

Paper 351-2011

CONTRAST and ESTIMATE Statements Made Easy: The LSMESTIMATE Statement

Kathleen Kiernan, Randy Tobias, Phil Gibbs, and Jill Tao; SAS Institute Inc., Cary, NC

ABSTRACT

In many SAS/STAT[®] modeling procedures, the CONTRAST and ESTIMATE statements enable a variety of custom hypothesis tests, but using these statements correctly is often challenging. The new LSMESTIMATE statement, available in ten procedures in SAS/STAT 9.22 software, greatly simplifies the use of these statements. The LSMESTIMATE statement enables you to side-step parameterization issues and to specify custom tests in terms of population quantities of direct interest (the LS-means). The LSMESTIMATE statement also implements a new nonpositional syntax for specifying contrasts. This paper discusses these new features and demonstrates them with examples from actual user questions to the Statistical Procedures group in SAS Technical Support.

INTRODUCTION

The strength of SAS/STAT software for linear models has always been its flexibility, in that it enables you to test what you need to. In the past, you might have used the CONTRAST, ESTIMATE, or LSMEANS statements to generate custom hypothesis tests as part of a post-fitting analysis. With the release of SAS/STAT[®] 9.22 software, the LSMESTIMATE statement has been added to ten procedures to simplify the task of specifying custom hypotheses. Also, additional functionality has been added to the CONTRAST and ESTIMATE statements for several procedures. This paper briefly discusses concepts and statements common to a post-fitting analysis. The paper showcases the LSMESTIMATE statement in particular; highlighting that statement's enhanced functionality as well as its ease of use.

INTRODUCTORY EXAMPLE

Consider a medical experiment that evaluates the response to several treatment regimens. The objective of an experiment like this is often more specific than merely determining whether all of the treatments have the same effect on the response. You would not spend the time, money, and effort on the study if you thought that was likely! For example, you might be concerned with which of several new drugs works best, or you might be interested in how the efficacy of these drugs compares to the efficacy of a standard drug.

Furthermore, in factorial experiments or any designed experiment, significant interactions might be viewed as a problem. Interactions, admittedly, complicate the interpretation of results.

Another challenge is in translating custom tests for hypotheses, such as "The effect of treatment A in group 1 is equal to the treatment A effect in group 2," in terms of the model parameters. However, with the advent of SAS/STAT 9.22, these are problems of the past. Many SAS procedures now offer a wealth of tools for easing post-fitting comparisons, especially in the presence of these significant interactions.

Table 1 provides an overview of the post-fitting analysis capabilities that are available in eleven procedures in the SAS/STAT 9.22 software. In this table, a check mark (✓) indicates that new statements have been added to the procedures. A star (*) indicates previously existing functionality. The combination of a star and a check mark (*✓) indicates that existing statements have been updated.

Procedure	CONTRAST Statement	ESTIMATE Statement	LSMEANS Statement	LSMESTIMATE Statement
GENMOD	*	*	*✓	✓
GLM	*	*	*	
GLIMMIX	*	*	*	*✓
LOGISTIC	*	✓	✓	✓
MIXED	*	*	*	✓

continued

Procedure	CONTRAST Statement	ESTIMATE Statement	LSMEANS Statement	LSMESTIMATE Statement
ORTHOREG		✓	✓	✓
PHREG	*	✓	✓	✓
PLM		✓	✓	✓
SURVEYLOGISTIC	*	✓	✓	✓
SURVEYPHREG		✓	✓	✓
SURVEYREG	*	*	✓	✓

Table 1. Post-Fitting Statements That Are Available in Linear Modeling Procedures

Both the CONTRAST and the ESTIMATE statements deal with custom general linear functions $K'\beta$ of the model parameters β . In older procedures, such as PROC GLM and PROC MIXED, you can specify and estimate only one such linear function, $L'\beta$, with the ESTIMATE statement. In the CONTRAST statement, you can specify multiple functions; however, you can test only whether they are all simultaneously zero, $H: K'\beta=0$. In contrast (no pun intended), the newer implementation of the ESTIMATE statement in PROC LOGISTIC, PROC ORTHOREG, and other procedures covers both of these tasks. This implementation also augments the procedures with features for multiplicity adjustment, one-sided testing, graphics, and more.

The LSMEANS statement computes and analyzes *LS-means*, which are certain particularly informative linear combinations of the fixed-effect parameter estimates. Each effect in the LSMEANS statement is computed as $L'\beta$ for a certain column vector L , where β is the vector of fixed-parameter estimates. In this sense, the LSMEANS statement covers a subset of the analyses that are provided by the ESTIMATE statement, but it is a very important subset. The LS-means essentially generalizes the notion of group averages as analytical tools for nonorthogonal, unbalanced data. If you collect data about men and women and you just want to know whether the means are different, you compare gender averages. However, if you want to know whether they are different, adjusting for age, height, hair color, and so on, you then compare gender LS-means. The LSMEANS statement has all of the same features as the ESTIMATE statement for multiplicity adjustment, specialized graphics, and so on. It also has additional capabilities for comparing the LS-means in various ways.

Finally, the LSMESTIMATE statement essentially is a combination of the LSMEANS and ESTIMATE statements. This LSMESTIMATE statement gives you a way to obtain custom hypothesis tests that are defined not in terms of the fundamental model parameters β , but in terms of the LS-means, which are defined as $L'\beta$ in the preceding paragraph. Thus, the computation $KL'\beta$ for an LSMESTIMATE statement from the fundamental model parameters β involves two coefficient matrices, K and L . The L matrix defines the LS-means as functions of β , and the K matrix defines the linear combinations of the LS-means that you are interested in. Once again, the LSMESTIMATE statement has the same additional analytical features as the ESTIMATE statement. It also has some features that are specific to LS-means.

A new feature, nonpositional syntax, in the ESTIMATE and LSMESTIMATE statements defines, respectively, how linear combinations of parameters and linear combinations LS-means are specified. The CONTRAST and ESTIMATE statements in older procedures offered a syntax that relies on you knowing the position of each parameter in an ordered listing of them. This syntax uses zeros to skip over those positions that are irrelevant for a particular contrast of interest. This alternative, nonpositional syntax is more succinct and clearer for models with many parameters.

USING CONTRAST OR ESTIMATE STATEMENTS

How do the ESTIMATE, LSMEANS, and LSMESTIMATE statements and the new nonpositional syntax ease post-fitting analysis? To answer this question, start by reviewing the following steps. These steps are important to follow when you write traditional CONTRAST or ESTIMATE statements in procedures such as GLM and MIXED:

1. Define the statistical model for the data.
2. Define the hypothesis of interest in terms of cell means.
3. Redefine the hypothesis in terms of the model parameters.
4. Compute the coefficients for the CONTRAST or the ESTIMATE statement.

In order to use traditional CONTRAST and ESTIMATE statements with their "positional" syntax, you must also understand the parameterization and parameter ordering for your model. In order to define your statistical hypothesis properly and to relate that hypothesis to your SAS syntax, you need to understand this ordering, as explained in the following discussion.

