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.

Key Words and Phrases: leverage, outliers, least squares, cluster analysis, data analysis, hat matrix, mathematical software.

CR Categories 5.14 and 5.5.

Table of Contents

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

$$X^TX\beta = X^Ty, \quad \hat{\beta} = (X^TX)^{-1}X^Ty, \quad X\hat{\beta} = \hat{y} = X(X^TX)^{-1}X^Ty.$$

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 i^{th} 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.

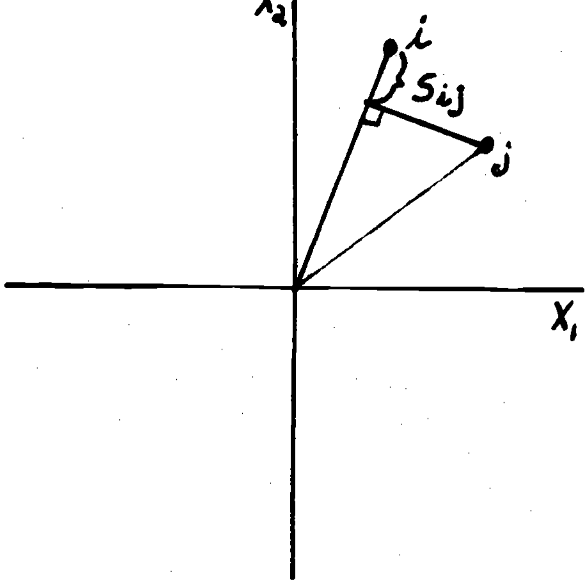
**See Gnanadesikan [3].

A Problem

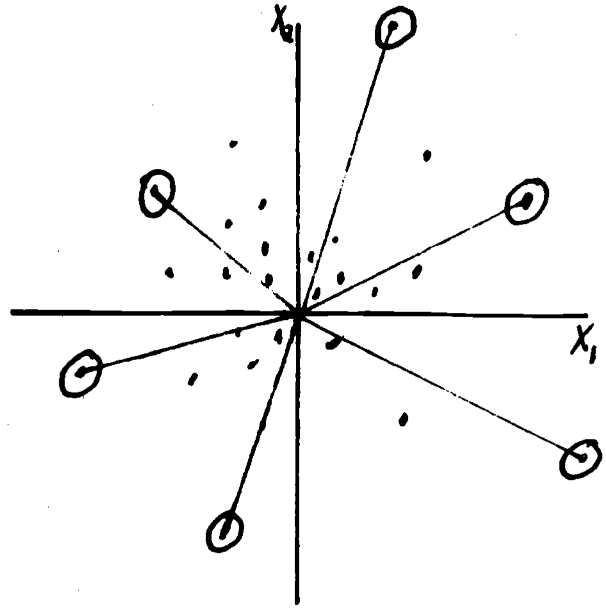
As discussed by Welsch and Kuh [8], the h_i effectively reveal individual leverage points, but may not reveal those leverage points that geometrically form a group (are in close geometric proximity to one another). Proximity to other data reduces the individual leverage, hence the h_i, of any given point.

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.

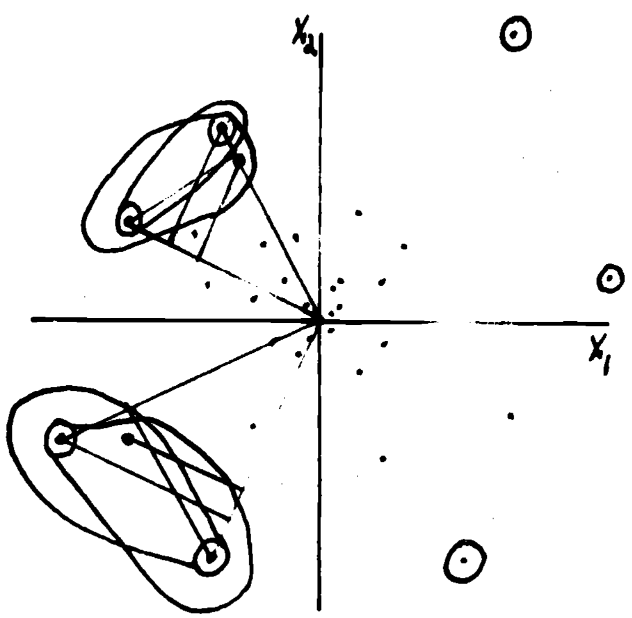
Sequential row deletion is unreliable because it is hard to determine what constitutes a group, and a group could collectively have high leverage, while the h_i of its members might be moderate. The sequential procedure proposed by Andrews and Pregibon [1] can also encounter difficulties for the same 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 $O(n^2p^2)$-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.



