Paper 3644-2015

Using and Understanding LSMEANS and LSMESTIMATE

David J. Pasta, ICON Clinical Research, San Francisco, CA

ABSTRACT

The concept of least squares means, or population marginal means, seems to confuse a lot of people. We explore least squares means as implemented by the LSMEANS statement in SAS®, beginning with the basics. Particular emphasis is paid to the effect of alternative parameterizations (for example, whether binary variables are in the CLASS statement) and the effect of the OBSMARGINS option. We use examples to show how to mimic LSMEANS using ESTIMATE statements and the advantages of the relatively new LSMESTIMATE statement. The basics of estimability are discussed, including how to get around the dreaded "non-estimable" messages. Emphasis is put on using the STORE statement and PROC PLM to test hypotheses without having to redo all the model calculations. This material is appropriate for all levels of SAS experience, but some familiarity with linear models is assumed.

INTRODUCTION

In a linear model, some of the predictors may be continuous and some may be discrete. A continuous predictor is one for which the numeric values are treated as meaningful and the estimated coefficient is interpreted as the effect of a one-unit change. A discrete (or categorical) predictor is one which is included in the CLASS statement. The individual values are not assumed to have any particular relationship to each other: they are treated as just "names" for the categories and are not to be interpreted quantitatively even if they are numbers. What is important for our purposes is that we want to estimate the effect of each value separately and not to assume specific spacing between values.

In addition, continuous variables can be grouped into categories and converted into discrete variables. This issue is discussed at length in Pasta (2009), but it is worthwhile to summarize a key point made there. Treating an ordinal variable as continuous allows you to estimate the linear component of the relationship, as recommended by Moses et al. (1984). On the other hand, treating an ordinal variable as discrete allows you to capture much more complicated relationships. It seems worthwhile to consider both aspects of the variable.

For categorical variables, it is possible to calculate least squares means, also known as population marginal means or adjusted means. These can be thought of as the means for a hypothetical population with a certain distribution of the predictor variables. In the simplest case, with a single categorical predictor, the least squares means are simply the observed sample means for the categories. In a model with a several continuous predictors along with a single categorical predictor, the least squares means are the predicted values for each category under the assumption that the continuous variables are set at fixed values (usually the overall mean of the continuous variable).

In more complicated situations with multiple categorical predictors and especially with interactions among categorical predictors, the least squares means can get complicated. Fortunately, SAS provides some convenient tools for understanding how the least squares means are calculated and some useful ways to work with the least squares means.

PARAMETERIZATIONS

Before getting into depth about models that include discrete variables, it is necessary to have some understanding of the way models are parameterized in SAS. This material is covered in numerous places, including several of my papers from previous conferences (Pritchard and Pasta 2004; Pasta 2005; Pasta 2009; Pasta 2010). One parameterization for discrete variables is the "less than full rank" approach in which dummy variables (indicator variables) are created for each category. This parameterization, also called the GLM parameterization, includes all the dummy variables but recognizes that there are redundancies and uses appropriate computational methods such as generalized inverses to obtain parameter estimates. The last category (as ordered using the formatted value) ends up as the reference category. To change the reference category it is necessary to reorder the categories of the variable.

It is now possible to specify the parameterization you want to use on the CLASS statement (but be aware that which procedures support this approach depends on which version of SAS you are using). You can specify REFERENCE coding, which allows you to specify a reference category which is omitted from the design matrix in various convenient ways. Alternatively you can specify EFFECT coding, which effectively compares each category to the overall average rather than to a single category, although there is still an omitted category that you can specify. My experience is that people find EFFECT coding rather confusing at first, so I recommend the use of REFERENCE coding. Note that LOGISTIC now uses EFFECT coding by default. You can specify different coding for different

variables and different reference categories (the default is LAST), making it much easier to manipulate the parameterization of discrete variables.

The examples presented here use GLM parameterization but the principles are all the same.

LEAST SQUARES MEANS – SOME SIMPLE EXAMPLES

Perhaps the simplest example of LSMEANS comes with a single discrete variable. Here's an example (with simulated data).

```
proc glm data=anal;
class site;
model y4 = site / solution;
lsmeans site / stderr pdiff;
lsmeans site / stderr pdiff OM;
title3 "y4 = site with and without OM";
run;
```

Here we have a study with multiple sites and we want to understand how the response variable, Y4, varies across site. We put SITE in the CLASS statement and as the only variable on the right hand side of the model statement. The least squares fit for this linear model is to assign the sample mean to each site. The SOLUTION shows us the estimates for the parameters and the LSMEANS provides the least squares means. The default parameterization, the GLM parameterization, creates a dummy variable for each of the 5 sites but one of the parameters is redundant (the intercept is equal to the sum of the dummy variables for the 5 sites). Therefore the last site is arbitrarily treated as the reference and gets a parameter estimate of 0; the parameters for the other sites are relative to that site. The LSMEANS are easier to understand and, in this case, the least squares means are simply equal to the sample means. The OBSMARGINS or OM option has no effect in this simple example.

