Title
Least Absolute Residuals Regression Routine

Authors
Rosenberg, Barr
Carlson, Daryl

Publication Date
1970-11-01

Peer reviewed

COMMITTEE ON ECONOMETRICS AND MATHEMATICAL ECONOMICS

Working Paper IP-162

LEAST ABSOLUTE RESIDUALS REGRESSION ROUTINE

by

Barr Rosenberg and Daryl Carlson

November 1970

Note: Preparation of this program was supported by NSF Grant GS-2102 and in part by subsidized funds from the University of California Computer Center. Copies of the program deck may be secured by writing to Professor Barr Rosenberg, Institute of Business and Economic Research, 156 Barrows Hall, University of California, Berkeley, 94720. Please indicate on the envelope "RE: LAR REGRESSION PROGRAM," and enclose $10.00 to cover variable costs.

INSTITUTE OF BUSINESS AND ECONOMIC RESEARCH
CENTER FOR RESEARCH IN MANAGEMENT SCIENCE
University of California, Berkeley

INDEX

I. GENERAL DESCRIPTION OF THE ROUTINE AND ITS CAPABILITIES

This routine computes parameter estimates for the linear regression model by minimizing the sum of absolute values of the residuals. Linear constraints on the parameter estimates may be utilized. A modified version of the Linear Programming Simplex Routine from the SHARE Library (SDA 3384) is used as the core of this routine.

The main storage and control variables are located in either blank or labeled COMMON to reduce the amount of core required by the routine, as well as to give the user complete access to, and control of, intermediate steps in the calculation procedure. The routine is currently dimensioned for a maximum of 204 observations and 20 explanatory variables.

Three basic modes of operation have been designed to allow the user, through his own FORTRAN program, complete flexibility in utilizing the many features of this routine.

Mode 1

Mode 1 simply computes the least absolute residuals (LAR) parameter estimates of a single regression equation. The output currently provided by the routine operating in this mode includes the parameter estimates, the unadjusted R^2, the Durbin-Watson statistic, the sum of absolute values of residuals, the sum of squared residuals, and the standard error of the regression.

Mode 2

The second mode enables the user to perform a Monte Carlo study as efficiently as possible. For this mode, the output, as described for Mode 1, can be requested for each regression equation or it can be suppressed. That is, if the user is interested only in retrieving and storing the parameter estimates from each equation, the printed output can be suppressed and the parameter estimates retrieved from the array BHAT(20) which is in labeled common DATA. In Mode 2, the user must call the Least Absolute Residuals subroutine (LAR) for each simulated regression equation and retrieve any results wanted from the COMMON blocks before the subroutine LAR is called for the next equation. Example (c) in Section V will illustrate, in detail, this interaction between the user's program and the LAR subroutine.

Mode 3

Mode 3 calculates the forecasting performance of the LAR parameter estimates over a subset of the data history in a time-series problem as follows:

The parameters are initially estimated over some base period of the observations, say, $t = 1,2,\ldots,L$, where L is at least as great as the number of parameters to be estimated and is smaller than the total number of observations available (T). These initial estimates are used to forecast the next period's $(L + 1)$ value of the dependent variable, using the known values for the explanatory variables in Period $L + 1$. This forecasted value is compared to the actual value for that period, yielding an evaluation of forecasting performance. Then, observation L + 1 is incorporated into the estimation period to yield "updated" parameter estimates, reflecting information for observations 1 through L + 1. This procedure is continued recursively, first forecasting and then incorporating each successive observation, until the data history is exhausted.

Thus, the forecasting performance, with a lead time of one observation, of the regression model can be evaluated over the (T - L) forecasted periods. The forecasting performance can also be evaluated for leads of greater than one observation. The routine will currently handle a maximum lead of 10 observations. If M is the number of periods lead, the forecasting performance of the regression model can then be evaluated over (T - L - M + 1) forecasting periods. The output currently provided with Mode 3 for each period is the parameter estimates and the forecast error for whatever lead is requested. The program will continue to recursively estimate and print out the parameter estimates until the last period in the data history is reached, but forecast errors will be omitted when the forecast period extends beyond the data history. Also, the complete printout (R^2, Durbin-Watson, etc.) as in Mode 1 is given for the parameter estimates over the first and last estimation periods. If the user desires to have this information printed for all periods, a control parameter can be set appropriately as illustrated in Example (d) of Section V. After the parameter estimates and statistics are printed for the last estimation period, the sum of absolute values of forecast errors over the evaluation interval is printed.

