Journal of Economics Bibliography
www.kspjournals.org
Volume 3 September 2016 Issue 3

Shapley Value Regression and the Resolution of Multicollinearity

By Sudhanshu K. MISHRA †

Abstract. Multicollinearity in empirical data violates the assumption of independence among the regressors in a linear regression model that often leads to failure in rejecting a false null hypothesis. It also may assign wrong sign to coefficients. Shapley value regression is perhaps the best methods to combat this problem. The present paper simplifies the algorithm of Shapley value decomposition of R^2 and develops a Fortran computer program that executes it. It also retrieve regression coefficients from the Shapley value. However, Shapley value regression becomes increasingly impracticable as the number of regressor variables exceeds 10, although, in practice, a good regression model may not have more than ten regressors..
Keywords. Multicollinearity, Shapley value, regression, computer program, Fortran.
JEL. C63, C71.

1. Introduction

In the econometric literature multicollinearity is defined as the incidence of high degree of correlation among some or all regressor variables. Strong multicollinearity has deleterious effects on the confidence intervals of linear regression coefficients (β in the linear regression model $y=X\beta+u$). Although it does not affect the explanatory power (R^2) of the regressors or unbiasedness of the estimated coefficients associated with them, it does inflate their standard error of estimate rendering test of hypothesis misleading or paradoxical, often such that although R^2 could be very high, individual coefficients may all have poor Student's t-values. Thus, strong multicollinearity may lead to failure in rejecting a false null hypothesis of ineffectiveness of the regressor variable to the regressand variable (type II error). Very frequently, it also affects the sign of the regression coefficients. However, it has been pointed out that the incidence of high degree of correlation (measured in terms of a large condition number; Belsley et al., 1980) among some or all regressor variables alone (unsupported by large variance of error in the regressand variable, y) has little effect on the precision of regression coefficients. Large condition number coupled with a large variance of error in the regressand variable destabilizes the regression estimator; either of the two in isolation cannot cause much harm, although the condition number is relatively more potent in determining the stability of estimated regression coefficients (Mishra, 2004a).

† North-Eastern Hill University, Shillong, 91 - 793022, New Delhi, India.
☎. 918130397754
✉. mishrasknehu@hotmail.com

2. Statistical resolution of multicollinearity problem

Some econometricians suggest that the problem of multicollinearity is a data problem – the data do not contain enough information to disentangle the effects of individual regressor variables on the regressand variable – and therefore, the solution of the problem lay in getting more data with enough variability. However, since it is often hard or impossible to obtain more informative data, and the incidence of a high degree of multicollinearity is frequently met with in empirical studies rendering the estimated regression coefficients either unreliable or misleading (sign-wise) or both, the resolution of multicollinearty problem has attracted intense efforts leading to development of many methods (mostly statistical in nature) that appear to contain the problem with the given data set. A compact but nearly exhaustive survey of such methods (e.g. (i) ordinary ridge regression, (ii) restricted ridge regression, (iii) the family of Liu estimators, (iv) restricted Liu estimators, (v) generalized maximum entropy estimator, (vi) maximum entropy Leuven estimators, (vii) modular max entropy Leuven estimator, etc.) is available (Mishra, 2004b). Works of Wu (2009), Macedo et al. (2010), Chen (2012), Özkale (2012), York (2012), Özkale (2014), Ročková & George (2014), Huang et al. (2015), Gómez et al. (2016) are some recent efforts in this direction. These methods modify the method of statistical estimation of the regression model and a few of them need some information from the analyst.

3. Shapley value regression

This is an entirely different strategy to assess the contribution of regressor variables to the regressand variable. It owes its origin in the theory of cooperative games (Shapley, 1953). The value of R^2 obtained by fitting a linear regression model y=Xβ+u is considered as the value of a cooperative game played by X (whose members, $x_j \epsilon X$; j=1, m, work in a coalition) against y (explaining it). The analyst does not have enough information to disentangle the contributions made by the individual members $x_j \epsilon X$; j=1, m, but only their joint contribution (R^2) is known. The Shapley value decomposition imputes the most likely contribution of each individual $x_j \epsilon X$; j=1, m, to R^2.

4. An algorithm to impute the contribution of individual variables to Shapley value

Let there be m number of regressor variables in the model y=Xβ+u. Let X(p, r) be the r-membered subset of X in which the p^{th}regressor appears and X(q, r) be the r-membered subset of X in which the p^{th}regressor <u>does not</u> appear. Further, let R^2(p, r) be the R^2 obtained by regression of y on X(p, r) and R^2(q, r) be the R^2 obtained by regression of y on X(q, r). Then, the share of the regressor variable p (that is x_p ϵ X) is given by $S(p)=(1/m)\sum_{r=1}^{m}\left\{\sum_{c=1}^{k}[\mathrm{R}^2(p,r)-R^2(q,r-1)]_c\right\}/\mathrm{k}.$

Moreover, $\mathrm{R}^2(q,0)=0$. Here k is the number of cases in which the evaluation in [.] was carried out. The sum of all S(p) for p=1, m (that is, $\sum_{p=1}^{m}S(p)$) is the R^2 of y=Xβ+u : (all x_j ϵ X) or the total value of the game = R^2 = $\sum_{p=1}^{m}S(p)=\sum_{p=1}^{m}(1/m)\sum_{r=1}^{m}\left\{\sum_{c=1}^{k}[\mathrm{R}^2(p,r)-R^2(q,r-1)]_c\right\}/\mathrm{k}.$

5. Retrieval of regression coefficients from Shapley value

As explained by Lipovetsky (2006), we may retrieves standardized regression coefficients, denoted by, say, α. Denoting pair-wise correlation matrix among regressors by S, pair-wise correlation vector between regressand and regressors by T and Shapley value vector by V, we formulate a quadratic programming problem to minimize $f(\alpha)$ given by

$$f(\alpha)=\sum_{j=1}^{m}(\alpha_j(2T-S\alpha)_j-V_j)^2.$$

Optimization may be done by any suitable method. Further, regular or non-standardized (unit and scale retaining) regression coefficients may be obtained by $\beta_j=\alpha_j(\sigma(y)/\sigma(x_j))$, where $\sigma(y)$ and $\sigma(x_j)$ are standard deviations of y and x_j, respectively. Further, the regression constant may be computed by the relationship $\beta_0=\bar{y}-\sum_{j=1}^{m}\beta_j\bar{x}_j$ in which $\bar{y}$ and $\bar{x}_j$ are arithmetic mean of y and x_j,respectively.

6. A numerical Example to impute Shapley value

Table-1 presents a dataset (available in Arumairajan & Wijekoon, 2013) of y and four regressor variables x_1 through x_4. The regression model is y=Xβ+u. Since the total of percentage weight of different chemicals in the composition is close to 100 for each observation (replicate), the regression model based on this dataset would suffer from a high multicollinearity problem.

The ordinary least squares estimation of the model y=Xβ+u provides:

$$\hat{Y}=\ 62.405\ +\ 1.551\,x_1\ +\ 0.510\,x_2\ +\ 0.102\,x_3\ -\ 0.144\,x_4\ ;R^2=0.982$$

The details of regression analysis on this dataset are presented in Table-2. As it is indicated in Table-2, in spite of a large value of R^2(0.982), the regression coefficients associated with x_2, x_3 and x_4 are statistically insignificant (due to inflated standard error of estimate). The regression coefficient associated with x_1 is poorly significant (different from zero) at 7.1% level. Interestingly, the regression coefficient associated with x_4 is negative (although statistically insignificant). A large value of R^2 suggests that the chemicals in the composition of cement explain the evolution of heat in setting, but none of the chemicals, individually, contribute significantly to the said evolution of heat. This is paradoxical. Also, the negative contribution of x_4 (although statistically insignificant), goes against the chemical science. This example amply suggests that strong multicollinearity mars the scientific validity of the findings of regression analysis.

Table1. *The Portland Cement Dataset (cf. Woods, Steinour and Starke, 1932)*

sl	1	2	3	4	5	6	7	8	9	10	11	12	13
y	78.5	74.3	104.3	87.6	95.9	109.2	102.7	72.5	93.1	115.9	83.8	113.3	109.4
x_1	7	1	11	11	7	11	3	1	2	21	1	11	10
x_2	26	29	56	31	52	55	71	31	54	47	40	66	68
x_3	6	15	8	8	6	9	17	22	18	4	23	9	8
x_4	60	52	20	47	33	22	6	44	22	26	34	12	12
Total	99	97	95	97	98	97	97	98	96	98	98	98	98

Variables: x_1 = 3CaO.Al2O3 (% weight of tricalcium aluminate); x_2 = 3CaO.SiO2 (% weight of tricalcium silictate); x_3 = 4CaO.Al2O3.Fe2O3 (% weight of tetracalcium alumino ferrite); x_4 = 2CaO.SiO2 (% weight of beta-dicalcium silictate); y = heat (calories per gram of cement) evolved while the sample was setting for 180 days of curing. Total is the sum of x_1 through x_4.

