United States Department of Agriculture

National Agricultural Statistics Service

Research and Applications Division

SRB Research Report Number SRB-89-12

October 1989

A COMPUTER ALGORITHM FOR MARKOV CHAIN FORECASTS OF COTTON OBJECTIVE YIELD

James H. Matis
Charles R. Perry
Donald E. Boudreaux

A COMPUTER ALGORITHM FOR MARKOV CHAIN FORECASTS OF COTTON OBJECTIVE YIELD

by James H. Matis*, Charles R. Perry and Donald E. Boudreaux*, National Agricultural Statistics Service, U.S. Department of Agriculture, Washington, D.C. 20250, October 1989, Research Report No. SRB 89–12.

Abstract

This paper documents the computer algorithm developed by Matis, Perry, Boudreaux and Aune (1989) in evaluating a revised Markov chain procedure for forecasting final cotton objective yield. The algorithm was executed in three programs. Each program's function is summarized by a statement of purpose, a procedure outline, and a few comments. The complete code along with a detailed annotation is provided for each program.

* Dr. Matis and Mr. Boudreaux are with Texas A&M University, Department of Statistics, College Station, Texas 77843.

Keywords: Markov Chain, Cross Validation, Forecast Errors, Objective Yield.

Acknowledgements

The authors express their appreciation to Ben Klugh and George Hanuschak for their helpful suggestions and moral support during this project. We express our gratitude to our colleagues Bill Donaldson, Barry Ford, and Phil Kott for their thoughtful reviews of this report. However, we bear full responsibility for any errors.

Contents

Summary

Existing computer programs were revised and expanded to accomplish two objectives. The first objective was to implement a new procedure for defining the categorical Markov transition states. The second objective was to automate the variable selection procedure so that the predictor variables are determined solely on the basis of the statistical evidence from the data at the time of forecast.

The revised algorithm was divided into five basic steps: read and edit the data, select the predictor variables, create the new categorical states, calculate the transition matrices, and generate the forecasts and estimate the forecast errors. The five steps were executed in three computer programs. Each program is described by a statement of purpose, a procedure outline, and a few comments. The complete Statistical Analysis System (SAS) code is given for each program along with a detailed annotation.

Introduction

This paper provides documentation for the algorithm and programs developed by Matis, Perry, Boudreaux and Aune (1989) in evaluating a revised Markov chain procedure for forecasting final cotton objective yield. Computer programs existing from earlier related cooperative research between the National Agricultural Statistics Service (NASS) and Texas A and M University (TAMU) (Matis, *et. al.*, 1985, 1989) were revised and expanded to accomplish two objectives. The first was to implement the new procedure for defining states. The second was to automate the program so that the primary and secondary predictor variables for each forecast were determined without human intervention. Previously these predictor variables were selected on the basis of expert judgment of the user. The revised procedures select the variables based solely on the statistical evidence available at the time of each forecast.

Computer Algorithm Overview

The process of constructing and evaluating Markov forecasts was accomplished in a five steps.

1. Read and edit the data.
2. Select independent variables.
3. Create new categorical variables based upon the selected variables and user defined number of breaks.
4. Calculate the Markov transition matrices.
5. Generate the estimated forecasts and check the accuracy of the simulation.

These steps were executed within three SAS programs – XDAT (step 1.), XRSQ (step 2.), and YRXX (steps 3., 4., 5.). Each program is described by its basic purpose, a overview of the procedure, a detailed annotated program outline, and a few comments. The complete SAS code is provided for each program. However, it should be pointed out that the programs were written for researching the feasibility of utilizing the Markov chain forecast procedure on USDA/NASS data. Thus, the programs are not necessarily optimally coded nor are they intended for "production" use. Furthermore, the programs were written in an older mainframe version of SAS, therefore they would have to be updated for use with PC-SAS or newer versions of mainframe SAS. This is especially evident with the replacement of PROC MATRIX with PROC IML.

Program: XDAT

Purpose:

This program edits the USDA/NASS cotton objective yield SAS data sets (from tape) for consistency among the values of the various variables, create new variables, and develop a data structure appropriate for Markov analysis.

Procedure:

The original tapes contained a series of SAS data sets each representing a single year. These yearly data sets were read and processed to identify and categorize the monthly sequence of information associated with each individual plot. A unique identification number, ID, and month variable, MONTH, were created. Some data items were combined, edited, and used to create new variables. Other data sets were created and merged to obtain an overall cumulative yield variable. Certain independent variables were then ranked and a sequence of data steps were executed to organize and rename the independent variables by month (8,9, and 10). Finally, these monthly data sets were recombined and a SAS data set was created on a mainframe disk pack.

Program Outline:

Lines 001–009 JCL card for SAS batch processing.