By setting the control parameter NRET = 1, the routine will return control to the user's program after each successive estimation and forecast evaluation to enable the user to save or use the parameter estimates and forecast error at each step. The estimates are stored in BHAT(20) and the forecast error for the specified lead in FORSAV. Example (e) in Section V illustrates the use of NRET = 1. With NRET = 0, the LAR routine will not return to the user's program until the parameter estimates of the final period have been computed and printed.

Options

In addition to the three basic modes, two optional features can be implemented with any mode. First, linear inequality constraints on the parameters to be estimated can be introduced. The general form of these addition constraints is:

$$\sum_{i=1}^{K} C_{ni} b_i \geq C_{n,K+1} \qquad n = 1,\ldots,N$$

where: N is the number of parameter constraints,

K is the number of explanatory variables, and

C is an N X (K + 1) matrix of constraint coefficients with elements:

$$
\begin{array}{c}
C_{11}b_1 + C_{12}b_2 + \ldots + C_{1K}b_K \geq C_{1,K+1} \\
C_{21}b_1 + C_{22}b_2 + \ldots + C_{2K}b_K \geq C_{2,K+1} \\
\vdots \\
C_{N1}b_1 + C_{N2}b_2 + \ldots + C_{NK}b_K \geq C_{N,K+1}.
\end{array}
$$

For example, if the regression equation has four parameters to be estimated (b_i, i = 1,2,3,4), then the following constraints:

$$b_1 \geq 0, \qquad 2b_2 + 3b_3 \geq 0, \qquad 3b_4 \geq 7$$

would be imposed by the C matrix:

$$
\begin{array}{ccccc}
1 & 0 & 0 & 0 & 0 \\
0 & 2 & 3 & 0 & 0 \\
0 & 0 & 0 & 3 & 7
\end{array}
$$

The routine is currently dimensioned for a maximum of ten parameter constraints. See Example (b) of Section V for a detailed illustration of the use of this feature.

The second optional feature is the use of extraneous initial parameter estimates to reduce the amount of searching necessary to reach the optimal solution. As with the parameter constraints, this feature may be implemented with any mode of operation. Example (c) of Section V illustrates the details of inputting extraneous initial estimates into the LAR subroutine.

II. FORMULATION OF AN LAR REGRESSION AS A BOUNDED VARIABLES LINEAR PROGRAMMING PROBLEM

For the standard linear regression model,

$$Y_t = \sum_{i=1}^{K} \beta_i X_{it} + u_t, \qquad t=1,\ldots,T$$

the parameter estimates b_i, $i=1,\ldots,K$ and residuals (or fitting errors) e_t, $t=1,\ldots,T$ are related by the equations

$$Y_t = \sum_{i=1}^{K} b_i X_{it} + e_t, \qquad t=1,\ldots,T.$$

The following linear programming model can be constructed to minimize the sum of absolute values of the regression residuals:

$$\underset{b_i}{\text{MINIMIZE}} \quad \sum_{t=1}^{T} | e_t |$$

subject to the constraints:

$$\sum_{i=1}^{K} b_i X_{it} + e_t = Y_t \qquad t=1,\ldots,T$$

$$e_t, b_i \quad \text{unrestricted in sign.}$$

To facilitate solution by a standard linear programming algorithm, the objective function can be transformed, using the method in Charnes, Cooper, and Fergusen [2], by defining two additional variables, e_t^+ and e_t^-, to take the value of the positive and negative residuals, respectively. That is, if $e_t \geq 0$, then $e_t^+ = e_t$ and $e_t^- = 0$. Similarly, if $e_t < 0$, then $e_t^- = -e_t$ and $e_t^+ = 0$. The objective function can then be written as:

$$\underset{b_i}{\text{MINIMIZE}} \quad \sum_{t=1}^{T} e_t^+ + \sum_{t=1}^{T} e_t^-.$$

With constraints on the parameters added, the linear programming model with N parameter constraints will be:

$$\underset{b_i}{\text{MINIMIZE}} \quad \sum_{t=1}^{T} e_t^+ + \sum_{t=1}^{T} e_t^-$$

subject to the constraints:

$$\sum_{i=1}^{K} b_i X_{it} + e_t^+ - e_t^- = Y_t \qquad t=1,\ldots,T$$

$$\sum_{i=1}^{K} C_{ni} b_i \geq C_{n,K+1} \qquad n=1,\ldots,N$$

$$b_i \text{ unrestricted in sign, } e_i^+ > 0,\ e_i^- > 0.$$

This yields a linear programming model consisting of $T + N$ (the number of observations plus the number of parameter constraints) linear relations in $K + 2T$ (the number of parameters to be estimated plus the number of error terms) unknowns, which can become computationally unwieldy if $(T + N)$ is large.