1a)



1b)



1c)

Figure 1

1a) Measuring the parallel distance of point j from point i .

1b) Finding outdistancers.

1c) Finding leverage groups headed by outdistancers.

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; $X \leftarrow X - \bar{X}$, where the rows of $\bar{X}$ 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). Tally those 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 $x = (x_1, x_2, \ldots x_n)^T$, the ℓ_∞ norm of x, $||x||_\infty = \max_{1\le i\le n} |x_i|$

**Given vector $x = (x_1, x_2, \ldots, x_n)^T$, the ℓ_2 norm of x, $||x||_2 = (\sum_{i=1}^{n} x_i^2)^{1/2}$

$= (x^T x)^{1/2}$

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 non-symmetric, $d_1 R d_2 \not\leftrightarrow d_2 R d_1$, 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) 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_i, 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.

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_i^*, and the h_i and r_i^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_i 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.

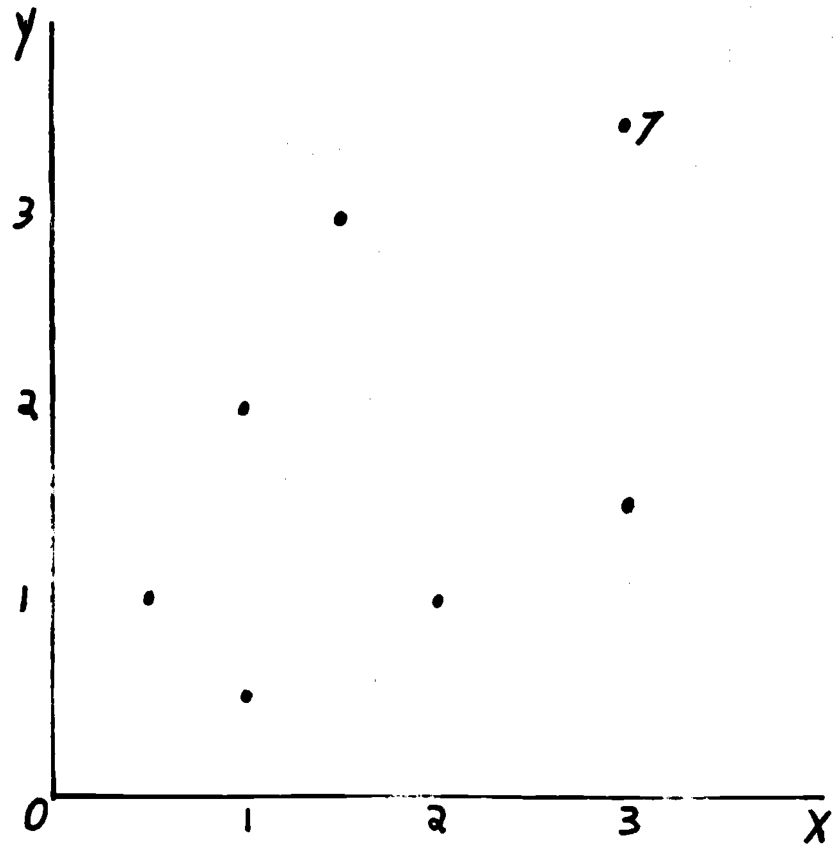


Figure 2

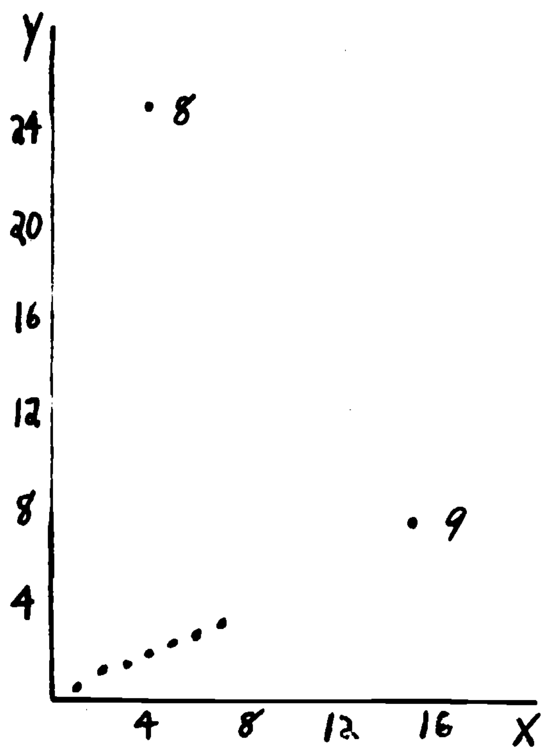


Figure 3

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

As a second example, consider the data: $(i, (i/2) + \varepsilon_i)$ for $i = 1, 2, \ldots, 7$ 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_i^* 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_i and r_i^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 $X^TX = X^T\beta$.

