Paper CC84

Using MULTILABEL Formats in SAS® to Analyze Data Over Moving Periods of Time

Christopher M. Aston, USDA - Food Safety and Inspection Service

ABSTRACT

The Food Safety Inspection Service (FSIS) collects a plethora of data from all over the country on a daily basis. Many of the Agency's performance measures that it uses to identify potential trends and to assess the effectiveness of its Policies on the Meat and Poultry Industry are based on the most recent 12 months of data. Furthermore, these performance measures are normally assessed on a monthly or quarterly basis, so that these data are used multiple times in overlapping windows when we seek to do an analysis of performance over time (multiple windows). The purpose of this paper is to present the method I devised to analyze time dependent data that is evaluated as a "moving window," i.e. each data point is used multiple times in overlapping windows, so that the data are only analyzed one time. This is accomplished specifically using MULTILABEL formats in SAS® to assign specific dates to more than one period of time.

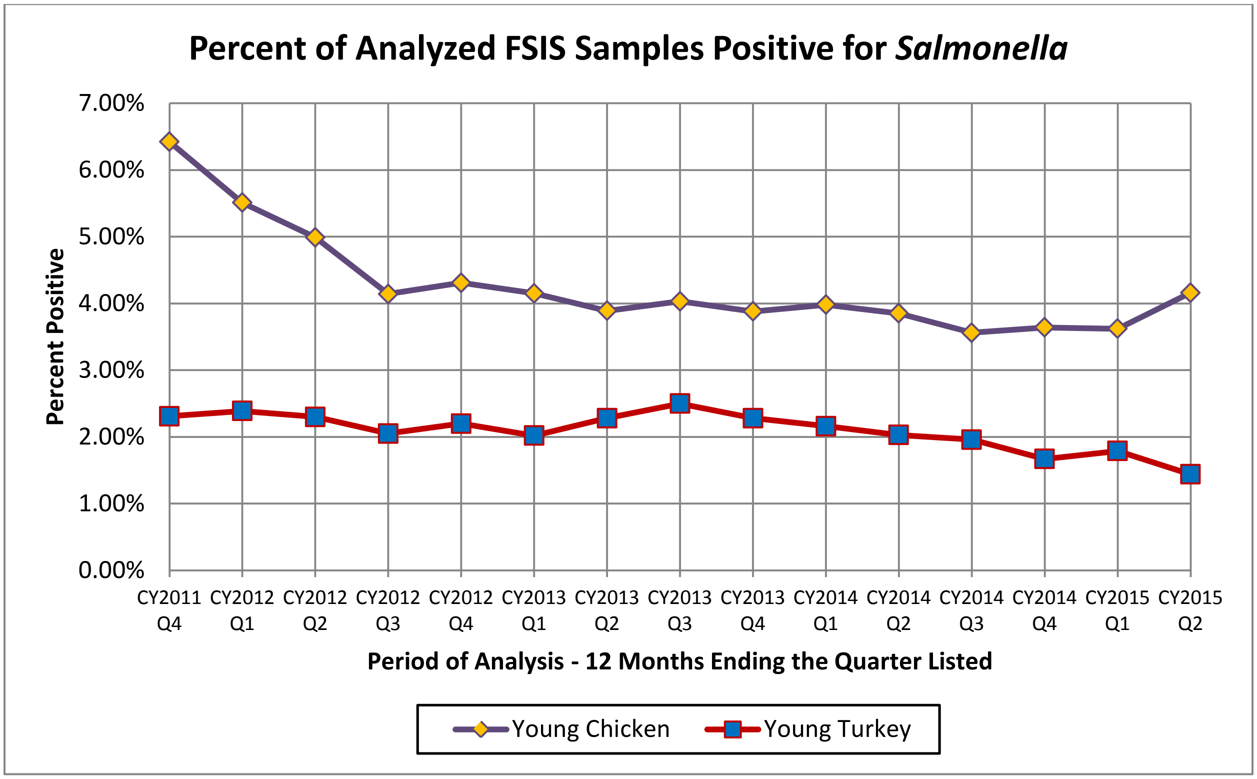

Figure 1. Example of Moving Window Statistics

INTRODUCTION

When tasked with tracking data over time, there are several different methods that can be employed, with varying amounts of efficiency, depending on the type of data being analyzed. For discrete periods of time that have no overlap, this is normally not an issue, because each observation belongs in one and only one period. The big issue, in terms of processing efficiency, arises when a single data point belongs to more than one measured period of time, as represented above in Figure 1. By using the MULTILABEL option of PROC FORMAT, this issue can easily be overcome with great ease and eloquence. It must be noted, the specific FSIS data used for this paper are not valid for time series analysis, nor should the method presented in this paper be used as a statistically appropriate form to do so.[1] Therefore, I will refrain from using that specific term throughout this paper; however, readers should be aware that SAS has other tools available precisely for that purpose.

