Paper PH-06-2015

Plotting LSMEANS and Differences in Generalized Linear Models with GTL

Robin High, University of Nebraska Medical Center, Omaha, NE

ABSTRACT

A visual display of LsMeans and their pairwise differences in a generalized linear model is an important component of data analysis which allows one to view and compare differences Lsmeans. The SGPLOT procedure from SAS® software will produce graphs from an ANOVA for LSMeans and their differences with confidence intervals including the forest plot and the mean-mean multiple comparison (MMC) plot. The mean-mean scatter plot is available with ODS Graphics. However, greater flexibility of making a combined plot of the LSMeans and their differences with a forest plot can be achieved with the Graph Template Language (GTL). The process consists of appending two data sets, one with the file of LSMeans and the other file containing their differences produced with ODS OUTPUT statements, into one data set in block diagonal form and then submitting this data set to a template with SGRENDER. This graph can also be enhanced to include results from the "lines" option in the LSMEANS statement with PROC GLIMMIX and interpret pairwise differences sorted by the decreasing values of the LSMeans. These plots also provide an introduction to producing more complex graphs directly with graph templates including the use of macro variables for modifying graph structure and appearance and with dynamic variables for graph content. These techniques require a basic statistics background including ANOVA and experience with the SGPLOT procedures.

INTRODUCTION

Visual displays of LsMeans and their differences are important components of the output from a generalized linear model. The techniques to produce these plots are basically the same for one-way ANOVAs and the most complex models having random effects or repeated measures. Tables of means and pvalues for differences are also helpful, but a well-constructed graphical display can summarize the output in a manner that can offer insight for interpretation, especially to understand the pvalues for the model and individual comparisons. The techniques described in High (2014) are the background for this paper which continues the development of SAS code available with the Graph Template Language (GTL) to visually display means and their differences with confidence intervals in addition to the pvalues.

Visual Displays of LSMeans and their Differences in a Generalized Linear Model

Assume that continuous response data are classified into two or more groups of varying sample sizes and the ideal conditions of normality, equal variances, and independence are reasonably satisfied (these ideal conditions can be relaxed with PROC GLIMMIX). In the example data shown in the example, both unequal group variances and unequal sample sizes are present. Response data assuming normal residuals are usually analyzed with an ANOVA using a SAS/STAT procedure such as PROCs GLM, MIXED, or GLIMMIX.

Means with their confidence intervals are often all that is displayed in a graph, perhaps with annotations or explanations to state which pairs of means are significantly different. However, an effective graphical display, which can be produced for analyses ranging from a two-sample, equal variance t-test to a repeated measures ANOVA with interactions and a complex covariance structure, is to visually show a plot of both the LsMeans and differences of selected or all possible pairs of means.

The layout of the graph produced with the GTL code described here is shown in Figure 1. The top panel displays the LsMeans and their 95% confidence intervals with horizontal lines and the bottom panel displays the 95% confidence intervals for their pairwise differences also with horizontal lines indicating which pairs are significant or not significant (based on the relevant Type I error rate, α) with two distinct patterns: a solid line indicates the pvalue is less than α=0.05, where the difference of 0 is not included in the interval. The length of the corresponding 95% confidence intervals for the differences depends on the magnitude of the standard errors. One notable feature of this graph is the lines for both the means and their differences are horizontal. By producing the two panels in this manner, the horizontal scale for both panels can be given widths having the same range. Also, labels of the group means and especially their pairwise differences have more printing options and are easier to read on the vertical axis. Plotting either or both panels with vertical lines for the confidence intervals follows the same process, depending on the preferred orientation; however, GTL generally provide features with greater flexibility to label horizontal lines for both the means and their differences on one of the two vertical axes.

Combined Plot of LsMeans and Differences

Methods of plotting and interpreting multiple comparisons of differences in LsMeans are illustrated with data from Chapter 10 of Westfall, et. al. (2011) which relates an experiment to study the olfactory sense of smell to age treated categorically. For illustration of the graphical display of means and differences, only the first four of the five age groups are selected. The continuous response variable, smell, is analyzed with a one-way unequal variance ANOVA with one fixed factor, agegroup (coded numerically as 1, 2, 3, 4) which is assigned a format ag. having associated values of "Age 1" through "Age 4" (the recommended coding method for producing graphs with GTL). The pvalue from the overall F-test=0.005 is printed in the upper right corner of the panel for adjusted differences which indicates evidence that the means for levels of age group are different (ProbF=0.005). The range of the horizontal axis is the same for both the display of the means and their differences (with a range of 0.5), so the spacing of the means in the top panel matches the spacing of the associated differences from 0 on the bottom panel.