Parameterization refers to the coding that is used to define the design variables that are generated by the CLASS statement. There are several coding schemes available in SAS; for example, glm (indicator or dummy) coding, effects (or deviation from mean) coding, and reference cell coding. The *glm coding* is the default coding for procedures such as the GENMOD, GLM, GLMSELECT, GLIMMIX, LIFEREG, MIXED, and SURVEYPHREG procedures. The *effect coding* is the default coding for the CATMOD, LOGISTIC, and SURVEYLOGISTIC procedures. The *reference cell coding* is the default coding for PHREG and TRANSREG procedures. Some procedures (for example, PROC LOGISTIC, PROC GENMOD, PROC GLMSELECT, PROC PHREG, PROC SURVEYLOGISTIC, and PROC SURVEYPHREG) allow different parameterizations of the CLASS variables. Note that many procedures (for example, PROC GLM, PROC MIXED, PROC GLIMMIX, and PROC LIFEREG) do not allow different parameterizations of CLASS variables. The examples in this paper are based on the glm coding of the CLASS variables.

The parameter ordering typically depends on the order in which the variables are specified in the CLASS statement and also on the setting of the ORDER= option in the PROC or CLASS statement. It is also helpful to know the order of the parameters within effects that have multiple parameters (such as interactions or nested effects). Several methods are available for determining and confirming the parameter ordering. Those methods involve examining the Class Level Information table, the Parameter Estimates table, or the Least Squares Means table in the fitting procedure's displayed results. In addition, the E option in the CONTRAST, ESTIMATE, LSMEANS, or LSMESTIMATE statement is useful in confirming the ordering of parameters for specifying L vector of coefficients when you define custom hypothesis tests.

The key to writing successful CONTRAST or ESTIMATE statements with the traditional positional syntax is to use the parameter multipliers as coefficients. You must also be careful to order the coefficients so that they match the order of the model parameters in the procedure, including the appropriate model parameters.

In a CONTRAST or ESTIMATE statement, the syntax for testing whether the general linear combination $L'\beta$ is equal to 0 is based on the way the parameters are assigned to the respective effects in the model parameters. The traditional way to define a linear combination of parameters to test a hypothesis is with positional syntax. In the positional syntax, you specify the name of each effect, followed by a list of coefficients for the parameters that correspond to the effect to be tested. The following data from Kutner (1974, p.98) illustrates a two-way ANOVA model and uses the ESTIMATE statement to perform additional custom hypothesis tests:

```

title "Two-way ANOVA Model, Kutner (1974, p.98)";
data a;
  input drug disease @;
  do i=1 to 6;
    input y @;
    output;
  end;
  datalines;
1 1 42 44 36 13 19 22
1 2 33 . 26 . 33 21
1 3 31 -3 . 25 25 24
2 1 28 . 23 34 42 13
2 2 . 34 33 31 . 36
2 3 3 26 28 32 4 16
3 1 . . 1 29 . 19
3 2 . 11 9 7 1 -6
3 3 21 1 . 9 3 .
4 1 24 . 9 22 -2 15
4 2 27 12 12 -5 16 15
4 3 22 7 25 5 12 .
;

proc glm data=a;
  class drug disease;
  model y=drug disease drug*disease;
  lsmeans drug/pdiff;
run;

```

The previous PROC GLM code generates several hypothesis tests in the default output (not shown). For example, the ANOVA table shows the overall significance of the model, the TYPE III SS shows the test for each effect, and the PDIFF option in the LSMEANS statement shows the comparison of LS-means.

In addition, suppose that you want to test some custom hypothesis tests such as the following:

- The average of drugs 1 and 2 is equal to the average of drugs 3 and 4.
- The mean of drug 3 is zero.
- The mean of drug 1 is the same as the mean of drug 2 for disease 2.

For these tests, you can use the CONTRAST or the ESTIMATE statement. For example, you can add the following statements to the previous PROC GLM code to test the aforementioned hypotheses:

```
estimate "Drug pair 1,2 vs drug pair 3,4"
      drug 1 1 -1 -1 /divisor=2;
estimate "Drug 3 mean"
      intercept 1 drug 0 0 1 0;
estimate "Drug 1 disease 2 vs drug 2 disease 2"
      drug 1 -1 drug*disease 0 1 0 0 -1;
```

It is important to remember the following facts about using positional syntax in a CONTRAST or ESTIMATE statement to specify the *L* vector values:

- In both the CONTRAST and the ESTIMATE statements, the coefficients of the specified main effect (drug) are equally distributed to the respective levels of the higher-ordered effect (drug*disease interaction). For this example, suppose you specify the following ESTIMATE statement:

```
estimate "drug 3 mean" intercept 1 drug 0 0 1 0;
```

In this case, PROC GLM assumes that the coefficients for the drug*disease term are as follows:

```
0 0 0 0 0 0.333333 0.333333 0.333333 0 0 0
```

Therefore, this ESTIMATE statement is equivalent to this statement:

```
estimate "drug 3 mean"
      intercept 1 drug 0 0 1
      drug*disease 0 0 0
                  0 0 0
                  0.333333 0.333333 0.333333
                  0 0 0;
```

In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and it does not contain any specified effects, then all of its coefficients in *L* are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

- Trailing zeros are not necessary. However, leading zeros and intermittent zeros are necessary placeholders that cannot be omitted.
- If too many values are specified for an effect, the extra ones are ignored. If too few are specified, the remaining ones are set to 0.
- It is good practice to use the E option in either the CONTRAST or the ESTIMATE statement in order to do the following:
 - examine the coefficients that are assigned to each of the effects and levels
 - verify the defined hypothesis that is being tested

USING THE LSMESTIMATE STATEMENT

As mentioned before, the LSMESTIMATE statement is essentially a combination of the LSMEANS and ESTIMATE statements. The syntax for the LSMESTIMATE statement is defined as follows:

```
LSMESTIMATE fixed-effect <'label'> values <divisor=>
              <, <'label'> values <divisor=>> <,> ...>
              </options>;
```

This syntax follows the same general form as that of the ESTIMATE statement except that it pertains to a single fixed effect rather than to the coefficients for all the effects in the model. You name this fixed effect first, before naming the (optional) label. You must do this first because, although multiple rows are allowed, they all pertain to combinations of the LS-means of the same effect. The coefficients are defined in terms of the LS-means of the specified fixed effect rather than directly in terms of the model parameters.

For example, you can write the ESTIMATE statements for the two-way ANOVA model in the previous section by using the following three LSMESTIMATE statements in the appropriate procedure from [Table 1](#).

```
lsmestimate drug "drug pair 1,2 vs drug pair 3,4" 1 1 -1 -1 /divisor=2;
lsmestimate drug "drug 3 mean" 0 0 1 ;
lsmestimate drug*disease "drug 1 disease 2 vs drug 2 disease 2" 0 1 0 0 -1;
```

The LSMESTIMATE statement provides a mechanism for obtaining custom hypothesis tests among LS-means. As compared to the LSMEANS statement, the LSMESTIMATE statement does not (automatically) produce the LS-means or their differences. However, you can use the statement to estimate any linear function of the LS-means.

