MULTIPLE LINEAR REGRESSION VIEWPOINTS
Volume 11, Number 1, September, 1981

PREDICTION OF MISSING DATA USING REGRESSION MODELS: A PROGRAMMED APPROACH FOR LARGE SPSS SYSTEM FILES

RON BOBNER
YOUNGSTOWN STATE UNIVERSITY
HAL STALCUP
U.S. AIR FORCE, WRIGHT-PATTERSON AFB
ISADORE NEWMAN AND CAROLYN BENZ
THE UNIVERSITY OF AKRON

Abstract

The purpose of this article is to present computer programming capable of automatically predicting missing data in an SPSS system file using multiple regression techniques. Two versions are presented. The first is an interactive approach designed to run via VSPC under OS/VS MVS. The second is designed to be used in a card batch environment. In addition, an equation which predicts the amount of CPU time required is included.

Incomplete data are often a major concern in behavioral research. SPSS, SASS, and BIOMED, three of the most commonly available statistical packages, all have provisions for handling missing data. These subroutines generally take the form of some type of case deletion. This approach, however, decreases the N size and therefore the power of the analysis. In addition, if regression methods are used, a decrease in the N size will increase the upward bias of the R (Newman, 1972) and decrease the stability of the weights (Newman, 1973).

Two other alternatives for handling missing data are sometimes used. One method is the substitution of the mean

value calculated from the complete cases. However, this technique tends to decrease the variance and may therefore be inappropriate for analysis of variance procedures. The other approach is to predict the missing values through the use of a regression equation whose weights are calculated from the complete cases. This latter approach tends to bias any subsequent analysis to a lesser extent than the insertion of means.

While neither SPSS, SASS, or BIOMED makes direct provision for the insertion of means for missing values, it is possible by utilizing various calculation and data management subroutines within the packages. By comparison, the calculation and insertion of missing data based on the regression approach, utilizing subroutines existing within the packages, is so complicated that this procedure is impractical even when dealing with small amounts of missing data.

The following programming was designed to provide a solution to this problem. Two versions of the program are presented. The first was designed to run interactively via VSPC under OS/VS MVS where VS FORTRAN, the RUML subroutine of the IMSL (International Mathematical and Statistical Libraries, Inc., 1980), and SPSS Release 8.1 (McGraw-Hill, 1979) routines have been implemented. The second version was designed for use with a card deck in a batch mode and only requires the implementation of SPSS, the IMSL, and FORTRAN. The final section of this paper presents a method for predicting the amount of CPU time that will be required to predict and insert the missing data.

Interactive Programming

The interactive programming consists of two routines designed to run consecutively: INTRO and INTRO1. They will call on SPSS system file, sort the complete from the incomplete cases, and further sort the incomplete cases into two categories: $\leq$ 10% and > 10% missing data. Next, the program will scan the first case in the $\leq$ 10% missing data file and locate the first missing variable, build a regression equation predicting this variable utilizing only those predictors available from that particular case, and calculate the weights utilizing the complete case file. These weights, along with the values for the predictors from the first case, will be used to predict the missing value for the variable. An a priori decision was made to round the predicted value up or down as appropriate if the first three digits dropped are > ±.455 of the last significant digit kept. If the first three digits dropped are in the range of < ±.455 of the last significant digit kept, the value of the last significant digit is randomly rounded up or down. This value is then inserted into the data matrix and the program then continues its scan of the first case in the $\leq$ 10% missing data file for other missing variables. If it should find another missing value the process is repeated. In no instance, however, are previously predicted variables entered into the prediction process of subsequent variables for that particular case.

Once the scan and prediction of missing variables for the first case is completed, the program repeats the process for the second and each subsequent case in the ≤ 10% missing data file. After the last case has been completed, the program combines the complete case file with newly completed version of the ≤ 10% missing data file. This new file is in BCD format and can be used as a raw data file for input into SPSS. Its name is USER FINISH CASES.

This new system file will contain complete data on all cases which initially were complete or had ≤ 10% missing data. Those cases which had > 10% missing data will have been discarded. The choice of the 10% cutoff is based on the fact that > 10% missing data for any one case will not unduly bias the prediction of the missing data (Cohen & Cohen, 1975).

