FINDING LEVERAGE GROUPS*

David E. Coleman

Working Paper 195

National Bureau of Economic Research, Inc. 575 Technology Square Cambridge, Massachusetts 02139

August 1977

* This work was supported by the National Science Foundation under Grant MCS76-11989 to the NBER Computer Research Center.

Abstract

A brief discussion of recent methods using the Hat Matrix for identifying leverage points, and clustering techniques for finding groups of data points is presented. The problem of identifying leverage groups is addressed, and a heuristic algorithm for identifying both leverage points and leverage groups is proposed. Semi-portable FORTRAN code implementing the algorithm, a sample terminal session, and a discussion of the terminal session are included.

Acknowledgements

The author, especially, would like to thank Steve Peters for many helpful discussions, a useful derivation, and for running the software on a 506 by 13 matrix. Virginia Klema, Roy Welsch, Ed Kuh, and Sally Coleman also provided useful insight. Sandra Moriarty gave extensive technical assistance.

<u>Key Words and Phrases</u>: leverage, outliers, least squares, cluster analysis, data analysis, hat matrix, mathematical software. CR Categories 5.14 and 5.5.

Table of Contents

																											-	Page
Introduction	•	•	•	•	•	•	• .	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	1
A problem	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	3
The Data Poir	nt	A.	lgo	ori	itł	m	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	5
An Example .	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•	7
Appendix 1 .	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	10
Appendix 2 .	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• .	•	•	•	•	•	•	13
Appendix 3.	•	•	•	•	•	•	•	•	•	•	•.	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	14
Appendix 4 .	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	19
Appendix 5.	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	23
References .	•	•	•	•		•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	24
Figures																												
Figure 1	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	4
Figure 2	•					•		•	•	•		•		•		•	•		•	•	•		•.	•	•	•	•	11

.

. 11

. .

Figure 3 . . .

. . . .

Introduction

Of primary concern in regression (least squares), $y = X\beta + \varepsilon$, is that the X matrix be non-singular and well-conditioned. A secondary concern, sometimes neglected, is the distribution of data (sample) points (rows of X) over the space spanned by the columns of non-singular X. Although it is desirable, and frequently assumed to be true that the data is normally distributed (in each column), this often is not the case. Two issues then arise, the presence of leverage points, and the presence of clusters (groups) of points.

Conceptually, a leverage point is far away (in some sense) from other points and their centroid; it is an outlier in X. If p (for X, n by p) is larger than, say, 3 it is hard to spot leverage points by eye or scatter plot because the hyper-parallelopiped representing the observation space has 2^{p} vertices. Furthermore, leverage is a relative property involving n(n-1)/2 interpoint relationships. What is needed is a metric under which each data point can be assigned a number indicating its leverage.

Hoaglin and Welsch [5] present the use of the so called "Hat Matrix", H, to examine the distribution of data points. In particular, they use the diagonal elements, h_i , of H as indicators of leverage, as is motivated by the derivation of H: Letting X^T stand for the transpose of X, $(X^TX)^{-1}$ stand for the matrix inverse of X^TX , $\hat{\beta}$ stand for the computed approximation to β , and \hat{y} stand for the fit realized at the least squares solution $X\hat{\beta}$ we have

 $x^{T}x\beta = x^{T}y, \hat{\beta} = (x^{T}x)^{-1}x^{T}y, x\hat{\beta} = \hat{y} = x(x^{T}x)^{-1}x^{T}y.$

If we set $H = X(X^TX)^{-1}X^T$ we have $\hat{y} = Hy$; H "puts the hat" on y. The leverage of the ith row of X, X_i , is seen in the influence of y_i on the fit \hat{y}_i , through h_i . Since H is a symmetric, idempotent matrix (a projection matrix), the h_i lie between 0 and 1. In their recent paper, Welsch and Kuh [8] develop the use of the h_i and related regression statistics. They define a cutoff level of 2p/n (for n > 2p) above which an h_i is considered significant and row i is called a leverage point.* Andrews and Pregibon [1] have developed another technique in which points with large h_i 's are considered leverage points, and minors of X^TX are computed in order to identify groups of leverage points (leverage groups).

The problem of identifying clusters, or groups, has been approached in many ways. As in the leverage point problem, nonhierarchical cluster analysis^{***} is multidimensional in nature, and seeks to reduce $O(n^2)$ interpoint relationships to n relationships, where each point is assigned to a cluster on the basis of some specified criterion, often involving Euclidean distance. Kendall and Stuart [4] give a heuristic procedure using ranking which is moderately successful in partitioning data into groups. Gnanadesikan [3], in his chapter, "Multidimensional Classification and Clustering," and Oliver [6] in his software documentation on Cluster Analysis routines describe a number of different clustering criteria and clustering procedures, but the complexity of the problem constrains the algorithm to be molded by its context. Since we are interested only in leverage groups, we will want to use criteria peculiar to assessing leverage.

See Appendices 1 and 2.

** See Gnanadesikan [3]. - 2 -

A Problem

As discussed by Welsch and Kuh [8], the h_i effectively reveal individual leverage points, but may not reveal those leverage points that geometrically form a group (are in close geometric proximity to one another). Proximity to other data reduces the individual leverage, hence the h_i , of any given point.

A simple example makes this clear. Consider X which consists of a cloud of 20 points centered at the origin, uniformly randomly distributed within a 5-space hypercube of side length 4, plus a point at (10, 10, 10, 10, 10). The latter point has h_{21} of about .951, close to the maximum value of 1. When a 22nd point is added nearby, at (10.1, 10.1, 10.1, 10.1, 10.1) we find that h_{21} and h_{22} are about .483 and .492. A 23rd point at (10.2, 10.2, 10.2, 10.2, 10.2) yields h_{21} , h_{22} , and h_{23} of .321, 328, and .334. These h_i contrast to others corresponding to points within the cloud, which are as high as .340, .425, .469, and 482.

Sequential row deletion is unreliable because it is hard to determine what constitutes a group, and a group could collectively have high leverage, while the h_i of its members might be moderate. The sequential procedure proposed by Andrews and Pregibon [1] can also encounter difficulties for the same reasons. Welsch and Kuh [8] mention the possibility of identifying groups through the correlation matrix of the residuals, but as they note, this requires the computation of the n(n-1)/2 elements, h_{ij} , which requires either considerably more storage or an $0(n^2p^2)$ -operations algorithm. If groups can be identified, we might prefer to replace row deletion with the substitution of a group by the mean (or some other summary measure) of its members. This way, crucial or expensive data is not lost, and the h_i convey more information. Welsch and Kuh [8] discuss other possible remedies.

- 3 -



la)

X,

0

0

X,





Figure 1

- la) Measuring the parallel
 distance of point j from
 point i .
- 1b) Finding outdistancers.
- lc) Finding leverage groups headed by outdistancers.





 \odot

The above comprises the motivation for a heuristic algorithm which can be used to help identify leverage points and leverage groups. The "Data Point Algorithm" (DPA) is $O(n^2p)$ operations, and requires little extra storage beyond that of the X matrix, and thus is comparable in cost to obtaining the h_i 's, and less expensive than obtaining the h_{ij} 's or $R_{ij}^{(k)}$'s proposed by Andrews and Pregibon [1]:

Data Point Algorithm

- 1. Given X, n by p with all constant columns deleted.
- 2. Center the data; $X \leftarrow X \overline{X}$, where the rows of \overline{X} are identically the column means of X. (The origin is now the centroid).
- 3. Normalize each column by dividing by its l_{∞} norm* times 2(p^{1/2}) (The main diagonal of the observation space hypercube is now of length 1).
- 4. Compute and store the l_2 norm** of each point (row).
- 5. Compute for each point the "normal" distance to all other points, that is, distance parallel to its normal vector, (see Figure 1a). Tallythose points further out in the normal direction (those with negative parallel distances). Sum the (scaled) inverses of these distances for each point, to obtain a measure of local density.
- 6. Single out those points with outdistance (further out) tallies of 0, particularly those that have large l_2 norms (relative to the others, and to the maximum, 0.5). We call these points "outdistancers" (see Figure 1b).

* Given vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$, the ℓ_{∞} norm of \mathbf{x} , $||\mathbf{x}||_{\infty} = \max_{\substack{\mathbf{x} \in \mathbf{x}_1 \\ 1 \leq i \leq n}} |\mathbf{x}_i|$ ** Given vector $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n)^T$, the ℓ_2 norm of \mathbf{x} , $||\mathbf{x}||_2 = (\sum_{i=1}^n \mathbf{x}_i^2)^{1/2}$

 $= (\mathbf{x}^{T_{\mathbf{x}}})^{1/2}$

5.

- 7. Each outdistancer is a leverage point, or the point furthest out in a leverage group. A relatively low "density" value means a point is isolated, a high value indicates the proximity (in the normal direction) of other points.
- 8. Get a sorted listing (possibly via Tukey [7], and Hoaglin and Wasserman's "Stem-and-Leaf" display) of all points and their normal distance to each outdistancer. Establish a cutoff level for normal distances, below which points form a leverage group "headed" by the outdistancer (see Figure 1c).

A listing of a semi-portable interactive driver, DPA FORTRAN, and the initialization routine, MATRIX FORTRAN, which implement the DPA algorithm can be found in Appendix 3.