Following the procedure suggested by Wagner [3], a more manageable dual form of this problem can be derived. The direct dual of the above model is:

$$\underset{z_t,\, v_j}{\text{MAXIMIZE}} \quad \sum_{t=1}^{T} z_t Y_t + \sum_{j=1}^{N} v_j c_{j,K+1}$$

subject to the constraints:

$$\sum_{t=1}^{T} z_t X_{it} + \sum_{j=1}^{N} v_j c_{ji} = 0 \qquad i=1,\ldots,K$$

$$z_t \leq 1 \qquad t=1,\ldots,T$$

$$-\, z_t \leq 1 \qquad t=1,\ldots,T$$

which has $K + 2T$ linear relations in $T + N$ unknowns. However, by letting $w_t = z_t + 1$ $(t=1,\ldots,T)$, the dual can be written as:

$$\underset{w_t,\, v_j}{\text{MAXIMIZE}} \quad \sum_{t=1}^{T} w_t Y_t - \sum_{t=1}^{T} Y_t + \sum_{j=1}^{N} v_j c_{j,K+1}$$

subject to the constraints:

$$\sum_{t=1}^{T} w_t X_{it} + \sum_{j=1}^{N} v_j c_{ji} = \sum_{t=1}^{T} X_{it} \qquad i=1,\ldots,K$$

$$0 \le w_t \le 2, \qquad v_j \ge 0.$$

Now the dual form involves only K linear relations in $T + N$ bounded nonnegative variables. This formulation allows the "bounded variables" algorithm proposed by Dantzig [1] to be used. This algorithm differs from the standard simplex algorithm in that it contains one additional step. The constraints which impose bounds on the variables are deleted from the constraints in the "bounded variables" tableau. Instead, each time a new variable is to be included in the basis, a check is made to see if inclusion of the new variable will cause either the new variable (if it is bounded), or some bounded variable already present in the basis, to be forced against its upper bound. Let us establish some notation and formalize the bounded-variables algorithm. At any stage in the simplex algorithm, define:

γ as the vector of shadow prices of the observations included in the present basis

X_t' as the vector $(X_{1t}\ X_{2t}\ \ldots\ X_{Kt})$

A as the matrix of coefficients of the included basis activities in the constraint tableau. Note--if the basis activities are observations, $A = (X_{i_1} \ldots X_{i_K})$ where i_j is the index of the j^{th} included observation

RHS' as the right-hand side vector $\left(\sum_{t=1}^{T} X_{1t}\ \sum_{t=1}^{T} X_{2t} \ldots \sum_{t=1}^{T} X_{Kt}\right)$

η as $A^{-1} X_t$ or in words, the expression for the t^{th} observation vector in terms of the basis activities.

In the general simplex algorithm subject to equality constraints, in order to bring in an observation X_t at the level λ, we must adjust γ by $-\lambda\eta$, since

$$A\gamma = \text{RHS} \Longrightarrow A(\gamma-\lambda\eta) + \lambda X_t = \text{RHS}.$$

In the bounded variables algorithm, λ is set to that value which will force exactly one of the b_i either to 0 or to its upper bound of 2 if it is a bounded variable. However, if the new activity is also bounded above at 2, we do not bring it into the basis, but instead transform it, if a value of λ greater than 2 is needed to force out a basis activity. Examining the constraints, we see that λ is given by

$$\lambda = \text{MINIMUM } (\lambda_1, \lambda_2, 2)$$

where $\lambda_1 = \text{MINIMUM } [\gamma_i/\eta_i]$

$$i\varepsilon \{\text{sign } (\gamma_i) = \text{sign } (\eta_i)\}$$

$$\lambda_2 = \text{MINIMUM } [(2-\gamma_i)/\eta_i]$$

$$i\varepsilon \{\text{sign } (\gamma_i) \neq \text{sign } (\eta_i) \text{ and } \gamma_i \text{ is bounded at } 2\}.$$

If λ_1 is minimal, a basis activity goes to zero. If λ_2 is minimal, a basis activity is forced to its upper bound and removed by the transformation described below. If 2 is minimal, the newly introduced observation reaches its upper bound without a change in the existing basis, and is transformed. The following operations are applied whenever some variable (i.e. the ℓ^{th}) is forced against its upper bound.