The STDERR provides the standard error of each LSMEAN and a test of whether that particular LSMEAN is different from zero. This test may or may not be of any interest. The PDIFF option asks for the *P* value testing whether each possible pairwise difference is statistically significantly different from zero. This is frequently of interest. It should be noted that various methods for adjusting for multiple comparisons are available on the LSMEANS, including the TUKEY method. Here are parts of the output.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
site	4	16214.20560	4053.55140	1.42	0.2269

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		184.9503361	B	7.49456720	24.68	<.0001
site	1	-10.3378153	B	7.92951397	-1.30	0.1926
site	2	-15.9374299	B	7.94703438	-2.01	0.0452
site	3	-13.3317596	B	10.45067465	-1.28	0.2024
site	4	-17.4115336	B	10.27414782	-1.69	0.0904
site	5	0.0000000	B	.	.	.

```
The X'X matrix has been found to be singular, and a generalized inverse was used to solve the
normal equations.  Terms whose estimates are followed by the letter 'B' are not uniquely
estimable.
```

There is no overall site effect, although the estimate for site 2 is (barely) significantly different from zero. What is that testing? It is actually testing the difference between site 2 and site 5, which is why site 5 is referred to as the "reference site." The message at the bottom, which will not be shown again, appears essentially whenever you have discrete variables in your model.

site	y4 LSMEAN	Standard Error	Pr > |t|	LSMEAN Number
1	174.612521	2.590107	<.0001	1
2	169.012906	2.643259	<.0001	2
3	171.618576	7.283410	<.0001	3
4	167.538802	7.027772	<.0001	4
5	184.950336	7.494567	<.0001	5

With or Without OBSMARGINS

The lsmeans are presented along with their standard errors and a test of whether they are different from zero, which is of no interest here. But the comparison of each site against each other site is of interest. Note that the *P* values for the comparisons to site 5 (last row and last column) are the same as the *P* values from the SOLUTION above.

Least Squares Means for effect site Pr > |t| for H0: LSMean(i)=LSMean(j) Dependent Variable: y4					
i/j	1	2	3	4	5
1		0.1306	0.6986	0.3452	0.1926
2	0.1306		0.7367	0.8444	0.0452
3	0.6986	0.7367		0.6870	0.2024
4	0.3452	0.8444	0.6870		0.0904
5	0.1926	0.0452	0.2024	0.0904	

Now consider what happens when we add another discrete variable, SEX. In this case SEX is a character variable that takes on two values, FEMALE and MALE. The reference category will be MALE, the last value when sorted alphabetically by the formatted value.

```
proc glm data=anal;
class site sex;
model y4 = site sex / solution;
lsmeans site sex / stderr pdiff;
lsmeans site sex / stderr pdiff OM;
title3 "y4 = site sex with and without OM";
run;
```

Source	DF	Type III SS	Mean Square	F Value	Pr > F
site	4	17550.2748	4387.5687	1.61	0.1695
sex	1	141518.9367	141518.9367	51.93	<.0001

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		192.4432989	B	7.38339614	26.06	<.0001
site	1	-9.1199895	B	7.73588713	-1.18	0.2387
site	2	-14.9797927	B	7.75226862	-1.93	0.0536
site	3	-14.6915936	B	10.19479831	-1.44	0.1499
site	4	-18.7550993	B	10.02261086	-1.87	0.0616
site	5	0.0000000	B	.	.	.

Parameter		Estimate		Standard Error	t Value	Pr > |t|
sex	Female	-25.4760735	B	3.53522261	-7.21	<.0001
sex	Male	0.0000000	B	.	.	.

site	y4 LSMEAN	Standard Error	Pr > |t|	LSMEAN Number
1	170.585273	2.587332	<.0001	1
2	164.725469	2.645858	<.0001	2
3	165.013668	7.162746	<.0001	3
4	160.950163	6.915234	<.0001	4
5	179.705262	7.345963	<.0001	5

Without OBSMARGINS

site	y4 LSMEAN	Standard Error	Pr > |t|	LSMEAN Number
1	175.069062	2.527052	<.0001	1
2	169.209258	2.578243	<.0001	2
3	169.497457	7.109960	<.0001	3
4	165.433952	6.860748	<.0001	4
5	184.189051	7.310580	<.0001	5

With OBSMARGINS

This is a little more interesting. We have a big SEX effect and now the OM option makes a difference. What is going on? We can see more by asking SAS to tell us how it is calculating the least squares means with the E option on the LSMEANS statement. That option requests the coefficients the LSMEANS statement is using to calculate the least squares means. Now we can see that without the OM option the site effects are assuming that the sexes are exactly balanced (half and half). With the OM option, the sexes are assumed to be in the same proportion in each site as they are in the overall sample.

Coefficients for site Least Square Means						
		site Level				
Effect		1	2	3	4	5
Intercept		1	1	1	1	1
site	1	1	0	0	0	0
site	2	0	1	0	0	0
site	3	0	0	1	0	0
site	4	0	0	0	1	0
site	5	0	0	0	0	1
sex	Female	0.5	0.5	0.5	0.5	0.5
sex	Male	0.5	0.5	0.5	0.5	0.5

Without OBSMARGINS

Coefficients for site Least Square Means						
		site Level				
Effect		1	2	3	4	5
Intercept		1	1	1	1	1
site	1	1	0	0	0	0
site	2	0	1	0	0	0
site	3	0	0	1	0	0
site	4	0	0	0	1	0
site	5	0	0	0	0	1
sex	Female	0.324	0.324	0.324	0.324	0.324
sex	Male	0.676	0.676	0.676	0.676	0.676

With OBSMARGINS

So which are the correct least squares means for SITE – those with OM or those without OM? Both; they are equally valid. The tests for differences among the least squares means are exactly the same – the differences, the standard errors, and the *P* values are all exactly the same. They are not affected by the assumption about how the sexes are distributed within the site in this model with only main effects. Whether you're interested in the hypothetical "what would be the average level of Y4 at each site if there were half males and half females at each site" or "what would be the average level of Y4 at each site if there were 67.6% males and 32.4% females at each site" is entirely up to you. In practice, I generally find that I prefer the OBSMARGINS version.

This example suggests one of the reasons I tend to prefer OMSMARGINS to the default of perfect balance. What if my study were about breast cancer, for which only about 1% of the patients are male? Then calculating a value assuming about half of the patients are male and half female would be very unrealistic. It would be better to use the observed proportion as a better estimate of the population distribution.

The OBSMARGINS option allows you to specify a dataset to use to calculate the distribution of the discrete variables. By default, it is the same as the data being analyzed and that is normally the desired choice. But if you have a different dataset that you want to use as your standard – maybe a population to which you plan to apply the model – you can specify that dataset.

