

INTERNATIONAL JOURNAL OF THE FACULTY OF AGRICULTURE AND BIOLOGY,  
 WARSAW UNIVERSITY OF LIFE SCIENCES, POLAND

## SOFTWARE TRICKS AND TIPS

### Using SAS to Generate Trial Designs and Plot Maps

Previous contribution in this series on SAS tips and tricks introduced the basic aspects of macro variables (van Santen, 2009a,b) and PROC SQL introduced as time saving tools of the SAS/BASE installation (van Santen, 2010). In this contribution I demonstrate how these tools might be combined and used in conjunction with an EXCEL spreadsheet to generate field trial designs and plot plans.

#### Scenario:

The incentive for creating this approach was a call for help from my colleague Dr. Mark Sulc at The Ohio State University, who had been asked on short notice to supervise the management of the soybean cultivar evaluation program. The 2011 program consisted of four trials (NV, RRE, RRL, SEED) conducted as a standard randomized complete block with four replicates at two locations (1, 2) in each of three regions (N, C, S). The objective was to generate a SAS program that would be simple yet flexible enough to be usable for several years. The trials are laid out in the same manner at all six locations. In planting direction, each block would be four ranges deep but the number of entries within each group might vary from location to location. The four blocks are stacked, resulting in 16 ranges (Fig. 1). If a trial did not have enough entries for four complete ranges, extra entries were added to create complete rectangular blocks. The process was broken down into three steps.

Range 16	NV	RRE	RRL	SEED
Range 15	NV	RRE	RRL	SEED
	.	.	.	.
	.	.	.	.
Range 2	NV	RRE	RRL	SEED
Range 1	NV	RRE	RRL	SEED

Figure 1. Trial arrangement

#### Step 1. Prepare Trial Entry Spreadsheet

In cooperation with my colleague I decided that the entries for each trial x region combination be listed in a separate worksheet; the two locations in a region had the same entries. This would also offer the flexibility of substitution if an entry had to be replaced in a trial. A meaningful label was associated with a given sheet, e.g., 11N\_RRE refers to the early-maturing 2011 Roundup-Ready cultivar trial conducted in the Northern Ohio region. Thus there were 12 (= 3 regions x 4 trials) worksheets with trial entries ([see supplemental EXCEL file](#)).

The “START” worksheet is used to create a dummy SAS dataset as a template for combining all entries into a single SAS dataset. In particular, the repeat function (REPT) is used to create a template for string variables. The number used in this function is the maximum string length encountered for each variable, easily obtainable through the LEN function in EXCEL.

The “TRIALS” worksheet (Table 1) contains the data for the basic trial layout. It calculates the starting and ending row number for each trial from the number of entries (Entry\_N). The only requirement is that this number be a multiple of 4. Variables Row\_Start and Row\_End are used by SAS to create the plot plan. Simply enter another multiple of 4 into the ENTRY\_N column and watch the resulting changes. The macro variables for reading the data into SAS are create in EXCEL. Please read the notes in this worksheet in the supplemental EXCEL file ([Ohio Soybean Trials 2011.xls](#)).

Table 1. Excerpt from the “TRIALS” worksheet.

Loc	Trial_name	Entry_N	Block_N	Range_N	Row_N	Trial_Order	TEST	Row_Start	Row_End	SAS_macro_var
N1	11N_NV	24	4	16	6	1	TRUE	1	6	%import(11N_NV);
N1	11N_RRE	24	4	16	6	2	TRUE	7	12	%import(11N_RRE);
N1	11N_RRL	68	4	16	17	3	TRUE	13	29	%import(11N_RRL);
N1	11N_SEED	24	4	16	6	4	TRUE	30	35	%import(11N_SEED);
N2	11N_NV	24	4	16	6	1	TRUE	1	6	%import(11N_NV);
N2	11N_RRE	24	4	16	6	2	TRUE	7	12	%import(11N_RRE);
N2	11N_RRL	68	4	16	17	3	TRUE	13	29	%import(11N_RRL);
N2	11N_SEED	24	4	16	6	4	TRUE	30	35	%import(11N_SEED);

## Step 2. Import necessary trial information into SAS

The program 01\_IMPORT ([see supplemental files](#)) does four things: (1) It creates a SAS dataset from the TRIALS worksheet; (2) uses a macro to read all trial entries; (3) uses PROC SQL to create a permanent dataset for trial entries; and (4) creates and prints to the output window the temporary dataset TRIALS with location and test information. This gives the operator another chance to verify the trials to be included.

