How to Get What You Need – Part 1

Chris Toppe, Independent Sector, Washington, DC

This is a two-part Coder's Corner. In this first part I'll show you the output, the results, of the code I'll go over in Part 2 of this presentation. First, let me set the stage.

People give money to charities. They do this for a variety of reasons. Many things effect how much a person gives, with two of the most important being whether or not they attend church weekly and whether or not they volunteer with their families. However, we also know that how much people give is strongly related to their household income: giving increases as income increases. The first question, then, is do people who attend church and volunteer with their families give more to charities after the effects of income are taken into consideration. The second question is, what can I say about the differences in giving when these variables are taken into effect.

These kinds of questions come up in data analysis over and over again. Therefore, it makes sense to have one master program that addresses these questions in a consistent and predictable way. First, there a tremendous programming advantage – you need not write new code each time the questions arise. Second, there is a cognitive advantage in that once you've learned how to read the output, you read it the same way every time.

But first, a little about the data. The analysis variable is GIVING, a continuous variable. The classifying variables are WEEKLY and FAMILY, two binomial variables that indicate, respectively, if respondents attend weekly religious services and if they volunteer with their families. The co-variate is INCOME, a continuous variable.

Because I am interested in the interaction between WEEKLY and FAMILY, I create a character variable called INTERAXN that is the combination of the four possible combinations of those two variables. When WEEKLY=1 and FAMILY=1, the value of INTERAXN is "11". Therefore, the four possible values of INTERAXN are "00", "01", "10" and "11", with WEEKLY first and FAMILY second. This is pretty simple code even if one is a novice programmer. There are more elegant ways to program this step, but elegance is not the topic of this paper

```
IF WEEKLY = 0 AND FAMILY = 0 THEN
INTERAXN = "00";
ELSE IF WEEKLY = 0 AND FAMILY = 1
THEN INTERAXN = "01";
ELSE IF WEEKLY = 1 AND FAMILY = 0
THEN INTERAXN = "10";
ELSE IF WEEKLY = 1 AND FAMILY = 1
THEN INTERAXN = "11";
```

The GLM procedure later confirms that this worked as planned:

Class Level Information		
Class	Levels	Values
interaxn	4	00 01 10 11

This done, the first analysis step run is Proc GLM® to test if there are differences in the means of INCOME when grouped by the values if INTERAXN, which are dummy variables for the combinations of WEEKLY and FAMILY. For those of you who don't know GLM, it is like PROC REG®, only more powerful. One of the most powerful things it does is let you remove the effects of a co-variate, in this case INCOME. That is, it removes the effects of INCOME and then calculates new, INCOME-adjusted means for your analysis groups. Pretty cool stuff. Just a few lines of code is all it takes:

```
PROC GLM DATA=GANDV01.D;
WEIGHT HHWGT;
CLASS INTERAXN;
MODEL GIVING=INCOME
INTERAXN/SOLUTION;
MEANS INTERAXN/TUKEY LINES
ALPHA=.01;
LSMEANS INTERAXN  / PDIFF;
```

Line 4 is the model statement, which says to predict giving based on INCOME and INTERAXN. Line 5 says to get the means for each value of INTERAXN and test to see if they are statistically different using the Tukey method with an alpha level of .01. Line 6 says to get the Least Square Means based on INTERAXN and tell me how statistically different they are. Compare Line 4 and Line 6. The basic model is INCOME and INTERAXN, while the LSMeans statement is only INTERAXN. Therefore, the means produced in Line 6 are after the effects of INCOME have been removed.

GLM produces a lot of useful data, but here I'll just show you a few of the key outputs. The first chart shows the GIVING for each value of INTERAXN along with an indication as to whether or not the means are

statistically different from each other. Note the letters assigned by GLM to each mean in the Tukey column. You read them as follows: Group A (INTERAXN="11") is greater than the group B and Group C means: The group B means are the same, so there is no measurable difference in the means for INTERAXN="01" or "10". The group=C mean, when INTERAXN="00" is lower than the others.

Means with the same letter are not significantly different.			
Tukey Grouping	Mean	N	interaxn
A	$3,457.0	527	11
B	$1,963.9	360	01
B			
B	$1,744.0	1043	10
C	$707.1	1845	00

The second set of charts are much more interesting. Here we see the income-adjusted means, called Least Square Means and denoted LSMEANS in the table. If you compare the normal and adjusted means, you can see that the adjustment process changed the values of the means. Most important, the second of the charts shows the results of the tests to see if the means are different. You can see in that chart that each of the means is significantly different from the others, a different conclusion than obtained by the normal means test. That is, after the effects of income are removed, there are significant differences in GIVING based in church attendance and family volunteering.