Just to drive home how lsmeans are calculated, here are the coefficients for sex without and with OMBSMARGINS.

Coefficients for sex Least Square Means			
		sex Level	
Effect		Female	Male
Intercept		1	1
site	1	0.2	0.2
site	2	0.2	0.2
site	3	0.2	0.2
site	4	0.2	0.2
site	5	0.2	0.2
sex	Female	1	0
sex	Male	0	1

Without OBSMARGINS

Coefficients for sex Least Square Means			
		sex Level	
Effect		Female	Male
Intercept		1	1
site	1	0.427	0.427
site	2	0.41	0.41
site	3	0.054	0.054
site	4	0.058	0.058
site	5	0.051	0.051
sex	Female	1	0
sex	Male	0	1

With OBSMARGINS

Clearly the very unbalanced distribution of subjects across the sites – two sites account for over 80% of the subjects – mean the OBSMARGINS has the potential to make a substantial difference. For this data and model, in fact it makes almost no difference at all.

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	155.457930	3.585676	<.0001	<.0001
Male	180.934004	2.758138	<.0001	

Without OBSMARGINS

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	155.050133	2.904531	<.0001	<.0001
Male	180.526207	2.009247	<.0001	

With OBSMARGINS

WORKING WITH BINARY VARIABLES

The SEX variable is coded as 1 and 2, with 1 for FEMALE and 2 for MALE. I know that and now you know that, but the output from GLM did not tell you that. The values of the variable could just as well have been 1 and 317 or 61 and -3.1416; only the formatted values matter in this context. My recommendation when working with binary variables – variables that take on exactly two values – is to code them 0 and 1 and give the name that represents the 1. That is easier to talk about and you won't have to guess the direction of the effect. Here's what we get when we substitute the variable MALE for SEX. The output is identical with only the names changed. The parameterization is the same because MALE=1 is the last category and therefore becomes the reference category.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
site	4	17550.2748	4387.5687	1.61	0.1695
male	1	141518.9367	141518.9367	51.93	<.0001

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		192.4432989	B	7.38339614	26.06	<.0001
site	1	-9.1199895	B	7.73588713	-1.18	0.2387
site	2	-14.9797927	B	7.75226862	-1.93	0.0536
site	3	-14.6915936	B	10.19479831	-1.44	0.1499
site	4	-18.7550993	B	10.02261086	-1.87	0.0616
site	5	0.0000000	B	.	.	.
male	0	-25.4760735	B	3.53522261	-7.21	<.0001
male	1	0.0000000	B	.	.	.

As you probably know, a binary variable can be treated as continuous – it does not need to be in the CLASS statement. In fact you may have been told it makes no difference whether it is in the CLASS statement or not. That is only partially true. It is true that the model is equivalent whether or not binary variables are in the CLASS statement, but it does make a difference in the parameterization and therefore the interpretation of the results. It is especially important to remember whether binary variables are continuous or discrete when interpreting least squares means (LSMEANS). Generally, my recommendation is to treat binary variables as discrete (include them in the CLASS) statement, because then you can look at their effects in LSMEANS statements, but sometimes is it better to treat them as continuous. Suppose we go back to the SEX variable and omit it from the CLASS statement.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
site	4	17550.2748	4387.5687	1.61	0.1695
sex	1	141518.9367	141518.9367	51.93	<.0001

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		141.4911518	B	9.47641500	14.93	<.0001
site	1	-9.1199895	B	7.73588713	-1.18	0.2387
site	2	-14.9797927	B	7.75226862	-1.93	0.0536
site	3	-14.6915936	B	10.19479831	-1.44	0.1499
site	4	-18.7550993	B	10.02261086	-1.87	0.0616
site	5	0.0000000	B	.	.	.
sex		25.4760735		3.53522261	7.21	<.0001

We would get the identical model if we used MALE instead of SEX except for the Intercept. The only difference is that with MALE (coded 0/1) instead of SEX (coded 1/2), the Intercept needs to be 25.4760735 larger to give the same model. What happens to the LSMEANS when the binary variable is omitted from the CLASS statement? The first thing to note is that we cannot get LSMEANS for the binary variable (only variables in the CLASS statement can be included on the LSMEANS statement). Here is what we get for the LSMEANS for SITE.

site	y4 LSMEAN	Standard Error	Pr > |t|	LSMEAN Number
1	175.069062	2.527052	<.0001	1
2	169.209258	2.578243	<.0001	2
3	169.497457	7.109960	<.0001	3
4	165.433952	6.860748	<.0001	4
5	184.189051	7.310580	<.0001	5

Without OBSMARGINS

Wait, those are exactly the same values as we got before when we specified OBSMARGINS. And it turns out if we specify OBSMARGINS we get the same answer. Is SAS forcing OBSMARGINS on us? In a sense, yes. The LSMEANS statement by default puts all the continuous variables at their overall mean. For the binary variable not in the CLASS statement, this is equivalent to the weighting you would get with OBSMARGINS.

In a more complicated model with more than one variable in the CLASS statement, the LSMEANS could vary depending on whether OBSMARGINS was specified or not. The point here is the binary variables that are omitted from the CLASS statement are set to their mean value for purposes of calculating least squares means.