+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 $H = X(X^TX)^{-1}X^T$, $X = QR$. Therefore, $H = QR(R^TQ^TQR)^{-1} R^TQ^T = QQ^T$ (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 i^{th} 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 NM,MN,N,P,I,J,K,OUT,IN,IPLUS1,IERR,IV1(300),OUTDIS(510)
      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:::::GET DATA MATRIX AND PARAMETER VALUES.
C
      CALL MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN)
      DFP = 2.0D0 * DSQRT(DFLOAT(P))
C
C:::::CENTER THE DATA.
C
      DO 20 I=1,P
         TEMP = 0.0D0
         DO 10 J=1,N
              TEMP = TEMP + X(J,I)
   10    CONTINUE
         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))
   15    CONTINUE
C
C:::::NORMALIZE THE DATA SUCH THAT THE OBSERVATION SPACE IS SCALED INTO
C:::::A HYPERCUBE OF MAIN DIAGONAL LENGTH 1.
C
         DO 20 J=1,N
              X(J,I) = (X(J,I) / MAX) / DFP
   20 CONTINUE
      DO 30 I=1,N
         DENSE(I) = 0.0D0
         OUTDIS(I) = 0
   30 CONTINUE
C
C:::::COMPUTE ROW L2 NORMS.
C
      DO 50 I=1,N
         TEMP = 0.0D0
         DO 40 J=1,P
              TEMP = TEMP + X(I,J)*X(I,J)
   40    CONTINUE
         NORMS(I) = DSQRT(TEMP)
   50 CONTINUE
C
C:::::COMPUTE DISTANCES SQUARED.
C
      DO 105 I=1,N
         IF (I .EQ. N) GOTO 105
         IPLUS1 = I + 1
         NRM1 = NORMS(I)
         DO 100 J=IPLUS1,N
              DIST = 0.0D0
```