By centering and normalizing the data, norms and distances can be compared. The further out a given point is from the origin (the centroid) and the fewer points are further out - the more leverage it exerts. The point furthest out in any normal direction exerts the most leverage in that direction. Any such point may be isolated, part of a tight group, or anywhere on the continuum in-between. Again, we emphasize that the group-inclusion function imposes a discrete, binary set of relationships on a complex, continuous configuration, so there always is some arbitrariness and simplification. For our purposes, we would seem to reduce complexity by measuring distances only in the normal directions (perpendicular distances are not used), but we increase complexity because normal distances are nonsymmetric, $d_1Rd_2+ d_2 Rd_1$, unlike Euclidean distances. Thus leverage groups are "headed" by outdistancing leverage points. An example makes the above discussion clearer.



An Example

We return to the example discussed above, X comprised of twenty points in a cloud about the origin and three points around (10, 10, 10, 10, 10). Appendix 4 contains the terminal session with DPA FORTRAN, to which the reader should refer.*

DPA FORTRAN carries out steps 1) - 5) of the Data Point Algorithm. Examining the OUTDIS column, we see that points 8, 10, 17, 18, and 23 are outdistancers. Point 23 especially catches our eye because its norm is listed as .5, the highest possible value. We now proceed to sequentially examine the 5 points singled out by step 6), using the Stem-and-Leaf display (SLD) [7]. The SLD for point 8 is done in units of 10^{-2} , first of all indicating that all but the three points isolated at the bottom of the display are relatively close to point 8 (.01 is small relative to .5). Nonetheless, the SLD does show a well defined break in distances, at about .04. DPA identifies points 17 and 19 to be part of the indicated group. We adopt a convenient notation for leverage groups: (norm, cutoff value, cutoff separation, outdistancer: other points in group), so we list the first leverage group identified as (.134, .04, .02, 8: 17, 19). The norm indicates the extent of leverage, (low in this case). The cutoff distance indicates the approximate minimum normal-distance radius used to define (contain) the group, (small, in this case). The cutoff separation indicates the extent to which the group is isolated from the other points (also small, in this case). Lastly, the header (outdistancer) of the group, and the group members are listed.

Execution was on an IBM VM370/158 computer, FORTRAN H(OPT(2)) compiler.

Continuing with the example, DPA finds (.112, .01, .02, 10:17, 18) which means that two weak leverage groups overlap at point 17, (.147, -, -, 17:-)which has no well-defined cutoff value, and (.114, .01, .03, 18:10). DPA clearly identifies the leverage group near (10, 10, 10, 10, 10, 10) in this contrived example: (.500, .02, .38, 23:21, 22).

Turning to some "real" data, the example considered by Welsch and Kuh [8] taken from an econometric study of life-cycle savings rates) serves as a good case for comparison of the use of the h;, and the Data Point Algorithm." The h_i identify points 49, 44, 23, and 21 to be leverage points (in order of decreasing h_i) and 37, 6, 47, 14, and 39 to be "contenders". DPA FORTRAN indicates that of 49, 44, 23, and 21, only 49 is an outdistancer; 44 is outdistanced by 39, 23 by 28, and 21 by 2, 3, 14, 25, 34, 40 and 43. No clear leverage groups are indicated; 18, 37, 39, and 49 are all outdistancers, but SLD's reveal no significant breaks in the sorted normal distances. The design of DPA FORTRAN allows the user to identify "secondary" leverage groups those headed by a point outdistanced by only a few other points. We call such points "k-outdistancers" where k is the number of outdistancing points. DPA FORTRAN lists as 1-outdistancers points 14, 23, 25, 43, 44, and 50. By defining a new generalized data structure for leverage groups headed by k-outdistancers: (norm, cutoff value, cutoff separation, k-outdistancer : (outdistancing points), other points in group) we can conveniently display the fact that point 25 has a norm of .311, is a 1-outdistancer (outdistanced by point 39) and with cutoff value of .05 and cutoff separation of .03 it heads a group containing points 2, 3, 11, 14, 15, 40, and 43:

(.311, .05, .03, 25:(39), 2, 3, 11, 14, 15, 40, 43).

See Appendix 5.

- 8 -

We also have

(.320, .05, .03, 43:(39), 2, 3, 11, 14, 25, 40).

The other 1-outdistancers are uninteresting.

In conclusion, DPA FORTRAN shows points 39, 49, 18, and 37 (in order of decreasing norm) to be outdistancers, each with a roughly uniformly distributed set of neighbors in the direction towards the origin (centroid). Loosely speaking, points 25 and 43 head up a leverage group outdistanced only by point 39, and containing points 2, 3, 11, 14, and 40. This set of data does not appear to contain any remarkable features in the way of leverage points or groups.



Appendix 1

X and Augmented X

An issue in the leverage point (group) problem is whether to search for leverage points in X, or in X augmented by the right-hand side; y: X|y. The appeal of using X|y is that it contains all input data, and a leverage measure, such as h_i^* (the diagonal of the hat matrix for X|y) can be computed for each point $X_i|y_i$. The crucial disadvantage of using X|y is that such a measure as h_i^* can blur what are two distinct cases: leverage points in X, and outliers in y. A leverage point in X, X_j , is a point that (because of its position relative to other points in X) has considerable influence on the fit, regardless of the value y_j . An outlier in X|y is a point, $X_j|y_j$, with a y_j significantly deviant from the fit at X_j obtained by fitting with all but point j.

Some indication of the distinction between these two cases in evident in the relation: $h_i^* = h_i + r_i^2/SSR^+$, where SSR is the Sum of the Squared Residuals. The h_i^* measure leverage in X|y space. The h_i measure leverage in X space. The r_i^2/SSR depend upon X and y, but for moderate h_i they can provide an indication of outliers in y.

Two examples contrast the use of the h_1^* , and the h_1 and r_1^2/SSR . First, consider the data, in (x,y) pairs: (1, .5), (2, 1), (3, 1.5), (.5, 1), (1, 2), (1.5, 3), and (2.49, 3.5) (see Figure 2). Point 7 is clearly an outlier in X|y though not a leverage point in X. We find h_7^* = .609, higher than any other h_1 by .031, so h_7^* reveals the isolation of point 7 in X|y space. This contrasts to h_7 = .419, less than h_3 = .424, and r_7^2/SSR = .190, less than r_6^2/SSR = .300, revealing that point 7 is second in leverage in X, and second in the list of outliers in y (though h_7 is large enough to cause

⁺The author is indebted to Steve Peters for deriving this important relationship.





us to perhaps consider r_7^2/SSR more significant⁺).

As a second example, consider the data: (i, (i/2) + ε_i) for i = 1, 2,...,7 and ε_i is a random variable of uniform distribution in the interval (0, .1); plus the points (4, 25) and (15, 7.5) (see Figure 3). Points 8 and 9 are both outliers in X|y, but point 8 is an outlier in y, not X, and point 9 is a leverage point in X, not an outlier in y. We find h_8^{*} = .999989 and h_9^{*} = .817, followed by h_1^{*} = .268, so the h_1^{*} distinguish points 8 and 9 from the other points, but not from each other. However, h_8 = .122, h_9 = .816, r_8^2/SSR = .878, and r_9^2/SSR = .001. Clearly, the h_1 and r_1^2/SSR distinguish the leverage point in X from the outlier in y.

The above serves as motivation to search for leverage points (or more generally, leverage groups) strictly in the X matrix, using the scaled residuals to identify outliers in y. If hat matrix diagonals are being used to identify leverage points, this approach has the added advantage that the h_i , unlike the h_i^* , are directly computable from the QR decomposition of X - which can be used to solve $X^T X = X^T \beta$.

^{*}See Welsch and Kuh [8] for the possibly more useful statistic , the studentized residual, $r_i^* = r_i/(s_{(i)} (1-h_i)^{1/2})$, where $s_{(i)}$ is the estimated error variance for the "not i" fit.

Appendix 2

H is most reliably computed via the QR decomposition of X [2], which uses Householder transformations (forming orthogonal Q) to reduce X to upper-triangular R. QR decomposition by Householder transformations, with column pivoting, is more stable than Gram-Schmidt orthogonalization, and yeilds a more nearly orthogonal Q than Modified Gram-Schimidt in the event of rank degeneracy.

To compute H, we have $H = X(X^{T}X)^{-1}X^{T}$, X = QR. Therefore, $H = QR(R^{T}Q^{T}QR)^{-1}R^{T}Q^{T} = QQ^{T}$ (Q is m by n here). The QR decomposition routine used need not store Q explicitly, storing instead the u's which define the Householder transformations, I-uu^T (the u's can be stored in a lower triangular matrix). Each h_{i} is computed by applying the Householder transformations to a vector representing the ith column of I_{n} , then setting h_{i} to the dot product of the vector (the first p elements) with itself. The h_{ij} are more cheaply computed (at the price of extra storage) by forming Q explicitly. Appendix 3