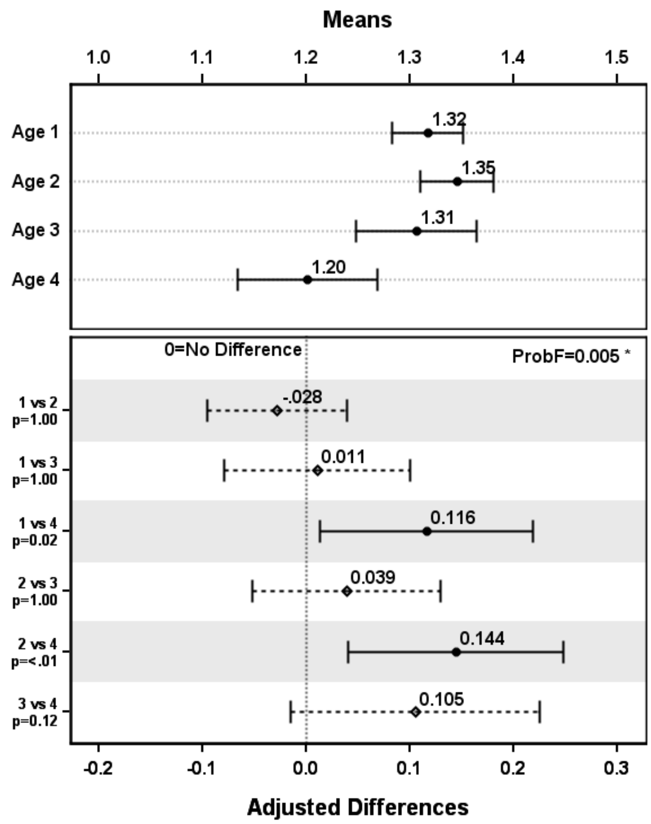

Figure 1. Plot of LsMeans and 95% confidence intervals for their differences according to factor levels.

To produce the graph in Figure 1 the data are first analyzed with PROC GLIMMIX from SAS/STAT software, V. 9.4, with the unequal variance ANOVA model. ODS OUTPUT statements save results of the LsMeans and their differences into two SAS datasets which provide the summary data necessary to produce the graph:

```
ODS OUTPUT lsmeans=lsm(KEEP = agegroup estimate lower upper)
           diffs=dfs(KEEP = agegroup _agegroup estimate adjlower adjupper adjp);

ODS SELECT nobs tests3 covparms fitstatistics;
PROC GLIMMIX DATA=upsit;
WHERE agegroup LE 4;
CLASS agegroup ;
MODEL smell = agegroup / ddfm=satterth;
RANDOM _residual_ / group=agegroup ;
LSMEANS agegroup / cl diff adjust=bonferroni;
RUN;
```

In the ODS OUTPUT statements, the KEEP= option saves both the LsMeans, their differences, and endpoints of the confidence intervals. In this example, the Bonferroni method was chosen to adjust the confidence intervals of the differences for multiple comparisons and are printed on the left vertical axis with the adjusted pvalue indicating the significance of each of the six comparisons.

