Paper 269-2010

Practicalities of Using ESTIMATE and CONTRAST Statements

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

ABSTRACT

One of the tasks that many analysts find daunting is the construction of ESTIMATE and CONTRAST statements. Admittedly, these can get very complicated, but if you start with simple examples and work your way to more complex situations it can help demystify the process. This paper provides practical guidance on how to code ESTIMATE and CONTRAST statements, especially when the group sizes are unequal. It includes an example of how to mimic the effect of the OBSMARGINS option on LSMEANS by using ESTIMATE statements. It also illustrates the use of the SYMPUT function to obtain the necessary ESTIMATE coefficients from the data "automatically." Concrete examples are used to help you learn many of the principles you need to successfully create ESTIMATE and CONTRAST statements. Some familiarity with linear models is essential, but you do not need experience with any specific SAS® procedure. The emphasis is on the practical, not the theoretical: no Greek letters!

INTRODUCTION

A situation that arises frequently in our work is that what starts as a reasonably straightforward linear statistical model ends up getting more and more complicated with more and more specific comparisons of interest. This frequently leads to a large number of ESTIMATE or CONTRAST statements to calculate specific values with associated standard errors, confidence intervals, or *P* values. The construction of those ESTIMATE and CONTRAST statements are sometimes somewhat tricky and many programmers find them confusing. This paper presents concrete examples of the construction of ESTIMATE and CONTRAST statements, building up to a complex example that uses a stratified piecewise linear model.

We will begin by describing the scientific background for the statistical model and presenting the final result we are trying to achieve. Next we will go "back to basics" and consider some fairly simple examples of ESTIMATE and CONTRAST statements and then some more complex examples. Along the way we will illustrate the effect of the OBSMARGINS option on LSMEANS and how to mimic the effect by using ESTIMATE statements. Then we will show the ESTIMATE and CONTRAST statements used in the complex model and show how to use the SYMPUT function to automate the calculation of the ESTIMATE coefficients from the data.

SCIENTIFIC BACKGROUND

The complex model used as our example arose in connection with a study of cystic fibrosis. Cystic fibrosis is a hereditary disease that leads to long term decline in lung function. One measure of lung function is FEV_1, the Forced Expiratory Volume in 1 second, which is obtained during a pulmonary function test (PFT). Because the volume of air that can be expelled in 1 second varies considerably based on the size of the lung, it is common in CF to calculate a "% predicted" measure that is based on the sex, age, height, and race/ethnicity of the patient. This relates the absolute FEV_1 to the expected (mean) value based on the patient's characteristics, expressed as a percentage. It is common practice to track FEV_1 % predicted over time to document patients' lung function decline. There are some disadvantages to modeling the % predicted values (rather than z-scores or other alternative measures), but this approach models the measures most commonly used clinically.

Consider choosing an arbitrary point in time and assessing the lung function of a cystic fibrosis patient, as measured by FEV_1 % predicted, before and after that index time. At first thought, it may seem that there is no reason to believe the rate of decline would be any different before and after the arbitrary index time. However, previous research has shown that high lung function is an independent risk factor for decline. Therefore, patients with higher than average lung function are expected to experience a steeper than average decline going forward and patients with lower lung function are expected to experience a less steep decline going forward. Furthermore, it stands to reason that patients with relatively high lung function at that index time are likely to have had more gradual prior decline than patients with relatively low lung function. (This is a sort of regression to the mean effect looking backwards in time.) These two factors combine to produce the expectation that patients with relatively high lung function at the index time are likely to show a change from mild decline to steeper decline, whereas those with relatively low

lung function are likely to show a change from steep decline to milder decline. Thus, the null hypothesis of no change in average decline before and after an arbitrary index time may need to be adjusted depending on the measured lung function at that index time.

THE STATISTICAL MODEL

In the statistical model, we wanted to quantify the average rates of decline in FEV_1 % predicted before and after an index time, separately by severity group. An index pulmonary function test was defined as the PFT closest (within 30 days) to the first encounter within one year following the eighth or subsequent even numbered birthday. (Even numbered birthdays were used to avoid having overlapping pre-index periods.) The pre-index and post-index periods – each 2 years in duration – were each required to have ≥1 encounter and ≥3 FEV_1 values spanning at least six months to estimate the slope of FEV_1. Patients were included for as many sets of pre-index and post-index periods as they had available data.

When we characterized severity using FEV_1 % predicted values, we ran into the difficulty that there were few younger patients in the most severe categories and few older patients in the least severe categories. To provide for a more balanced distribution across categories by age, we characterized lung function relative to other CF patients at every age from 8 to 38 years using all PFTs in the dataset to establish age-specific deciles of FEV_1 % predicted.

For every patient and index value, separate regression lines were fit during each of the two-year pre-index and post-index periods. The index PFT was used to establish the age-adjusted decile of severity, but was excluded from both the pre-index and post-index periods to minimize issues associated with regression to the mean. The regression lines were fit using PROC MIXED in SAS® with four random effects: intercept (at the index PFT) and slope before the index event, and change in intercept and change in slope after the index event.

THE RESULTS

Figure 1 shows the average pre-index and post-index fitted lines by decile; Table 1 provides the details.

Figure 1: Pre- and post-index slopes and increment at index event by FEV_1 decile

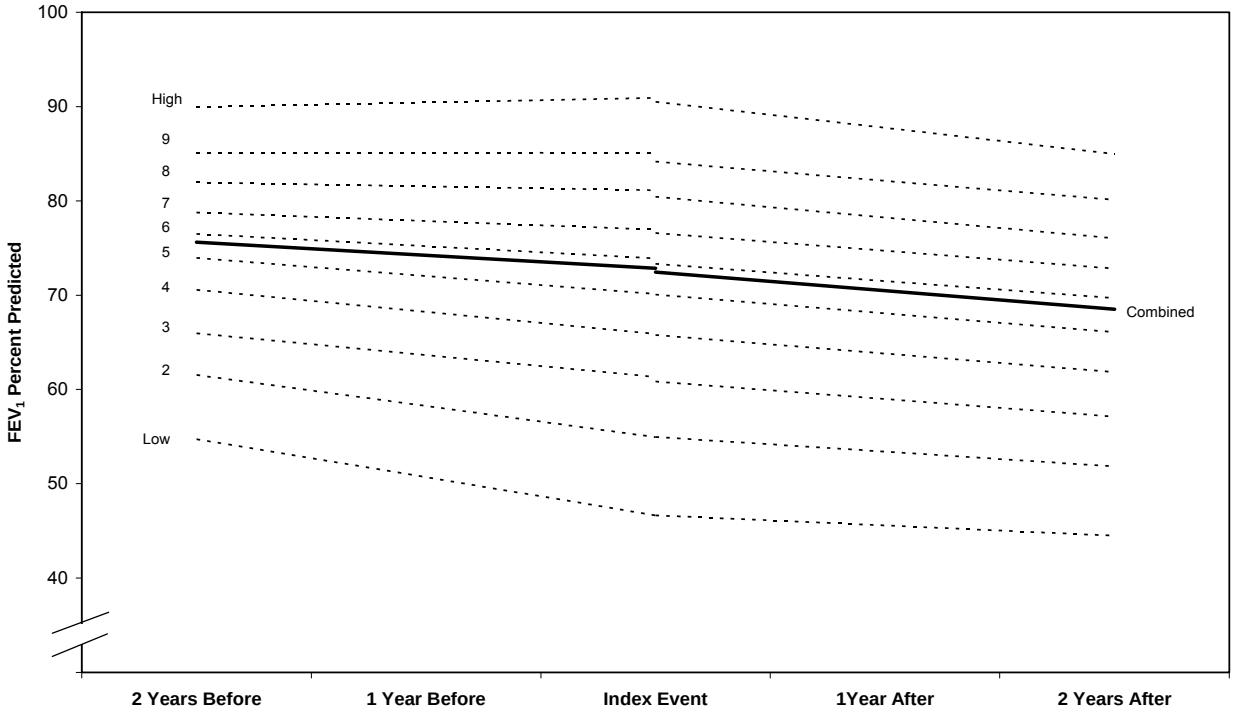

Table 1: Details of pre- and post-index slopes and increment at index event by FEV_1 decile and combined

