Introduction

The choice among methods for generating equivalent statistics often is based on the form of the printed results. Steps in transforming computer output to informative displays can be minimized by the careful selection and combination of SAS procedures and programming statements.

SAS code, printed results and time estimates relative to alternatives are presented for two examples. The first example shows an automation of transforming PROC PLAN random numbers to a tear sheet schedule for assigning patients to treatments in a clinical trial. Here with a modicum of SAS statements, PROC PRINTTO is used to make the results of PROC PLAN accessible to DATA programming statements in the same job. PROC MATRIX programming statements are used to consolidate the long two-column list output of the DATA step and provide corresponding column spacing and replicate variable names. PROC PRINT prints the results.

The second example provides a hierarchical display of distributional statistics relative to classification variables and analysis variables in a similar format to simple PROC TABULATE displays. The statistics shown in one line of the display correspond to a summary of one page of selected PROC UNIVARIATE output including percentiles, symmetry measures and parametric/nonparametric location tests. These statistics are generated in DATA programming statements. The alternative for comparison is the use of PROC PRINTTO of PROC UNIVARIATE output.

The purpose of this paper is to make available two routines that have been useful in health science applications and to identify SAS options for optimizing the processing of obtaining statistical information.

A Randomization Schedule

A common task in medical statistics involves providing researchers with randomization schedules. Typically, these schedules might be used to assign patients to treatment in a clinical trial assessing treatment differences. One approach is to generate random permutations of a particular blocksize for a number of randomly ordered blocks. Here, the blocksize represents the desired number of random assignments with known value result in planned proportional allocation to each treatment. Blocksize is also used to control a run of random assignments to one treatment. The specific values of the numerical sequence [1 to blocksize] represent treatment assignment. The order of the permuted sequence determines the order of treatment assignment. Finally, a double shuffle can be achieved by randomly ordering the use of each block for assigning treatment order. PROC PLAN provides random permutations of random blocks for this purpose. However, these results need to be ordered and transformed into treatment assignments for practical use in the clinic.

A convenient format for a randomization schedule involves replicated paired columns on each page. Here, the paired columns list patient number with treatment assignment. The researcher would sequentially read down the paired columns as assignments were made and then start at the top of the page when the paired columns are complete. For clarity the replicated paired columns should have the same headings, e.g., "PATIENT" "DRUG" and should have additional spacing between listings.

SAS code in Figure 1 for a schedule in Figure 2 shows the use of PROC MATRIX programming statements to construct the formatting requirements of a single page schedule. Here PROC MATRIX is used as an alternative to detailed pointer instructions. PROC PRINT prints the page layout as a table which is readily printed by PROC PRINT. Additional spacing between the paired columns is obtained by an imbedded blank column with a null variable name. More specifically a column of missing values is printed as blank when using the OPTIONS MISSING = " " statement. Replicate variable names are assigned using the COLNAME = option in the OUTPUT statement. Replicate variable names cannot be assigned in the RENAME statement or referred to in the VAR statement without error. For this reason, PROC MATRIX needs to provide a dataset that can be printed without reference to the VAR list. In PROC PRINT, the observation number can be suppressed by using the ID statement. Here ID row variable; is used where the row numbers have been blanked out using the ROWNAME = option in the matrix OUTPUT statement. The row variable reference in the ID statement does not involve the reference of a replicated variable name. Finally, the replicated column structure is created in PROC MATRIX by dividing the augmented instream observations (LIST) having variables PATIENT, PERM, and BLANK into three equal lengths and concatenating them side by side.

The linkage between PROC PLAN and PROC MATRIX involves PROC PRINTTO to write the PROC PLAN output onto an external file and a DATA step to read PROC PLAN output creating one observation per treatment assignment (PERM) which can be sorted according to the random permutation of blocks (A) while maintaining the randomized order within blocks (2). The variable PERM (treatment assignment) has values of the permuted blocksize. The actual decoding of numbers to treatment names takes place at the PROC PRINT stage with a FORMAT statement. Apparently, the replicated variable structure at this stage does not generate errors in the FORMAT statement.

