A note on the ordinal canonical correlation analysis of two sets of ranking scores

Mishra, SK

North-Eastern Hill University, Shillong (India)

16 January 2009

A Note on the Ordinal Canonical Correlation Analysis of Two Sets of Ranking Scores

SK Mishra
Department of Economics
North-Eastern Hill University
Shillong, Meghalaya (India)
Contact: mishrasknehu@yahoo.com

Abstract

In this paper we have proposed a method to conduct the ordinal canonical correlation analysis (OCCA) that yields ordinal canonical variates and the coefficient of correlation between them, which is analogous to (and a generalization of) the rank correlation coefficient of Spearman. The ordinal canonical variates are themselves analogous to the canonical variates obtained by the conventional canonical correlation analysis (CCCA). Our proposed method is suitable to deal with the multivariable ordinal data arrays. Our examples have shown that in finding canonical rank scores and canonical correlation from an ordinal dataset, the CCCA is suboptimal. The OCCA suggested by us outperforms the conventional method. Moreover, our method can take care of any of the five different schemes of rank ordering. It uses the Particle Swarm Optimizer which is one of the recent and prized meta-heuristics for global optimization. The computer program developed by us is fast and accurate. It has worked very well to conduct the OCCA.

Keywords: Ordinal, Canonical correlation, rank order, rankings, scores, standard competition, modified competition, fractional, dense, Repulsive Particle Swarm, global optimization, computer program, FORTRAN

A Note on the Ordinal Canonical Correlation Analysis of Two Sets of Ranking Scores

SK Mishra
Department of Economics
North-Eastern Hill University
Shillong, Meghalaya (India)
Contact: mishrasknehu@yahoo.com

I. Introduction: Let us consider a scenario in which thirty players of badminton were rank-ordered for their skill and acumen by two committees of judges, each committee caring for a certain specified aspect of the game. The first committee had four judges on it while the second committee had five members. Each judge rank-ordered the players according to his own perception of competence in the specified aspect of the game, without any consultation with the fellow judges. The problem is to find the degree of concordance between the two specified aspects of the game as exhibited by the thirty players and adjudicated by the two committees. We will denote the rankings awarded by the four judges (on committee-1) by X_1 making a 30x4 matrix and the rankings awarded by the five judges (on committee-2) by X_2 making a 30x5 matrix. The array of pooled rank scores $[X_1|X_2]$ may be called X, a 30x9 matrix.

The problem can be solved in a number of alternative ways, some of which are: (i) finding the best composite scores (Y_1 and Y_2) separately from the ranking scores X_1 and X_2 (assuming independence of X_1 and X_2) and then finding $r(Y_1,Y_2)$ the coefficient of correlation between the two composite scores; (ii) rank-ordering Y_1 and Y_2 to obtain $Z_1=\mathfrak{R}(Y_1)$ and $Z_2=\mathfrak{R}(Y_2)$, where $\mathfrak{R}(.)$ is a suitable rule to obtain the ranking score of (.), and then finding $r(Z_1,Z_2)$; (iii) finding Z_1 and Z_2 that maximize the sum of their squared correlation with $x_{1j;j=1,4} \in X_1$ and $x_{2j:j=1,5} \in X_2$, respectively, and then finding $r(Z_1,Z_2)$; (iv) finding the best composite scores (Y_1 and Y_2) jointly from the ranking scores X_1 and X_2 so as to maximize $r^2(Y_1,Y_2)$; (v) finding the best composite scores (Y_1 and Y_2) jointly from the ranking scores X_1 and X_2 so as to maximize $r^2(Y_1,Y_2)$, to obtain $Z_1=\mathfrak{R}(Y_1)$ and $Z_2=\mathfrak{R}(Y_2)$ and then finding $r(Z_1,Z_2)$; and (vi) finding the best composite scores (Y_1 and Y_2) jointly from the ranking scores X_1 and X_2 so as to maximize $r^2(Z_1,Z_2)$, while $Z_1=\mathfrak{R}(Y_1)$ and $Z_2=\mathfrak{R}(Y_2)$. The first three approaches do not take advantage of joint estimation and thus disregard the information available to them. The last three approaches use the available information and therefore can perform better. Indeed, the numerical exercises on the data given in Table-0 reveal that the coefficients of correlation obtained for the six approaches are: (0.985244), (0.982647),

Table-0. Rankings of Thirty Badminton Players by Two Committees, Each considering a Particular Aspect of the Game

	Rankings by the First Committee Members				Rankings by the Second Committee members				
Sl.No	J_{11}	J_{12}	J_{13}	J_{14}	J_{21}	J_{22}	J_{23}	J_{24}	J_{25}
1	3	8	9	8	14	11	11	6	4
2	25	20	16	22	22	24	25	19	19
3	13	5	4	13	9	8	8	7	14
4	4	6	2	1	2	2	4	1	3
5	27	27	27	25	25	28	24	28	28
6	2	3	3	4	3	3	2	5	1
7	5	4	5	6	8	7	9	3	8
8	18	16	17	17	15	18	19	15	17
9	26	26	25	26	30	27	22	26	20
10	28	30	28	29	26	30	29	30	27
11	11	18	19	21	19	15	15	14	23
12	23	21	22	24	20	23	20	21	24
13	16	10	8	10	17	16	16	12	11
14	8	9	13	7	6	6	3	11	13
15	7	7	7	2	4	5	7	4	7
16	22	23	23	20	23	20	27	25	25
17	9	12	12	11	5	9	14	13	9
18	20	19	24	18	21	22	23	24	21
19	21	25	18	23	24	21	26	22	22
20	14	13	14	15	13	17	10	16	12
21	29	28	29	28	29	26	28	29	29
22	19	22	20	16	16	19	17	23	18
23	24	24	26	27	28	25	21	20	26
24	10	14	11	19	10	12	12	17	10
25	17	15	21	12	18	13	18	8	16
26	15	17	15	14	12	14	6	18	15
27	6	2	1	5	7	4	5	9	5
28	30	29	30	30	27	29	30	27	30
29	12	11	6	9	11	10	13	10	6
30	1	1	10	3	1	1	1	2	2

(0.982647), (0.991362), (0.989321) and (0.995996) respectively. The fourth approach gives us what is known as the 'canonical correlation' that maximizes $r^2(Y_1,Y_2)$: $Y_1=X_1w_1$; $Y_2=X_2w_2$. The fifth approach gives $r(Z_1,Z_2)$ while $Z_1=\Re(Y_1)$, $Z_2=\Re(Y_2)$; $Y_1=X_1w_1$, $Y_2=X_2w_2$ that maximizes $r^2(Y_1,Y_2)$. It may be noted that since this approach aspires to maximize $r^2(Y_1,Y_2)$ rather than $r^2(Z_1,Z_2)$, it performs poorer than the sixth approach that goes in for maximization of $r^2(Z_1,Z_2)$ and hence outperforms all other approaches. This sixth approach gives us the coefficient that we would call the '*ordinal canonical correlation coefficient*'.

Then, the ordinal canonical correlation coefficient, $r(Z_1,Z_2)$, is the coefficient of correlation between two ordinal variables (Z_1 and Z_2), both of them being the composite (ordinal) ranking scores derived from two ordinal multidimensional data sets of ranking scores, X_1 and X_2, such that $r(Z_1,Z_2)$ is of the largest magnitude. It may be considered analogous to the conventional coefficient of canonical correlation in which the composite canonical variates (Y_1 and Y_2) are cardinally measured. It may be noted that while X_1 and X_2 are in themselves the ordinal variables, their transformation to cardinally measured canonical variates is problematic. Therefore, in such conditions, the ordinal coefficient of correlation (an analog of Spearman's rank correlation) would be a more appropriate measure of concordance between two sets of variables (that is, the ranking scores).

II. The Conventional Canonical Correlation Analysis: The conventional canonical correlation analysis (Hotelling, 1936) maximizes the squared (product moment) coefficient of correlation between two composite variates (Y_1 and Y_2) obtained as a linear combination of two sets of data, X_1 and X_2, on m_1 and m_2 variables (respectively) each in n observations [$n > \max(m_1, m_2)$ linearly independent cases]. It is a straightforward (multivariate) generalization of (Karl Pearson's product moment coefficient of) correlation. It is well known that in case of two variables, x_1 and x_2, we have two lines of regression, the one of x_1 on x_2 (i.e. $x_1 = a_0 + x_2a_1 + u$) and the other of x_2 on x_1 (i.e. $x_2 = b_0 + x_1b_1 + v$), and the product of the two regression coefficients is $r^2(x_1,x_2) = a_1b_1 = [\{(x_2'x_2)^{-1}x_2'x_1\}\{(x_1'x_1)^{-1}x_1'x_2\}]$. If x_1 and x_2 both contain multiple variables, which we will call X_1 and X_2 respectively to highlight that both of them are sets of variables (e.g. X_1 containing k number of variables and X_2 containing l number of variables, each in $n > \max(k,l)$ observations), then we obtain $AB = [\{(X_2'X_2)^{-1}X_2'X_1\}\{(X_1'X_1)^{-1}X_1'X_2\}]$. This AB is diagonalized so as to yield D, which is a diagonal matrix containing the eigenvalues (λ_s) of AB in its principal diagonal (and zero elsewhere). This matrix contains $\min(k,l)$ positive elements in its principal diagonal, each being a squared canonical correlation. They canonize $[X_1, X_2]$ into $Z = [Y_1,Y_2]$ such that $(1/n)[Y_1'Y_1] = I$, $(1/n)[Y_2'Y_2] = I$ and $(1/n)[Y_1'Y_2] = D$. Here I is the identity matrix. The largest element in D explains the largest part of standardized co-variation or squared correlation between X_1 and X_2 and so on. Presently we are concerned with the largest squared correlation only.

When the variables in X_1 and X_2 are ordinal, it is mathematically awkward to obtain Y_1 and Y_2 which are the cardinal variables. The conventional canonical correlation analysis does not provide a procedure to obtain ordinal Y_1 and Y_2. Then what remains with us is the option to rank-order Y_1 and Y_2 and obtain $Z_1=\Re(Y_1)$ and $Z_2=\Re(Y_2)$, where $\Re(.)$ is a suitable rule to obtain the ranking score of (.). However, $r^2(Z_1,Z_2)$ does not necessarily preserve (or inherit) the optimality of $r^2(Y_1,Y_2)$. This means that there could be an alternative method to obtain Z_1^* and Z_2^* both of which are ordinal and maximize $r^2(Z_1^*, Z_2^*)$ outperforming the conventional canonical correlation that yields a suboptimal $r^2(Z_1,Z_2)$.