Decile	N	Pre-index Slope (SE)	Post-index Slope (SE)	Slope Difference (SE)	P Difference	Post-index Increase (SE)	P Increase	Pre-index Start	Pre-index Stop	Post-index Start	Post-index Stop
Combined (observed)	32355	-1.38 (0.05)	-1.98 (0.04)	-0.60 (0.06)	<.001	-0.40 (0.06)	<.001	75.61	72.85	72.45	68.50
Combined (uniform)	32355	-1.60 (0.05)	-1.92 (0.04)	-0.31 (0.06)	<.001	-0.36 (0.06)	<.001	73.91	70.70	70.33	66.50
1	2155	-4.05 (0.16)	-1.07 (0.16)	2.97 (0.23)	<.001	0.00 (0.21)	1.00	54.74	46.65	46.65	44.50
2	2511	-3.29 (0.15)	-1.56 (0.15)	1.73 (0.21)	<.001	0.01 (0.20)	0.95	61.53	54.95	54.96	51.84
3	2874	-2.33 (0.15)	-1.86 (0.14)	0.47 (0.20)	0.021	-0.47 (0.19)	0.014	65.97	61.32	60.85	57.12
4	3091	-2.35 (0.14)	-1.97 (0.13)	0.38 (0.20)	0.057	-0.10 (0.19)	0.59	70.57	65.88	65.78	61.85
5	3292	-1.87 (0.14)	-2.00 (0.13)	-0.13 (0.20)	0.52	-0.14 (0.19)	0.46	73.95	70.21	70.07	66.08
6	3428	-1.30 (0.14)	-1.82 (0.13)	-0.52 (0.19)	0.007	-0.58 (0.19)	0.002	76.49	73.90	73.32	69.68
7	3639	-0.90 (0.14)	-1.90 (0.13)	-1.00 (0.19)	<.001	-0.36 (0.19)	0.059	78.76	76.96	76.60	72.81
8	3768	-0.43 (0.14)	-2.20 (0.13)	-1.77 (0.19)	<.001	-0.69 (0.19)	<.001	81.97	81.12	80.43	76.03
9	3782	-0.02 (0.14)	-2.03 (0.13)	-2.01 (0.19)	<.001	-0.92 (0.19)	<.001	85.13	85.08	84.16	80.10
10	3815	0.49 (0.14)	-2.77 (0.14)	-3.26 (0.20)	<.001	-0.40 (0.20)	0.040	89.94	90.92	90.51	84.98

In addition to estimating the average lines by decile, an overall estimate was obtained by combining the deciles using equal weighting (each decile counted equally) and the observed distribution (each decile counted according to the number of patients represented). These two ways of combining the deciles differ because the number of patients with available data varied by decile; the figure presents the version based on the observed distribution.

The results show the anticipated "bowing." The middle deciles have similar slope pre- and post-index with little change in intercept. For the lower deciles, the pre-index slopes are fairly steep compared to the post-index slopes, which are fairly flat. The opposite is the case for the higher deciles, where the pre-index slopes are fairly flat and the post-index slopes are fairly steep. The differences in estimated intercept are an indication that the straight lines do not adequately fit what is presumably a curved trajectory. Although it is reasonable to approximate the rate of change over short times using a straight line, fitting straight lines to up to two years of data may be more problematic. The more curved the true underlying trends, the more likely there is to be an observed difference in intercept when straight lines are fit in the two time periods.

Table 1 includes many different values, some with standard errors and some with *P* values, most of which came from ESTIMATE statements. Let's go back to basics and work up to how to produce Table 1.

ESTIMATE STATEMENT BASICS

The idea behind the ESTIMATE statement is that you want to calculate the value of a linear combination of parameters and (generally) the associated standard error, confidence bounds, or *P* value for testing whether it differs from zero. If the model is of full rank, as would be typical in the context of a regression, each individual parameter is uniquely estimated and any linear combination of the parameters can also be estimated. When the model includes categorical variables (variables listed on the CLASS statement), as is typical in the context of analysis of variance (ANOVA), the model may be parameterized to be less

than full rank. As a simple example, consider a variable Sex that takes on two values: Female and Male. If you include Sex in the CLASS statement and estimate a simple model Y=Sex using, for example, the GLM or MIXED procedure, SAS will not estimate an overall intercept and a value for Female but set the value of the estimated parameter for Male to 0. The estimated value for Males is determined by adding the Intercept and the coefficient for Males; the estimated value for Females is determined by adding the Intercept and the coefficient for Females. The best fit model in this case is the group mean, so it turns out the Intercept is the mean for Males and the coefficient for Females is actually the difference (Group mean for Females minus Group mean for Males). The t-test that the coefficient for Females is nonzero is the same as you would get from PROC TTEST or from LSMEANS in GLM. It is important to understand this simple example thoroughly, as it illustrates a number of principles. We can code some simple ESTIMATE statements can confirm those calculations:

CODE FOR GLM:

```
proc glm data=anal;
class sex;
model y1 = sex / solution;
lsmeans sex / stderr tdiff e;
estimate 'male' intercept 1 sex 0 1;
estimate 'female' intercept 1 sex 1 0;
estimate 'female-male' sex 1 -1;
title3 'GLM by sex';
run;
```

The GLM procedure makes it relatively easy to see what is going on. By specifying / SOLUTION on the MODEL statement, the parameter estimates are provided (along with standard errors and the t-tests that the parameters are zero). This is almost always useful, if only to make sure the model is being estimated as you expect; I essentially always specify the SOLUTION option. The LSMEANS statement requests the Least Squares Means for the specified terms in the model. In this case, these are the same as the ordinary means. The STDERR option requests the standard errors of each least squares mean, the TDIFF option asks for t-tests on all possible pairwise differences among levels of the effects specified on the LSMEANS statement, and the E option requests the coefficients used to calculate the least squares mean. In this case, the result is rather simple, but in more complicated situations this is very handy.

The syntax for an ESTIMATE statement is a name in quotes (technically optional but essential for identifying the estimate on the output) and then one or more effects from the model each followed by coefficients. The intercept is an implied effect in every model. So the interpretation of the first ESTIMATE statement is to multiply 1 times the intercept and 0 times the first coefficient for SEX and 1 times the second coefficient for SEX and add them up. In this case the ESTIMATE statements mimic the information obtained from LSMEANS in this case but are a good test of understanding simple ESTIMATE statements. The final 0 on the 'female' ESTIMATE is optional: trailing zeroes are assumed. I think it is useful to include them to emphasize the number of coefficients associated with the effect.