The LSMESTIMATE statement was first made available in PROC GLIMMIX in SAS 9.1.3. In SAS/STAT 9.22 software, this statement is enhanced with additional functionality for multiple-comparison adjustments. Moreover, this statement is now available in the additional ten procedures shown (along with PROC GLIMMIX) in [Table 1](#).

SELECTED FEATURES OF THE LSMESTIMATE STATEMENT

Similar to the ESTIMATE statement, the LSMESTIMATE statement also supports the following features:

- nonpositional syntax
- multiple row tests and multiple-comparison adjustments
- ODS graphics
- TESTVALUE=, EXP, and ILINK options

NONPOSITIONAL SYNTAX

The nonpositional syntax that is available in the new ESTIMATE and LSMESTIMATE statements enables you to do the following:

- ignore the underlying ordering of parameters
- define directly only the nonzero coefficients that are involved in your hypothesis test

You can use either traditional positional or the new nonpositional syntax in the LSMESTIMATE statement, and you can even mix and match within the same statement. The following two LSMESTIMATE statements contrast using the positional and nonpositional syntax to specify the comparison for testing the difference of Drug A versus Drug B in a one-way ANOVA model:

```
lsmestimate drug "Drug: A vs. B" 1 -1; /* Positional */
lsmestimate drug "Drug: A vs. B" [1, 1] [-1, 2]; /* Nonpositional */
```

Each bracketed term in the nonpositional syntax defines a coefficient of L , where the first argument in brackets defines the coefficient and the second argument defines the level of the effect. If the effect involves continuous variables, then the values of continuous variables needed for the construction of L must precede the level indicators of the CLASS variables. The ESTIMATE and LSMESTIMATE statements support both the positional and nonpositional syntax. You can combine the traditional positional syntax with the nonpositional syntax for different effects in the same statement.

The following three LSMESTIMATE statements use nonpositional syntax. Using the appropriate procedure from [Table 1](#), compare these statements with the three previously shown ESTIMATE statements that use the traditional positional syntax for the two-way ANOVA model:

```
lsmestimate drug "drug pair 1,2 vs drug pair 3,4"
      [ 1,1] [ 1,2]
      [-1,3] [-1,4] / divisor=2;

lsmestimate drug "drug 3 mean"
      [ 1,3];

lsmestimate drug*disease "drug 1 disease 2 vs drug 2 disease 2"
      [ 1,1 2]
      [-1,2 2];
```

In these LSMESTIMATE statements, the first value (typically, 1 or -1) is the coefficient, and the values after the comma are the levels of the specified effect. In nonpositional syntax, you do not need to have zeros to occupy positions as you do in the positional syntax; this difference makes using the new nonpositional syntax more intuitive.

MULTIPLE ROW TESTS FOR JOINT HYPOTHESES AND ADJUSTMENTS FOR MULTIPLE COMPARISONS

If you have more than one LS-means hypothesis to test, you can control the overall error using one of two methods. One method is to test all the hypotheses jointly, enabling you to say whether it is likely that they are all simultaneously null. As with an overall ANOVA F test, a significant result in this case tells you that *something* is going on, but it does not say what. Multiplicity adjustments, in addition to jointly testing the hypothesis, provide another way to control the overall error. The new ESTIMATE and LSMESTIMATE statements support multiple row tests with multiplicity adjustments (the ADJUST= option) and joint tests (the JOINT or FTEST option). Multiple comparisons and multiplicity adjustments, with many examples of LSMEANS, LSMESTIMATE, and ESTIMATE statements, are covered in the forthcoming second edition of the book by Westfall et al. (**Note:** The anticipated publication date for this book is summer 2011.)

To see the joint-test capacity of the LSMESTIMATE statement in action, consider one-way data on the yield of an industrial process when the machine is operated by each of nine different people.

```
data Quality;
  do Operator=1 to 9;
    do i=1 to 4; input Yield @@; output; end;
  end;
datalines;
  5.8 3.9 4.4 2.9
  6.2 3.4 4.5 3.9
  3.4 4.0 3.3 3.7
  6.7 5.2 5.3 5.2
  6.3 5.7 4.7 6.4
  10.3 10.4 8.6 10.6
  7.9 8.1 10.0 9.4
  8.9 10.2 8.4 8.4
  8.7 8.9 10.0 7.8
;
```

You expect to see a significant difference between operators, because you think the first three operators all do things the same way, as do the next two, as do the last four. You can test for this type of discovered nesting by using two LSMESTIMATE statements similar to the following:

```
proc orthoreg data=Quality;
  class Operator;
  model Yield = Operator;
  lsmestimate Operator "Clustered operators"
    [1,1] [-1,2], [1,1] [-1,3],
    [1,4] [-1,5],
    [1,6] [-1,7], [1,6] [-1,8], [1,6] [-1,9] / joint;
  lsmestimate Operator "Different clusters"
    [4,1] [4,2] [4,3] [-6,4] [-6,5],
    [4,1] [4,2] [4,3] [-3,6] [-3,7] [-3,8] [-3,9] / joint;
  ods select Contrasts;
run;
```

The F test is indeed significant. The LSMESTIMATE statement for the clustered operators tests for differences within the three hypothesized clusters; the associated p -value is > 0.4 . This value indicates that there are no significant differences within operator clusters. On the other hand, the p -value for the different clusters, which corresponds to differences between the three cluster averages, is highly significant.

ODS GRAPHICS

The LSMEANS statement includes the use of full-featured SAS ODS Graphics for depicting LS-means and their differences. ODS Graphics is also supported in both the LSMESTIMATE statement and the new ESTIMATE statement, for the two procedures (PROC GENMOD and PROC PHREG) that can perform Bayesian analysis. The plots are available from these procedures directly and also from the PLM procedure when you use an item store that is created by these procedures.

TESTVALUE=, EXP, AND ILINK OPTIONS

Most linear tests involve hypotheses of no difference, meaning that you are testing whether some linear function of the parameters or LS-means is zero. However, sometimes you need to test a quantity against a prespecified nonzero value. For example, to test the hypothesis that the difference between the first two LS-means is 50, you can use the TESTVALUE= option as shown in this example:

```
lsmestimate A "A1 - A2" 1 -1/ testvalue=50;
```

For nonnormal data, the EXP and ILINK options give you a way to obtain the quantity of interest on the scale of the mean (inverse link). Results presented in this fashion can be much easier to interpret than data on the link scale. For example, the following LOGISTIC procedure returns the odds ratio and log odds ratio for comparing A1 to A2:

```
proc logistic;
  class A/param=glm;
  model y=A / link=logit;
  lsmestimate A "odds ratio: A1/A2" 1 -1/ exp;
run;
```

In summary, the syntax in an LSMESTIMATE statement is simpler than that in the corresponding ESTIMATE statement. You can simplify multiple-comparison methods by using the LSMESTIMATE statement when the comparisons involve a single factor. If the comparison includes more than one factor, there is no inherent advantage to using the LSMESTIMATE statement. However, it can be useful to use the statement initially with the E option in order to make sure that the ESTIMATE statement is correct.