```
            DO 70 K=1,P
                DIFF = X(I,K) - X(J,K)
                DIST = DIST + DIFF*DIFF
   70       CONTINUE
C
C:::::COMPUTE NORMAL (PARALLEL) DISTANCES.
C
   75       NRM2 = NORMS(J)
            T1 = (DIST + NRM1*NRM1 - NRM2*NRM2) / (2.0D0*NRM1)
            T2 = (DIST + NRM2*NRM2 - NRM1*NRM1) / (2.0D0*NRM2)
            DENSE(I) = DENSE(I) + 1.0D0 / (EPS + DABS(T1))
            DENSE(J) = DENSE(J) + 1.0D0 / (EPS + DABS(T2))
C
C:::::TALLY OUTDISTANCING POINTS.
C
            IF (T1 .LE. 0.0D0) OUTDIS(I) = OUTDIS(I) + 1
            IF (T2 .LE. 0.0D0) OUTDIS(J) = OUTDIS(J) + 1
  100     CONTINUE
  105 CONTINUE
      WRITE(OUT,1001)
      DO 110 I=1,N
         WRITE(OUT,1002) I,NORMS(I),DENSE(I),OUTDIS(I)
  110 CONTINUE
C
C:::::CHECK INDIVIDUAL POINTS OF INTEREST.
C
  120 WRITE(OUT,1003)
C
C:::::GET POINT INDEX.
C
      READ(IN,1004) K
      IF (K*(2*N + 1 - 2*K)) 130,200,150
  130 WRITE(OUT,1006) N
      GO TO 120
C
C:::::COMPUTE DISTANCES.
C
  150 NRM1 = NORMS(K)
      DENSE(K) = 0.0D0
      RV1(K) = 0.0D0
      DO 170 I=1,N
         OUTDIS(I) = I
         IF (I .EQ. K) GO TO 170
         DIST = 0.0D0
         DO 160 J=1,P
             DIFF = X(K,J) - X(I,J)
             DIST = DIST + DIFF*DIFF
  160    CONTINUE
         NRM2 = NORMS(I)
         T1 = (DIST + NRM1*NRM1 - NRM2*NRM2) / (2.0D0*NRM1)
         DENSE(I) = T1
         RV1(I) = T1
  170 CONTINUE
      IF (.NOT. SORTOR) GOTO 175
C
C:::::SORT AND PRINT NORMAL DISTANCES TO POINT K.
C
      CALL ISORT1(N,OUTDIS,DENSE)
      WRITE(OUT,1010)
      DO 172 I=1,N
         J = OUTDIS(I)
         WRITE(OUT,1011) I,J,DENSE(J)
  172 CONTINUE
      GO TO 120
C
C:::::DO STEM & LEAF DISPLAY OF NORMAL DISTANCES TO POINT K.
```

```
.C
  175 WRITE(OUT,1008) K
      CALL SLDSPY(RV1,IV1,IV2,IV3,OUTDIS,80,N,300,IERR,OUT)
      CALL IERRIO(IERR,OUT,16,16H STEM & LEAF       )
C
C::::ESTABLISH CUTOFF DISTANCE.
C
      WRITE(OUT,1012)
      READ (IN,1013) DIST
      WRITE(OUT,1009) K
      DO 180 I=1,N
         IF (I .EQ. K) GO TO 180
         IF (DABS(DENSE(I)) .LE. DIST) WRITE(OUT,1004) I
         IF (DENSE(I) .LE. 0.0D0) WRITE(OUT,1005) I
  180 CONTINUE
      GO TO 120
C
  200 STOP
C
 1001 FORMAT(/40H    I      NORMS       DENSITY     OUTDIS       )
 1002 FORMAT(I4,2D12.3,2I8)
 1003 FORMAT(/35H POINT CHECKING (TYPE 0 TO STOP):       /)
 1004 FORMAT(I4)
 1005 FORMAT(I8)
 1006 FORMAT(/25H INDEX MUST BE FROM 1 TO    ,I4)
 1007 FORMAT(I12,3D12.3)
 1008 FORMAT(/18H STEM & LEAF   FOR   ,I4)
 1009 FORMAT(/15H NEB OUT   FOR   ,I4)
 1010 FORMAT(/20H    I  PT    DIST    /)
 1011 FORMAT(2I4,D12.3)
 1012 FORMAT(/20H INPUT CUTOFF VALUE     )
 1013 FORMAT(F10.2)
C
      END
```