Here are excerpts from output from MEANS, TTEST, and GLM:

FROM PROC MEANS:

sex	N Obs	N	Mean	Std Dev	Std Error
Female	326	326	65.4034956	37.5623024	2.0803835
Male	674	674	68.5128249	36.2429821	1.3960275

FROM PROC TTEST:

TTEST by sex

Variable	sex	N	Mean	Std Dev	Std Err
y1	Female	326	65.403	37.562	2.0804
y1	Male	674	68.513	36.243	1.396
y1	Diff (1-2)		-3.109	36.678	2.4744

```
                              T-Tests
Variable    Method           Variances      DF    t Value    Pr > |t|
y1          Pooled           Equal         998      -1.26      0.2092
y1          Satterthwaite    Unequal       623      -1.24      0.2150
```

FROM PROC GLM:

```
                                     Sum of
Source                       DF         Squares     Mean Square    F Value    Pr > F
Model                         1        2124.276        2124.276       1.58    0.2092
Error                       998     1342572.807        1345.263
Corrected Total             999     1344697.083

Source                       DF     Type III SS     Mean Square    F Value    Pr > F
sex                           1     2124.275971     2124.275971       1.58    0.2092

                                                Standard
Parameter                 Estimate                 Error    t Value    Pr > |t|
Intercept              68.51282491 B      1.41277729      48.50      <.0001
sex       Female       -3.10932930 B      2.47437150      -1.26      0.2092
sex       Male          0.00000000 B       .                .         .

NOTE: 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.

Least Squares Means
       Coefficients for sex Least Square Means
                                        sex Level
Effect                                  Female    Male
Intercept                                    1       1
sex       Female                             1       0
sex       Male                               0       1

                              Standard    H0:LSMEAN=0     H0:LSMean1=LSMean2
sex          y1 LSMEAN           Error       Pr > |t|    t Value    Pr > |t|
Female      65.4034956       2.0313972         <.0001      -1.26      0.2092
Male        68.5128249       1.4127773         <.0001

Standard
Parameter                  Estimate           Error    t Value    Pr > |t|
male                    68.5128249      1.41277729      48.50      <.0001
female                  65.4034956      2.03139721      32.20      <.0001
female-male             -3.1093293      2.47437150      -1.26      0.2092
```

In this simple example, the means, standard deviations, standard errors, t-statistics, and *P* values all match across the various ways of obtaining them. This gives us confidence to try a harder example.

ESTIMATE STATEMENTS WITH TWO FACTORS

Once you go beyond a single factor, things get more complicated. Among other things, the question arises whether to use the OBSMARGINS option (abbreviated OM) on LSMEANS. This option causes the LSMEANS to use the observed marginal distributions 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.

Let's consider an example with two categorical variables, race (with five levels) and sex (with two levels). In order to get a handle on estimability, we can ask for the general form of all estimable functions by include the E option on the MODEL statement.To illustrate the difference, we include one LSMEANS statement with the OM option and one without:

CODE FOR GLM:

```
proc glm data=anal;
class race sex;
model y1 = race sex / solution e;
lsmeans race sex / stderr tdiff e;
lsmeans race sex / stderr tdiff e om;
estimate 'male' intercept 1 sex 0 1;
estimate 'female' intercept 1 sex 1 0;
estimate 'female-male' sex 1 -1;
title3 'GLM by race sex';
run;
```

FROM PROC GLM:

```
GLM by race sex
                Class Level Information
Class         Levels    Values
race               5    Asian Black Hispanic Other White
sex                2    Female Male

Number of Observations Read        1000
Number of Observations Used        1000

General Form of Estimable Functions
Effect                 Coefficients
Intercept              L1
race      Asian        L2
race      Black        L3
race      Hispanic     L4
race      Other        L5
race      White        L1-L2-L3-L4-L5
sex       Female       L7
sex       Male         L1-L7
```

Notice that the levels of RACE are in alphabetic order by formatted value. The interpretation of the "General Form of Estimable Functions," obtained by specifying the E option on the MODEL statement, is that the coefficients given as L followed by a number can be assigned any numerical value, but that the coefficient for some effects are derivable from the others. For example, if L2 is assigned a 1 and all the other coefficients are assigned 0, then for the function to be estimable the coefficient for White would need to be -1. This would then estimate the difference between Asian and White. If you wanted just the value for Asian, you could assign L1 and L2 a value of 1, and the estimate would be of the Intercept plus Asian (and the coefficient for White would be 1-1=0).

```
 Dependent Variable: y1
                                      Sum of
Source                     DF         Squares     Mean Square    F Value    Pr > F
Model                       5       82152.751       16430.550      12.94    <.0001
Error                     994     1262544.332        1270.165
Corrected Total           999     1344697.083

Source                     DF     Type III SS     Mean Square    F Value    Pr > F
race                        4     80028.47498     20007.11875      15.75    <.0001
sex                         1      2590.81573      2590.81573       2.04    0.1535
```

Parameter		Estimate		Standard Error	t Value	Pr > |t|
Intercept		63.39550656	B	1.65404269	38.33	<.0001
race	Asian	0.96269902	B	3.79755107	0.25	0.7999
race	Black	15.72610471	B	3.33992898	4.71	<.0001
race	Hispanic	23.67301601	B	3.48135125	6.80	<.0001
race	Other	-2.94225607	B	5.41062521	-0.54	0.5867
race	White	0.00000000	B	.	.	.
sex	Female	-3.43990976	B	2.40856801	-1.43	0.1535
sex	Male	0.00000000	B	.	.	.

NOTE: 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.

The SOLUTION, given above, includes tests of the difference between each race and the White category (the last category). These are more attractively displayed using the LSMEANS and the TDIFF option. For more information about SOLUTION, see Usage Note *38384:* How to interpret the results of the SOLUTION option in the MODEL statement of PROC GLM? at //support.sas.com/notes/index.html.

The first LSMEANS are *without* the OM option:

Least Squares Means

Coefficients for race Least Square Means

Effect		race Level Asian	Black	Hispanic	Other	White
Intercept		1	1	1	1	1
race	Asian	1	0	0	0	0
race	Black	0	1	0	0	0
race	Hispanic	0	0	1	0	0
race	Other	0	0	0	1	0
race	White	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

race	y1 LSMEAN	Standard Error	Pr > |t|	LSMEAN Number
Asian	62.6382507	3.5119521	<.0001	1
Black	77.4016564	3.0099679	<.0001	2
Hispanic	85.3485677	3.1771906	<.0001	3
Other	58.7332956	5.2036501	<.0001	4
White	61.6755517	1.5532976	<.0001	5

Least Squares Means for Effect race
t for H0: LSMean(i)=LSMean(j) / Pr > |t|
Dependent Variable: y1

i/j	1	2	3	4	5
1		-3.20959	-4.82644	0.623286	0.253505
		0.0014	<.0001	0.5332	0.7999
2	3.209586		-1.82924	3.112197	4.708515
	0.0014		0.0677	0.0019	<.0001
3	4.826444	1.829242		4.376613	6.79995
	<.0001	0.0677		<.0001	<.0001
4	-0.62329	-3.1122	-4.37661		-0.54379
	0.5332	0.0019	<.0001		0.5867
5	-0.25351	-4.70851	-6.79995	0.543792	
	0.7999	<.0001	<.0001	0.5867	

NOTE: To ensure overall protection level, only probabilities associated with pre-planned comparisons should be used.

```
Least Squares Means
       Coefficients for sex Least Square Means
                                        sex Level
Effect                                  Female    Male
Intercept                                    1       1
race      Asian                            0.2     0.2
race      Black                            0.2     0.2
race      Hispanic                         0.2     0.2
race      Other                            0.2     0.2
race      White                            0.2     0.2
sex       Female                             1       0
sex       Male                               0       1

                              Standard    H0:LSMEAN=0     H0:LSMean1=LSMean2
sex          y1 LSMEAN           Error       Pr > |t|    t Value    Pr > |t|
Female      67.4395095       2.2041923         <.0001      -1.43      0.1535
Male        70.8794193       1.7677880         <.0001
```

