THE POWER TO KNOW.

An Introduction to Generalized Linear Mixed Models Using SAS PROC GLIMMIX

Phil Gibbs
Advanced Analytics Manager
SAS Technical Support
November 22, 2008
UC Riverside

THE POWER TO KNOW®

What We Will Cover Today

- What is PROC GLIMMIX and how do I get access to the procedure?
- What does the procedure do and how does it compare to PROC MIXED?
- What are the new features in PROC GLIMMIX?
- Are there any pitfalls in using PROC GLIMMIX?

THE POWER TO KNOW.

But first … Let’s talk about SAS Technical Support

Who Can contact Tech Support?

- Available to all SAS customers
- Free, Unlimited Support
- Telephone: (919)-677-8008
- Email: support@sas.com
- Web: http://support.sas.com/techsup/contact/index.html

When is Tech Support Available?

- New Problems
  - 9:00 AM - 8:00 PM ET (limited support from 6:00-8:00 PM)
- Tracked Problems
  - 9:00 AM - 5:00 PM ET
- Emergencies (system down situations)
  - 24 hour support

What sets SAS Tech Support Apart?

- Tech Support is a career at SAS
- Statistics Group – average of 12 years experience in tech support
- Talk to an actual live person

… Back to GLIMMIX

What is PROC GLIMMIX?

- PROC GLIMMIX is a procedure for fitting **G**eneralized **Li**near **Mix**ed **M**odels
- GLiM’s (or GLM’s) allow for non-normal data and random effects
- GLiM’s allow for correlation amongst responses

How can I access PROC GLIMMIX?

- SAS 9.1
  - Download add-on (Windows, Unix, Linux) from
    http://support.sas.com
    http://www.sas.com/statistics
  - Supported on a limited number of platforms and platform configurations
- SAS 9.2 (available now for most academic sites)

Distributions Supported in PROC GLIMMIX

- Discrete
  - Binary
  - Binomial
  - Poisson
  - Geometric
  - Negative Binomial
  - Multinomial (nominal and ordinal)

- Continuous
  - Beta
  - Normal
  - “Lognormal”
  - Gamma
  - Exponential
  - Inverse Gaussian
  - Shifted T

Distributions specified through DIST= (and LINK=) options on the MODEL statement

Syntax: PROC GLIMMIX vs. PROC MIXED

PROC GLIMMIX	PROC MIXED
BY	BY
CLASS	CLASS
CONTRAST	CONTRAST
EFFECT	
ESTIMATE	ESTIMATE
FREQ	
ID	ID
LSMEANS	LSMEANS
LSMESTIMATE	
MODEL	MODEL
NLOPTIONS	
OUTPUT	
PARMS	PARMS
	PRIOR
RANDOM	RANDOM
	REPEATED
WEIGHT	WEIGHT
<Programming Statements>	

Ooops!
Did we miss something?

G- and R-side Random Effects in MIXED and GLIMMIX

- MIXED uses RANDOM statement for G-side effects and REPEATED statement for R-side effects.

G-Side

R-Side

```
proc mixed;
   class treatment patient clinic;
   model y = treatment;
   random clinic;
   repeated / subject=patient type=ar(1);
run;
```

G- and R-side Random Effects in MIXED and GLIMMIX

Both types of effects are specified with the RANDOM statement in GLIMMIX

G-Side

R-Side

```
proc glimmix;
   class treatment patient clinic;
   model y = treatment;
   random clinic;
   random _residual_ / subject=patient type=ar(1);
run;
```

What are G- and R-side Random Effects?

- Remember from mixed models: Y = X*Beta + Z*Gamma + E
- G-side effects enter through Z*Gamma
- R-side effects apply to the covariance matrix on E
- G-side effects are “inside” the link function, making them easier to interpret and understand
- R-side effects are “outside” the link function and are more difficult to interpret

What PROC GLIMMIX Is Not …

- PROC GLIMMIX is NOT PROC MIXED with a DIST= and LINK= option
- PROC GLIMMIX is NOT a direct replacement for the %GLIMMIX macro
- PROC GLIMMIX has its own set of specialized options and features not found in other procedures or macros