Lines 010–013 titles to document SAS output.

Lines 014–026 macro _MKMO determines state and year it will also begin the process of creating the variable MONTH.

Lines 027–035 read 1980 data and create MONTH and ID.

Lines 036–044 read 1981 data and create MONTH and ID.

Lines 045–053 read 1982 data and create MONTH and ID.

Lines 054–062 read 1983 data and create MONTH and ID.

Lines 063–071 read 1984 data and create MONTH and ID.

Lines 072–080 read 1985 data and create MONTH and ID.

Lines 081–089 read 1986 data and create MONTH and ID.

Lines 090–137 combine 80 to 86 data sets, edit, and create new variables (note cumulative variables BURR, OPEN, and YLD).

Lines 138–142 segment out the final cumulative yield.

Lines 143–151 recombine the final yield for each observation and select independent variables to keep.

Lines 152–160 rank the independent variables.

Lines 161–170 segment out month 8 and rename variables.

Lines 171–180 segment out month 9 and rename variables.

Lines 181–190 segment out month 10 and rename variables.

Lines 191–197 recombine the data sets for months 8,9, and 10 then save the result to mainframe disk space.
Line 198 printout 20 observations to check data.

Comments:

The creation of the variables MONTH and ID was necessary because of the way the data was accumulated and structured for analysis. The macro (Lines 014–026) and data set steps (Lines 027–089) used in creating these variables are not efficient procedures. They were employed to overcome a SAS problem utilizing a "by" statement within the macro language. Attention should be given to the use of SAS's automatic data set naming convention within this particular set on previous page of data steps and procedures (Lines 031, 040, 049, 058, 067, 076, 084).