DPA FORTRAN

INTEGER NN, NN, N, P, I, J, K, OUT, IN, IPLUS1, IERR, IV1(300), OUTDIS(510) DFA000 INTEGER IV2(300), IV3(300) DOUBLE PRECISION MAX, NNM, SID, DENSE(510), TENP, DFF DFA000 DOUBLE PRECISION NF, NNM, IPT, TI, T2, DIST, EPS, RV1(510) DFA000 DOUBLE PRECISION NF, NNM, IPT, T1, T2, DIST, EPS, RV1(510) DFA000 DOUBLE PRECISION NF, NNM, IPT, T1, T2, DIST, EPS, RV1(510) DFA000 C DATA NH/510/, HN/15/ DFA001 C DATA NH/510/, HN/15/ DFA001 C C DATA NH/510/, HN/15/ DFA001 C C DATA NH/S10/, HN/N, P, X, EPS, SORTOR, OUT, IN) DFA001 DFF = 2,000 X BGRT(DFLOAT(F)) DFA001 C D 20 I=1, P DFA000 T EMF = 0,000 DFA000 D 10 J=1, N DFA001 T EMF = TEMP / TENP + X(J, I) DFA001 D 10 J=1, N DFA001 T EMF = TEMP / DFLOAT(N) DFA002 D 10 J=1, N DFA001 T EMF = TEMP / DFLOAT(N) DFA002 D 10 J=1, N DFA001 D 10 J=1, N DFA002 T F (DABS(X(J,I)).GT. HAX) MAX = DABS(X(J,I)) DFA002 IF (DABS(X(J,I)).GT. HAX) MAX = DABS(X(J,I)) DFA002 D 15 CONTINUE DFA002 D 15 CONTINUE DFA003 C D 10 J=1, N DFA003 D 20 J=1, N DFA003 C D 10 J=1, N DFA003 D 10 J=1, N DFA003 C D 10 J=1, N DFA003 D 10 J=1, N DFA003 C D 10 J=1, N DFA003 D 10 J=1, N DFA003 D 10 J=1, N DFA003 C D 10 J=1, N DFA003 D 10 J=1, N DFA004 D 10 J=1, N DFA004 D 10 J=1, N DFA005 D 10 J=1, N DFA004 D 10 J=1, N DFA005 D 10 J=1	<pre>INTEGER NM, MN, N, P, I, J, K, OUT, IN, IPLUS1, IERR, IV1(300), OUTDIS(51) INTEGER IV2(300), IV3(300) DOUBLE PRECISION X(510, 15), NORMS(510), DENSE(510), TEMP, DFP DOUBLE PRECISION MAX, NRM1, NRM2, DIFF, T1, T2, DIST, EPS, RV1(510) DOUBLE PRECISION DFLOAT, DSQRT, DABS LOGICAL SORTOR C DATA NM/510/, MN/15/ C C CALL MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM, MN, N, P, X, EPS, SORTOR, OUT, IN) DFP = 2,0D0 * DSQRT(DFLOAT(P))</pre>	0) DPA000 DPA000 DPA000 DPA000 DPA000 DPA000 DPA000 DPA000 DPA000	010 020 030 040 050 060 070
<pre>INTEGER 1/2(300)/JUS(300) DPA000 DOUBLE PRECISION X(30)/IS)/NORMS(S10)/DENSE(S10)/TEMP,DFP DPA000 DOUBLE PRECISION X(30/IS)/NORMS(S10)/DENSE(S10)/TEMP,DFP DPA000 DOUBLE PRECISION DFLOAT,DSORT,DABS LOGICAL SORTOR DPA000 DATA NM/S10/,HN/15/ DPA000 DEATA NM/S10/,HN/15/ DPA000 C. CALL MATRIX(AND PARAMETER VALUES. DPA001 DFF 2.0D0 * DSGRT(DFLOAT(F)) DFA000 DFF 2.0D0 * DSGRT(DFLOAT(F)) DFA001 DFF 2.0D0 * DSGRT(DFLOAT(F)) DFA001 DFF 2.0D0 * DSGRT(DFLOAT(F)) DFA001 DFF 2.0D0 * DSGRT(DFLOAT(F)) DFA001 D 0 20 [=1-P TEMP = 0.0D0 D 10 J=1+N TEMP = TEMP / X(J,I) DFA001 D 10 J=1+N TEMP = TEMP / X(J,I) DFA001 D 15 J=1,N MAX = 0.0D0 D 15 J=1,N C II::INORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DFA002 CI::I::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DFA002 DFA002 DI 15 J=1,N X(J,I) = X(J,I) / MAX) / DFP DFA003 DFA003 DFA004 DFA005 C II::INORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DFA002 CI::I:INORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DFA002 DFA002 DFA002 D 30 I=1,N DFA003 DO 130 I=1,N DFA003 DFA004 DFA005 DFA0</pre>	<pre>INTEGER IV2(300),IV3(300) DOUBLE PRECISION X(510,15),NORMS(510),DENSE(510),TEMP,DFP DOUBLE PRECISION MAX,NRM1,NRM2,DIFF,T1,T2,DIST,EPS,RV1(510) DOUBLE PRECISION DFLOAT,DSQRT,DABS LOGICAL SORTOR C DATA NM/510/,MN/15/ C C CALL MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP = 2,0D0 * DSQRT(DFLOAT(P))</pre>	DPA000 DPA000 DPA000 DPA000 DPA000 DPA000 DPA000 DPA000	020 030 040 050 060 070
DUBLE PRECISION X(510,15),NORMS(510),DEMSE(510),TEMP,DFP DPA000 DUBLE PRECISION MAX,NRM1,NRM2,DIF,T1,T2,DIST,EPS,RV1(510) DPA000 DUBLE SURTOR DFLOAT,DSORT,DABS DFA000 DATA NH/510/,MN/15/ DFA000 C.DATA NH/510/,MN/15/ DFA000 C.DATA NH/510/,MN/15/ DFA000 DFA000 DFA000 DFA000 DFA000 DFA000 DFP = 2,000 * DSGRT(DFLOAT(P)) DFA001 C.TITICENTER THE DATA. DFARAMETER VALUES. DFA001 DFP = 2,000 * DSGRT(DFLOAT(P)) DFA001 C.TEMP = 0,000 DFA000 DI 10 _=1:P DFLOAT(N) DFA001 TEMP = TEMP + X(J,I) DFA001 DFA000 DFA000 DFA000 DFA000 DFA000 DFA000 DFA000 DFA001 DFA000 DFA000 DFA001 DFA000 DFA000 DFA001 DFA001 DFA001 DFA001 DFA001 DFA001 DFA002 TEMP = TEMP + X(J,I) DFA002 I CONTINUE DFLOAT(N) MAX = DABS(X(J,I)) DFA002 I G CONTINUE DFA002 I G 10 _=1:N DFLOAT(N) MAX = DABS(X(J,I)) DFA002 I G 10 _=1:N DFLOAT(N) MAX = DABS(X(J,I)) DFA002 I G 10 _=1:N DFLOAT(N) MAX = DABS(X(J,I)) DFA002 I G 0 _=1:N DFLOAT(N) MAX = DABS(X(J,I)) DFA002 I G 0 _=1:N DFLOAT(N) MAX = DABS(X(J,I)) DFA002 I G 0 _=1:N DFLOAT(N) DFA002 I DFA003 DC 0 _=1:N DFLOAT(N) DFA003 C _= DFLOAT DFLOAT(N) DFA003 D 0 _=1:N DFLOAT DFLOAT(N) DFA003 I DFLOAT DFA002 I D 10 _=1:N DFLOAT DFLOAT DFPA03 D 0 _=1:N DFLOAT DFPA03 D 0 _=1:N DFLOAT DFPA03 DFA03 D 0 _=1:N DFLOAT DFPA03 DFA03	DOUBLE PRECISION X(510,15),NORMS(510),DENSE(510),TEMP,DFP DOUBLE PRECISION MAX,NRM1,NRM2,DIFF,T1,T2,DIST,EPS,RV1(510) DOUBLE PRECISION DFLOAT,DSQRT,DABS LOGICAL SORTOR C DATA NM/510/,MN/15/ C C:::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DFA00(DFA00(DFA00(DFA00(DFA00(DFA00(DFA00(030 040 050 060 070
DUBLE PRECISION MAX:NRH:NRM2,DIFF.T1,T2,DIST.FPS;RV1(510) DPA000 DUBLE PRECISION DFLOAT,DSORT,DABS DPA000 LGGICAL SORTOR DPA000 DATA NH/510/,HN/15/ DPA000 C C C C C C C C C C C C C	DOUBLE PRECISION MAX;NRM1;NRM2;DIFF;T1;T2;DIST;EPS;RV1(510) DOUBLE PRECISION DFLOAT;DSQRT;DABS LOGICAL SORTOR C DATA NM/510/;MN/15/ C C:::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM;MN;N;P;X;EPS;SORTOR;OUT;IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DFA00(DFA00(DFA00(DFA00(DFA00(DFA00(DFA00(040 050 060 070
DUBLE PRECISION DFLOAT, DSGRT, DABS LOGICAL SORTOR PAA000 DATA NH/SIO/, MN/15/ C DATA NH/SIO/, MN/15/ C DATA NH/SIO/, MN/15/ C C DATA NH/SIO/, MN/15/ C C DATA NH/SIO/, MN/15/ C C DATA NH/SIO/, MN/15/ C C C DATA MATRIX AND PARAMETER VALUES. DFA000 DFP = 2.000 * DSGRT(DFLOAT(P)) D D D D D D D D D D D D D	DOUBLE PRECISION DFLOAT,DSQRT,DABS LOGICAL SORTOR C DATA NM/510/,MN/15/ C C:::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DPA00(DPA00(DPA00(DPA00(DPA00(050 060 070
LUGICAL SURTOR PPAGOO DATA NM/510/, HN/15/ DPAGOO C::::::GET DATA MATRIX AND PARAMETER VALUES. DPAGOO C::::::GET DATA MATRIX AND PARAMETER VALUES. DPAGOO CALL MATRIX(NM, NM, N, P, X, EPS, SORTOR, OUT, IN) DPAGOI DFP = 2.000 * DSGRT(DFLOAT(P)) DPAGOI DO 20 I=1,P BPAGOI TEMP = TEMP + X(J, I) PAGOI DO 10 J=1.N BPAGOI TEMP = TEMP / DFLOAT(N) BPAGOI MAX = 0.000 BPAGOI DF 15 J=1.N BPAGOI IF (DASS(X(J,I)) .GT. MAX) MAX = DASS(X(J,I)) DPAGOI DFFAGOI 15 CONTINUE DFAGONAL LENGTH 1. DPAGOI C:::::INORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPAGOI C:::::INORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPAGOI DFGGOI 20 CONTINUE DF MAIN DIAGONAL LENGTH 1. DPAGOI DO 20 J=1.N BPAGOI DO 30 I=1.N DPAGOI DO 30 I=1.N DPAGOI DC 30 I=1.N DPAGOI DC 30 I=1.N DPAGOI DC 30 I=1.N DPAGOI DFFAGO	C DATA NM/510/,MN/15/ C C:::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DPA00(DPA00(DPA00(DPA00(060 070
DATA NH/510/, HN/15/ DATA NH/510/, HN/15/ C DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NH, HN, N, P, X, EPS, SORTOR, OUT, IN) DFP = 2.0D0 * DSGRT(DFLOAT(P)) C C CALL MATRIX(NH, HN, N, P, X, EPS, SORTOR, OUT, IN) DFP = 2.0D0 * DSGRT(DFLOAT(P)) C C D D D D D D D D D D D D D	DATA NM/510/;MN/15/ C C:::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM;MN;N;P;X;EPS;SORTOR;OUT;IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DPA00(DPA00(DPA00(070
DHAR MUSIC/TMAINS C: C: C: CALL MATRIX(NH,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP 2001 DFP 2.0D0 * DSQRT(DFLOAT(P)) C: C: CALL MATRIX(NH,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP 001 DFP 002 TEMP = TEMP + X(J,I) 10 CONTINUE C: C: C: C: C: C: C: C: C: C:	C C::::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM;MN;N;P;X;EPS;SORTOR;OUT;IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DPA000 DPA000	
L CLITTIGET DATA MATRIX AND PARAMETER VALUES. CLITTIGET DATA MATRIX AND PARAMETER VALUES. CALL MATRIX(NH,HN,H,F,X,EPS,SORTOR,OUT,IN) DFA001 DFF = 2.0D0 * DSGRT(DFLOAT(P)) CLITTICENTER THE DATA. C DG 20 I=1,F DG 20 I=1,F DG 10 J=1,N TEMP = 0.0D0 DFA001 DFA001 DFA001 DFA001 DFA001 DFA001 DFA002 DFA003 DFA002 DFA003 DFA003 DFA003 DFA004 DFA003 DFA004 DFA003 DFA003 DFA003 DFA004 DFA003 DFA004 DFA003 DFA004 DFA003 DFA004 DFA004 DFA004 DFA005 DFA004 DFA004 DFA004 DFA005 DFA004 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005	C:::::GET DATA MATRIX AND PARAMETER VALUES. C CALL MATRIX(NM;MN;N;P;X;EPS;SORTOR;OUT;IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	DPA00(280
C C CALL MATRIX(NM,NM,P,YX,PFS,SORTOR,OUT,IN) DFP = 2.0D0 X BSORT(DFLOAT(P)) C C DD 20 I=1+P DD 20 I=1+P TEMP = 0.0D0 D 0 J =1+N TEMP = TEMP / X(J,I) D C CONTINUE C C CONTINUE C C CONTINUE C C CONTINUE C C CONTINUE C D 20 J=1+N TF (DABS(X(J,I)).GT. MAX) MAX = DABS(X(J,I)) D FA002 D 15 J=1+N X(J,I) = X(J,I) - TEMP X(J,I) = X(J,I).GT. MAX) MAX = DABS(X(J,I)) D FA002 C C CONTINUE C C CONTINUE C C CONTINUE C D 20 J=1+N X(J,I) = (X(J,I) / MAX) / DFP 20 CONTINUE C C CONTINUE C C CONTINUE C D 20 J=1+N X(J,I) = (X(J,I) / MAX) / DFP 20 CONTINUE C C CONTINUE C C CONTINUE C C CONTINUE C C CONTINUE C D 20 J=1+N D 4002 D 30 I=1+N D 4002 D 30 I=1+N D 4002 C C C CONTINUE C C C C CONTINUE C C C C CONTINUE C C C CONTINUE C C C CONTINUE C C C CONTINUE C C C C C C C C C C C C C	C C CALL MATRIX(NM;MN;N;P;X;EPS;SORTOR;OUT;IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))	<u> </u>	1 090
C CALL MATRIX(NM, NN, P, Y, EPS, SORTOR, OUT, IN) DFF = 2.010 * DSGRT(DFLOAT(P)) C::::::CENTER THE DATA. DO 20 I=1.P DO 20 I=1.P TEMP = 0.0D0 DI 10 J=1.N TEMP = TEMP + X(J,I) DFA001 DFA002 DFA003 DFA004	CALL MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN) DFP = 2.0D0 * DSQRT(DFLOAT(P))		110
DIFF = 2.000 * DSGR(DFLGAT(P)) DFA001 C DFF = 2.000 * DSGR(DFLGAT(P)) DFA001 C D 20 I=1+F DFA001 D 0 0 J=1+N TEMP = 0.0D0 D 10 J=1+N TEMP = TEMP + X(J,I) DFA002 DFA003 DFA004 DFA004 DFA003 DFA005 DFA005 DFA004 DFA004 DFA005 DFA005 DFA004 DFA004 DFA005 DFA005 DFA004 DFA005 DFA005 DFA005 DFA005 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA005 DFA005 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA005 DFA006 DFA006 DFA005 DFA006 DF	$DFP = 2.0D0 \times DSQRT(DFIDAT(P))$	DFAUU.	1 20
C LIT LIGENTER THE DATA. C:::::CENTER THE DATA. D 20 I=1;F TEMP = 0.0D0 D 10 J=1;N TEMP = 0.0D0 D 20 I=1;F TEMP = TEMP + X(J,I) D CONTINUE TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 D 15 J=1;N IF (DABS(X(J,I)) - TEMP IF (DABS(X(J,I)) .GT. MAX) MAX = DABS(X(J,I)) D 20 IS J=1;N C CONTINUE C CONTINUE C CONTINUE D 20 J=1;N D 20 CONTINUE D 30 CONTINUE C C C CONTINUE D 30 CONTINUE D 30 CONTINUE D 40 J=1;P D 4003 D 40 J=1;P D 4003 D 40 CONTINUE C CONTINUE D 50 CONTINUE C D 50 I=1;N TEMP = TEMP + X(I;J)*X(I;J) D 10 105 I=1;N D 100 105 I=1;N D 100 105 I=1;N D 100 I05 I=1;N D		DPACO	170
C:::::COMPUTE R THE DATA. C DD 20 I=1,P DD 4001 TEMP = 0.0D0 DD 10 J=1,N TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 DD 15 J=1,N TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 DD 15 J=1,N TF (DABS(X(J,I)).GT. MAX) MAX = DABS(X(J,I)) DPA002 C C C::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA002 C:::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1. DD 20 J=1,N T(J,I) = (X(J,I) / MAX) / DFP DFA003 DD 30 I=1,N DD 30 I=1,N DENSE(I) = 0.0D0 DUTDIS(I) = 0 S0 CONTINUE C C C C C C C C C C C C C C C C C C C	C	DPA00	140