The following sections document the interactive versions of INTRO and INTRO1.

```
      DIMENSION H(72),NAME(80),IFORM(80),NAMSYS(8),IP(80)
     +,NAMSUB(8)
      INTEGER YES,GO
      DATA YES,NO,GO/'Y','N','GO  '/
      ITEST=0
      CALL OPSYS('COMMAND','ALL',IRTN,IMESS,'FILE JOBOUT')
      CALL OPSYS('ALLOC','JOBOUT',9)
      CALL CLEAR
      WRITE(6,200)
  200 FORMAT(6X,'ENTRY INTO THE DATA MANAGEMENT SOFTWARE.'/
     +1X,'PURPOSE   : DATA EDITING AND CASE FILE ADDITIONS.'/
     +1X,'PROCEDURE:                                       '/
     +1X,'                1. PLACE SPSS FILE ON DISK FILE. '/
     +1X,'                2. SCAN CASES FOR COMPLETE CASES.'/
     +1X,'                3. SPLIT CASES AND DELETE )10% MISSING VALUES.'/
     +1X,'                4. USE COMPLETE CASES TO REPLACE MISSING VALUES.'/
     +1X,'                5. REWRITE SPSS FILE WITH CORRECT CASES.'//)
      CALL CLEAR
      WRITE(6,201)
  201 FORMAT(1X,'WELL, WELL RON;  HOW THE HELL ARE YOU')
      READ(5,100) IANS
  100 FORMAT(1A4)
      CALL REPLY(IANS)
    5 WRITE(6,202)
  202 FORMAT(1X,'ARE YOU READY TO DUMP THE SPSS DATA FILE???')
      CALL ANSWER(IRN)
      IF(IRN.EQ.0) GO TO 10
      IF(ITEST.EQ.99) GO TO 99
      WRITE(6,203)
```

```
300     203 FORMAT(6X,'THEN WHY DID YOU START THIS PROGRAM ?????'/
310        +1X,'I AM GOING TO ASSUME THAT YOU CAN NOT TYPE AND YOU NEED'/
320        +1X,'ANOTHER CHANCE TO ANSWER THE QUESTION.'/)
330         GO TO 5
340      10 WRITE(6,204)
350     204 FORMAT(1X,'VERY WELL THEN I AM READY TO TRANSFER THE DATA.')
360         WRITE(6,205)
370     205 FORMAT(1X,'I NEED SOME INFORMATION;')
380         WRITE(6,206)
390         READ(5,*) NVARS
400     206 FORMAT(1X,'FIRST, THE NUMBER OF VARIABLES IN THE FILE ')
410         WRITE(6,207)
420         READ(5,101) (NAME(II),II=1,55)
430     101 FORMAT(80A1)
440     207 FORMAT(1X,'NEXT, THE NAMES  OF THE VARIABLES IN THE FILE:')
450         WRITE(6,208)
460         READ(5,101) (IFORM(II),II=1,55)
470     208 FORMAT(1X,'NEXT, THE FORMAT TO READ THE VARIABLES IN THE FILE:')
480         WRITE(6,209)
490         READ(5,101) (NAMSYS(II),II=1,8)
500         WRITE(6,500)
510         READ(5,101) (NAMSUB(II),II=1,8)
520     209 FORMAT(1X,'LAST, THE NAME OF THE SPSS SYSTEM FILE TO READ ')
530     500 FORMAT(1X,'AND THE NAME OF THE SUBFILE (??):')
540         WRITE(9,210) (NAMSYS(II),II=1,8),(NAMSYS(II),II=1,8)
```

```
      210 FORMAT(
     +'//XXXXXXXX  JOB  04423,STALCUP,CLASS=A'
     +/'//*JOBPARM SKIP=YES,TIME=2'
     +/'// EXEC SPSS,DSN9=''&&OUTCASE'',UNIT9=SYSDA,'
     +/'// STATUS9=NEW,DISP9=PASS,TRK9=500,'
     +/'// DCB9=''(RECFM=FB,LRECL=400,BLKSIZE=4000)'',DSN3=',8A1
     +''GET FILE       ',8A1)
      WRITE(9,501) (NAMSUR(II),II=1,8)
  501 FORMAT(
     +'RUN SURFILES    ',8A1)
      WRITE(9,211) (IFORM(II),II=1,55),(NAME(II),II=1,55)
  211 FORMAT(
     +'WRITE CASES     ',55A1,
     +/'                ',55A1)
      WRITE(9,212) (IFORM(II),II=1,55)
  212 FORMAT(
     +'FINISH'
     +/'// EXEC FORT                                    '
     +/'       DIMENSION IDATA(200),DATA(200)'
     +/'  200 FORMAT(',55A1)
      WRITE(9,213) NVARS
  213 FORMAT(
     +'       IREC=0
     +/'       INVARS=',I5)
      WRITE(9,214)
  214 FORMAT(
     +'     5 CONTINUE                                  '
     +/'       READ(11,200,END=99) (DATA(I),I=1,INVARS)'
     +/'       DO 10 I=1,INVARS
     +/'       IDATA(I)=DATA(I)                        '
     +/'    10 CONTINUE
     +/'       WRITE(12) (IDATA(I),I=1,INVARS)
     +/'       IREC=IREC+1
     +/'       GO TO 5
```

```
      +//'    99 WRITE(6,300) IREC
      +//'   300 FORMAT(''1''///
      +//'      +25X,''   P R O C E S S I N G   E N D E D   W I T H  ''//'
      +//'      +25X,''                                               ''//'
      +//'      +25X,''   N O R M A L   R E T U R N   C O D E         ''//'
      +//'      +25X,''                                               ''//'
      +//'      +25X,''            # RECORDS=========='',I5)
      +//'       STOP                                    ')
       WRITE(9,215) NVARS
  215 FORMAT(
      +'       END
      +//'// EXEC GOFORT
      +//'//FT11F001 DD DSN=&&OUTCASE,DISP=(OLD,DELETE)'
      +//'//FT12F001 DD DSN=USER.CASE$(OUT),DISP=SHR
      +//'//*        UNIT=SYSDA,
      +//'//*        DCB=(RECFM=VBS,LRECL=800,BLKSIZE=804),'