---

[1] FSIS sampling methodology and data are available publicly on the agency website. http://www.fsis.usda.gov/wps/portal/fsis/topics/data-collection-and-reports

CREATING THE APPROPRIATE DATASET

This paper does not discuss specific data extraction methods, as they differ from agency to agency, each with its own data storage system. Moreover, this paper assumes that the programmer has already created a SAS dataset containing all the observations and variables of interest. In other words, there exists a SAS dataset with all observations starting from the first point in time of the first period to the very last point in time of the last period. For example, to create the calculations needed for Figure 1, the programmer would have already extracted *Salmonella* sampling data on young chickens and young turkeys from the FSIS Data Warehouse on all samples collected between January 1, 2011 and June 30, 2015. We will refer to this going forward as the “master” dataset.

For the purposes of the example featured in Figure 1, the master dataset contains the following existing variables from the FSIS Data Warehouse:

- Form_ID – the sampling form identification number
- Collect_Date – the date the sample was collected for analysis
- Product_Type – the type of sample that was collected for analysis
- Salmonella_Test – the result of the *Salmonella* analysis on the sample (‘Positive’ or ‘Negative’)

A subset of the master data would therefore look something like the table shown below in Table 1.

Form_ID	Collect_Date	Product_Type	Salmonella_Test
174876E801	1/3/2011	Young Chicken	Negative
174877308C	2/28/2013	Young Chicken	Negative
17487730BA	2/28/2013	Young Chicken	Positive
17487730BB	3/1/2013	Young Chicken	Negative
174877769C	2/4/2015	Young Chicken	Positive
174877769D	2/4/2015	Young Chicken	Negative
17487779C9	5/6/2015	Young Chicken	Negative
1748777A60	3/4/2011	Young Turkey	Negative
1748777A61	3/7/2011	Young Turkey	Negative
1748778469	8/3/2012	Young Turkey	Positive
174877846A	8/3/2012	Young Turkey	Negative
1748778ED2	9/19/2013	Young Turkey	Negative
1748779862	11/5/2014	Young Turkey	Negative
1748779A92	5/11/2015	Young Turkey	Negative
1748779A93	5/18/2015	Young Turkey	Negative

Table 1. Subset of Master Data

CREATING A NEW VARIABLE

Some of the SAS procedures that allow the use of MULTILABEL formats are PROC MEANS, PROC SUMMARY, and PROC TABULATE. In order to calculate the percent positive in any of these procedures, it makes sense to create another new binary variable to add into the master dataset. We could do this in the following way (although every programmer knows there are a million and one ways to do the same thing in SAS):

```
data Master_Dataset;
    set Master_Dataset;
    length Positive 8;
    if Salmonella_Test = ‘Positive’ then Positive = 1;
    else Positive = 0;
run;
```

DETERMINING THE LENGTH OF THE MOVING WINDOW

In Figure 1, we see that the moving window is twelve months and reported on a quarterly basis. Therefore, we know that most observations should be used to calculate the percent positive rates for 4 distinct quarters (with the exception of observations at the beginning and the end). For example, a young chicken sample collected on the date

2/1/2014 would belong to the periods “CY2014 Q1” (4/1/2013 – 3/31/2014), “CY2014 Q2” (7/1/2013 – 6/30/2014), “CY2014 Q3” (10/1/2013 – 9/30/2014), and “CY2014 Q4” (1/1/2014 – 12/31/2014).

CREATING A MULTILABEL FORMAT

At this point in the paper, it is very obvious how the MULTILABEL option can save a programmer a lot of time, but wait there’s more!

THE HARD WAY

In Figure 1, we see that there are 15 distinct periods which need to be accounted for. Just imagine if we wanted to look even further back into the data. Programming this in the traditional method in PROC FORMAT could take ages! For example:

```
proc format;
   value period (multilabel notsorted)
            "01Jan2011"d - "31Dec2011"d = "CY2011 Q4"
            "01Apr2011"d - "31Mar2012"d = "CY2012 Q1"
            "01Jul2011"d - "30Jun2012"d = "CY2012 Q2"
…
(It’s so tedious that I can’t even type all of it out)…
…
            "01Apr2014"d - "31Mar2015"d = "CY2015 Q1"
            "01Jul2014"d - "30Jun2015"d = "CY2015 Q2"
   ;
run;
```

You can see that this is a situation where the use of SAS macros would come in extremely handy to help us move this along more efficiently.