Interaxn	GIVING LSMEAN	LSMEAN Number
00	$887.4	1
01	$1,410.9	2
10	$2,063.4	3
11	$3,124.2	4

Least Squares Means for effect Interaxn Pr > |t| for H0: LSMean(i)=LSMean(j) Dependent Variable: GIVING				
I/j	1	2	3	4
1		0.0007	<.0001	<.0001
2	0.0007		<.0001	<.0001
3	<.0001	<.0001		<.0001
4	<.0001	<.0001	<.0001	

Note that the adjusted means do not exist in reality, so I can't report them as real means. However, I can talk about the real means being different after the effects of income have been removed. Now that I know that I have a real difference to talk about, there are a lot of things I want to know about giving by those who attend church and those who volunteer with their families. To answer most of the remaining questions I have, I use Proc Tabulate®.

```
PROC TABULATE DATA=GANDV01.D;
WEIGHT HHWGT;
CLASS WEEKLY FAMILY;
VAR GIVING
TABLE  WEEKLY*(FAMILY  ALL="SUB-
TOTAL") ALL="GRAND TOTAL",
GIVING*(
SUMWGT="WEIGHTED TOTAL NUMBER IN
POPLATION"*F=COMMA12.
PCTN="PERCENT OF WEIGHTED TOTAL
NUMBER"*F=7.1
PCTN<FAMILY ALL>="PERCENT OF GIVE
GROUP"*F=7.1
MEAN="WEIGHTED
AVERAGE"*F=DOLLAR8.
SUM="WEIGHTED TOTAL"*F=DOLLAR15.
PCTSUM="PERCENT    OF    WEIGHTED
TOTAL"*F=7.1
PCTSUM<FAMILY  ALL>="PERCENT  OF
ANALYSIS GROUP"*F=7.1);
TITLE1 "WEIGHTED STATISTICS";
TITLE2     "ANNUAL     HOUSEHOLD
CONTRIBUTION";
TITLE3 "BY WEEKLY AND FAMILY";
RUN;
```

To say this is a straight-forward Tabulate program would only be true to those of you who know the procedure. If you copy it and substitute your class variables and your analysis variable in the few places they are used (bolded in the program), you'll get the exact same results. Just be sure your class variables are binomials (0-1 values, for example) and your analysis variable is continuous (like dollars) and you'll be fine. If you want to actually learn Proc Tabulate, look for presentations or papers by Dan Bruns at SESUG. He teaches Tabulate as well as anyone I know. As this is not a Tabulate paper, I'll just move to the results.

		TOTAL GIVING						
		Weighted Total Number in Population	Percent of Weighted Total Number	Percent of Give Group	Weighted Average	Weighted Total	Percent of Weighted Total	Percent of Analysis Group
Weekly	Family							
0	0	1,892	48.9	83.7	$707	$1,338,171	24.4	67.2
	1	333	9.5	16.3	$1,964	$653,290	11.9	32.8
	Sub-Total	2,225	58.4	100.0	$895	$1,991,461	36.3	100.0
1	Family							
	0	1,073	27.6	66.4	$1,744	$1,871,533	34.1	53.4
	1	472	14.0	33.6	$3,457	$1,630,469	29.7	46.6
	Sub-Total	1,545	41.6	100.0	$2,267	$3,502,002	63.7	100.0
Grand Total		3,770	100.0	100.0	$1,457	$5,493,463	100.0	100.0

This output gives me pretty much everything I need to know about how GIVING differs across my classification groups, WEEKLY and FAMILY. Here are a few of the statements I can make:

I can talk about percentages: Those who attend weekly and volunteer with their families are 14.0% of all people yet they give nearly 30% of all money.

I can talk about averages: Those who do not attend church and who do not volunteer with their families are nearly half of all people. They give about 25% of all donations to charities, averaging about $700 per year.

I can compare within groups: Those who attend church weekly give more when they also volunteer with their families, $3,457 to $1,744.

I can compare between groups: Those who attend weekly give more, $2,267, than those who do not attend weekly, $895.

Basically, I can write a report from this table. And by changing the classifying variables to others, I can get out a totally new set of data that I can read the same way. For example, I could use MARRIED or HOMEOWNER or COLLEGE or REGION and create a whole new set of comparisons. As a matter of fact, except for the variable names, nothing changes in the program. If I made a few changes, like putting in macro variables instead of hard-coded variable names, I could automate this program. That is the topic of Part 2 of this presentation.

About the Author

Chris Toppe is Director of Philanthropic Studies at Independent Sector. In this role he manages the collection, analysis, and reporting of data on the charitable activities of Americans. His focus is on uncovering facts and findings that are useful to practioners and policy makers. Chris also teaches in the graduate school of Georgetown University where he supervises original research for graduate students enrolled in the Georgetown Public Policy Institute. Chris has been using SAS since the 1980s and has presented papers at over two dozen local, regional, and SUGI conferences. Chris Toppe, Ph.D. Director, Philanthropic Studies, Independent Sector, 1200 18th Street, NW Suite 200 Washington, DC 20036 202.467.6115 (office) 202.467.6101 (fax) chris@IndependentSector.org.