EXAMPLES USING THE LSMESTIMATE STATEMENT

The following examples illustrate different ways you can use the LSMESTIMATE statement to simplify the coding of custom hypothesis tests. With simpler models, the savings that are realized through the use of the LSMESTIMATE statement might not be that great. With more complicated models, however, this new statement can greatly shorten the length of the statement needed to represent the hypothesis and lessen the chances of coding the hypothesis incorrectly.

EXAMPLE 1: SPLIT-PLOT DESIGN WITH CUSTOM HYPOTHESIS TEST USING THE MIXED PROCEDURE

This example represents a balanced split-plot design. A custom hypothesis of interest in this experiment concerns whether the average of the first two levels of A is equal to a quarter of the third level. This hypothesis can be expressed as follows:

$$H_0: (\mu_1 + \mu_2)/2 = \mu_3/4 \text{ (or, equivalently: } H_0: \mu_1 + \mu_2 - \mu_3/2 = 0)$$

The data and the initial analysis, without the custom test, follow:

```
data sp;
  input Block A B Y @@;
  datalines;
1 1 1 56 1 1 2 41
1 2 1 50 1 2 2 36
1 3 1 39 1 3 2 35
2 1 1 30 2 1 2 25
2 2 1 36 2 2 2 28
2 3 1 33 2 3 2 30
3 1 1 32 3 1 2 24
3 2 1 31 3 2 2 27
3 3 1 15 3 3 2 19
4 1 1 30 4 1 2 25
4 2 1 35 4 2 2 30
4 3 1 17 4 3 2 18
;

proc mixed data=sp;
  class A B Block;
  model Y = A B A*B;
  random intercept A / subject=Block;
  lsmeans A;
run;
```


The results from the LSMEANS statement from the PROC MIXED program show the following:

Least Squares Means						
Effect	A	Estimate	Standard Error	DF	t Value	Pr > t
A	1	32.8750	4.5403	6	7.24	0.0004
A	2	34.1250	4.5403	6	7.52	0.0003
A	3	25.7500	4.5403	6	5.67	0.0013

The estimated difference that you are interested in testing is $32.875 + 34.125 - 0.5(25.570) = 54.125$. However, you cannot easily calculate the standard error and, therefore, the significance of this estimated difference, because of correlation between the LS-means.

A naïve first attempt to use an ESTIMATE statement to test this hypothesis test might be as follows:

```
estimate "A1 + A2 vs 0.5A3" a 1 1 -0.5/E;
```

However, this statement returns a nonestimable result because this hypothesis also should include a term for the intercept when the test is expressed in terms of the model parameters. If you express this test in terms of the LSMEANS by using the LSMESTIMATE statement, then this syntax yields a valid result for this test. This next LSMESTIMATE statement uses both the positional and nonpositional syntax:

```
lsmestimate A "positional A1 + A2 vs 0.5A3" 1 1 -0.5,
             "nonpositional A1 + A2 vs 0.5A3"
             [1, 1] [1,2] [-0.5,3];
```

This statement generates the following results:

Least Squares Means Estimates						
Effect	Label	Estimate	Standard Error	DF	t Value	Pr > t
A	positional A1 + A2 vs 0.5A3	54.1250	6.8105	6	7.95	0.0002
A	nonpositional A1 + A2 vs 0.5A3	54.1250	6.8105	6	7.95	0.0002

As expected, the two versions of the test that use the different syntax yield the same result.

Adding the E option to the LSMESTIMATE statement shows the coefficients that are necessary to generate this test result with the ESTIMATE statement.

EXAMPLE 2: THREE-WAY FACTORIAL DESIGN WITH SIGNIFICANT INTERACTIONS

The following example demonstrates a three-way factorial design with significant interaction effects. It also provides guidance on how to proceed with some comparisons of interest. The objective of this study is to determine the effect of three training programs on exercise tolerance, measured as minutes until fatigue, in running a marathon. There are three factors in the study: the gender of the study participant, the training program that is followed, and the terrain type that is used under the training program.

In general, a full-factorial model with three factors includes the three main effects, all two-way interactions of the main effects, and the three-way interaction of the three main effects. All of these effects can make for a lengthy model, one that can be especially difficult to work with in postprocessing and creating custom hypothesis tests.

The data for this example is simulated using the following DATA step:

```
data test;
  call streaminit(6123451);
  do gender = 1 to 2;
    do program = 1 to 3;
      do terrain = 1 to 3;
        do rep=1 to ceil(rand('uniform')*3)+3;
          y=165 + 3*gender + program - terrain
            + gender*program - gender*terrain;
        end;
      end;
    end;
  end;
```



```

+ program*terrain - 2*gender*program*terrain
+ rand('normal');
output;
end;
end;
end;
end;
run;

```

The simulation code creates data with two levels for gender, three levels for program, and three levels for terrain type. You can use the following code to estimate the full-factorial model:

```

proc mixed data=test namelen=25;
class gender program terrain;
model y=gender|program|terrain;
run;

```

The results from the MODEL statement show in this procedure are as follows:

Type 3 Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
gender	1	75	427.69	<.0001
program	2	75	62.57	<.0001
gender*program	2	75	42.48	<.0001
terrain	2	75	963.72	<.0001
gender*terrain	2	75	144.30	<.0001
program*terrain	4	75	32.88	<.0001
gender*program* terrain	4	75	5.25	0.0009

Running this full-factorial model produces results that are significant for all of the effects in the model (not a surprise, considering that the data are simulated that way). A hypothesis of interest to the study is to compare gender results across levels of the exercise program. One specific case of interest is to compare the results across the genders for the first level of the training program and the first terrain type. A CONTRAST statement for that comparison requires coefficients on the main effects and two-way interactions, as well as on the three-way interaction effect:

```

contrast "diff GPT111 - GPT211"
gender 1 -1
gender*program 1 0 0 -1 0 0
gender*terrain 1 0 0 -1 0 0
gender*program*terrain 1 0 0 0 0 0 0 0 0 -1;

```

The results from the CONTRAST statement in the MIXED procedure are as follows:

Contrasts				
Label	Num DF	Den DF	F Value	Pr > F
diff GPT111 - GPT211	1	75	2.04	0.1578

If you re-express this test as a function of only the LS-means, then you only need coefficients for the comparisons of the LS-means. You do not need to determine the coefficients for the lower-ordered interaction terms as well. The LSMESTIMATE statement for this hypothesis test, using both the positional and nonpositional syntax, is as follows:

```
lsmestimate gender*program*terrain
  "Positional    GPT111 - GPT211" 1 0 0 0 0 0 0 0 0 -1,
  "Nonpositional GPT111 - GPT211" [1, 1 1 1] [-1, 2 1 1];
```

With the need for all of the place-holding zeros, the placement of the coefficients for the LS-means in the positional syntax can be difficult. The nonpositional syntax removes that difficulty, and the compactness of the syntax makes the statement much easier to interpret and work with.