SAS Code follows:

```
001  //XDAT   JOB (C774,2C,2,25,JM),'BOUDREAUX',
002  //       MSGCLASS=Z,MSGLEVEL=(0,0)
003  // EXEC SAS,OPTIONS=MACRO,REGION=4096K
004  //TAPE   DD DSN=USR.E413.JM.USDA1.INDEX1.YANG,DISP=SHR
005  //DSK    DD DSN=USR.E413.JM.STATE48,
006  //          DISP=(NEW,CATLG,DELETE),
007  //          UNIT=SYSDA,
008  //          SPACE=(TRK,(100,20))
009  //SYSIN DD *

010  TITLE1 ' MARKOV PROJECT                 ' ;
011  TITLE2 ' DATA FILTER PROC (NEW)         ' ;
012  TITLE3 ' AUG 88, MATIS & BOUDREAUX      ' ;
013  TITLE4 ' STATE=48(ALL)  YRS 81-84       ' ;

014  %MACRO _MKMO ;
015       IF STATE = 48 ;
016       IF YEAR = 0 THEN DELETE ;
017       IF YEAR = 5 THEN DELETE ;
018       IF YEAR = 6 THEN DELETE ;
019       N = _N_ ;
020       PROC SORT ;
021            BY SAMPLE ;
022       PROC RANK ;
023            BY SAMPLE ;
024            VAR N ;
025            RANKS RNK ;
026  %MEND _MKMO ;

027  DATA X80 ;
028       SET TAPE.CTWK80 ;
029       %_MKMO ;
030  DATA Y80 ;
031       SET DATA1 ;
```

```
          ID = 80000 + SAMPLE ;
          MONTH = RNK + 7 ;
          IF MONTH GE 14 THEN
             DELETE ;

     DATA X81 ;
          SET TAPE.CTWK81 ;
          %_MKMO ;
     DATA Y81 ;
          SET DATA2 ;
          ID = 81000 + SAMPLE ;
          MONTH = RNK + 7 ;
          IF MONTH GE 14 THEN
             DELETE ;

     DATA X82 ;
          SET TAPE.CTWK82 ;
          %_MKMO ;
     DATA Y82 ;
          SET DATA3 ;
          ID = 82000 + SAMPLE ;
          MONTH = RNK + 7 ;
          IF MONTH GE 14 THEN
             DELETE ;

     DATA X83 ;
          SET TAPE.CTWK83 ;
          %_MKMO ;
     DATA Y83 ;
          SET DATA4 ;
          ID = 83000 + SAMPLE ;
          MONTH = RNK + 7 ;
          IF MONTH GE 14 THEN
             DELETE ;

     DATA X84 ;
          SET TAPE.CTWK84 ;
          %_MKMO ;
     DATA Y84 ;
          SET DATA5 ;
          ID = 84000 + SAMPLE ;
          MONTH = RNK + 7 ;
          IF MONTH GE 14 THEN
             DELETE ;

     DATA X85 ;
          SET TAPE.CTWK85 ;
          %_MKMO ;
     DATA Y85 ;
          SET DATA6 ;
```

```
        ID = 85000 + SAMPLE ;
        MONTH = RNK + 7 ;
        IF MONTH GE 14 THEN
           DELETE ;

DATA X86 ;
        SET TAPE.CTWK86 ;
        %_MKMO ;
DATA Y86 ;
        SET DATA7 ;
        ID = 86000 + SAMPLE ;
        MONTH = RNK + 7 ;
        IF MONTH GE 14 THEN
           DELETE ;

DATA DSO ;
        SET Y80 Y81 Y82 Y83 Y84 Y85 Y86 ;
        IF (C509 = 0) THEN
           CONFACT = 0 ;
        ELSE
           CONFACT = C510 / C509 ;
        CURRWT = 1.0526 * CONFACT *
        (C316+C317+C325+C326+C327+C336+C337+C345+C346+C347) ;
        ROWSP = (C303+C304) / 8 ;
        IF (MONTH = 8) THEN DO ;
           BURR = C312+C322+C332+C342;
           OPEN = C313+C314+C323+C324+C333+C334+C343+C344 ;
           IF (OPEN > 0 AND CURRWT = 0) THEN
              CURRWT = . ;
           CUMWT = CURRWT ;
           IF (ROWSP = 0) THEN
              ROWSP = 3.225 ;
           END ;
        ELSE DO ;
           BURR = C312+C322+C332+C342+ BURR ;
           OPEN = C313+C314+C323+C324+C333+C334+C343+C344+OPEN ;
           IF (OPEN > 0 AND CURRWT = 0) THEN
              CURRWT = . ;
           CUMWT = CUMWT+CURRWT ;
           IF (ROWSP = 0 OR ROWSP = .) THEN
              ROWSP = ROWXX ;
           END ;
        ROWXX = ROWSP ;
        IF (OPEN = 0) THEN
           Z21 = 0 ;
        ELSE
           Z21 = CUMWT / OPEN ;
        UNOPB = C319+C329+C339+C349 ;
        PARTB = C318+C328+C338+C348 ;
        LB = BURR+OPEN +PARTB +UNOPB ;
```

```
      P3 = C350+C365 ;
      P  = C311 + C321 + C331 + C341 ;
      IF (P = 0) THEN
         MATUR = 0 ;
      ELSE
         MATUR = LB / P ;
      X1 = (0.870 * LB) + (0.867 * (C366+C367)) ;
      X2 = (6.667 * C368) ;
      X3 = 6.667*(C364+C374) ;
      MO_YLD = 2.401 * 0.368 * CUMWT / ROWSP ;
      IF (MO_YLD NE .) THEN
         YLD = MO_YLD ;
      RETAIN BURR OPEN CUMWT ROWXX YLD ;

DATA DSF ;
      SET DS0 ;
      IF MONTH = 13 ;
      YIELD = YLD ;
      KEEP ID YIELD ;

DATA DS2 ;
      MERGE DS0 DSF ; BY ID ;
      IF (YIELD = 0 OR YIELD = .) THEN
         DELETE ;
      KEEP    BURR    OPEN    UNOPB   PARTB   LB
              X1      X2      X3      P       P3
              MATUR   CONFACT CURRWT  CUMWT   ROWSP
              Z21     MO_YLD  C380    MONTH   STATE
              YEAR    YLD     YIELD   ID ;

PROC RANK DATA=DS2 OUT=DSRK ;
      VAR     BURR    OPEN    UNOPB   PARTB   LB
              X1      X2      X3      P       P3
              MATUR   CONFACT CURRWT  CUMWT   ROWSP
              Z21     C380    YIELD ;
      RANKS   RBURR    ROPEN    RUNOPB   RPARTB   RLB
              RX1      RX2      RX3      RP       RP3
              RMATUR   RCONFACT RCURRWT  RCUMWT   RROWSP
              RZ21     RC380    RYIELD ;

DATA M08 ;
      SET DSRK ;
      IF MONTH = 8 ;
      RBUR_8 = RBURR    ; ROPEN_8 = ROPEN   ; RUNO_8  = RUNOPB ;
      RPRT_8 = RPARTB   ; RLB_8   = RLB     ; RX1_8   = RX1 ;
      RX2_8  = RX2      ; RX3_8   = RX3     ; RP_8    = RP ;
      RP3_8  = RP3      ; RMAT_8  = RMATUR  ; RCON_8  = RCONFACT ;
      RCUR_8 = RCURRWT  ; RCUM_8  = RCUMWT  ; RROW_8  = RROWSP ;
      RZ21_8 = RZ21     ; RC380_8 = RC380   ;
      KEEP ID RBUR_8 -- RC380_8 ;
```

```
DATA M09 ;
      SET DSRK ;
      IF MONTH = 9 ;
      RBUR_9 = RBURR     ; ROPEN_9 = ROPEN    ; RUNO_9   = RUNOPB ;
      RPRT_9 = RPARTB    ; RLB_9   = RLB      ; RX1_9    = RX1 ;
      RX2_9  = RX2       ; RX3_9   = RX3      ; RP_9     = RP ;
      RP3_9  = RP3       ; RMAT_9  = RMATUR   ; RCON_9   = RCONFACT ;
      RCUR_9 = RCURRWT   ; RCUM_9  = RCUMWT   ; RROW_9   = RROWSP ;
      RZ21_9 = RZ21      ; RC380_9 = RC380    ;
      KEEP ID RBUR_9 -- RC380_9 ;

DATA M10 ;
      SET DSRK ;
      IF MONTH = 10 ;
      RBUR_10= RBURR     ; ROPEN_10= ROPEN    ; RUNO_10 = RUNOPB ;
      RPRT_10= RPARTB    ; RLB_10  = RLB      ; RX1_10  = RX1 ;
      RX2_10 = RX2       ; RX3_10  = RX3      ; RP_10   = RP ;
      RP3_10 = RP3       ; RMAT_10 = RMATUR   ; RCON_10 = RCONFACT ;
      RCUR_10= RCURRWT   ; RCUM_10 = RCUMWT   ; RROW_10 = RROWSP ;
      RZ21_10= RZ21      ; RC380_10= RC380    ;
      KEEP YEAR YIELD ID RBUR_10 -- RC380_10 ;

PROC SORT DATA=M08 ; BY ID ;
PROC SORT DATA=M09 ; BY ID ;
PROC SORT DATA=M10 ; BY ID ;
DATA DSK.STATE482 ;
      MERGE M08 M09 M10 ; BY ID ;
      IF (YIELD = .) THEN
         DELETE ;
PROC PRINT DATA = DSK.STATE482 (OBS = 20) ;
```