C DP 20 I=1;P DP 4001 TEMP = 0.0D0 DP 4001 D0 10 J=1;N DP 4001 TEMP = TEMP + X(J;I) DP 4002 10 CONTINUE DF 4002 TEMP = TEMP / DFLOAT(N) DP 4002 TEMP = TEMP / DFLOAT(N) DP 4002 D0 15 J=1;N DP 4002 X(J;I) = X(J;I) - TEMP DP 4002 X(J;I) = X(J;I) - TEMP DP 4002 C C C C C C C C C D0 20 J=1;N DP 4002 D0 20 J=1;N DP 4002 C D0 20 J=1;N DP 4002 D0 20 J=1;N DP 4002 C C D0 20 J=1;N DP 4002 D0 20 J=1;N DP 4002 C C D0 20 J=1;N DP 4002 D0 30 I=1;N DP 4002 C C C C D0 20 J=1;N DP 4002 D0 30 I=1;N DP 4002 C C C C D0 20 J=1;N DP 4003 D 20 CONTINUE DP 4003 C C D 20 J=1;N DP 4003 C C D 20 J=1;N DP 4003 C C D 20 J=1;N DP 4003 D 30 CONTINUE DP 4003 C C C C C C C C C C C C C C C	C:::::CENTER THE DATA.	DPA00	150
D0 20 I=1.P TEMP = 0.0D0 D0 10 J=1.N TEMP = TEMP + X(J,I) 10 CONTINUE TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 D0 15 J=1.N X(J,I) = X(J,I) - TEMP IPA002 IF (DABS(X(J,I)) .GT. MAX) MAX = DABS(X(J,I)) DPA002 C C C C D0 20 J=1.N C D0 30 I=1.N DENSE(I) = 0.0D0 DTDIS(I) = 0 30 CONTINUE C C C C C C C C D0 50 I=1.N C C D0 50 I=1.N C C D0 50 I=1.N C D0 50 D I=1.N C D0 50 I=1.N C D0 50 I=1.N C C D0 50 I=1.N C D0 50 I=1.N C D0 50 I=1.N C D0 50 I=1.N C D0 50 I=1.N C D0 50 I=1.N DPA003 DD 105 I=1.N DPA004 DD 40 J=1.P C D0 105 I=1.N DPA005 DD 105 I=1.N DPA004 DPA004 DPA004 DPA005 DPA004 DPA005 DPA004 DPA004 DPA005 DPA004 DPA004 DPA005 DPA004 DPA004 DPA005 DPA004 DPA004 DPA005 DPA005 DPA004 DPA005 DPA004 DPA004 DPA005 DPA004 DPA004 DPA005 DPA005 DPA004 DPA005 DPA05 DPA		DFA00	160
TEMP = 0.0D0 DPA001 D0 10 J=1;N DPA002 10 CONTINUE DPA002 10 CONTINUE DPA002 MAX = 0.0D0 DPA002 MAX = 0.0D0 DPA002 D0 15 J=1;N DPA002 X(J;I) = X(J;I) - TEMP DPA002 IF (DABS(X(J;I)) .GT. MAX) MAX = DABS(X(J,I)) DPA002 DFA002 IF (DABS(X(J,I)) .GT. MAX) MAX = DABS(X(J,I)) DPA002 C::::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA002 DPA003 C D0 20 J=1;N DPA002 X(J;I) = (X(J;I) / MAX) / DFP DPA003 D0 30 I=1;N DPA003 D0 30 I=1;N DPA003 D0 30 I=1;N DPA003 D0 30 CONTINUE DPA003 C DPA003 C DPA004 D0 30 I=1;N DPA005 D0 50 I=1;N DPA004 D0 40 J=1;P DPA004 D0 40 J=1;P DPA004 D0 40 J=1;P DPA004 D0 50 CONTINUE DPA004 D0 40 J=1;P DPA004 D0 40 J=1;P DPA004 <tr< td=""><td>DO 20 I=1,P</td><td>DPA00:</td><td>170</td></tr<>	DO 20 I=1,P	DPA00:	170
D0 10 J=1;N TEMP = TEMP + X(J,I) D CONTINUE TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 D0 15 J=1;N X(J,I) = X(J,I) - TEMP IF (DABS(X(J,I)) .GT. MAX) MAX = DABS(X(J,I)) DPA002 C C C D0 20 J=1;N X(J,I) = (X(J,I) / MAX) / DFP D0 20 J=1;N X(J,I) = (X(J,I) / MAX) / DFP D0 20 J=1;N X(J,I) = (X(J,I) / MAX) / DFP D0 30 I=1;N DD 20 J=1;N DD 20 J=1;N DD 20 J=1;N DD 20 J=1;N DD 20 J=1;N DPA003 DC D0 30 I=1;N DFA004 DD 30 I=1;N DFA005 DD 20 J=1;N DFA005 DD 20 J=1;N DFA004 DFA005 DD 20 J=1;N DFA005 DFA05 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA005 DFA05 DF	TEMP = 0.000	DPA00:	180
TEMP = TEMP + X(J,I) DPA002 10 CONTINUE DPA002 TEMP = TEMP / DFLOAT(N) DPA002 MAX = 0.0D0 DPA002 D0 15.J=1,N DPA002 X(J,I) = X(J,I) - TEMP DPA002 IF (DABS(X(J,I)).GT. MAX) MAX = DABS(X(J,I)) DPA002 C DPA002 C:::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA002 C::::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA003 C D0 20 J=1,N X(J,I) = (X(J,I) / MAX) / DFP DPA003 D0 30 I=1,N DPA003 D0 30 CONTINUE DPA003 D0 30 CONTINUE DPA003 D0 30 CONTINUE DPA004 D0 30 I=1,N DPA003 C DPA004 D0 30 I=1,N DPA005 C DPA004 D0 40 J=1,P DPA004 D0 50 I=1,N DPA004 D0 40 J=1,P DPA004 D0 40 J=1,P DPA004 D0 50 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 <td>DO 10 J=1,N</td> <td>DFA00:</td> <td>190</td>	DO 10 J=1,N	DFA00:	190
10 CONTINUE TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 D0 15 J=1,N (J,I) = X(J,I) - TEMP DFA002 X(J,I) = X(J,I) - TEMP DFA002 C C C C C C D0 20 J=1,N C D0 20 J=1,N X(J,I) = (X(J,I) / MAX) / DFP D0 20 J=1,N X(J,I) = (X(J,I) / MAX) / DFP DFA003 D0 30 I=1,N DENSE(I) = 0.0D0 OUTDIS(I) = 0 30 CONTINUE DD 50 I=1,N DD 50 DFA003 DD 7004 DD	$TEMP = TEMP + X(J_{F}I)$	DPA002	200
TEMP = TEMP / DFLOAT(N) MAX = 0.0D0 DO 15 J=1:N X(J,I) = X(J,I) - TEMP IF (DABS(X(J,I)).GT. MAX) MAX = DABS(X(J,I)) DFA002 C C C C C C C C C C C C C	10 CONTINUE	DFA002	210
MAX = 0.0D0 DPA002 D0 15 J=1;N DPA002 IF (DABS(X(J,I)) .GT. MAX) MAX = DABS(X(J,I)) DPA002 D1 15 CONTINUE DPA002 C DPA002 C::::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA002 DPA002 C::::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA002 DPA003 C::::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1. DPA003 D0 20 J=1;N DPA003 20 CONTINUE DPA003 D0 30 I=1;N DPA003 D0 30 I=1;N DPA003 D0 30 I=1;N DPA003 00 30 CONTINUE DPA003 00 30 CONTINUE DPA004 C DPA004 C DPA004 C DPA005 C DPA004 D0 50 I=1;N DPA004 TEMP = TEMP + X(I,J)*X(I,J) DPA004 D0 40 J=1;P DPA004 00 40 J=1;P DPA004 01 05 I=1;N DPA004 10 105 I=1;N DPA005 10 105 I=1;N DPA005 D0 105 I=1;N DPA005	TEMP = TEMP / DFLOAT(N)	DPA002	220
<pre>DPA002 X(J,I) = X(J,I) - TEMP IF (DABS(X(J,I)).GT. MAX) MAX = DABS(X(J,I)) DPA002 C C C C C C C C C C C C C C C C C C</pre>	MAX = 0.0D0	DPA00	230
<pre>X(J,I) = X(J,I) - TEMP IF (DABS(X(J,I)) .GT. MAX) MAX = DABS(X(J,I)) DPA002 CC CC CC CC CC CC CC CC CC C</pre>	$\square 15 J=1,N$	DPA002	240
TF (DABS(X(J,T)) .GT. MAX) MAX = DABS(X(J,T)) 15 CONTINUE C C:::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DFA002 C:::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1. DD 20 J=1;N X(J;I) = (X(J;I) / MAX) / DFP DD 20 CONTINUE DD 30 I=1;N DENSE(I) = 0.0D0 DD 10 I=1;N C C C:::::COMPUTE ROW L2 NORMS. C DD 50 I=1;N TEMP = TEMP + X(I;J)*X(I;J) 40 CONTINUE DD 40 J=1;P TEMP = TEMP + X(I;J)*X(I;J) DD 40 J=1;N DFA004 DD 40 J=1;P TEMP = TEMP + X(I;J)*X(I;J) DD 105 I=1;N DD 105 I=1;N IF (I : EQ. N) GOTO 105 IF (I : EQ. N) GOTO 105 IF (I : EQ. N) GOTO 105 IF (I) = I + 1 NRM1 = NORMS(I) DD 100 J=TEUSIN NCM1 = NORMS(I) DD 100 J=TEUSIN DD 100 J=TEUSIN DD 100 J=TEUSIN NCM1 = NORMS(I) DD 100 J=TEUSIN DD 100 J	$\mathbf{X}(\mathbf{J}_{\mathbf{F}}\mathbf{I}) = \mathbf{X}(\mathbf{J}_{\mathbf{F}}\mathbf{I}) - \mathbf{T}\mathbf{E}\mathbf{M}\mathbf{P}$	DPA00	250
15 CUNTINUE DPA002 CC CC C::::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA002 C:::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1. DPA003 C DO 20 J=1,N X(J,I) = (X(J,I) / MAX) / DFP DPA003 DO 30 I=1,N DENSE(I) = 0.0D0 DPA003 DUTDIS(I) = 0 DPA003 30 CONTINUE DPA003 C C C C C C C D D D D D D D D D D D D D	$IF (DABS(X(J_{F}I)) \cdot GT \cdot MAX) MAX = DABS(X(J_{F}I))$	DPA00	260
L DPA002 C:::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO DPA003 C:::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1. DD 20 J=1,N X(J,I) = (X(J,I) / MAX) / DFP DD 30 I=1,N DD 30 I=1,N DD 30 I=1,N DD 30 I=1,N DFA003 30 CONTINUE C C C C DD 50 I=1,N DD 50 I=1,P TEMP = 0.0D0 DD 40 J=1,P TEMP = TEMP + X(I,J)*X(I,J) 40 CONTINUE DD 50 I=1,N TEMP = TEMP + X(I,J)*X(I,J) DFA004 DFA004 DFA004 DFA005 DFA005 DFA004 DFA005 DFA0	15 CUNTINUE	DPA002	270
C:::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1. DD 20 J=1,N X(J,I) = (X(J,I) / MAX) / DFP 20 CONTINUE DD 30 I=1,N DENSE(I) = 0.0D0 OUTDIS(I) = 0 30 CONTINUE C C C DD 50 I=1,N DFA003 C C C DD 50 I=1,N DFA004 DD 50 I=1,N DFA004 DD 40 J=1,P TEMP = TEMP + X(I,J)*X(I,J) 40 CONTINUE C C DD 105 I=1,N DFA004 DFA005	U CTTTTTNORMALIZE THE DATA CHON THAT THE ODCEDUATION CDACE TO COALED.	LIPAUU	280
C C DD 20 J=1;N DPA003 C DD 30 J=1;N DPA003 DD 30 J=1;N DPA003 DD 30 J=1;N DPA003 DD 30 CONTINUE DPA003 30 CONTINUE C C C C C C C C C C C C C C C C C C C	CITITIAN AND EDGINE OF MAIN DIAGONAL LENGTH 1	INTO DEADO	270
D0 20 J=1,N DFA003 X(J,I) = (X(J,I) / MAX) / DFP DFA003 20 CONTINUE DFA003 D0 30 I=1,N DFA003 D0 S0 I=1,N DFA003 0UTDIS(I) = 0 DFA003 30 CONTINUE DFA003 C DFA003 C DFA004 D 40 J=1,P DFA004 TEMP = TEMP + X(I,J)*X(I,J) DFA004 V DFA004 NORMS(I) = DSQRT(TEMP) DFA004 S0 CONTINUE DFA004 DFA005 DFA004 MORMS(I) = DSQRT(TEMP) DFA005 D0 105 I=1,N DFA005 IF (I.EQ. N) GOTO 105 DFA005 IFLUS1 = I + 1 DFA005 NRM1 = NORMS(I) DFA005 </td <td>CTTTTT A ALLERCODE OF AMIN DIMODAME LENGIA IT</td> <td>DF HOU.</td> <td>300</td>	CTTTTT A ALLERCODE OF AMIN DIMODAME LENGIA IT	DF HOU.	300
X(J,I) = (X(J,I) / MAX) / DFP DPA003 20 CONTINUE DPA003 D0 30 I=1;N DPA003 DENSE(I) = 0.0D0 DPA003 0UTDIS(I) = 0 DPA003 30 CONTINUE DPA003 C DPA003 C DPA004 C DPA004 D0 50 I=1;N DPA004 D0 40 J=1;P DPA004 D0 40 J=1;P DPA004 D0 40 J=1;P DPA004 MORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 MORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 MORMS(I) = DSQRT(TEMP) DPA004 S0 CONTINUE DPA004 MORMS(I) = DSQRT(TEMP) DPA004 S0 CONTINUE DPA005 ID 105 I=1;N DPA005 IF (I :EQ, N) GOTO 105 DPA005 IFLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 DD 100 LETUS1:N DPA005	τΩ 20 l≡1•N	DP 400.	320
20 CONTINUE DFA003 D0 30 I=1,N DFA003 DENSE(I) = 0.0D0 DFA003 0UTDIS(I) = 0 DFA003 30 CONTINUE DFA003 C: DFA003 C: DFA003 C: DFA003 C DFA004 D0 50 I=1,N DFA004 C DFA004 D0 40 J=1,P DFA004 TEMP = TEMP + X(I,J)*X(I,J) DFA004 40 CONTINUE DFA004 NORMS(I) = DSQRT(TEMP) DFA004 50 CONTINUE DFA004 C DFA004 D1 105 I=1,N DFA004 C DFA004 MORMS(I) = DSQRT(TEMP) DFA004 D0 105 I=1,N DFA005 IF (I .EQ. N) GOTO 105 DFA005 IFLUS1 = I + 1 DFA005 NRM1 = NORMS(I) DFA005 D0 100 I=IFLUS1.N DFA005	X(J,I) = (X(J,I) / MAX) / DFP	TIPAOO	330
DO 30 I=1,N DPA003 DENSE(I) = 0.0D0 DPA003 OUTDIS(I) = 0 DPA003 30 CONTINUE DPA003 C::::::COMPUTE ROW L2 NORMS. DPA004 DD 50 I=1,N DPA004 DD 50 I=1,N DPA004 DD 40 J=1,P DPA004 TEMP = TEMP + X(I,J)*X(I,J) DPA004 40 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 C::::COMPUTE DISTANCES SQUARED. DPA005 DD 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IFLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005	20 CONTINUE	DPA00	340
DENSE(I) = 0.0D0 OUTDIS(I) = 0 30 CONTINUE C C::::::COMPUTE ROW L2 NORMS. C D0 50 I=1,N D0 40 J=1,P TEMP = TEMP + X(I,J)*X(I,J) 40 CONTINUE NORMS(I) = DSQRT(TEMP) 50 CONTINUE D0 105 I=1,N D0 105 I=1,	DO 30 I=1,N	DPA00	350
OUTDIS(I) = 0 DPA003 30 CONTINUE DPA003 C DPA003 C:::::COMPUTE ROW L2 NORMS. DPA004 D0 50 I=1,N DPA004 D0 50 J=1,N DPA004 D0 40 J=1,P DPA004 V TEMP = TEMP + X(I,J)*X(I,J) DPA004 40 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 V:::COMPUTE DISTANCES SQUARED. DPA005 D0 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IFLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 D0 100 L=IELUS1.N DPA005	DENSE(I) = 0.0D0	DPA00	360
30 CONTINUE DPA003 C DFA003 C::::::COMPUTE ROW L2 NORMS. DPA004 D0 50 I=1,N DPA004 TEMP = 0.0D0 DPA004 D0 40 J=1,P DPA004 TEMP = TEMP + X(I,J)*X(I,J) DPA004 40 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 C DPA004 MORMS(I) = DSQRT(TEMP) DPA004 D0 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IPLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 D0 100 LETPLUS1.N DPA005	OUTDIS(I) = 0	DPA00	370