      +//'//*        SPACE=(TRK,(200,5),RLSE),VOL=SER=ACAD01'
      +//'// EXEC FORT'
      +//'       DIMENSION IDATA(200)'
      +//'       DATA INC1,INC2,INC3,INC4/-9,-8,-7,-6/'
      +//'       IREC=0
      +//'       IREC1=0
      +//'       IREC2=0
      +//'       IFLAG=0
      +//'       NREC=',I5
      +//'     5 READ(10,END=99) (IDATA(I),I=1,NREC)'
       WRITE(7,216)
  216 FORMAT(
      +'       IREC=IREC+1
      +//'       DO 10 I=1,NREC
      +//'       IF(IDATA(I).EQ.INC1) IFLAG=IFLAG+1
      +//'       IF(IDATA(I).EQ.INC2) IFLAG=IFLAG+1
      +//'C      IF(IDATA(I).EQ.INC3) IFLAG=IFLAG+1
      +//'C      IF(IDATA(I).EQ.INC4) IFLAG=IFLAG+1
      +//'    10 CONTINUE
```

```
1200        +/.         IF(IFLAG.EQ.0) GO TO 20
1210        +/.         IF(IFLAG.LE.7) GO TO 25
1220        +/.         IFLAG=0
1230        +/.         GO TO 5
1240        +/.      20 WRITE(12) (IDATA(I),I=1,NREC)'
1250        +/.         IREC1=IREC1+1
1260         WRITE(9,217)
1270    217 FORMAT(
1280        +'          IFLAG=0
1290        +/.         GO TO 5
1300        +/.      25 WRITE(11) (IDATA(I),I=1,NREC)
1310        +/.         IREC2=IREC2+1
1320        +/.         IFLAG=0
1330        +/.         GO TO 5
1340        +/.      99 CONTINUE
1350        +/.         IREC3=IREC-(IREC1+IREC2)
1360        +/.         WRITE(6,202) IREC,IREC1,IREC2,IREC3
1370        +/.     202 FORMAT(1X,I6,' RECORDS PROCESSED'/
1380        +/.        +/1X,I6,' RECORDS COMPLETE'/
1390        +/.        +1X,I6,' RECORDS INCOMPLETE'/
1400        +/.        +/1X,I6,' RECORDS DISCARDED') ')
1410         WRITE(9,218)
1420    218 FORMAT(
1430        +'          STOP
1440        +/.         END
1450        +/.//  EXEC GOFORT'
1460        +/.//FT10F001 DD DSN=USER.CASES(OUT).DISP=SHR'
```

```
1470         +'//FT11F001  DD DSN=USER.INC.CASES,DISP=SHR
1480         +'//          UNIT=SYSDA,
1481         +'//          DCB=(RECFM=VBS,LRECL=800,BLKSIZE=804),'
1490         +'//          SPACE=(TRK,(100,5),RLSE),VOL=SER=ACAD01'
1500         +'//FT12F001  DD DSN=USER.COM.CASES,DISP=SHR '
1510         +'//          UNIT=SYSDA,
1511         +'//          DCB=(RECFM=VBS,LRECL=800,BLKSIZE=804),'
1520         +'//          SPACE=(TRK,(100,5),RLSE),VOL=SER=ACAD01')
1530      99 CONTINUE
1540         CALL CLEAR
1550         CALL OPSYS('COMMAND','ALL',IRTN,IMESS,'PF 12 ''SUB JOBOUT''   C E')
1560         CALL OPSYS('COMMAND','ALL',IRTN,IMESS,'PF 11 ''STATUS      ''  C E')
1570         CALL OPSYS('COMMAND','ALL',IRTN,IMESS,'PF 10 ''RUN INTRO1''   C E')
1580         WRITE(6,300)
1590     300 FORMAT(1X,'ALL RIGHT RON, NOW YOU NEED TO SUBMIT THE JOB: ',
1600        +/11X,'PRESS PF KEY #12.'
1610        +/11X,'TO CHECK THE STATUS OF THE JOB'
1620        +/11X,'PRESS PF KEY #11.'
1630        +/11X,'AND TO START THE NEXT PROGRAM'
1640        +/11X,'PRESS PF KEY #10.'//)
1650         STOP
1660         END
1670         SUBROUTINE CLEAR
1680         WRITE(6,200)
1690     200 FORMAT(1X,'PLEASE CLEAR YOUR SCREEN AND PRESS ENTER . . . .')
1700         READ(5,100)
1710     100 FORMAT(1A1)
1720         RETURN
```

```
      END
      SUBROUTINE REPLY(IANS)
      INTEGER*4 YES,OK,FINE,GOOD,BL,BAD
      DATA YES,NO,OK,FINE,GOOD,BL,BAD/'Y','N','OK  ',
     1'FINE','GOOD','    ','BAD '/
      IF(IANS.EQ.FINE) GO TO 20
      IF(IANS.EQ.GOOD) GO TO 30
      IF(IANS.EQ.BAD) GO TO 40
      IF(IANS.EQ.OK) GO TO 50
      WRITE(6,500)
      GO TO 99
   20 WRITE(6,100)
      GO TO 99
   30 WRITE(6,200)
      GO TO 99
   40 WRITE(6,300)
      GO TO 99
   50 WRITE(6,400)
   99 CONTINUE
  100 FORMAT(1X,'JUST FINE, NOW WHAT KIND OF ANSWER IS THAT ????')
  200 FORMAT(1X,'WELL, WE WILL SEE ABOUT CHANGING THAT.')
  300 FORMAT(1X,'SORRY TO HEAR THAT, PLEASE DON'T TAKE IT OUT ON ME.')
  400 FORMAT(1X,'WELL DON'T COMMIT YOURSELF TO A POSITIVE ANSWER.')
  500 FORMAT(1X,'NEXT TIME TRY HARDER, THINK OF A BETTER ANSWER.')
      RETURN
      END
      SUBROUTINE ANSWER(IRN)
      INTEGER YES
      DATA YES,NO/'Y','N'/
      WRITE(6,200)
  200 FORMAT(1X,'(Y/N):')
   10 READ(5,100) IANS
  100 FORMAT(1A1)
      IF(IANS.EQ.YES) GO TO 20
      IF(IANS.EQ.NO) GO TO 30
      WRITE(6,201)
```

```
2090   201 FORMAT(1X,'PLEASE ANSWER Y OR N........')
2100       GO TO 10
2110    20 IRN=0
2120       GO TO 99
2130    30 IRN=1
2140    99 CONTINUE
2150       RETURN
2160       END
```