Combined File with Both Lsmeans and Differences

Figure 1 displays means and confidence intervals for the LsMeans (top panel) and the confidence intervals for the pairwise differences (bottom panel) of all LsMeans computed for each factor level as part of the ANOVA summary. The ODS file of LsMean differences produced with the LSMEANS statement listed above contains all the information needed to plot the adjusted confidence intervals of the pairwise differences displayed as horizontal lines with the value of the difference as the center point of the interval. With α=0.05, significant differences are displayed when the adjusted 95% confidence intervals of the comparisons do not intersect a dashed vertical line placed at a difference equal to 0. The required modifications to the file of differences is to first rename the variable estimate to mndif1 (the same name also appears in the lsmeans file) and the adjusted confidence limits. Also, add a text variable called label which contains the two factor levels involved in the comparison of LsMeans, in this example identified by the variables agegroup and _agegroup. The plotted variable for the vertical axis to display the differences shows the contents for the comparison. The variable label made with the CAT function from the data set of differences in a DATA step. The actual names of the factor levels to display on the graph can be entered into the label with a format. The adjusted pvalue is also included by appending it with a pvalue5.2 format.

```
DATA dfs1; SET dfs(rename=(estimate=mndif1 adjlower=adjlower1 adjupper=adjupper1) );
KEEP label mndif1 adjlower1 adjupper1 mndif2 adjlower2 adjupper2;
LENGTH label $30 ptxt $4;
IF adjp < .01 then ptxt=', p'; else ptxt=', p=';
label = CAT(PUT( agegroup,ag.),' vs ',
            PUT(_agegroup,ag.),ptxt,STRIP(PUT(adjp,pvalue5.2)));

* copy data into 3 different variables for non-significant differences,
  set these 3 variables to missing;
IF adjp GE .05 then
 DO; mndif2 = mndif1; adjlower2 = adjlower1; adjupper2 =adjupper1;
     CALL MISSING(mndif1, adjlower1, adjupper1);
END;
RUN;
```

The contents of ptxt either contains p= for pvalues greater than or equal to 0.01; for smaller values, ptxt omits the equal sign and the attached format pvalue5.2 adds the less than sign, so the pvalue is printed as p <.01. The format for the integer agegroup variable is ag in the CAT function; the length of the label must be long enough to contain the contents of the longest expression; otherwise, the label will be blank.

Next, append the two data sets into one data set that is in a block diagonal structure of means and differences.

```
DATA lsmdf;
SET lsm dfs1;
RUN;
PROC PRINT DATA=lsmdf NOObs;
VAR agegroup estimate lower upper
    mndif1 adjlower1 adjupper1 label mndif2 adjlower2 adjupper2;
RUN;
```

```
Agegroup Estimate Lower Upper mndif1 adjlower1 adjupper1 label       mndif2 adjlower2 adjupper2
   1        1.32   1.28  1.35    .        .         .                    .        .         .
   2        1.35   1.31  1.38    .        .         .                    .        .         .
   3        1.31   1.25  1.36    .        .         .                    .        .         .
   4        1.20   1.13  1.27    .        .         .                    .        .         .
   .          .      .     .     .        .         .     1 vs 2,p=1.00 -0.03   -0.10      0.04
   .          .      .     .     .        .         .     1 vs 3,p=1.00  0.01   -0.08      0.10
   .          .      .     .   0.12     0.01      0.22    1 vs 4,p=0.02   .        .         .
   .          .      .     .     .        .         .     2 vs 3,p=1.00  0.04   -0.05      0.13
   .          .      .     .   0.14     0.04      0.25    2 vs 4,p<0.01   .        .         .
   .          .      .     .     .        .         .     3 vs 4,p=0.12  0.11   -0.01      0.22
```

The block structure defines how the data will appear in the two panels of the graph: the first four columns and the top four rows (upper left block) contain data to plot the lsmeans with the first LAYOUT OVERLAY section; the remaining rows and columns (lower right block) contain the data to plot the differences with the second LAYOUT OVERLAY section. The reason for two sets of columns containing the differences and confidence limits is to provide different line types; a solid line when the pvalue < 0.05 and a dashed line for the non-significant differences. Ideally, to retain the same column names for all differences and adjusted endpoints is desired. However, a discrete attribute approach (the DISCRETEATTRMAP and relevant statements in GTL) to specify line types the two types of comparisons is not possible here since it currently does not support the specification of ERRORBARATTRS=().

