HortScience 29(2): 123–125. 1994.

A Simple Technique to Randomize and Label Large Experimental Plantings

W.R. Okie, W.R. Joyner, and T.G. Beckman
Southeastern Fruit and Tree Nut Research Laboratory, U.S. Department of Agriculture, 111 Dunbar Road, Byron, GA 31008

Additional index words. experimental design

Abstract. Large **field plantings are often difficult to label and to plant randomly. A DOS computer program was developed in SAS and BASIC to randomize lists of experimental factors and print sorted paper labels to apply to trees or plants. Tagged trees can be resorted readily by block or row to speed planting. The computer lists are useful for plot verification and subsequent data collection, especially if data are collected and inputted directly to a computer. Copies of the programs are available from W.R. Joyner if a formatted diskette and self-addressed mailer are supplied.**

Breeding and cultivar testing often require field or greenhouse experiments with many varieties or genotypes as treatments, arranged in blocks either singly or in multiple-plant plots. The logistics of labeling and planting can be difficult. To minimize the difficulty, we have developed a computer program in SAS (SAS Institute, 1985) and BASIC programming languages (Waite et al., 1990) that randomizes the appropriate factors (i.e., row and plot numbers, treatment names, and color codes) and produces a sorted file that can be printed on paper labels (Fig. 1). As an example, we used each program to describe a planting consisting of three randomized complete blocks of four varieties each, to be planted in two rows each with six single-tree plots. These experimental factors should be adjusted as needed for a particular design. Slight program modifications can be made for completely randomized, split plot, or multiple-plant plot configurations.

In SAS (Fig. 2), a treatment list is inputted (Fig. 2A), and merged with a list of random numbers (IDX). The RANUNI function returns a random number between 0 and 1. By sorting on appropriate identification variables and the random variable using PROC SORT (Fig. 2B), a randomized layout for the testis created. Plot and row values are generated in order (Fig. 2C). Then these values are merged with the randomized layout list (Fig. 2D) that is saved to a map file (plot.map) for use in data collection and displayed on the screen for verification. The map is resorted by variety name (Fig. 2E), and then outputted (Fig. 2F) as a DOS print file (labels.prt) in a format suitable for printing labels. This program will work even if the block consists of rows and parts of rows. For multiple-plant plots, either use one tag per plot bundle or increment the DO TREES loop (Fig. 2G) to make copies of each label. For a completely randomized design, remove the term "BLOCK' in the first PROC SORT (Fig. 2B).

As an alternative, the list and print file are generated using BASIC (Fig. 3). Creating the identification variable list, using FOR-NEXT loops, and then selecting at random from an array or a random file obviates the need for a complicated sort routine. The key to quick randomization in BASIC is simply to replace the selected value with the last value in the selection range and to truncate the selection range. This technique precludes selection of empty (already chosen) cells.

After initializing variables (Fig. 3A), array MAP is loaded with block numbers and variety names and then loaded with ROW and PLOT values (Fig. 3B). Variety names are randomized within the MAP array (Fig. 3C). A DOS map file (plot.map) is printed for use in data collection (Fig. 3D). The MAP array, grouped by variety, is outputtted (Fig. 3E) as a DOS file (label.prt) suitable for printing 10 × 3.5-cm labels.

If the "labels.prt" output file from either program contains printer control codes to change print size or spacing, use the DOS COPY command to send it to the printer: "copy labels.prt 1pt1 :". Pin-fed vinyl or self-adhesive paper labels can be printed out in