INTRO1


```
  10       DIMENSION N(72),NAME(80),IFORM(80),NAMSYS(8),IP(80)
  20       INTEGER YES,GO
  30       DATA YES,NO,GO/'Y','N','GO  '/
  31       WRITE(6,200)
  32   200 FORMAT(1X,'READY?    IF SO TYPE THE WORD GO.')
  33   100 FORMAT(1A4)
1610    50 READ(5,100) IANS
1620       IF(IANS.EQ.GO) GO TO 55
1630       GO TO 50
1640    55 CONTINUE
1650       CALL OPSYS('COMMAND','ALL',IRTN,IMESS,'FILE JOBOUT')
1660       CALL OPSYS('ALLOC','2078 REPLACE',10)
1661       CALL OPSYS('ALLOC','JOBOUT',9)
1670       REWIND 9
1671       DO 60 J=1,4
1672   110 FORMAT(80A1)
1680       READ(10,110) (IP(I),I=1,80)
1690       WRITE(9,110) (IP(I),I=1,80)
```

```
1700     60 CONTINUE
1701        WRITE(6,201)
1702    201 FORMAT(1X,'ENTER THE NUMBER OF VARIABLES: ')
1703        READ(5,*) NVARS
1710        WRITE(9,303) NVARS
1720    303 FORMAT(1X,'       NVARS=',I5)
1731     65 CONTINUE
1732        READ(10,110,END=88) (IP(I),I=1,80)
1740        WRITE(9,110) (IP(I),I=1,80)
1750        GO TO 65
1760     88 CALL CLEAR
1780        WRITE(6,304)
1781    304 FORMAT(//1X,'NOW THE JOB FOR THE MLR IS READY.'
1782       +//11X,'PRESS PF KEY # 12 TO SUBMIT THE JOB'
1783       +//11X,'PRESS PF KEY # 11 TO CHECK THE STATUS OF THE JOB')
1929        WRITE(6,305)
1930    305 FORMAT(//1X,'CONTROL RETURNED TO YOU, OVER AND OUT.....')
2030        STOP
2040        END
2050        SUBROUTINE CLEAR
2060        WRITE(6,200)
2070    200 FORMAT(1X,'PLEASE CLEAR YOUR SCREEN AND PRESS ENTER.....')
2080        READ(5,100)
2090    100 FORMAT(1A1)
2100        RETURN
2110        END
2120        SUBROUTINE REPLY(IANS)
2130        INTEGER*4 YES,OK,FINE,GOOD,BL,BAD
2140        DATA YES,NO,OK,FINE,GOOD,BL,BAD/'Y','N','OK  ',
2150       +'FINE','GOOD','    ','BAD '/
2160        IF(IANS.EQ.FINE) GO TO 20
2170        IF(IANS.EQ.GOOD) GO TO 30
2180        IF(IANS.EQ.BAD) GO TO 40
2190        IF(IANS.EQ.OK) GO TO 50
2200        WRITE(6,500)
```

```
      GO TO 99
   20 WRITE(6,100)
      GO TO 99
   30 WRITE(6,200)
      GO TO 99
   40 WRITE(6,300)
      GO TO 99
   50 WRITE(6,400)
   99 CONTINUE
  100 FORMAT(1X,'JUST FINE, NOW WHAT KIND OF ANSWER IS THAT ????')
  200 FORMAT(1X,'WELL, WE WILL SEE ABOUT CHANGING THAT.')
  300 FORMAT(1X,'SORRY TO HEAR THAT, PLEASE DON'T TAKE IT OUT ON ME')
  400 FORMAT(1X,'WELL DON'T COMMIT YOURSELF TO A POSITIVE ANSWER. ')
  500 FORMAT(1X,'NEXT TIME TRY HARDER, THINK OF A BETTER ANSWER.')
      RETURN
      END
      SUBROUTINE ANSWER(IRN)
      INTEGER YES
      DATA YES,NO/'Y','N'/
      WRITE(6,200)
  200 FORMAT(1X,'(Y/N):')
   10 READ(5,100) IANS
  100 FORMAT(1A1)
```

```
      IF(IANS.EQ.YES) GO TO 24
      IF(IANS.EQ.NO) GO TO 30
      WRITE(6,201)
  201 FORMAT(1X,'PLEASE ANSWER Y OR N..... ')
      GO TO 10
   20 IRN=0
      GO TO 99
   30 IRN=1
   99 CONTINUE
      RETURN
      END
```