III. Ordinal Canonical Correlation Analysis by Constrained Integer Programming: If Z_1 and Z_2 are ordinal variables obtained by the ordinal (1-2-3-4) ranking rule (see Wikipedia on ranking) then, following the formulation analogous to the one suggested by Korhonen (1984), Korhonen and Siljamaki (1998) and Li and Li (2004), the ordinal canonical correlation may be computed. However, if the scheme of rank ordering is standard competition ranking (1-2-2-4 rule), modified competition ranking (1-3-3-4 rule), dense ranking (1-2-2-3 rule) or fractional ranking (1-2.5-2.5-4 rule), the formulation of constraints in the integer programming problem would be extremely difficult or impracticable.

IV. Ordinal Canonical Correlation Analysis by Particle Swarm Optimization: We propose in this paper to solve the problem of obtaining ordinal composite rankings arrays, Z_1 and Z_2, by an application of the Particle Swarm Optimization (PSO) proposed by Eberhart and Kennedy (1995). We propose to directly optimize $r^2(Z_1, Z_2)$: $Z_1= \mathfrak{R}(Y_1)$, $Z_2= \mathfrak{R}(Y_2)$; $Y_1=X_1w_1$, $Y_2=X_2w_2$, with w_1 and w_2 as decision variables and $\mathfrak{R}(.)$ as the rule of assigning rankings to the individuals. The rule may be that of ordinal, standard competition, modified competition, dense or fractional ranking. The details of the PSO may be obtained on the Wikipedia. Fleischer (2005) gives a lucid description of this approach to global optimization. In particular, we use the Repulsive Particle Swarm (RPS) optimizer (see Wikipedia). This method has been successfully used by the author (Mishra, 2009) for obtaining the leading ordinal principal components from the ordinal datasets.

V. Some Simulated Examples: In Table-1.1 we present the simulated dataset $X=[X_1|X_2]$, the canonical variates ($Y_1=X_1v_1$ and $Y_2=X_2v_2$) obtained by the conventional canonical correlation analysis (CCCA), the canonical variates (¥$_1$=X_1w_1 and ¥$_2$=X_2w_2) obtained by the ordinal canonical correlation analysis (OCCA), the composite ranking scores ($Z_1=\mathfrak{R}(Y_1)$, $Z_2=\mathfrak{R}(Y_2)$) obtained by the CCCA and the composite ranking scores ($\zeta_1=\mathfrak{R}$(¥$_1$), $\zeta_2=\mathfrak{R}$(¥$_2$)) obtained by the OCCA. The ordinal ranking (1-2-3-4) rule has been used for rank-ordering Y_1, Y_2, ¥$_1$ and ¥$_2$. The weights (v for CCCA and w for OCCA) on different variables (X_{11} through X_{24}) are presented in Table-1.2. For the CCCA, $r^2(Y_1,Y_2)$ is 0.759435 and $r^2(Z_1,Z_2)$ is 0.703061. Against these, for the OCCA, r^2(¥$_1$,¥$_2$) is 0.773341 and $r^2(\zeta_1,\zeta_2)$ is 0.768694. Thus, the OCCA outperforms the CCCA.

In Table-2.1(a) we present another simulated dataset $X=[X_1|X_2]$, the canonical variates ($Y_1=X_1v_1$ and $Y_2=X_2v_2$) obtained by the CCCA, the canonical variates (¥$_1$=X_1w_1 and ¥$_2$=X_2w_2) obtained by the OCCA, the composite ranking scores ($Z_1=\mathfrak{R}(Y_1)$, $Z_2=\mathfrak{R}(Y_2)$) obtained by the CCCA and the composite ranking scores ($\zeta_1=\mathfrak{R}$(¥$_1$), $\zeta_2=\mathfrak{R}$(¥$_2$)) obtained by the OCCA. The ordinal ranking (1-2-3-4) rule has been used for rank-ordering Y_1, Y_2, ¥$_1$ and ¥$_2$. The weights (v for CCCA and w for OCCA) on different variables (X_{11} through X_{26}) are presented in Table-2.2(a). For the CCCA $r^2(Y_1,Y_2)$ is 0.727651 and $r^2(Z_1,Z_2)$ is 0.711292. Against these, for the OCCA, r^2(¥$_1$,¥$_2$) is 0.78319 and $r^2(\zeta_1,\zeta_2)$ is 0.79307. Thus, the OCCA outperforms the CCCA.

It may be noted that this dataset has three ties: the couples of individuals (#3, #4), (#12, #13) and (#29, #30) have the same ranking scores in X_1. Thus, the overall rankings based on X_1 will be different for different ranking schemes (standard competitive, modified competitive, dense, ordinal and fractional ranking rules).

In Table-2.1(b) we present (for the dataset in Table-2.1(a)) in two panels the canonical variates (Y_1=X_1v_1 and Y_2=X_2v_2) obtained by the CCCA, the canonical variates (¥$_1$=X_1w_1 and ¥$_2$=X_2w_2) obtained by the OCCA, the composite ranking scores ($Z_1=\Re(Y_1)$, $Z_2=\Re(Y_2)$) obtained by the CCCA and the composite ranking scores ($\zeta_1=\Re$(¥$_1$), $\zeta_2=\Re$(¥$_2$)) obtained by the OCCA. Two different ranking rules (standard competition, 1-2-2-4 and modified competition, 1-3-3-4 rules) have been used for rank-ordering Y_1, Y_2, ¥$_1$ and ¥$_2$. The weights (v for CCCA and w for OCCA) on different variables (X_{11} through X_{26}) are presented in Table-2.2(b). When the standard competition ranking rule is used, the CCCA $r^2(Y_1,Y_2)$ is 0.727651 and $r^2(Z_1,Z_2)$ is 0.71018. Against these, for the OCCA, r^2(¥$_1$,¥$_2$) is 0.83919 and $r^2(\zeta_1,\zeta_2)$ is 0.790452. Once again, the OCCA outperforms the CCCA. When the modified competition ranking rule is used, the CCCA $r^2(Y_1,Y_2)$ is 0.72765 and $r^2(Z_1,Z_2)$ is 0.710437. Against these, for the OCCA correlation r^2(¥$_1$,¥$_2$) is 0.835028 and $r^2(\zeta_1,\zeta_2)$ is 0.790459. In this instance too (when the modified competition ranking rule is used), the OCCA outperforms the CCCA.

The results regarding some other schemes of ranking are presented in Table-2.1(c) and 2.2(c). When the dense (1-2-2-3) ranking rule is used, the CCCA $r^2(Y_1,Y_2)$ is 0.727651 and $r^2(Z_1,Z_2)$ is 0.704068. Against these, for the OCCA, r^2(¥$_1$,¥$_2$) is 0.774216 and $r^2(\zeta_1,\zeta_2)$ is 0.799843. Once again, the OCCA outperforms the CCCA. When the fractional ranking rule is used, the CCCA $r^2(Y_1,Y_2)$ is 0.727653 and $r^2(Z_1,Z_2)$ is 0.710641. Against these, for the OCCA, r^2(¥$_1$,¥$_2$) is 0.780932 and $r^2(\zeta_1,\zeta_2)$ is 0.781356. In this instance too (when fractional ranking rule is used), the OCCA outperforms the CCCA.

VI. A Computer Program for Ordinal Canonical Correlation Analysis: We have developed a computer program (in FORTRAN) for obtaining the results of the ordinal canonical correlation analysis reported in this paper. This program consists of a main program, ORDCANON, and other thirteen subroutines. The subroutine RPS is the central program for the Repulsive Particle Swarm Optimization. It uses LSRCH, NEIGHBOR, RANDOM, FUNC and FSELECT for searching the optimal value. GINI is used to measure the degree of diversity in the population on termination of the optimization program. DORANK obtains rank-ordering according to different schemes on the choice of a parameter, NRL. The subroutines CORD, CORLN, CORA, CORREL and DOCORA are meant for computation of the correlation coefficient. In particular, CORA and DOCORA obtain Bradley's absolute correlation (Bradley, 1985; not discussed or illustrated in this paper), while CORLN and CORREL compute Karl Pearson's coefficient of correlation. CORD obtains the canonical variates and coordinates the rank-ordering as well as the correlation programs and returns the values of decision variables and objective function to FUNC.

The user has to specify two parameters (NOB= n = no. of observations or cases and MVAR = m = no. of variables) in the main program (ORDCANON) as well as CORD. The parameter NRL, which chooses the rank-ordering scheme, is specified in the DORANK subroutine. The RPS also has a number of parameters, which need not normally be changed. However, comments have been given at different places how to change them if required. These parameters relate to tuning of the search algorithm and modifying the dimensions, if required so.

VII. Concluding Remarks: In this paper we have proposed a method to conduct the ordinal canonical correlation analysis that yields ordinal canonical variates and the coefficient of correlation between them, which is analogous to (and a generalization of) the rank correlation coefficient of Spearman. The ordinal canonical variates are themselves analogous to the canonical variates obtained by the conventional canonical correlation analysis. Our proposed method is suitable to deal with the multivariable ordinal data arrays. Our examples have shown that in finding canonical rank scores and canonical correlation from an ordinal dataset, the conventional canonical correlation analysis is suboptimal. The ordinal canonical correlation analysis suggested by us outperforms the conventional method. Moreover, our method can take care of any of the five different schemes of rank ordering. It uses the Particle Swarm Optimizer which is one of the recent and prized meta-heuristics for global optimization. The computer program developed by us is fast and accurate. It has worked very well to conduct the ordinal canonical correlation analysis.

References

Bradley, C. (1985) “The Absolute Correlation”, *The Mathematical Gazette*, 69(447): 12-17.

Eberhart R.C. and Kennedy J. (1995): “A New Optimizer using Particle Swarm Theory”, *Proceedings Sixth Symposium on Micro Machine and Human Science:* 39–43. IEEE Service Center, Piscataway, NJ.

Fleischer, M. (2005): “Foundations of Swarm Intelligence: From Principles to Practice”, Swarming Network Enabled C4ISR, arXiv:nlin.AO/0502003 v1.

Hotelling, H. (1936) “Relations Between Two Sets of Variates”, *Biometrica*, 28: 321-377.

Korhonen, P. (1984) Ordinal Principal Component Analysis, *HSE Working Papers*, Helsinki School of Economics, Helsinki, Finland.

Korhonen, P. and Siljamaki, A. (1998) Ordinal Principal Component Analysis. Theory and an Application”, *Computational Statistics & Data Analysis,* 26(4): 411-424.