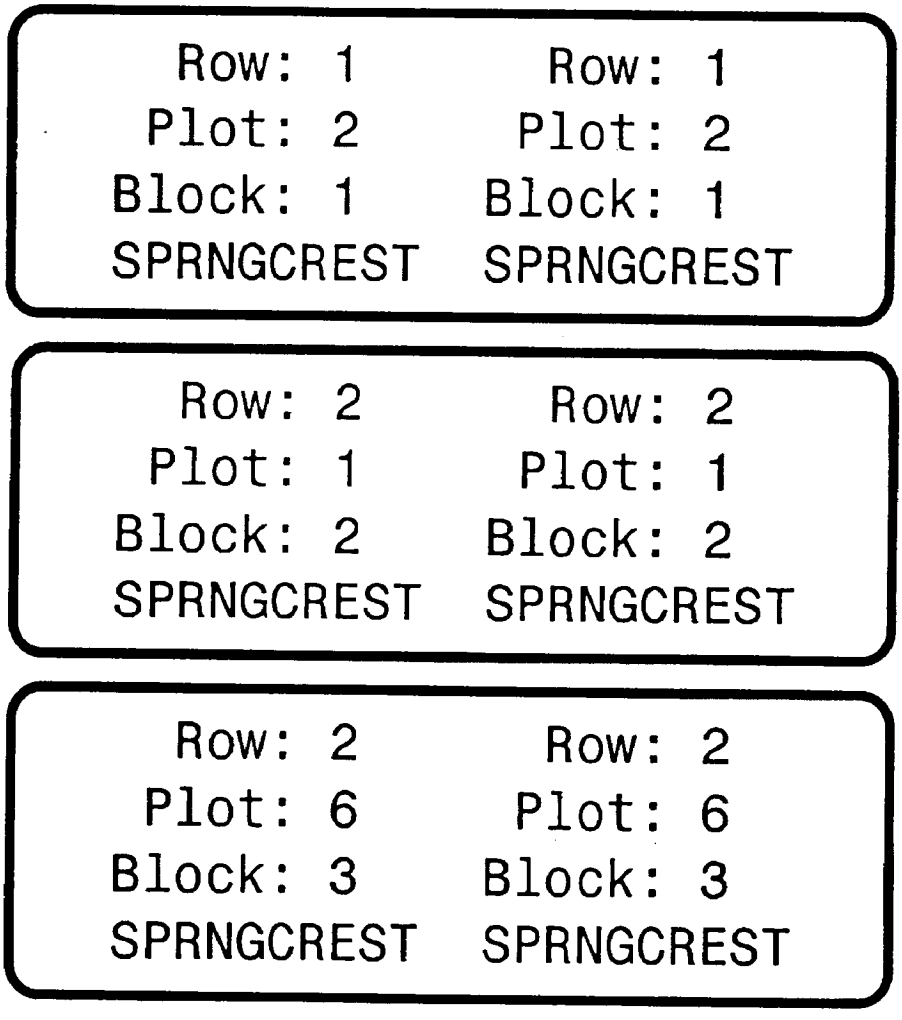

Fig. 1. Typical paper label generated by program in Fig. 2.

Received for publication 22 Mar. 1993. Accepted for publication 25 Aug. 1993. The cost of publishing this paper was defrayed in part by the payment of page charges. Under postal regulations, this paper therefore must be hereby marked *advertisement* solely to indicate this fact.

sequence by variety name and tree number (Fig. 1). Labeled trees subsequently can be sorted into groups based on a block or row number on the label. Paper labels remain legible for up to 6 months outside, unless placed near the soil.

This program may be applied to any experiment where labeled items are needed and where sorting experimental units is burdensome. Additional information or color codes can be put on tags depending on particular needs. Copies of these programs and versions

```
/*PROGRAM TO GENERATE PLOT MAP AND LABELS FOR RANDOMIZED
  COMPLETE BLOCKS:  ROWS=2 PLOTS=6 BLOCKS=3 VARIETIES=4* /

A: *Input list of VARIETIES;
    data vlist; input VARIETY $10, ;
      do BLOCK = 1 to 3; *specify no. blocks per variety;
        IDX = ranuni (0); *generate random numbers;
      output; end; cards;
SPRNGCREST
REDRAVEN
REDGLOBE
OHENRY
    run;

B: *Sort each BLOCK by random number;
   *For COMPLETELY RANDOMIZED design remove BLOCK from next line;
    proc sort; by BLOCK IDX; run;

C: *Make list for ROWS and PLOTS;
    data rlist;
      do ROW = 1 to 2; *specify first and last ROW;
        do PLOT = 1 to 6; *specify PLOTS per ROW;
      output; end; end; run;

D: *Merge VARIETIES list with ROW-PLOT list;
    data vrlist;
      merge vlist rlist;
      file 'plot.map' ;  *designate MAP file;
      if VARIETY ne ' ' ; *delete any missing VARIETIES;
      put ROW PLOT BLOCK VARIETY; *write to MAP file;
      drop IDX; run;

   *Print MAP by ROW and PLOT for verification;
    proc print; run;

E: *Sort by VARIETY to print labels;
    proc sort; by VARIETY ROW PLOT; run;

F: *Print Labels file;
    data _null_; set vrlist;
    file ' labels. prt' ; *designate output file;
   *Control codes for 10x3. 5 cm labels on Epson printer;
    FF=byte ( 12 ); *skip to top of page= label;
    FG=byte(27) ||byte(67) ||byte(9); *page=9 lines;
    DARK=byte(27) ||byte(33) ||byte(41) ; *bold & double wide;
    BIG=byte(27) ||byte(119) ||byte(l) ; *double high;

   if _n_=1 then put DARK BIG PG
   *Print both halves of labels;
G: *for 1 label/plot; do TREES=1 to 1;
    put @3 'Row: ' ROW      @14 'Row: ' ROW /;
    put @2 ' Plot: ' PLOT    @13 'Plot: ' PLOT / ;
   *Delete next line for COMPLETELY RANDOMIZED design;
    put @1 'Block:  ' BLOCK @12 'Block: ' BLOCS /;
    put @1 VARIETY         @12 VARIETY FF; end;
    run;
```