```
      SUBROUTINE MATRIX(NM,MN,N,P,X,EPS,SORTOR,OUT,IN)
      INTEGER NM,MN,N,P,OUT,IN
      DOUBLE PRECISION X(NM,MN),EPS
      LOGICAL SORTOR
C
C:::::PARAMETER DECRIPTION:
C
C     ON INPUT:
C
C        NM IS THE DECLARED ROW DIMENSION OF X.
C
C        MN IS THE DECLARED COLUMN DIMENSION OF X.
C
C     ON OUTPUT:
C
C        N IS THE NUMBER OF ROWS IN X.
C
C        P IS THE NUMBER OF COLUMNS IN X.
C
C        X IS THE DATA MATRIX (WITH NO CONSTANT COLUMNS).
C
C        EPS IS A SMALL SCALING CONSTANT USED IN COMPUTING
C             THE DENSITY VALUES FOR EACH POINT.
C
C        SORTOR IS A LOGICAL FLAG WHICH CONTROLS THE
C             POINT-CHECKING PROCEDURE:
C             IF SORTOR IS .TRUE.  SORTED DISTANCES ARE DISPLAYED.
C             IF SORTOR IS .FALSE. STEM & LEAF AND A USER-SPECIFIED
C                                  CUTOFF POINT IS USED.
C
C        OUT IS THE UNIT OUTPUT DEVICE.
C
C        IN IS THE UNIT INPUT DEVICE.
C
      EPS = 1.0D-6
      SORTOR = .FALSE.
      OUT = 6
      IN = 5
C
C:::::USER SHOULD SUPPLY THE DESIRED MATRIX CALL HERE.
C
      CALL GETMAT(NM,MN,N,P,X)
C
      RETURN
      END
```

Appendix 3 (cont.)

Other FORTRAN Routines
Used by DPA FORTRAN

ISORT1 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.

Appendix 4

```
start

EXECUTION BEGINS...

  I       NORMS        DENSITY    OUTDIS
  1    0.454D-01    0.207D+04      14
  2    0.937D-01    0.114D+04       3
  3    0.114D+00    0.560D+03       1
  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    0.248D-01    0.154D+05       5
  8    0.134D+00    0.273D+03       0
  9    0.864D-01    0.302D+04       4
 10    0.112D+00    0.700D+03       0
 11    0.924D-01    0.113D+04       2
 12    0.797D-01    0.293D+04       8
 13    0.117D+00    0.418D+03       1
 14    0.912D-01    0.246D+04       2
 15    0.102D+00    0.838D+03       2
 16    0.672D-01    0.563D+04       7
 17    0.147D+00    0.271D+03       0
 18    0.114D+00    0.412D+03       0
 19    0.100D+00    0.904D+03       1
 20    0.742D-01    0.120D+04       1
 21    0.489D+00    0.303D+03       2
 22    0.494D+00    0.392D+03       1
 23    0.500D+00    0.303D+03       0

POINT CHECKING (TYPE 0 TO STOP):

>   8

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


  3          HI  I     0.4893     0.4934     0.4976
```

```
IERR =     0 STEM & LEAF

INPUT CUTOFF VALUE
>.04

NEB OUT   FOR    8
 17
 19

POINT CHECKING (TYPE 0 TO STOP):

>  10

STEM & LEAF   FOR  10


        STEM-AND-LEAF DISPLAY,  N =    23



             ( UNIT =   0.1000D-02 )

    3              0  I 057
    3              1  I
    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  I
    4             10  I
    4             11  I 3


    3             HI I     0.5431    0.5480    0.5530
IERR =     0 STEM & LEAF

INPUT CUTOFF VALUE
>.01

NEB OUT   FOR   10
 17
 18

POINT CHECKING (TYPE 0 TO STOP):

>  17

STEM & LEAF   FOR  17


        STEM-AND-LEAF DISPLAY,  N =    23



             ( 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              T I
    4              F I 5