Li, J. and Li, Y. (2004) Multivariate Mathematical Morphology based on Principal Component Analysis: Initial Results in Building Extraction”, http://www.cartesia.org/geodoc/isprs2004/comm7/papers/223.pdf

Mishra, S. K. (2009) “The Most Representative Composite Rank Ordering of Multi-Attribute Objects by the Particle Swarm Optimization”, Available at SSRN: http://ssrn.com/abstract=1326386

Wikipedia (2008-a) “Ranking”, available at Wikipedia http://en.wikipedia.org/wiki/Rank_order

Table-1.1: Simulated Data Set for Canonical Correlation Analysis: Conventional vs Ordinal – (Example-1)

	Dataset of Ordinal Ranking: X_1[30,5], X_2[30,4]									Conventional Canonical Correln				Ordinal Canonical Correlation			
	Ordinal Variables Set-1					Ordinal Variables Set-2				Canonical Variates		Rankings		Canonical Variates		Rankings	
Sl No.	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{21}	X_{22}	X_{23}	X_{24}	Y_1	Y_2	Z_1	Z_2	¥$_1$	¥$_2$	ζ_1	ζ_2
1	6	2	3	1	17	16	12	19	11	4.76670	15.46251	3	12	6.10938	14.68026	2	10
2	2	18	5	8	7	2	6	22	8	8.20955	9.38513	6	5	9.54035	7.25369	5	5
3	22	12	17	9	27	9	21	8	12	13.35808	17.27114	15	13	18.95964	17.89666	16	15
4	26	23	13	25	30	29	19	20	21	22.47362	26.37883	25	23	26.79523	26.42061	24	23
5	3	6	2	14	9	17	11	14	16	10.61049	17.86988	9	16	10.56237	17.58613	6	13
6	29	20	14	24	24	12	26	28	30	20.05824	32.38621	21	28	25.75121	30.98147	21	26
7	23	26	10	27	29	13	22	26	23	23.28528	26.61762	27	24	26.31755	25.33733	22	21
8	28	28	28	22	21	21	28	30	22	21.86111	29.88751	24	25	31.21551	28.91349	27	25
9	9	8	8	12	13	4	7	15	14	11.10433	13.12832	12	8	12.48180	11.87593	8	7
10	16	4	20	23	8	14	17	13	26	17.59774	25.61613	18	21	22.57954	25.52915	18	22
11	27	25	30	19	26	15	8	4	19	21.78132	17.38091	23	15	30.19972	17.91865	26	16
12	14	19	19	7	11	26	25	2	6	10.17343	17.33471	8	14	17.46078	19.78599	13	18
13	18	27	27	30	23	28	27	27	29	27.96003	34.67873	30	30	34.94031	34.25203	30	30
14	19	9	24	10	12	23	9	12	17	11.96302	18.54930	13	17	19.59914	18.66177	17	17
15	25	7	4	13	5	10	4	25	2	6.63456	6.55750	5	4	9.07812	4.43795	7	2
16	1	1	1	5	3	1	13	7	1	3.73966	6.19434	2	3	3.63201	6.19775	1	4
17	5	13	16	3	15	20	14	21	18	9.05792	21.31282	7	20	14.58903	20.53819	10	19
18	13	15	9	20	16	24	18	18	4	16.51748	14.62106	17	10	18.11849	14.78012	15	11
19	24	3	23	21	25	6	30	29	7	20.20942	18.90851	22	18	24.85693	17.59310	20	14
20	17	29	18	26	19	25	29	17	28	22.96789	33.42606	26	29	28.91541	34.10874	25	29
21	15	22	12	15	6	19	5	6	15	12.00524	14.66210	14	11	16.59385	15.04450	12	12
22	7	5	11	6	2	5	3	1	5	5.70188	5.07206	4	2	8.48668	5.32391	4	3
23	8	21	15	11	18	11	2	5	13	14.22496	10.93191	16	6	17.42150	10.87844	14	6
24	11	10	21	16	22	22	10	16	20	18.32167	20.94321	19	19	23.04441	20.57709	19	20
25	30	24	25	29	20	30	24	24	25	24.43786	31.18817	28	26	32.41155	31.07194	29	27
26	20	30	22	28	28	27	23	23	27	26.87835	31.46096	29	27	32.99932	31.22719	28	28
27	4	16	26	4	10	3	15	3	9	10.89577	11.79539	11	7	17.39686	12.37365	11	8
28	21	17	29	18	14	18	20	11	24	18.33108	25.97915	20	22	27.24893	26.51493	23	24
29	12	14	6	2	4	8	1	10	3	2.70122	4.40220	1	1	5.95886	3.69263	3	1
30	10	11	7	17	1	7	16	9	10	10.80839	13.96415	10	9	12.67738	14.10951	9	9

.

Table-1.2: Weights on Variables for Construction of Canonical Variates [Y1, Y2 and ¥$_1$, ¥$_2$] – (Example-1)

Variables	First Set of Variables (X_1)					Second Set of Variables (X_2)			
	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{21}	X_{22}	X_{23}	X_{24}
Weights (v for Y)	-0.132450	0.060239	0.219398	0.569803	0.247819	0.163968	0.369060	0.088763	0.611256
Weights (w for ¥)	0.122675	0.126949	0.422850	0.414331	0.173285	0.213664	0.427263	-0.024279	0.599615

.

Table-2.1(a): Simulated Data Set for Canonical Correlation Analysis: Conventional vs Ordinal – (Example-2)

	Dataset of Ordinal Ranking: X_1[30,3], X_2[30,6]									Conventional Canonical Correln				Ordinal Canonical Correlation			
	Ord Var Set-1			Ordinal Variables Set-2						Canonical Variates		Rankings		Canonical Variates		Rankings	
Sl No.	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}	Y_1	Y_2	Z_1	Z_2	¥$_1$	¥$_2$	ζ_1	ζ_2
1	12	1	3	6	2	7	2	7	10	5.56743	8.26115	5	6	2.93062	9.64908	3	6
2	8	10	10	13	3	4	3	3	17	8.60885	6.28686	8	3	8.64823	10.18923	9	8
3	13	15	12	17	25	10	20	25	29	12.91567	24.37481	15	20	13.99462	27.65074	13	18
4	13	15	12	17	25	10	20	25	16	12.91567	23.37397	14	17	13.39738	25.09363	12	16
5	22	23	29	29	23	27	18	23	25	22.01740	34.92023	25	28	24.59520	40.30905	26	29
6	30	27	26	30	13	23	27	29	30	26.44517	30.62335	30	25	24.70628	35.73747	29	27
7	11	9	9	26	16	22	30	27	19	9.25563	29.34996	9	22	11.90437	32.75253	10	21
8	14	16	13	11	15	16	6	15	22	13.85591	21.94492	16	14	13.42606	24.80406	14	14
9	9	6	1	3	7	3	16	13	6	6.17440	8.08058	6	5	3.36901	7.58854	4	3
10	28	13	16	28	21	21	22	14	23	18.56872	27.47321	22	21	15.77477	33.54191	21	23
11	25	26	22	23	27	15	19	8	15	23.52356	23.08959	28	16	21.22057	28.14989	27	20
12	3	2	6	15	4	1	1	1	7	2.80304	3.41229	1	2	2.87502	6.69095	1	2
13	3	2	6	15	4	1	1	1	3	2.80304	3.10434	2	1	2.57640	5.90414	2	1
14	1	3	20	2	10	8	9	2	1	4.26193	9.29777	3	7	8.32832	9.70169	6	7
15	21	18	17	27	29	29	24	16	27	17.98305	36.37321	20	30	19.96881	42.70603	22	30
16	10	8	8	10	18	9	7	10	14	8.31539	16.19411	7	10	7.99362	18.51297	7	10
17	4	5	11	5	1	6	4	9	5	5.09315	7.30465	4	4	5.86301	7.61435	5	4
18	29	20	7	24	17	25	12	20	12	20.66659	29.67092	23	23	15.75391	32.99602	23	22

19	20	28	24	22	28	17	25	24	26	22.62676	29.72480	26	24	23.66690	33.76221	25	24
20	6	21	27	25	26	12	28	19	18	14.58239	23.74624	17	19	20.35714	27.89038	17	19
21	18	4	15	18	8	18	15	5	9	10.76148	17.31617	11	13	8.90846	21.09445	11	12
22	27	29	28	19	24	30	23	18	20	26.34188	35.05868	29	29	26.25676	38.90305	30	28
23	17	19	18	16	22	13	10	22	11	16.93952	23.40787	18	18	16.11543	24.83417	18	15
24	15	7	4	4	9	2	14	21	2	9.36131	9.99752	10	8	5.10514	8.10706	8	5
25	16	24	14	8	14	20	26	12	21	18.07709	22.84247	21	15	17.53396	25.47672	20	17
26	23	12	23	9	11	5	13	17	4	17.09311	12.02022	19	9	13.02001	11.83955	19	9
27	26	17	30	14	20	26	17	26	24	21.26369	33.37791	24	27	21.73915	35.66802	24	26
28	24	25	25	12	19	24	21	28	28	23.10914	32.34754	27	26	23.11367	34.44265	28	25
29	2	22	21	20	12	14	5	6	8	12.61868	16.46741	12	11	16.79872	20.33217	15	11
30	2	22	21	20	12	14	5	6	13	12.61868	16.85235	13	12	17.39596	21.31567	16	13

.

Table-2.2(a): Weights on Variables for Construction of Canonical Variates [Y1, Y2 and ¥$_1$, ¥$_2$] – (Example-2)

Variables	First Set of Variables (X_1)			Second Set of Variables (X_2)					
	X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}
Weights (v for Y)	0.39675	0.41203	0.13145	0.03963	0.33921	0.67897	-0.02408	0.26721	0.07699
Weights (w for ¥)	0.29862	0.36944	0.25648	0.20761	0.35009	0.69753	-0.02791	0.12991	0.19670

.

