

USING SAS® MACRO FACILITIES TO SIMPLIFY A LARGE MULTI-EVENT STATISTICAL ANALYSIS

Douglas P. DeYoung, The Upjohn Co.

Introduction

How can one simplify the development of a system for analysis and formatted output of a large research project? This paper will look at how SAS macros and macro variables are utilized to generalize the statistical analysis and resulting creation of over 11,000 statistical tables. Frequency distributions, analysis of variance, statistical means and statistical outliers are analyzed, as well as some 600 graphical analyses of variables. Utilizing the SAS macro capabilities and SAS/GRAPH®, we have been able to both streamline and standardize the programming for this drug safety study.

The task at hand

The project is a target animal safety study which will last approximately three years and include analysis of physical and reproductive health of the animals and their offspring. Statistical analysis will include primarily frequency distributions of event occurrence and/or a listing of missing/bad data values, arithmetic and least-squares means, statistical outliers and analysis of (co)variance. The analysis includes some 285 different measurements for 150 animals in 5 different dose levels, and 7 phases. In all, approximately 2 million data points in 49 different source data sets are included in the analysis.

The goals

Quality of analysis

Quality of any drug study must always be first. Cost and time required for this project make it imperative that the analysis be complete and correct the first time.

Flexibility

Inherent change in a project of this size, duration and nature compel us to develop a system with as much flexibility as possible. We needed to be able to analyze sub groups as well as all subjects, be able to create new analyses quickly, and add, drop or change parameters to provide prompt reporting which could impact the conduct of the project as well as the time requirements for completing the final report.

Validation

As this is a trial which requires FDA approval, validation of the programming is required for the analysis to be acceptable.

Economy of programming resources

When isn't this an issue?

Readability

Review by the FDA necessitates that the programs be understandable by the non-programmer, non-statistician.

Consistency

The analyses utilized needed to be consistent across all aspects of the study for effective comparison, and reduced validation time.

The approach

The initial plan was to develop a consistent approach which would then be utilized by modifying the programs for each analysis to reflect changes to the files, fields, parities, phases and analysis characteristics. As we began to address the project, it became evident that the magnitude of the data to be analyzed required a different approach if the project was to be completed in either a timely or economical fashion. We began to consider how

we might develop a system which would provide the flexibility needed without the "copy and modify" requirements. We concluded that a modular approach would be beneficial in reducing the need to rewrite procedures. SAS macros provided the mechanism to achieve this, but, did not provide flexibility which would allow them to be utilized in different ways, without writing multiple routines. By using macro variables, we were able to develop one macro for a statistical analysis providing the flexibility to identify varying elements (e.g. different models in proc GLM). Further, by utilizing SAS macros which can be included in any program, changes can be implemented in a single location and carried through into all calling programs. We have utilized the autocall library feature to provide "automatic" incorporation of called macro routines.

The above approach has been implemented through three major areas as noted and illustrated below.

1) Macro variables are employed to identify various values used to specify a situation analyzed as well as for output formatting. Macro variables expected to be used in multiple programs, and their "default" values are maintained in a master file (MVGENERAL SASVAR). This file is "included" in each SAS program utilizing the macro variables. Default values can be overridden within the individual program.

```

*-----*
[ MVGENERAL SASVAR ]
[ ]
[ Program description: ]
[ Macro variable definitions for all program ]
[ ]
[ Written by: DP DeYoung ]
[ Date: 16 Jun 1992 ]
[ ]
[ Modification History: ]
[ ]
*-----*

*-----*
[ DATE OF LAST DATA RECEIVED ]
[ Change to reflect current date of data received ]
*-----*

%let asofdate=15 June 92;

```

```

*-----*
[ Dose groups in study ]
*-----*
%let numgrps=5;

*-----*
[ Lactation initial values ]
*-----*
%let numlact=3;
%let strtlact=1; %let endlact=3;

*-----*
[ Periods in each lactation ]
*-----*
%let p=Pre-treatment t=Treatment d=Dry *;
%let numpers=3;
%let strtpers=1; %let endpers=3; %let allpers=ptd;

*-----*
[ Parity initial values ]
*-----*
%let numpars=2;
%let strtpars=1; %let endpars=2; %let allpars=cowhef;

*-----*
[ Variables measured initial values ]
*-----*
%let nummeas=1;
%let strtmeas=1; %let endmeas=1; %let allmeas=a;

*-----*
[ Analysis requested initial values ]
*-----*
%let i=missing values m=means o=outliers v=analysis of variance *;
%let strtanly=1; %let endtanly=4; %let reqtanly=imov;

* Frequency Analysis values *;
%let numvals=1;
%let strtval=1; %let endval=1; %let allvals=one;
%let ncrttp=1; %let numvald=2;

*-----*
[ Output formatting variables ]
*-----*
%let duringb=@25 'During the ';
%let duringl=@32 'During the ';
%let duringp=@20 'During the Pre-Treatment Period of ';
%let duringt=@22 'During the Treatment Period of ';
%let duringd=@25 'During the Dry Period of ';
%let duringa=@30 'Through end of ';
%let lactno0= ;
%let lactno1='First Lactation';
%let lactno2='Second Lactation';
%let lactno3='Third Lactation';
%let lactnob='First & Second Lactations';
%let glmoutn=@36 'Analysis of Variance';
%let glmoutc=@35 'Analysis of Covariance';
%let glmoutr=@27 'Repeated Measures Analysis of Variance';
%let glmouth=@26 'Repeated Measures Analysis of Covariance';