Introductory Example: Logistic Regression with Random Effect

- Observed a binary response Y on 3 groups of patients (j= 1 to 3)
- 10 patients in each group
- Each patient could have received 1 of 3 treatments (i=1 to 3)
- Two covariates X1 and X2
- Assume patient group is a random effect

- LOG(p/(1-p)) = B0 + TRTi + B1*X1 + B2*X2 + GRPj
- GRPj ~ N(0,SIGMA**2)

Introductory Example: Simulating Data

```
data test;
   call streaminit(25345278);
   do grp=1 to 3;
      rgrp=rand('normal')*.7;
      do i=1 to 10;
         x1=rand('uniform');
         x2=rand('uniform');
         trt=ceil(rand('uniform')*3);
         logit=-2 + 2*x1 + x2 + (trt-2) + rgrp;
         p=exp(-logit)/(1+exp(-logit));
         if rand('uniform')>p then y=1; else y=0;
         output;
      end;
   end;
   drop rgrp i logit p;
run;
```

Introductory Example: The Data

Obs	grp	x1	x2	trt	y
1	1	0.03473	0.02817	1	0
2	1	0.06804	0.47722	2	0
3	1	0.30478	0.74750	2	1
4	1	0.71212	0.30261	1	0
5	1	0.35231	0.92217	2	1
6	1	0.14616	0.19462	2	0
7	1	0.29740	0.16734	2	0
8	1	0.66109	0.63280	2	0
9	1	0.88732	0.53281	2	0
10	1	0.28241	0.29755	2	0
11	2	0.57008	0.57124	2	0
12	2	0.77971	0.69519	3	1
13	2	0.73923	0.64358	1	0
14	2	0.08526	0.64158	2	0
15	2	0.61839	0.99100	1	1

Introductory Example: PROC GLIMMIX Code

```
proc glimmix data=test;
   class trt grp;
   model y=trt x1 x2;
   random int /subject=grp;
run;
```

Model Information	
Data Set	WORK.TEST
Response Variable	y
Response Distribution	Gaussian
Link Function	Identity
Variance Function	Default
Variance Matrix Blocked By	grp
Estimation Technique	Restricted Maximum Likelihood
Degrees of Freedom Method	Containment

Oops!

Introductory Example: Specifying the LINK= and DIST= Options

```
proc glimmix data=test;
   class trt grp;
   model y=trt x1 x2 / link=logit dist=binomial;
   random int / subject=grp;
run;
```

You will not get the correct model without the LINK= and DIST= options!

Introductory Example: Output

Model Information	
Data Set	WORK.TEST
Response Variable	y
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix Blocked By	grp
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

Class Level Information		
Class	Levels	Values
trt	3	1 2 3
grp	3	1 2 3

Number of Observations Read	30
Number of Observations Used	30

Dimensions	
G-side Cov. Parameters	1
Columns in X	6
Columns in Z per Subject	1
Subjects (Blocks in V)	3
Max Obs per Subject	10

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

Iteration History

Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	5	123.44818639	0.41987088	6.769E-8
1	0	4	130.00304388	0.18471388	8.274E-7
2	0	4	132.59220249	0.04430941	6.55E-8
3	0	3	133.03032197	0.00613896	4.786E-8
4	0	2	133.07142356	0.00057998	9.41E-8
5	0	1	133.07492066	0.00007304	9.381E-6
6	0	0	133.07535621	0.00000000	6.892E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

Introductory Example: Output (cont.)

Fit Statistics	
-2 Res Log Pseudo-Likelihood	133.08
Generalized Chi-Square	21.50
Gener. Chi-Square / DF	0.86

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	grp	2.7200	3.6860

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
trt	2	23	0.72	0.4955
x1	1	23	1.47	0.2376
x2	1	23	3.05	0.0939

Where are the information criteria statistics???