C C C::::::COMPUTE ROW L2 NORMS. D D D O S O I = 1, N D D O O O O O O O O O O O O O O O O O	30 CONTINUE	DPA00	380
C:::::COMPUTE ROW L2 NORMS. D0 50 I=1+N TEMP = 0.0D0 D0 40 J=1+P TEMP = TEMP + X(I+J)*X(I+J) 40 CONTINUE NORMS(I) = DSQRT(TEMP) 50 CONTINUE C :::COMPUTE DISTANCES SQUARED. D0 105 I=1+N IF (I +EQ. N) GOTO 105 IPLUS1 = I + 1 NRM1 = NORMS(I) D0 40 J=1+N DFA004 DFA004 DFA005 DFA005 DFA004 DFA005 DFA05 DFA05 DFA05 DFA05 DFA05 DFA05 DFA05 DFA05 DFA05 DFA05		DF'AOO	390
C D0 50 I=1,N TEMP = 0.0D0 D0 40 J=1,P TEMP = TEMP + X(I,J)*X(I,J) 40 CONTINUE NORMS(I) = DSQRT(TEMP) 50 CONTINUE C C C D1 105 I=1,N IF (I .EQ. N) GOTO 105 IF (U .EQ. N) GOTO 105 IPLUS1 = I + 1 NRM1 = NORMS(I) D0 100 J=IFUS1.N DPA005 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DPA05 DP	C:::::COMPUTE ROW L2 NORMS.	DF AOO	400
DU 50 I=1;N TEMP = 0.0D0 DU 40 J=1;P TEMP = TEMP + X(I,J)*X(I,J) 40 CONTINUE NORMS(I) = DSQRT(TEMP) 50 CONTINUE C C DD 105 I=1;N IF (I .EQ. N) GOTO 105 IF (I .EQ. N) GOTO 105 IPLUS1 = I + 1 NRM1 = NORMS(I) DO 100 J=IPLUS1;N DD 400 J=IPLUS1;N DD		DPA004	410
TEMP = 0.000 DPA004 DD 40 J=1,P DPA004 TEMP = TEMP + X(I,J)*X(I,J) DPA004 40 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 C DPA004 D0 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IPLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 D0 100 J=TPLUS1-N DPA005	DO 50 I=1 N	DPA004	420
DU 40 J=1,P DPA004 TEMP = TEMP + X(I,J)*X(I,J) DPA004 40 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA004 C DPA004 C DPA005 D0 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IPLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 D0 100 .I=TELUS1.N DPA005	IEMP = 0.000	DPA004	430
40 CONTINUE DPA004 40 CONTINUE DPA004 NORMS(I) = DSQRT(TEMP) DPA004 50 CONTINUE DPA005 50 ISTANCES SQUARED. DPA005 50 105 IF (I .EQ. N) GOTO 105 DPA005 50 IF (I .EQ. N) GOTO 105 DPA005 DPA005 50 IPLUS1 = I + 1 DPA005 DPA005	UU 40 J=1≠P TEMD - TEMD I V/T D+V/T D	DPA004	440
40 CONTINUE DFA004 NORMS(I) = DSQRT(TEMP) DFA004 50 CONTINUE DFA004 C DFA004 D DSTANCES SQUARED. DFA005 D0 105 I=1+N DFA005 IF (I .EQ. N) GOTO 105 DFA005 IFLUS1 = I + 1 DFA005 NRM1 = NORMS(I) DFA005 D0 100 I=IFLUS1+N	IENF = IENF + X(I)J(I)	DPA004	430
50 CONTINUE DFA004 C DFA004 D::::COMPUTE DISTANCES SQUARED. DFA005 D0 105 I=1;N DFA005 IF (I .EQ. N) GOTO 105 DFA005 IFLUS1 = I + 1 DFA005 NRM1 = NORMS(I) DFA005 D0 100 .I=IFLUS1:N DFA005	NORMS(I) = DCORT(TEMP)		400
C C C C C C C C C C C C C C	50 CONTINUE	DF HOO- DEACO.	4/0
Image: Structure Distances Squared. DPA005 D0 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IFLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 D0 100	C	DPA00	490
DO 105 I=1;N DFA005 IF (I .EQ. N) GOTO 105 DFA005 IFLUS1 = I + 1 DFA005 NRM1 = NORMS(I) DFA005 DD 100	CALL::COMPUTE DISTANCES SQUARED.	DPA00	500
DO 105 I=1,N DPA005 IF (I .EQ. N) GOTO 105 DPA005 IPLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 DO 100		DFA00	510
IF (I .EQ. N) GOTO 105 DPA005 IPLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 DD 100 J=IPLUS1.N DPA005	DO 105 I=1,N	DPA00	520
IPLUS1 = I + 1 DPA005 NRM1 = NORMS(I) DPA005 D0 100 J=IPLUS1-N DPA005	IF (I .EQ. N) GOTO 105	DPA00	530
NRM1 = NORMS(I) DPA005	IPLUS1 = I + 1	DPA00	540
		DPA00	550
DPAVU3	NKM1 = NUKMS(1)		540
DIST = 0.000 pp.404	NKMI = NUKMS(I) DO 100 J=IFLUS1;N	DPA00	200

```
DO 70 K=1,P
                                                                             DPA00580
                     DIFF = X(I_{F}K) - X(J_{F}K)
                                                                             DPA00590
                     DIST = DIST + DIFF*DIFF
                                                                             DPA00600
    70
               CONTINUE
                                                                             DPA00610
                                                                             DPA00620
   ::::COMPUTE NORMAL (PARALLEL) DISTANCES.
                                                                             DPA00630
 С
                                                                             DPA00640
    75
               NRM2 = NORMS(J)
                                                                             DPA00650
               T1 = (DIST + NRM1*NRM1 - NRM2*NRM2) / (2.0D0*NRM1)
                                                                             DF'A00660
               T2 = (DIST + NRM2*NRM2 - NRM1*NRM1) / (2.0D0*NRM2)
                                                                             DPA00670
               DENSE(I) = DENSE(I) + 1.0D0 / (EPS + DABS(T1))
                                                                             DPA00680
               DENSE(J) = DENSE(J) + 1.0D0 / (EPS + DABS(T2))
                                                                             DPA00690
С
                                                                             DPA00700
C:::::TALLY OUTDISTANCING POINTS.
                                                                             DPA00710
С
                                                                             DPA00720
               IF (T1 .LE. 0.0D0) OUTDIS(I) = OUTDIS(I) + 1
                                                                             DPA00730
               IF (T2 .LE. 0.0D0) OUTDIS(J) = OUTDIS(J) + 1
                                                                             DPA00740
   100
          CONTINUE
                                                                             DPA00750
   105 CONTINUE
                                                                             DPA00760
       WRITE(OUT,1001)
                                                                             DPA00770
       DO 110 I=1,N
                                                                             DPA00780
          WRITE(OUT,1002) I,NORMS(I),DENSE(I),OUTDIS(I)
                                                                             DFA00790
  110 CONTINUE
                                                                             DPA00800
С
                                                                             DPA00810
C:::::CHECK INDIVIDUAL POINTS OF INTEREST.
                                                                             DPA00820
С
                                                                             DPA00830
  120 WRITE(OUT,1003)
                                                                             DPA00840
С
                                                                             DPA00850
C:::::GET FOINT INDEX.
                                                                             DPA00860
С
                                                                             DPA00870
       READ(IN,1004) K
                                                                             DPA00880
       IF (K*(2*N + 1 - 2*K)) 130,200,150
                                                                             DPA00890
   130 WRITE(OUT,1006) N
                                                                             DPA00900
       GO TO 120
                                                                             DPA00910
С
                                                                             DPA00920
C:::::COMPUTE DISTANCES.
                                                                             DPA00930
С
                                                                             DPA00940
  150 NRM1 = NORMS(K)
                                                                             DPA00950
      DENSE(K) = 0.0D0
                                                                             DPA00960
      RV1(K) = 0.000
                                                                             DPA00970
      DO 170 I=1,N
                                                                             DPA00980
          OUTDIS(I) = I
                                                                             DPA00990
          IF (I .EQ. K) GO TO 170
                                                                             DPA01000
         DIST = 0.0D0
                                                                             DPA01010
         DO 160 J=1,P
                                                                             DPA01020
               DIFF = X(K_{i}J) - X(I_{i}J)
                                                                            DFA01030
               DIST = DIST + DIFF*DIFF
                                                                             DPA01040
         CONTINUE
  160
                                                                             DFA01050
         NRM2 = NORMS(I)
                                                                            DPA01060
         T1 = (DIST + NRM1*NRM1 - NRM2*NRM2) / (2.0D0*NRM1)
                                                                            DPA01070
         DENSE(I) = T1
                                                                             DPA01080
         RV1(I) = T1
                                                                            DPA01090
  170 CONTINUE
                                                                            DPA01100
      IF (.NOT. SORTOR) GOTO 175
                                                                            DPA01110
C
                                                                            DPA01120
C:::::SORT AND PRINT NORMAL DISTANCES TO POINT K.
                                                                            DPA01130
С
                                                                            DPA01140
      CALL ISORT1(N,OUTDIS,DENSE)
                                                                            DPA01150
      WRITE(OUT,1010)
                                                                            DPA01160
      DO 172 I=1,N
                                                                            DPA01170
         J = OUTDIS(I)
                                                                            DPA01180
         WRITE(OUT,1011) I,J,DENSE(J)
                                                                            DPA01190
  172 CONTINUE
                                                                            DPA01200
      GO TO 120
                                                                            DPA01210
С
                                                                            DPA01220
C:::::DO STEM & LEAF DISPLAY OF NORMAL DISTANCES TO POINT K.
                                                                            DPA01230
```