DUMMY VARIABLE CODING

When working with discrete variables in procedures that do not support the CLASS statement, you have probably been taught to create a series of dummy (indicator) variables. You might accomplish this with code such as the following for the SITE variable:

```
site01=(site eq 1);
site02=(site eq 2);
site03=(site eq 3);
site04=(site eq 4);
site05=(site eq 5);
```

Actually better code would be something along the following lines:

```
site01=0;
site02=0;
site03=0;
site04=0;
site05=0;
if (site eq 1) then site01=1;
else if (site eq 2) then site02=1;
else if (site eq 3) then site03=1;
else if (site eq 4) then site04=1;
else if (site eq 5) then site05=1;
else error “error site missing or invalid”;
```

Let’s see what happens when we put the SITExx dummy variables in the model and in CLASS.

```
proc glm data=anal;
class site01-site05 sex;
model y4 = site01-site04 sex / solution;
lsmeans site01-site04 sex / stderr pdiff E;
lsmeans site01-site04 sex / stderr pdiff E OM;
title3 "y4 = site01-site04 sex with and without OM";
run;
```

Here we have specified the first 4 SITExx dummy variables, as one of them is redundant and can be omitted. This makes site 5 the reference category. We get an equivalent model to the one with SITE in the CLASS statement, but the coefficient for each SITExx variable has changed sign because it represents the 0 value for the site rather than the 1 value. Also, the estimated INTERCEPT is quite different. However, if you do the arithmetic, the estimated value for males and females for each site is the same as for the previous model.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
site01	1	3787.4903	3787.4903	1.39	0.2387
site02	1	10175.0624	10175.0624	3.73	0.0536
site03	1	5659.3120	5659.3120	2.08	0.1499
site04	1	9542.4546	9542.4546	3.50	0.0616
sex	1	141518.9367	141518.9367	51.93	<.0001

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		134.8968237	B	24.33747910	5.54	<.0001
site01	0	9.1199895	B	7.73588713	1.18	0.2387
site01	1	0.0000000	B	.	.	.
site02	0	14.9797927	B	7.75226862	1.93	0.0536
site02	1	0.0000000	B	.	.	.
site03	0	14.6915936	B	10.19479831	1.44	0.1499
site03	1	0.0000000	B	.	.	.
site04	0	18.7550993	B	10.02261086	1.87	0.0616
site04	1	0.0000000	B	.	.	.
sex	Female	-25.4760735	B	3.53522261	-7.21	<.0001
sex	Male	0.0000000	B	.	.	.

What will the lsmeans look like with and without the OBSMARGINS option? Here are the results for SITE01.

site01	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
0	155.492019	6.332701	<.0001	0.2387
1	146.372030	12.373097	<.0001	

Without OBSMARGINS

Those lsmeans do not seem to bear much relationship to what we saw before. What is SAS doing? Here are the coefficients.

Coefficients for site01 Least Square Means			
		site01 Level	
Effect		0	1
Intercept		1	1
site01	0	1	0
site01	1	0	1
site02	0	0.5	0.5
site02	1	0.5	0.5
site03	0	0.5	0.5
site03	1	0.5	0.5
site04	0	0.5	0.5
site04	1	0.5	0.5
sex	Female	0.5	0.5
sex	Male	0.5	0.5

Without OBSMARGINS

So we are acting as though half the subjects are at site 2 and half not at site 2. And half the subjects are at site 3 and half not at site 3. And half at site 4 and not at site 4. I am sure that's useful for some purpose, but I do not know what. No doubt OBSMARGINS will save the day. Here is SITE01 with OBSMARGINS specified.

site01	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
0	176.166194	3.692749	<.0001	0.2387
1	167.046205	4.730075	<.0001	

With OBSMARGINS

Below are the coefficients being used. Do they really make any more sense than the previous version? If we want to know what happens for site 1, why are we looking at the coefficients for the other sites? The problem is that SAS does not know that the SITExx variables are an interconnected set.

Coefficients for site01 Least Square Means			
		site01 Level	
Effect		0	1
Intercept		1	1
site01	0	1	0
site01	1	0	1
site02	0	0.59	0.59
site02	1	0.41	0.41
site03	0	0.946	0.946
site03	1	0.054	0.054
site04	0	0.942	0.942
site04	1	0.058	0.058
sex	Female	0.324	0.324
sex	Male	0.676	0.676

With OBSMARGINS

We elected to omit SITE05 but we could instead omit SITE01, thereby making it the reference category. Given that SITE05 is not in the model, we could have omitted it from the CLASS statement. Here is the SOLUTION we get when we omit SITE01 instead.

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		171.3767818	B	14.66463974	11.69	<.0001
site02	0	5.8598032	B	3.60969194	1.62	0.1048
site02	1	0.0000000	B	.	.	.
site03	0	5.5716041	B	7.54816613	0.74	0.4606
site03	1	0.0000000	B	.	.	.
site04	0	9.6351098	B	7.31388165	1.32	0.1880
site04	1	0.0000000	B	.	.	.
site05	0	-9.1199895	B	7.73588713	-1.18	0.2387
site05	1	0.0000000	B	.	.	.

Parameter		Estimate		Standard Error	t Value	Pr > |t|
sex	Female	-25.4760735	B	3.53522261	-7.21	<.0001
sex	Male	0.0000000	B	.	.	.

We get an exactly equivalent model (I will leave it to you to do the calculations) but with a different parameterization. By the way, perhaps now it is clear why I put SITE01-SITE05 in the CLASS statement: to guard against forgetting which SITExx variables I was going to use and ending up with some in the CLASS statement and some not in the CLASS statement. It is a little more overhead for SAS to process the unused dummy variable, but so much less frustrating than forgetting to include it. Suppose some but not all of the SITExx variables in the MODEL statement are in the CLASS statement?

We get a hybrid in this example with SITE02-SITE04 discrete and SITE05 continuous. This means when we go to look at LSMEANS we will have SITE05 set at its mean but the other SITExx variables balanced or not according to whether OBSMARGINS is specified. Confusing!

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		162.2567923	B	11.39452136	14.24	<.0001
site02	0	5.8598032	B	3.60969194	1.62	0.1048
site02	1	0.0000000	B	.	.	.
site03	0	5.5716041	B	7.54816613	0.74	0.4606
site03	1	0.0000000	B	.	.	.
site04	0	9.6351098	B	7.31388165	1.32	0.1880
site04	1	0.0000000	B	.	.	.
site05		9.1199895		7.73588713	1.18	0.2387
sex	Female	-25.4760735	B	3.53522261	-7.21	<.0001
sex	Male	0.0000000	B	.	.	.