Table2. *Summary of Regression Analysis of Portland Cement Dataset*

Regressors	β	Std Error	Std. coeff.	t-value	Signif.
Constant	62.405	70.071	-	0.891	0.399
x_1	1.551	0.745	0.607	2.083	0.071
x_2	0.510	0.724	0.528	0.705	0.501
x_3	0.102	0.755	0.043	0.135	0.896
x_4	-0.144	0.709	-0.160	-0.203	0.844

7. Can partial correlation ameliorate the multicollinearity problem?

Partial correlation between two variables y and x_p measures the correlation between the residuals u_{yz} ($u_{yz} = y - Z\hat{\alpha}$ obtained from the model y = Zα + u_{yz}) and u_{pz} ($u_{pz} = x_p - Z\hat{\gamma}$ obtained from the model x_p = Zγ + u_{pz}), where Z = X(q, m-1) ϵ X or Z is the subset of X in which x_p is not there. Thus, the partial correlation coefficient r(u_{yz} , u_{pz}) measures the correlation between y and x_p (both of which are net of the effects of Z). From the dataset under investigation we obtained the partial correlation coefficients, presented in Table-3. A perusal of Table-3 reveals that x_3 and x_4 continue to show poor effectiveness to y and the correlation between y and x_4 continues to be negative.

Table 3. *Partial Correlation Coefficients of Regressors in the Portland Cement Dataset*

Regressor	x_1	x_2	x_3	x_4
Partial r	0.592932583580143	0.241809386003352	0.047686489890471	-0.071648285536613
Partial r^2	0.351569048671023	0.058471779159318	0.002274001318074	0.005133476820336

8. Decomposition of Shapley value of regression analysis

The coalition (regressor variables, X) gains 0.9823756204 (which is equal to the R^2 of the regression model y=Xβ+u). Individual members of the coalition share this gain as detailed out in Table-4.

Table 4. *Share of Individual Regressors and the Shapley Value of the Cooperative Game*

Regressor	x_1	x_2	x_3	x_4	Total
Share	0.2488891693	0.2912502074	0.1348865135	0.3073497303	0.9823756204
Share(%)	25.3354	29.6475	13.7306	31.2864	100.00

9. Computational illustration

It will be worthwhile to demonstrate how the values in Table-4 were obtained. Table-5 details out the computation for the share of x_1. For (p=1, because the computation is done for x1), r=4 (and, hence, r-1=3), there is only one entry in $[\mathrm{R}^2(p,r) - R^2(q,r-1)]$ and hence k=1. Thus, we obtain (0.982376-0.97282)/1 = 0.009556. For r=3 (and, hence, r-1=2) we have 3 entries in $[\mathrm{R}^2(p,r) - R^2(q,r-1)]$ and accordingly (0.982285+0.982335+0.981281-0.847025-0.68006-0.93529)/3 = 0.161175. Similarly, for r=2 (and, hence, r-1=1) we have 3 entries in $[\mathrm{R}^2(p,r) - R^2(q,r-1)]$ and accordingly (0.978678+0.548167+0.972471-0.666268-0.285873-0.674542)/3 = 0.290878. For r=1 (and, hence, r-1=0) we have only one entry in $[\mathrm{R}^2(p,r) - R^2(q,r-1)]$ and accordingly we have 0.533948. All the for accumulated and divided by m gives (0.009556+0.161175+0.290878+0.533948)/4 = 0.248889. This is the expected share of x_1 in R^2 presented in Table-4. For other regressors (= 2, 3 and 4) the similar computational scheme is used.

Table 5. *Computational details of share of x1 in R^2 of the Portland cement dataset*

r	r-1	x_1	x_2	x_3	x_4	R^2	K	operation	values	Sum/k	Grand value
4		1	2	3	4	0.982376		plus	0.982376		
	3		2	3	4	0.97282		minus	-0.97282		
							k=1	Sum/k		0.009556	
3		1	2	3		0.982285		plus	0.982285		
3		1	2		4	0.982335		plus	0.982335		
3		1		3	4	0.981281		plus	0.981281		
	2		2	3		0.847025		minus	-0.847025		
	2		2		4	0.68006		minus	-0.68006		
	2			3	4	0.93529		minus	-0.93529		
							k=3	Sum/k		0.161175	
2		1	2			0.978678		plus	0.978678		
2		1		3		0.548167		plus	0.548167		
2		1			4	0.972471		plus	0.972471		
	1		2			0.666268		minus	-0.666268		
	1			3		0.285873		minus	-0.285873		
	1				4	0.674542		minus	-0.674542		
							k=3	Sum/3		0.290878	
1		1				0.533948		plus	0.533948		
							k=1	Sum/k		0.533948	
								Sum(sum/k)/m			0.248889

Now, we proceed to retrieve regression coefficients. The matrix of pair-wise correlation among the regressor variables and the vector of pair-wise correlation between the regressand (y) and regressor variables are presented in Table-6. We optimize the quadratic function $f(\alpha)$, as noted in section 5, by the Host-Parasite Co-evolutionary Algorithm or HPC (Mishra, 2013). To be doubly sure, we have also optimized $f(\alpha)$ by the Differential Evolution (DE) method of global optimization and found the results identical. The DE has been found very effective in difficult nonlinear optimization problems (Mishra, 2007).

Table 6. *Pair-wise correlation matrix among regressors (S) and y and regressors (T)*

Pair-wise correlation matrix among regressor variables (S)				Correlation (y,x) or (T)
1.0000000000	0.2285794703	-0.8241337644	-0.2454451074	0.7307174720
0.2285794703	1.0000000000	-0.1392423761	-0.9729549989	0.8162525698
-0.8241337644	-0.1392423761	1.0000000000	0.0295370033	-0.5346706755
-0.2454451074	-0.9729549989	0.0295370033	1.0000000000	-0.8213050372

We obtain the minimal value of $f(\alpha)$ = 0.0000995876416053835 by HPC (Table-7), which gives R^2 = 0.9639077953492068 (squared correlation between y and $\hat{y} = x\hat{\alpha}$). The minimal value of $f(\alpha)$ = 0.0000995876416053834 by DE (Table-8), which gives R^2 = 0.9639077954654629 (squared correlation between y and $\hat{y} = x\hat{\alpha}$). Although, numerically, the R^2 obtained from DE is slightly better than that obtained from HPC (at the 10^{th} place after decimal), for practical purposes HPC and DE both yield (almost) identical R^2, but this R^2 is noticeably smaller than the R^2 obtained from OLS regression (0.982), or the one (0.9823756204) that was decomposed by the Shapley value reported in Table-4. Ideally, the min($f(\alpha)$) should be zero, but it is about 0.0001. Some of the possible reasons might be the effectiveness of an optimizing algorithm and its coding, or the rounding off errors accumulated in course of computation, or near-flatness of a quadratic function at and about its peak. This near-flatness may be dependent on the correlation structure of the dataset being analyzed. But the most potent reason is the fact that the imputed coefficients (α) derived from Shapley value departs from the OLS coefficients (β; this β characterizing min(u'u) and hence max(R^2)) and, therefore, must always pay in terms of loss in R^2. The R^2(y, Xα) must be smaller than R^2(y, Xβ) if α≠β. The magnitude of this loss would be dependent on the extent of departure of α from β. By the way, we also note the negative signs of coefficients associated with x_3 and x_4. This sign would be a serious concern for a scientist.

Table-7. *Regression coefficients retrieved from Shapley value by quadratic optimization by HPC*

Regressors	x_0	x_1	x_2	x_3	x_4
Standardized	-	0.32409026578	0.34345113757	-0.26775995747	-0.34897780483
Regular	-5.36219673960	0.82883324530	0.33203669858	-0.62888792954	-0.31364971095

Table 7a. *Regression coefficients retrieved from Shapley value by quadratic optimization by DE*

Regressors	x_0	x_1	x_2	x_3	x_4
Standardized	-	0.32409026433	0.34345113662	-0.26775995561	-0.34897780345
Regular	-5.36219675675	0.82883324160	0.33203669767	-0.62888792517	-031364970971

10. Some additional observations

In the traditional regression analysis most of the regressors appeared to be redundant, and yet, together, they had exhibited a very high explanatory power. It was paradoxical. The Shapley value regression has explained that such a paradox was due to inability of the traditional regression analysis, which basically assumes independence among the regressors, in dealing with the coalition (cooperative action of the regressors). It is well known that the traditional regression method (ordinary least squares) performs poorly when the Gauss-Markov assumptions are not fulfilled by the data and, therefore, econometricians had in the past invented many techniques such as the generalized least squares (when errors in the dependent variable or the regressand are heteroskedasic or correlated or both), the instrumental variables method (when regressor variables are not fixed, but random), the maximum likelihood estimation (which may be suitable if errors in the regressand are not normally distributed) and so on. These techniques, nevertheless, presume independence among explanatory variables. For dealing with a departure from the assumption of independence among the regressors, several methods (noted in section 2 above) have been proposed, but most of them perform poorly or require additional information (beyond the dataset) from the analyst. Most of those statistical methods for dealing with the multicollinear regressors also are biased estimators. Against these odds, the Shapley value regression needs no additional information and works out the performance individual regressors, which has several desirable properties, such as efficiency, symmetry, linearity, anonymity, marginalism, etc. well discussed in Hart (1989). However, if α≠β, it must pay in terms of loss in R^2, since R^2(y, Xα) based on the Shapley value must be smaller than R^2(y, Xβ) based on OLS.