.C		DPA01240
175	WRITE(OUT,1008) K	DPA01250
	CALL SLDSPY(RV1,IV1,IV2,IV3,OUTDIS,80,N,300,IERR,OUT)	DPA01260
	CALL IERRIO(IERR,OUT,16,16H STEM & LEAF)	DPA01270
<u> </u>	•	DPA01280
	ESTABLISH CUTOFF DISTANCE.	DPA01290
Ĉ		DPA01300
	WRITE(OUT,1012)	DPA01310
	READ (IN,1013) DIST	TIPA01320
	WRITE(OUT,1009) K	DPA01330
	DO 180 I=1,N	DPA01340
	IF (I .EQ. K) GO TO 180	DPA01350
	IF (DABS(DENSE(I)) .LE. DIST) WRITE(OUT,1004) I	DPA01360
	IF (DENSE(I) .LE. 0.0D0) WRITE(OUT,1005) I	DPA01370
180	CONTINUE	DPA01380
	GO TO 120	DPA01390
C		DPA01400
200	STOP	DPA01410
C		DPA01420
1001	FORMAT(/40H I NORMS DENSITY OUTDIS)	DPA01430
1002	FORMAT(14,2D12,3,218)	DPA01440
1003	FORMAT(/35H POINT CHECKING (TYPE 0 TO STOP): /)	DPA01450
1004	FORMAT(I4)	DPA01460
1005	FORMAT(18)	DPA01470
1006	FORMAT(/25H INDEX MUST BE FROM 1 TO ,I4)	DPA01480
1007	FORMAT(112,3D12.3)	DPA01490
1008	FORMAT(/18H STEM & LEAF FOR +14)	DPA01500
1009	FORMAT(/15H NEB OUT FOR ,I4)	DPA01510
1010	FORMAT(/20H I PT DIST /)	DPA01520
1011	FORMAT(214,D12.3)	DPA01530
012	FORMAT(/20H INPUT CUTOFF VALUE)	DPA01540
013	FORMAT(F10+2)	DPA01550
C		DPA01560
	END	DPA01570