The graph is produced by listing the necessary ODS GRAPHICS statements with the desired options and reading the input data file with PROC SGRENDER and assigning the template name. The purpose of the DYNAMIC statement will be described in the next section.

```
ODS GRAPHICS ON / reset = all height= 6 in width= 5 in border= off;
ODS LISTING image_dpi=200 gpath="u:\sas\plots";
ODS GRAPHICS ON /reset=index imagename="anova_means_diffs";

PROC SGRENDER DATA=lsmdf TEMPLATE=anvmns;
DYNAMIC _title ="Olfactory Study" _mnttl="Response Means"
        _difttl="Adjusted Differences"
 _x='agegroup' _nlvls='4' _xtckfmt="ag."
 _ystr='1'    _yend='1.5'  _yincr='.1' _ytckfmt='4.1'
 _dstr='-.2'  _dend='.3'  _dincr='.1' _dtckfmt='5.1' ;
FORMAT agegroup ag. estimate mndif: 5.2;
RUN;
ODS GRAPHICS off;
```

Description of the GTL Code

This section briefly describes some of the statements from the graph template language (GTL) needed to produce Figure 1 (the complete GTL code to produce this graph is found in the Appendix). Referring to the template code, the template name “anvmns” appears at the end of the DEFINE STATGRAPH statement just below PROC TEMPLATE. The data set with the LsMeans and differences in block diagonal structure are then specified with PROC SGRENDER as DATA=lsmdf along with the template name TEMPLATE=anvmns as shown above.

The key feature of the figure is the two-panel display. PROC SGPANEL is designed for situations that display the same variables across multiple levels of one or two PANELBY variables. However, the graph template approach is relevant when displaying different columns of data and thus specifying that different types of graphs need to appear in the panels and also that the width and/or height of the panels may be different. The ODS Graphics Designer and SG procedures will produce template code, which is often a quick and useful starting point to produce the GTL code for many graphs.

The graph is produced with PROC SGRENDER with values specific to the graph content entered as dynamic variables within PROC SGRENDER as shown above. A DYNAMIC statement defines variables with data values, variable names, formats, and other items that can be sent directly to the template without the need to enter them in the graph template statements. All dynamic variables begin with an underscore and are also included within PROC TEMPLATE after the BeginGraph statement where only the dynamic variable names are listed on the DYNAMIC statement. They provide a convenient way to enter values which depend only on the summary data set for the ANOVA model external to the template, yet do not affect the graph structure or visual appearance. Also, within the graph template code, character values on the DYNAMIC statement do not need to be enclosed in quotes, as they have been entered within quotes in SGRENDER. For example, the ENTRYTITLE statement only needs a dynamic variable name `_title` which would otherwise be entered as “Olfactory Analysis” at the same position. The same template can thus be utilized with summary data sets from other ANOVAs where only the variables defined with the dynamic values will change.

For repetitive use of one type of multi-panel graph, as is the situation when plotting means and differences from ANOVA models, it is usually more efficient to store statements of the template with values associated with its structure and appearance that may change entered as macro variables in the template and then make edits as needed to the macro variables. Variables that define the content of the graph can be specified with dynamic variables within PROC SGRENDER. The following table lists a few of the items that can be modified with macro of dynamic variables:

Examples of graph specifications entered external to the template code			
Macro Variables		**Dynamic Variables**	
Name	Purpose	Name	Content
rwa	proportional height, top panel	_title	Overall graph title
rwb	proportional height, bottom panel	_grpttl	Panel title for LsMeans
mmrg	Lsmeans: top/bot margins	_difttl	Panel title for Differences
dfmrg	Differences: top margin	_x	Name of ANOVA factor
tksz	axis tick value size	_nlvls	Number of factor levels
dlsz	data value size	_xtckfmt	factor format (="ag.")
txtE	text entry size	_ystr	lower limit of means
mrksz	marker symbol size	_yend	upper limit of means
lblsz	axis label size	_yincr	increment for means
errbrthk	error bar thickness	_ytckfmt	numeric format (=5.1)
		_dstr	lower limit of differences
		_dend	upper limit of differences
		_dincr	increment for differences
		_dtckfmt	numeric format (=5.2)

To define the graph structure, macro variables entered into the template code allow for easy and quick edits since variations in structure and appearance may be desired. Edits to these macro variables only need to be made external to the template code. Structural options such as the panel heights and graph margins or changes to the graph's visual appearance, such as sizes for text, tick values, or data labels, markers, and other items not part of the graph content. These entries are often the same across the panels, and it may be difficult to locate and edit all of them when they are embedded in the template code.

Multiple entries of the macro variables located within the template code are then quickly processed by running it with PROC TEMPLATE. If specific types of character data specifications entered as macro variables are desired to be smaller or larger, they can be adjusted within the template code, such as modifying text size by entering %EVAL(&txtE. - 1) to make it smaller.