Table-2.1(b): Canonical Correlation Analysis: Conventional vs Ordinal – using 1-2-2-4 and 1-3-3-4 Ranking Rules

	Panel-1:Ranking by 1-2-2-4 or Standard Competition Rule								Panel-2:Ranking by 1-3-3-4 or Modified Competition Rule							
	Conventional Canonical Correln				Ordinal Canonical Correlation				Conventional Canonical Correln				Ordinal Canonical Correlation			
Sl No.	Canonical Variates		Rankings		Canonical Variates		Rankings		Canonical Variates		Rankings		Canonical Variates		Rankings	
	Y_1	Y_2	Z_1	Z_2	¥$_1$	¥$_2$	ζ_1	ζ_2	Y_1	Y_2	Z_1	Z_2	¥$_1$	¥$_2$	ζ_1	ζ_2
1	7.6276	6.0657	5	6	3.5098	9.0598	5	6	6.6936	7.2298	5	6	3.2442	9.4008	5	6
2	11.7960	4.6173	8	3	9.2485	9.9121	8	8	10.3497	5.5009	8	3	8.6769	10.1471	8	8
3	17.6971	17.8976	14	20	15.8573	26.8126	14	18	15.5274	21.3312	15	20	14.8573	27.2997	15	18
4	17.6971	17.1623	14	17	15.0078	24.1966	14	16	15.5274	20.4556	15	17	14.0738	24.8426	15	16
5	30.1686	25.6403	25	28	26.9024	38.3411	25	29	26.4697	30.5608	25	28	25.1274	39.7063	25	29
6	36.2348	22.4856	30	25	27.0767	33.8264	30	27	31.7929	26.8013	30	25	25.3737	35.2521	30	27
7	12.6819	21.5502	9	22	14.1857	31.0047	10	21	11.1273	25.6877	9	22	13.2159	32.3728	10	21
8	18.9854	16.1127	16	14	14.7432	23.6390	16	14	16.6577	19.2050	16	14	13.8444	24.2221	16	14
9	8.4596	5.9328	6	5	4.2691	7.2703	6	4	7.4230	7.0731	6	5	4.0448	7.5088	6	4
10	25.4417	20.1732	22	21	17.9394	32.2253	22	23	22.3241	24.0438	22	21	16.7253	33.2700	22	23
11	32.2318	16.9546	28	16	22.9541	27.4188	28	20	28.2804	20.2068	28	16	21.5682	28.0702	28	20
12	3.8409	2.5068	1	2	2.7717	6.5917	1	2	3.3699	2.9847	2	2	2.5704	6.8464	2	2
13	3.8409	2.2806	1	1	2.3469	5.7868	1	1	3.3699	2.7153	2	1	2.1786	6.0903	2	1
14	5.8408	6.8265	3	7	7.9251	9.3501	3	7	5.1238	8.1382	3	7	7.2882	9.5670	3	7
15	24.6400	26.7074	20	30	23.0859	40.9763	21	30	21.6196	31.8333	20	30	21.5608	42.1522	21	30
16	11.3936	11.8907	7	10	8.9280	17.9444	7	10	9.9970	14.1717	7	10	8.3520	18.2487	7	10
17	6.9790	5.3631	4	4	5.3276	7.0196	4	3	6.1231	6.3931	4	4	4.9642	7.4133	4	3
18	28.3158	21.7854	23	23	18.6013	31.0821	23	22	24.8459	25.9671	23	23	17.5000	32.4457	23	22
19	31.0037	21.8258	26	24	25.8233	32.5619	26	24	27.2020	26.0141	26	24	24.2439	33.3829	26	24
20	19.9824	17.4367	17	19	21.4845	27.0978	17	19	17.5308	20.7819	17	19	20.1009	27.8614	17	19
21	14.7447	12.7148	11	13	9.5058	20.0059	11	12	12.9381	15.1557	11	13	8.7839	20.9012	11	12
22	36.0937	25.7412	29	29	28.6715	37.0028	29	28	31.6686	30.6840	29	29	26.8740	38.2117	29	28
23	23.2106	17.1868	18	18	17.2991	23.6882	18	15	20.3649	20.4848	18	18	16.2382	24.4515	18	15
24	12.8260	7.3398	10	8	5.6550	7.5923	9	5	11.2545	8.7501	10	8	5.3300	7.9757	9	5
25	24.7694	16.7716	21	15	19.3815	24.3587	20	17	21.7323	19.9939	21	15	18.2668	24.9998	20	17
26	23.4206	8.8255	19	9	12.9116	11.2487	19	9	20.5500	10.5199	19	9	12.0426	11.7283	19	9
27	29.1354	24.5065	24	27	23.4342	33.7197	24	26	25.5639	29.2124	24	27	21.8135	34.8112	24	26
28	31.6642	23.7500	27	26	25.2451	32.6180	27	25	27.7822	28.3110	27	26	23.6512	33.5979	27	25
29	17.2918	12.0918	12	11	17.3551	19.3900	12	11	15.1698	14.4108	13	11	16.3338	20.1671	13	11
30	17.2918	12.3746	12	12	18.2046	20.3962	12	13	15.1698	14.7476	13	12	17.1174	21.1121	13	13

.

Table-2.2(b): Weights on Variables for Construction of Canonical Variates [Y1, Y2 and ¥$_1$, ¥$_2$] Using 1-2-2-4 and 1-3-3-4 Ranking Rules (for Dataset in Table-2.1)

Ranking Rule	Variables as in Table-2.1(a)	First Set of Variables (X_1)			Second Set of Variables (X_2)					
		X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}
1-2-2-4 Rule	Weights (v for Y)	0.54354	0.56460	0.18017	0.02918	0.24907	0.49847	-0.01767	0.19614	0.05656
	Weights (w for ¥)	0.42479	0.39699	0.18803	0.20113	0.36199	0.63581	-0.01781	0.10025	0.20123
1-3-3-4 Rule	Weights (v for Y)	0.47701	0.49531	0.15805	0.03459	0.29680	0.59431	-0.02095	0.23383	0.06735
	Weights (w for ¥)	0.39179	0.38467	0.16958	0.22373	0.34952	0.66922	-0.01693	0.11694	0.18901

Table-2.1(c): Canonical Correlation Analysis: Conventional vs Ordinal – using 1-2-2-3 and 1-2.5-2.5-4 Ranking Rules																
	Panel-1:Ranking by 1-2-2-3 or Dense Ranking Rule								Panel-2:Ranking by 1-2.5-2.5-4 Fractional Rule							
	Conventional Canonical Correln				Ordinal Canonical Correlation				Conventional Canonical Correln				Ordinal Canonical Correlation			
Sl No.	Canonical Variates		Rankings		Canonical Variates		Rankings		Canonical Variates		Rankings		Canonical Variates		Rankings	
	Y_1	Y_2	Z_1	Z_2	¥$_1$	¥$_2$	ζ_1	ζ_2	Y_1	Y_2	Z_1	Z_2	¥$_1$	¥$_2$	ζ_1	ζ_2
1	6.2161	7.7083	4	6	2.6979	6.7817	2	6	3.0253	9.6713	5	6	1.9110	8.5044	3	6
2	9.6123	5.8677	7	3	8.2084	7.0625	8	7	4.6774	7.3605	8	3	5.2552	7.2823	9	4
3	14.4210	22.7453	12	20	13.3203	22.6741	11	17	7.0176	28.5388	14.5	20	9.4776	26.4785	12.5	17
4	14.4210	21.8098	12	17	12.7911	19.8683	11	15	7.0176	27.3702	14.5	17	9.2883	24.7449	12.5	16
5	24.5839	32.5827	22	28	23.4521	31.5439	22	28	11.9626	40.8868	25	28	16.6221	38.2450	25	28
6	29.5273	28.5739	27	25	23.0101	25.5320	26	23	14.3687	35.8587	30	25	16.5256	32.4041	29	24
7	10.3344	27.3850	8	22	10.7479	24.5981	9	19	5.0290	34.3696	9	22	8.0241	31.5176	10	21
8	15.4708	20.4766	13	14	13.6914	20.5121	12	16	7.5284	25.6899	16	14	9.3594	23.9940	14	14
9	6.8937	7.5400	5	5	3.9499	6.4301	3	5	3.3550	9.4634	6	5	2.8323	8.5043	4	5
10	20.7327	25.6354	19	21	15.3254	27.7675	17	24	10.0894	32.1691	22	21	10.6897	31.8762	20	23
11	26.2652	21.5454	25	16	21.3978	25.5067	24	22	12.7813	27.0366	28	16	14.8306	28.0222	27	20
12	3.1299	3.1846	1	2	2.7499	3.3513	1	2	1.5229	3.9979	1.5	2	1.9287	3.7263	1.5	2
13	3.1299	2.8967	1	1	2.4853	2.4880	1	1	1.5229	3.6383	1.5	1	1.7394	3.1928	1.5	1
14	4.7595	8.6754	2	7	8.3234	10.6558	5	9	2.3151	10.8860	3	7	5.6285	11.7373	6	8
15	20.0789	33.9397	17	30	19.1416	37.5738	18	30	9.7710	42.5866	20	30	13.6135	42.6157	21	30
16	9.2845	15.1110	6	10	7.9955	16.3211	7	11	4.5181	18.9592	7	10	5.4916	18.3867	7	10
17	5.6871	6.8151	3	4	5.3961	4.5922	4	3	2.7671	8.5531	4	4	3.7926	6.8997	5	3
18	23.0745	27.6830	20	23	14.9186	24.6245	20	20	11.2295	34.7419	23	23	11.0470	31.5591	23	22
19	25.2640	27.7366	23	24	23.4628	28.4030	23	25	12.2938	34.8043	26	24	16.6768	33.2472	26	25
20	16.2827	22.1581	14	19	19.5350	23.2410	16	18	7.9225	27.8086	17	19	13.8292	27.1434	18	18
21	12.0159	16.1569	10	13	8.5932	17.5683	10	13	5.8472	20.2768	11	13	6.0493	20.5538	11	13
22	29.4121	32.7119	26	29	25.8689	34.4148	27	29	14.3124	41.0469	29	29	18.3718	40.2991	30	29
23	18.9139	21.8402	15	18	15.5438	18.7640	15	14	9.2038	27.4083	18	18	11.2233	24.1583	19	15
24	10.4521	9.3273	9	8	4.7956	4.6449	6	4	5.0867	11.7094	10	8	3.9860	8.5934	8	7
25	20.1838	21.3150	18	15	18.4377	24.9438	19	21	9.8219	26.7430	21	15	12.7660	27.7241	22	19
26	19.0856	11.2150	16	9	12.3553	8.1244	14	8	9.2873	14.0774	19	9	8.8902	11.7500	17	9
27	23.7424	31.1433	21	27	21.0354	29.4759	21	27	11.5532	39.0771	24	27	14.8724	36.0938	24	27
28	25.8026	30.1826	24	26	22.7700	28.8172	25	26	12.5560	37.8705	27	26	16.1673	35.0344	28	26
29	14.0899	15.3647	11	11	16.3923	15.3731	13	10	6.8555	19.2826	12.5	11	11.8153	18.4322	15.5	11
30	14.0899	15.7245	11	12	16.9215	16.4523	13	12	6.8555	19.7321	12.5	12	12.0046	19.0990	15.5	12