The following table shows the results of this LSMESTIMATE statement:

Least Squares Means Estimates							
Effect	Label		Estimate	Standard Error	DF	t Value	Pr > t
gender*program*terrain	Positional	GPT111 - GPT211	-0.9832	0.6891	75	-1.43	0.1578
gender*program*terrain	Nonpositional	GPT111 - GPT211	-0.9832	0.6891	75	-1.43	0.1578

Both sets of results from the CONTRAST and LSMESTIMATE statements in this example indicate that there is no significant difference between gender for the first level of the training program and the first terrain type.

EXAMPLE 3: USING THE SLICE OPTION AND THE SLICE STATEMENT FOR MULTIPLE COMPARISONS

Typically, when the values of your response depend on multiple classification factors jointly, you look for simpler models that explain that joint behavior. One familiar way of doing this is to model main effects and interactions of various orders, hoping to find that higher-order interactions are all insignificant. Another way is exemplified by the contrast that is illustrated in the previous example. That case looks for differences due to one factor (Gender) while holding other factors (Program and Terrain) fixed. In effect, that example sliced the three-way Gender*Program*Terrain interaction by Program and Terrain, considering only the Gender differences within one slice. Another term for the sliced Gender differences is the *simple effects* of Gender (Winer 1971).

To perform this analysis easily, use the SLICE= option in the LSMEANS statement or the new SLICE statement that is available in some procedures. For example, with the previous data and model, consider the following statement:

```
lsmeans gender*program*terrain / slice=program*terrain;
```

This statement results in 36 three-way LS-means as well as a table of nine Gender tests, one for each combination of Program and Terrain:

Tests of Effect Slices						
Effect	program	terrain	Num DF	Den DF	F Value	Pr > F
gender*program*terrain	1	1	1	75	2.04	0.1578
gender*program*terrain	1	2	1	75	13.21	0.0005
gender*program*terrain	1	3	1	75	68.07	<.0001
gender*program*terrain	2	1	1	75	0.44	0.5077
gender*program*terrain	2	2	1	75	40.18	<.0001
gender*program*terrain	2	3	1	75	192.70	<.0001

continued

Tests of Effect Slices						
Effect	program	terrain	Num DF	Den DF	F Value	Pr > F
gender*program*terrain	3	1	1	75	2.24	0.1387
gender*program*terrain	3	2	1	75	129.16	<.0001
gender*program*terrain	3	3	1	75	378.51	<.0001

The p -value in the first row (0.1578) matches the one from the LSMESTIMATE statement in the previous example. The p -value in the second row (0.0005) indicates that there is a significant difference between gender for the first level of the training program and the second terrain type.

EXAMPLE 4: REPEATED-MEASURES DESIGN WITH ADJUSTED COMPARISONS OF THE SIMPLE EFFECTS WITHIN A SLICE

Several procedures in SAS/STAT 9.22 software offer the SLICEDIFF= and ADJUST= options for producing the adjusted comparisons of the simple effects within a slice. (In order to perform a similar analysis in previous releases, you had to save all comparisons to a SAS data set, retain only those comparisons of interest in a subsequent DATA step, and then use the MULTTEST procedure to obtain the comparisons.)

The following example illustrates how you can use the SLICEDIFF= and ADJUST= options together. First, simulate some data to work with. The following DATA step simulates data from an experiment on 20 subjects (ID). Suppose that the subjects were exposed to one of three treatments (TRT) and observed over two time (TIME) points:

```
data test;
  call streaminit(1451345);
  do id=1 to 20;
    rid=rand('normal');
    trt=ceil(rand('uniform')*3);
    do time=1 to 2;
      y=trt + trt*time + rand('normal') + rid;
    output;
  end; end;
run;
```

You can analyze this data using the following MIXED procedure:

```
proc mixed data=test;
  class id trt time;
  model y=trt time trt*time;
  repeated time / subject=id(trt) type=cs;
run;
```

The results from this MIXED procedure show the following:

Type 3 Tests of Fixed Effects				
Effect	Num DF	Den DF	F Value	Pr > F
trt	2	17	56.45	<.0001
time	1	17	26.15	<.0001
trt*time	2	17	6.75	0.0070

Given that the TRT*TIME interaction is significant, you might want to explore differences in the levels of that interaction using the LSMEANS statement. Specifically, you can explore treatment differences at each of the two time points by adding the LSMEANS statement with the SLICE= option.

```
proc mixed data=test;
  class id trt time;
  model y=trt time trt*time;
  repeated time / subject=id(trt) type=cs;

  lsmeans trt*time / slice=time;
  ods select slices;
run;
```

The result from the LSMEANS statement with the SLICE= option in this procedure shows the following:

Tests of Effect Slices					
Effect	time	Num DF	Den DF	F Value	Pr > F
trt*time	1	2	17	23.35	<.0001
trt*time	2	2	17	54.26	<.0001

The result of the SLICE= option gives you two *F* tests for treatment differences, at time point 1 and time point 2. Since these *F* tests are both significant, the next logical step is to explore the pairwise differences of the treatments within each of the time points. To do this in PROC MIXED, you need to write ESTIMATE or LSMESTIMATE statements. However, PROC GLIMMIX, which can also fit this model, has a newer form of the LSMEANS statement that makes this task easier.

The new SLICEDIFF= option can take the place of several LSMESTIMATE statements, as shown in this example:

```
proc glimmix data=test;
  class id trt time;
  model y = trt time trt*time;
  random time / subject=id(trt) type=cs rside;
  lsmeans trt*time/slice=time slicediff=time adjust=bon;
  ods select slicediffs;
run;
```

The LSMEANS statement here gives you Bonferroni-adjusted comparisons of the levels of TRT within each level of TIME. The SLICEDIFF= option requests simple effects differences. In this case, you get all pairwise TRT differences. However, if comparisons with a control are of interest, you can use the SLICEDIFFTYPE= option to specify which levels of the effects are the controls. The ODS graphic DIFFPLOT, available in the LSMEANS statement through the PLOT=DIFF option, is useful in interpreting these differences.

The following results from the SLICEDIFF= option in the LSMEANS statement from the PROC GLIMMIX program shows the pairwise comparisons among TRT for each level of TIME. It also provides adjusted *p*-values for multiple comparisons.