11. A computer program in Fortran

The Fortran computer program (HPC based source code) is available which may be compiled by a suitable compiler and run. It needs that the data be stored in a text file (e.g. DATX.TXT) with as many rows as the number of observations (replicates, n), the first column containing the dependent (regressand) variable and the subsequent m (>1) columns containing the regressor variables. No header is to be provided, only numerical data be stored. The program and the data file must be in the same folder (directory). When the program runs, it needs the values of n and m (which may be coded in the program itself if fixed). Then it needs the input file name (e.g. DATX.TXT). It also needs the duration (in seconds) for which the program would run, which may not normally be more than 5 seconds. The program stores the output in a file (e.g. SHAPLEY_RESULTS.TXT). The source code uses the HPC algorithm. The DE based codes also may be obtained from the author on request.

12. Conclusion

Multicollinearity in empirical data violates the assumption of independence among the explanatory variables in a linear regression model and by inflating the standard error of estimates of the estimated regression coefficients leads to failure in rejecting a false null hypothesis of ineffectiveness of the regressor variable to the regressand variable (type II error). Very frequently, it also affects the sign of the regression coefficients. Shapley value regression is one of the best methods to combat this adversity to empirical analysis. To this end, the present paper has made two contributions, first in simplifying the algorithm to compute the Shapley value (decomposition of R^2 as fair shares to individual regressor variables) and secondly developing a computer program that works it out easily. Yet, it must be mentioned that the Shapley value regression becomes increasingly impracticable as the number of regressor variables exceeds 10 or 12, although, in practice, a good regression model should not have more than ten regressors.