THE EASY WAY

In order to expedite the process, I have written a SAS macro, which can easily be modified to suit the needs of any other programmer (i.e. when a different length or type of moving window is required).

```
options mprint nosymbolgen nomlogic;

%let qtrend = "30Jun2015"d; *any SAS date in the ending quarter of interest;
%let qtrback = 15; *how many quarters back to start the formats;

*get end of the quarter date from the ‘qtrend’ macro variable for SAS;
data _null_;
   call symput ('dtnd', intnx('qtr',&qtrend.,0,'E'));
run;

*create the macro variables needed for each period to create the MULTILABEL format;
*i.e. start date, end date, and label for that period of time;
data _null_;
   do i=&qtrback. to 1 by -1;
   call symput ('qst'||compress(i), put(intnx('qtr',&dtnd.,-i-2,'B'),date9.));
   call symput ('qnd'||compress(i), put(intnx('qtr',&dtnd.,-i+1,'E'),date9.));
   call symput ('qnm'||compress(i), "CY"||compress(year(intnx('qtr',&dtnd.,
-i+1,'E')))|| "Q"||compress(qtr(intnx('qtr',&dtnd.,-i+1,'E'))));
   end;
run;

*the next macro will be used to write out macro variables from above in the correct
   structure needed for the PROC FORMAT procedure;
%macro myformat();
   %do i=&qtrback. %to 1 %by -1;
     "&&qst&i"d - "&&qnd&i"d = "&&qnm&i"
   %end;
%mend myformat;

*create the MULTILABEL format;
proc format;
   value period (multilabel notsorted)
```

```
      %myformat
      ;
  run;
```

Once this code has been run in SAS, the programmer can check the log to ensure that everything executed properly. The last portion of the SAS log should resemble something like this:

```
17
18   %macro myformat();
19       %do i=&qtrback. %to 1 %by -1;
20           "&&qst&i"d - "&&qnd&i"d = "&&qnm&i"
21       %end;
22   %mend myformat;
23
24   proc format;
25       value period (multilabel notsorted)
26       %myformat
27       ;
MPRINT(MYFORMAT):   "01JAN2011"d - "31DEC2011"d = "CY2011 Q4" "01APR2011"d - "31MAR2012"d =
"CY2012 Q1" "01JUL2011"d - "30JUN2012"d = "CY2012 Q2" "01OCT2011"d - "30SEP2012"d = "CY2012 Q3"
"01JAN2012"d - "31DEC2012"d = "CY2012 Q4" "01APR2012"d - "31MAR2013"d = "CY2013 Q1" "01JUL2012"d
- "30JUN2013"d = "CY2013 Q2" "01OCT2012"d - "30SEP2013"d = "CY2013 Q3" "01JAN2013"d -
"31DEC2013"d = "CY2013 Q4" "01APR2013"d - "31MAR2014"d = "CY2014 Q1" "01JUL2013"d - "30JUN2014"d
= "CY2014 Q2" "01OCT2013"d - "30SEP2014"d = "CY2014 Q3" "01JAN2014"d - "31DEC2014"d = "CY2014
Q4" "01APR2014"d - "31MAR2015"d = "CY2015 Q1" "01JUL2014"d - "30JUN2015"d = "CY2015 Q2"
NOTE: Format PERIOD has been output.
28   run;

NOTE: PROCEDURE FORMAT used (Total process time):
      real time           0.04 seconds
      cpu time            0.01 seconds
```

Log 1. SAS Log Generated by %MYFORMAT()

CREATING THE OUTPUT TABLE

Now that the hardest part has been accomplished, we are ready to realize the fruits of our work by using this MULTILABEL format with an appropriate SAS procedure. As mentioned earlier in the paper, there are three main SAS procedures that allow programmers to utilize MULTILABEL formats; these are PROC SUMMARY, PROC MEANS, and PROC TABULATE, which all have the MLF option.

In this paper, I will be using PROC TABULATE to calculate the values for each of the quarterly moving windows seen in Figure 1. While such a graph can be generated within SAS using PROC GPLOT, I was more concerned with obtaining the table of statistics itself, and left it up to the reader to determine the best method graphing. In order to obtain this table, I employed the use of the Output Delivery System in SAS to generate an HTML file that can be read by Microsoft Excel.[2] The macro variable “lib” referenced in the ODS HTML FILE statement can be any output library the programmer wants to set using a %LET statement. Other programmers may prefer to use another means of outputting this table (I highly suggest looking into ODS TAGSETS.EXCELXP).

```
  ods html file="&lib.\SESUG Moving Window Calculations.xls" style=minimal;

  proc tabulate data= Master_Dataset;
      class Product_Type Salmonella_Test;
      class Collect_Date / MLF preloadfmt;
      var positive;
      table Product_Type=''*Collect_Date='', positive='Percent
      Positive'*mean=''*f=percent10.2
               Salmonella_Test='Salmonella Test Result'*n='' all='Total Samples
      Analyzed'*n='' / printmiss;
      format Collect_Date period.;
  run;
```