Simple Effect Comparisons of trt*time Least Squares Means by time								
Simple Effect Level	trt	_trt	Estimate	Standard Error	DF	t Value	Pr > t	Adj P
time 1	1	2	-0.7087	0.6569	17	-1.08	0.2957	0.8872
time 1	1	3	-4.0820	0.6747	17	-6.05	<.0001	<.0001
time 1	2	3	-3.3733	0.5964	17	-5.66	<.0001	<.0001
time 2	1	2	-3.0236	0.6569	17	-4.60	0.0003	0.0008
time 2	1	3	-6.9096	0.6747	17	-10.24	<.0001	<.0001
time 2	2	3	-3.8860	0.5964	17	-6.52	<.0001	<.0001

The first row of the table indicates that at Time 1, Trt 1 is 0.7087 below Trt 2 in terms of the mean of *y*. This difference is clearly not significant with the adjusted *p*-value that is equal to 0.8872.

EXAMPLE 5: MULTIPLE COMPARISONS FOR A NONPARAMETRIC MODEL

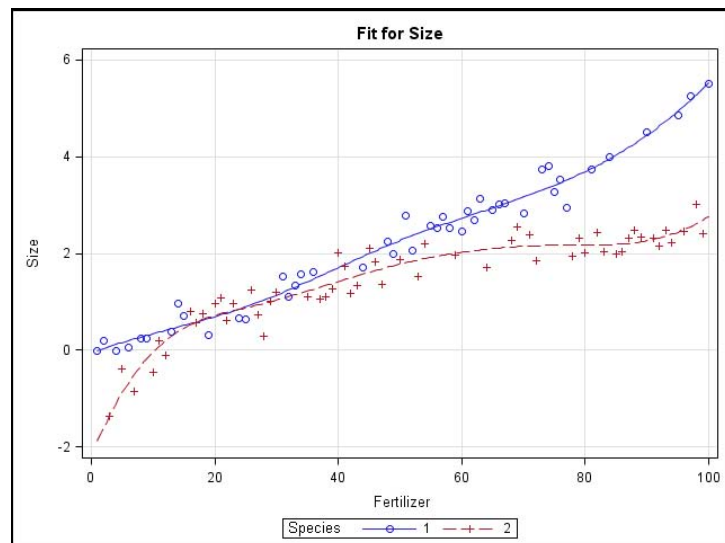
To explore new possibilities for multiple comparisons for a nonparametric model, a graduate student in agronomy decides to conduct an experiment to verify the amount of fertilizer that affects various species' growth rates. In this experiment, there are different nonlinear relationships and different forms of the relationship, not just different parameters, across the species.

This experiment is conducted using different amounts of fertilizer on each of 100 plants of different species and recording the size of their flowers. The following data set and code demonstrates the statistical analysis that is conducted by the graduate student. This code is similar to the methods described in Tobias and Cai (2010). This example uses the new EFFECT statement in PROC ORTHOREG in conjunction with the plotting and advanced multiple-comparison methods to determine the precise region of interest for a complicated nonparametric model.

```
ods graphics on;
data Flowers;
  input Species Size @@;
  Fertilizer = _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
;

title 'Flower Species Fertilizer Effects';
proc orthoreg data=Flowers;
  class Species;
  effect SmoothF=spline(Fertilizer);
  model Size=Species|SmoothF;
  effectplot / obs;
  test Species|SmoothF;
  store FlowerModel;
run;
```

The EFFECTPLOT statement prints the following summary plot of the data and the nonparametric fit:



The two curves are obviously different. This difference is corroborated by the significance of the SmoothF*Species effect in the table of Type 3 tests shown here:

Type III Tests of Model Effects				
Effect	Num DF	Den DF	F Value	Pr > F
Species	1	86	13.30	0.0005
SmoothF	6	86	290.42	<.0001
SmoothF*Species	6	86	30.47	<.0001

However, there seems to be an interval of fertilizer amounts where the difference between the species' growth rates is insignificant. The subsequent PROC PLM run (shown below), which estimates multiplicity-corrected species differences over a range of fertilizer amounts, focuses on this region.

```
%macro GroupDiff;
  %do x=0 %to 75 %by 5;
    "Diff at Fertilizer=&x" Species 1 -1 Species*SmoothF [1,1 &x] [-1,2 &x],
  %end;
  "Diff at Fertilizer=80" Species 1 -1 Species*SmoothF [1,1 80] [-1,2 80]
%mend;

proc plm restore=FlowerModel;
  estimate %GroupDiff / adjust=simulate seed=1 stepdown;
  filter adjp > 0.05;
  ods select Estimates;
run;
```

This code results in the following table:

Estimates Adjustment for Multiplicity: Holm-Simulated						
Label	Estimate	Standard Error	DF	t Value	Pr > t	Adj P
Diff at Fertilizer=10	0.3778	0.1540	86	2.45	0.0162	0.0545
Diff at Fertilizer=15	0.05822	0.1481	86	0.39	0.6952	0.9101
Diff at Fertilizer=20	-0.02602	0.1243	86	-0.21	0.8346	0.9565
Diff at Fertilizer=25	0.02014	0.1312	86	0.15	0.8783	0.9565
Diff at Fertilizer=30	0.1023	0.1378	86	0.74	0.4600	0.7418
Diff at Fertilizer=35	0.1924	0.1236	86	1.56	0.1231	0.2925

The graduate student determines that the difference between the two species is insignificant when the fertilizer levels are between 10 and 35 units. Therefore, he concludes that as long as you use that moderate amount of the fertilizer, you can apparently ignore the difference between species.

EXAMPLE 6: RUNNING WITH THE NEW PLM PROCEDURE

As is evident in Example 4, with the lack of a SLICEDIFF= option in the MIXED procedure's LSMEANS statement, the march of progress in SAS/STAT software sometimes collides with the commitment that SAS has to upward compatibility. This progression surpasses older forms of some statements in more mature procedures. The PLM procedure, new in SAS/STAT 9.22 software, gives older procedures (for example PROC GLM and PROC MIXED) a new lease on life. PROC PLM enables you to run additional post-fitting analyses using the most up-to-date linear modeling facilities with the stored fitted model. The following procedures support the STORE statement and, therefore, they can take advantage of PROC PLM for post-fitting analysis

- PROC GENMOD
- PROC GLIMMIX
- PROC GLM
- PROC LOGISTIC
- PROC MIXED

- PROC ORTHOREG
- PROC PHREG
- PROC SURVEYLOGISTIC
- PROC SURVEYPHREG
- PROC SURVEYREG.

Note: See Tobias and Cai (2010) for more details.

The following example uses PROC GLM to fit the model and store the results. Because the LSMESTIMATE statement is not available in PROC GLM, the example then uses the PLM procedure with the LSMESTIMATE statement.

