

Streamlining Data Analysis

Edzard van Santen, Auburn University, AL

ABSTRACT

One of the software tools used by researchers in agronomy for statistical data analysis is SAS® (SAS Institute, Cary, NC, USA). Even though new procedures offer superior results they may not be readily adopted if they are too cumbersome to use for the practicing researcher who is not a SAS professional. In agronomic research multiple response variables are often measured for a given experiment but newer analysis procedures available in SAS, such as PROC MIXED or PROC GLIMMIX, only allow the analysis of a single response variable at a time. In this paper I describe time saving measures that take some of the drudgery out of data analysis. The measures involve a judicious use of macro variables and a linearization of datasets with multiple response variables that, in SAS parlance, converts a dataset from a wide to a long data format. All original observations are now contained in a single variable "response" and the original variable names are assigned to a new variable such as "resp_name" that is then used as a BY group during analysis. All output that is collected during analysis through either an out= statement or ODS will be contained in a single dataset.

INTRODUCTION

One of the main software tools used for data analysis in agronomic research at many public academic institutions in the USA is SAS®. This software has the advantage that it is innovative, incorporating new analysis techniques on a regular basis, and generally affordable to university-based end users because of institutional licensing agreements. Users, who develop new solutions in response to new needs, often linking existing procedures through macros, drive the process of innovation to no small extent. If these macros then become used widely and, fulfilling a general need, the most successful ones may then become codified as new procedures. Yet while macros are extremely powerful, they often remain a mystery to researchers and students alike, who use SAS as a tool to fulfill their data analysis needs. These users have no interest in becoming SAS programmer analysts but simply need a convenient tool to accomplish their analysis tasks. They are familiar with basic input and output steps in addition to the SAS procedures commonly used for data analysis in their area of expertise.

Unlike least squares based analytical procedures, where all response variables of interest can be included on the left side of the model statement, likelihood-based procedures require a separate run for each response. One such procedure is PROC MIXED. As Piepho et al. (2003) pointed out rather succinctly: "While the mixed model framework has been well developed over the past 20 years and analyses are now easy to perform with modern statistical software, the use of mixed models in agricultural research does not seem to have reached the level it deserves". While these authors lay the burden on an overly complex presentation of the subject matter in statistical texts I present in this contribution simple procedures that may ease the load in more practical terms. These procedures are particularly helpful during the early part of model development such as checking assumptions and modeling the covariance structure. The approach could be described as the ABC of data analysis.

ASSEMBLE ALL RELATED INFORMATION

Efficiency is gained when all information pertaining to a particular analysis (research) project is collected into a single project folder with subfolders for various aspects such as literature or data analysis. Data destined for analysis and SAS programs are located in a single folder. I use the suggestion by Heaton (2003), who demonstrated how to modify the SAS icon, and place this icon into every analysis folder. This folder is then treated as home base when SAS is invoked by clicking on the icon. The other option would be to right-click on a SAS program icon, but clients tend to forget.

BREAK DOWN TASKS

My role as a consulting statistician in a university setting is to teach clients how to do the analysis. Breaking down the process in small chunks is a teaching aid. Furthermore, adopting a naming convention for programs that combines a number with a name describing what a program does leaves the clients not only with reusable program snippets devoted to a single or few related tasks but also reminds them of the order in which these programs were run. Examples are 01_IMPORT_XL AND LINEARIZE, 02_TABULATE, 03_RESIDUALS_GLIIMMIX, where the first program imports data from EXCEL spreadsheets and then converts the dataset from the wide to the long form for multiple response variables, the 2nd uses PROC TABULATE to investigate the data structure, and the 3rd uses PROC GLIMMIX to investigate the behavior of residuals.

CONVERT FROM WIDE TO LONG

Most experiments in agronomic research involve multiple response variables such as yield, dry matter percentage, plus content of various chemical compounds. Linearizing these data sets (WIDE to LONG conversion) through ARRAY processing in the data step streamlines data analysis.

```
%let array_columns = Height Density DM_Perc DM_yld;
data harvest_linear;
  set harvest1 harvest2;
  array raw(*) &array_columns;
  do resp_n=1 to dim(raw);
    response=raw(resp_n);
    resp_name=Vname(raw(resp_n));
    output;
  end;
  drop &array_columns;
run;
proc sort;by resp_name;run;
```

The original response variable name is assigned to the BY variable resp_name. The user does not have to remember how exactly a particular response variable was named or spelled. Furthermore, every time an analysis is done BY resp_name this variable name is included in any output dataset that is created. This is quite helpful in communicating with clients.