Program: XRSQ

Purpose:

This program uses the data created in XDAT to select the two best variables for predicting yield.

Procedure:

Given the nature of the study, one specific year was always excluded to allow the remaining data to be used to simulate a prediction. Then a PROC RSQUARE was run for each of the monthly time frames (8,9, and 10). The resulting set of "best" two variable models were used later in the Markov process.

Program Outline:

Lines 001–004 are JCL cards for batch processing.
Lines 005–007 are titles that will document the SAS output.
Lines 008–010 segment the year 1981 out of the data (as an example).

Lines 011–013 select the model for month 8.
Lines 014–016 select the model for month 9.
Lines 017–019 select the model for month 10.

Comments:

This process provides for an objective methodology of variable selection. However, it does assume that an "acceptable" set of independent variables are utilized.

SAS Code follows:

```
001  //RQ81   JOB (C774,2C,1,25,JM),'BOUDREAUX'
002  // EXEC SAS,REGION-1024K
003  //DSK    DD DSN-USR.E413.JM.STATE48,DISP-SHR
004  //SYSIN DD *

005  TITLE1 ' Variable Selection          ' ;
006  TITLE2 ' STATE - 48(ALL)             ' ;
007  TITLE3 ' YEARS - 82,83,84            ' ;

008  DATA DS_ONE ;
009       SET DSK.STATE482 ;
010       IF (YEAR NE 1) ;

011  PROC RSQUARE DATA-DS_ONE START-2 STOP-2 SELECT-1 CP ;
012       MODEL YIELD - ROPEN_8   RUNO_8   RLB_8   RCUR_8
013                     RX3_8     RP_8     RMAT_8  RZ21_8   RCUM_8 ;

014  PROC RSQUARE DATA-DS_ONE START-2 STOP-2 SELECT-1 CP ;
015       MODEL YIELD - ROPEN_9   RUNO_9   RLB_9   RCUR_9
016                     RX3_9     RP_9     RMAT_9  RZ21_9   RCUM_9 ;

017  PROC RSQUARE DATA-DS_ONE START-2 STOP-2 SELECT-1 CP ;
018       MODEL YIELD - ROPEN_10  RUNO_10  RLB_10  RCUR_10
019                     RX3_10    RP_10    RMAT_10 RZ21_10  RCUM_10 ;
```

Program: YRXX

Purpose:

This program takes the data from the program XDAT along with the selected variables from the program XRSQ and performs several Markov analysis forecast simulations.

Procedure:

The data was broken into two data sets one (ds_one) with the data to provide the forecasts and the other (ds_two) with the "actual" yield values, for later cross-validation analysis, (Efron 1982). Both of these data sets had new variables added to them based upon the selected variables, a set of user defined categories, and the breakpoints of the variable categories for the forecasts data set. These new categorical variables were then formed into Markov transition matrices and used to forecast the "actual" yields.