## Step 3. Generating the trial design

As a matter of principle I advise clients to separate data input from data processing. Hence, the program 02\_PLAN\_SQL contains only the processing commands. The datasets procedure at the top of the program `proc datasets library=work kill memtype=data;quit;` ensures that all temporary datasets are cleared before the randomizations are appended to the base file. The actual program to generate plot plans has six parts:

1. **PROC SQL to generate macro variables.** The SQL procedure offers a compact approach to generate macro variables ‘on the fly’. You can now see why I structured the TRIALS worksheet the way I did. It took several approaches to arrive at this solution that is easy to understand yet flexible enough to allow modifications down the road.
2. **The macro “doit”.** The approach in developing this macro was to initially work with a single trial at a single location to put all the parts in place. Once that worked the way I envisioned I wrapped the macro around it. The first part is to generate the randomization for a given trial at a given location (**PROC PLAN**). Range and row numbers (plots) are then created with do loops in a data step. Randomization and plots are then merged in another data step and the entry information added using PROC SQL. The combined randomization file for all trials and locations is created through PROC APPEND. This procedure is the main reason behind the START worksheet

because it sets the string length for company, cultivar names etc. to the maximum length encountered. It is consistent with my ‘pay me now or pay me later approach’, i.e., a little more work at the beginning of the process avoids problems down the road.

3. The next part “Exporting randomization list” has been commented out. If the “/\*” and “\*/” are removed it offers the use the opportunity to create a permanent randomization dataset for each location x trial combination as well as creating a separate EXCEL worksheet for each.
4. The final portion of the “doit” macro entitled ‘Creating Plot Plan and Exporting it’ uses the procedures SQL, TRANSPOSE, and EXPORT to convert all the information for the body of the plan to strings, transforming it to a rectangular layout, and transferring it to an EXCEL worksheet.
5. The macro call %doit is placed following the macro routine.
6. Finally, a permanent randomization dataset is created and exported to EXCEL.

#### **Step 4. Additional post-processing not involving SAS**

The next step is to copy and paste formats in EXCEL, a process that can also be automated through an EXCEL macro but that is beyond the scope of this contribution. In this case I included The “Template\_Map” worksheet containing the basic formatting elements. That template is copied and format pasted twice (1) in a row fashion for row height and cell attributes and (2) in column fashion to set column width.

Finally, to print labels for packeting seed the randomization worksheet is read into a database. Any database would work but we tend to use the cross-platform Filemaker database (Filemaker Inc, Santa Clara, CA, USA) because it is easy to use, particularly for printing labels.

A professional SAS programmer could achieve all that by issuing appropriate commands from within SAS. I contend that this exceeds the ability of most practicing agronomists, hence a solution that accommodates this reality.

#### **References**

- van Santen E. (2008). Make a project folder home base for SAS. *Communications in Biometry and Crop Science Crop Science* 3 (1), 1-2.
- van Santen E. (2009a). SAS Macro Variables. *Communications in Biometry and Crop Science Crop Science* 4 (1), 1-2.
- van Santen E. (2009b). SAS Macro Variables and ARRAY Processing. *Communications in Biometry and Crop Science Crop Science* 4 (2), 40-41.
- van Santen E. (2010). PROC SQL vs. DATA Step or Anything you can do, I can do better. *Communications in Biometry and Crop Science Crop Science* 5 (2), 66-68.
- van Santen E. (2011). SAS Functions. *Communications in Biometry and Crop Science Crop Science* 6 (1), 1-3.

**contributed by Edzard van Santen**

Forage Breeding and Genetics, Dept. of Agronomy and Soils,  
Auburn University, AL 36849-5412.  
E-mail: vanedza@auburn.edu