Appendix

```
      !SHAPLEY REGRESSION FOR MULTICOLLINEARITY
      PARAMETER (NMAX=100,MMAX=10)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(NMAX,MMAX),Y(NMAX),XX(MMAX,MMAX),XY(MMAX),B(MMAX)
      DIMENSION VX(MMAX,MMAX),VY(MMAX),YH(NMAX),CONTRIB(MMAX),Z(NMAX)
      DIMENSION ARRAY(MMAX),BARRAY(MMAX),RMAT(MMAX,MMAX),RVECT(MMAX)
      DIMENSION BETA(MMAX),AVX(MMAX),SDX(MMAX)
      COMMON /HP/RMAT,RVECT,CONTRIB
      CHARACTER *70 INFIL,OFIL,OUTFIL,FINRES
      COMMON /DAT/X,Y
      ! ---------- NO. OF  OBSERVATIONS AND REGRESSOR VAIABLES --------
      !WRITE(*,*)'FEED N AND M'
      ! READ(*,*) N,M
      N=13!NUMBER OF OBSERVATIONS
      M=4! NO. OF VARIABLES (REGRESSORS, NO CONSTANT)
      ! ------------------- FILE NAMES --------------------------------
      !WRITE(*,*)'FEED THE NAME OF INPUT & OUTPUT FILES'
      !READ(*,*) INFIL, OUTFIL, OFIL
      INFIL='DATX.TXT' ! CONTAINS Y AND X DATA
      OFIL='ALLCOMB.TXT' ! STORES ALL COMBINATIONS
      OUTFIL='SHAPLEY_R.TXT'! STORES ALL COMBINATIONS WITH R SQUARE
      FINRES='SHAPLEY_RESULTS.TXT'
      ! ------------------- FORMATS ----------------------------------
1     FORMAT(8F10.2)
2     FORMAT(2I3,4F5.0,2X,F12.9)! BETTER TO MAKE IT RUN-TIME FORMAT
3     FORMAT(40X,2I3,4F5.0,2X,F12.9)! BETTER TO MAKE IT RUN-TIME FORMAT
4     FORMAT('REGRESSOR #',I2,' SHARED R_SQR =',F18.15,' [',F7.4,' %]')
5     FORMAT(5(F15.10,','))
      ! ***************************************************************
      OPEN(9,FILE=FINRES)
      OPEN(7,FILE=INFIL) ! CONTAINS Y (REGRESSAND) AND X (REGRESSORS)
      DO I=1,N
      READ(7,*) Y(I),(X(I,J), J=1,M)
      ! DATA DOES NOT JAVE CONSTANT

      ENDDO
      CLOSE(7)
      ! MAKE DEVIATED FROM THE RESPECTIVE MEAN
      DO J=1,M
      AM=0.D0 ! MEAN
      SD=0.D0 ! STANDARD DEVIATION
      DO I=1,N
      AM=AM+X(I,J)
      SD = SD + X(I,J)**2
      ENDDO
      AM=AM/N
      SD = SQRT(SD/N - AM**2)
      AVX(J)=AM
      SDX(J)=SD
      DO I=1,N
      X(I,J)=(X(I,J)-AM)/SD
      ENDDO
      ENDDO

      AM=0.D0 ! MEAN
      SD=0.D0 ! STANDARD DEVIATION
      DO I=1,N
```

```
AM=AM+Y(I)
SD=SD+Y(I)**2
ENDDO
AM=AM/N
AMY=AM
SD = SQRT(SD/N - AM**2)
SDY=SD
DO I=1,N
Y(I)=(Y(I)-AM)/SD
ENDDO
! -----------------   PRINT DATA Y AND X  ----------------------
! DO I=1,N
! WRITE(*,*) Y(I),(X(I,J), J=1,M)
! ENDDO
! ----------------------------------------------------------------
! MAKE VARIANCE-COVARIANCE MATRIX
DO J=1,M
XY(J)=0.D0 ! COVARIANCE VECTOR OF Y WITH X
DO JJ=1,M  ! VAIANCE-COVARIANCE MATRIX OF X WITH ITSELF
XX(J,JJ)=0.D0
DO I=1,N
XX(J,JJ)=XX(J,JJ)+ X(I,J)*X(I,JJ)
ENDDO
XX(J,JJ)=XX(J,JJ)/N
ENDDO
DO I=1,N
XY(J)=XY(J)+X(I,J)*Y(I)
ENDDO
XY(J)=XY(J)/N
ENDDO
! -------------------PRINT VARIANCE COVARIANCE MATRIX ----------
!WRITE(*,*) 'VARIANCE COVARIANCE MATRIX'
!DO I=1,M
!WRITE(*,1) XY(I),(XX(I,J),J=1,M)
!ENDDO
!WRITE(*,*)'======================================================'
! STORE XX IN V
! CONSTRUCT CORREL MATRIX (RMAT:XX) AND CORREL VECTOR (RVAT: YX)
DO J=1,M
 DO I=1,N
 YH(I)=X(I,J)
 ENDDO
CALL RSQUARE(Y,YH,N,RM,RSQ)
RVECT(J)=RM
 DO JJ=1,M
   DO I=1,N
   Z(I)=X(I,JJ)
   ENDDO
 CALL RSQUARE(Z,YH,N,RM,RSQ)
 RMAT(J,JJ)=RM
 ENDDO
ENDDO
! CORRELATION MATRIX AND VECTOR
WRITE(*,*)'CORRELATION MATRIX AND VECTOR'
DO J=1,M
WRITE(*,5)(RMAT(J,JJ),JJ=1,M),RVECT(J)
ENDDO
!PAUSE
DO I=1,M
BARRAY(I)=0.D0 ! INTERMEDIATE VARIABLE FOR INTERNAL PURPOSES
ENDDO
OPEN(14,FILE=OUTFIL)! STORES ALL COMBINATIONS WITH R SQUARE
NSL=0
DO IX=1,M
KX=M-IX+1
NCOMB=NCR(M,KX)! NO. OF COMBINATION NCR
CALL COMBIN(M,KX,OFIL)
OPEN(7,FILE=OFIL) ! CONTAINS COMBINATIONS
DO I=1,NCOMB
READ(7,*)(ARRAY(J),J=1,KX)! COMBINATION ARRAY
CALL REGRESS(XX,XY,ARRAY,RSQ,N,KX)! CALLS ORDINARY LEAST SQUARES
NSL=NSL+1
WRITE(14,*)NSL,KX,(ARRAY(J),J=1,KX),RSQ !STORES REGRESSION RESULTS
!WRITE(*,*)'R-SQUARE=',RSQ
ENDDO
CLOSE(7)
! WRITE(*,*)'----------------------------------------------------'
ENDDO
CLOSE(14)
! MAKE TABLES
!WRITE(*,*)'///////////////// CHECKING /////////////////////////'
OPEN(14,FILE=OUTFIL)
NCOMBTOT=0
DO IX=1,M
KX=M-IX+1
MKX=M-KX
NCOMB=NCR(M,KX)
DO I=1,NCOMB
READ(14,*)NSL,KKX,(ARRAY(J),J=1,KX),RSQ
!WRITE(*,2)NSL,KKX,(ARRAY(J),J=1,KX),(BARRAY(J),J=1,MKX),RSQ
IF(KX.NE.KKX) THEN
!WRITE(*,*)'KKX AND KX ARE NOT EQUAL ',KX,KKX
!PAUSE
ENDIF
NCOMBTOT=NCOMBTOT+1 !TOTAL NO. OF REGRESSION
!PAUSE
ENDDO
```

```
ENDDO
CLOSE(14)
!PAUSE
!WRITE(*,*)'[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[[['
!WRITE(*,*) 'TOTAL NO. OF REGRESSION =', NCOMBTOT
OPEN(14,FILE=OUTFIL)
TSUMSRQ=0.D0
DO KC=1,M
!WRITE(*,*)'FEED KC (DESIRED VARIABLE)'
!READ(*,*) KC
SUMSRQ=0
DO KPP=1,M
NTR1=0
NTR0=0
KP=M-KPP+1
WRITE(*,*)'------- COMBINATION=',KP,' -------------------------'
KR=KP-1
! -----------------------------------------------------------------
SRSQ=0.D0
OPEN(14,FILE=OUTFIL)
DO I=1,NCOMBTOT
READ(14,*)SL,KX,(ARRAY(J),J=1,KX),RSQ
NT=0
DO J=1,KX
IF(KX.EQ.KP.AND.ARRAY(J).EQ.KC) THEN
NT=NT+1
WRITE(*,*) '(',KX,(ARRAY(JJ),JJ=1,KX),RSQ,').(+)'
ENDIF
ENDDO
IF(NT.NE.0)THEN
WRITE(*,*) '(',KX,(ARRAY(JJ),JJ=1,KX),RSQ,').(+)'
NTR1=NTR1+1
SRSQ = SRSQ + RSQ ! RSQ TO BE ADDED
ENDIF
NT=0
DO J=1,KX
IF(KX.EQ.KR.AND.ARRAY(J).NE.KC)THEN
NT=NT+1
WRITE(*,*) '[',KX,(ARRAY(JJ),JJ=1,KX),RSQ,'].(-)'
ENDIF
ENDDO
IF(NT.EQ.KX)THEN
WRITE(*,*) '[',KX,(ARRAY(JJ),JJ=1,KX),RSQ,'].(-)'
NTR0=NTR0+1
SRSQ = SRSQ - RSQ ! RSQ TO BE SUBTRACTED
ENDIF
ENDDO
CLOSE(14)

!WRITE(*,*)'NTR1 & NTR0,SRSQ,MEAN_SRSQ:',NTR1,NTR0,SRSQ,SRSQ/NTR1
SUMSRQ = SUMSRQ + SRSQ/NTR1
ENDDO ! FOR KPP
! WRITE(*,*)'SUM OF PROPERLY SIGNED RSQ & MEAN =',SUMSRQ,SUMSRQ/M
! -----------------------------------------------------------------
CONTRIB(KC) = SUMSRQ/M
TSUMSRQ = TSUMSRQ + SUMSRQ/M
ENDDO ! FOR KC
WRITE(9,*)' '
WRITE(9,*)'  -------- FINAL RESULTS OF SHAPLEY REGRESSION ------'
WRITE(9,*)' CONTRIBUTION OF EACH INDIVIDUAL VARIABLE TO R_SQUARE'
DO J=1,M
WRITE(9,4) J,CONTRIB(J), (CONTRIB(J)/TSUMSRQ)*100
ENDDO
WRITE(9,*)' '
WRITE(9,*)'TOTAL(JOINT) CONTRIBUTION R_SQ=F(X1,...,XM)=',TSUMSRQ
WRITE(9,*)'NOTE: TOTAL CONTRIBUTION SUMS UP TO 100 PERCENT.'

CALL HOST_PARASITE(M,BETA)
! COMPUTE IMPUTED R SQUARED VALUE
WRITE(9,*)'SHAPLEY-VALUE BASED STANDARDIZED REGRESSION COEFFS'
WRITE(9,*)(BETA(J),J=1,M)
WRITE(9,*)'SHAPLEY-VALUE BASED REGULAR REGRESSION COEFFICIENTS'
AMM=0.D0
DO J=1,M
B(J)=BETA(J)*SDY/SDX(J)
AMM=AMM+AVX(J)*B(J)
ENDDO
CONSTANT=AVY-AMM
WRITE(9,*) CONSTANT, (B(J),J=1,M)

DO I=1,N
YH(I)=0.D0
DO J=1,M
YH(I)=YH(I)+ X(I,J)*BETA(J)
ENDDO
ENDDO
CALL RSQUARE(Y,YH,N,RM,RSQ)
WRITE(9,*)'COMPUTED SHAPLEY REGRESSION R_SQUARED =',RSQ
CLOSE(9)
WRITE(*,*)' '
WRITE(*,*)' '
WRITE(*,*)'RESULTS ARE STORED IN FILE = ',FINRES
STOP
END
!-----------------------------------------------------------------
SUBROUTINE INV(A,M,D)! MATRIX INVERSION
PARAMETER(MMAX=10)! MMAX IS THE MAXIMUM DIMENSION.
```

```
!MATRIX INVERSION - EXCHANGE METHOD:KRISHNAMURTHY EV & SEN SK(1976)
!COMPUTER-BASED NUMERICAL ALGORITHMS, AFFILIATED EAST-WEST PRESS,
!NEW DELHI, P.161
