A Macro That Generates the PROC GLM LSMEANS Table From the LSMEANS Output Data Set

Michael A. Walega
Bio-Pharm Clinical Services, Inc.

Introduction

With each new release of the SAS® System, more and more PROCedural output has been made available to the user in the form of output data sets. Consider PROC GLM. Prior to v6.06, no procedural output was available in an output data set. To access the information contained in GLM output, programmers were forced to become proficient in the use of DATA step stripping methods. Version 6.06 heralded the availability of output data sets in PROC GLM. Currently, output data sets can be created from three (3) GLM statements as shown in Table 1.

Table 1

<table>
<thead>
<tr>
<th>GLM Statement</th>
<th>Option</th>
<th>Data Set</th>
<th>Data Set Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROC GLM</td>
<td>OUTSTAT=</td>
<td>dependent variable, sources/type of error, df, ss, f-statistic, p-value</td>
<td></td>
</tr>
<tr>
<td>OUTPUT</td>
<td>OUT=</td>
<td>original data set, diagnostics dependent variable, analysis variable, lsmean, se, covariance matrix</td>
<td></td>
</tr>
<tr>
<td>LSMEANS</td>
<td>OUT=</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Not all PROCedural output can be captured by these output data set options. For example, much of the output from a GLM repeated measures analysis is not available in an output data set. And while the LSMEANS output data set contains p-values, covariances, and the least squares means and standard errors, the actual LSMEANS table cannot be recreated without additional programming. Missing from the output data set is the matrix that contains the p-values from the pairwise comparisons \( \text{LSM}_i = \text{LSM}_j \), \( (i=1,...,t; j=1,...,t; i<j) \). One way to recreate this matrix is to strip it from the GLM output using PROC PRINTTO and a DATA step. Effective, but messy. A more elegant method is described below.

The LSTABLE Macro

The macro requires that the LSMEANS statement in PROC GLM be structured as follows:

```sas
LSMEANS depvar / cov stderr out=dsname;
```

`DEPVAR` is the dependent variable for which the pairwise comparisons will be generated; the `COV` option causes the variance/covariance matrix to be output to `dsname`; and `STDERR` puts the standard errors of the least squares means in the output data set.

Note that the macro is useful for only one (1) dependent variable, as the `COV` option is not appropriate for multiple dependent variables. The LSTABLE macro also requires two other macro variables: one that contains the number of unique values of the dependent variable (`&NDEPVAR`), and one that contains the error degrees of freedom from the overall GLM analysis (`&ERR_DF`). These can be generated via any number of different methods. The code for the macro is shown in Figure 1. A brief description of the inner workings of the macro follows.

The first DATA step accomplishes two tasks. The initial DO loop places all of the covariances into one record. This simplifies the calculations that are described below. The second and third DO loops are used to determine the number of unique pairwise comparisons that were made and place
the result in a macro variable. Examples of possible values for this macro variable (&COMBIN) for 2, 3, 4 or 5 unique values of the dependent variable are shown in Table 1.

<table>
<thead>
<tr>
<th># of Unique &amp;COMBIN</th>
<th>Dependent Unique Pairwise Comparisons</th>
</tr>
</thead>
<tbody>
<tr>
<td>2 (A, B) A/B</td>
<td>1 (2)</td>
</tr>
<tr>
<td>3 (A, B, C) A/B, A/C, B/C</td>
<td>3 (3)</td>
</tr>
</tbody>
</table>

The second DATA step places all of the least squares means and their respective standard errors into one record for ease of processing. Although the standard errors of the least squares means are not employed in any calculations, they will be displayed with the final output.

The last data step merges the data sets created by the two DATA steps described above and calculates the variance of the difference \( \text{LSM}_1 - \text{LSM}_2 \) as

\[
\text{vdif} = \sqrt{\text{var}(1) + \text{var}(2) + 2 \cdot \text{cov}(1,2)}
\]

This DATA step also generates the critical Student's t-statistic and the corresponding two-sided p-value. Lastly, a DATA _NULL_ (not shown) is used to generate the final output.

**Figure 1**

%macro lstable(dsn);
%let bign = eval(&NDEPVAR* &NDEPVAR);
data matrix(keep= _name_ var1-var &BIGN);
set &DSN end=eof;
array var (&BIGN) var1-var &BIGN;
array cov(&NDEPVAR) cov1-cov &NDEPVAR;
retain var1-var &BIGN;
do x = 1 to &NDEPVAR;
 var[&NDEPVAR* (number-1) + x] = cov[x];
end;
if eof;
numer = 1;
do x = 2 to &NDEPVAR;
 numer = numer * x;
end;
d1 = &NDEPVAR - 2;
denom1 = 1;
do x = 2 to d1;
 denom1 = denom1 * x;
end;
combin = numer / (denom1**2);
call symput('combin',
 trim(left(put(combin,8.))));
run;
data lsmean(keep= _name_ ls1-ls&NDEPVAR sel-se &NDEPVAR);
set &DSN end=eof;
array ls (&NDEPVAR) ls1-ls &NDEPVAR;
array se (&NDEPVAR) se &NDEPVAR;
retain ls1-ls&NDEPVAR
 sel-se &NDEPVAR;
ls[number] = lsmean;
se[number] = stderr;
if eof;
run;
data anal(keep= ls1-ls &NDEPVAR pv1-pv &COMBIN);
merge lsmean
 matrix by _name_
 array var (&BIGN) var1-var &BIGN;
 array ls (&NDEPVAR) ls1-ls &NDEPVAR;
 array pvval(&COMBIN) pv1-pv &COMBIN;
do x = 1 to &NDEPVAR-1;
p1 = 1;
p2 = 0;
p3 = 1;
do y = x+1 to &NDEPVAR;
 cntr + 1;
p1 + 1;
p2 + 1;
delta = abs( ls(x) - ls(y) );
 vdif = sqrt( var[ &NDEPVAR+1]*
 (x-1)+1)+
 var[(&NDEPVAR+2)*p1+
 ((&NDEPVAR+1)*p3-1))-2*var[(&NDEPVAR+1)*
 (p3+1)+p1])
 crit = delta / vdif;
pval(cntr) = 2*1-probt( crit,
 &ERR,DF )
end;
end;
run;
%mend lstable;

The author can be contacted at:
IBRD-Rostrum Global
Gwynedd Hall, Suite 100
1777 Sentry Parkway West
Blue Bell, PA 19422 215-540-8400

SAS is a registered trademark of the SAS Institute, Inc.