Table-2.2(c): Weights on Variables for Construction of Canonical Variates [Y1, Y2 and ¥$_1$, ¥$_2$] Using 1-2-2-3 and 1-2.5-2.5-4 Ranking Rules (for Dataset in Table-2.1)										
Ranking Rule	Variables as in Table-2.1(a)	**First Set of Variables (X_1)**			**Second Set of Variables (X_2)**					
		X_{11}	X_{12}	X_{13}	X_{21}	X_{22}	X_{23}	X_{24}	X_{25}	X_{26}
1-2-2-3 Rule	Weights (v for Y)	0.44297	0.46003	0.14682	0.03696	0.31654	0.63343	-0.02242	0.24922	0.07197
	Weights (w for ¥)	0.26459	0.39826	0.23737	-0.04295	0.46198	0.65507	0.10026	-0.11841	0.21583
1-2.5-2.5-4 Rule	Weights (v for Y)	0.21560	0.22387	0.07140	0.04671	0.39708	0.79477	-0.02801	0.31294	0.08989
	Weights (w for ¥)	0.18934	0.28717	0.16262	0.00408	0.44708	0.79115	0.07020	0.08198	0.13335

```fortran
C      !----------------- MAIN PROGRAM : ORDCANON ----------------------
C      PROVIDES TO USE REPULSIVE PARTICLE SWARM METHOD TO
C      OBTAIN THE LARGEST CANONICAL CORRELATION & COMPOSITE VARIATE RANKS
C      PRODUCT MOMENT AS WELL AS ABSOLUTE CORRELATION (BRADLEY, 1985) MAY
C      BE USED. PROGRAM BY SK MISHRA, DEPT. OF ECONOMICS, NORTH-EASTERN
C      HILL UNIVERSITY, SHILLONG (INDIA)
C      -----------------------------------------------------------------
C                     ADJUST THE PARAMETERS SUITABLY
C              IN THIS MAIN PROGRAM AND IN THE SOBROUTINE CORD
C      WHEN THE PROGRAM ASKS FOR ANY OTHER PARAMETERS, FEED THEM SUITABLY
C      -----------------------------------------------------------------
       PROGRAM ORDCANON
       PARAMETER(NOB=30,MVAR=9)!CHANGE THE PARAMETERS HERE AS NEEDED.
C      -----------------------------------------------------------------
C        NOB=NO. OF CASES AND MVAR=NO. OF VARIABLES IN ALL M= (M1+M2)
C        NOB AND MVAR TO BE ADJUSTED IN SUBROUTINE CORD(M,X,F) ALSO.
C       SET NRL TO DESIRED VALUE IN SUBROUTINE DORANK FOR RANKING SCHEME
C      -----------------------------------------------------------------
       IMPLICIT DOUBLE PRECISION (A-H, O-Z)
       COMMON /KFF/KF,NFCALL,FTIT ! FUNCTION CODE, NO. OF CALLS & TITLE
       CHARACTER *30  METHOD(1)
       CHARACTER *70 FTIT
       CHARACTER *40 INFILE,OUTFILE
       COMMON /CANON/MONE,MTWO
       COMMON /CORDAT/CDAT(NOB,MVAR),QIND1(NOB),QIND2(NOB),R(1),NORM,NCOR
       COMMON /XBASE/XBAS
       COMMON /RNDM/IU,IV ! RANDOM NUMBER GENERATION (IU = 4-DIGIT SEED)
       COMMON /GETRANK/MRNK
       INTEGER IU,IV
       DIMENSION XX(3,50),KKF(3),MM(3),FMINN(3),XBAS(1000,50)
       DIMENSION ZDAT(NOB,MVAR+1),FRANK1(NOB),FRANK2(NOB),RMAT(2,2)
       DIMENSION X(50)! X IS THE DECISION VARIABLE X IN F(X) TO MINIMIZE
C        M = DIMENSION OF THE PROBLEM, KF(=1) = TEST FUNCTION CODE AND
C               FMIN IS THE MIN VALUE OF F(X) OBTAINED FROM RPS
       WRITE(*,*)'====================      WARNING     =============== '
       WRITE(*,*)'ADJUST PARAMETERS IN SUBROUTINES RPS IF NEEDED '
C      ------------------ OPTIMIZATION BY RPS METHOD -------------------
       NORM=2!WORKS WITH THE EUCLIDEAN NORM (IDENTICAL RESULTS IF NORM=1)
       NOPT=1 ! ONLY ONE FUNCTION IS OPTIMIZED
       WRITE(*,*)'==================================================== '
       METHOD(1)=' : REPULSIVE PARTICLE SWARM OPTIMIZATION'
C      INITIALIZE. THIS XBAS WILL BE USED TO INITIALIZE THE POPULATION.
       WRITE(*,*)' '
       WRITE(*,*)'---------- FEED RANDOM NUMBER SEED, AND NCOR ---------'
       WRITE(*,*)' '
       WRITE(*,*)'FEED SEED [ANY 4-DIGIT NUMBER] AND NCOR[0,1]'
       WRITE(*,*)'NCOR(0)=PRODUCT MOMENT; NCOR(1)=ABSOLUTE CORRELATION'
       WRITE(*,*)' '
     1 READ(*,*) IU,NCOR
       IF(NCOR.LT.0.OR.NCOR.GT.1) THEN
       WRITE(*,*)'SORRY. NCOR TAKES ON[0,1] ONLY. FEED SEED & NCOR AGAIN'
       GOTO 1
       ENDIF
       WRITE(*,*)'WANT RANK SCORE OPTIMIZATION? YES(1); NO(OTHER THAN 1)'
       READ(*,*) MRNK
       WRITE(*,*)'INPUT FILE TO READ DATA:YOUR DATA MUST BE IN THIS FILE'
       WRITE(*,*)'CASES (NOB) IN ROWS ; VARIABLES (MVAR) IN COLUMNS'
       READ(*,*) INFILE
       WRITE(*,*)'SPECIFY THE OUTPUT FILE TO STORE THE RESULTS'
       READ(*,*) OUTFILE
       OPEN(9, FILE=OUTFILE)
       OPEN(7,FILE=INFILE)
       DO I=1,NOB
       READ(7,*),CDA,(CDAT(I,J),J=1,MVAR)
       ENDDO
       CLOSE(7)
       DO I=1,NOB
```

```
      DO J=1,MVAR
      ZDAT(I,J+1)=CDAT(I,J)
      ENDDO
      ENDDO
      WRITE(*,*)'DATA HAS BEEN READ. WOULD YOU UNITIZE VARIABLES? [YES=1
     & ELSE NO UNITIZATION]'
      WRITE(*,*)'UNITIZE MEANS TRANSFORMATION FROM X(I,J) TO UNITIZED X'
      WRITE(*,*)'[X(I,J)-MIN(X(.,J))]/[MAX(X(.,J))-MIN(X(.,J))]'
      READ(*,*) NUN
      IF(NUN.EQ.1) THEN
      DO J=1,MVAR
      CMIN=CDAT(1,J)
      CMAX=CDAT(1,J)
      DO I=2,NOB
      IF(CMIN.GT.CDAT(I,J)) CMIN=CDAT(I,J)
      IF(CMAX.LT.CDAT(I,J)) CMAX=CDAT(I,J)
      ENDDO
      DO I=1,NOB
      CDAT(I,J)=(CDAT(I,J)-CMIN)/(CMAX-CMIN)
      ENDDO
      ENDDO
      ENDIF
C     -----------------------------------------------------------------
C     THIS XBAS WILL BE USED AS INITIAL X
      DO I=1,1000
      DO J=1,50
      CALL RANDOM(RAND)
      XBAS(I,J)=RAND ! RANDOM NUMBER BETWEEN (0, 1)
      ENDDO
      ENDDO
C     -----------------------------------------------------------------
      WRITE(*,*)' *****************************************************'
C     -----------------------------------------------------------------
      K=1
      WRITE(*,*)'PARTICLE SWARM PROGRAM TO OBTAIN CANONICAL CORRELATION'
      CALL RPS(M,X,FMINRPS,Q1) !CALLS RPS AND RETURNS OPTIMAL X AND FMIN
      WRITE(*,*)'RPS BRINGS THE FOLLOWING VALUES TO THE MAIN PROGRAM'
      WRITE(*,*)(X(JOPT),JOPT=1,M),' OPTIMUM FUNCTION=',FMINRPS
      IF(KF.EQ.1) THEN
      WRITE(9,*)'REPULSIVE PARTICLE SWARM OPTIMIZATION RESULTS'
      WRITE(9,*)'THE LARGEST CANONICAL R BETWEEN THE SETS OF VARIABLES'
      WRITE(9,*)' ABS(R)=',DABS(R(1)),'; SQUARE(R)=',R(1)**2
      IF(NCOR.EQ.0) THEN
      WRITE(*,*)'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
      WRITE(9,*)'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
      ELSE
      WRITE(*,*)'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
      WRITE(9,*)'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
      ENDIF
      WRITE(*,*)'_____________________________________________________'
      WRITE(9,*)'_____________________________________________________'

      DO II=1,NOB
      FRANK1(II)=QIND1(II)
      FRANK2(II)=QIND2(II)
      ENDDO
      ENDIF
      FMIN=FMINRPS
C     -----------------------------------------------------------------
      DO J=1,M
      XX(K,J)=X(J)
      ENDDO
      KKF(K)=KF
      MM(K)=M
      FMINN(K)=FMIN
      WRITE(*,*)' '
      WRITE(*,*)' '
```

```
      WRITE(*,*)'----------------------- FINAL RESULTS==================='
      WRITE(*,*)'FUNCT CODE=',KKF(K),'  FMIN=',FMINN(K),' : DIM=',MM(K)
      WRITE(*,*)'OPTIMAL DECISION VARIABLES : ',METHOD(K)
      WRITE(*,*)'FOR THE FIRST SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
      WRITE(9,*)'FOR THE FIRST SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
      WRITE(9,*)(XX(K,J),J=1,MONE)
      WRITE(*,*)(XX(K,J),J=1,MONE)
      WRITE(*,*)'FOR THE SECOND SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
      WRITE(9,*)'FOR THE SECOND SET OF VARIABLES WEIGHTS ARE AS FOLLOWS'
      WRITE(9,*)(XX(K,J),J=MONE+1,M)
      WRITE(*,*)(XX(K,J),J=MONE+1,M)
      WRITE(*,*)'/////////////////////////////////////////////////////'
      WRITE(*,*)'OPTIMIZATION PROGRAM ENDED'
      WRITE(*,*)'*******************************************************'
      WRITE(*,*)'MEASURE OF EQUALITY/INEQUALITY'
      WRITE(*,*)'RPS: BEFORE AND AFTER OPTIMIZATION = ',Q0,Q1