This model estimated using pseudo-likelihood, so no IC's available

Pseudo-likelihoods are not comparable across models

Introductory Example: Quadrature Approximation

```
proc glimmix data=test method=quad;
   class trt grp;
   model y=trt x1 x2 / link=logit dist=binomial;
   random int / subject=grp;
run;
```

METHOD=QUAD uses Quadrature to approximate the likelihood, ala PROC NLMIXED

Quadrature only works on subset of models that GLIMMIX can fit

Introductory Example: Quadrature Output

Model Information	
Data Set	WORK.TEST
Response Variable	y
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix Blocked By	grp
Estimation Technique	Maximum Likelihood
Likelihood Approximation	Gauss-Hermite Quadrature
Degrees of Freedom Method	Containment

Fit Statistics	
-2 Log Likelihood	30.00
AIC (smaller is better)	42.00
AICC (smaller is better)	45.65
BIC (smaller is better)	36.59
CAIC (smaller is better)	42.59
HQIC (smaller is better)	31.13

Fit Statistics for Conditional Distribution	
-2 log L(y | r. effects)	24.82
Pearson Chi-Square	20.48
Pearson Chi-Square / DF	0.68

Covariance Parameter Estimates			
Cov Parm	**Subject**	**Estimate**	**Standard Error**
Intercept	grp	1.3642	1.9369

Introductory Example: Adding Odds Ratios and Predicted Probabilities

```
proc glimmix data=test method=quad;
   class trt grp;
   model y=trt x1 x2 / link=logit dist=binomial or;
   random int / subject=grp;
   lsmeans trt / ilink cl;
run;
```

Odds Ratios

Predicted Probabilities

Introductory Example: Odds Ratio Output

Odds Ratio Estimates									
trt	x1	x2	_trt	_x1	_x2	Estimate	DF	95% Confidence Limits	
1	0.4409	0.5538	3	0.4409	0.5538	0.196	23	0.012	3.240
2	0.4409	0.5538	3	0.4409	0.5538	0.271	23	0.016	4.734
	1.4409	0.5538		0.4409	0.5538	10.300	23	0.120	886.851
	0.4409	1.5538		0.4409	0.5538	39.075	23	0.376	>999.999

Effects of continuous variables are assessed as one unit offsets from the mean. The AT suboption modifies the reference value and the UNIT suboption modifies the offsets.

Introductory Example: Predicted Probabilities Output

trt Least Squares Means												
trt	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper	Mean	Standard Error Mean	Lower Mean	Upper Mean
1	-1.3627	1.1792	23	-1.16	0.2597	0.05	-3.8020	1.0765	0.2038	0.1913	0.02184	0.7458
2	-1.0386	1.1037	23	-0.94	0.3565	0.05	-3.3219	1.2446	0.2614	0.2131	0.03483	0.7764
3	0.2663	1.1722	23	0.23	0.8223	0.05	-2.1586	2.6912	0.5662	0.2879	0.1035	0.9365

Statistics on Predicted Probabilities

Interpretation depends on the distribution and link function used

New Features in GLIMMIX: EFFECT Statement

- The EFFECT statement allows you to create constructed effects from sets of columns in the design matrix
- COLLECTION effects allow you to collect one or more columns and create a single effect for testing and inference with multiple df
- MULTIMEMBER effects allow for effects with possibly more than one nonzero column for an observation
- SPLINE effects
- POLYNOMIAL effects for multivariate polynomials