Now let's take a look at what happens when we use the SITExx dummy variables but none of them are in the CLASS statement. First use SITE01-SITE04, so that SITE05 is the reference site.

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		192.4432989	B	7.38339614	26.06	<.0001
site01		-9.1199895		7.73588713	-1.18	0.2387
site02		-14.9797927		7.75226862	-1.93	0.0536
site03		-14.6915936		10.19479831	-1.44	0.1499
site04		-18.7550993		10.02261086	-1.87	0.0616
sex	Female	-25.4760735	B	3.53522261	-7.21	<.0001
sex	Male	0.0000000	B	.	.	.

Now let's make SITE01 the reference site instead.

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		183.3233093	B	2.80055152	65.46	<.0001
site02		-5.8598032		3.60969194	-1.62	0.1048
site03		-5.5716041		7.54816613	-0.74	0.4606
site04		-9.6351098		7.31388165	-1.32	0.1880
site05		9.1199895		7.73588713	1.18	0.2387
sex	Female	-25.4760735	B	3.53522261	-7.21	<.0001
sex	Male	0.0000000	B	.	.	.

Many coefficients change but the model is equivalent.

What is happening to the SEX least squares means with all these different equivalent models?

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	155.457930	3.585676	<.0001	<.0001
Male	180.934004	2.758138	<.0001	

SITE in CLASS Without OBSMARGINS

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	138.193988	9.352238	<.0001	<.0001
Male	163.670061	9.058654	<.0001	

SITE01-SITE04 in CLASS Without OBSMARGINS

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	151.873972	7.315993	<.0001	<.0001
Male	177.350046	6.806344	<.0001	

SITE02-SITE05 in CLASS Without OBSMARGINS

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	147.779097	5.953403	<.0001	<.0001
Male	173.255170	5.364892	<.0001	

SITE02-SITE04 in CLASS, SITE05 not in CLASS, Without OBSMARGINS

The lsmeans are very different when you use dummy variables in CLASS statements rather than SITE in the CLASS statement, and so are the standard errors. However, the difference between males and females is always the same.

When you specify OBSMARGINS in this situation, you end up with the same least squares means. You can accomplish the same thing by omitting SITE01-SITE04 from the CLASS statement.

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	155.050133	2.904531	<.0001	<.0001
Male	180.526207	2.009247	<.0001	

SITE in CLASS With OBSMARGINS

sex	y4 LSMEAN	Standard Error	H0:LSMEAN=0 Pr > |t|	H0:LSMean1=LSMean2 Pr > |t|
Female	155.050133	2.904531	<.0001	<.0001
Male	180.526207	2.009247	<.0001	

SITE01-SITE04 or SITE02-SITE05 in CLASS With OBSMARGINS

SITE01-SITE04 not in CLASS (With or Without OBSMARGINS)

LEAST SQUARES MEANS WITH INTERACTIONS

When a model includes interactions among discrete variables, the least squares means are more complicated and the issue of estimability arises. One way of thinking about estimability is to ask whether a given linear combination of parameters has the same value regardless of which parameterization is used. In the simple model with just SITE as a predictor, neither the intercept nor any of the individual values for the 5 sites is uniquely determined. But the sum of the intercept and the site effect is uniquely determined – it needs to equal the sample mean for that site. So the intercept plus a site effect is an estimable function. The SAS documentation has much more on estimability.

Let us turn our attention to a more realistic example. We have three variables that affect the outcome, SEX, RACE, and SITE. The racial-ethnic distribution varies considerably from site to site (which is typical of many studies). The distribution of SEX is similar across RACE and SITE but there are of course some random fluctuations. We think furthermore there may be some interactions among these three variables: the value of Y4 is thought to vary not just according to main effects but possibly with some additional complexity. We start with a fully-interacted model.

```
proc glm data=anal;
class sex race site;
model y4 = race sex site race*sex race*site sex*site race*sex*site / solution;
title3 "y4 = race sex site race*sex race*site sex*site race*sex*site";
run;
```

Source	DF	Type III SS	Mean Square	F Value	Pr > F
race	4	14774.31739	3693.57935	1.43	0.2209
sex	1	47263.32758	47263.32758	18.34	<.0001
site	4	5313.27168	1328.31792	0.52	0.7244
sex*race	4	61210.18136	15302.54534	5.94	0.0001
race*site	14	40126.34401	2866.16743	1.11	0.3422
sex*site	4	15408.09258	3852.02314	1.49	0.2017
sex*race*site	13	39332.13619	3025.54894	1.17	0.2933

Based on these results, I would remove the three-way interaction race*sex*site and retest the model. Then I would remove the least significant interaction and continue until all the effects are significant or are contained within a significant effect. We end up with a simpler model with just SITE RACE SEX RACE*SEX.