The second LSMEANS are *with* the OM option:

```
Least Squares Means
                    Coefficients for race Least Square Means
                                        race Level
Effect                                  Asian    Black    Hispanic    Other    White

Intercept                                   1        1           1        1        1
race      Asian                             1        0           0        0        0
race      Black                             0        1           0        0        0
race      Hispanic                          0        0           1        0        0
race      Other                             0        0           0        1        0
race      White                             0        0           0        0        1
sex       Female                        0.326    0.326       0.326    0.326    0.326
sex       Male                          0.674    0.674       0.674    0.674    0.674

                               Standard                   LSMEAN
race           y1 LSMEAN           Error    Pr > |t|      Number
Asian         63.2367950       3.4954641      <.0001           1
Black         78.0002007       2.9918561      <.0001           2
Hispanic      85.9471120       3.1501100      <.0001           3
Other         59.3318399       5.2019535      <.0001           4
White         62.2740960       1.4819289      <.0001           5

                  Least Squares Means for Effect race
                 t for H0: LSMean(i)=LSMean(j) / Pr > |t|
                         Dependent Variable: y1
i/j              1             2             3             4             5
   1                     -3.20959      -4.82644      0.623286      0.253505
                           0.0014        <.0001        0.5332        0.7999
   2       3.209586                    -1.82924      3.112197      4.708515
             0.0014                      0.0677        0.0019        <.0001
   3       4.826444      1.829242                    4.376613       6.79995
             <.0001        0.0677                      <.0001        <.0001
   4       -0.62329       -3.1122      -4.37661                    -0.54379
             0.5332        0.0019        <.0001                      0.5867
   5       -0.25351      -4.70851      -6.79995      0.543792
             0.7999        <.0001        <.0001        0.5867


NOTE: To ensure overall protection level, only probabilities associated with pre-planned
comparisons should be used.
```

```
Least Squares Means

       Coefficients for sex Least Square Means
                                        sex Level
Effect                                  Female     Male

Intercept                                    1        1
race      Asian                          0.104    0.104
race      Black                          0.142    0.142
race      Hispanic                       0.128    0.128
race      Other                          0.047    0.047
race      White                          0.579    0.579
sex       Female                             1        0
sex       Male                               0        1

                              Standard    H0:LSMEAN=0     H0:LSMean1=LSMean2
sex          y1 LSMEAN           Error       Pr > |t|     t Value     Pr > |t|
Female      65.1806844       1.9762366         <.0001       -1.43      0.1535
Male        68.6205941       1.3735697         <.0001
```

The ESTIMATE statements are designed to mimic the first set of LSMEANS for SEX:

```
                                          Standard
Parameter                  Estimate          Error    t Value    Pr > |t|
male                     70.8794193     1.76778797      40.09      <.0001
female                   67.4395095     2.20419228      30.60      <.0001
female-male              -3.4399098     2.40856801      -1.43      0.1535
```

The difference between the two LSMEANS are that the first assumes an equal distribution across the categories of the variables, whereas the second uses the observed marginal distribution. In this context, when comparing males and females we can either assume the five levels of the RACE variable each have 20% of the population (the default) or we can use the actual distribution of the RACE values (the OM option). In this constructed example, we get the same answer for the difference 'female-male' but rather different values for the least squares means. In general, both the least squares means and their differences will change when OM is specified.

ESTIMATE STATEMENTS WITH TWO FACTORS AND INTERACTION

Here is an example with a two-way interaction and therefore a somewhat more complicated ESTIMATE statement:

CODE FOR GLM:

```
proc glm data=anal;
class sex race;
model y1 = race sex race*sex / solution;
lsmeans race sex / stderr e;
lsmeans race sex / stderr e om;
estimate 'male'   intercept 1 sex 0 1 race .2 .2 .2 .2 .2  race*sex 0 0 0 0 0  .2 .2 .2 .2 .2;
estimate 'female' intercept 1 sex 1 0 race .2 .2 .2 .2 .2  race*sex .2 .2 .2 .2 .2  0 0 0 0 0;
estimate 'female-male' sex 1 -1 race*sex .2 .2 .2 .2 .2  -.2 -.2 -.2 -.2 -.2;
title3 'GLM by race sex race*sex';
run;
```

It turns out in this example, the OM version of the least squares means are nonestimable. This is because the observed proportion of males varies by race (or, to put it another way, the distribution across race is different for the two sexes). This makes the coefficients inconsistent. The ESTIMATE statements mimic the least squares means for the sex variable without the OM option:

```
                              Standard
sex          y1 LSMEAN           Error    Pr > |t|
Female      68.1117839       2.5611683      <.0001
Male        70.5239982       1.9646266      <.0001

                                          Standard
Parameter                    Estimate           Error    t Value    Pr > |t|
male                     70.5239982      1.96462660      35.90      <.0001
female                   68.1117839      2.56116830      26.59      <.0001
female-male              -2.4122143      3.22790036      -0.75      0.4551
```

Any consistent distribution for the other variables can be used to calculate least squares means – you are not limited to the uniform distribution or the observed distribution. For example, we might want to use the distribution over the RACE variable from some reference population. In this case, it might make sense to use the distribution used to generate the simulated data. Or, we might want to use the observed overall distribution of RACE (not separately by sex) instead. Here's how we might code the ESTIMATE statements and the associated output from GLM:

CODE FOR GLM:

```
proc glm data=anal;
class sex race;
model y1 = race sex race*sex / solution e;
estimate 'male (h)' intercept 1  sex 0 1  race  .10 .15 .15 .05 .55
          race*sex 0 0 0 0 0  .10 .15 .15 .05 .55;
estimate 'female (h)' intercept 1  sex 1 0 race  .10 .15 .15 .05 .55
          race*sex .10 .15 .15 .05 .55  0 0 0 0 0;
estimate 'female-male (h)' sex 1 -1 race*sex .10 .15 .15 .05 .55
          -.10 -.15 -.15 -.05 -.55;
estimate 'male (o)' intercept 1  sex 0 1  race  .104 .142 .128 .047 .579
          race*sex 0 0 0 0 0  .104 .142 .128 .047 .579;
estimate 'female (o)' intercept 1  sex 1 0 race  .104 .142 .128 .047 .579
          race*sex .104 .142 .128 .047 .579  0 0 0 0 0;
estimate 'female-male (o)' sex 1 -1 race*sex .104 .142 .128 .047 .579
          -.104 -.142 -.128 -.047 -.579;
estimate 'female-male (off)' sex 1 -1 race*sex .105 .142 .128 .047 .579
          -.105 -.142 -.128 -.047 -.579;
title3 'GLM by race sex race*sex with hypothetical and observed race weights';
run;
```

Notice the last ESTIMATE statement, with the difference marked "(off)" in the label. That ESTIMATE statement is nonestimable because the coefficients add up to 1.001 instead of 1.000 within each sex. It is easy to end up with values, even with many decimal places specified, that are off just a bit and which are therefore reported as nonestimable. One solution is to adjust the values slightly to make sure they add up to 1. Another solution is to use the DIVISOR option to ensure the values add up, as illustrated in the next sections. Here are the results from GLM:

```
                                          Standard
Parameter                    Estimate           Error    t Value    Pr > |t|

male (h)                 69.2219208      1.38231738      50.08      <.0001
female (h)               65.7862376      1.98334675      33.17      <.0001
female-male (h)          -3.4356832      2.41753297      -1.42      0.1556
male (o)                 68.6049088      1.37605701      49.86      <.0001
female (o)               65.1115576      1.98344075      32.83      <.0001
female-male (o)          -3.4933513      2.41403606      -1.45      0.1482
```

Notice that we get somewhat different estimates of the female-male difference depending on how we weight the levels of RACE (and both differ from the default values in the previous output).