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION A(MMAX,MMAX)
! ------------------------------------------------------------
! INVERSION BEGINS
D=1.D0 ! D IS THE DETERMINANT OF MATRIX A.
! THE RESULT (INVERSE OF A) IS STORED IN A ITSELF. A IS LOST
DO I=1,M
  D=D*A(I,I)
  A(I,I)=1.D0/A(I,I)
  DO J=1,M
    IF(I.NE.J) A(J,I)=A(J,I)*A(I,I)
  ENDDO
  DO J=1,M
    DO K=1,M
      IF(I.NE.J.AND.K.NE.I) A(J,K)=A(J,K)-A(J,I)*A(I,K)
    ENDDO
  ENDDO
  DO J=1,M
   IF(J.NE.I) A(I,J)=-A(I,J)*A(I,I)
  ENDDO
ENDDO
! INVERSION ENDS
! ------------------------------------------------------------
! WRITE(*,*)'DETERMINANT=',D
RETURN
END
! ==================================================================
SUBROUTINE VINIT(VX,VY) ! INITIALIZES (INTERNAL USE)
PARAMETER (MMAX=10)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION VX(MMAX,MMAX),VY(MMAX)
DO I=1,MMAX
DO J=1,MMAX
IF(I.EQ.J) THEN
VX(I,J)=1
ELSE
VX(I,J)=0
ENDIF
ENDDO
VY(I)=0
ENDDO
RETURN
END
! -----------------------------------------------------------------
SUBROUTINE RSQUARE(Y,YH,N,RM,RSQ) ! FINDS REGRESSION R SQUARE
PARAMETER (NMAX=100)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION Y(NMAX),YH(NMAX)
AY=0
AYH=0
VY=0
VYH=0
VYYH=0
DO I=1,N
AY=AY+Y(I)
AYH=AYH+YH(I)
VY=VY+Y(I)**2
VYH=VYH+YH(I)**2
VYYH=VYYH+Y(I)*YH(I)
ENDDO
AY=AY/N
AYH=AYH/N
VY=VY/N-AY**2
VYH=VYH/N-AYH**2
VYYH=VYYH/N-AY*AYH
RM = VYYH/SQRT(VY*VYH)
RSQ=(VYYH**2)/(VY*VYH)
RETURN
END
!-----------------------------------------------------------------
SUBROUTINE COMBIN(IN,IR,OFIL)! LISTS COMBINATIONS
PARAMETER (NMAX=10,MMAX=2000)
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
DIMENSION A(NMAX),B(NMAX),C(MMAX)
CHARACTER *70 OFIL
OPEN(15,FILE=OFIL)
DO I=1,IR
A(I)=I
ENDDO
B(1)=IN+1-IR
DO J=1,IR
B(J+1)=B(J)+1
ENDDO
MM=1
1  DO K=1,IR
C(MM) = A(K)
MM=MM+1
ENDDO

IF(A(1).NE.B(1)) THEN
DO M=1,IR
IM=IR-M+1
A(IM)=A(IM)+1
IF(A(IM).LE.B(IM)) GO TO 2
```

```
      ENDDO
2     DO M=IM,IR
      A(M+1)=A(M)+1
      ENDDO
      GO TO 1
      ENDIF
      MM=MM-1
      I=1
      IX=IR
      DO WHILE(IX.LE.MM)
      WRITE(15,*)(INT(C(J)),J=I,IX)
      I=I+IR
      IX=IX+IR
      ENDDO
      CLOSE(15)
      RETURN
      END
! ================================================================
      SUBROUTINE REGRESS(XX,XY,ARRAY,RSQ,N,MX)! OLS SUBROUTINE
      PARAMETER (NMAX=100,MMAX=10)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(NMAX,MMAX),Y(NMAX),XX(MMAX,MMAX),XY(MMAX),B(MMAX)
      DIMENSION VX(MMAX,MMAX),VY(MMAX),YH(NMAX)
      DIMENSION ARRAY(MMAX)
      CHARACTER *70 INFIL
      COMMON /DAT/X,Y
      CALL VINIT(VX,VY)! INITIALIZE VX AND VY
      M=MX
      DO I=1,M
      II=INT(ARRAY(I))
      DO J=1,M
      JJ=INT(ARRAY(J))
      VX(I,J)=XX(II,JJ)
      ENDDO
      VY(I)=XY(II)
      ENDDO
      CALL INV(VX,M,DET)
! COMPUTE REGRESSION COEFFICIENT
      DO I=1,M
      B(I)=0
      DO J=1,M
      B(I) = B(I)+VX(I,J)*VY(J)
      ENDDO
      ENDDO
! FIND R SQUARE
      DO I=1,N
      YH(I)=0
      DO J=1,M
      JJ=INT(ARRAY(J))
      YH(I)=YH(I)+X(I,JJ)*B(J)
      ENDDO
      ENDDO
      CALL RSQUARE(Y,YH,N,RM,RSQ)
! ----- PRINT ORDINARY REGRESSION COEFFICIENTS AND R SQUARE -----
! WRITE(*,*)'ORDINARY REGRESSION COEFFICIENTS AND R SQUARE'
! WRITE(*,*)(B(J),J=1,M), ' RSQUARE =',RSQ
      RETURN
      END
! -----------------------------------------------------------------
      FUNCTION NCR(IN,IR) ! FINDS NCR
      NF=1
      NR=1
      DO I=1,IR
      NF=NF*(IN-I+1)
      NR=NR*I
      ENDDO
      NCR=NF/NR
      RETURN
      END
!-----------------------------------------------------------------
      SUBROUTINE HOST_PARASITE(M,RBETA)
! ALGORITHM & PROGRAM BY PROF. SK MISHRA, DEPT. OF ECONOMICS
! NORTH-EASTERN HILL UNIVERSITY (SHILLONG), INDIA
      PARAMETER(NMAX=1000, MMAX=10)
      PARAMETER(MAXREP=15, NREP=1)
! NMAX = MAXIMUM NUMBER OF BIRDS TO GENERATE
! MMAX = MAXIMUM DIMENSION OR NO. OF DECISION VARIABLES
      PARAMETER(IPRN=1000) ! DISPLAY INTERMEDIATE RESULTS AT EVERY IPRN
!-----------------------------------------------------------------
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
!-----------------------------------------------------------------
      PARAMETER(FSIGN=1, NSORT=0) !(FSIGN = -1 FOR MAXIMIZATION)
      PARAMETER(MAXITER=10000000, EPS=1.0D-14)
! MAXITER = MAXIMUM NO. OF ITERATIONS
! !EPS IS USED AS A CONVERGENCE CRITERION)
!-----------------------------------------------------------------
      DOUBLE PRECISION LEVY ! FUNCTION LEVY(BETA) IS DOUBLE PRECISION
      COMMON /RNDM/IU,IV ! TO GENERATE UNIFORM DISTR. RANDOM NUMBERS
      COMMON /KFF/KF,NFCALL,FTIT ! KF IS FUNCTION CODE; NFCALL IS THE
!NUMBER OF FUNCTION CALLS; FTIT IS THE TITLE OF THE FUNCTION
      CHARACTER *70 TIT(200),FTIT ! TIT IS TITLE OF FUNCTIONS, A BATTERY
! OF 100 TEST FUNCTIONS; FTIT = TITLE OF THE FUNCTION
      CHARACTER *70 HISTORY ! OUTPUT FILE TO STORE HISTORY OF COVERGENCE
      INTEGER IU,IV! FOR GENERATING UNIFORMLY DISTRIBUTED RANDOM NUMBERS
      DIMENSION CUCKOO(NMAX,MMAX),CROW(NMAX,MMAX),ICU(NMAX),ICR(NMAX)
      DIMENSION A(MMAX),FCU(NMAX),FCR(NMAX),TCUC(MMAX),TCRO(MMAX)
      DIMENSION OPTVAL(MAXREP),NRAND(MAXREP),EXTIME(MAXREP)
```

```
      DIMENSION EXCYCLE(MAXREP)
      CHARACTER *8 CLOCK, START_TIME, NOW_TIME
      COMMON /HP/RMAT,RVECT,CONTRIB
      DIMENSION RMAT(MMAX,MMAX),RVECT(MMAX),CONTRIB(MMAX),RBETA(MMAX)

      DATA (NRAND(I),I=1,MAXREP)/45331,44431,44421,44401,45671,53277,
     &34567,23171,98267,49821,11387,17869,12352,12017,10501/
!-----------------------------------------------------------------
! LINEAR FUNCTION
!DPROB(PROB)=0.3D0*(1.D0-PROB)
! GOMPERTZ CURVE
      DPROB(PROB)=0.7D0*(EXP(-2*EXP(-(1.0D0/(1+DLOG(1+PROB))))))
! LOGISTIC FUNCTION
!DPROB(PROB)=(0.5-0.35D0/(1.D0+EXP(-PROB)))
! LOGIT FUNCTION
!DPROB(PROB)=-0.05D0*DLOG(PROB/(1.D0-PROB))


!PROB(IT,FN1,FN2)=0.7*(EXP(-2*EXP(-(0.00001D0/(1+DLOG(1+DABS(FN1
!* -FN2))))*IT)))
! THIS STATEMENT FUNCTION DEFINES THE DETECTION/REJECTION FUNCTION
! OF A CUCKOO EGG BY THE HOST (0.7 IS THE UPPER LIMIT OF PROB)
!-----------------------------------------------------------------
!-----------------------------------------------------------------
!WRITE(*,*)'FILE (NAME.TXT) TO STORE HISTORY OF CONVERGENCE ?'
! THIS FILE STORES THE HISTORY OF CONVEGENCE OF CUCKOOS AND CROWS
!READ(*,*) HISTORY
      HISTORY='HIST.TXT'
      OPEN(15,FILE=HISTORY)
! ===============================================================
      DO IREP=1,NREP
 !----------------------------------------------------------------
      IF(IREP.EQ.1) THEN
! SELECT/CHOOSE THE FUNCTION TO OPTIMIZE
      CALL FSELECT(KF,M,FTIT) ! CHOOSES THE FUNCTION TO OPTIMIZE
!WRITE(*,*)'NO. OF CUCKOOS (EQUAL TO NO. OF CROWS) ?'
!WRITE(*,*)'THIS COULD BE BETWEEN 30 AND 100, SAY.'
!READ(*,*) NCU, NCR  ! NO. OF CUCKOOS (& CROWS) TO GENETE.
!NCU SHOULD BE NOT BE MORE THAN A HALF OF NMAX (NMAX > 2*NCU)
!NCR=NCU ! THE CROWS ARE AS MANY AS THE CUCKOOS
!WRITE(*,*)'FEED THE RANDOM NUMBER SEED'