    3             HI I     0.6097    0.6150    0.6203
 IERR =     0 STEM & LEAF

 INPUT CUTOFF VALUE
 >0

 NEB OUT   FOR   17

 POINT CHECKING (TYPE 0 TO STOP):

 >  18

 STEM & LEAF   FOR  18


          STEM-AND-LEAF DISPLAY,  N =    23



              ( UNIT =   0.1000D-02 )

    2              0  I 09
    2              1  I
    2              2  I
    3              3  I 7
    6              4  I 066
   10              5  I 1236
    2              6  I 48
   11              7  I 0457
    7              8  I 7
    6              9  I 04
    4             10  I
    4             11  I 8


    3             HI I     0.4677    0.4718    0.4759
 IERR =     0 STEM & LEAF

 INPUT CUTOFF VALUE
 >.01

 NEB OUT   FOR   18
  10

 POINT CHECKING (TYPE 0 TO STOP):

 >  23
```

```
        STEM-AND-LEAF DISPLAY,  N =    23


   3              LO I    0.0       0.0056    0.0112


              ( UNIT =   0.1000D-01 )

   4              4. I 9
   4              5  I
   6               T I 33
   8               F I 44
   7               S I 6777777
   8              5. I 88999
   3              6  I 00
   1               T I 3

IERR =    0 STEM & LEAF

INPUT CUTOFF VALUE
>.02

NEB OUT   FOR   23
 21
 22

POINT CHECKING (TYPE 0 TO STOP):

>0
R; T=0.20/1.16 16:42:39

>
```

Appendix 5

The Sterling Data (X Matrix)

POSITION	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	2108.47	3.82
4	BOLIVIA	41.89	1.67	189.13	0.22
5	BRAZIL	42.19	0.83	728.47	4.56
6	CANADA	31.72	2.85	2982.88	2.43
7	CHILE	39.74	1.34	662.86	2.67
8	CHINA(TAIWAN)	44.75	0.67	289.52	6.51
9	COLOMBIA	46.64	1.06	276.65	3.08
10	COSTA RICA	47.64	1.14	471.24	2.8
11	DENMARK	24.42	3.93	2496.53	3.99
12	ECUADOR	46.31	1.19	287.77	2.19
13	FINLAND	27.84	2.37	1681.25	4.32
14	FRANCE	25.06	4.7	2213.82	4.52
15	GERMANY F.R.	23.31	3.35	2457.12	3.44
16	GREECE	25.62	3.1	870.85	6.28
17	GUATEMALA	46.05	0.87	289.71	1.48
18	HONDURAS	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
27	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	568.56	3.61
32	PARAGUAY	41.18	1.05	220.56	1.03
33	PERU	44.19	1.28	400.06	0.67
34	PHILLIPINES	46.26	1.12	152.01	2.
35	PORTUGAL	28.96	2.85	579.51	7.48
36	SOUTH AFRICA	31.94	2.28	651.11	2.19
37	SOUTH RHODESIA	31.92	1.52	250.96	2.
38	SPAIN	27.74	2.87	768.79	4.35
39	SWEDEN	21.44	4.54	3299.49	3.01
40	SWITZERLAND	23.49	3.73	2630.96	2.7
41	TURKEY	43.42	1.08	389.66	2.96
42	TUNISIA	46.12	1.21	249.87	1.13
43	UNITED KINGDOM	23.27	4.46	1813.93	2.01
44	UNITED STATES	29.81	3.43	4001.89	2.45
45	VENEZUELA	46.4	0.9	813.39	0.53
46	ZAMBIA	45.25	0.56	138.33	5.14
47	JAMAICA	41.12	1.73	380.47	10.23
48	URUGUAY	28.13	2.72	766.54	1.88
49	LIBYA	43.69	2.07	123.59	16.71
50	MALAYSIA	47.2	0.66	242.69	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.

2. 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, 1965.

3. 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.