Plots for each panel are defined here with two sets of LAYOUT LATTICE statements. The types of plot statements in the graph template within the LAYOUT sections for each panel are essentially self-explanatory having similar names found in SGPLOT; however, some keywords and options may differ in spelling or how they are entered. The template code indicates the graph consists of two panels laid out in 2 rows and 1 column. The rowweights (which must sum to 1) indicate the proportions of the specified graph height to be assigned to each panel. The two proportions for rowweights are easy to modify and their values depend on the numbers of means and pairs of differences to appear in each panel.

```
LAYOUT lattice / rows=2 rowweights=(.4 .6) rowgutter=3 columns=1;
```

For a two-panel graph, the layout of the template code consists of two sets of paired statements which surround the plot statements specified, each defined by a LAYOUT OVERLAY statement and concludes with an ENDLAYOUT statement. Much of the editing focusses on specifications for the vertical and horizontal axes, including how to print tick values and labels. Unlike SGPLOT which requires separate XAXIS and YAXIS statements, the specified options for the lower horizontal axis (X) and left vertical axis (Y) in GTL are placed within the brackets identified by xaxisopts=() and yaxisopts=() respectively on the LAYOUT statement. GTL also provides the capability to define values on the top horizontal axis x2axisopts=(), as used here for the top panel of LSMeans and the right vertical axis, y2axisopts=(). One of the difficulties in making these graphs is to understand choice of options available for each axis in the top and bottom panels and how to apply them for they often differ from SGPLOT.

Axis Specifications for the Panel of LsMeans

For the top panel of Figure 1 which displays the LsMeans, the left vertical axis is defined as numeric (recall that factor levels are coded numerically as 1, 2, 3, .., p, where p = number of factor levels):

```
yaxisopts=(display=(line tickvalues)
           offsetmin=.05 offsetmax=.05
           griddisplay=on gridattrs=(color=grey pattern=dot thickness=1)
           linearopts=( tickvaluesequence= (start=1 end=_nlvls increment=1)
                        tickvalueformat=_xtckfmt )
           tickvalueattrs=(weight=bold size=&tksz.)
           reverse=true);
```

Numeric values for the left vertical axis with a format are entered with a *linearopts()* option, in this case with a specified layout with equally spaced values assumed to begin at 1 (conceptually, the sorted order of factor levels is printed, but specifying actual values gives you more control). The tick values are printed sequentially with the *tickvaluesequence(*start=## end=## increment=##*)* option; here three keywords specify the lower and upper boundary values and the increment, which is defined here to be 1 for the factor levels. In the example on the previous page two dynamic variables are utilized, one for the maximum tick value (_nlvls = number of factor levels) and for the factor level format (_xtckfmt), which are allowed to vary depending on the values of the data. The *reverse=true* option prints the numerically coded values with the smallest number on top, so they increase from top to bottom.

For display purposes the LsMeans and confidence intervals may be defined for either the top or bottom horizontal axis; in this case the upper axis, x2, is defined as:

```
x2axisopts=(display=(line ticks tickvalues)
            offsetmin=0.05 offsetmax=0.05
            linearopts=( tickvaluesequence=(start=_ystr end=_yend increment=_yincr)
                   TICKVALUEPRIORITY =true tickvaluefitpolicy= none tickvalueformat=_ytckfmt)
            tickvalueattrs=(color=black weight=bold size=&tksz.) )
```

Most of the entries are similar to the left vertical axis; the predefined minimum and maximum values for the range are specified along with an incremental value and are entered with dynamic values _ystr, _yend, and _yincr. Also necessary or desirable (esp. with other graph types), the *tickvaluepriority=true* option prints the endpoints of the range on the graph, which may not be the case if the actual data range of the data for the axis is smaller than the range specified.