Batch Version

Both INTRO and INTRO1 are documented below in a form suitable for batch entry via cards. The cards which must be changed in each instance are noted.

```
/*JOBPARM SKIP=YES,TIME=2
// EXEC SPSS,DSN9='  OUTCASE',UNIT9=SYSDA,
// STATUS9=NEW,DISP9=PASS,TRK9=600,
// DCB9='(RECFM=FB,LRECL=400,BLKSIZE=4000)',DSN3=ABOD
GET FILE      ABOD
RUN SUBFILES  S10
WRITE CASES   (67F3.0)
              SCHOOL,ITEM01,ITEM66,AGE,GRADE
FINISH
// EXEC FORT
      DIMENSION IDATA(200),DATA(200)
  200 FORMAT(67F3.0)
      IREC=0
      INVARS=  67
    5 CONTINUE
      READ(11,200,END=99)  (DATA(I),I=1,INVARS)
      DO 10 I=1,INVARS
      IDATA(I)=DATA(I)
   10 CONTINUE
      WRITE(12)  (IDATA(I),I=1,INVARS)
      IREC=IREC+1
      GO TO 5
   99 WRITE(6,300)  IREC
  300 FORMAT('1'///
     +25X,'   P R O C E S S I N G   E N D E D   W I T H  '/
     +25X,'                                              '/
     +25X,'   N O R M A L   R E T U R N   C O D E       '/
     +25X,'                                              '/
     +25X,'              RECORDS=========>',I5)
      STOP
      END
```