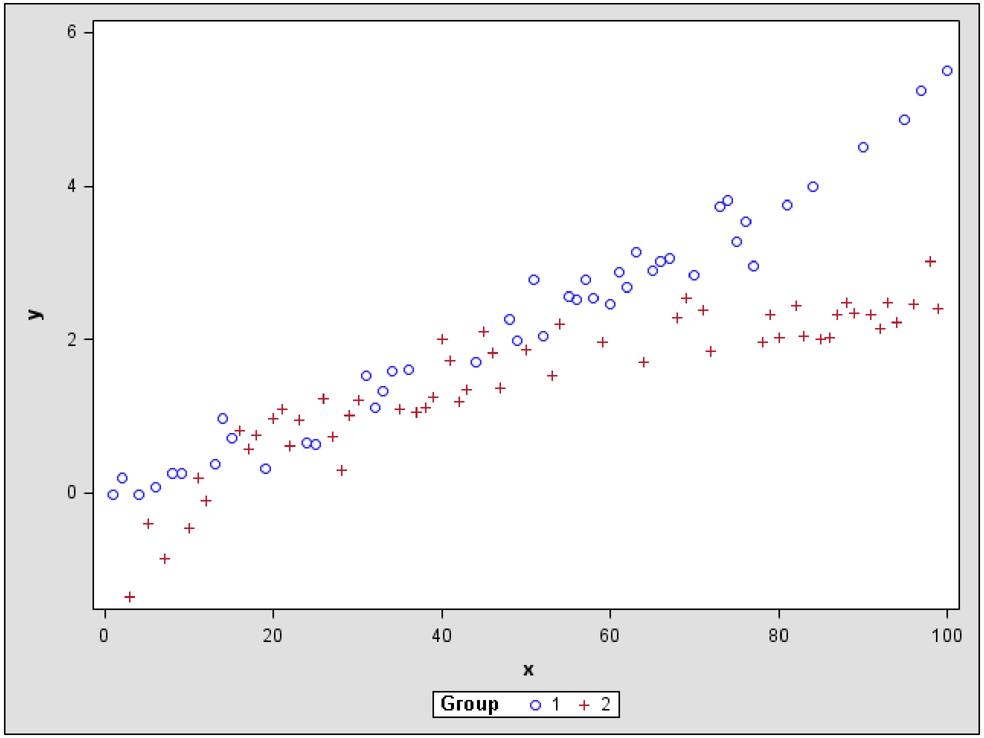
Example 38.15: Creating Spline Effects

```
data spline;
     input group y @@;
     x = _n_;
     datalines;
    1     -.020 1     0.199 2     -1.36 1     -.026
    2     -.397 1     0.065 2     -.861 1     0.251
    1     0.253 2     -.460 2     0.195 2     -.108
    1     0.379 1     0.971 1     0.712 2     0.811
    2     0.574 2     0.755 1     0.316 2     0.961
    2     1.088 2     0.607 2     0.959 1     0.653
    1     0.629 2     1.237 2     0.734 2     0.299
    2     1.002 2     1.201 1     1.520 1     1.105
    1     1.329 1     1.580 2     1.098 1     1.613
    2     1.052 2     1.108 2     1.257 2     2.005
    2     1.726 2     1.179 2     1.338 1     1.707
    2     2.105 2     1.828 2     1.368 1     2.252
    1     1.984 2     1.867 1     2.771 1     2.052
    2     1.522 2     2.200 1     2.562 1     2.517
    1     2.769 1     2.534 2     1.969 1     2.460
    1     2.873 1     2.678 1     3.135 2     1.705
    1     2.893 1     3.023 1     3.050 2     2.273
    2     2.549 1     2.836 2     2.375 2     1.841
    1     3.727 1     3.806 1     3.269 1     3.533
    1     2.948 2     1.954 2     2.326 2     2.017
    1     3.744 2     2.431 2     2.040 1     3.995
    2     1.996 2     2.028 2     2.321 2     2.479
    2     2.337 1     4.516 2     2.326 2     2.144
    2     2.474 2     2.221 1     4.867 2     2.453
    1     5.253 2     3.024 2     2.403 1     5.498
  ;
```

Two groups of data measured on X and Y

Example 38.15: Plotting the Data

```
ods html;
proc sgplot data=spline;
   scatter y=y x=x / group=group name="data";
   keylegend "data"  / title="Group";
run;
ods html close;
```