Axis Specifications for the Panel of Differences in LsMeans

For the LsMean adjusted differences displayed in the lower panel, the vertical axis is defined as discrete which here are defined with discreteopts=( ), since the character variable label, containing the combination of factor levels and the associated pvalue, is plotted:

```
yaxisopts=(display=(line ticks tickvalues)
           offsetmin=&dfmrg. offsetmax=.075 /* reverse=true at the end flips min and max */
           discreteopts=(tickvaluefitpolicy=splitalways
                         tickvaluesplitchar= ","
                         tickvaluesplitjustify= right
                     colorbands=odd colorbandsattrs=(transparency=0.5 color=lightgray))
           tickvalueattrs=(color=black weight=bold size=&tksz.)
           reverse=true);
```

One of the primary reasons for make the graph with horizontal lines for the confidence intervals is the readability of labels. With two factor levels and a pvalue printed on the label, it is typically long enough to split its contents into two rows printed here on the left vertical axis, which works better to read them divided in that manner than if they were read from below on a horizontal axis. This feature is achieved here with the tickvaluefitpolicy=splitalways and the other specified "split" options. A comma "," was added to the label when it was produced; instead of being printed, it is now the value which defines where the split is to occur (tickvaluesplitchar=","). Also, the color bands for each difference help to differentiate them and the reverse order is chosen (reverse=true) so that the order they appear in the data step matches the order observed in the top panel for the LsMeans.

Referring back to Figure 1, the lower panel shows the forest plot for the differences in LsMeans for the example data. The values of the differences are printed at the center of each interval; the approximate values of the endpoints of the intervals can be determined with vertical lines drawn to the horizontal axis. The labels for the differences can be attached to either the left vertical axis (YAXIS) or to the right vertical axis (Y2AXIS). These comparisons are positioned with equal spacing between the horizontal lines without any visual representation of the values of the two LsMeans for each difference. Also, since the label for each difference is character data, transparent color bands surround the horizontal bands for the differences strictly to enhance visual clarity.

Although beyond the scope of this paper, it is straightforward to design the template code to plot LsMeans and differences for two variables that have a significant interaction by holding one factor constant and looking at differences of the other factor (slicing). The additions to the template in order to make this type of graph include entering a group variable with associated group options on the SCATTERPLOT statement and a DISCRETELEGEND statement for the lsmeans, and to add the variable held constant at the beginning of the labels for the differences. In this situation, the size of the graph can grow considerably large, yet the basic process to make this graph is nearly the same as for one variable. If the graph gets too large, the variable held constant can be entered on a BY statement in SGRENDER.

Enhanced Plot with the Lines Option

PROC GLIMMIX provides a "lines" option on the LSMEANS statement which outputs the LsMeans in sorted descending order with vertical lines showing which subsets of the adjacent means for which the adjusted differences are statistically equal and which are different (preferred over PROC GLM for this type of output). The ODS output file to save this information is LSMLINES=lsmlnes. This method of summarizing differences can also be incorporated into the graph to plot the lsmeans in a descending order. Also the differences should appear in the same relative order implied by the order of the lsmeans, as shown in Figure 2. In the example data, the LsMean for Age 2 is larger than the LsMean for Age 1, so it appears as the first line in the top panel. The placement of the letters A & B appear in the SAS output as vertical lines to indicate which subset of means are statistically equal and which are different. These vertical lines can also be plotted on the top panel of the graph for the LsMeans at the largest values (slightly above the largest upper confidence limit) which here are defined by the "A" "B" symbols taken directly from the lsmlines output.

Since the means are plotted in descending order in the top panel, the differences in the lower panel will always be non-negative, as the first mean in the comparison is defined to be greater than or equal to the second mean which is also greater or equal than all subsequent means for the first mean under consideration. To make this plot requires manipulation of the differences when the smaller of the two means involved in each difference appears first in order to reverse the factor levels in the comparison and also the upper and lower limits. To place the differences in the desired descending order requires that first the file of differences be sorted in the order of comparisons that the LSMeans files assumes. Once this is achieved, a number represented as a two-digit character value (i.e., xplt=PUT(nmbr,z2.), will produce 01, 02, 03, etc. is added and a format of the labels associated with these values to appear in the discreteopts=() specifications for the vertical axis. They are defined as character so that the same features of the graph described earlier are maintained, such as the ability to add colored bands and to split the label of the comparison into two lines. The code to produce the labels in this order requires some processing of the axis values with DATA steps and to make formats with PROC FORMAT. The internal character values for the labels with the comparisons are in the sorted order for the graph and do not need to be entered with a tickvaluesequence=() option.