      WRITE(*,*)' '
      WRITE(*,*)'RESULTS STORED IN FILE= ',OUTFILE
      WRITE(*,*)'OPEN BY MSWORD OR EDIT OR ANY OTHER EDITOR'
      WRITE(*,*)' '
      WRITE(*,*)'NOTE:VECTORS OF CORRELATIONS & INDEX(BOTH TOGETHER) ARE
     & IDETERMINATE FOR SIGN &   MAY BE MULTIPLED BY (-1) IF NEEDED'
      WRITE(*,*)'THAT IS IF R(J) IS TRANSFORMED TO -R(J) FOR ALL J THEN
     &THE INDEX(I) TOO IS         TRANSFORMED TO -INDEX(I) FOR ALL I'
      WRITE(9,*)' '
      WRITE(9,*)'NOTE: VECTORS OF CORRELATIONS AND INDEX (BOTH TOGETHER)
     & ARE IDETERMINATE FOR SIGN AND MAY BE MULTIPLED BY (-1) IF NEEDED'
      WRITE(9,*)'THAT IS IF R(J) IS TRANSFORMED TO -R(J) FOR ALL J THEN
     &THE INDEX(I) TOO IS TRANSFORMED TO -INDEX(I) FOR ALL I'
      CALL DORANK(FRANK1,NOB)
      CALL DORANK(FRANK2,NOB)
      DO I=1,NOB
      ZDAT(I,1)=FRANK1(I)
      ZDAT(I,2)=FRANK2(I)
      ENDDO
      IF(NCOR.EQ.0) THEN
      CALL CORREL(ZDAT,NOB,2,RMAT)
      ELSE
      CALL DOCORA(ZDAT,NOB,2,RMAT)
      ENDIF
      WRITE(9,*)'=================================================== '
      WRITE(*,*)'=================================================== '
      WRITE(9,*)'1ST 2 ARE CANONICAL SCORES AND LAST 2 ARE THEIR RANK'
      WRITE(*,*)'1ST 2 ARE CANONICAL SCORES AND LAST 2 ARE THEIR RANK'
      WRITE(9,*)'=================================================== '
      WRITE(*,*)'=================================================== '
      DO I=1,NOB
      IF(MRNK.EQ.1) THEN
      QIND1(I)=0.D0
      QIND2(I)=0.D0
      DO J=1,MONE
      QIND1(I)=QIND1(I)+ZDAT(I,J+1)*XX(NOPT,J)
      ENDDO
      DO J=MONE+1,MVAR
      QIND2(I)=QIND2(I)+ZDAT(I,J+1)*XX(NOPT,J)
      ENDDO
      ENDIF
      WRITE(9,2)I,QIND1(I),QIND2(I),(ZDAT(I,J),J=1,2)
      WRITE(*,2)I,QIND1(I),QIND2(I),(ZDAT(I,J),J=1,2)
      ENDDO
    2 FORMAT(I5,2F15.6,2F10.3)
      WRITE(9,*)'SQUARE OF CANONICAL CORRELATION =',RMAT(1,2)**2
      WRITE(*,*)'SQUARE OF CANONICAL CORRELATION =',RMAT(1,2)**2
      WRITE(9,*)'ABSOLUTE OF CANONICAL CORRELATION =',DABS(RMAT(1,2))
      WRITE(*,*)'ABSOLUTE OF CANONICAL CORRELATION =',DABS(RMAT(1,2))
      IF(NCOR.EQ.0) THEN
      WRITE(*,*)'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
```

```
      WRITE(9,*)'NOTE: THESE ARE KARL PEARSON TYPE CORRELATION (NCOR=0)'
      ELSE
      WRITE(*,*)'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
      WRITE(9,*)'NOTE: THESE ARE BRADLEY TYPE CORRELATION (NCOR=1)'
      ENDIF
      CLOSE(9)
      WRITE(*,*) 'THE JOB IS OVER'
      END
C     -----------------------------------------------------------------
      SUBROUTINE RPS(M,ABEST,FBEST,G1)
C     PROGRAM TO FIND GLOBAL MINIMUM BY REPULSIVE PARTICLE SWARM METHOD
C     WRITTEN BY SK MISHRA, DEPT. OF ECONOMICS, NEHU, SHILLONG (INDIA)
C     -----------------------------------------------------------------
      PARAMETER (N=50,NN=10,MX=100,NSTEP=7,ITRN=10000,NSIGMA=1,ITOP=1)
      PARAMETER (NPRN=50) ! DISPLAYS RESULTS AT EVERY 500 TH ITERATION
C     PARAMETER(N=50,NN=25,MX=100,NSTEP=9,ITRN=10000,NSIGMA=1,ITOP=3)
C     PARAMETER (N=100,NN=15,MX=100,NSTEP=9,ITRN=10000,NSIGMA=1,ITOP=3)
C     IN CERTAIN CASES THE ONE OR THE OTHER SPECIFICATION WORKS BETTER
C     DIFFERENT SPECIFICATIONS OF PARAMETERS MAY SUIT DIFFERENT TYPES
C     OF FUNCTIONS OR DIMENSIONS - ONE HAS TO DO SOME TRIAL AND ERROR
C     -----------------------------------------------------------------
C     N = POPULATION SIZE. IN MOST OF THE CASES N=30 IS OK. ITS VALUE
C     MAY BE INCREASED TO 50 OR 100 TOO. THE PARAMETER NN IS THE SIZE OF
C     RANDOMLY CHOSEN NEIGHBOURS. 15 TO 25 (BUT SUFFICIENTLY LESS THAN
C     N) IS A GOOD CHOICE. MX IS THE MAXIMAL SIZE OF DECISION VARIABLES.
C     IN F(X1, X2,...,XM) M SHOULD BE LESS THAN OR EQUAL TO MX. ITRN IS
C     THE NO. OF ITERATIONS. IT MAY DEPEND ON THE PROBLEM. 200(AT LEAST)
C     TO 500 ITERATIONS MAY BE GOOD ENOUGH. BUT FOR FUNCTIONS LIKE
C     ROSENBROCKOR GRIEWANK OF LARGE SIZE (SAY M=30) IT IS NEEDED THAT
C     ITRN IS LARGE, SAY 5000 OR EVEN 10000.
C     SIGMA INTRODUCES PERTURBATION & HELPS THE SEARCH JUMP OUT OF LOCAL
C     OPTIMA. FOR EXAMPLE : RASTRIGIN FUNCTION OF DMENSION 3O OR LARGER
C     NSTEP DOES LOCAL SEARCH BY TUNNELLING AND WORKS WELL BETWEEN 5 AND
C     15, WHICH IS MUCH ON THE HIGHER SIDE.
C     ITOP <=1 (RING); ITOP=2 (RING AND RANDOM); ITOP=>3 (RANDOM)
C     NSIGMA=0 (NO CHAOTIC PERTURBATION);NSIGMA=1 (CHAOTIC PERTURBATION)
C     NOTE THAT NSIGMA=1 NEED NOT ALWAYS WORK BETTER (OR WORSE)
C     SUBROUTINE FUNC( ) DEFINES OR CALLS THE FUNCTION TO BE OPTIMIZED.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /RNDM/IU,IV
      COMMON /KFF/KF,NFCALL,FTIT
      INTEGER IU,IV
      CHARACTER *70 FTIT
      DIMENSION X(N,MX),V(N,MX),A(MX),VI(MX),TIT(50),ABEST(*)
      DIMENSION XX(N,MX),F(N),V1(MX),V2(MX),V3(MX),V4(MX),BST(MX)
C     A1 A2 AND A3 ARE CONSTANTS AND W IS THE INERTIA WEIGHT.
C     OCCASIONALLY, TINKERING WITH THESE VALUES, ESPECIALLY A3, MAY BE
C     NEEDED.
      DATA A1,A2,A3,W,SIGMA /.5D00,.5D00,.0005D00,.5D00,1.D-03/
      EPSILON=1.D-12 ! ACCURACY NEEDED FOR TERMINATON
C     --------------------CHOOSING THE TEST FUNCTION -------------------'
      CALL FSELECT(KF,M,FTIT)
C     -----------------------------------------------------------------
      FFMIN=1.D30
      LCOUNT=0
      NFCALL=0
      WRITE(*,*)'4-DIGITS SEED FOR RANDOM NUMBER GENERATION'
      READ(*,*) IU
      DATA FMIN /1.0E30/
C     GENERATE N-SIZE POPULATION OF M-TUPLE PARAMETERS X(I,J) RANDOMLY
      DO I=1,N
        DO J=1,M
        CALL RANDOM(RAND)
         X(I,J)=RAND
C     WE GENERATE RANDOM(-5,5). HERE MULTIPLIER IS 10. TINKERING IN SOME
C     CASES MAY BE NEEDED
         ENDDO
```

```
        F(I)=1.0D30
      ENDDO
C     INITIALISE VELOCITIES V(I) FOR EACH INDIVIDUAL IN THE POPULATION
      DO I=1,N
      DO J=1,M
      CALL RANDOM(RAND)
       V(I,J)=(RAND-0.5D+00)
C       V(I,J)=RAND
      ENDDO
      ENDDO
      DO 100 ITER=1,ITRN
C     WRITE(*,*)'ITERATION=',ITER
C     LET EACH INDIVIDUAL SEARCH FOR THE BEST IN ITS NEIGHBOURHOOD
        DO I=1,N
           DO J=1,M
           A(J)=X(I,J)
           VI(J)=V(I,J)
           ENDDO
           CALL LSRCH(A,M,VI,NSTEP,FI)
           IF(FI.LT.F(I)) THEN
            F(I)=FI
            DO IN=1,M
            BST(IN)=A(IN)
            ENDDO
C     F(I) CONTAINS THE LOCAL BEST VALUE OF FUNCTION FOR ITH INDIVIDUAL
C     XX(I,J) IS THE M-TUPLE VALUE OF X ASSOCIATED WITH LOCAL BEST F(I)
             DO J=1,M
             XX(I,J)=A(J)
             ENDDO
             ENDIF
        ENDDO
C      NOW LET EVERY INDIVIDUAL RANDOMLY COSULT NN(<<N) COLLEAGUES AND
C      FIND THE BEST AMONG THEM
      DO I=1,N
C     ------------------------------------------------------------------
      IF(ITOP.GE.3) THEN
C     RANDOM TOPOLOGY ******************************************
C     CHOOSE NN COLLEAGUES RANDOMLY AND FIND THE BEST AMONG THEM
          BEST=1.0D30
           DO II=1,NN
                 CALL RANDOM(RAND)
               NF=INT(RAND*N)+1
                IF(BEST.GT.F(NF)) THEN
                 BEST=F(NF)
                NFBEST=NF
                 ENDIF
            ENDDO
      ENDIF
C-----------------------------------------------------------------------
      IF(ITOP.EQ.2) THEN
C     RING + RANDOM TOPOLOGY ******************************************
C     REQUIRES THAT THE SUBROUTINE NEIGHBOR IS TURNED ALIVE
       BEST=1.0D30
          CALL NEIGHBOR(I,N,I1,I3)
          DO II=1,NN
                IF(II.EQ.1) NF=I1
                 IF(II.EQ.2) NF=I
                  IF(II.EQ.3) NF=I3
                      IF(II.GT.3) THEN
                     CALL RANDOM(RAND)