```

2) Macros were written for each of the statistical procedures (frequency, missing

values, arithmetic and least-squares means, analysis of variance and statistical outliers) which were to be used multiple times. Parameters and macro variables provide the means for passing control information between the macros and the program.

A macro is utilized to call the routine which determines and outputs e.g. the arithmetic and least-squares means with appropriate independent variable(s), and type of analysis for each phase. Parity is checked in the mainline program and only records with the appropriate parity are included in the data set being analyzed. The macro to loop for all requested means follows.

```

*-----*
[ MACRO: ALLMEANS ]
[ ]
[ Program description ]
[ Call the routine to get and output means ]
[ for KEEP data sets for each lactation ]
[ and period. ]
[ ]
[ Parameters: independent variable(s) ]
[ variance type: ]
[ N - ANOVA ]
[ C - ANCOVA ]
[ R - Repeated Measure ANOVA ]
[ B - Repeated Measure ANCOVA ]
[ ]
[ Written by: DP DeYoung ]
[ Date: 03 Apr 1992 ]
[ ]
[ Modification History: ]
[ ]
*-----*

```

```
%macro allmeans(indepvar,vartype);
```

```
%* FOR EACH LACTATION AND PERIOD ;
%local l; %local p;
```

```
%do l=&strilact %to &endlact;
  %if &l=3 %then %do /* Only one period in lactation 3 */;
    %let period=%substr(&allpers,1,1);
    %writmean (&l,&period,&indepvar,&vartype);
  %end;
  %else %do;
    %do p=&strippers %to &endpers;
      %let period=%substr(&allpers,&p,1) /* Get single char
period */;
    %end;
  %end;
```

```
%* NO COVARIATE IF FIRST LACTATION AND FIRST
PERIOD ;
  %if &l=1 and &p=1 %then %do;
    %if %upcase(&vartype)=C %then
      %writmean(&l,&period,&indepvar,n);
```

```

  %else %if %upcase(&vartype)=B %then
    %writmean(&l,&period,&indepvar,r);
  %else %writmean(&l,&period,&indepvar,&vartype);
%end;
```

```
%* NOT REPEATED MEASURE IN DRY PERIOD OR
THIRD LACTATION ;
```

```

  %else %if %upcase(&period)=D or &l=3 %then %do;
    %if %upcase(&vartype)=R %then
      %writmean(&l,&period,&indepvar,n);
    %else %if %upcase(&vartype)=B %then
      %writmean(&l,&period,&indepvar,c);
    %else %writmean(&l,&period,&indepvar,&vartype);
  %end;
```

```
%* MUST BE NORMAL ;
```

```

  %else
    %writmean (&l,&period,&indepvar,&vartype);
  %end;
%end;
```

```
%mend allmeans;
```

3) Analysis and output formatting

Analysis and output methods likewise were developed as macro programs for consistency and simplification. Again we have made significant use of macro variables for values which are changed (e.g. table numbers, variable measured) based on the analysis and variables being reviewed. The following is the macro for determination and output of arithmetic and least-squares means.

```

*-----*
[ MACRO: WRITMEAN ]
[ ]
[ Program description: ]
[ Get and write files for output of ]
[ least-squares and arithmetic mean values ]
[ for both parities and all measures of ]
[ the files. ]
[ ]
[ Parameters: lactation ]
[ period ]
[ independent variable(s) ]
[ variance type ]
[ N - ANOVA ]
[ C - ANCOVA ]
[ R - Repeated Measure ANOVA ]
[ B - Repeated Measure ANCOVA ]
[ ]
[ Written by: D.P. DeYoung ]
[ Date: 17 Apr 1992 ]
[ ]
[ Modification History: ]
[ ]
*-----*