Program Outline:

Lines 001–005 JCL card for SAS batch processing.

Lines 006–009 titles to document SAS output.

Lines 010–015 read disk data and segment it into a forecast data set (ds_one) and an actual data set (ds_two).

Lines 016–079 macro _CAT1 : used to create a one level categorical break for a single variable.

Lines 080–187 macro _CAT2 : used to create a two level nested categorical break.

Lines 188–191 these lines are where the user inputs the selected variables and the number of splits per variable that are to be used.

Lines 192–203 the index variables are incremented by one for future readability.

Lines 204–207 this PROC FREQ provides the cell counts for the transition matrices.

Lines 208–216 univariate statistics are calculated for the final yield categories.

Lines 217–220 the statistics are printed and merged back into the original data set.

Lines 221–234 macro _MTX : creates the transition matrices.

Lines 235–297 macro _CMX : calculates the cumulative distribution and other statistics for the forecasted data.

Lines 298–315 macro _ESX : determines the estimate to be used (mean value or median), the actual, and the residual values. Univariate statistics and plots are then produced for the forecast simulation.

Lines 316–323 these lines are the code and macro combinations necessary to "run" the markov simulation for months 8, 9, 10 ⟶ forecast.

Lines 324–330 simulation for months 9, 10 ⟶ forecast.

Lines 331–336 simulation for month 10 ⟶ forecast.

Comments:

This SAS program was developed to satisfy a specific research need. Thus, it is not optimally coded and should not be used for production work. There are parts of the code that may be highly dependent on the version of SAS used. These include Lines 221–234 where the order of the information read from the PROC FREQ may vary and the use of PROC MATRIX in Lines 016–079, Lines 080–187, or in Lines 316–336 which has been replaced by IML. Also in some cases the number of categories asked for by the user in Lines 188–191 may not be supported by the data. When this occurs the PROC MATRIX procedures will fail to form the transition matrices. This can easily be corrected by choosing a smaller number of classifications and resubmitting the program.