                      NF=INT(RAND*N)+1
                     ENDIF
                  IF(BEST.GT.F(NF)) THEN
                  BEST=F(NF)
                  NFBEST=NF
                 ENDIF
             ENDDO
```

```
       ENDIF
C---------------------------------------------------------------------
      IF(ITOP.LE.1) THEN
C     RING TOPOLOGY **************************************************
C     REQUIRES THAT THE SUBROUTINE NEIGHBOR IS TURNED ALIVE
        BEST=1.0D30
           CALL NEIGHBOR(I,N,I1,I3)
              DO II=1,3
              IF (II.NE.I) THEN
             IF(II.EQ.1) NF=I1
              IF(II.EQ.3) NF=I3
                  IF(BEST.GT.F(NF)) THEN
                   BEST=F(NF)
                   NFBEST=NF
                  ENDIF
                  ENDIF
            ENDDO
       ENDIF
C---------------------------------------------------------------------
C     IN THE LIGHT OF HIS OWN AND HIS BEST COLLEAGUES EXPERIENCE, THE
C     INDIVIDUAL I WILL MODIFY HIS MOVE AS PER THE FOLLOWING CRITERION
C     FIRST, ADJUSTMENT BASED ON ONES OWN EXPERIENCE
C     AND OWN BEST EXPERIENCE IN THE PAST (XX(I))
           DO J=1,M
           CALL RANDOM(RAND)
           V1(J)=A1*RAND*(XX(I,J)-X(I,J))

C     THEN BASED ON THE OTHER COLLEAGUES BEST EXPERIENCE WITH WEIGHT W
C     HERE W IS CALLED AN INERTIA WEIGHT 0.01< W < 0.7
C     A2 IS THE CONSTANT NEAR BUT LESS THAN UNITY
           CALL RANDOM(RAND)
           V2(J)=V(I,J)
           IF(F(NFBEST).LT.F(I)) THEN
           V2(J)=A2*W*RAND*(XX(NFBEST,J)-X(I,J))
           ENDIF
C     THEN SOME RANDOMNESS AND A CONSTANT A3 CLOSE TO BUT LESS THAN UNITY
           CALL RANDOM(RAND)
           RND1=RAND
           CALL RANDOM(RAND)
            V3(J)=A3*RAND*W*RND1
C            V3(J)=A3*RAND*W
C     THEN ON PAST VELOCITY WITH INERTIA WEIGHT W
           V4(J)=W*V(I,J)
C     FINALLY A SUM OF THEM
           V(I,J)= V1(J)+V2(J)+V3(J)+V4(J)
           ENDDO
      ENDDO
C     CHANGE X
      DO I=1,N
      DO J=1,M
      RANDS=0.D00
C     ------------------------------------------------------------------
      IF(NSIGMA.EQ.1) THEN
       CALL RANDOM(RAND) ! FOR CHAOTIC PERTURBATION
       IF(DABS(RAND-.5D00).LT.SIGMA) RANDS=RAND-0.5D00
C     SIGMA CONDITIONED RANDS INTRODUCES CHAOTIC ELEMENT IN TO LOCATION
C     IN SOME CASES THIS PERTURBATION HAS WORKED VERY EFFECTIVELY WITH
C     PARAMETER (N=100,NN=15,MX=100,NSTEP=9,ITRN=100000,NSIGMA=1,ITOP=2)
      ENDIF
C     -----------------------------------------------------------------
      X(I,J)=X(I,J)+V(I,J)*(1.D00+RANDS)
      ENDDO
      ENDDO
       DO I=1,N
         IF(F(I).LT.FMIN) THEN
         FMIN=F(I)
         II=I
```

```fortran
         DO J=1,M
         BST(J)=XX(II,J)
         ENDDO
         ENDIF
         ENDDO

      IF(LCOUNT.EQ.NPRN) THEN
      LCOUNT=0
      WRITE(*,*)'OPTIMAL SOLUTION UPTO THIS (FUNCTION CALLS=',NFCALL,')'
      WRITE(*,*)'X = ',(BST(J),J=1,M),' MIN F = ',FMIN
C      WRITE(*,*)'NO. OF FUNCTION CALLS = ',NFCALL
      DO J=1,M
      ABEST(J)=BST(J)
      ENDDO
      IF(DABS(FFMIN-FMIN).LT.EPSILON) GOTO 999
      FFMIN=FMIN
      ENDIF
      LCOUNT=LCOUNT+1
  100 CONTINUE
  999 WRITE(*,*)'----------------------------------------------------'
      DO I=1,N
      IF(F(I).LT.FBEST) THEN
      FBEST=F(I)
      DO J=1,M
      ABEST(J)=XX(I,J)
      ENDDO
      ENDIF
      ENDDO
      CALL FUNC(ABEST,M,FBEST)
      CALL GINI(F,N,G1)
      WRITE(*,*)'FINAL X = ',(BST(J),J=1,M),' FINAL MIN F = ',FMIN
      WRITE(*,*)'COMPUTATION OVER:FOR ',FTIT
      WRITE(*,*)'NO. OF VARIABLES=',M,'  END.'
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE LSRCH(A,M,VI,NSTEP,FI)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      CHARACTER *70 FTIT
      COMMON /KFF/KF,NFCALL,FTIT
      COMMON /RNDM/IU,IV
      INTEGER IU,IV
      DIMENSION A(*),B(100),VI(*)
      AMN=1.0D30
      DO J=1,NSTEP
         DO JJ=1,M
         B(JJ)=A(JJ)+(J-(NSTEP/2)-1)*VI(JJ)
         ENDDO
      CALL FUNC(B,M,FI)
        IF(FI.LT.AMN) THEN
        AMN=FI
        DO JJ=1,M
        A(JJ)=B(JJ)
        ENDDO
        ENDIF
      ENDDO
      FI=AMN
      RETURN
      END
C     -----------------------------------------------------------------
C     THIS SUBROUTINE IS NEEDED IF THE NEIGHBOURHOOD HAS RING TOPOLOGY
C     EITHER PURE OR HYBRIDIZED
       SUBROUTINE NEIGHBOR(I,N,J,K)
       IF(I-1.GE.1 .AND. I.LT.N) THEN
       J=I-1
       K=I+1
       ELSE
```

```
       IF(I-1.LT.1) THEN
       J=N-I+1
       K=I+1
       ENDIF
       IF(I.EQ.N) THEN
       J=I-1
       K=1
       ENDIF
       ENDIF
       RETURN
       END
C     ------------------------------------------------------------------
C     RANDOM NUMBER GENERATOR (UNIFORM BETWEEN 0 AND 1 - BOTH EXCLUSIVE)
      SUBROUTINE RANDOM(RAND1)
       DOUBLE PRECISION  RAND1
       COMMON /RNDM/IU,IV
      INTEGER IU,IV
       IV=IU*65539
       IF(IV.LT.0) THEN
       IV=IV+2147483647+1
       ENDIF
       RAND=IV
       IU=IV
       RAND=RAND*0.4656613E-09
       RAND1= DBLE(RAND)
       RETURN
       END
C     ------------------------------------------------------------------
      SUBROUTINE GINI(F,N,G)
      PARAMETER (K=1) !K=1 GINI COEFFICENT; K=2 COEFFICIENT OF VARIATION
C     THIS PROGRAM COMPUTES MEASURE OF INEQUALITY
C     IF K =1 GET THE GINI COEFFICIENT. IF K=2 GET COEFF OF VARIATIONE
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION F(*)
      S=0.D0
      DO I=1,N
      S=S+F(I)
      ENDDO
      S=S/N
      H=0.D00
      DO I=1,N-1
      DO J=I+1,N
      H=H+(DABS(F(I)-F(J)))**K
      ENDDO
      ENDDO
      H=(H/(N**2))**(1.D0/K)! FOR K=1 H IS MEAN DEVIATION;
C                              FOR K=2 H IS STANDARD DEVIATION
      WRITE(*,*)'MEASURES OF DISPERSION AND CENTRAL TENDENCY = ',G,S
      G=DEXP(-H)! G IS THE MEASURE OF EQUALITY (NOT GINI OR CV)
C     G=H/DABS(S) !IF S NOT ZERO, K=1 THEN G=GINI, K=2 G=COEFF VARIATION
      RETURN
      END
C     ------------------------------------------------------------------
      SUBROUTINE FSELECT(KF,M,FTIT)
      COMMON /CANON/MONE,MTWO
C      THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING ------
C     (1) FUNCTION CODE (KF), (2) NO. OF VARIABLES IN THE FUNCTION (M);
      CHARACTER *70 TIT(100),FTIT
      NFN=1
      KF=1
      WRITE(*,*)'-----------------------------------------------------'
      DATA TIT(1)/'COMPUTE CANONICAL CORRELATION FROM 2 DATA SETS'/
C     ------------------------------------------------------------------
      DO I=1,NFN
      WRITE(*,*)TIT(I)
      ENDDO
      WRITE(*,*)'-----------------------------------------------------'
```

```
      WRITE(*,*)'SPECIFY NO. OF VARIABLES IN SET-1[=M1] AND SET-2[=M2]'
      READ(*,*) MONE, MTWO
      M=MONE+MTWO
      FTIT=TIT(KF) ! STORE THE NAME OF THE CHOSEN FUNCTION IN FTIT
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE FUNC(X,M,F)
C     TEST FUNCTIONS FOR GLOBAL OPTIMIZATION PROGRAM
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /RNDM/IU,IV
      COMMON /KFF/KF,NFCALL,FTIT
      INTEGER IU,IV
      DIMENSION X(*)
      CHARACTER *70 FTIT
      NFCALL=NFCALL+1 ! INCREMENT TO NUMBER OF FUNCTION CALLS
C     KF IS THE CODE OF THE TEST FUNCTION
      IF(KF.EQ.1) THEN
      CALL CORD(M,X,F)
      RETURN
      ENDIF
C     =================================================================
      WRITE(*,*)'FUNCTION NOT DEFINED. PROGRAM ABORTED'
      STOP
      END
C     ------------------------------------------------------------------
      SUBROUTINE CORD(M,X,F)
      PARAMETER (NOB=30,MVAR=9)! CHANGE THE PARAMETERS HERE AS NEEDED.
C     ------------------------------------------------------------------