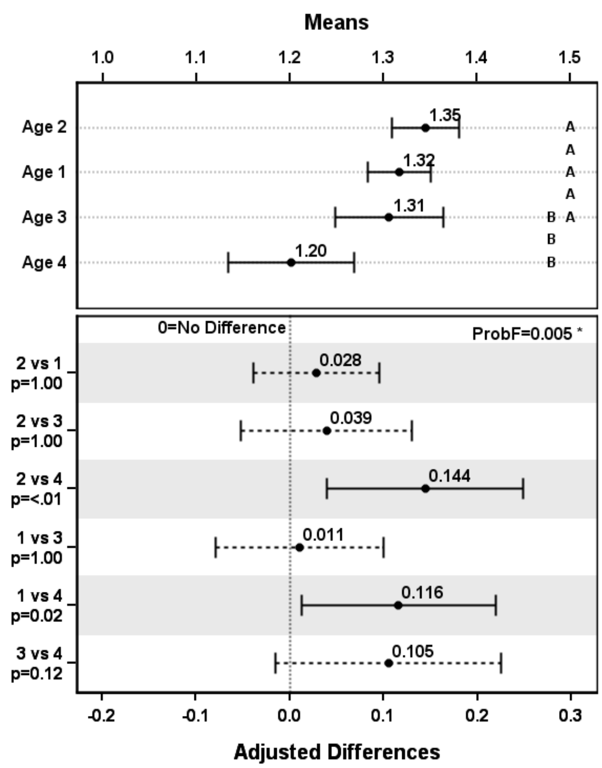

Figure 2. Plot of LsMeans and 95% confidence intervals for their differences according to the "lines" option.

Summary and Conclusions

The graphing technique described here allows interpretation of both Lsmeans and their differences and could become a more common graph to summarize differences from ANOVA. The degree of difficulty to make them varies, as the basic mean and forest plot is the easiest to produce whereas the file to make the graph illustrating interpretation from output produced by the lines option requires some data manipulations and programming steps. With GTL, a straightforward application of this code can be written to add extra columns and new variable names in block diagonal structure, such that multiple graphs can appear for more than one response variable.

REFERENCES

1. High, Robin “Plotting Differences among LSMEANS in Generalized Linear Models” Paper 1902, SGF 2014
2. Hsu, Jason, Mario Peruggia, (1994) *Graphical Representations of Tukey’s Multiple Comparison Method*, Journal of Computational and Graphical Statistics, Volume 3, No. 2, pp. 143-161.
3. Hsu, Jason (1996) *Multiple Comparisons Theory and Methods*, Chapman & Hall/CRC: Boca Raton.
4. Westfall, Peter, Randall D. Tobias, and Russell D. Wolfinger. 2011. *Multiple Comparisons and Multiple Tests Using* SAS®, Second Edition. Cary, NC, SAS Institute

CONTACT INFORMATION

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

Robin High
Statistician III
College of Public Health
Department of Biostatistics
University of Nebraska Medical Center
984375 Nebraska Medical Center
Omaha, NE 68198-4375
Phone: (402) 559-2929
email: rhigh@unmc.edu