Computer Center

Initial SPSS file

Format of variables to be read from INITIAL SPSS file

Number of variables to be read from INITIAL SPSS file

```
// EXEC GOFORT
//FT11F001 DD DSN=  OUTCASE,DISP=(OLD,DELETE)
//FT12F001 DD DSN=USER.CASES(OUT),DISP=SHR
//*        UNIT=SYSDA,
//*        DCB=(RECFM=FBS,LRECL=800,BLKSIZE=804),
//*        SPACE=(TRK,(200,5),RLSE),VOL=SER=ACAD01
// EXEC FORT
      DIMENSION IDATA(200)
      DATA INC1,INC2,INC3,INC4/-9,-8,-7,-6/
      IREC=0
      IREC1=0
      IREC2=0
      IFLAG=0
      NREC=   67
    5 READ(10,END=99) (IDATA(I),I=1,NREC)
      IREC=IREC+1
      DO 10 I=1,NREC
      IF(IDATA(I).EQ.INC1) IFLAG=IFLAG+1
      IF(IDATA(I).EQ.INC2) IFLAG=IFLAG+1
C     IF(IDATA(I).EQ.INC3) IFLAG=IFLAG+1
C     IF(IDATA(I).EQ.INC4) IFLAG=IFLAG+1
   10 CONTINUE
      IF(IFLAG.EQ.0) GO TO 20
      IF(IFLAG.LE.7) GO TO 25
```

Computer Center VCL

```
      IFLAG=0
      GO TO 5
   20 WRITE(12) (IDATA(I),I=1,NREC)
      IREC1=IREC1+1
      IFLAG=0
      GO TO 5
   25 WRITE(11) (IDATA(I),I=1,NREC)
      IREC2=IREC2+1
      IFLAG=0
      GO TO 5
   99 CONTINUE
      IREC3=IREC-(IREC1+IREC2)
      WRITE(6,202) IREC,IREC1,IREC2,IREC3
  202 FORMAT(1X,I6,' RECORDS PROCESSED'
     +/1X,I6,' RECORDS COMPLETE'/
     +1X,I6,' RECORDS INCOMPLETE'/
     +/1X,I6,' RECORDS DISCARDED')
      STOP
      END
```

```
// EXEC GOFORT
//FT10F001 DD DSN=USER.CASES(OUT),DISP=SHR
//FT11F001 DD DSN=USER.INC.CASES,DISP=SHR
//*         UNIT=SYSDA,
//*         DCB=(RECFM=VBS,LRECL=800,BLKSIZE=804),
//*         SPACE=(TRK,(100,5),RLSE),VOL=SER=ACAD01
//FT12F001 DD DSN=USER.COM.CASES,DISP=SHR
//*         UNIT=SYSDA,
//*         DCB=(RECFM=VBS,LRECL=800,BLKSIZE=804),
//*         SPACE=(TRK,(100,5),RLSE),VOL=SER=ACAD01

//XXXXXXXX  JOB  04423,STALCUP,CLASS=C
/*JOBPARM SKIP=YES,TIME=15
// EXEC FORT
      COMMON /COM1/ INCOM(200)
       NVARS=  67
      I=0
      NNNN=NVARS-1
   10 READ(10,END=99)   ISC,(INCOM(J),J=1,NNNN)
      WRITE(6,201) ISC,(INCOM(J),J=1,NNNN)
      CALL REP(NVARS,ISC)
      CALL WRITE(NVARS,ISC)
      I=I+1
      GO TO 10
   99 CONTINUE
      WRITE(6,200) I
  200 FORMAT('1','  J O B   S U M M A R Y',
     +//1X,'JOB FINISHED;  COMPLETION CODE=0'
     +/1X,'NUMBER OF CASES=',I5)
  201 FORMAT(1X,'INCOMPLETE CASE LISTED BELOW:',
     +50(/1X,30I4))
       STOP
       END
```