Discussion

With special use of PROC MATRIX statements and options, the randomized schedule was generated with relatively few SAS statements. Computer time for an IBM 3081 with an MVS operating system increased from 2.9 seconds (1.4 cpu) for PROC PLAN alone to 9.9 seconds (4.7 cpu) for the
completed randomized schedule. The computing routine from PROC PLAN to PROC PRINT requires only a few changes to adapt it to changes in blocksize and number of blocks. In fact, these changes can be made automatically by having blocksize and number of blocks as parameters to a macro language version of the routine.

Figure 3 demonstrates an extension to a more detailed clinical tear sheet which provides spaces for tallies of numbers of patients assigned/number of patients participating in a multi-treatment/pooled control design. Here, twenty pages of formatted group assignment was specified by a DO loop operation in PROC MATRIX.

**Distributional Statistics Display**

The distributional statistics routine is a SAS MACRO system in Figure 4 that augments statistics available in PROC MEANS with rank type statistics generated in DATA programming steps. The decision to create this routine was based on the need to display this combination of statistics in one line per classification or analysis variable(s). The choice of combining PROC MEANS output with DATA programming statements was based on (1) the accessibility of PROC MEANS output through the use of the OUTPUT statement and (2) the ability to specify the algorithm of order statistics and generate other statistics in alternative forms that are not available in other SAS PROCedures. A similar set of statistics can be obtained through PROC UNIVARIATE using the PROC PRINTTO mechanism to collect each variable. However, the input-output (I/O) task and the DATA step for collecting variables is cumbersome and does not include the user defined alternative statistics. The difficulty arises because the OUTPUT statement in PROC UNIVARIATE does not provide all of the desired statistics.

**Example Statistical Display**

The printed results of the distributional statistics routine is shown in Figure 5 for repeated measurements over time in a clinical trial with randomized treatment assignment. The page displays statistical information concerning the percent increase in hormone level (HL) over a baseline level at month zero relative to months 1, 3, 6, 9, 12 and endpoint for each treatment and pooled across treatment.

The statistics include:
- **MIN** the minimum value observed
- **M25** the value of 25th percentile, see definition A
- **MED** the median value or 50th percentile, see definition A
- **M75** the value of the 75th percentile, see definition A
- **MAX** the maximum value observed
- **RANGE** the difference between MIN and MAX
- **QRATIO** the quartile ratio (M50-M25)/(M75-M25)
- **NMIN** the lowest sum of signed ranks
- **STDEV** the standard deviation of N observations
- **PROBST** PROC MEANS p-value for the Student's t statistic testing the hypothesis that the population mean is 0.
- **PROBZ** a normal approximation p-value for the Wilcoxon Signed Rank statistic testing the hypothesis that values are centered about 0.
- **N** the sample size
- **MEAN** the mean of N observations
- **STDV** the standard deviation of N observations
- **QRATIO** the quartile ratio (M50-M25)/(M75-M25)
- **NMIN** the lowest sum of signed ranks

The example in Figure 5 shows that the present increase in HL is rarely symmetric with extreme values for QRATIO and large discrepancies between PSO and MEAN. With N < 10, PSO and PROBZ are probably more appropriate to use than MEAN and PROBST in describing the location of the distribution. For this data, HL shows a rise in the median value with time in each treatment group. This rise is significant in the pooled group at 6 months, 12 months and at endpoint.

**Percentiles Definition A**

Definition A is based on a principle by which features in definitions 1 to 5 in PROC UNIVARIATE are combined resulting in the empirical distribution function with weighted averaging aimed at $x(np + g)$.

Definition A: weighted average aimed at $x(np + g)$

\[
y = (1 - g) \cdot x(j) + g \cdot x(j + 1)
\]

where $np + g = j + g$ with $j$ being the integer part of $(np + g)$ and $g$ being the fractional part.
User Instructions for Distributional Statistics Routine

The following MACRO's define the hierarchical display:

MACRO ITEM lists the page variables.
MACRO TAGVAR lists page labeling variables.
MACRO OTHERVAR lists major page variables.
MACRO BASIC GR identifies the inner most page variable which distinguishes one row from the next.