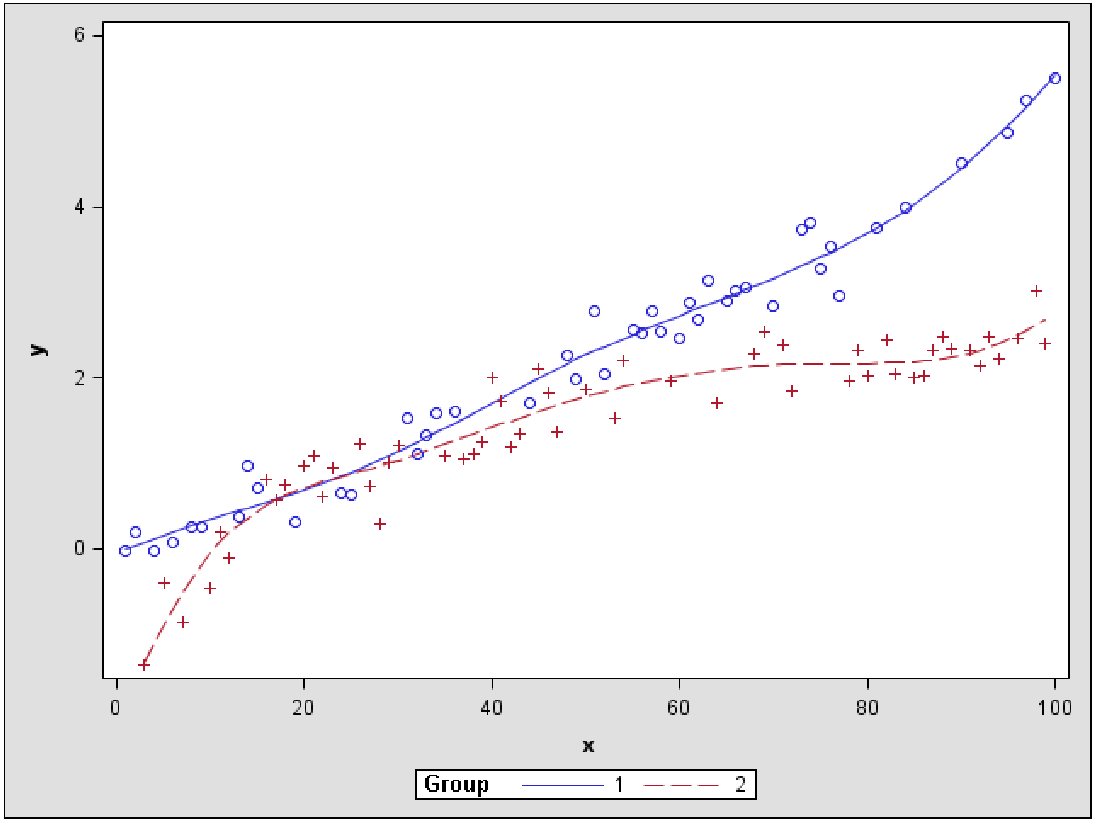
Example 39.15: Fitting the Spline Model

EFFECT statement fits b-spline of degree 3 with 7 knot points

```
proc glimmix data=spline outdesign=x;
   class group;
   effect spl = spline(x);
   model y = group spl*group / s noint;
   output out=gmxout pred=p;
run;
```

Example 38.15: Seeing the Fit

```
proc sgplot data=gmxout;
   series  y=p x=x / group=group name="fit";
   scatter y=y x=x / group=group;
   keylegend "fit" / title="Group";
run;
```



EFFECT Statement: Polynomial Effects

- Polynomial effects provide a programatic way to express polynomial fits in a model
- model y = x1 x2 x3 x1*x1 x1*x2 x1*x3 x2*x2 x2*x3 x3*x3;
- effect MyPoly = polynomial(x1-x3/degree=2); model y = MyPoly;

New Features in GLIMMIX: LSMEANS Statement Options

- SLICE= gives tests of simple effects
- Assume a model where A has 4 levels and B has 3 levels

```
proc glimmix data=test;
    class a b;
    model y=a b a*b;
    lsmeans a*b / slice=a;
run;
```

SLICE= will give tests for differences among the levels of B for each level of A

LSMEANS Statement: SLICE= Option Results

Tests of Effect Slices for a*b Sliced By a				
a	Num DF	Den DF	F Value	Pr > F
1	2	48	2.43	0.0989
2	2	48	81.61	<.0001
3	2	48	49.38	<.0001
4	2	48	79.21	<.0001

LSMEANS Statement Options: SLICEDIFF=

- Use SLICEDIFF= to explore the differences in the levels of one effect inside the levels of another effect

```
proc glimmix data=test;
   class a b;
   model y=a b a*b;
   lsmeans a*b / slicediff=a;
run;
```