Computer Center VCC

Number of variables

```
C.....................................................................
      SUBROUTINE REP(NREC,ISC)
      DIMENSION XX(1350,200),TEMP(200),
     +B(200,7),VARS(200),IREP(200),NRR(6),
     +INDEX(200)
      COMMON /COM1/INDATA(200)
      COMMON /COM2/ISUB(1350,200)
      COMMON /COM3/IFILE(1350,200)
      COMMON /COM4/INFILE(200)
      ICOMP=1
      IT=ISC
      NNREC=NREC-1
      REWIND 11
   10 READ(11,END=99)  ISC,(INFILE(I),I=1,NNREC)
C     WRITE(6,777)  ISC,(INFILE(I),I=1,NNREC)
  777 FORMAT(1X,3013/)
      IF(IT.EQ.ISC) GO TO 20
      GO TO 10
   20 CONTINUE
      J=0
      L=0
C.................      J = COUNTER FOR INCOMPLETE VARS
C.................      L = COUNTER FOR COMPLETE VARS
C.................   IREP = INDEX OF LOCATION FOR COMPLETE VARS
C.................  INDEX = INDEX OF LOCATION FOR INCOMPLETE VARS
C.................  IFILE = VALUES OF THE DEPENDENT VARS (COMPLETE)
C.................   ISUB = VALUES OF THE INDEPENDENT VARS (INCOMPLETE)
```

```
      DO 30 I=1,NNREC
      IF(INDATA(I).EQ.-9.OR.INDATA(I).EQ.-8) GO TO 40
      GO TO 50
   40 J=J+1
      INDEX(J)=I
      GO TO 30
   50 CONTINUE
      L=L+1
      INREP(L)=I
   30 CONTINUE
      M=0
   35 M=M+1
      DO 60 K=1,J
      IFILE(M,K)=INFILE(INDEX(K))
   60 CONTINUE
      DO 70 K=1,L
      ISUB(M,K)=INFILE(INREP(K))
   70 CONTINUE
      READ(11,END=88)   ISC,(INFILE(LL),LL=1,NNREC)
      IF(IT.EQ.ISC) GO TO 35
      LL=L+J
      L1=L+1
      M1=L+1
      NTM=M1+M1
C     WRITE(6,888)
      CALL SUB(J,L,M,NVARS,IREP,INDEX,XY,0,LL,L1,M1)
      RETURN
   88 CONTINUE
C     WRITE(6,666)
```

```
  555 FORMAT(1X,'=REC=',I5)
  666 FORMAT(1X,'CALL AT END')
  888 FORMAT(1X,'CALL AT MID')
      LL=L+J
      L1=L+1
      M1=L+1
      MTH=M1*M1
      CALL SUB(J,L,M,NVARS,IREP,INDEX,XY,B,LL,L1,M1)
   99 CONTINUE
      RETURN
      END
C.....
      SUBROUTINE SUB(J,L,M,NVARS,IREP,INDEX,XY,B,LL,L1,M1)
      DIMENSION XY(M,M1),TEMP(200),
     *XYBAR(200),A(5400),ANOVA(14),B(M1,7),VARB(2100),IREP(200),
     *RHR(6),INDEX(200)
      COMMON /COM1/INDATA(200)
      COMMON /COM2/ISUB(1350,200)
      COMMON /COM3/IFILE(1350,200)
      COMMON /COM4/INFILE(200)
C.....      J = COUNTER FOR INCOMPLETE VARS
C.....      L = COUNTER FOR COMPLETE VARS
C.....      M = COUNTER FOR NUMBER OF CASES
C.....   IREP = INDEX OF LOCATION FOR COMPLETE VARS
C.....  INDEX = INDEX OF LOCATION FOR INCOMPLETE VARS
C.....  IFILE = VALUES OF THE DEPENDENT VARS (COMPLETE)
C.....   ISUB = VALUES OF THE INDEPENDENT VARS (INCOMPLETE)
C     WRITE(6,666) M,L,J
```

```
  666 FORMAT(1X,3I10/20(20I5/))
      DO 10 LL=1,M
      DO 10 KK=1,L
      IY(LL,KK)=ISUB(LL,KK)
   10 CONTINUE
C     WRITE(6,666)  M,L,J,(IFILE(LL,1),LL=1,M)
      ICON=0
    5 CONTINUE
      ICON=ICON+1
      IF(ICON.GT.J) GO TO 999
      KK=L+1
      DO 20 LL=1,M
      IY(LL,KK)=IFILE(LL,ICON)
   20 CONTINUE
C     WRITE(6,666)  KK,ICON,LL
      MM=M
      NN=L
      II=M
      MBR(1) = NN + 1
      MBR(2) = MM
      MBR(3) = MM
      MBR(4) = 1
```

```
      NBR(5) = 1
      NBR(6) = 1
      ALPA = 0.10
      IB = 81
C     WRITE(6,555)
  555 FORMAT(1X,'JUST BEFORE THE CALL',I10)
      CALL RECOVR (IY,IX,NBR,TEMP,IYBAR,A,IER)

      CALL RLSBL (A,IYBAR,BB,NR,ALPA,AMOVA,B,IB,VARR,IER)
C     WRITE(6,666) B(1,1),B(NR+1,1)
      SUM=B(NR+1,1)
      DO 30 II=1,NR
      SUM=SUM+B(II,1)*INDATA(IREP(II))
   30 CONTINUE
      CALL ROUND(SUM,IS)
      INDATA(INDEX(ICOR))=IS
      GO TO 5
  999 CONTINUE
      RETURN
      END
C.....
      SUBROUTINE ROUND(I,I)
      I=I
      II=I*100
      II=I*100
      Y=II-II
      IF(Y.GE.45.0.AND.Y.LE.55.0) GO TO 10
      IF(Y.LT.45.0) RETURN
      IF(Y.GT.55.0) I=I+1
      RETURN
```

```
  200 FORMAT(1X,'FINISHED CASE LISTED BELOW:',
     +50(/1X,30I4))
         RETURN
         END
// EXEC GOFORT,GOSIZE=3500K
//FT10F001 DD DSN=USER.INC.CASES,DISP=SHR
//FT11F001 DD DSN=USER.COM.CASES,DISP=SHR
//FT16F001 DD DSN=USER.MAL01,DISP=SHR
// EXEC FORT
         DIMENSION M(80)
         K=0
       1 READ(12,END=88) (M(J),J=1,65)
         WRITE(16,100) (M(I),I=1,65)
         K=K+1
         GO TO 1
      88 CONTINUE
       5 READ(13,100,END=99) (M(I),I=1,65)
         WRITE(16,100) (M(I),I=1,65)
         K=K+1
         GO TO 5
      99 CONTINUE
         WRITE(6,201) K
     100 FORMAT(80I3)
     200 FORMAT(/1X,3(1X,30I3/))
     201 FORMAT(///1X,'  OF CASES ADDED=',I6)
         STOP
         END
// EXEC GOFORT
//FT12F001    DD DSN=USER.COM.CASES,DISP=SHR
//FT13F001    DD DSN=USER.MAL01,DISP=SHR
//FT16F001    DD DSN=USER.FINISH.CASES,DISP=(MOD,KEEP)
//
```