APPENDIX

The data set upsit used in the example is available from Chapter 10 of Westfall et. al.,

```
* The SAS code to produce inputs for the graph is entered in the document
* the ANOVA assumes coding categorical factors as integers 1,2,3,.. ;
* Apply a format to attach factor level descriptions;

PROC FORMAT;
VALUE ag 1='Age 1' 2='Age 2' 3='Age 3' 4='Age 4';
RUN;

* Macro Variables;
* plot means horizontally on top panel, differences in lower panel;

%LET rwa = .4;  * proportional height, top panel;
%LET rwb = .6;  * proportional height, bottom panel;

%LET mmrg    = .2;  * lsmeans: top and bottom margins;
%LET dfmrg   = .18; * differences: top margin (bottom fixed in template);
%LET tksz    =  6;  * axis tick value size;
%LET dlsz    =  6;  * data label size;
%LET txtE    =  7;  * text entry size;
%LET mrksz   =  5;  * marker symbol size;
%LET lblsz   =  0;  * axis label size;
%LET errbrthk= 1.5; * error bar thickness;
```

```
PROC TEMPLATE;
DEFINE STATGRAPH anvmns;
BeginGraph;
DYNAMIC _title _mnttl _difttl _x _xlbl _nlvls _xtckfmt
        _ystr _yend _yincr _ytckfmt _dstr _dend _dincr _dtckfmt _pv;
ENTRYTITLE halign=center _title / pad=(bottom=2)
           textattrs=(color=black weight=bold size=11) ;
LAYOUT lattice / rows=2 rowweights=(&rwa. &rwb.) rowgutter=3  columns=1;
* Panel for LsMeans;
LAYOUT overlay /
    x2axisopts=(display=(line ticks tickvalues)
                offsetmin=0.05 offsetmax=0.05
                linearopts=( tickvaluesequence=(start=_ystr end=_yend increment=_yincr)
                       tickvaluepriority=true tickvaluefitpolicy= none tickvalueformat=_ytckfmt)
                tickvalueattrs=(color=black weight=bold size=&tksz.) )
     yaxisopts=(display=(line ticks tickvalues)
          offsetmin=&mmrg. offsetmax=&mmrg.
          griddisplay=on gridattrs=(color=grey pattern=dot thickness=1)
          linearopts=( tickvaluesequence=(start=1 end=_nlvls increment=1)
                       tickvalueformat=_xtckfmt)
          tickvalueattrs=(color=black weight=bold size=&tksz.)
          reverse=true);
   ENTRY _mnttl / location=outside valign=top textattrs=(color=black weight=bold size=&txtE.);
   SCATTERPLOT x=estimate y=_x / xaxis=x2 xerrorlower=lower xerrorupper=upper
         errorbarattrs=(color=black pattern=solid thickness=&Errbrthk.) ERRORBARCAPSHAPE=serif
         markerattrs=(color=black symbol=circlefilled size=&mrksz.)
         datalabel=estimate datalabelattrs=(color=black weight=bold size=&dlsz. );
ENDLAYOUT;

* Panel for Differences;
LAYOUT overlay /
     xaxisopts=(display=(line ticks tickvalues)
                offsetmin=0.05 offsetmax=0.05
                linearopts=( TICKVALUESEQUENCE=(start=_dstr end=_dend increment=_dincr)
                             tickvaluepriority =true tickvalueformat=_dtckfmt )
                 tickvalueattrs=(weight=bold size=&tksz. ) )
     yaxisopts=(display=(line ticks tickvalues) offsetmin=&dfmrg. offsetmax=.075
                reverse=true /* reverse=true flips offset- min and max */
                discreteopts=(TICKVALUEFITPOLICY=splitalways tickvaluesplitchar=","
                              TICKVALUESPLITJUSTIFY=right
                colorbands=odd colorbandsattrs=(color=lightgray transparency=0.5))
                tickvalueattrs=(color=black weight=bold size=&tksz.));
  ENTRY _difttl / location=outside valign=bottom textattrs=(color=black weight=bold size=&txtE.);
  ENTRY _pv / border=false location=inside autoalign=(topleft topright)
              textattrs=(weight=bold size= %eval(&txtE. -1));
  REFERENCELINE x=0 / lineattrs=(color=black thickness=1 pattern=33) datatransparency=.4
     CURVELABEL='0=No Difference'
     CURVELABELattrs=(color=black weight=bold size= %eval(&txtE.-1) )
     CURVELABELPOSITION=min;
  SCATTERPLOT y=label x=mndif1 / xerrorlower=adjlower1 xerrorupper=adjupper1
             errorbarattrs=(color=black pattern=1 thickness=&Errbrthk.) ERRORBARCAPSHAPE=serif
             markerattrs=(color=black symbol=circlefilled size=&mrksz.)
             datalabel=mndif1 datalabelattrs=(color=black weight=bold size=&dlsz.);
  SCATTERPLOT y=label x=mndif2 / xerrorlower=adjlower2 xerrorupper=adjupper2
             errorbarattrs=(color=black pattern=2 thickness=&Errbrthk.) ERRORBARCAPSHAPE=serif
             markerattrs=(color=black symbol=diamond size=&mrksz.)
             datalabel=mndif2 datalabelattrs=(color=black weight=bold size=&dlsz.);
endlayout;
endlayout;
EndGraph;
END;
RUN;
```