```

```

%macro writemean(lactloop, period, indepvar, vartype);

%local i; %local m;

/* DETERMINE TYPE OF MEAN */
/* NOT REPEATED MEASURE */
%if %upcase(&vartype)=N or %upcase(&vartype)=C %then %do;
  %let reptime=NO;
  %let srtparam=
    by par &indepvar;
%end;
/* REPEATED MEASURE */
%else %do;
  %let reptime=YES;
  %let srtparam=
    by par &indepvar interval;
%end;

/* FOR EACH MEASURED VALUE ;
%do m=&strtimeas %to &endtimeas;
  %let measure=%substr(&allmeas,&m,1);
  %if &reptime=NO %then %do;
    %let glmparam=
      %str(class &indepvar;
        model &&varnam&measure=&indepvar;
        lsmeans &indepvar /out=lsnout;);
  %end;
  %else %do;
    %let glmparam=
      %str(class &indepvar interval;
        model &&varnam&measure=&indepvar
&indepvar.*interval;
        lsmeans &indepvar.*interval /out=lsnout;);
  %end;

  /* GET THE LEAST-SQUARES MEANS ;
  proc glm data=keep&lactloop&period noprint;
    by par;
    &glmparam;
  run;
  proc sort data=lsnout;
    &srtparam;
  run;

  /* GET THE ARITHMETIC MEANS ;
  proc means data=keep&lactloop&period noprint mean std n;
    var &&varnam&measure;
    &srtparam;
    output out=meansout mean=mean&measure;
  run;

  /* COMBINE BOTH MEANS ;
  data lsm&lactloop&period&measure;
    merge meansout lsmout;
    &srtparam;
    %getdose;
  run;

  proc sort;
    &srtparam;
  run;

  /* OUTPUT BY PARITY ;
  %do i=&strtpars %to &endpars;

```

```

%let partype=%substr(&allpars,8*(&i-1)+1,1);
%let parity=%substr(&allpars,3*(&i-1)+1,3);

data temp1;
  set lsm&lactloop&period&measure;
  retain noreps 'no' yesreps 'yes';
  if par=&parity;
run;

proc sort;
  by tmnt_grp;
run;

data _null_;
  set temp1;
  by tmnt_grp;
  file print notitles header=pagetop;
  if upcase(&reptime.reps)='NO' then
    put / @17 dose @42 _freq_ @50 mean&measure 9.4 @65
lsmean 9.4;
  else do;
    if first.tmnt_grp then put;
    put @10 dose @26 interval @42 _freq_
      @50 mean&measure 9.4 @65 lsmean 9.4;
  end;
  return;

pagetop:
  if upcase(&reptime.reps)='NO' then
    put @39 " T a b l e
&&tmn&lactloop&period&measure&partype" //
    @43 'Means'//
    &&labl&measure&partype/
    &&during&period &&lactno&lactloop//
    @69 'Least'//
    @38 'Sample' @49 'Arithmetic' @67 'Squares'//
    @17 'Dose' @40 'Size' @55 'Mean' @70 'Mean'//
    @17 '-----';
  else
    put @39 " T a b l e
&&tmn&lactloop&period&measure&partype" //
    @43 'Means'//
    &&labl&measure&partype/
    &&during&period &&lactno&lactloop//
    @26 'Sample' @69 'Least'//
    @21 'Time Within' @38 'Sample' @49 'Arithmetic' @67
'Squares'//
    @10 'Dose' @23 'Lactation' @40 'Size' @55 'Mean' @70
'Mean'//
    @10 '-----'
    @50 '-----';
  return;
run;
%end;
%end;

%mend writemean;

```

The results

When the initial program was written, it was 28 pages in length. There were multiple

occurrences of similar procedures included (e.g. GLM with different models for ANOVA, ANCOVA and repeated measures depending on the variable and phase). It was expected this program would be copied and altered for differences in files, fields, models etc. As project parameters changed, every program which included that parameter would need to be modified.

We currently have 6 major analysis routines: missing values, least-squares means, analysis of (co)variance, outliers, frequency and graphs. Generally the calling program consists of only a page or two of code (including documentation). The mainline program need only read the required data sets, do necessary calculations, construct data set(s) for analysis and call the appropriate macros.