Source	DF	Type III SS	Mean Square	F Value	Pr > F
site	4	36683.9562	9170.9890	3.53	0.0072
race	4	23387.3465	5846.8366	2.25	0.0618
sex	1	126491.4775	126491.4775	48.70	<.0001
sex*race	4	77996.2271	19499.0568	7.51	<.0001

Note that SITE is highly significant now that we have adjusted for SEX and RACE and their interaction (this is the same data as we used for the earlier examples). Let's look at the LSMEANS.

race	y4 LSMEAN	Standard Error	Pr > |t|
1 White	161.998278	3.541149	<.0001
2 Black	175.088940	5.116060	<.0001
3 Hispanic	157.883347	5.234228	<.0001
4 Asian	170.846191	5.436250	<.0001
5 Other	164.445322	7.681423	<.0001

Without OBSMARGINS

race	y4 LSMEAN
1 White	Non-est
2 Black	Non-est
3 Hispanic	Non-est
4 Asian	Non-est
5 Other	Non-est

With OBSMARGINS

sex	y4 LSMEAN	Standard Error	Pr > |t|
Female	149.918237	4.196126	<.0001
Male	182.186594	3.010243	<.0001

Without OBSMARGINS

sex	y4 LSMEAN
Female	Non-est
Male	Non-est

With OBSMARGINS

sex	race	y4 LSMEAN	Standard Error	Pr > |t|
Female	1 White	154.688623	4.633583	<.0001
Female	2 Black	139.437614	8.213185	<.0001
Female	3 Hispanic	148.548572	7.899383	<.0001
Female	4 Asian	156.241477	8.775861	<.0001
Female	5 Other	150.674899	12.523136	<.0001
Male	1 White	169.307934	3.778542	<.0001

sex	race	y4 LSMEAN	Standard Error	Pr > |t|
Male	2 Black	210.740265	5.393662	<.0001
Male	3 Hispanic	167.218121	5.865276	<.0001
Male	4 Asian	185.450906	5.909248	<.0001
Male	5 Other	178.215744	8.916867	<.0001

Without OBSMARGINS

sex	race	y4 LSMEAN	Standard Error	Pr > |t|
Female	1 White	157.639603	3.848519	<.0001
Female	2 Black	142.388594	7.991616	<.0001
Female	3 Hispanic	151.499551	7.509113	<.0001
Female	4 Asian	159.192457	8.801103	<.0001
Female	5 Other	153.625879	12.556880	<.0001
Male	1 White	172.258913	2.778982	<.0001
Male	2 Black	213.691244	5.348196	<.0001
Male	3 Hispanic	170.169101	5.515505	<.0001
Male	4 Asian	188.401885	5.897934	<.0001
Male	5 Other	181.166723	9.086486	<.0001

With OBSMARGINS

And we have our first encounter with non-estimability. What went wrong? Well, the proportion of males and females differs across the race categories. So with OBSMARGINS the LSMEANS for the RACE variable doesn't give the same weight to males and females in each category and that means we don't have an unambiguous value for each race category. When we allow for equal weights (omitting the OM option), we assume half males and half females for every race and the RACE lsmeans are estimable. The same problem occurs for the SEX variable – the distribution of RACE is different for males than for females, so the OM version cannot calculate lsmeans for SEX but the equally-weighted version has no problem.

But the equally-weighted version assumes the races are equally represented (clearly unreasonable) and that the sites are all the same size (also unreasonable for this study). We really want the OM version – how can we make the lsmeans estimable? All we need to do is specify the same sex distribution for each race and otherwise use the observed margins. We can use the overall proportion of males and females in the study as a whole as the proportions. This unfortunately involves tedious specification of all the coefficients for the estimates. Fortunately we have the E option on the LSMEANS statement to tell us what SAS was going to use which gives us a nice head start. A simple PROC MEANS will provide the necessary values. However, you need to be careful to provide the coefficients to enough accuracy to ensure that the coefficients add up to exactly 1 (within the FUZZ value). Otherwise you will still be faced with non-estimability despite all your efforts. See Pasta (2010) for more.

COMPARING LEAST SQUARES MEANS IN MODELS WITH INTERACTIONS

Often when the model includes interactions among discrete variables, there are specific subgroups of interest to be compared or there is interest in which combinations of levels of the interacted variables is creating the interaction effect. The PDIFF option can be used with the interacted effects to compare every subgroup with every other subgroup. This produces a lot of tests, generally only a few of which will be of interest. It is a convenient way to get the combinations without using too much of your brainpower, however, and it's often the fastest way to get to where you want to be. I would be remiss if I did not point out the danger of interpreting *P* values in this context, as the issue of multiple comparisons looms large. If you do just a few pre-planned comparisons, the situation is not dire but you should be sure to remember (and alert your audience) to the number of tests you have considered and the expected number statistically significant by chance. Consider the following example.

```
proc glm data=anal;
class sex race site;
model y4 = site race sex race*sex / solution;
lsmeans race*sex / pdiff slice=race E;
store out=store01;
estimate "sex effect race=1" sex 1 -1  sex*race 1 0 0 0 0  -1 0 0 0 0;
estimate "sex effect race=2" sex 1 -1  sex*race 0 1 0 0 0  0 -1 0 0 0;
estimate "sex effect race=3" sex 1 -1  sex*race 0 0 1 0 0  0 0 -1 0 0;
estimate "sex effect race=4" sex 1 -1  sex*race 0 0 0 1 0  0 0 0 -1 0;
estimate "sex effect race=5" sex 1 -1  sex*race 0 0 0 0 1  0 0 0 0 -1;
title3 "y4 = site race sex race*sex with pdiff and slice=race";
run;
```

This illustrates the underutilized SLICE= option for use with interacted variables. The tests from both the PDIFF and SLICE= options can also be obtained by coding ESTIMATE statements but that is less convenient.

sex*race Effect Sliced by race for y4					
race	DF	Sum of Squares	Mean Square	F Value	Pr > F
1 White	1	26019	26019	10.02	0.0016
2 Black	1	149384	149384	57.51	<.0001
3 Hispanic	1	10776	10776	4.15	0.0419
4 Asian	1	20974	20974	8.07	0.0046
5 Other	1	8324.739096	8324.739096	3.20	0.0737

SLICE=RACE

Parameter	Estimate	Standard Error	t Value	Pr > |t|
sex effect race=1	-14.6193107	4.6190904	-3.16	0.0016
sex effect race=2	-71.3026501	9.4021124	-7.58	<.0001
sex effect race=3	-18.6695496	9.1659626	-2.04	0.0419
sex effect race=4	-29.2094289	10.2790386	-2.84	0.0046
sex effect race=5	-27.5408441	15.3838186	-1.79	0.0737