*

SUBROUTINE MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN) MAT00010 INTEGER NM, MN, N, P, OUT, IN MAT00020 DOUBLE PRECISION X(NM,MN),EPS MAT00030 LOGICAL SORTOR MAT00040 С MAT00050 C:::::PARAMETER DECRIPTION: MAT00060 C MAT00070 C **ON INPUT:** MAT00080 С MAT00090 C NM IS THE DECLARED ROW DIMENSION OF X. MAT00100 C MAT00110 C MN IS THE DECLARED COLUMN DIMENSION OF X. MAT00120 C MAT00130 С **ON OUTPUT:** MAT00140 C MAT00150 C N IS THE NUMBER OF ROWS IN X. MAT00160 C MAT00170 C P IS THE NUMBER OF COLUMNS IN X. MAT00180 C MAT00190 C X IS THE DATA MATRIX (WITH NO CONSTANT COLUMNS). MAT00200 C MAT00210 C EPS IS A SMALL SCALING CONSTANT USED IN COMPUTING MAT00220 C THE DENSITY VALUES FOR EACH POINT. MAT00230 C MAT00240 C SORTOR IS A LOGICAL FLAG WHICH CONTROLS THE MAT00250 **POINT-CHECKING PROCEDURE:** MAT00260 IF SORTOR IS .TRUE. SORTED DISTANCES ARE DISPLAYED. MAT00270 C IF SORTOR IS .FALSE. STEM & LEAF AND A USER-SPECIFIED MAT00280 С CUTOFF POINT IS USED. MAT00290 C MAT00300 C OUT IS THE UNIT OUTPUT DEVICE. MAT00310 С MAT00320 C IN IS THE UNIT INPUT DEVICE. MAT00330 С MAT00340 EPS = 1.0D-6MAT00350 SORTOR = .FALSE.MAT00360 OUT = 6MAT00370 IN = 5MAT00380 C MAT00390 C:::::USER SHOULD SUPPLY THE DESIRED MATRIX CALL HERE. MAT00400 С MAT00410 CALL GETMAT(NM, MN, N, P, X) MAT00420 C MAT00430 RETURN MAT00440 END MAT00450

Appendix 3 (cont.)