Define a new variable $w_\ell^* = 2 - w_\ell$

then set $w_\ell^* = 0$ or equivalently $w_\ell = 2$

and $y_\ell^* = -y_\ell$

and $x_{i\ell}^* = -x_{i\ell}$ $\qquad i=1,\ldots,K.$

Also, the objective function constant is incremented:

$$\left(\sum_{t=1}^{T} Y_t\right)^* = \sum_{t=1}^{T} Y_t - 2.0\, y_\ell$$

as well as the right-hand side of the constraints,

$$\left(\sum_{t=1}^{T} X_{it}\right)^* = \sum_{t=1}^{T} X_{it} - 2.0\, X_{it} \qquad i=1,\ldots,K.$$

After these operations, the activity w_ℓ^* is absent from the basis (for its value is zero), but the problem has been transformed so that the original variable w_ℓ is at its upper bound. These operations have the effect of removing the variable w_ℓ from the basis. The levels of the variables in the basis are adjusted to reflect the transformation of w_ℓ, and the simplex algorithm continues.

III. EXPLOITING THE LINEAR PROGRAMMING MODEL FOR COMPUTATIONAL EFFICIENCY

Additional insight into the meaning and capabilities of the dual form of the linear programming model developed in the previous section can be gained by performing some algebraic manipulations on the dual objective function and constraints. Ignoring the inequality parameter constraints for the time being, the objective function

$$\underset{w_t}{\text{MAXIMIZE}} \quad \sum_{t=1}^{T} w_t Y_t - \sum_{t=1}^{T} Y_t \quad \text{can be written as} \quad \underset{w_t}{\text{MAXIMIZE}} \quad \sum_{t=1}^{T} (w_t - 1)\, Y_t .$$

The constraints $\sum_{t=1}^{T} (w_t X_{it}) = \sum_{t=1}^{T} X_{it}$ for $i=1,\ldots K$

can be written as $\sum_{t=1}^{T} (w_t - 1)\, X_{it} = 0$ for $i=1,\ldots,K$

and multiplying by b_i and combining yields $\sum_{i=1}^{K} b_i \sum_{t=1}^{T} (w_t - 1)\, X_{it} = 0$.

Therefore, the optimal solution also maximizes

$$\sum_{t=1}^{T} (w_t - 1)(Y_t) - \left(\sum_{i=1}^{K} b_i \sum_{t=1}^{T} (w_t - 1)\, X_{it} \right)$$

or rewritten

$$\sum_{t=1}^{T} (w_t - 1) \left(Y_t - \sum_{i=1}^{K} b_i X_{it} \right) .$$

Thus it is clear that at any solution, the shadow prices (the w_t's) will equal two when the residual is positive, zero when the residual is negative. For those cases when the residual is zero (these are the observations chosen as a basis) the shadow prices are free to vary between zero and two. When parameter constraints are included, the form which is maximized becomes:

$$\sum_{t=1}^{T} (w_t - 1)\left(Y_t - \sum_{i=1}^{K} b_i X_{it} \right) + \sum_{j=1}^{N} v_j \left(c_{j,K+1} - \sum_{i=1}^{K} b_i c_{ji} \right) = 0$$

The v_j are zero when the constraints are not satisfied and may be positive when the constraints are satisfied. The w_t behave as in the previous case.

This interpretation of the dual can be exploited to hasten the computation process when initial extraneous estimates of the parameters are available. The residuals for the regression model, relative to any initial estimates, can be calculated and their signs used as initial estimates of the signs of the residuals in the final LAR solution. Using these estimated signs of the residuals for each observation, the w_t can be initially set to the appropriate bounds. Thus, for each observation t, if the residual $y_t - \sum_{i=1}^{K} \hat{b}_i x_{it}$ is positive, the transformation described at the end of Section II is carried out. This initial step can save computation time relative to that required for the linear programming routine to hit each bound separately in its standard search procedure.

In a Monte Carlo application, successive replications are made without changes to the explanatory variables. Successive runs can therefore be made with the entire constraint set from the previous run as a starting point. Only the objective function, involving the newly generated values of the dependent variable, need be recalculated. Since the explanatory variables and constraints will have been transformed during the bounded-variables procedure in the preceding replication, the newly constructed dependent variables must be comparably transformed. Each new observation of the dependent variable is given a positive or negative sign as a coefficient in the objective function and in the objective function constant according as that observation was or was not transformed in the previous replication.