LSMEANS Statement: SLICEDIFF= Option Results

Simple Effect Comparisons of a*b Least Squares Means By a							
Simple Effect Level	b	_b	Estimate	Standard Error	DF	t Value	Pr > |t|
a 1	1	2	-0.7680	0.5984	48	-1.28	0.2055
a 1	1	3	-1.3125	0.5984	48	-2.19	0.0332
a 1	2	3	-0.5445	0.5984	48	-0.91	0.3674
a 2	1	2	-3.5416	0.5984	48	-5.92	<.0001
a 2	1	3	-7.6379	0.5984	48	-12.76	<.0001
a 2	2	3	-4.0964	0.5984	48	-6.85	<.0001
a 3	1	2	-3.4324	0.5984	48	-5.74	<.0001
a 3	1	3	-5.9214	0.5984	48	-9.90	<.0001
a 3	2	3	-2.4890	0.5984	48	-4.16	0.0001
a 4	1	2	-3.4859	0.5984	48	-5.83	<.0001
a 4	1	3	-7.5245	0.5984	48	-12.58	<.0001
a 4	2	3	-4.0387	0.5984	48	-6.75	<.0001

Within each level of A we get pairwise comparisons of the levels of B

Use the PDIFF= option to get multiplicity adjustments within each level of A

New Features in GLIMMIX: LSMESTIMATE Statement

- Allows ESTIMATES that involve coefficients on the LSMEANS rather than on the parameter estimates
- Can dramatically shorten the length and complexity of an ESTIMATE statement

LSMESTIMATE Statement Syntax

```
proc glimmix data=test;
   class a b;
   model y=a b a*b;
   estimate 'ab12 vs ab21' a 1 -1 b -1 1 a*b 0 1 0 -1;


run;
```

[coefficient level_of_effect_A   level_of_effect_B]

New Features in GLIMMIX: Multiplicity Adjustments in ESTIMATE Statement

- Multiple DF contrasts have been allowed before
- Now the ESTIMATE statement can accept multiple tests within the same statement
- This family of tests can be adjusted for multiplicity

Multiplicity Adjustments on an ESTIMATE Statement

```
proc glimmix data=test method=quad;
   class trt grp;
   model y=trt x1 x2 / link=logit dist=binomial or;
   random int / subject=grp;
   lsmeans trt / ilink cl;
   estimate '1 vs 2' trt 1 -1,
            '1 vs 3' trt 1 0 -1 / adjust=bon;
run;
```

Estimates Adjustment for Multiplicity: Bonferroni						
Label	**Estimate**	**Standard Error**	**DF**	**t Value**	**Pr > |t|**	**Adj P**
1 vs 2	-0.3241	1.3518	23	-0.24	0.8126	1.0000
1 vs 3	-1.6290	1.3558	23	-1.20	0.2418	0.4836

Multiplicity Adjustments on an LSMESTIMATE Statement

```
proc glimmix data=test method=quad;
   class trt grp;
   model y=trt x1 x2 / link=logit dist=binomial or;
   random int / subject=grp;
   lsmeans trt / ilink cl;
   lsmestimate trt '1 vs 2' 1 -1,
                   '1 vs 3' 1 0 -1 / ftest adjust=bon;
run;
```

Least Squares Means Estimates
Adjustment for Multiplicity: Bonferroni

Effect	Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P
trt	1 vs 2	-0.3241	1.3518	23	-0.24	0.8126	1.0000
trt	1 vs 3	-1.6290	1.3558	23	-1.20	0.2418	0.4836

Least Squares Means Ftest

Effect	Num DF	Den DF	F Value	Pr > F
trt	2	23	0.79	0.4637

New Features in PROC GLIMMIX: ODS Graphics

- DIFFOGRAM from LSMEANS statement
- Interaction plots from LSMEANS statement
- Analysis of means plots from LSMEANS statement
- (Residual and Box Plots)

