649	133.0116	91.79647
BELEMNOID LEFT FOOT X	127.5787963	42.9294	90.92409	124.0766	31.71769
BELEMNOID LEFT FOOT Y	10	21.52349	46.16694	125.4343	88.10743
BELEMNOID APEX X	138.7893982	58.78259	89.37199	44.70818	34.89256
BELEMNOID APEX Y	44.50268472	29.59977	79.45217	100.4246	105.2062
BELEMNOID RIGHT FOOT X	150	44.0815	101.9318	70.43082	40.83451
BELEMNOID RIGHT FOOT Y	10	19.5784	48.58849	179.5649	88.86175

Standard Belemnoid Mu = 628.8 Metric Tonnes

BELTEST0

CLUSTER NAME	STANDARD	1960	1970	1980	1990
ITEMS IN CLUSTER	0	3	5	5	6
CLUSTER CENTROID EASTING	0	44.21205	95.54826	77.58964	36.00208
CLUSTER CENTROID NORTHING	0	20.96947	51.37649	133.0116	91.79647
CLUSTER DYNAMIC RADIUS	0	12.69431	29.18961	23.97703	10.42951
VALUES ARITHMETIC MEAN	0	404	685.8	1104.4	321
VALUES POPULATION SD	0	53.93206	268.8832	1818.1	218.2376
VALUES SAMPLE SD	0	66.05301	300.6205	2032.698	239.0674
VALUES SKEWNESS	0	-0.525032	-0.73474	1.478166	-0.226761
VALUES KURTOSIS	0	1.5	2.055144	3.221091	1.592853
NORMATIVE SAMPLE SIZE	0	33	279	12741	152
LOWER CONFIDENCE LIMIT	0	239.9152	312.5303	-1419.528	70.11421
UPPER CONFIDENCE LIMIT	0	568.0848	1059.07	3628.328	571.8858
BELEMNOID CENTROID X	138.7893982	44.21205	95.54826	77.58964	36.00208
BELEMNOID CENTROID Y	18.144979	20.96947	51.37649	133.0116	91.79647
BELEMNOID LEFT FOOT X	127.5787963	42.9294	90.92409	124.0766	31.71769
BELEMNOID LEFT FOOT Y	10	21.52349	46.16694	125.4343	88.10743
BELEMNOID APEX X	138.7893982	58.78259	89.37199	44.70818	34.89256
BELEMNOID APEX Y	44.50268472	29.59977	79.45217	100.4246	105.2062
BELEMNOID RIGHT FOOT X	150	44.0815	101.9318	70.43082	40.83451
BELEMNOID RIGHT FOOT Y	10	19.5784	48.58849	179.5649	88.86175

Standard Belemnoid Mu = 628.8 Metric Tonnes

BELTEST1

CLUSTER NAME	STANDARD	1960	1970	1980	1990
ITEMS IN CLUSTER	0	3	5	5	6
CLUSTER CENTROID EASTING	0	44.21205	95.54826	77.58964	36.00208
CLUSTER CENTROID NORTHING	0	20.96947	51.37649	133.0116	91.79647
CLUSTER DYNAMIC RADIUS	0	12.69431	29.18961	23.97703	10.42951
VALUES ARITHMETIC MEAN	0	404	685.8	1104.4	321
VALUES POPULATION SD	0	53.93206	268.8832	1818.1	218.2376
VALUES SAMPLE SD	0	66.05301	300.6205	2032.698	239.0674
VALUES SKEWNESS	0	-0.525032	-0.73474	1.478166	-0.226761
VALUES KURTOSIS	0	1.5	2.055144	3.221091	1.592853
NORMATIVE SAMPLE SIZE	0	33	279	12741	152
LOWER CONFIDENCE LIMIT	0	239.9152	312.5303	-1419.528	70.11421
UPPER CONFIDENCE LIMIT	0	568.0848	1059.07	3628.328	571.8858
BELEMNOID CENTROID X	141.4934919	44.21205	95.54826	77.58964	36.00208
BELEMNOID CENTROID Y	16.18033989	20.96947	51.37649	133.0116	91.79647
BELEMNOID LEFT FOOT X	132.9869838	43.44706	92.79036	105.3149	33.44683
BELEMNOID LEFT FOOT Y	10	21.2999	48.26947	128.4924	89.59629
BELEMNOID APEX X	141.4934919	52.90206	91.86467	57.97881	35.34035
BELEMNOID APEX Y	36.18033989	26.11667	68.12111	113.5764	99.79414
BELEMNOID RIGHT FOOT X	150	44.13419	99.35546	73.32005	38.88419
BELEMNOID RIGHT FOOT Y	10	20.13982	49.7137	160.7764	90.04618

Standard Belemnoid Mu = 800 Metric Tonnes