After this initialization of the new objective function the "true values" (those set by the experimenter) of the parameters are used as extraneous initial estimates of the parameters. The procedure described at the end of Section II is used to transform the bounded variables in accordance with the sign of the residuals of the new set of dependent variables relative to these true values. (These signs are just the signs of the simulated disturbances, since the disturbances are the residuals relative to the true parameter values, with the exception that the sign is changed whenever an observation appeared transformed in the previous replication). Thus,an error variable is transformed to the opposite bound whenever the sign of the newly simulated disturbance differs from the sign of the previously estimated residual. These stratagems substantially reduce computation time.

The dual linear programming structure also permits recursive forecast evaluation to be accomplished efficiently, since the "re-estimation" process only involves adding one observation. One new activity, corresponding to the error in fitting this new observation, is added. The objective function and the constraint conditions are also incremented by the new observation. For example, if we want to add the T + 1 observation, the following sums must be incremented.

$$\sum_{t=1}^{T+1} Y_t = \sum_{t=1}^{T} Y_t + Y_{T+1}$$

$$\sum_{t=1}^{T+1} X_{it} = \sum_{t=1}^{T} X_{it} + X_{i,T+1} \qquad i = 1, \ldots, K$$

With these two calculations performed, the next step is simply to check to see if the previous base activities are still feasible (now over T + 1 observations), and if they are not we need some economical way to constructing a feasible basis for the new problem. In order to explain our procedure, let us return to the notation of Section II.

From the fact that the previous basis is feasible, we know that $A\,\gamma = RHS$. Therefore,letting $\eta = A^{-1}X_{T+1}$

$$A\,(\gamma + \lambda\eta) + (1 - \lambda)\,X_{T+1} = RHS + X_{T+1}$$

where the right hand side of this equation is the new right hand side after the addition of observation T + 1. Therefore we can bring in the new observation at an activity level $(1 - \lambda)$, if addition of $\lambda\eta$ forces out a basis activity. If no value of $\lambda < 1$ forces out a basis activity, the old basis remains feasible with the new observation added. To implement this we simply use a modification of the simplex algorithm, seeking the value of λ such that,

$$\lambda = \text{MINIMUM } (\lambda_1, \lambda_2, 1)$$

where

$$\lambda_1 = \text{MINIMUM } (\gamma_i/\eta_i)$$

$$i \in \{\text{sign } (\gamma_i) = \text{sign } (-\eta_i)\}$$

$$\lambda_2 = \text{minimum } ((2-\gamma_i)/\eta_i)$$

$$i \in \{\text{sign } (\gamma_i) \neq \text{sign } (-\eta_i) \text{ and } \gamma_i \text{ is bounded at } 2\}$$

If λ_1 or λ_2 is minimal, an included activity is removed as before and the T + 1 observation is brought in at level $(1 - \lambda)$. If 1 is minimal, the new observation can be brought in at level zero, i.e., the old basis remains feasible over the T + 1 observations.

The standard simplex routine can be used directly in this case of adding a new observation. We attempt to introduce X_{T+1} into the extant basis, with the changes that η is multiplied by -1, the new observation (if brought in) is brought in at level (1 - λ) instead of λ, and the range in this case of λ is 0 to 1.

IV. GLOSSARY OF VARIABLES

To give the user full flexibility in utilizing the many features of this routine, all of the control parameters and storage arrays are defined and dimensioned below.

Data Arrays:

(1) C(204) - dependent variable

(2) A(20,204) - explanatory variables

Parameter Constraints:

(1) STRAIN (10, 21) - maximum of 10 constraints and 20 variables

Final and Intermediate Storage Arrays:

(1) BHAT(20) - estimated parameters

(2) TRANS(204) - sign of $\left(\sum_{i=1}^{K} b_i X_{it} - Y_t \right)$ for each observation t

(3) B(20) - sum of explanatory variables over observations

(4) E(20,20) - inverse of the transposed matrix of coefficients of the explanatory variables (or constraint coefficients) for the observation (or constraint) in the basis

(5) P(21) - identical to BHAT

(6) X(20) - shadow prices of observations included in the basis

(7) PE(20) - if IFOPT = 0, PE is the estimate of the parameters when the optimum is first achieved. This will differ from BHAT if there were computational errors built up during the iterative computation of E. If IFOPT = 1, PE is an external estimate.

(8) JH(20) - indices of constraints which are not effective, or (for LAR) the indices of the observations or parameter constraints included in the basis.

(9) KB(204) - KB(JH(I)) = I ... for recall purposes.

(10) Y(21) - equals E*A(.,JT) where JT is the latest observation to enter the basis.