TESTING FOR LINEAR TREND

One use of the ESTIMATE statement is to assess a linear trend across levels of a categorical variable. These can be equally spaced or unequally spaced. The DIVISOR option is used to specify a number that each coefficient is divided by.

CODE FOR GLM:

```
proc glm data=anal;
class sex educat;
model y1 = sex educat / solution;
lsmeans sex educat / stderr e;
lsmeans sex educat / stderr e om;
estimate 'educat linear (no divisor)' educat -2 -1 0 1 2;
estimate 'educat linear (divisor=10)' educat -2 -1 0 1 2 / divisor=10;
estimate 'educat linear (divisor=40)' educat -4 -2 0 2 4 / divisor=40;
estimate 'educat by year (not centered)' educat 8 12 14 16 20;
estimate 'educat by year (centered)' educat -6 -2 0 2 6 / divisor=80;
title3 'GLM by sex educat';
run;
```

It turns out that not centering the coefficents results in a nonestimable function:

Parameter	Estimate	Standard Error	t Value	Pr > |t|
educat linear (no divisor)	55.7056111	10.3314131	5.39	<.0001
educat linear (divisor=10)	5.5705611	1.0331413	5.39	<.0001
educat linear (divisor=40)	2.7852806	0.5165707	5.39	<.0001
educat by year (centered)	1.8542778	0.3765941	4.92	<.0001

Perhaps the hypothesis test is of greater interest in this context than the estimated parameter. If you are interested in the estimated slope the coefficients should be normalized by dividing by the sum of squares (so the Euclidean length of the vector of parameters is 1). This example shows that changing the divisor or multiplying all the coefficients by the same value changes the estimate and standard error proportionally and the t-statistic is unchanged. Alternatively, unequally-spaced values can be used.

USE OF COUNTS AND THE DIVISOR OPTION

It is common for there to be changes in the analyses, for example deciding to omit a few subjects or a whole group. This can mean redoing all the carefully calculated and transcribed coefficients. A separate problem is making sure that the coefficients add exactly to 1. Both of these can be helped by using counts of subjects and the DIVISOR option on the ESTIMATE statement. Here's an example where it was decided to omit the Asian and Other categories of RACE:

CODE FOR GLM:

```
proc glm data=anal;
class sex race;
model y1 = race sex race*sex / solution e;
estimate 'male (333)' intercept 1  sex 0 1  race  .333 .333 .333
          race*sex 0 0 0  .333 .333 .333;
estimate 'female (333)' intercept 1  sex 1 0 race  .333 .333 .333
          race*sex  .333 .333 .333  0 0 0;
estimate 'female-male (333)' sex 1 -1 race*sex  .333 .333 .333  -.333 -.333 -.333;
estimate 'male (334)' intercept 1  sex 0 1  race  .333 .333 .334
          race*sex 0 0 0  .333 .333 .334;
estimate 'female (334)' intercept 1  sex 1 0 race  .333 .333 .334
          race*sex  .333 .333 .334  0 0 0;
estimate 'female-male (334)' sex 1 -1 race*sex  .333 .333 .334  -.333 -.333 -.334;
estimate 'male (d)' intercept 3  sex 0 3  race  1 1 1  race*sex 0 0 0  1 1 1 / divisor=3;
estimate 'female (d)' intercept 3  sex 3 0 race  1 1 1  race*sex 1 1 1  0 0 0 / divisor=3;
estimate 'female-male (d)' sex 3 -3 race*sex 1 1 1   -1 -1 -1 / divisor=3;
```

```
estimate 'male (o)' intercept 849  sex 0 849  race  142 128 579
          race*sex 0 0 0  142 128 579 / divisor=849;
estimate 'female (o)' intercept 849  sex 849 0 race  142 128 579
          race*sex 142 128 579  0 0 0 / divisor=849;
estimate 'female-male (o)' sex 849 -849 race*sex 142 128 579   -142 -128 -579 / divisor=849;
title3 'GLM without Asian/Other; race weighted using default and observed using counts';
run;
```

We show the ESTIMATE statements we might use to get the default (equally-weighted) values, namely coefficients of .333. Unfortunately, those do not add up to 1 exactly and so are nonestimable. We can solve the problem by putting one of them to .334 (or adding more digits and still making sure one ends in 4 instead of 3, or adding enough digits that the sum is close enough to 1), but that is not very elegant. Instead, we can specify a DIVISOR of 3 and all the coefficients are divided by 3. This means that where we would normally want a 1 we need to put a 3, which is a bit strange looking but does the trick.

To get the values for the observed proportions, instead of using long decimals (and making sure they add up to 1 exactly) we can instead simply put in the counts and the overall N as the divisor. The counts will add up to the overall N and we should have no trouble with estimability:

Parameter	Estimate	Standard Error	t Value	Pr > |t|
male (334)	76.4417741	1.88286738	40.60	<.0001
female (334)	73.1831213	2.63245041	27.80	<.0001
female-male (334)	-3.2586528	3.23650811	-1.01	0.3143
male (d)	76.4543608	1.88419071	40.58	<.0001
female (d)	73.1974355	2.63417346	27.79	<.0001
female-male (d)	-3.2569253	3.23867943	-1.01	0.3149
male (o)	69.8106240	1.48329152	47.06	<.0001
female (o)	65.6371940	2.17514619	30.18	<.0001
female-male (o)	-4.1734301	2.63275800	-1.59	0.1133

These are some examples of the use of the ESTIMATE statement but there are many other contexts where they are useful, such as in complex multifactor models. A good resource for coding ESTIMATE statements with multifactor models is Usage Note *24447:* Are there any examples of writing proper CONTRAST and ESTIMATE statements? at //support.sas.com/notes/index.html.

THE STRATIFIED PIECEWISE LINEAR MODEL AND SOME ESTIMATE STATEMENTS

Now that we have a better grasp on the ESTIMATE statement, let us return to the cystic fibrosis model we introduced at the beginning. Recall that the model stratifies patients into ten deciles (based on their lung function at an index event) and fits two separate lines for each patient, one before the index event and one after the index event. Shown below is the model and the first group of ESTIMATE statements. The model includes four variables that define the two lines. The intercept estimates the value at time 0 (the index event) using the pre-index data. The variable t represents time and ranges from -2 to +2. The variable tafter is 1 for the time after index and 0 before; it represents the change in intercept from pre- to post-index. The variable t0 equals max(t,0) and therefore represents the change in slope between the pre-index and post-index periods. This is a convenient parameterization for piecewise linear models because it provides a direct test of the change in intercept and the change in slope, both of which are likely to be of interest. For more on parameterization of piecewise linear models, see Pasta 2005.

The model includes the decile variable (which takes on values 1 to 10 to represent the 10 deciles of severity) alone and interacted with t, t0, and tafter. This causes the MIXED procedure to calculate, for each decile, an estimated average value for the intercept, t, t0, and tafter.