LSMEANS Diffogram Plot

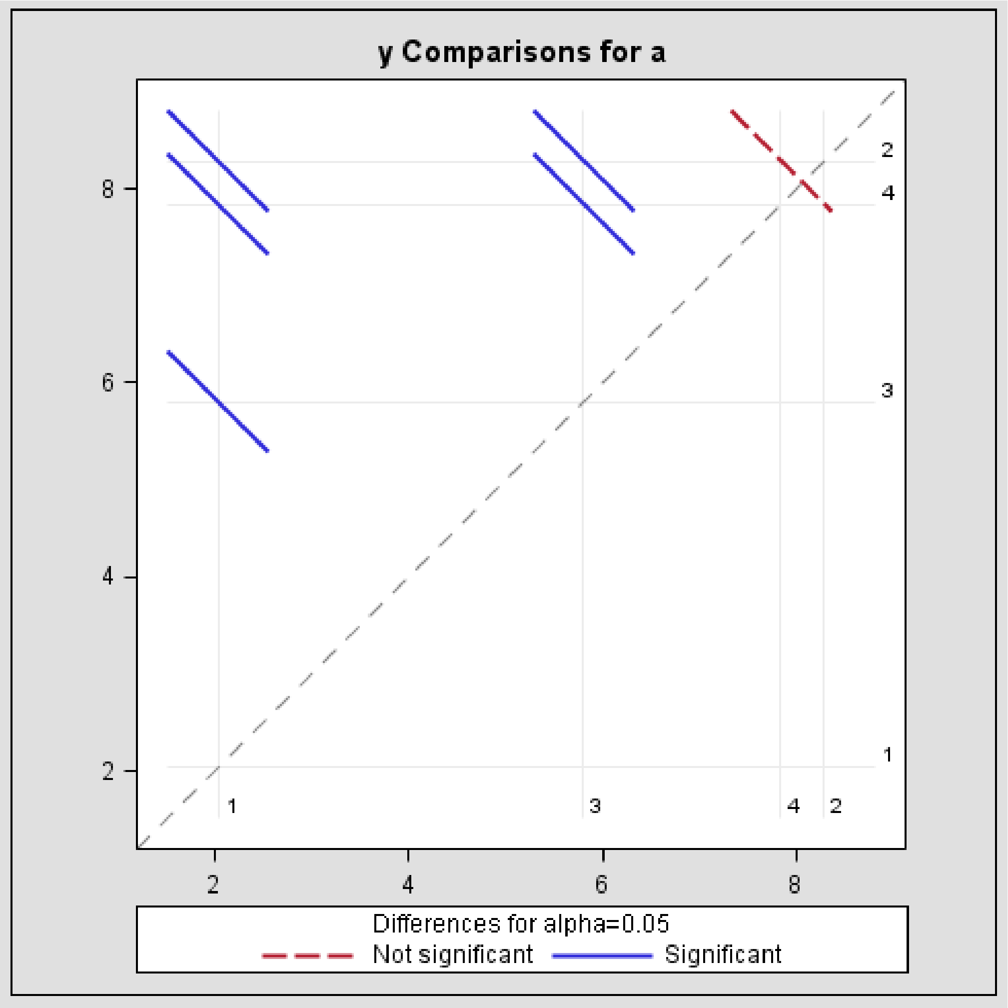
```
proc glimmix data=test plots=diffogram;
   class a b;
   model y=a b;
   lsmeans a / pdiff cl;
run;
```

LSMEANS Statement Output

a Least Squares Means								
a	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	2.0330	0.3687	54	5.51	<.0001	0.05	1.2938	2.7722
2	8.2873	0.3687	54	22.48	<.0001	0.05	7.5481	9.0265
3	5.8051	0.3687	54	15.74	<.0001	0.05	5.0659	6.5443
4	7.8383	0.3687	54	21.26	<.0001	0.05	7.0991	8.5775

Differences of a Least Squares Means									
a	_a	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1	2	-6.2543	0.5214	54	-11.99	<.0001	0.05	-7.2997	-5.2089
1	3	-3.7721	0.5214	54	-7.23	<.0001	0.05	-4.8175	-2.7266
1	4	-5.8053	0.5214	54	-11.13	<.0001	0.05	-6.8507	-4.7599
2	3	2.4822	0.5214	54	4.76	<.0001	0.05	1.4368	3.5277
2	4	0.4490	0.5214	54	0.86	0.3930	0.05	-0.5964	1.4944
3	4	-2.0332	0.5214	54	-3.90	0.0003	0.05	-3.0786	-0.9878

DIFFOGRAM Plot



Pitfalls in Working with PROC GLIMMIX

- Simplify, **Simplify**, **Simplify!**!!
- Just because you can syntactically estimate a model does not mean you will get results – or that you should even try to
- Check your data for sufficient variability before estimating a model
- NLOPTIONS TECH=NRRIDG for discrete responses
- Always specify DIST= and LINK= on MODEL statement

sas

THE POWER TO KNOW