In this example, the data set contains the test scores on phonics for second graders in a few US school districts. The factors are Gender, School, and Semesters. Semesters specifies the number of semesters that the student has been in the school district at the time of measurements.

```

data school2;
  input school :$10. reps @;
  do i=1 to reps;
    input gender $ semesters phonics3 @;
    output;
  end;
  datalines;
Cottonwood 43          F 4 67 F 4 55 F 4 .
F 4 41 F 6 38 F 6 49 F 6 51 F 6 51 F 6 55 F 6 62 F 6 40
F 6 41 F 6 45 F 6 44 F 6 41 F 6 61 F 6 50 F 6 33 M 4 63
M 4 56 M 4 41 M 4 32 M 6 47 M 6 52 M 6 10 M 6 47 M 6 55
M 6 31 M 6 40 M 6 58 M 6 48 M 6 47 M 6 54 M 6 . M 6 .
M 6 44 M 6 53 M 6 50 M 6 32 M 6 42 M 6 45 M 8 46 M 8 51
Dogwood 49            F 4 50
F 4 57 F 4 53 F 6 58 F 6 58 F 6 65 F 6 61 F 6 67 F 6 74
F 6 39 F 6 64 F 6 68 F 6 70 F 6 67 F 6 36 F 6 . F 6 62
F 6 . F 6 63 F 6 71 F 6 78 F 6 77 F 6 64 F 6 54 F 6 57
F 8 30 M 4 63 M 4 54 M 4 52 M 4 71 M 6 45 M 6 0 M 6 35
M 6 47 M 6 48 M 6 . M 6 57 M 6 39 M 6 52 M 6 42 M 6 42
M 6 43 M 6 32 M 6 56 M 8 56 M 8 39 M 8 . M 8 61 M 8 0
Maple 31
F 4 53 F 6 64 F 6 25 F 6 45 F 6 76 F 6 64 F 6 52
F 6 68 F 6 65 F 6 64 F 6 49 F 6 72 F 6 57 F 8 56 M 4 .
M 4 18 M 6 53 M 6 60 M 6 65 M 6 60 M 6 59 M 6 59 M 6 .
M 6 59 M 6 53 M 6 64 M 6 53 M 6 55 M 6 64 M 6 65 M 8 50
Pine 67              F 4 65 F 4 38 F 4 .
F 4 58 F 4 . F 4 61 F 4 54 F 6 69 F 6 64 F 6 68 F 6 41
F 6 66 F 6 57 F 6 60 F 6 48 F 6 60 F 6 52 F 6 62 F 6 68
F 6 58 F 6 69 F 6 62 F 6 67 F 6 59 F 8 45 M 4 65 M 4 49
M 4 50 M 4 56 M 4 46 M 4 72 M 4 72 M 4 55 M 4 49 M 6 64
M 6 48 M 6 60 M 6 63 M 6 53 M 6 60 M 6 66 M 6 65 M 6 55
M 6 65 M 6 52 M 6 46 M 6 57 M 6 55 M 6 42 M 6 54 M 6 44
M 6 69 M 6 45 M 6 58 M 6 56 M 6 63 M 6 71 M 6 48 M 6 74
M 6 58 M 6 62 M 8 54 M 8 39 M 8 49 M 8 . M 8 62 M 8 53
;
run;

proc glm data=school2 namelen=30;
  class gender semesters school;
  model phonics3=gender|semesters|school;
  store glmresults;
run;

```


The STORE statement creates an item store that stores the model fitting information from PROC GLM. Then, the PLM procedure uses the LSMESTIMATE statement to estimate the differences between two groups.

```
proc plm source=glmresults;
  lsmestimate gender*semesters*school
    "positional 6 semesters girls: Cottonwood vs. Maple"
    0 0 0 0 1 0 -1 0,
    "nonpositional 6 semesters girls: Cottonwood vs. Maple"
    [1, 1 2 1][-1, 1 2 3];
run;
```

The results from the LSMESTIMATE statement from the PROC PLM program show the following:

Least Squares Means Estimates						
Effect	Label	Estimate	Standard Error	DF	t Value	Pr > t
gender*semesters*school	positional 6 semesters girls: Cottonwood vs. Maple	-11.2024	4.2451	155	-2.64	0.0092
gender*semesters*school	nonpositional 6 semesters girls: Cottonwood vs. Maple	-11.2024	4.2451	155	-2.64	0.0092

The result of the LSMESTIMATE statement indicates there is a significant difference between the girls who attended six semesters in the Cottonwood school district and the girls who attended six semesters in the Maple school district.

EXAMPLE 7: A TWO-FACTOR LOGISTIC MODEL WITH INTERACTION USING GLM (INDICATOR) CODING

Consider this fictitious study in which a group of statisticians with either a master's (MS) degree or a doctoral (PhD) degree participate in one of three games (blackjack, poker, or slots). The result of playing the games (winner or loser) is measured. The following DATA step and LOGISTIC procedure enable you to model the probability of losing and to determine whether there are differences due to the type of game, the education level, or an interaction between them. In addition, this example estimates two odds—the odds of a statistician with a master's degree losing at blackjack and the odds of the statistician losing at slots. The example then illustrates how the CONTRAST, LSMESTIMATE, and SLICE statements can be used to estimate the ratio of the two odds.

```
data jackpot;
  input degree $ game :$9. response $ count @@;
  datalines;
MS blackjack winner 78 MS blackjack loser 28
MS poker winner 101 MS poker loser 11
MS slots winner 68 MS slots loser 46
PhD blackjack winner 40 PhD blackjack loser 95
PhD poker winner 54 PhD poker loser 5
PhD slots winner 34 PhD slots loser 6
;
run;
proc logistic data=jackpot;
  freq count;
  class degree game / param=glm;
  model response(event="loser") = degree game degree*game;
  estimate 'blackjack master' int 1 degree 1 game 1 degree*game 1 / ilink exp;
  estimate 'slots master' int 1 degree 1 game 0 0 1
    degree*game 0 0 1 / ilink exp;
  contrast 'blackjack v slots master' game 1 0 -1
    degree*game 1 0 -1 0 0 0 /estimate=exp;
  lsmestimate degree*game 'blackjack v slots master' 1 0 -1,
    'nonpos blackjack v slots master'
    [1,1 1] [-1, 1 3]/exp cl;
run;
```

The following table provides a partial listing of the PROC LOGISTIC results for the previous program code:

Type 3 Analysis of Effects			
Effect	DF	Wald Chi-Square	Pr > ChiSq
degree	1	0.2312	0.6306
game	2	52.4352	<.0001
degree*game	2	36.7575	<.0001

Notice that the TYPE 3 Analysis of Effects results indicate a significant effect for game and degree*game interaction (p -value <0.0001). It also indicates that the main effect of degree is not significant (p -value 0.6306).

The result of the first ESTIMATE statement, shown in the following table, indicates that the probability of an MS statistician being a loser at blackjack is 0.2642 and that the odds of losing are $0.2642/(1-0.2642)=0.3591$. It is the ILINK option that provides the mean (probability) estimate and that estimate (-1.0245) is the log odds; therefore, $\exp(-1.0245)=0.3591$. The expectation is only 0.3591 losers for every winner, or 2.7850 (1/0.3591) winners for every loser.

