How to Get What You Need – Part 2

Chris Toppe, Independent Sector, Washington, DC

In part 1 of this two-part Coders Corner, I presented a set of programs that both analyze and produce tables for my data. Please refer to that paper for a description of the data and to see what the programs do.

Now I want to combine those programs into one master program and automate it so that I can run it for any combination of classifying variables. While I'm at it, I might as well add in the Output Delivery System code to create Word-ready output.

If you look back at the code, there really are only two things that I need to change if I want to run these programs with different variables: the classifying variable names. I could, and have, just used the SAS editor to change the names, but that gets rather tedious. If I can change those variable names into macro names, then I can run this program over and over with only one line of code.

First, give the macro a name and declare the expected input. This line of code says I'm creating a macro named TWOVARS and that there will be two input variables, VAR1 and VAR2, which will become the macro variables.

```
%MACRO TWOVARS(VAR1= , VAR2= );
```

Next, I subset the data with a KEEP list, showing the first time I use the macro variables. When the code executes, the names of the real variables will be substituted for the &VAR1 and &VAR2 macro names.

```
DATA GANDV01.D; SET GANDV01.FAMILIES
(KEEP = &VAR1 &VAR2 GIVING INCOME
HHWGT);
```

Next, I create the INTERAXN variable. Note that this is an expansion of the code that was used in the old code. This code allows the classification variables to have up to six values each (0 – 5), whereas the old code only allowed binary values (0 1). Note that the variable INTERAXN is not a macro variable, but is built from the two input macro variables.

```
IF &VAR1=0 AND &VAR2=0 THEN
INTERAXN='00';
IF &VAR1=0 AND &VAR2=1 THEN
INTERAXN='01';
IF &VAR1=0 AND &VAR2=2 THEN
INTERAXN='02';
IF &VAR1=0 AND &VAR2=3 THEN
INTERAXN='03';
IF &VAR1=0 AND &VAR2=4 THEN
INTERAXN='04';
IF &VAR1=0 AND &VAR2=5 THEN
INTERAXN='05';
IF &VAR1=1 AND &VAR2=0 THEN
INTERAXN='10';
IF &VAR1=1 AND &VAR2=1 THEN
INTERAXN='11';
IF &VAR1=1 AND &VAR2=2 THEN
INTERAXN='12';
IF &VAR1=1 AND &VAR2=3 THEN
INTERAXN='13';
IF &VAR1=1 AND &VAR2=4 THEN
INTERAXN='14';
IF &VAR1=1 AND &VAR2=5 THEN
INTERAXN='15';

RUN;
```

At this point, the macro variables have been stored, the data subset, and the INTERAXN variable created. All that is left for me to do is run the Proc GLM® and Proc Tabulate® programs. But first, I'll put in the ODS code. Note that I create a Word file that is named for the variables that are in the calling macro, variables &VAR1 and &VAR2. Also note, and this is important, that the macro names are followed by a period (".") so that SAS knows when the name ends. That is, the period is required! If I run this job using the variables WEEKLY and FAMILY, then the file will be named "WEEKLY_BY_FAMILY_GLM.DOC".

```
ODS RTF FILE=
"C:\&VAR1._BY_&VAR2._GLM.DOC";
```

Now I can run Proc GLM. Note that in this case the macro variables appear only in the TITLE statement. This produces the exact same output as in the Part 1 example, except it will be based on whatever variables I declare with I invoke the macro. Also notice that I have to close the ODS file. The GLM output will appear in the Output window as well as in a Word file.

```
PROC GLM DATA=GANDV01.D;
WEIGHT HHWGT;
CLASS INTERAXN;
MODEL GIVING=INCOME
INTERAXN/SOLUTION;
MEANS INTERAXN/TUKEY LINES
ALPHA=.01;
LSMEANS INTERAXN  / PDIFF;
TITLE1 "ANALYSIS OF &VAR1 AND
&VAR2";
RUN;
QUIT;

ODS RTF CLOSE;
```