Technical Note: *Because the same patient could contribute more than one index event, the mixed model allows for within-patient correlation through the use of two RANDOM statements. At the level of the index event, all four parameters of the lines were treated as random effects with an unstructured covariance matrix parameterized as fa0(4) to ensure it is positive semi-definite. At the level of the patient, the slope and intercept of the pre-index line were treated as random effects with unstructured covariance; we found*

empirically that the patient-level variances associated with the change in intercept and the change in slope were near zero, so we set them to zero to avoid numerical instabilities. This implies that there is a correlation within patient for the overall slope and intercept but not for the change values.

```
proc mixed data = anal01 noclprint;
  class patid patid_age decile;
  model fev1pct = decile
                  decile*t
                  decile*t0
                  decile*tafter / solution ddfm=bw;
  random intercept t / sub=patid type=fa0(2) g gcorr;
  random intercept t t0 tafter / sub=patid_age(patid) type=fa0(4) g gcorr;
  *** Estimates for slope BEFORE index event ***;
  estimate 'Before:1'   decile*t   1 0 0 0 0 0 0 0 0 0;
  estimate 'Before:2'   decile*t   0 1 0 0 0 0 0 0 0 0;
  estimate 'Before:3'   decile*t   0 0 1 0 0 0 0 0 0 0;
  estimate 'Before:4'   decile*t   0 0 0 1 0 0 0 0 0 0;
  estimate 'Before:5'   decile*t   0 0 0 0 1 0 0 0 0 0;
  estimate 'Before:6'   decile*t   0 0 0 0 0 1 0 0 0 0;
  estimate 'Before:7'   decile*t   0 0 0 0 0 0 1 0 0 0;
  estimate 'Before:8'   decile*t   0 0 0 0 0 0 0 1 0 0;
  estimate 'Before:9'   decile*t   0 0 0 0 0 0 0 0 1 0;
  estimate 'Before:10'  decile*t   0 0 0 0 0 0 0 0 0 1;
  estimate 'Before:U'   decile*t   1 1 1 1 1 1 1 1 1 1 / divisor=10;
  estimate 'Before:O'   decile*t   0.0666048524 0.0776077886 0.0888270746
                                   0.0955339206 0.1017462525 0.1059496214
                                   0.1124710246 0.1164580436 0.1168907433
                                   0.1179106784 ;
  [MORE . . . ]
```

These estimate statements start out pretty easy. To get the estimated slope before the index event for each decile, we just need to pick out the interaction of the decile and the time variable for that decile. Note that the 0s after the 1 are not strictly necessary; I like to include them to make clear what is going on. The decile*t effect represents 10 parameters, so I like to have all 10 coefficients appear.

Is it possible to get these same estimates simply by specifying "solution" on the model statement? Indeed it is. Later values will not be possible to get that way and this is a good check on the coding.

The next two ESTIMATE statements obtain an overall average "before" slope. The one labeled U uses a uniform distribution across the deciles – each decile is given a weight of 0.1. This could be expressed as a coefficient of 0.1 for each or, as is done here, by specifying a 1 for each coefficient and a divisor of 10. The divisor approach is especially convenient for avoiding long fractions when there are, say, 7 or 13 categories. The one labeled O uses the observed distribution across the deciles. The extent to which some deciles are over- or under-represented here is a reflection of the actual data available. There are times when the uniform approach makes the most sense and there are times when the observed approach makes the most sense. This choice corresponds to the default in LSMEANS (the uniform approach) or the version you get when you specify OBSMARGINS (the observed approach). Note that the observed version requires calculating to many decimal places to get good accuracy and making sure the coefficients sum exactly to 1.0 (if they do not, because of rounding, they need to be adjusted so that they do). We will see a couple of easier ways to specify these coefficients later.

MORE ESTIMATE STATEMENTS

In addition to the pre-index slope, we want to look at the post-index slope and various other values derived from the four values (intercept, t, t0, and tafter). Here are more ESTIMATE statements.

```
  *** Estimates for slope AFTER index event ***;
  estimate 'After:1'    decile*t      1 0 0 0 0 0 0 0 0 0
                        decile*t0     1 0 0 0 0 0 0 0 0 0;
  estimate 'After:2'    decile*t      0 1 0 0 0 0 0 0 0 0
                        decile*t0     0 1 0 0 0 0 0 0 0 0;
```

```
  estimate 'After:3'    decile*t      0 0 1 0 0 0 0 0 0 0
                        decile*t0     0 0 1 0 0 0 0 0 0 0;
  estimate 'After:4'    decile*t      0 0 0 1 0 0 0 0 0 0
                        decile*t0     0 0 0 1 0 0 0 0 0 0;
  estimate 'After:5'    decile*t      0 0 0 0 1 0 0 0 0 0
                        decile*t0     0 0 0 0 1 0 0 0 0 0;
  estimate 'After:6'    decile*t      0 0 0 0 0 1 0 0 0 0
                        decile*t0     0 0 0 0 0 1 0 0 0 0;
  estimate 'After:7'    decile*t      0 0 0 0 0 0 1 0 0 0
                        decile*t0     0 0 0 0 0 0 1 0 0 0;
  estimate 'After:8'    decile*t      0 0 0 0 0 0 0 1 0 0
                        decile*t0     0 0 0 0 0 0 0 1 0 0;
  estimate 'After:9'    decile*t      0 0 0 0 0 0 0 0 1 0
                        decile*t0     0 0 0 0 0 0 0 0 1 0;
  estimate 'After:10'   decile*t      0 0 0 0 0 0 0 0 0 1
                        decile*t0     0 0 0 0 0 0 0 0 0 1;
  estimate 'After:U'    decile*t      1 1 1 1 1 1 1 1 1 1
                        decile*t0     1 1 1 1 1 1 1 1 1 1 / divisor=10;
  estimate 'After:O'    decile*t      0.0666048524 0.0776077886 0.0888270746
                                      0.0955339206 0.1017462525 0.1059496214
                                      0.1124710246 0.1164580436 0.1168907433
                                      0.1179106784
                        decile*t0     0.0666048524 0.0776077886 0.0888270746
                                      0.0955339206 0.1017462525 0.1059496214
                                      0.1124710246 0.1164580436 0.1168907433
                                      0.1179106784 ;

*** Estimates for DIFFERENCE in slope between before and after index event ***;
  estimate 'Diff:1'     decile*t0     1 0 0 0 0 0 0 0 0 0;
  estimate 'Diff:2'     decile*t0     0 1 0 0 0 0 0 0 0 0;
  estimate 'Diff:3'     decile*t0     0 0 1 0 0 0 0 0 0 0;
  estimate 'Diff:4'     decile*t0     0 0 0 1 0 0 0 0 0 0;
  estimate 'Diff:5'     decile*t0     0 0 0 0 1 0 0 0 0 0;
  estimate 'Diff:6'     decile*t0     0 0 0 0 0 1 0 0 0 0;
  estimate 'Diff:7'     decile*t0     0 0 0 0 0 0 1 0 0 0;
  estimate 'Diff:8'     decile*t0     0 0 0 0 0 0 0 1 0 0;
  estimate 'Diff:9'     decile*t0     0 0 0 0 0 0 0 0 1 0;
  estimate 'Diff:10'    decile*t0     0 0 0 0 0 0 0 0 0 1;
  estimate 'Diff:U'     decile*t0     1 1 1 1 1 1 1 1 1 1 / divisor=10;
  estimate 'Diff:O'     decile*t0     0.0666048524 0.0776077886 0.0888270746
                                      0.0955339206 0.1017462525 0.1059496214
                                      0.1124710246 0.1164580436 0.1168907433
                                      0.1179106784 ;

  *** Estimates of INCREASE AFTER the index event ***;
  estimate 'IncAfter:1'  decile*tafter  1 0 0 0 0 0 0 0 0 0;
  estimate 'IncAfter:2'  decile*tafter  0 1 0 0 0 0 0 0 0 0;
  estimate 'IncAfter:3'  decile*tafter  0 0 1 0 0 0 0 0 0 0;
  estimate 'IncAfter:4'  decile*tafter  0 0 0 1 0 0 0 0 0 0;
  estimate 'IncAfter:5'  decile*tafter  0 0 0 0 1 0 0 0 0 0;
  estimate 'IncAfter:6'  decile*tafter  0 0 0 0 0 1 0 0 0 0;
  estimate 'IncAfter:7'  decile*tafter  0 0 0 0 0 0 1 0 0 0;
  estimate 'IncAfter:8'  decile*tafter  0 0 0 0 0 0 0 1 0 0;
  estimate 'IncAfter:9'  decile*tafter  0 0 0 0 0 0 0 0 1 0;
  estimate 'IncAfter:10' decile*tafter  0 0 0 0 0 0 0 0 0 1;
  estimate 'IncAfter:U'  decile*tafter  1 1 1 1 1 1 1 1 1 1 / divisor=10;
  estimate 'IncAfter:O'  decile*tafter  0.0666048524 0.0776077886 0.0888270746
                                        0.0955339206 0.1017462525 0.1059496214
                                        0.1124710246 0.1164580436 0.1168907433
                                        0.1179106784 ;
```

To get the slope after the index event, it is necessary to include the coefficient of t and of t0 for the corresponding decile. Note that there is *not* a semicolon between the two lines; you are creating a single estimate using coefficients from two different terms in the model.