Number of variables

Number of variables

Prediction of Time Required

The programming requires fairly large amounts of CPU time when dealing with large files. The total time required is a function of the number of variables, the number of cases, the number of cases with missing data, and the interaction between the number of missing cases and the total number of cases.

To date, run time data have been gathered on 23 files which ranged in size from an N = 122 to N = 1,081. The number of missing data cases ranged from N = 12 to N = 191. All the files had 67 variables. Required CPU times ranged from 2.145 minutes to 203.072 minutes.

This information was used to build to test a regression equation which could be used to predict the CPU time required. The R^2 = .9960 had a $p < .00000$, but the equation is limited by the fact that all the test files had 67 variables. Even considering this limitation, the following regression equation may be useful in predicting CPU time.

$$CPUT = (.22248840)u + (.00779903)NC + (.08595389)NM + (.00100721)NC * NM + E$$

Where:

CPUT = CPU time required
U = Unit vector (+)
NC = Number of cases
NM = Number of cases with $\geq$ 10% missing data
E = Error vector (ignored when predicting CPUT)

References

Cohen, J., & Cohen, P. Applied multiple regression/correlation analysis for the behavioral sciences. New York: John Wiley & Sons, 1975.

International Mathematical and Statistical Libraries, Inc. The IMSL library, "RLMUL" (vol. 3). Houston, Texas: International Mathematical and Statistical Libraries, Inc., June 1980.

Newman, I. Variations between shrinkage estimation formulas and the appropriateness of their interpretation. Multiple Linear Regression Viewpoints, 1973, 4, 45-48.

Newman, I., & Fry, J. A response to "A note on multiple comparisons" and comment on shrinkage. Multiple Linear Regression Viewpoints, 1972, 2, 36-39.