!READ(*,*) IU ! RANDOM NUMBER SEED (5 DIGITS ODD INTEGER NUMBER)
      NCU=30
      NCR=30

      WRITE(*,*)' '
      WRITE(*,*)' '
      WRITE(*,*)'------------------------------------------------------'
      WRITE(*,*)'MAX TIME(SEC) TO RUN?. FIVE SECS ARE ENOUGH, FEED 5.'
!READ(*,*) AMAXSEC,VTHEN ! VTHEN IS REFERENCE VALUE
      READ(*,*) AMAXSEC
      VTHEN=9999
      WRITE(*,*)'------------------------------------------------------'
      ENDIF
! INITIALIZATION ------------------------------------------------
      BETA=3/2.D0  ! NEEDED TO GENERATE LEVY FLIGHTS (CUCKOOS)
      GAMMA=5/3.D0 ! NEEDED TO GENERATE LEVY FLIGHTS  (CROWS)
! ---------------------------------------------------------------
      BET=0.2D0  ! NEEDED TO GENERATE CAUCHY FLIGHTS (CUCKOOS)
      GAM=0.8D0  ! NEEDED TO GENERATE CAUCHY FLIGHTS  (CROWS)
! ---------------------------------------------------------------
      SCALE=10 !(SCALING OF INITIAL VALUES OF DECISION VARIABLES)
      FACTOR=SCALE ! SCALING FACTOR
      NFCALL=0  ! NO. OF FUNCTION CALLS : INITIALIZED
      CUSD=1.0D30 ! USED FOR CONVERGENCE CRITERION
      CRSD=1.0D30 ! USED FOR CONVERGENCE CRITERION
      PROX=0.00D0 ! DETERMINES CHOICE BETWEEN LEVY AND CAUCHY FLIGHTS
      PROB=0.5D0
      ALIF=1.0D-06 ! AFFECTS THE RATE OF COVERGENCE
      SUCCESS=0.0D0
      GHZ=2.4D0 ! CLOCK CYCLES (PER SECOND) OF THE CPU
!---------------------------------------------------------------
!GENERATE CUCKOOS RANDOMLY AND EVALUATE
!CALL TIME(CLOCK)
!START_TIME=CLOCK
      CALL CPU_TIME(START)
      IU=NRAND(IREP)
      KSEED=IU
      CALL RANDOM(RAND)
      DO I=1,NCU
      DO J=1,M
      CALL RANDOM(RAND)
      A(J)=(RAND-0.5)*FACTOR
      CUCKOO(I,J)=A(J)
      ENDDO
      CALL FUNC(M,A,F)
      FCU(I)=F*FSIGN
      ENDDO
!GENERATE CROWS RANDOMLY AND EVALUATE
      DO I=1,NCR
      DO J=1,M
      CALL RANDOM(RAND)
      A(J)=(RAND-0.5)*FACTOR
      CROW(I,J)=A(J)
      ENDDO
      CALL FUNC(M,A,F)
      FCR(I)=F*FSIGN
```

```
ENDDO
IF(NSORT.EQ.1) THEN
CALL SORT(CUCKOO,FCU,NCU,M) ! SORT CUCKOO POPULATION
CALL SORT(CROW,FCR,NCR,M)  ! SORT CROW POPULATION
LOCU=1
LOKR=1
ELSE
CALL FINDBEST(FCU,NCU,TOPCU,LOCU)
CALL FINDBEST(FCR,NCR,TOPKR,LOKR)
ENDIF
!-----------------------------------------------------------------
ICOUNT=0
!----------------------------------------------------------------
IT=0 ! INITIALIZATION OF ITERATION
FEPS=0.0D0 ! INITIALIZATION OF TERMINATION CONDITION
!DO IT=1,MAXITER
DO WHILE (IT.LE.MAXITER.AND.FEPS.EQ.0.0)
FN1=FCU(LOCU) ! BEST VALUE OF CUCKOOS
FN2=FCR(LOKR) ! BEST VALUE OF CROWS
PDET=DPROB(PROB) ! DEFINED IN THE STATEMENT FUNCTION
! SET ICU AND ICR TO ZERO
DO I=1,NCU
ICU(I)=0
ENDDO
DO I=1,NCR
ICR(I)=0
ENDDO
! CUCKOOS REGENERATE THEMSELVES (FLY) WITH LEVY FLIGHT
DO I=1,NCU
DO J=1,M

CALL RANDOM(RAND)
ALPHA=ALIF+(RAND)**2  ! AFFECTS THE SPEED OF CONVERGENCE
CALL RANDOM(RAND)
OMEGA=ALIF+(RAND)**2  ! AFFECTS THE SPEED OF CONVERGENCE
CALL RANDOM(RC)
CALL RANDOM(RAND)
L=1+INT(NCR*RAND)
CALL RANDOM(RAND)
DIFFN=(CROW(L,J)-CUCKOO(I,J))
IF(RAND.GE.PROX) THEN
A(J)=CUCKOO(I,J)+ALPHA*(RC-0.5)*LEVY(BETA)*DIFFN
!A(J)=CUCKOO(I,J)+ALPHA*(RC-0.5)*BURR12()*DIFFN
!A(J)=CUCKOO(I,J)+ALPHA*(RC-0.5)*GAUSS()*DIFFN
!A(J)=CUCKOO(I,J)+ALPHA*(RC-0.5)*CAUCHY(BET)*DIFFN
ELSE
A(J)=CUCKOO(I,J)+ALPHA*(RC-0.5)*CAUCHY(BET)*DIFFN
ENDIF
ENDDO
CALL FUNC(M,A,F)
! A NEW S0LUTION IS ADMITTED ONLY IF IT IS BETTER
FNEW=F*FSIGN
IF(FCU(I).GT.FNEW) THEN
FCU(I)=FNEW
ICU(I)=1
DO J=1,M
CUCKOO(I,J)=A(J)
ENDDO
ENDIF
ENDDO
! TRY TO PLACE THE EGGS OF CUCKOOS INTO CROW-NESTS
DO I=1,NCU
CALL RANDOM(RAND)
IX=1+INT(NCR*RAND)
CALL RANDOM(RAND)
MK=0
IF(RAND.GT.PDET.AND.ICR(IX).EQ.0) MK=1
IF(MK.EQ.1.AND.ICU(I).EQ.1.AND.FCR(IX).GT.FCU(I)) THEN
ICR(IX)=1
ICU(I)=1
FCR(IX)=FCU(I)
DO J=1,M
CROW(IX,J)=CUCKOO(I,J)
ENDDO
ENDIF
ENDDO
! SET ICU TO ZERO
DO I=1,NCU
ICU(I)=0
ENDDO
! SET ICR TO ZERO AND CROW(I,J) TO RANDOM. ALSO FIND FITNESS
DO I=1,NCR
IF(ICR(I).NE.0) THEN
DO J=1,M
CALL RANDOM(RK)
CALL RANDOM(RAND)
L=1+INT(NCU*RAND)
CALL RANDOM(RAND)
DIFFN=(CUCKOO(L,J)-CROW(I,J))
IF(RAND.GE.PROX) THEN
A(J)=CROW(I,J)+OMEGA*(RK-0.5)*LEVY(GAMMA)*DIFFN
!A(J)=CROW(I,J)+OMEGA*(RK-0.5)*BURR12()*DIFFN
!A(J)=CROW(I,J)+OMEGA*(RK-0.5)*GAUSS()*DIFFN
!A(J)=CROW(I,J)+OMEGA*(RK-0.5)*CAUCHY(GAM)*DIFFN
ELSE
A(J)=CROW(I,J)+OMEGA*(RK-0.5)*CAUCHY(GAM)*DIFFN
ENDIF
```

```
    ENDDO
    CALL FUNC(M,A,F)
    IF(FCR(I).GT.F*FSIGN) THEN
    FCR(I)=F*FSIGN
    DO J=1,M
    CROW(I,J)=A(J)
    ENDDO
    ICR(I)=0
    SUCCESS=SUCCESS+1
    ENDIF
    ENDIF
    ENDDO
    PROB=SUCCESS/(NCU*(IT+1))
    IF(NSORT.EQ.1) THEN
    CALL SORT(CUCKOO,FCU,NCU,M) ! SORT CUCKOO POPULATION
    CALL SORT(CROW,FCR,NCR,M)  ! SORT CROW POPULATION
    LOCU=1
    LOKR=1
    ELSE
    CALL FINDBEST(FCU,NCU,TOPCU,LOCU)
    CALL FINDBEST(FCR,NCR,TOPKR,LOKR)
    ENDIF
    BESTVAL=FCR(LOKR)
    ! DISPLAY RESULTS AT EVERY IPRN ITERATIONS
    IF(INT(ICOUNT/IPRN).EQ.(FLOAT(ICOUNT)/IPRN))THEN
    ICOUNT=0
    WRITE(*,1)
1   FORMAT(/39('*='))
    WRITE(*,*)'PROBLEM NO.=',KF,' DIMENSION=',M,' RANDOM SEED=',KSEED,
& ' EXPERIMENT NO. = ', IREP
    WRITE(*,*)'CUCKOO COORDINATE VALUES'
    WRITE(*,*)(CUCKOO(LOCU,J),J=1,M)
    WRITE(*,*)'-----------------------------------------------------'
    WRITE(*,*)'CROW COORDINATE VALUES'
    WRITE(*,*)(CROW(LOKR,J),J=1,M)
    WRITE(*,*)'-----------------------------------------------------'
    WRITE(*,*)'FITNESS OF CUCKOOS AND CROWS =',FCU(LOCU), FCR(LOKR)
    WRITE(*,*)'NO.OF FUNCTION CALLS=',NFCALL,' PROB OF REJECT=',PDET
    WRITE(15,*)(NFCALL+.0D0),FCU(LOCU),FCR(LOKR),PROB,PDET
    CALL MEANSD(FCU,NCU,CUMEAN,CUSD,CUSKEW)
    CALL MEANSD(FCR,NCR,CRMEAN,CRSD,CRSKEW)
    WRITE(*,*) 'FMEANS =',CUMEAN, CRMEAN,' FSD =',CUSD,CRSD
    WRITE(*,*)'SKEWNESS IN CUCKOO & CROW POPULATIONS =',CUSKEW,CRSKEW
    ! CUMEAN & CRMEAN ARE MEAN FUNCTION VALUES - CUCKOOS & CROWS
    ! CUSD & CRSD ARE STD DEV OF FUNCTION VALUES - CUCKOOS & CROWS
    ! CUSKEW & CRSKEW ARE SKEWNESS FUNCTION VALUES - CUCKOOS & CROWS
    ! CUMED & CRMED ARE MEDIANS OF FUNCTION VALUES FOR CUCKOOS & CROWS

    ! IF(CUSD.LT.EPS.OR.CRSD.LE.EPS) FEPS=1 ! TERMINATION CONDITION
    !CUSD=DABS(FCU(1)-FCU(N))
    !CRSD=DABS(FCR(1)-FCR(N))
    CALL CPU_TIME(FINISH)
    !CALL TIME(CLOCK)
    !NOW_TIME=CLOCK
    !NOW_SEC=SECNDS(0.0)
    !NSEC=NOW_SEC - START_SEC
    CPUT=(FINISH-START)
    WRITE(*,*) 'TIME_ELAPSED=', CPUT,' SECONDS.'
    WRITE(*,*)'-----------------------------------------------------'
    CYCL=CPUT*GHZ
    WRITE(*,*)'CPU TIME(S) TAKEN =',CPUT,' CLOCK CYCLES(GIGA) =',CYCL,
&'  EXPERIMENT #', IREP
    WRITE(*,*)'-----------------------------------------------------'
    ENDIF
    CALL CPU_TIME(FINISH)
    CPUT=(FINISH-START)
    IF(CPUT.GE.AMAXSEC) GOTO 2
    IF(DABS(BESTVAL-VTHEN).LT.1.0D-12) GOTO 2

    !---------------------------------------------------------------
    IF(CUSD.LT.EPS.OR.CRSD.LE.EPS) FEPS=0 ! TERMINATION CONDITION
    !---------------------------------------------------------------