Estimate							
Label	Estimate	Standard Error	z Value	Pr > z	Mean	Standard Error of Mean	Exponentiated
blackjack master	-1.0245	0.2203	-4.65	<.0001	0.2642	0.04282	0.3590

The result of the second ESTIMATE statement, shown below, indicates that the probability of an MS statistician being a loser at slots is 0.4035, yielding the odds of losing at $0.4035/(1-0.4035)=0.6764$. The expectation here is 0.6764 losers for every winner, or 1.4783 winners for every loser. Note that the ratio of odds is $0.3951/0.6764 \approx 0.5307$, which is found by the CONTRAST and LSMESTIMATE statements.

Estimate							
Label	Estimate	Standard Error	z Value	Pr > z	Mean	Standard Error of Mean	Exponentiated
slots master	-0.3909	0.1909	-2.05	0.0406	0.4035	0.04595	0.6765

The CONTRAST and LSMESTIMATE statements both estimate the odds ratio. As shown in the exponentiated estimate in the following tables, an MS statistician is 0.5307 times more likely to lose at blackjack than at slots. These results indicate that for those statisticians with a master's degree, the probability of being a loser is significantly different between blackjack and slots.

Contrast Test Results			
Contrast	DF	Wald Chi-Square	Pr > ChiSq
blackjack v slots master	1	4.7246	0.0297

Contrast Estimation and Testing Results by Row									
Contrast	Type	Row	Estimate	Standard Error	Alpha	Confidence Limits		Wald Chi-Square	Pr > ChiSq
blackjack v slots master	EXP	1	0.5307	0.1547	0.05	0.2997	0.9396	4.7246	0.0297

continued

Least Squares Means Estimates											
Effect	Label	Estimate	Standard Error	z Value	Pr > z	Alpha	Lower	Upper	Exponentiated		
									Estimate	Lower	Upper
degree* game	blackjack v slots master	-0.6336	0.2915	-2.17	0.0297	0.05	-1.2050	-0.06228	0.5307	0.2997	0.9396
degree* game	non-pos blackjack v slots master	-0.6336	0.2915	-2.17	0.0297	0.05	-1.2050	-0.06228	0.5307	0.2997	0.9396

The new SLICE statement in PROC LOGISTIC makes the comparison easier to program. When you have classification variables in your model, PROC LOGISTIC allows the SLICE statement only if you also specify the PARAM=GLM option. The following example demonstrates the use of the SLICE statement:

```
proc logistic data=jackpot;
  freq count;
  class degree game / param=glm;
  model response(event="loser")=degree game degree*game;
  slice degree*game/sliceby=degree diff exp cl;
run;
```

The following results show how the games compare to one another, for each level of educational degree. In addition to the game comparisons for the MS group, the results provide the game comparisons for the PhD group.

Chi-Square Test for degree*game Least Squares Means Slice			
Slice	Num DF	Chi-Square	Pr > ChiSq
degree MS	2	24.60	<.0001

Simple Differences of degree*game Least Squares Means												
Slice	game	_game	Estimate	Standard Error	z Value	Pr > z	Alpha	Lower	Upper	Exponentiated		
										Estimate	Lower	Upper
degree MS	black jack	poker	1.1927	0.3865	3.09	0.0020	0.05	0.4353	1.9502	3.2960	1.5454	7.0297
degree MS	black jack	slots	-0.6336	0.2915	-2.17	0.0297	0.05	-1.2050	-0.06228	0.5307	0.2997	0.9396
degree MS	poker	slots	-1.8264	0.3705	-4.93	<.0001	0.05	-2.5525	-1.1002	0.1610	0.07789	0.3328

Chi-Square Test for degree*game Least Squares Means Slice			
Slice	Num DF	Chi-Square	Pr > ChiSq
degree PhD	2	61.76	<.0001

Simple Differences of degree*game Least Squares Means												
Slice	game	_game	Estimate	Standard Error	z Value	Pr > z	Alpha	Lower	Upper	Exponentiated		
										Estimate	Lower	Upper
degree PhD	black jack	poker	3.2445	0.5040	6.44	<.0001	0.05	2.2567	4.2324	25.6498	9.5511	68.8830
degree PhD	black jack	slots	2.5996	0.4813	5.40	<.0001	0.05	1.6564	3.5428	13.4583	5.2402	34.5649
degree PhD	poker	slots	-0.6449	0.6439	-1.00	0.3165	0.05	-1.9069	0.6171	0.5247	0.1485	1.8535

For the statisticians with a master's degree, the results indicate that the log odds of losing differ significantly within each pair of games. The odds of losing at blackjack are 3.296 times the odds of losing at poker. The odds of being a loser for poker are 0.1610 of the odds for slots. For the statisticians with a PhD, the log odds of losing at blackjack differ significantly from the log odds of losing at either poker or slots

CONCLUSION

This paper describes several of the new post-fitting analysis statements and options that are available in SAS/STAT 9.22 software. The paper also provides examples to demonstrate methods and techniques that you can use to test custom hypotheses using the ESTIMATE or LSMESTIMATE statements. The LSMESTIMATE statement combines important and convenient features of both the LSMEANS and ESTIMATE statements. These features enable you to specify coefficients in a more straightforward and intuitive way, in terms of the LS-means rather than the model parameters.

For additional examples of how to write CONTRAST, ESTIMATE, or LSMESTIMATE statements, see SAS Note 24447, "Examples of writing proper CONTRAST, ESTIMATE, and LSMESTIMATE statements." (SAS Institute Inc. 2005). This note is a tutorial that provides several examples that will help you in writing these statements properly.

REFERENCES

- Kutner, M. H. 1974. "Hypothesis Testing in Linear Models (Eisenhart Model)." *The American Statistician* 28:98-100.
- SAS Institute Inc. 2005. SAS Note 24447, "Examples of writing proper CONTRAST, ESTIMATE, and LSMESTIMATE statements." Cary, NC: SAS Institute Inc. Available at support.sas.com/kb/24/447.html.
- Tobias, R. and Cai, W. 2010. "Introducing PROC PLM and Postfitting Analysis for Very General Linear Models in SAS/STAT® 9.22." *Proceedings of the SAS Global Forum 2010 Conference*, Cary, NC: SAS Institute Inc. Available at support.sas.com/resources/papers/proceedings10/258-2010.pdf.
- Westfall, P.H.; Tobias, R.; and Wolfinger, R. 2011. *Multiple Comparisons and Multiple Tests Using the SASⁱ System, Second Edition.*, Cary, NC: SAS Institute Inc. (Anticipated publication date is summer 2011.)
- Winer, B. J. 1971. *Statistical Principles in Experimental Design, Second Edition*, New York: McGraw-Hill.

CONTACT INFORMATION

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

Kathleen Kiernan
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
E-mail: support@sas.com
Web: support.sas.com

Randy Tobias
SAS Institute Inc.
Research Drive
Cary, NC 27513
E-mail: Randy.Tobias@sas.com
Web: www.sas.com

Phil Gibbs
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
E-mail: support@sas.com
Web: support.sas.com

Jill Tao
SAS Institute Inc.
SAS Campus Drive
Cary, NC 27513
E-mail: support@sas.com
Web: support.sas.com

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.