Other FORTRAN Routines Used by DPA FORTRAN

ISORTI sorts N real values in increasing order through an integer index vector.

SLDSPY is part of a FORTRAN package implementing Tukey's Stem-and-Leaf Display [7]. It was written by D. Hoaglin and S. Wasserman and appears in ROSEPACK version 0.4, developed at NBER/CRC.

IERRIO is also in ROSEPACK version 0.4. It prints an integer error return code along with a message. It can be replaced by a WRITE statement and FORMAT statement.

XECUTION BEGINS...

I	NORMS	DENSITY	OUTDIS
1	0.454D-01	0.207D+04	14
2	0.9370-01	0.114D+04	3
3	0.114D+00	0.560D+03	ī
4	0.966D-01	0.669D+03	2
5	0.697D-01	0.366D+04	5
6	0.903D-01	0.405D+04	1
7	0.2480-01	0.154D+05	5
8	0.1340+00	0.2730+03	0
9	0.8640-01	0.3020+04	4
10	0.1120+00	0.7000+03	0
11	0.924D-01	0.113D+04	2
12	0.797D-01	0.2930+04	. 8
13	0.1170+00	0.4180+03	1
14	0.9120-01	0.246D+04	2
15	0.1020+00	0.8380+03	2
16	0.6720-01	0.5630+04	7
17	0.1470+00	0.271D+03	0
18	0.114D+00	0.412D+03	Ō
19	0.1000+00	0.9040+03	1
20	0.7420-01	0.1200+04	1
21	0.4890+00	0.3030+03	2
22	0.494D+00	0.3920+03	1
23	0.5000+00	0.3030+03	Ň

POINT CHECKING (TYPE O TO STOP):

> 8

3

STEM & LEAF FOR 8

STEM-AND-LEAF DISPLAY, N = 23

(UNIT = 0.1000D-02)

1	0	I	0
1	1	I	
2	2	I	7
3	3	I	5
3	4	I	
5	5	I	47
8	6	I	178
9	7	I	3
3	8	I	348
11	9	I	169
8	10	I	
8	11	I	26
6	12	I	359

HII 0.4893 0.4934

0.4976

IERR = 0 STEM & LEAF

INPUT CUTOFF VALUE >.04

EBOUT FOR 8

POINT CHECKING (TYPE O TO STOP):

> 10

STEM & LEAF FOR 10

STEM-AND-LEAF DISPLAY, N = 23

0.10000-02) (UNIT = 3 0 I 057 3 Ι 1 6 2 I 019 8 3 I 89 11 4 I 456 3 5 I 666 9 6 I 3 8 7 I 1245 4 8 I 4 9 Ι 4 I 10 11 4 I 3

3

HI I

0.5431

0.5480 0.5530

23

IERR = 0 STEM & LEAF

INPUT CUTOFF VALUE >.01

NEB OUT FOR 10 17 18

POINT CHECKING (TYPE O TO STOP):

> 17

STEM & LEAF FOR 17

STEM-AND-LEAF DISPLAY, N =

(UNIT = 0.1000D-01)

1 0 I 0 2 T I 3 5 F I 455 7 S I 6667777 11 0. I 8889 7 1 I 011 4 ΤI 4 F I 5 3 HI I 0.6097 0.6150 0.6203 IERR =O STEM & LEAF INPUT CUTOFF VALUE >0 NEB OUT FOR 17 POINT CHECKING (TYPE O TO STOP): > 18 STEM & LEAF FOR 18 STEM-AND-LEAF DISPLAY, N = 23 (UNIT = 0.1000D-02)2 0 I 09 2 1 Ι 2 2 Ι 3 3 I 7 6 4 I 066 10 5 I 1236 2 I 48 6 11 7 I 0457 7 8 I 7 6 9 I 04

3

4

4

IERR = 0 STEM & LEAF

10

11

Ι

HI I

I 8

0.4677

0.4718

0.4759

INPUT CUTOFF VALUE

POINT CHECKING (TYPE 0 TO STOP):

STEM-AND-LEAF DISPLAY, N = 23

LO I 0.0056 0.0112 3 0.0 (UNIT = 0.1000D-01)4. I 9 4 4 5 I T I 33 6 F I 44 8 7 S I 6777777 5. I 88999 8 3 6 I 00 TI3 1 IERR = 0 STEM & LEAF INPUT CUTOFF VALUE >.02 NEB OUT FOR 23 21 22 POINT CHECKING (TYPE O TO STOP): 50 R; T=0.20/1.16 16:42:39

>

The Sterling Data (X Matrix)

6.681	TION	LABEL

1	AUSTRALIA	29.35	2.87	2329.68	2.87
2	AUSTRIA	23.32	4.41	1507.99	3.93
3	BELGIUM	23.8	4.43	2109.47	3.82
4	BOLIVIA	41.89	1.67	199.17	0.22
5	BRAZIL	42.19	0.83	707+13	4.56
6	CANATIA	31.72	2.95	/∠0+4/ ว000 00	2.43
2 .	CHTLE	39.74	1.30	4702+08 449 07	2.67
8	CHINA(TATWAN)	44.75	0 47	002+00	4.51
9	COLOMBIA	46.64	1.04	207132	3.08
10	COSTA RICA	47.64	1.14	∠/0+0J A71 0A	2.8
11	DENMARK	24.42	7 07	9/1+24 940/ 57	7.00
12	ECHATIOR	46.71	1 10	≪470+00 107 mm	2.10
13	ETNLAND	22 08	1 · 17	28/1//	A 70
14	EDANCE	27+04	2.137	1681+20	4.52
15	CEDMANN E D	23108	4.7	2213.82	4+02
14	ODEROF	23:31	3,35	2457.12	3+44
10	OKELUE	20+62	3.1	870+85	6.28
10	UUATEMALA	46.05	0.87	289.71	1.48
18	HUNDURAS	47.32	0.58	232.44	3.19
19	ICELAND	34+03	3.08	1900.1	1.12
20	INDIA	41.31	0.96	88.94	1.54
21	IRELAND	31.16	4.19	1139.95	2.99
22	ITALY	24.52	3.48	1390.	3.54
23	JAPAN	27.01	1,91	1257.28	8.21
24	KOREA	41.74	0.91	207.69	5.81
25	LUXEMBOURG	21.8	3.73	2449.39	1.57
26	MALTA	32.54	2.47	601.05	8.12
22	NORWAY	25.95	3.67	2231.03	3,62
28	NETHERLANDS	24.71	3.25	1740.7	7.66
29	NEW ZEALAND	32.61	3.17	1487.52	1.76
30	NICARAGUA	45.04	1.21	325.54	2.48
31	PANAMA	43.56	1.2	548.54	3.61
32	PARAGUAY	41.18	1.05	290 54	1.03
33	PERU	44.19	1 78	400.04	0.47
34	PHTLLTPINES	46.25	1,10	157 01	2.
35	PORTUGAL	28.96	2 05	IJ2+01 E70 E4	7.48
36	SOUTH AFRICA	31.94	·	U/7+UI /54 14	2.10
37	SOUTH ENODERTA	31.92	1 50	2001+11 2001+11	2.1/
38	SPATN SPATN	07 74	1.12	200,96	~ · ·
39	CHETCH	27 1 74	2.67	768.79	4+00
10	CUTTZEELAND	22.144	4.04	3299.49	3.01
11	TUDELA	AT AD	3./3	2630.96	2.1/
12		43+42	1.08	389.66	2.70
13	INTER KENDERN	40+12	1.21	249.87	1+10
14	UNITED KINGDOM	2342/	4.46	1813.93	2.01
55	UNITED STATES	27+UI A/ A	3.43	4001.89	2.45
16	VENEZUELA	46+4	0.9	813.39	0.53
17	ZAMELA	45.25	0.56	138.33	5.14
161	JAMAICA	41.12	1.73	380+47	10+23
	URUGUAY	28+13	2.72	766.54	1.89
50	LIBYA	43.69	2.07	123.59	16.71
	MALAYSIA	47.2	0.66	242.65	5.08



References

- 1. Andrews, D. F. and Pregibow, D., "Finding the Outliers that Matter," Technical Report No. 1, University of Toronto, Graduate Department of Statistics, March 1977.
- Businger, P. and Golub, G., "Linear Least Squares Solutions by Householder Transformations, in Wilkinson," T. and Reinsch, C., (Eds.), Handbook for Automatic Computation, V.2: Linear Algebra, Springer-Verlag, p. 111-118, 1971, Prepub. in Math. 7, p. 269-76, 1985.
- Gnanadesikan, R., "Methods for Statistical Data Analysis of Multivariate Observations," John Wiley and Sons, New York, 177, p. 82-120, 258-284.
- 4. Kendall, M. G. and Stuart, A., "The Advanced Theory of Statistics," Vol. 3, Ed. 2, Hafner Publ. Co., New York, 1968, p. 338-9.
- 5. Hoaglin, D. C. and Welsch, R. E., "The Hat Matrix in Regression and ANOVA," WP 901-77, Alfred P. Sloan School of Management, MIT, Jan. 1977.
- 6. Oliver, D., "Troll Experimental Programs: CLUSTER ANALYSIS," Computer Research Center for Economics and Management Science, National Bureau of Economic Research, Inc., August 1975.
- 7. Tukey, J. W., "Exploratory Data Analysis," Addison-Wesley, 1977.
- 8. Welsch, R. E. and Kuh, E., "Linear Regression Diagnostics," MIT and NBER, Working Paper No. 173, March 1977.