(11) FORSAV - forecast error for requested lead time.

User Set Control Parameters:

(1) MODE - equals 1, 2, or 3 (see Section I).

(2) NOBS - number of observations to be fitted.

(3) KVAR - number of explanatory variables.

(4) NSTRAT - number of inequality parameter constraints.

(5) IFBHAT - equals 1 if initial estimates of the parameters are to be utilized; equals 0 otherwise.

(6) INDHI - index of the last observation to be used in the estimation period. In forecasting evaluation mode, this is the terminator for the base period, after which recursive estimation begins. Equals MAXLEN in modes 1 and 2.

(7) MAXLEN - index of the last observation to be used in the regression (generally NOBS).

(8) IFOPT - equals 0 if inversion after the optimum is desired; equals 1 if not. Inversion of the optimum is a precaution to avoid numerical errors built up in the iterative computation of E. We have found this to be unnecessary in scientific computers.

(9) IPSUP - equals 1 if output is to be suppressed; equals 0 otherwise.

(10) LEAD - number of periods lead desired for forecast evaluation in Mode 3.

(11) NRET - equals 1 if LAR routine is to return to user's program after each estimation period in Mode 3. Equals 0 otherwise.

Internally Set Control Parameters:

(1) INDLO - index of the first observation to be included in the problem, set to be + 1.

(2) KRTRAN - controls the retransformation of dependent and explanatory variables.

(3) OBJECT - sum of dependent variables for observations in the basis.

(4) IFFAIL - equals 0 if the problem was feasible
- equals 1 if the problem was infeasible
- equals 2 if the problem had an infinite solution
- equals 4 if the algorithm did not terminate

(5) NTRANS - number of bounds hit during the operation of the program (number of instances where the transformation procedure was carried out).

(6) ITER - number of iterations taken (number of variables brought into the basis).

(7) NUMVR - number of inversions performed (an inversion is performed after [KVAR/ 2 + 5] iterations).

(8) INVC - number of iterations performed since the last inversion.

(9) NUMPV - number of pivot steps performed (equals ITER plus NUMVR times KVAR).

(10) NPIV - number of pivot steps taken since the last inversion.

(11) NVER - maximum number of iterations allowed before an inversion is performed. Currently set at (KVAR/2 + 5) * 10.

(12) NCUT - maximum number of iterations allowed. Currently set at (4 * KVAR + INDHI + NSTRAT + 15) * 10. The routine will terminate if this limit is exceeded.

V. EXAMPLES OF PROCEDURES FOR EACH MODE OF OPERATION

EXAMPLE (a) - SINGLE EQUATION -

To construct the data matrices and set the control parameters to solve a single regression problem, the following Fortran program would be needed.

```
      PROGRAM DRIVE(INPUT,OUTPUT)

      COMMON/DATA/A(20,204),C(204),B(20),TRANS(204),E(20,20),P(21),
     1 X(20),PE(20),JH(20),KB(204),Y(21),BHAT(20),STRAIN(21,10)
     2,FORSAV

      COMMON IFBHAT, IFFAIL, IFOPT, INDHI, INDLO, INVC, ITER, KRTRAN,
     1 KVAR, MAXLEN, MODE, NOBS, NPIV, NSTRAT, NTRANS, NUMPV, NUMVR,
     2 OBJECT,IPSUP,LEAD,NRET

      READ 10, (C(N),N=1,68)
   10 FORMAT(6F12.4)
      DO 11 I=2,7
      READ 10, (A(I,N),N=1,68)
   11 CONTINUE
      DO 12 N=1,68
   12 A(1,N)=1.0

*     NOBS=INDHI=MAXLEN=68
*     NSTRAT=IFOPT=IFBHAT=0
*     NRET=0
*     IPSUP=0
*     KVAR=7
*     MODE=1
*     CALL LAR

      STOP
      END
```

The printed output from this run would appear as on the following page.

```
LEAST ABSOLUTE RESIDUALS ESTIMATES

ESTIMATED COEFFICIENTS ......
          .408474E+01
          .146083E-01
          .112307E-02
         -.198452E-01
         -.739088E-02
         -.596843E-02
         -.354726E-02

R-SQUARED =      .7236

DURBIN-WATSON =  1.5182

NUMBER OF OBSERVATIONS =    68

SUM OF SQUARED RESIDUALS =      .721261E+01

STANDARD ERROR OF THE REGRESSION =      .34386CE+00

SUM OF ABSOLUTE VALUES OF RESIDUALS =      .163987E+02
```

