ALGORITHM 39
CORRELATION COEFFICIENTS WITH MATRIX MULTIPLICATION

PAPKEN SASSOUNI
Burroughs Corporation, Pasadena, California

```
procedure  NORM (x) number of rows: (m) number of columns:
             (n) normalized output: (y) standard deviations:
             (s) ;
value      m, n ; integer m, n ; array x, y, s ;
comment    Given an observation matrix [x] consisting of ob-
             servations x_ij on a population, NORM will cal-
             culate
```

$$y_{ij} = \frac{x_{ij} - \bar{x}_j}{\sqrt{\sum_{i=1}^{m} (x_{ij} - \bar{x}_j)^2}} \qquad \text{for } i = 1, \cdots, m \quad j = 1, \cdots, n$$

and the standard deviations

$$s_j = \sqrt{\frac{\sum_{i=1}^{m} (x_{ij} - \bar{x}_j)^2}{m}}$$

```
             where x̄_j is the mean of observations on the j-th
               factor ;
begin        integer i, j ; real r, h, c, b ;
             r := sqrt (m) ; for j := 1 step 1 until n do
1: begin     h := 0 ;
             for i := 1 step 1 until m do
             h := h + x[i, j] ; h := h/m ; b := 0 ;
             for i := 1 step 1 until m do
2: begin     c := x[i, j] - h ; b := b + c↑2 ; y[i, j] := c
             end 2 ;
             b := sqrt (b) ;
             for i := 1 step 1 until m do
             y[i, j] := y[i, j]/b ; s[j] := b/r
             end 1
             end NORM ;
comment      The normalization is now completed, and we are
               ready to compute the correlation matrix ;
procedure    TRANSMULT (y) number of rows: (m) number of
               columns: (n) symmetrical square matrix result:
               (z) ;
value        m, n ; integer m, n ; array y, z ;
comment      This procedure multiplies two matrices, the first
               being the transpose of the second. The result is a
               symmetrical matrix with respect to the main diag-
               onal, therefore only the lower part of it, including
               the main diagonal, is computed. The upper half is
               obtained by equating corresponding elements;
begin        integer i, j, k ; real h ;
             for j := 1 step 1 until n do
             for i := j step 1 until n do
begin        h := 0 ;
             for k := 1 step 1 until m do
             h := h + y[k, i] × y[k, j] ; z[i, j] := h ;
             if i ≠ j then z[j, i] := h
             end i
end TRANSMULT. [z] is the square matrix of the
  correlation coefficients of the initial observation
  matrix [x]
```