SAS Code follows:

```
001  //YR81   JOB (C774,2C,1,25,JM),'BOUDREAUX',
002  //       MSGCLASS=Z,MSGLEVEL=(0,0)
003  // EXEC SAS,OPTIONS=MACRO,REGION=2048K
004  //DSK    DD DSN=USR.E413.JM.STATE48,DISP=OLD
005  //SYSIN DD *
```

```
TITLE1 ' MARKOV ANALYSIS              ' ;
TITLE2 ' XCAT & XMAT PROCEDURES       ' ;
TITLE3 ' AUG 88, MATIS & BOUDREAUX    ' ;
TITLE4 ' STATE-48(ALL) MODEL YR-81    ' ;

DATA DS_ONE ;
     SET DSK.STATE48 ;
     IF (YEAR NE 1) ;
DATA DS_TWO ;
     SET DSK.STATE48 ;
     IF (YEAR EQ 1) ;

%MACRO _CAT1(V1,N1,BK) ;
     /* CLASSIFY DATA IN A SECOND DATA SET BY
     /* ONE VARIABLE  IN DATA SET ONE .
     /*
     /* V1     : FIRST VARIABLE
     /* N1     : NO. OF CATEGORIES TO BREAK VAR1
     /* BK     : RESULTING CATEGORICAL VARIABLE
     */

     DATA DS_ONE ;
          SET DS_ONE ;
          IF (&V1 - .) THEN
             DELETE ;

     DATA DS_TWO ;
          SET DS_TWO ;
          IF (&V1 - .) THEN
             DELETE ;

     PROC RANK DATA-DS_ONE OUT-RS1 GROUPS-&N1 ;
          VAR &V1 ;
          RANKS BREAK1 ;
     PROC SORT DATA-RS1 ;
          BY BREAK1 ;
     PROC UNIVARIATE NOPRINT DATA-RS1 ;
          VAR &V1 ;
          OUTPUT OUT-US1 MAX-MAX1 ;
          BY BREAK1 ;

     PROC MATRIX ;
          FETCH MAX1 DATA-US1 (KEEP-BREAK1 MAX1) ;

          SPLIT1 - J.(1,&N1,0) ;
          SPLIT1(1,1) - 0 ;
          DO I - 2 TO &N1 ;     ;
             SPLIT1(1,I) - MAX1(I-1,2) ;
          END ;
```

```
          PRINT SPLIT1 ;

          FETCH NEW DATA=DS_TWO (KEEP=&V1) ;
          NR = NROW (NEW) ;
          YY = J.(NR,2,0) ;
          DO I = 1 TO NR ;
             DO J = 1 TO &N1 ;
                IF (J LT &N1) THEN DO ;
                   IF (NEW(I,1) GT SPLIT1(1,J)) AND
                      (NEW(I,1) LE SPLIT1(1,J+1)) THEN
                      YY(I,1) = J ;
                   END ;
                ELSE IF (J EQ &N1) THEN DO ;
                   IF (NEW(I,1) GT SPLIT1(1,J)) THEN
                      YY(I,1) = J ;
                   END ;
                ELSE
                   YY(I,1) = . ;
             END ;
          END ;
          OUTPUT YY OUT=YY ;

     DATA YY ;
          SET YY  ;
          &BK = COL1 - 1 ;
          DROP COL1 ;

     DATA DS_ONE ;
          SET RS1 ;
          &BK = BREAK1 ;
          DROP BREAK1 ;

     DATA DS_TWO ;
          MERGE DS_TWO YY ;
          DROP ROW ;
%MEND _CAT1 ;

%MACRO _CAT2(V1,N1,V2,N2,BK) ;
     /* CLASSIFY DATA IN A SECOND DATA SET BY
     /* TWO VARIABLES (NESTED) IN DATA SET ONE .
     /*
     /* V1     : FIRST VARIABLE
     /* N1     : NO. OF CATEGORIES TO BREAK VAR1
     /* V2     : SECOND VARIABLE
     /* N2     : NO. OF CATEGORIES TO BREAK VAR2
     /* BK     : RESULTING CATEGORICAL VARIABLE
     */
```

```
DATA DS_ONE ;
     SET DS_ONE ;
     IF (&V1 - . OR &V2 - .) THEN
        DELETE ;

DATA DS_TWO ;
     SET DS_TWO ;
     IF (&V1 - . OR &V2 - .) THEN
        DELETE ;

PROC RANK DATA-DS_ONE OUT-RS1 GROUPS-&N1 ;
     VAR &V1 ;
     RANKS BREAK1 ;
PROC SORT DATA-RS1 ;
     BY BREAK1 ;
PROC RANK DATA-RS1 OUT-RS2 GROUPS-&N2 ;
     VAR &V2 ;
     RANKS BREAK2 ;
     BY BREAK1 ;
PROC SORT DATA-RS2 ;
     BY BREAK1 BREAK2 ;
PROC UNIVARIATE NOPRINT DATA-RS2 ;
     VAR &V1 ;
     OUTPUT OUT-US1 MAX-MAX1 ;
     BY BREAK1 ;
PROC UNIVARIATE NOPRINT DATA-RS2 ;
     VAR &V2 ;
     OUTPUT OUT-US2 MAX-MAX2 ;
     BY BREAK1 BREAK2 ;

PROC MATRIX ;
     FETCH MAX1 DATA-US1 (KEEP-BREAK1        MAX1) ;
     FETCH MAX2 DATA-US2 (KEEP-BREAK1 BREAK2 MAX2) ;
     NR - NROW (MAX2) ;

     SPLIT1 - J.(1,&N1,0) ;
     SPLIT1(1,1) - 0 ;
     DO I - 2 TO &N1 ;      ;
        SPLIT1(1,I) - MAX1(I-1,2) ;
     END ;
     PRINT SPLIT1 ;

     SPLIT2 - J.(&N1,&N2,0) ;
     DO I - 1 TO NR ;
        R - MAX2(I,1) + 1 ;