EXAMPLE (b) - SINGLE EQUATION WITH PARAMETER CONSTRAINTS -

To solve a single regression problem with linear inequality parameter constraints, the following Fortran instructions should be used in place of the starred (*) instructions in EXAMPLE (a). This example is estimating the same equation as in the previous case but with the following two parameter constraints: $b_4 \geq 0.10$ and $b_5 \geq b_3$.

```
      NOBS=INDHI=MAXLEN=68
      IFOPT=IFBHAT=0
      NRET=0
      IPSUP=0
      KVAR=7
      MODE=1
      NSTRAT=2

      DO 20 I=1,NSTRAT
      DO 20 J=1,8
   20 STRAIN(J,I)=0.0

      STRAIN(4,1)=1.0
      STRAIN(8,1)=0.10
      STRAIN(5,2)=1.0
      STRAIN(3,2)=-1.0

      CALL LAR
```

The printed output from this run would appear as below.

```
LEAST ABSOLUTE RESIDUALS ESTIMATES

ESTIMATED COEFFICIENTS ......
        -.834952E+01
        -.542897E-01
         .159307E-02
         .100000E+00
         .159307E-02
         .962673E-02
        -.237222E-01

R-SQUARED =        .5382

DURBIN-WATSON =   1.4776

NUMBER OF OBSERVATIONS =    70

SUM OF SQUARED RESIDUALS =      .120527E+02

STANDARD ERROR OF THE REGRESSION =      .444506E+00

SUM OF ABSOLUTE VALUES OF RESIDUALS =      .208698E+02
```

EXAMPLE (c) - MONTE CARLO APPLICATION -

The following set of instructions illustrates the basic program structure required to perform a Monte Carlo experiment with the LAR routine. Note that with IPSUP = 0 the complete printout of parameter estimates and statistics would be given for each equation estimated. By setting IPSUP = 1, as in this example, the printout is suppressed.

```
      DIMENSION BSTORE(3,4),BETA(3),F(80),DUMMY(80)
      BEGIN=0.5875
      SIGMA=SQRT(50.0)
      CALL GOODRAN(RANVAR,0,BEGIN,SIGMA)
      NREP=4
      NOBS=INDHI=MAXLEN=68
      NSTRAT=IFOPT=0
      NRET=0
      IFBHAT=1
      IPSUP=1
      KVAR=3
      MODE=2
      BETA(1)=5.0
      BETA(2)=0.04
      BETA(3)=0.002
      DO 16 I=1,NOBS
   16 DUMMY(I)=BETA(1)+(BETA(2)*A(2,I))+(BETA(3)*A(3,I))
      DO 1000 N=1,NREP
      CALL ERRGEN(F)
      DO 15 I=1,NOBS
   15 C(I)=DUMMY(I)+F(I)
      DO 17 K=1,KVAR
   17 BHAT(K)=BETA(K)
      CALL LAR
      DO 22 K = 1,KVAR
   22 BSTORE(K,N) = BHAT(K)
 1000 CONTINUE
      PRINT 14, ((BSTORE(I,J),I=1,3),J=1,4)
   14 FORMAT(1H0,*PARAMETER ESTIMATES *,/,4(10X,3F15.6,/))
```