MACRO VARIABLES TO SIMPLIFY THE ESTIMATE STATEMENTS

We can use macro variables to simplify the ESTIMATE statements and make them easier to follow. Here we are getting the two intercepts at time zero: the one for the end of the BEFORE period and the one at the beginning of the AFTER period. We need to include the INTERCEPT term (included in the model by default).

```
%let S_1    = 1 0 0 0 0 0 0 0 0 0;
%let S_2    = 0 1 0 0 0 0 0 0 0 0;
%let S_3    = 0 0 1 0 0 0 0 0 0 0;
%let S_4    = 0 0 0 1 0 0 0 0 0 0;
%let S_5    = 0 0 0 0 1 0 0 0 0 0;
%let S_6    = 0 0 0 0 0 1 0 0 0 0;
%let S_7    = 0 0 0 0 0 0 1 0 0 0;
%let S_8    = 0 0 0 0 0 0 0 1 0 0;
%let S_9    = 0 0 0 0 0 0 0 0 1 0;
%let S_10   = 0 0 0 0 0 0 0 0 0 1;
%let S_none = 0 0 0 0 0 0 0 0 0 0;
%let P_U    = .1 .1 .1 .1 .1 .1 .1 .1 .1 .1;
%let P_O    = 0.0666048524 0.0776077886 0.0888270746
              0.0955339206 0.1017462525 0.1059496214
              0.1124710246 0.1164580436 0.1168907433
              0.1179106784 ;
  *** Intercept at the end of the BEFORE and at the beginning of the AFTER ***;
  estimate 'LS:1_Before'   intercept             1
                           decile                &S_1.
                           decile*tafter         &S_none.;
  estimate 'LS:1_After'    intercept             1
                           decile                &S_1.
                           decile*tafter         &S_1.;
  estimate 'LS:2_Before'   intercept             1
                           decile                &S_2.
                           decile*tafter         &S_none.;
  estimate 'LS:2_After'    intercept             1
                           decile                &S_2.
                           decile*tafter         &S_2.;
  . . .
  estimate 'LS:10_Before'  intercept             1
                           decile                &S_10.
                           decile*tafter         &S_none.;
  estimate 'LS:10_After'   intercept             1
                           decile                &S_10.
                           decile*tafter         &S_10.;
  estimate 'LS:Before_U'   intercept             1
                           decile                &P_U.
                           decile*tafter         &S_none.;
  estimate 'LS:After_U'    intercept             1
                           decile                &P_U.
                           decile*tafter         &P_U.;
  estimate 'LS:Before_O'   intercept             1
                           decile                &P_O.
                           decile*tafter         &S_none.;
  estimate 'LS:After_O'    intercept             1
                           decile                &P_O.
                           decile*tafter         &P_O.;
```

The S_none macro variable is not necessary here but it helps show the parallelism of the constructs. Note that P_U is expressed as a fraction. If we use the DIVISOR option here, we need the coefficient of the INTERCEPT to be equal to the DIVISOR, which is a bit obscure. It seems easier to follow this way.

TESTING HYPOTHESES WITH CONTRAST STATEMENTS

In general, ESTIMATE statements provide the necessary functionality to calculate estimated combinations of parameter values, their standard errors, and the associated confidence intervals. At times, however, a CONTRAST statement is needed because you want to test a hypothesis with more than one degree of freedom. In this context, we wanted to test the null hypothesis that both the change in intercept and the change in slope are simultaneously zero, i.e. that the AFTER line segment is not statistically different from the BEFORE line segment.

They syntax of CONTRAST statements is very similar to the syntax of ESTIMATE statements except that multiple effects are specified and separated by commas. Using the macro variables created above, the statements are quite compact using this parameterization:

```
*** Simultaneous test of change in intercept and slope ***;
contrast  'PIntSlope:1'  decile*t0      &S_1.,
                         decile*tafter  &S_1.;
contrast  'PIntSlope:2'  decile*t0      &S_2.,
                         decile*tafter  &S_2.;
. . .
contrast  'PIntSlope:10' decile*t0      &S_10.,
                         decile*tafter  &S_10.;
contrast  'PIntSlope:U'  decile*t0      &P_U.,
                         decile*tafter  &P_U.;
contrast  'PIntSlope:O'  decile*t0      &P_O.,
                         decile*tafter  &P_O.;
```

OBTAINING SUBGROUP COUNTS AND PROPORTIONS AUTOMATICALLY

One considerable annoyance when constructing ESTIMATE statements is the need to specify proportions very precisely. It is possible to use the macro language and SYMPUT to get the subgroup counts and automatically construct the necessary proportions. In addition to being less error-prone, this is a lifesaver when the counts change because the analysis is slightly revised.

```
*** Obtaining subgroup Ns to automatically feed into model ***;
*** First get unique patient_age records with decile ***;
proc sort data=temp01(keep=patid_age decile) out=temp02 nodupkey;
  by patid_age decile;
run;

data _null_;
  set temp02;
  by patid_age decile;
  if not(first.patid_age and last.patid_age)
          then error 'ERROR: decile varies within patid_age';
run;

ods listing close;
ods output onewayfreqs = temp03(keep=decile frequency cumfrequency);
proc freq data = temp02;
  tables decile;
run;
ods listing;

data _null_;
  set temp03;
  call symput('N_'||left(trim(put(decile,2.))),frequency);
run;
```

```
data _null_;
  set temp03;
  if (decile eq 10) then call symput('Ntot',cumfrequency);
run;

data _null_;
  call symput('P_1', &N_1. / &Ntot.));
  call symput('P_2', &N_2. / &Ntot.);
  call symput('P_3', &N_3. / &Ntot.);
  call symput('P_4', &N_4. / &Ntot.);
  call symput('P_5', &N_5. / &Ntot.);
  call symput('P_6', &N_6. / &Ntot.);
  call symput('P_7', &N_7. / &Ntot.);
  call symput('P_8', &N_8. / &Ntot.);
  call symput('P_9', &N_9. / &Ntot.);
  call symput('P_10', &N_10. / &Ntot.);
run;

*** Macro variable for model specifications ***;
%let P_O = &P_1. &P_2. &P_3. &P_4. &P_5. &P_6. &P_7. &P_8. &P_9. &P_10.;
```