    !ENDIF
    ICOUNT=ICOUNT+1
    IT=IT+1
ENDDO ! END OF WHILE LOOP
    !----------------------------------------------------------------
    WRITE(*,*)'TOTAL NO. OF FUNCTION CALLS =',NFCALL
    CLOSE(15)
2   OPTVAL(IREP)=BESTVAL
    ! EXTIME(IREP)=NSEC
    EXTIME(IREP)=CPUT
    EXCYCLE(IREP)=CYCL
    DO JH=1,M
    TCUC(JH)=TCUC(JH)+CUCKOO(LOCU,JH)
    TCRO(JH)=TCRO(JH)+CROW(LOKR,JH)
    ENDDO
ENDDO ! ENDS THE IREPEAT LOOP
    WRITE(*,*)'------------------------------------------------------'
    ! MEAN VALUE OF COORDINATES
    DO JH=1,M
    TCUC(JH)=TCUC(JH)/NREP
    TCRO(JH)=TCRO(JH)/NREP
    ENDDO
    WRITE(*,*)' '
    WRITE(*,*)'OPT F =',(OPTVAL(I),I=1,NREP)
    WRITE(*,*)' '
```

```
EXTM=0.D0
EXTS=0.D0
OPTM=0.D0
OPTS=0.D0
EXCYCM=0.D0
EXCYCS=0.D0
DO I=1,NREP
OPTM=OPTM+OPTVAL(I)
EXTM=EXTM+EXTIME(I)
EXCYCM=EXCYCM+EXCYCLE(I)
OPTS=OPTS+OPTVAL(I)**2
EXTS=EXTS+EXTIME(I)**2
EXCYCS=EXCYCS+EXCYCLE(I)**2
ENDDO
OPTM=OPTM/NREP
EXTM=EXTM/NREP
EXCYCM=EXCYCM/NREP
OPTS=DSQRT(DABS(OPTS/NREP-OPTM**2))
EXTS=DSQRT(DABS(EXTS/NREP-EXTM**2))
EXCYCS=DSQRT(DABS(EXCYCS/NREP-EXCYCM**2))
IF(OPTM.EQ.0.D0) THEN
CV=OPTS/(1+OPTM)
ELSE
CV=OPTS/OPTM
ENDIF
WRITE(*,*) 'MEAN, SD & CV',OPTM,OPTS,CV
WRITE(*,*) 'MEAN TIME & SD',EXTM,EXTS
WRITE(*,*) 'MEAN GIGA CYCLES & SD',EXCYCM,EXCYCS
CLOSE(15)
DO J=1,M
!RBETA(J)= TCUC(J)
RBETA(J)=CUCKOO(LOCU,J)
ENDDO
!WRITE(*,*)'PROGRAM ENDS. THANK YOU'
RETURN
END

!-----------------------------------------------------------------
SUBROUTINE NORMAL(R1,R2)
! PROGRAM TO GENERATE N(0,1) FROM RECTANGULAR RANDOM NUMBERS
! IT USES BOX-MULLER VARIATE TRANSFORMATION FOR THIS PURPOSE.
 !-----------------------------------------------------------------
!----- BOX-MULLER METHOD BY GEP BOX AND ME MULLER (1958) ---------
! BOX, G. E. P. AND MULLER, M. E. "A NOTE ON THE GENERATION OF
! RANDOM NORMAL DEVIATES." ANN. MATH. STAT. 29, 610-611, 1958.
! IF U1 AND U2 ARE UNIFORMLY DISTRIBUTED RANDOM NUMBERS (0,1),
! THEN X=[(-2*LN(U1))**.5]*(COS(2*PI*U2) IS N(0,1)
! ALSO,  X=[(-2*LN(U1))**.5]*(SIN(2*PI*U2) IS N(0,1)
! PI = 4*ARCTAN(1.0)= 3.1415926535897932384626433832795
! 2*PI = 6.283185307179586476925286766559
!-----------------------------------------------------------------
IMPLICIT DOUBLE PRECISION (A-H,O-Z)
COMMON /RNDM/IU,IV
INTEGER IU,IV
!-----------------------------------------------------------------
CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
U1=RAND ! U1 IS UNIFORMLY DISTRIBUTED [0, 1]
CALL RANDOM(RAND) ! INVOKES RANDOM TO GENERATE UNIFORM RAND [0, 1]
U2=RAND ! U1 IS UNIFORMLY DISTRIBUTED [0, 1]
X=DSQRT(-2.D0*DLOG(U1))
R1=X*DCOS(U2*6.283185307179586476925286766559D00)
R2=X*DSIN(U2*6.283185307179586476925286766559D00)
RETURN
END
!-----------------------------------------------------------------
! RANDOM NUMBER GENERATOR (UNIFORM BETWEEN 0 AND 1,BOTH EXCLUSIVE)
SUBROUTINE RANDOM(RAND)
 DOUBLE PRECISION  RAND
 COMMON /RNDM/IU,IV
INTEGER IU,IV
 RANDX=REAL(RAND)
 IV=IU*65539
 IF(IV.LT.0) THEN
 IV=IV+2147483647+1
 ENDIF
 RANDX=IV
 IU=IV
 RANDX=RANDX*0.4656613E-09
 RAND= RANDX
 RETURN
 END
!-----------------------------------------------------------------
DOUBLE PRECISION FUNCTION LEVY(BETA)
!GENERATING LEVY FLIGHT
! REFERENCE: GUTOWSKI, M. (2001) "LEVY FLIGHTS AS AN UNDERLYING
! MECHANISM FOR GLOBAL OPTIMIZATION ALGORITHMS", [JUNE 2001].
! HTTP://ARXIV.ORG/ABS/MATH-PH/0106003V1.
DOUBLE PRECISION BETA, R
COMMON /RNDM/IU,IV
INTEGER IU,IV
CALL RANDOM(R)
LEVY = 1.0D0/R**(1.0D0/BETA) - 1.0D0
RETURN
END
!-----------------------------------------------------------------
DOUBLE PRECISION FUNCTION CAUCHY(BETA)
! FOLDED CAUCHY DISTRIBUTION
```

```
      DOUBLE PRECISION R1,R2,BETA
      COMMON /RNDM/IU,IV
      INTEGER IU,IV
1     CALL NORMAL(R1,R2)
      CAUCHY=DABS(R1/R2)
      IF(CAUCHY.GT.500) GOTO 1
      RETURN
      END
!-----------------------------------------------------------------
      SUBROUTINE SORT(X,F,N,M)
! ARRANGING F(I) IN ORDER
      PARAMETER(NMAX=1000,MMAX=100)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION F(NMAX),X(NMAX,MMAX)
      DO I=1,N-1
      DO II=I+1,N
      IF(F(I).GT.F(II)) THEN
      T=F(I)
      F(I)=F(II)
      F(II)=T
      DO J=1,M
      T=X(I,J)
      X(I,J)=X(II,J)
      X(II,J)=T
      ENDDO
      ENDIF
      ENDDO
      ENDDO
      RETURN
      END
!----------------------------------------------------------------
      SUBROUTINE FINDBEST(F,N,BEST,LO)
! ARRANGING F(I) IN ORDER
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION F(*)
      BEST=F(1)
      LO=1
      DO I=1,N
      IF(F(I).LT.BEST) THEN
      BEST=F(I)
      LO=I
      ENDIF
      ENDDO
      RETURN
      END
! ---------------------------------------------------------------
      SUBROUTINE MEANSD(X,N,A,S,SKEW)
      PARAMETER(NMAX=1000,MMAX=200)
      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      DIMENSION X(*)
      A=0.D0
      S=0.D0
      DO I=1,N
      A=A+X(I)
      S=S+X(I)**2
      ENDDO
      S=DSQRT(DABS(N*S-A**2)/(N*N))
      A=A/N
      SKEW=0.D0
      IF(S.GT.0.0001) THEN
      DO I=1,N
      SKEW=SKEW+((X(I)-A)/S)**3
      ENDDO
      SKEW=N*SKEW/((N-1)*(N-2))
      ENDIF
      RETURN
      END
!-----------------------------------------------------------------
      SUBROUTINE FSELECT(KF,M,FTIT)
!THE PROGRAM REQUIRES INPUTS FROM THE USER ON THE FOLLOWING ------
!(1) FUNCTION CODE (KF), (2) NO. OF VARIABLES IN THE FUNCTION (M);
      CHARACTER *70 TIT(200),FTIT
      WRITE(*,*)'----------------------------------------------------'
      DATA TIT(1)/'KF=1 SHAPLEY VALUE REGRESSION M-VARIABLES M=?'/
!-----------------------------------------------------------------
!DO I=1,1
!WRITE(*,*) TIT(I)
!ENDDO
!WRITE(*,*)'----------------------------------------------------'
!WRITE(*,*)'FUNCTION CODE [KF] AND NO. OF VARIABLES [M] ?'
!READ(*,*) KF,M
      KF=1
      FTIT=TIT(KF) ! STORE THE NAME OF THE CHOSEN FUNCTION IN FTIT
      RETURN
      END
!-----------------------------------------------------------------
      SUBROUTINE FUNC(M,X,F)
!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
      PI=4.D+00*DATAN(1.D+00)! DEFINING THE VALUE OF PI
      NFCALL=NFCALL+1 ! INCREMENT TO NUMBER OF FUNCTION CALLS
!  KF IS THE CODE OF THE TEST FUNCTION
```

```
! -----------------------------------------------------------------
 IF(KF.EQ.1) THEN
 CALL CALCBET(M,X,F)
 RETURN
 ENDIF
 RETURN
 END
 !-----------------------------------------------------------------
 SUBROUTINE CALCBET(M,BETA,F)
 IMPLICIT DOUBLE PRECISION (A-H,O-Z)
 PARAMETER (MMAX=10)
 COMMON /HP/RMAT,RVECT,CONTRIB
 COMMON /RNDM/IU,IV
 COMMON /KFF/KF,NFCALL,FTIT
 CHARACTER *70 FTIT
 DIMENSION RMAT(MMAX,MMAX),RVECT(MMAX),CONTRIB(MMAX),T(MMAX)
 DIMENSION BETA(*)

 IF(NFCALL.EQ.1) THEN
 DO J=1,M
 BETA(J)= CONTRIB(J)/RVECT(J)
 ENDDO
 ENDIF

 DO J=1,M
 T(J)=0.D0
   DO JJ=1,M
   T(J)=T(J)+RMAT(J,JJ)*BETA(JJ)
   ENDDO
 ENDDO
 F=0.D0
 DO J=1,M
 F=F + (BETA(J)*(2.0D0*RVECT(J)-T(J))- CONTRIB(J))**2
 ENDDO
 RETURN
 END
```