ESTIMATE STATEMENTS TO MIMIC SLICE=RACE

STORE AND PROC PLM FOR ADDITIONAL PROCESSING WITHOUT RE-ESTIMATING THE MODEL

The last example used the STORE statement, which allows you to store the model estimates in an “item store.” This statement is now available in most of the procedures that estimate linear or generalized linear models. We use PROC PLM to retrieve the item store to get additional information and perform various tests.

```
proc plm restore=store01;
lsmeans site race sex race*sex / pdiff E OM=anal;
estimate "sex=F using overall race" intercept 1 site .427 .410 .054 .058 .051
     race .547 .144 .139 .119 .051 sex 1 0
     sex*race .547 .144 .139 .119 .051  0 0 0 0 0;
estimate "sex=M using overall race" intercept 1 site .427 .410 .054 .058 .051
     race .547 .144 .139 .119 .051 sex 0 1
     sex*race 0 0 0 0 0  .547 .144 .139 .119 .051;
estimate "sex=F-M using overall race" sex 1 -1
     sex*race .547 .144 .139 .119 .051  -.547 -.144 -.139 -.119 -.051;
title3 "PLM with ESTIMATE statements";
run;
```

In order to use the OM option, we need to specify a dataset because the individual values needed to calculate the observed margins are not included in the item store. Also, the format of the LSMEANS output is different in PLM.

sex Least Squares Means					
sex	Estimate	Standard Error	DF	t Value	Pr > |t|
Female	149.92	4.1961	986	35.73	<.0001
Male	182.19	3.0102	986	60.52	<.0001

Without OBSMARGINS

Differences of sex Least Squares Means						
sex	_sex	Estimate	Standard Error	DF	t Value	Pr > |t|
Female	Male	-32.2684	4.6240	986	-6.98	<.0001

Without OBSMARGINS

Estimate					
Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex=F using overall race	154.57	2.8400	986	54.43	<.0001

Estimate					
Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex=M using overall race	180.31	1.9621	986	91.90	<.0001

Estimate					
Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex=F-M using overall race	-25.7399	3.4546	986	-7.45	<.0001

We have been able to specify the overall distribution of RACE to get estimable least squares means for females and males and the difference. Note that we get a different answer than we get without the OM option, which assumes each of the 5 races represents 20% of the population.

THE LSMESTIMATE STATEMENT

Another advantage of PROC PLM is that it provides access to certain new capabilities that have not been made available in legacy procedures. For example, the new LSMESTIMATE statement is not available in PROC GLM. However, if we STORE the results of the GLM model and analyze it in PROC PLM, we have access to the LSMESTIMATE statement. LSMESTIMATE is a combination of the LSMEANS and ESTIMATE statements. It allows you to calculate linear combinations (like the ESTIMATE statement), but instead of linear combinations of parameters it allows you to specify linear combinations of least squares means.

There are some variations in the syntax used in PROC PLM compared to that used in PROC GLM. For example, the LSMEANS statement in PROC PLM does not have a SLICE= option. Instead, there is a SLICE statement that provides the functionality. This also illustrates the use of the DIVISOR= option. Only selected output is shown.

```
proc plm restore=store01;
lsmeans race*sex / E;
slice race*sex / sliceby=race;
slice race*sex / sliceby=sex;
estimate "sex effect race=1" sex 1 -1  sex*race 1 0 0 0 0  -1 0 0 0 0;
estimate "sex effect race=2" sex 1 -1  sex*race 0 1 0 0 0  0 -1 0 0 0;
estimate "sex effect race=3" sex 1 -1  sex*race 0 0 1 0 0  0 0 -1 0 0;
estimate "sex effect race=4" sex 1 -1  sex*race 0 0 0 1 0  0 0 0 -1 0;
estimate "sex effect race=5" sex 1 -1  sex*race 0 0 0 0 1  0 0 0 0 -1;
lsmestimate sex "sex effect equal weights" 1 -1 / E;
lsmestimate sex "sex effect OM" 1 -1 / OM=anal E;
lsmestimate race "white v. black" 1 -1 0 0 0 ,
   "w+a v. rest no divisor" 3 -2 -2 3 -2 ,
   "w+a v. rest divisor=6" 3 -2 -2 3 -2 divisor=6 ,
   "w+a v. rest weighted" .82132 -.43114 -.41617 .17868 -.15269 / E;
lsmestimate race*sex "F: white v. black" 1 -1 0 0 0  0 0 0 0 0 ,
   "F: w+a v. rest weighted" .82132 -.43114 -.41617 .17868 -.15269  0 0 0 0 0 ,
   "M: w v. b" 0 0 0 0 0  1 -1 0 0 0 ,
   "M: w+a v. rest weighted"  0 0 0 0 0  .82132 -.43114 -.41617 .17868 -.15269 / E;
estimate "sex=F-M using overall race" sex 1 -1
   sex*race .547 .144 .139 .119 .051  -.547 -.144 -.139 -.119 -.051;
title3 "PLM with LSMESTIMATE and ESTIMATE statements";
run;
```

F Test for sex*race Least Squares Means Slice

Slice	Num DF	Den DF	F Value	Pr > F
race 1 White	1	986	10.02	0.0016

F Test for sex*race Least Squares Means Slice

Slice	Num DF	Den DF	F Value	Pr > F
sex Female	4	986	0.82	0.5093

Least Squares Means Estimate

Effect	Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex	sex effect equal weights	-32.2684	4.6240	986	-6.98	<.0001

Least Squares Means Estimate

Effect	Label	Margins	Estimate	Standard Error	DF	t Value	Pr > |t|
sex	sex effect OM	WORK.ANAL	Non-est	.	.	.	.