Next on my list of things to do is run the Proc Tabulate. Note that the macro variables are used in several places in

this code. First, an ODS statement opens a new file with the variable names in the file name.

```
ODS RTF
FILE="C:\&VAR1._BY_&VAR2._TAB.DOC";

PROC TABULATE DATA=GANDV01.D;
WEIGHT HHWGT;
CLASS &VAR1 &VAR2;
VAR GIVING;

TABLE &VAR1*(&VAR2 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<&VAR2 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<&VAR2 ALL>="PERCENT OF
ANALYSIS GROUP"*F=7.1);
TITLE1 "BY &VAR1 AND &VAR2";
RUN;
```

I then close the ODS file…

```
ODS RTF CLOSE;
```

And end the macro.

```
%MEND TWOVARS;
```

With that done, the macro is ready to go. I first have to compile the macro by running all the code from the top of the macro to the bottom. After that, I can invoke the macro as often as I wish. Below are some examples of variables I have used with this code, with the first one being the example used in Part 1 of this Coders Corner presentation. Note there is no ";" when you call a macro!!!

- `%TWOVARS(VAR1=WEEKLY, VAR2=FAMILY)`
- `%TWOVARS(VAR1=COLLEGE, VAR2=MARRIED)`
- `%TWOVARS(VAR1=ITEMIZERS, VAR2=HOMEOWNERS)`
- `%TWOVARS(VAR1=INCCLS, VAR2=REGION)`

Note that in the last example INCCLS is a five-value income class variable that puts people into one of five income groups. The program, then, controls for income effects within income group, a very powerful technique.

The entire program is below so you can cut and paste easily. When you copy it, make the changes you need to match your system, like LIBNAME. Remember your class variables must be categorical variables with up to six mutually exclusive values (0 – 5), and that your analysis variable must be a continuous variable. In GLM, you can use any kind of variable as your co-variate (where I use INCOME) as long as you know what you're doing. Hope you find this useful. And because these papers are limited to 3 pages, you'll have to add in your own code comments.

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. Contact Information: 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.

```
%MACRO TWOVARS(VAR1= , VAR2= );

DATA GANDV01.D;
SET GANDV01.FAMILIES (KEEP = &VAR1 &VAR2 GIVING HHWGT);
IF &VAR1=0 AND &VAR2=0 THEN INTERAXN='00';
IF &VAR1=0 AND &VAR2=1 THEN INTERAXN='01';
IF &VAR1=0 AND &VAR2=2 THEN INTERAXN='02';
IF &VAR1=0 AND &VAR2=3 THEN INTERAXN='03';
IF &VAR1=0 AND &VAR2=4 THEN INTERAXN='04';
IF &VAR1=0 AND &VAR2=5 THEN INTERAXN='05';

IF &VAR1=1 AND &VAR2=0 THEN INTERAXN='10';
IF &VAR1=1 AND &VAR2=1 THEN INTERAXN='11';
IF &VAR1=1 AND &VAR2=2 THEN INTERAXN='12';
IF &VAR1=1 AND &VAR2=3 THEN INTERAXN='13';
IF &VAR1=1 AND &VAR2=4 THEN INTERAXN='14';
IF &VAR1=1 AND &VAR2=5 THEN INTERAXN='15';
RUN;

ODS RTF FILE= "C:\&VAR1._BY_&VAR2._GLM.DOC";

PROC GLM DATA=GANDV01.D;
WEIGHT HHWGT;
CLASS INTERAXN;
MODEL GIVING=INCOME INTERAXN/SOLUTION;
MEANS INTERAXN/TUKEY LINES ALPHA=.01;
LSMEANS INTERAXN  / PDIFF;
TITLE1 "ANALYSIS OF &VAR1 AND &VAR2";

RUN;
QUIT;

ODS RTF CLOSE;

ODS RTF FILE="C:\&VAR1._BY_&VAR2._TAB.DOC";

PROC TABULATE DATA=GANDV01.D;
WEIGHT HHWGT;
CLASS &VAR1 &VAR2;
VAR GIVING;
TABLE &VAR1*(&VAR2 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<&VAR2 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<&VAR2 ALL>="PERCENT OF ANALYSIS GROUP"*F=7.1);
TITLE1 "BY &VAR1 AND &VAR2";
RUN;

ODS RTF CLOSE;

%MEND TWOVARS;

%TWOVARS(VAR1=WEEKLY, VAR2=FAMILY) /*YOUR VARIABLE NAMES GO HERE*/
```