Specifying the SYMBOLGEN option results in the assigned values appearing in the log:

```
SYMBOLGEN:  Macro variable NTOT resolves to       32355

SYMBOLGEN:  Macro variable N_1 resolves to        2155
SYMBOLGEN:  Macro variable N_2 resolves to        2511
SYMBOLGEN:  Macro variable N_3 resolves to        2874
SYMBOLGEN:  Macro variable N_4 resolves to        3091
SYMBOLGEN:  Macro variable N_5 resolves to        3292
SYMBOLGEN:  Macro variable N_6 resolves to        3428
SYMBOLGEN:  Macro variable N_7 resolves to        3639
SYMBOLGEN:  Macro variable N_8 resolves to        3768
SYMBOLGEN:  Macro variable N_9 resolves to        3782
SYMBOLGEN:  Macro variable N_10 resolves to        3815

SYMBOLGEN:  Macro variable P_1 resolves to 0.0666048524
SYMBOLGEN:  Macro variable P_2 resolves to 0.0776077886
SYMBOLGEN:  Macro variable P_3 resolves to 0.0888270746
SYMBOLGEN:  Macro variable P_4 resolves to 0.0955339206
SYMBOLGEN:  Macro variable P_5 resolves to 0.1017462525
SYMBOLGEN:  Macro variable P_6 resolves to 0.1059496214
SYMBOLGEN:  Macro variable P_7 resolves to 0.1124710246
SYMBOLGEN:  Macro variable P_8 resolves to 0.1164580436
SYMBOLGEN:  Macro variable P_9 resolves to 0.1168907433
SYMBOLGEN:  Macro variable P_10 resolves to 0.1179106784
```

Additional programming can be used to capture the results of the ESTIMATE and CONTRAST statements and construct Table 1 automatically. This involves careful use of the ODS tables and some attention to getting the rows and columns in the right order. The effort is repaid many times over when (not if!) the analysis needs to be rerun with slight variations.

OUTPUT FROM PROC MIXED

The output from MIXED includes some information that helps you check the ESTIMATE statements were coded correctly. Specifically, you can compare the solution for fixed effects with the corresponding estimates to see if the values match (where they should match) and can do a little arithmetic to see if they are about right in other cases.

Solution for Fixed Effects

Effect	Age-adjusted FEV1 decile	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		90.9166	0.2888	14E3	314.76	<.0001
decile	1	-44.2671	0.4340	11E3	-101.99	<.0001
decile	2	-35.9684	0.3977	11E3	-90.44	<.0001
. . .						
decile	10	0	.	.	.	.
t*decile	1	-4.0451	0.1646	58E4	-24.58	<.0001
t*decile	2	-3.2907	0.1547	58E4	-21.27	<.0001
. . .						
t*decile	10	0.4887	0.1409	58E4	3.47	0.0005
t0*decile	1	2.9726	0.2320	58E4	12.81	<.0001
t0*decile	2	1.7332	0.2149	58E4	8.06	<.0001
. . .						
t0*decile	10	-3.2569	0.1966	58E4	-16.57	<.0001
tafter*decile	1	0.000519	0.2103	58E4	0.00	0.9980
tafter*decile	2	0.01151	0.1998	58E4	0.06	0.9541
. . .						
tafter*decile	10	-0.4046	0.1971	58E4	-2.05	0.0401

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
decile	9	11E3	1669.15	<.0001
t*decile	10	58E4	183.24	<.0001
t0*decile	10	58E4	74.12	<.0001
tafter*decile	10	58E4	5.94	<.0001

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|
Before:1	-4.0451	0.1646	58E4	-24.58	<.0001
Before:2	-3.2907	0.1547	58E4	-21.27	<.0001
. . .					
Before:10	0.4887	0.1409	58E4	3.47	0.0005
Before:U	-1.6042	0.04690	58E4	-34.20	<.0001
Before:0	-1.3799	0.04678	58E4	-29.50	<.0001
After:1	-1.0725	0.1620	58E4	-6.62	<.0001
After:2	-1.5574	0.1465	58E4	-10.63	<.0001
. . .					
After:10	-2.7682	0.1373	58E4	-20.17	<.0001
After:U	-1.9179	0.04498	58E4	-42.64	<.0001
After:0	-1.9758	0.04484	58E4	-44.06	<.0001
Diff:1	2.9726	0.2320	58E4	12.81	<.0001

Diff:2	1.7332	0.2149	58E4	8.06	<.0001
. . .					
Diff:10	-3.2569	0.1966	58E4	-16.57	<.0001
Diff:U	-0.3137	0.06375	58E4	-4.92	<.0001
Diff:O	-0.5959	0.06354	58E4	-9.38	<.0001
IncAfter:1	0.000519	0.2103	58E4	0.00	0.9980
IncAfter:2	0.01151	0.1998	58E4	0.06	0.9541
. . .					
IncAfter:10	-0.4046	0.1971	58E4	-2.05	0.0401
IncAfter:U	-0.3633	0.06130	58E4	-5.93	<.0001
IncAfter:O	-0.4000	0.06174	58E4	-6.48	<.0001
LS:1_Before	46.6494	0.3465	58E4	134.61	<.0001
LS:1_After	46.6500	0.3369	58E4	138.48	<.0001
LS:2_Before	54.9482	0.3042	58E4	180.66	<.0001
LS:2_After	54.9597	0.2934	58E4	187.31	<.0001
. . .					
LS:10_Before	90.9166	0.2888	58E4	314.76	<.0001
LS:10_After	90.5120	0.2834	58E4	319.33	<.0001
LS:Before_U	70.6974	0.1506	58E4	469.43	<.0001
LS:After_U	70.3342	0.1491	58E4	471.84	<.0001
LS:Before_O	72.8524	0.1502	58E4	484.89	<.0001
LS:After_O	72.4524	0.1488	58E4	486.96	<.0001

Contrasts

Label	Num DF	Den DF	F Value	Pr > F
PIntSlope:1	2	58E4	82.72	<.0001
PIntSlope:2	2	58E4	32.84	<.0001
PIntSlope:3	2	58E4	6.27	0.0019
PIntSlope:4	2	58E4	2.07	0.1261
PIntSlope:5	2	58E4	0.44	0.6444
PIntSlope:6	2	58E4	7.53	0.0005
PIntSlope:7	2	58E4	14.46	<.0001
PIntSlope:8	2	58E4	47.18	<.0001
PIntSlope:9	2	58E4	62.39	<.0001
PIntSlope:10	2	58E4	137.91	<.0001
PIntSlope:U	2	58E4	27.38	<.0001
PIntSlope:O	2	58E4	60.41	<.0001

This example also illustrates the importance of clear labeling of the individual estimates and contrasts.

CONCLUSION

With a good understanding of the parameterization of your model, you should be able to code ESTIMATE and CONTRAST statements. A number of examples have been given to try to demystify the process. A complicated mixed model with ten levels of stratification across patients and two fitted lines within patients provides a context for many examples of ESTIMATE and CONTRAST statements. Some shortcuts in creating ESTIMATE and CONTRAST statements can make the process simpler and make it easier to check the results. To check your code, it is helpful to use the SOLUTION option on the MODEL statement and to use LSMEANS statements (with OBSMARGINS specified). You can use the SYMPUT function to get counts from the data into macro variables to allow you to automatically calculate coefficients for ESTIMATE and CONTRAST statements. Although the examples given here use GLM and MIXED, most of the sample principles can be applied to other procedures that use ESTIMATE or CONTRAST statements.

REFERENCES

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, Paper 212-30. http://www2.sas.com/proceedings/sugi30/212-30.pdf

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

ACKNOWLEDGEMENT

The project on which this paper is based is joint with my colleague Stefanie Silva Millar, who wrote code and text describing the model as well as producing the figure and table.

CONTACT INFORMATION

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

David J. Pasta
Vice President, Statistics & Data Operations
ICON Clinical Research
188 Embarcadero, Suite 200
San Francisco, CA 94105
+1.415.371.2111
david.pasta@iconplc.com
www.iconplc.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.