These MACRO's can be left empty. A DATA step needs to be specified as follows:

DATA STUDY (KEEP = BYVAR VALUE TAGVAR);
SET yourset;
* let PAGE be a character string identifying your page variable;
* let VALUE take on the value of the page variable;
* OUTPUT only if VALUE is NOT MISSING;

PAGE = 'HL'; VALUE = HL; IF VALUE > . THEN OUTPUT;
*REPEAT FOR EACH PAGE VARIABLE;

The MACRO is simply referenced by a statement call following the DATA step as follows:

ORDER

Results can be printed using PROC PRINT:
PROC SORT DATA = SIMPLE; BY BYVAR;
PROC PRINT DATA = SIMPLE; BY ITEM: ID OTHERVAR;
VAR BASIC GR MIN P25 P75 MAX RANGE QRATIO NMIN SMIN PROZ N MEAN STDV PROBT;

References

Figure 1

TEAR SHEET RANDOMIZATION SCHEDULES

INGRID AMARA AND GABY KOCH
THE UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

OPTIONS LINESIZE=72 MISSING=' ' NODATE;
PROC PRINT UNIT=20 NEW;
PROC PLAN: FACTORS A=15 B=6;
PROC PRINT;
OPTIONS LINESIZE=80;
TITLE1 'RANDOMIZATION SCHEDULES';
TITLE2 'A MULTICENTER RANDOMIZED COMPARISON OF EFFECTIVENESS & SAFETY';
TITLE3 'OF THREE DRUGS: A, B & C';
TITLE4 'IN THE TREATMENT OF CLINICAL AILMENT TYPE A';
TITLE5 'INVESTIGATOR: JOHN DOCTOR';
TITLE6 'BLOCKSIZE: f(2A + 2B + 2C)';
DATA ALLOC;
FILE PRINT;
INFILE FT20F001;
* WRITE OUT PROC PLAN RESULTS;
DO K=1 TO 8; INPUT #; PUT _INFILE_; END;
INPUT # 9 0;
ARRAY B(J) B1-B6;
BLANK=;
KEEP A J BLANK;
DO I=1 TO 15;
   INPUT #5 A #1-B6;
   PUT _INFILE_;
   DO J=1 TO 6;
      PERM=E;
      OUTPUT;
   END:
END:
PROC SORT: BY I J;
FFOC MATRIX,
FETCH PRDSE1=ALLOC(KEEP=PERM BLANK);
PATIENT= (1:NROW(FAILS));
LIST=PATIENT || PAIR;
TABLE= LIST(1:30,) || LIST(31:60,) || LIST(61:90,1 2);
NAMES = 'PATIENT' || 'DRUG' || 'PATIENT' || 'DRUG';
BLANK = ' ';
R = J1:NROW(TABLE),BLANK;
OUTPUT TABLE OUT=PRDSET COLNAME=NAMES DOWNNAME=R;
PROC FORMAT:VALUE TCOD=1-2=A 3-4=E 5-6=C;
PROC PRINT DATA=PRDSET;
ID R;
FORMAT DRUG TCODE,';
### Figure 2
**Randomization Schedules**
A multicenter randomized comparison of effectiveness & safety of three drugs: A, B & C in the treatment of classical alveolar type A