        C - MAX2(I,2) + 1 ;
        SPLIT2 (R,C) - MAX2(I,3) ;
     END ;
```

```
          DO I = 1 TO &N1 ;
             DO J = &N2 TO 2 BY -1 ;
                SPLIT2 (I,J) = SPLIT2 (I,J-1) ;
             END ;
             SPLIT2 (I,1) = 0.0 ;
          END ;
          PRINT SPLIT2 ;

          FETCH NEW DATA=DS_TWO (KEEP=&V1 &V2) ;
          NR = NROW (NEW) ;
          YY = J.(NR,2,0) ;
          DO I = 1 TO NR ;
             DO J = 1 TO &N1 ;
                IF (J LT &N1) THEN DO ;
                   IF (NEW(I,1) GT SPLIT1(1,J)) AND
                      (NEW(I,1) LE SPLIT1(1,J+1)) THEN
                      YY(I,1) = J ;
                   END ;
                ELSE IF (J EQ &N1) THEN DO ;
                   IF (NEW(I,1) GT SPLIT1(1,J)) THEN
                      YY(I,1) = J ;
                   END ;
                ELSE
                   YY(I,1) = . ;
             END ;
             W = J.(1,1,0) ;
             W = YY(I,1) ;
             DO J = 1 TO &N2 ;
                IF (J LT &N2) THEN DO ;
                   IF (NEW(I,2) GT SPLIT2(W,J)) AND
                      (NEW(I,2) LE SPLIT2(W,J+1)) THEN
                      YY(I,2) = J ;
                   END ;
                ELSE IF (J EQ &N2) THEN DO ;
                   IF (NEW(I,2) GT SPLIT2(W,J)) THEN
                      YY(I,2) = J ;
                   END ;
                ELSE
                   YY(I,2) = . ;
             END ;
          END ;
          OUTPUT YY OUT=YY ;

     DATA YY ;
          SET YY  ;
          COL1 = COL1 - 1 ;
          COL2 = COL2 - 1 ;
          &BK = &N2*COL1 + COL2 ;
          DROP COL1 COL2 ;
```

```
          DATA DS_ONE ;
               SET RS2 ;
               &BK - &N2*BREAK1 + BREAK2 ;
               DROP BREAK1 BREAK2 ;

          DATA DS_TWO ;
               MERGE DS_TWO YY ;
               DROP ROW ;
     %MEND _CAT2 ;

     %_CAT2(RX3_8    ,4 ,RLB_8    ,2 ,INX1) ;
     %_CAT2(RLB_9    ,4 ,RX3_9    ,2 ,INX2) ;
     %_CAT2(RUNO_10 ,8 ,RCUM_10 ,2 ,INX3) ;
     %_CAT1(YIELD,50,SY1 ) ;

     DATA ONE ;
          SET DS_ONE ;
          INX1 - INX1 + 1 ;
          INX2 - INX2 + 1 ;
          INX3 - INX3 + 1 ;
          SY1  - SY1  + 1 ;
     DATA TWO ;
          SET DS_TWO ;
          INX1 - INX1 + 1 ;
          INX2 - INX2 + 1 ;
          INX3 - INX3 + 1 ;
          SY1  - SY1  + 1 ;

     PROC FREQ DATA-ONE ;
          TABLES INX1*INX2 / NOPRINT OUT-T12 ;
          TABLES INX2*INX3 / NOPRINT OUT-T23 ;
          TABLES INX3*SY1  / NOPRINT OUT-T3Y ;

     PROC SORT DATA-ONE ; BY SY1 ;
     PROC UNIVARIATE NOPRINT DATA-ONE ; BY SY1 ;
          VAR YIELD ;
          OUTPUT OUT-UNI
                 MEAN   - MNY1
                 MEDIAN - MDY1
                 Q1     - Q1
                 Q3     - Q3
                 VAR    - VAR ;

     PROC PRINT DATA-UNI ;
     PROC SORT DATA-UNI ; BY SY1 ;
     DATA ONE ;
          MERGE ONE UNI ; BY SY1 ;
```

```
%MACRO _MTX (FRQ,AMX,A_ROW,A_COL) ;
      /* READ PROC FREQ DATA INTO MATRICIES
      /*
      /* FRQ      : PROC FREQ OUTPUT DATASETS
      /* AMX      : RESULTING MATRIX
      */
      FETCH &FRQ DATA=&FRQ ;
      &AMX = J.(&A_ROW,&A_COL,0) ;
      DO I = 1 TO NROW(&FRQ) ;
         &AMX(&FRQ(I,1),&FRQ(I,2)) = &FRQ(I,3) ;
      END ;
      /* SCALE THE ROWS OF MATRIX : SUM = 1.0 */
      &AMX = (1 #/ (DIAG(&AMX(,+)))) * &AMX ;
%MEND _MTX ;

%MACRO _CMX (AMX,CMX,STA,INX) ;
      /* GENERATE STATISTICS
      /*
      /* AMX     : ORIGINAL MATRIX
      /* CMX     : CUM DIST OF ORIGINAL MATRIX
      /* STA     : Q1,Q3,MEDIAN,MEAN,VAR OF AMX
      /* INX     : INDEX NUMBER, MUST CORRESPOND TO AMX
      */
      FETCH UNI DATA=UNI (KEEP=MNY1) ;
      NC = NCOL(&AMX) ;
      NR = NROW(&AMX) ;
      &CMX = J.(NR,NC+1,0) ;