A mainline analysis program example:

```

*-----*
[ ANLFILE? SAS ]
[ ]
[ Program description: ]
[ Analysis of the variables in ??? file ]
[ ]
[ Written by: D.P. DeYoung ]
[ Date: 28 May 1992 ]
[ ]
[ Modification History: ]
[ ]
*-----*

*-----*
[ CMS DEFINITIONS ]
*-----*
options mautosource sasautos='allmacs machib f
      center pageno=1 pagesize=60 linesize=90;
libname sasdata 'g';
libname tempdata 'f';
proc printto print='anlfile? listing a' new;
run;

*-----*
[ ASSIGN MACRO VARIABLES ]
*-----*
%include 'mvfile? sasvar f;
%let vartype=b /* Repeated Measures ANCOVA */;
%let indepvar=tmnt_grp;

*-----*
[ TEMPORARY BATCH RUN CHANGES ]
*-----*
%macro newchoic;
%let endlact=1;
%let endpers=2;

```

```

%let reganly=v /* TEMP TEST */;
%mend newchoic;

```

```

*-----*
[ SELECT INTERACTIVE OR BATCH AND ]
[ SET VARIABLES MEASURED ]
[ ]
*-----*
%choice (interact=N);
%let strtmeas=1; %let endmeas=2; %let allmeas=ws;

*-----*
[ PROGRAM ENTRY POINT ]
*-----*

```

READ APPROPRIATE FILES AND DO REQUIRED CALCULATIONS TO OBTAIN DATA FOR ANALYSIS DEPENDING ON SPECIFIC ANALYSIS REQUIRED

```

*-----*
[ SORT KEEP FILES BY PAR AND TMNT_GRP ]
[ THEN CALL OUTPUT SUBROUTINES ]
[ ]
*-----*

```

```

%keepsort;
%analyses(&reganly);

```

Output generated has the following format:

Table A7.1.3
Means
Body Weight (Lbs) for Cows
During the Pre-Treatment Period of First Lactation

Dose	Sample Time Within Lactation	Sample Size	Arithmetic Mean	Least Squares Mean
Control	1	15	1374.8000	1374.8000
Control	2	15	1362.2000	1362.2000
Control	3	15	1304.4667	1304.4667
Dose 1	1	15	1413.4667	1413.4667
Dose 1	2	15	1361.6667	1361.6667
Dose 1	3	15	1332.6000	1332.6000
Dose 2	1	15	1306.3333	1306.3333
Dose 2	2	15	1354.2667	1354.2667
Dose 2	3	15	1302.3333	1302.3333
Dose 3	1	15	1405.1333	1405.1333
Dose 3	2	15	1355.4000	1355.4000
Dose 3	3	15	1295.5333	1295.5333
Dose 4	1	15	1407.2667	1407.2667
Dose 4	2	15	1370.6000	1370.6000
Dose 4	3	15	1322.0667	1322.0667

Two other major areas have been tied in to the macro concept effectively which provide us with additional capability in the system. First, as noted above, a number of analyses need to include a graphical comparison of individual dose to control. This has been done using SAS/GRAPH with the routines called

from the main SAS programs when required. Second is an interactive user screen allowing the user to choose to run any analysis for all considerations or select only specific values of phase, animal, parity, and type of analysis. The interactive programming is accomplished using %window, with a macro variable in the program used to indicate whether interactive or batch processing is to be used.

Keys to success

Planning/design

Careful planning and continuous design of the modules is required to provide effective and flexible modules. In addition, the database design, though not directly relevant, is critical to achieving the same goals and will facilitate the macro utilization.

Teamwork, communication,

The importance of communication within the project team to provide awareness of applications and potential refinements has been the foundation on which the benefits have been obtained. Maintaining consistency of labeling, and programming approach likewise are critical to successful implementation.

Standards/conventions

Consistency in identifying both macro routines and macro variables is imperative with a project of any size, but especially important when dealing with this magnitude. Naming conventions for each have been established, with a goal of consistency in names, and attempting to use names which clarify function.

Documentation

Program in-line documentation is critical to understanding the programs especially as using SAS macros and macro variables tends to reduce visibility within the programs. Also, documentation of macro routines and macro variables and their definition, application and default values is necessary to provide maximum application.

Consistency

Our goal is to use available developed macros wherever possible, and where unique situations are involved attempt to utilize existing macro routines as a starting point.

Summary

The use of SAS macro variables for project parameters and SAS macros for various analysis and output modules has contributed significantly toward several of our major project goals including quality of analysis, flexibility, validation, economy of resources, program readability and consistency, and has enabled us to achieve a more manageable project.

Acknowledgements

I wish to thank H. Jeff Rostami and Mohammed Al-Adhami for their significant roles in formulating and implementing these ideas and their technical review of this paper.

Trademark Notice

SAS and SAS/GRAPH are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries.
® indicates USA registration.

Author

Douglas P. DeYoung
The Upjohn Co.
301 Henrietta St. 9162-32-2
Kalamazoo, MI 49007-3776
(616) 384-9661 FAX (616)385-5286