C     NOB=NO. OF OBSERVATIONS (CASES) & MVAR= NO. OF VARIABLES
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      COMMON /CANON/MONE,MTWO
      COMMON /RNDM/IU,IV
      COMMON /CORDAT/CDAT(NOB,MVAR),QIND1(NOB),QIND2(NOB),R(1),NORM,NCOR
      COMMON /GETRANK/MRNK
      INTEGER IU,IV
      DIMENSION X(*),Z(NOB,2)
      DO I=1,M
      IF(X(I).LT.-1.0D0.OR.X(I).GT.1.0D0) THEN
      CALL RANDOM(RAND)
      X(I)=(RAND-0.5D0)*2
      ENDIF
      ENDDO
      XNORM=0.D0
      DO J=1,M
      XNORM=XNORM+X(J)**2
      ENDDO
      XNORM=DSQRT(XNORM)
      DO J=1,M
      X(J)=X(J)/XNORM
      ENDDO
C     CONSTRUCT INDEX
      DO I=1,NOB
      QIND1(I)=0.D0
      QIND2(I)=0.D0
      DO J=1,MONE
      QIND1(I)=QIND1(I)+CDAT(I,J)*X(J)
      ENDDO
      DO J=MONE+1,M
      QIND2(I)=QIND2(I)+CDAT(I,J)*X(J)
      ENDDO
      ENDDO

C     ------------------------------------------------------------------
C      !FIND THE RANK OF QIND
      IF(MRNK.EQ.1) THEN
      CALL DORANK(QIND1,NOB)
```

```
      CALL DORANK(QIND2,NOB)
      ENDIF
C     ------------------------------------------------------------------
C     COMPUTE CORRELATIONS
      DO I=1,NOB
      Z(I,1)=QIND1(I)
      Z(I,2)=QIND2(I)
      ENDDO

      IF(NCOR.EQ.0) THEN
      CALL CORLN(Z,NOB,RHO)
      ELSE
      CALL CORA(Z,NOB,RHO)
      ENDIF
      R(1)=RHO
      F= DABS(R(1))**NORM
C     ------------------------------------------------------------------
      F=-F
      RETURN
      END
      SUBROUTINE CORLN(Z,NOB,RHO)
C     NOB = NO. OF CASES
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Z(NOB,2),AV(2),SD(2)
      DO J=1,2
      AV(J)=0.D0
      SD(J)=0.D0
      DO I=1,NOB
      AV(J)=AV(J)+Z(I,J)
      SD(J)=SD(J)+Z(I,J)**2
      ENDDO
      ENDDO
      DO J=1,2
      AV(J)=AV(J)/NOB
      SD(J)=DSQRT(SD(J)/NOB-AV(J)**2)
      ENDDO
C      WRITE(*,*)'AV AND SD ', AV(1),AV(2),SD(1),SD(2)
      RHO=0.D0
      DO I=1,NOB
      RHO=RHO+(Z(I,1)-AV(1))*(Z(I,2)-AV(2))
      ENDDO
      RHO=(RHO/NOB)/(SD(1)*SD(2))
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE CORA(Z,N,R)
C     COMPUTING BRADLEY'S ABSOLUTE CORRELATION MATRIX
C     BRADLEY, C. (1985) "THE ABSOLUTE CORRELATION", THE MATHEMATICAL
C     GAZETTE, 69(447): 12-17.
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION Z(N,2),X(N),Y(N)
C     ------------------------------------------------------------------
C     PUT Z INTO X AND Y
      DO I=1,N
      X(I)=Z(I,1)
      Y(I)=Z(I,2)
      ENDDO
C     ARRANGE X ANY IN AN ASCENDING ORDER
      DO I=1,N-1
      DO II=I+1,N
      IF(X(I).GT.X(II)) THEN
      TEMP=X(I)
      X(I)=X(II)
      X(II)=TEMP
      ENDIF
      IF(Y(I).GT.Y(II)) THEN
      TEMP=Y(I)
```

```fortran
      Y(I)=Y(II)
      Y(II)=TEMP
      ENDIF
      ENDDO
      ENDDO
C     FIND MEDIAN
      IF(INT(N/2).EQ.N/2.D0) THEN
      XMED=(X(N/2)+X(N/2+1))/2.D0
      YMED=(Y(N/2)+Y(N/2+1))/2.D0
      ENDIF
      IF(INT(N/2).NE.N/2.D0) THEN
      XMED=X(N/2+1)
      YMED=Y(N/2+1)
      ENDIF
C     SUBTRACT RESPECTIVE MEDIANS FROM X AND Y AND FIND ABS DEVIATIONS
      VX=0.D0
      VY=0.D0
      DO I=1,N
      X(I)=X(I)-XMED
      Y(I)=Y(I)-YMED
      VX=VX+DABS(X(I))
      VY=VY+DABS(Y(I))
      ENDDO
C     SCALE THE VARIABLES X AND Y SUCH THAT VX=VY
      IF(VX.EQ.0.D0.OR.VY.EQ.0.D0) THEN
      R=0.D0
      RETURN
      ENDIF
      DO I=1,N
      X(I)=X(I)*VY/VX
      ENDDO
C     COMPUTE CORRELATION COEFFICIENT
      VZ=0.D0
      R=0.D0
      DO I=1,N
      VZ=VZ+DABS(X(I))+DABS(Y(I))
      R=R+DABS(X(I)+Y(I))-DABS(X(I)-Y(I))
      ENDDO
      R=R/VZ
      RETURN
      END
C     -----------------------------------------------------------------
      SUBROUTINE DORANK(X,N)! N IS THE NUMBER OF OBSERVATIONS
      PARAMETER (NRL=0) ! THIS VALUE IS TO BE SET BY THE USER
C                       !THE VALUE OF NRL DECIDES THE SCHEME OF RANKINGS
C     !THIS PROGRAM RETURNS RANK-ORDER OF A GIVEN VECTOR
      PARAMETER (MXD=1000)! MXD IS MAX DIMENSION FOR TEMPORARY VARIABLES
      ! THAT ARE LOCAL AND DO NOT GO TO THE INVOKING PROGRAM
      ! X IS THE VARIABLE TO BE SUBSTITUTED BY ITS RANK VALUES
C     NRULE=0 FOR ORDINAL RANKING (1-2-3-4 RULE);
C     NRULE=1 FOR DENSE RANKING (1-2-2-3 RULE);
C     NRULE=2 FOR STANDARD COMPETITION RANKING (1-2-2-4 RULE);
C     NRULE=3 FOR MODIFIED COMPETITION RANKING (1-3-3-4 RULE);
C     NRULE=4 FOR FRACTIONAL RANKING (1-2.5-2.5-4 RULE);
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(N),NF(MXD),NCF(MXD),RANK(MXD),ID(MXD),XX(MXD)
C     GENERATE ID(I),I=1,N
      DO I=1,N
      ID(I)=I
      NF(I)=0
      ENDDO
C     ARRANGE DATA (X) AND THE IDS IN ASCENDING ORDER
      DO I=1,N-1
      DO II=I,N
      IF(X(II).LT.X(I)) THEN
      TEMP=X(I)
      X(I)=X(II)
```

```
      X(II)=TEMP
      ITEMP=ID(I)
      ID(I)=ID(II)
      ID(II)=ITEMP
      ENDIF
      ENDDO
      ENDDO
C     MAKE DISCRETE UNGROUPED FREQUENCY TABLE
      K=0
      J=1
    1 K=K+1
      XX(K)=X(J)
      NF(K)=0
      DO I=J,N
      IF(XX(K).EQ.X(I)) THEN
      NF(K)=NF(K)+1
      ELSE
      J=I
      IF(J.LE.N) THEN
      GOTO 1
      ELSE
      GOTO 2
      ENDIF
      ENDIF
      ENDDO
    2 KK=K
      DO K=1,KK
      IF(K.EQ.1) THEN
      NCF(K)=NF(K)
      ELSE
      NCF(K)=NCF(K-1)+NF(K)
      ENDIF
      ENDDO
      DO I=1,N
      RANK(I)=1.D0
      ENDDO

      IF(NRL.GT.4) THEN
      WRITE(*,*)'RANKING RULE CODE GREATER THAN 4 NOT PERMITTED',NRL
      STOP
      ENDIF

      IF(NRL.LT.0) THEN
      WRITE(*,*)'RANKING RULE CODE LESS THAN 0 NOT PERMITTED',NRL
      STOP
      ENDIF

      IF(NRL.EQ.0) THEN
      DO I=1,N
      RANK(I)=I
      ENDDO
      ENDIF
C     -----------------------------------------------------------------
      IF(NRL.GT.0) THEN
       DO K=1,KK
       IF(K.EQ.1) THEN
       K1=1
       ELSE
       K1=NCF(K-1)+1
       ENDIF
       K2=NCF(K)
       DO I=K1,K2
       SUM=0.D0
       DO II=K1,K2
       SUM=SUM+II
       ENDDO
       KX=(K2-K1+1)
```

```
        IF(NRL.EQ.1)RANK(I)=K ! DENSE RANKING (1-2-2-3 RULE)
        IF(NRL.EQ.2)RANK(I)=K1!STANDARD COMPETITION RANKING(1-2-2-4 RULE)
        IF(NRL.EQ.3)RANK(I)=K2!MODIFIED COMPETITION RANKING(1-3-3-4 RULE)
        IF(NRL.EQ.4)RANK(I)=SUM/KX !FRACTIONAL RANKING (1-2.5-2.5-4 RULE)
        ENDDO
        ENDDO
       ENDIF
C      ---------------------------------------------------------------
       DO I=1,N
       X(ID(I))=RANK(I) ! BRINGS THE DATA TO ORIGINAL SEQUENCE
       ENDDO
       RETURN
       END
C      --------------------------------------------------------------
       SUBROUTINE CORREL(X,N,M,RMAT)
       PARAMETER (NMX=30)!DO NOT CHANGE UNLESS NO. OF VARIABLES EXCEED 30
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION X(N,M),RMAT(2,2),AV(NMX),SD(NMX)
       DO J=1,2
       AV(J)=0.D0
       SD(J)=0.D0
       DO I=1,N
       AV(J)=AV(J)+X(I,J)
       SD(J)=SD(J)+X(I,J)**2
       ENDDO
       AV(J)=AV(J)/N
       SD(J)=DSQRT(SD(J)/N-AV(J)**2)
       ENDDO
       DO J=1,2
       DO JJ=1,2
       RMAT(J,JJ)=0.D0
       DO I=1,N
       RMAT(J,JJ)=RMAT(J,JJ)+X(I,J)*X(I,JJ)
       ENDDO
       ENDDO
       ENDDO
       DO J=1,2
       DO JJ=1,2
       RMAT(J,JJ)=RMAT(J,JJ)/N-AV(J)*AV(JJ)
       RMAT(J,JJ)=RMAT(J,JJ)/(SD(J)*SD(JJ))
       ENDDO
       ENDDO
       RETURN
       END
C      ---------------------------------------------------------------
       SUBROUTINE DOCORA(ZDAT,N,M,RMAT)
       IMPLICIT DOUBLE PRECISION (A-H,O-Z)
       DIMENSION ZDAT(N,M),RMAT(2,2),Z(N,2)
       DO I=1,N
       Z(I,1)=ZDAT(I,1)
       Z(I,2)=ZDAT(I,2)
       ENDDO
       CALL CORA(Z,N,R)
       RMAT(1,2)=R
       RMAT(2,1)=R
       DO J=1,2
       RMAT(J,J)=1.D0
       ENDDO
       RETURN
       END
```