DETECT DATA INTEGRITY AND STRUCTURE

This precedes the actual analysis of the data. The objective is to ascertain that the data is correct and treatments are correctly assigned to experimental units. This is accomplished primarily with PROC TABULATE using statistics such as N, MEAN, MIN, and MAX in constructing tables.

```
proc tabulate noseps formchar='          ';
class loc year harvest block resp_name;
var response;
table year*loc*harvest, resp_name*response*N*f=6.0/rt=15;
run;
```

The 2-way table generated by this program would give an immediate view of the number of observations present for each year x location x harvest x response variable combination without having to remember the name of each response. If the result does not confirm the client's expectation the table command can be modified easily for an in-depth look at the data.

EVALUATE RESIDUAL BEHAVIOR

The analysis needs for most of my clients would fall in the broad category analysis of variance, including regression (covariance) approaches. PROC GLIMMIX is a marvelous tool because a panel of studentized residuals can be created with a simple programming statement. ODS can be used to exclude all output not of immediate interest and ODS OUTPUT to capture particular tables.

```
ods html; ods graphics on; ods select StudentPanel;
proc glimmix data=a plots=StudentPanel;
  by resp_name;
  class          ;
  model response =          / dist=    ddfm=  ;
  random          ;
  random _residual_/ ;
  ods output FitStatistics=Fit_out;
run;
ods graphics off; ods html close; quit;
```

Again, this initial look at the data is made easier because all residual panels are created in a single run. Output of interest is collected via ODS; each response variable is identified through the BY statement. Often response variables can now be grouped based on their residual behavior and then analyzed/ modeled separately.

FIT COVARIANCE STRUCTURE

The analysis can be streamlined by creating a series of macro variables that identify groups of response variables or individual responses. Selected covariance structures can then be obtained by wrapping a macro around PROC MIXED or PROC GLIMMIX and collection of all relevant information into a data set, followed by PROC RANK to identify the best fitting structure. My approach to teaching clients the macro approach is to first test the components

and then wrap a simple macro around a group of programming steps. Again, the objective is not to teach macro programming but help clients with simple tools to accomplish their data analysis needs.

```
proc datasets library=work kill;run;
%macro Fit_Cov(type);
  ods exclude all;
  proc glimmix data=Select;
    by resp_name;
    class Block Plot Factor1 Time;
    model response = Factor1|Time/dist=normal;
    random Block Block*Factor1 Block*Time;
    random _residual_/subject=Plot type=&type;
    ods output Fitstatistics=Fit_out;
  run;
  Data Fit_out1; set Fit_out;
  where substr(Descr,1,4)="AICC";
  Cov_type="&type";
run;
proc datasets;
  append Base = Fit_all data=Fit_out1 force;
run;
ods select all; quit;
%mend Fit_Cov;

% Fit_Cov(CS);
% Fit_Cov(UN);

proc rank data = Fit_all out=AICCranks;
  by resp_name; var value; ranks rank;
run;

proc print data=AICCranks_sorted; where rank=1;
run;
```

The code above does some R-side modeling of the covariance structure that is induced by taking repeated observations from a given experimental unit (plot) over time. For sake of simplicity only the two extreme covariance structures in terms of number of fitted parameters are used in this example. The fit statistics are collected via PROC DATASETS, and then ranked and the lowest rank printed. A similar approach can be used when dealing with heterogeneous variances that might require the creation of variance groups.

CONCLUSION

I have developed and used the approach to streamlined data analysis for close to 10 years. Analysis of a linearized dataset conducted BY group results in selected output being transferred to a single dataset; no additional programming is needed to collect the output. Actual data analysis, hypothesis testing, and means comparisons will be somewhat anticlimactic once these basic issues have been dealt with. The main advantage to breaking down the overall data analysis project into small tasks is that clients for whom I have conducted data analyses repeatedly over the years begin to take over the more routine aspects. This frees up my time that I can then devote to more challenging problems.

REFERENCES

Heaton, E. 2003. SAS® Systems Options Are Your Friends. Proc. 28th SUGI meeting (<http://www2.sas.com/proceedings/sugi28/067-28.pdf> ; verified 10 July, 2009).
Piepho, H.P. Büchse, A., and Emrich, K. 2003. A hitchhiker's guide to mixed models for randomized experiments. Journal of Agronomy and Crop Science 189: 310-322.

ACKNOWLEDGMENTS

I gratefully acknowledge my former student Dr. Ravi Vaylay (Cybrid Inc, Harrisburg, PA) for his willingness to explain some of the finer points of SAS programming. I further acknowledge current and former students in AGRN 7080

(Experimental Methods) whom I have had the pleasure to teach and who provided the challenge to improve my skills in teaching and statistical consulting.

CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Edzard van Santen, PhD

Professor

Dept. of Agronomy and Soils

202 Funchess Hall

Auburn University, AL 3684905412

Work Phone: 334-844-3975

Fax: 334-844-3945

E-mail: evsanten@acesag.auburn.edu

Web: <http://www.ag.auburn.edu/agrn/faculty/VanSanten.html>

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.