**Investigator:** John Ectoe

**Blocksize:** 6 (2A + 2B + 2C)

<table>
<thead>
<tr>
<th>Row</th>
<th>Patient</th>
<th>Drug</th>
<th>Patient</th>
<th>Drug</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>31</td>
<td>A</td>
<td>61</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>32</td>
<td>B</td>
<td>62</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
<td>33</td>
<td>B</td>
<td>63</td>
</tr>
<tr>
<td>4</td>
<td>A</td>
<td>34</td>
<td>A</td>
<td>64</td>
</tr>
<tr>
<td>5</td>
<td>B</td>
<td>35</td>
<td>C</td>
<td>65</td>
</tr>
<tr>
<td>6</td>
<td>A</td>
<td>36</td>
<td>C</td>
<td>66</td>
</tr>
<tr>
<td>7</td>
<td>A</td>
<td>37</td>
<td>B</td>
<td>67</td>
</tr>
<tr>
<td>8</td>
<td>C</td>
<td>38</td>
<td>B</td>
<td>68</td>
</tr>
<tr>
<td>9</td>
<td>C</td>
<td>39</td>
<td>A</td>
<td>69</td>
</tr>
<tr>
<td>10</td>
<td>A</td>
<td>40</td>
<td>C</td>
<td>70</td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>41</td>
<td>B</td>
<td>71</td>
</tr>
<tr>
<td>12</td>
<td>B</td>
<td>42</td>
<td>A</td>
<td>72</td>
</tr>
<tr>
<td>13</td>
<td>B</td>
<td>43</td>
<td>D</td>
<td>73</td>
</tr>
<tr>
<td>14</td>
<td>A</td>
<td>44</td>
<td>A</td>
<td>74</td>
</tr>
<tr>
<td>15</td>
<td>C</td>
<td>45</td>
<td>C</td>
<td>75</td>
</tr>
<tr>
<td>16</td>
<td>A</td>
<td>46</td>
<td>C</td>
<td>76</td>
</tr>
<tr>
<td>17</td>
<td>C</td>
<td>47</td>
<td>B</td>
<td>77</td>
</tr>
<tr>
<td>18</td>
<td>A</td>
<td>48</td>
<td>A</td>
<td>78</td>
</tr>
<tr>
<td>19</td>
<td>B</td>
<td>49</td>
<td>A</td>
<td>79</td>
</tr>
<tr>
<td>20</td>
<td>B</td>
<td>50</td>
<td>B</td>
<td>80</td>
</tr>
<tr>
<td>21</td>
<td>C</td>
<td>51</td>
<td>A</td>
<td>81</td>
</tr>
<tr>
<td>22</td>
<td>A</td>
<td>52</td>
<td>C</td>
<td>82</td>
</tr>
<tr>
<td>23</td>
<td>A</td>
<td>53</td>
<td>B</td>
<td>83</td>
</tr>
<tr>
<td>24</td>
<td>C</td>
<td>54</td>
<td>A</td>
<td>84</td>
</tr>
<tr>
<td>25</td>
<td>C</td>
<td>55</td>
<td>A</td>
<td>85</td>
</tr>
<tr>
<td>26</td>
<td>A</td>
<td>56</td>
<td>C</td>
<td>86</td>
</tr>
<tr>
<td>27</td>
<td>B</td>
<td>57</td>
<td>B</td>
<td>87</td>
</tr>
<tr>
<td>28</td>
<td>A</td>
<td>58</td>
<td>C</td>
<td>88</td>
</tr>
<tr>
<td>29</td>
<td>C</td>
<td>59</td>
<td>B</td>
<td>89</td>
</tr>
<tr>
<td>30</td>
<td>B</td>
<td>60</td>
<td>A</td>
<td>90</td>
</tr>
</tbody>
</table>

### Figure 3
**Family Assignment**
1. For best family sequence number.
2. Count at serial groups that are not recruiting families.
3. Double not family groups that are nonrecruiting.
4. Create the first eligible pair of serial.
5. Select group from the corresponding family group schedule.
6. Discourage assignment, assign another group to a nonrecruiting.
7. For recruitment, sign eligible groups above line, circle next task. Repeat steps 5 to 7.
## FIGURE 5

% CHANGE OVER MOO

<table>
<thead>
<tr>
<th>Measure</th>
<th>Meas2. Hl</th>
<th>GECU</th>
<th>TEST DRUG</th>
<th>PLACEBO</th>
<th>TOTAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Min</td>
<td>89.117</td>
<td>-69.917</td>
<td>-6.426</td>
<td>-8.063</td>
<td>-6.455</td>
</tr>
<tr>
<td>25%</td>
<td>95.333</td>
<td>71.457</td>
<td>7.473</td>
<td>12.487</td>
<td>13.035</td>
</tr>
<tr>
<td>50%</td>
<td>90.417</td>
<td>71.457</td>
<td>7.473</td>
<td>12.487</td>
<td>13.035</td>
</tr>
<tr>
<td>75%</td>
<td>89.117</td>
<td>-69.917</td>
<td>-6.426</td>
<td>-8.063</td>
<td>-6.455</td>
</tr>
<tr>
<td>Range</td>
<td>95.333</td>
<td>71.457</td>
<td>7.473</td>
<td>12.487</td>
<td>13.035</td>
</tr>
<tr>
<td>Mean</td>
<td>90.417</td>
<td>71.457</td>
<td>7.473</td>
<td>12.487</td>
<td>13.035</td>
</tr>
<tr>
<td>SD</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
<td>0.75</td>
</tr>
</tbody>
</table>

* Denotes significance at α = 0.05 Type I error level by Wilcoxon signed ranks statistic.