The printed output from this run would appear as below.

```
 PARAMETER ESTIMATES
                   5.293435         .033575          .006930
                   9.079679         .012218         -.002519
                   6.539467         .026785          .003537
                   3.503639         .061087          .000804
```

```
LEAST ABSOLUTE RESIDUALS ESTIMATES

ESTIMATED COEFFICIENTS ......
        .633753E+01
        .139279E-01
        .102283E-02
       -.487775E-01

R-SQUARED =      .7068

DURBIN-WATSON =  1.2793

NUMBER OF OBSERVATIONS =   60

SUM OF SQUARED RESIDUALS =      .577262E+01

STANDARD ERROR OF THE REGRESSION =      .321065E+00

SUM OF ABSOLUTE VALUES OF RESIDUALS =      .136884E+02


PARAMETER ESTIMATES FOR PERIOD    1       ( 60 OBSERVATIONS )
   .6337534E+01    .1392786E-01    .1022833E-02  -.4877746E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...          -.3169345E+00

PARAMETER ESTIMATES FOR PERIOD    2       ( 61 OBSERVATIONS )
   .5637449E+01    .1087093E-01    .1042459E-02  -.4163253E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...           .8167995E+00

PARAMETER ESTIMATES FOR PERIOD    3       ( 62 OBSERVATIONS )
   .8764314E+01    .2248324E-01    .9735627E-03  -.7167085E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...           .1845641E+00

PARAMETER ESTIMATES FOR PERIOD    4       ( 63 OBSERVATIONS )
   .1097212E+02    .3032581E-01    .8668462E-03  -.9231087E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...           .5485820E+00

PARAMETER ESTIMATES FOR PERIOD    5       ( 64 OBSERVATIONS )
   .6239987E+01    .1184804E-01    .9249539E-03  -.4621173E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...           .5567798E+00

PARAMETER ESTIMATES FOR PERIOD    6       ( 65 OBSERVATIONS )
   .4573667E+01    .5529964E-02    .9622760E-03  -.3003375E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...          -.3770107E-01

PARAMETER ESTIMATES FOR PERIOD    7       ( 66 OBSERVATIONS )
   .3122152E+01   -.8806393E-03    .9666649E-03  -.1520997E-01

PARAMETER ESTIMATES FOR PERIOD    8       ( 67 OBSERVATIONS )
   .5759550E+01    .1061990E-01    .7645089E-03  -.4138362E-01
```

```
LEAST ABSOLUTE RESIDUALS ESTIMATES

ESTIMATED COEFFICIENTS ......
        .627140E+01
        .127332E-01
        .754930E-03
       -.463892E-01

R-SQUARED =      .6002

DURBIN-WATSON =    .9669

NUMBER OF OBSERVATIONS =    68

SUM OF SQUARED RESIDUALS =      .102235E+02

STANDARD ERROR OF THE REGRESSION =      .399678E+00

SUM OF ABSOLUTE VALUES OF RESIDUALS =      .198155E+02

TOTAL FORECAST ERROR FOR A  3 PERIOD LEAD ...          .2461361E+01
```

EXAMPLE (e) - FORECAST EVALUATION WITH INTERACTION -

The following instructions illustrate the capability of the user to interact with LAR during the recursive forecast evaluation procedure with NRET = 1, the LAR routine will return to the user's program after each estimation period. At this time the user can retrieve and store any results desired from that estimation period and then call the LAR routine for the next period of estimation.

```
      DIMENSION FORSTO(10)
      NOBS=MAXLEN=68

      INDHI=63
      NSTRAT=IFOPT=IFBHAT=0
      IPSUP=1
      NRET=1
      KVAR=4
      LEAD=3
      MODE=3
      L=0
   13 CALL LAR
      L=L+1
      FORSTO(L)=FORSAV
      IF((INDHI+LEAD+L-1).LT.MAXLEN) GO TO 13
      PRINT 100, (FORSTO(I),I=1,3)
  100 FORMAT(1H0,*FORECAST ERRORS ..*,3F12.5)
```

The printed output from this run would appear as below.

```
LEAST ABSOLUTE RESIDUALS ESTIMATES

ESTIMATED COEFFICIENTS ......
       .691359E+01
       .152991E-01
       .100970E-02
      -.538676E-01

R-SQUARED =      .7212

DURBIN-WATSON =  1.2277

NUMBER OF OBSERVATIONS =   63

SUM OF SQUARED RESIDUALS =      .655631E+01

STANDARD ERROR OF THE REGRESSION =      .333353E+00

SUM OF ABSOLUTE VALUES OF RESIDUALS =      .151514E+02


PARAMETER ESTIMATES FOR PERIOD   1      ( 63 OBSERVATIONS )
   .6913593E+01   .1529915E-01   .1009701E-02  -.5386757E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...          .3341872E+00

PARAMETER ESTIMATES FOR PERIOD   2      ( 64 OBSERVATIONS )
   .5542743E+01   .1051260E-01   .1045259E-02  -.4071591E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...          .5339710E+00

PARAMETER ESTIMATES FOR PERIOD   3      ( 65 OBSERVATIONS )
   .5125984E+01   .9355635E-02   .1054627E-02  -.3696703E-01
FORECAST ERROR FOR A  3 PERIOD LEAD ...         -.1512203E-01

FORECAST ERRORS ..       .33419       .53397      -.01512
```

REFERENCES

(1) Dantzig, G. B., Linear Programming and Extensions, Princeton University Press, 1963.

(2) Charnes, A., Cooper, W, and Ferguson, R., "Optimal Estimation of Executive Compensation by Linear Programming," Management Science, 1955.

(3) Wagner, H., "Linear Programming Techniques for Regression Analysis," Journal of American Statistical Assocation, March 1959.