---

[2] Note that when opening the HTML file for the first time, the user will be asked by the computer to verify that the file is not corrupted and is from a trusted source. This is perfectly fine to accept, and hit “Yes.”

```
ods html close;
```

From the SAS code above, it can be seen that I used the MLF and PRELOADFMT options in a CLASS statement for just the Collect_Date variable. I put all other class variables within a different CLASS statement to avoid confusion, and in case there are other formats the programmer may want to use. For all purposes, we only need the percent positive by quarter to create the line graph seen in Figure 1; however, I also included a summary count of samples positive and negative in the TABLE statement because these statistics are often also reported by FSIS. The HTML file that is created by the PROC TABULATE procedure above, yields the following table (with minimal editing to add headers for the “Product” and “Period” columns).

Product Type Sampled	Period Ending	Percent Positive	Salmonella Test Result		Total Samples Analyzed
			Negative	Positive	
Young Chicken	CY2011 Q4	6.42%	4750	326	5076
	CY2012 Q1	5.51%	5089	297	5386
	CY2012 Q2	4.99%	7059	371	7430
	CY2012 Q3	4.14%	8858	383	9241
	CY2012 Q4	4.31%	10462	471	10933
	CY2013 Q1	4.15%	12489	541	13030
	CY2013 Q2	3.89%	12985	526	13511
	CY2013 Q3	4.03%	11683	491	12174
	CY2013 Q4	3.88%	10650	430	11080
	CY2014 Q1	3.98%	8353	346	8699
	CY2014 Q2	3.85%	8493	340	8833
	CY2014 Q3	3.56%	8820	326	9146
	CY2014 Q4	3.64%	8479	320	8799
	CY2015 Q1	3.62%	8406	316	8722
	CY2015 Q2	4.16%	5364	233	5597
Young Turkey	CY2011 Q4	2.31%	1568	37	1605
	CY2012 Q1	2.39%	1796	44	1840
	CY2012 Q2	2.30%	2171	51	2222
	CY2012 Q3	2.05%	2195	46	2241
	CY2012 Q4	2.20%	2135	48	2183
	CY2013 Q1	2.02%	2186	45	2231
	CY2013 Q2	2.28%	2319	54	2373
	CY2013 Q3	2.50%	2337	60	2397
	CY2013 Q4	2.28%	2357	55	2412
	CY2014 Q1	2.16%	2220	49	2269
	CY2014 Q2	2.03%	2028	42	2070
	CY2014 Q3	1.96%	1998	40	2038
	CY2014 Q4	1.67%	1881	32	1913
	CY2015 Q1	1.79%	1753	32	1785
	CY2015 Q2	1.44%	1369	20	1389

Table 2. HTML Output Derived by PROC TABULATE with MLF Option

Once this table is generated, creating the accompanying line graph, as seen in Figure 1, in Microsoft Excel is quite easy, but the opportunity to use PROC GPLOT still exists for those programmers who are ambitious! If this were an automated report, or if it were required on a more routine basis, I would probably employ SAS to create the graph; however, FSIS only asks for the data in Table 2 on a regular basis, and the accompanying line graph is not required. Therefore, I chose to use Excel since it seemed intuitive given the structure of the table.

CONCLUSION

When faced with the need to calculate statistics over a moving window of time, the MULTILABEL format options in SAS can be a huge help and timesaver when employed correctly. It removes the need to repetitively run the same procedure on subsets of data and then append the results. Programmers are strongly advised to make sure any SAS procedure they use supports MULTILABEL formats (i.e. has the MLF option within its syntax). Otherwise, the results from the output will be incomplete.

ACKNOWLEDGMENTS

I offer thanks to my colleagues at the Food Safety and Inspection Service for their continuous support, and help in solving new coding challenges as they arise. I would especially like to thank Christopher Alvares, Director of the Data Analysis and Integration Staff, for his encouragement to write and present this paper.

CONTACT INFORMATION

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

Christopher M. Aston
USDA – Food Safety and Inspection Service
Patriot's Plaza
355 E Street, SW #9-147B
Washington, DC 20024
Work Phone: (202) 690-3986
E-mail: Christopher.Aston@fsis.usda.gov