References

Arumairajan, S.,& Wijekoon, P. (2013).Improvement of the preliminary test estimator when stochastic restrictions are available in linear regression model.*Scientific Research*, 3(4), 283-292.doi. 10.4236/ojs.2013.34033

Belsley, D.A., Kuh, E. & Welsch, R.E. (1980). *Regression Diagnostics, Identifying Influential Data and Sources of Collinearity*, Wiley, New York.

Chen, G.J. (2012). A simple way to deal with multicollinearity. *Journal of Applied Statistics*, 39(9), 1893-1909. doi. 10.1080/02664763.2012.690857

Gómez, R.S., Pérez, J.G., Martín, M.D.M.L. & García, C.G. (2016). Collinearity diagnostic applied in ridge estimation through the variance inflation factor. *Journal of Applied Statistics*, 43(10), 1831-1849. doi. 10.1080/02664763.2015.1120712

Hart, S. (1989). Shapley Value. In Eatwell, J., Milgate, M., and Newman, P (eds.). *The New Palgrave: Game Theory*. Norton, p.210-216.

Huang, C.C.L., Jou, Y.J.,& Cho, H.J. (2015). A new multicollinearity diagnostic for generalized linear models. *Journal of Applied Statistics*, 43(11), 2029-2043. doi. 10.1080/02664763.2015.1126239

Lipovetsky, S. (2006). Entropy criterion in logistic regression and Shapley value of predictors. *Journal of Modern Applied Statistical Methods*, 5(1), 95-106.

Macedo, P., Scotto, M.,& Silva, E. (2010). A general class of estimators for the linear regression model affected by collinearity and outliers. *Communications in Statistics - Simulation and Computation*, 39(5), 981-993. doi. 10.1080/03610911003695719

Mishra, S.K. (2004a). Multicollinearity and maximum entropy leuven estimator. *Economics Bulletin*, 3(25), 1-11.

Mishra, S.K. (2004b). Estimation under multicollinearity: Application of restricted Liu and maximum entropy estimators to the Portland cement dataset. [Retrieved from].

Mishra, S.K. (2007). Performance of differential evolution method in least squares fitting of some typical nonlinear curves. *Journal of Quantitative Economics*, 5(1), 140-177.

Mishra, S.K. (2013). Global optimization of some difficult benchmark functions by host-parasite coevolutionary algorithm. *Economics Bulletin*, 33(1), 1-18.

Özkale, M.R. (2012). Combining the unrestricted estimators into a single estimator and a simulation study on the unrestricted estimators. *Journal of Statistical Computation and Simulation*, 82(5): 653-688. doi. 10.1080/00949655.2010.550293

Özkale, M.R. (2014). The relative efficiency of the restricted estimators in linear regression models. *Journal of Applied Statistics*, 41(5), 998-1027. doi. 10.1080/02664763.2013.859234

Ročková, V.,& George, E.I. (2014). Negotiating multicollinearity with spike-and-slab priors. *Metron*, 72(2),217-229. doi. 10.1007/s40300-014-0047-y

Shapley, L.S. (1953). A Value for n-person Games. In *Kuhn, H.W. and Tucker, A.W. (eds.). Contributions to the theory of games. Annals of Mathematical Studies 28*. Princeton University Press, p.307-317.

Woods, H., Steinour, H.H.,& Starke, H.R. (1932). Effect of composition of Portland cement on heat evolved during hardening. *Industrial and Engineering Chemistry*, 24(11), 1207-1214. doi. 10.1021/ie50275a002

Wu, X. (2009). A weighted generalized maximum entropy estimator with a data-driven weight. *Entropy*, 11(4), 917-930. doi. 10.3390/e11040917

York, R. (2012). Residualization is not the answer: Rethinking how to address multicollinearity. *Social Science Research*, 41(6), 1379–1386. doi. 10.1016/j.ssresearch.2012.05.014