Fig. 2. Sample SAS program to randomize and print labels for a randomized complete-block design with many varieties. Control characters in final DATA step are typical for Epson dot-matrix printers.

```
   Program to create map file and label file for RANDOM COMPLETE
   BLOCK design of 3 BLOCKS, 4 VARIETIES, 2 ROWS, 6 PLOTS

A: 'Define types and initialize variables
     TYPE LABEL
       Row AS INTEGER
       Plot AS INTEGER
       Block AS INTEGER
       variety AS STRING * 10
     END TYPE

       CONST NoOfBlocks =3, NoOfVarieties =4
       CONST NoOfRows =2, NoOfPlots =6, NoOfTree. =1
       DIM Map(NoOfRows * NoOfPlots) AS LABEL
       FFS =CHRS(12), CRS = CNRS(13)                    'Form Feed and Carriage Return
       PGLENS - CHRS(27) + CHRS(67) + CNRS(9)           'Printer settings for page length=9
       EMPHS - CSRS(27) + CHRS(33) + CHRS(41)           'Printer bold
       BIGS - CHRS(27) + CHRS(119) + CHRS(l)            'Printer size

     RANDOMIZE TIMER                                    'Initialize RND Generator
     OPEN "plot.map" FOR OUTPUT AS 1                    'OpEN MAP file
     OPEN "labels.prt" FOR OUTPUT AS 2                  'Open LABEL file

B: 'Load array with BLOCK, VARIETIES, ROW, PLOT
     FOR 1% =1 TO NoOfBlocks: FOR j% - 1 TO NoOfVarieties
         Map((i %- 1) *NoOfVarieties + j%). Block =i%
         READ Map((i% - 1) * NoOfVarieties + j %), Variety
     NEXT j%: RESTORE: NEXT i%
     FOR 1% =1 TO NoOfRows: FOR J% - 1 TO NoOfPlots
         Map((i% - 1) 1 NoOfPlots + j%). Row - i%
         Map((i %- 1) 1NoOfPlots + j%). Plot - j%
     NEXT J%: NEXT i%

C: 'Randomize VARIETY within BLOCKS by swapping names and trucating selection range
     FOR i% - 1 TO NoOfBlocks
       FOR J% = NoOfVarieties TO 1 STEP -1
         idx% = ((i% - 1) * NoOfVarieties) + (INT(RNO * j%, + 1))
         IF idx% <> (i% - 1) * NoOfvarieties + j% THEN
           tempS - Map((i% - 1) * NoOfVar ieties + j%). variety
           Map((i% - 1) l NoOfVarieties + J%), Variety = Map(idx%), Variety
           Map(idx%), Variety = tempS
         END IF
       NEXT j%
     NEXT i%

D: 'Print to MAP file
     FOR i% - 1 TO NoOfBlocks * NoOfVarieties
       PRINT #1, USING ### ### ### &"; Map(i%). Row; Map(i%). Plot; Map(i%). Block; Map(i%). Variety
     NEXT i%: CLOSE #1


E: 'Print to LABEL by VARIETY
      PRINT #2, FGLENS; EMPHS; BIGS                     -Initialize printer
      RESTORE                                           'Reset Variety names
      FOR 1% -1 TO NoOfVarieties: READ tempS            'Get Variety name
        FOR j% - 1 TO NoOfBlocks *NoOfVarieties
          IF LEFTS (Map(j%). Variety, LEN(tempS)) = temp$ THEN  'Match to name in ARRAY
          FOR k% =1 TO NoOfTrees                            'Print LABEL
            PRINT #2, USING "ROW:###     Row:###&"; Map(j%). Row; Map(j%). Row; CRS
            PRINT #2, USING " Plot:###    Plot: ###&"; Map(j%). Plot; Map(j%). Plot; CRS
            PRINT #2, USING " Block:###   Block: ###&"; Map(j%). Block; Map(j%). Block; CRS
            PRINT #2, USING "\         \ \         \"'; Map(j%) Variety; Map(j%), Variety
            PRINT #2, FFS
          NEXT k%
          END IF
        NEXT j%
      NEXT i%
      cLOSE #2

   'List VARIETIES here
     DATA SPRNGCREST, REDHAVEN, REDGLOBE, OHENRY
```

Fig. 3. Sample BASIC program to randomize and print labels for a randomized complete-block design with many varieties. This version written in QuickBASIC (Microsoft, Bellevue, Wash.).

for more complex designs are available from W.R. Joyner, if the requester supplies a formatted diskette and self-addressed mailer.

Literature Cited

SAS Institute. 1985. SAS language guide for personal computers. SAS Institute, Inc., Cary, N.C.

Waite, M.,R. Arnson, C. Gemmell, and H. Henderson. 1990. The Waite group's MicrosoftQuickBASIC bible. Microsoft Press, Redmond, Wash.