The SEX effect with equal weights mimics what we got from LSMEANS without the OM option. Here again we confirm that the SEX effect is non-estimable when specifying OM.

Least Squares Means Estimates						
Effect	Label	Estimate	Standard Error	DF	t Value	Pr > |t|
race	white v. black	-13.0907	5.6360	986	-2.32	0.0204
race	white+asian v. rest no divisor	3.6982	28.3227	986	0.13	0.8961
race	white+asian v. rest divisor=6	0.6164	4.7204	986	0.13	0.8961
race	white+asian v. rest weighted	-2.7241	4.0817	986	-0.67	0.5047

Least Squares Means Estimates						
Effect	Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex*race	F: white v. black	15.2510	8.9579	986	1.70	0.0890
sex*race	F: white+asian v. rest weighted	10.0209	6.2950	986	1.59	0.1117
sex*race	M: white v. black	-41.4323	6.1744	986	-6.71	<.0001
sex*race	M: white+asian v. rest weighted	-15.4691	4.5682	986	-3.39	0.0007

Estimate					
Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex effect race=1	-14.6193	4.6191	986	-3.16	0.0016

Estimate					
Label	Estimate	Standard Error	DF	t Value	Pr > |t|
sex=F-M using overall race	-25.7399	3.4546	986	-7.45	<.0001

Here we have used LSMESTIMATE to compare White and Asian (combined with equal weights and combined according to their representation in the data) with Black/Hispanic/Other combined. We get very different results for Females and Males. Note that although the ESTIMATE value changes with DIVISOR, it does not affect the significance test.

CONCLUSION

When a model includes discrete variables, the parameter estimates are often difficult to interpret and the test that they are zero may not be of interest. The LSMEANS statement allows the calculation of least squares means, also called adjusted means, for the values of a variable (or of interactions among discrete variables). There are also options to compare least squares means with or without adjustments for multiple comparisons. One of the things to pay attention to when a model includes more than one predictor variable is whether to specify the OBSMARGINS option (abbreviated OM) on LSMEANS. This option causes the LSMEANS to use the observed marginal distribution of the variable rather than using equal coefficients across classification effects (thereby assuming balance among the levels). Sometimes you want one version and sometimes you want the other, but in my work I generally find that OBSMARGINS more often gives me the LSMEANS I want. The issue of estimability also arises (assuming the model is less than full rank). It is quite possible for the LSMEANS to be nonestimable with the OM option but estimable without, or vice versa. Some time spent understanding the model, together with some tools that SAS provides, make the determination of estimability less mysterious. See Pasta (2010) for additional details.

When you have discrete (categorical) variables in your model, you are likely to want to use the LSMEANS statement to help interpret the results and to test specific hypotheses. It is important to understand what LSMEANS is doing and the implications of the OBSMARGINS (OM) option as well as the implication of including binary variables in the CLASS statement. Sometimes you need to use ESTIMATE or CONTRAST statements to test the desired hypotheses, but the SLICE and LSMESTIMATE statements can make construction of those tests much easier. The STORE statement and PROC PLM provide a convenient method for estimating combinations of parameters and testing hypotheses without having to re-estimate a model. I think STORE should be used routinely.

REFERENCES

Moses, Lincoln E., Emerson John D., and Hosseini, Hossein (1984), "Analyzing data from ordered categories," New England Journal of Medicine, 311:442-8. Reprinted as Chapter 13 in Bailar, John C. III and Mosteller, Frederick (1992) Medical Uses of Statistics, 2nd Ed., Boston, MA: NEJM Books

Pasta, David J. (2005), "Parameterizing models to test the hypotheses you want: coding indicator variables and modified continuous variables," Proceedings of the Thirtieth Annual SAS Users Group International Conference, 212-30 http://www2.sas.com/proceedings/sugi30/212-30.pdf

Pasta David J. (2009), "Learning when to be discrete: continuous vs. categorical predictors," Proceedings of the SAS Global Forum 2009, 248-2009 http://support.sas.com/resources/papers/proceedings09/248-2009.pdf

Pasta, David J. (2010), "Practicalities of using ESTIMATE and CONTRAST statements," Proceedings of the SAS Global Forum 2010, 269-2010 http://support.sas.com/resources/papers/proceedings10/269-2010.pdf

Pasta, David J. (2011), "Those confounded interactions: Building and interpreting a model with many potential confounders and interactions," Proceedings of the SAS Global Forum 2011, 347-2011 http://support.sas.com/resources/papers/proceedings11/347-2011.pdf

Pasta, David J. (2012) "Being continuously discrete (or discretely continuous): Understanding models with continuous and discrete predictors and testing associated hypotheses," Proceedings of the 2012 Western Users of SAS Software Regional Users Group Conference, Long Beach, California: Western Users of SAS Software

Potter, Lori and Pasta, David J (1997), "The sum of squares are all the same—how can the LSMEANS be so different?", Proceedings of the Fifth Annual Western Users of SAS Software Regional Users Group Conference, San Francisco: Western Users of SAS Software

Pritchard, Michelle L. and Pasta, David J. (2004), "Head of the CLASS: impress your colleagues with a superior understanding of the CLASS statement in PROC LOGISTIC," Proceedings of the Twenty-Ninth Annual SAS Users Group International Conference, 194-29 http://www2.sas.com/proceedings/sugi29/194-29.pdf

ACKNOWLEDGMENTS

Some of the material in this paper previously appeared in Pasta (2009-2012). My thanks to my coauthors on previous papers, Stefanie Silva Millar, Lori Potter, and Michelle Pritchard Turner, for their help.

CONTACT INFORMATION

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

David J. Pasta, Vice President
Medical Affairs Statistical Analysis
ICON Clinical Research
456 Montgomery Street, Suite 2200
San Francisco, CA 94104
(415) 371-2111
david.pasta@iconplc.com
www.iconplc.com