      &CMX = &AMX || J.(NR,1,0) ;
      &STA = J.(NR,6,0) ;
      DO I = 1 TO NR ;
         CUM  = 0 ;
         VAR  = 0 ;
         MEAN = 0 ;
         DO J = 1 TO NC ;
            MEAN = MEAN + (UNI(J,1)*&AMX(I,J)) ;
            VAR  = VAR  + ((UNI(J,1)**2)*&AMX(I,J)) ;
            CUM  = CUM  + &AMX(I,J) ;
            &CMX(I,J) = CUM ;
            IF (CUM LE 0.25) THEN &STA(I,1) = J ;
            IF (CUM LE 0.50) THEN &STA(I,2) = J ;
            IF (CUM LE 0.75) THEN &STA(I,3) = J ;
         END ;
         &CMX(I,NC+1) = I ;
         IF (&STA(I,1) = NC) THEN
            GOTO LX ;
         /* *** Q1      */
         &STA(I,1) = UNI(&STA(I,1),1)
            + (UNI(&STA(I,1)+1,1)
            - UNI(&STA(I,1),1))
```

```
                * ((0.25 - &CMX(I,&STA(I,1)))
                #/ (&CMX(I,&STA(I,1)+1)
                - &CMX(I,&STA(I,1)))) ;
             /* *** MEDIAN */
             &STA(I,2) = UNI(&STA(I,2),1)
                + (UNI(&STA(I,2)+1,1)
                - UNI(&STA(I,2),1))
                * ((0.50 - &CMX(I,&STA(I,2)))
                #/ (&CMX(I,&STA(I,2)+1)
                - &CMX(I,&STA(I,2)))) ;
             /* *** Q3     */
             &STA(I,3) = UNI(&STA(I,3),1)
                + (UNI(&STA(I,3)+1,1)
                - UNI(&STA(I,3),1))
                * ((0.75 - &CMX(I,&STA(I,3)))
                #/ (&CMX(I,&STA(I,3)+1)
                - &CMX(I,&STA(I,3)))) ;
             /* *** MEAN   */
             &STA(I,4) = MEAN ;
             /* *** VAR    */
             &STA(I,5) = VAR - MEAN**2 ;
             /* *** INDEX  */
             &STA(I,6) = I     ;
LX : END ;
     OUTPUT &STA OUT=&STA
            (RENAME = (COL1=Q1 COL2=MEDIAN COL3=Q3
                       COL4=MEAN COL5=VAR COL6=&INX)) ;
     FREE CUM MEAN VAR ;
%MEND _CMX ;

%MACRO _ESX (EST,STA,INX) ;
     /* MERGE STATISTICS WITH NEW DATA
     /*
     /* EST    : WHICH ESTIMATE (MEAN, MEDIAN)
     /* STA    : DATASET OF MATRIX STATISTICS
     /* INX    : WHICH INDEX VARIABLE TO USE
     */
     PROC PRINT DATA=&STA ;
     PROC SORT DATA=&STA ; BY &INX ;
     PROC SORT DATA=TWO  ; BY &INX ;
     DATA TWO ;
          MERGE TWO &STA ; BY &INX ;
          EST = &EST      ; /* ESTIMATES YIELD */
          ACT = YIELD     ; /* ACTUAL YIELD    */
          RES = EST - ACT ; /* RESIDUAL        */
     PROC UNIVARIATE PLOT ;
          VAR EST ACT RES ;
%MEND _ESX ;
```

```
PROC MATRIX ;
     %_MTX (T12,A1,8,8)
     %_MTX (T23,A2,8,16)
     %_MTX (T3Y,A3,16,50)
     EST = A1*A2*A3 ;
     %_CMX (EST,CDF,STATS,INX1)
     %_ESX (MEAN,STATS,INX1)
     TITLE5 ' MONTHS 8,9,10               ' ;

PROC MATRIX ;
     %_MTX (T23,A2,8,16)
     %_MTX (T3Y,A3,16,50)
     EST =    A2*A3 ;
     %_CMX (EST,CDF,STATS,INX2)
     %_ESX (MEAN,STATS,INX2)
     TITLE5 ' MONTHS   9,10               ' ;

PROC MATRIX ;
     %_MTX (T3Y,A3,16,50)
     EST =       A3 ;
     %_CMX (EST,CDF,STATS,INX3)
     %_ESX (MEAN,STATS,INX3)
     TITLE5 ' MONTHS     10               ' ;
```

References

Efron, B. (1982). The Jackknife, the Bootstrap and Other Resampling Plans. Society for Industrial and Applied Mathematics, Philadelphia, PA.

Matis, J. H., Saito, T., Grant, W. E. Iwig, W. C., and Richie, J. T. (1985). A Markov chain approach to crop yield forecasting. *Agricultural Systems* 18:171–187.

Matis, J. H., Birkett, T., and Boudreaux, D. (1989). An application of the Markov Chain approach to forecasting cotton yield from surveys. *Agricultural Systems* 29:357–370.

Matis, J.H., Perry, C.R., Boudreaux, D.E., and D.J. Aune (1989). ***Markov Chain Forecasts of Cotton Objective Yield***, National Agricultural Statistics Service, U.S. Department of Agriculture, Washington, D.C. 20250, Research Report No. SRB 89–11.

*U.S. GOVERNMENT PRINTING OFFICE:1989-