

Paper 153

Streamlining Medicaid Enrollment Reporting and Calculation of Enrollment Trends Using Macro and PROC Statements in SAS

Deniz Soyer, MBA, District of Columbia Department of Health Care Finance

ABSTRACT

The Division of Analytics and Policy Research (DAPR) within the District of Columbia Department of Health Care Finance (DHCF) produces a monthly Medicaid enrollment report for the District's Medical Care Advisory Committee (MCAC) to document enrollment trends in the District's various Fee-For-Service and Managed Care programs. This retrospective report requires retrieving Medicaid eligibility data for multiple monthly enrollment spans, organization of the data by program type and month, and calculation of enrollment growth for programs of interest. Previously, DAPR used Microsoft Excel to organize multiple outputs of monthly enrollment data and perform calculations on trends. To minimize time spent to manually produce recurring reports, DAPR sought to develop a SAS program to streamline and automate its monthly report.

The use of %LET statements created macro variables to represent each monthly enrollment span, which allowed for the formatting and transposing steps, occurring later in the program, to automatically reference each month. Functions PUT and %SCAN were used to reference and redefine variable formats. PROC FREQ was used to obtain total enrollment counts, by month, for each Medicaid program. PROC TRANSPOSE was used to convert each month from an observation into a variable type, which allowed each row to display enrollment for each Medicaid program type, while the columns served to classify enrollment by month. DATA steps were used to perform calculations for enrollment trends.

INTRODUCTION

DHCF is responsible for administering the District of Columbia's Medicaid program, which is the joint federal-state health insurance program providing health care coverage for low-income and disabled adults, children, and families. Eligibility for D.C. Medicaid is determined through a series of financial and non-financial requirements. Individuals eligible for D.C. Medicaid may be enrolled in a Fee-for-Service (FFS) program, Managed Care (MCO) program and may be dually eligible for Medicare. DHCF also administers the locally-funded D.C. Healthcare Alliance program, which serves low-income District residents who have no other health insurance and are not eligible for either Medicaid or Medicare.

DAPR is housed within DHCF's Health Care Policy and Research Administration. DAPR's roles and responsibilities include developing and conducting original data analysis, including research around health services utilization, expenditures, and program enrollment. Each month, DAPR produces an enrollment report for the D.C. Medical Care Advisory Committee (MCAC) monthly meeting. Below is an example of the standard template for the monthly MCAC Report:

MCAC Enrollment Report - July 2015 - DC Medicaid and Alliance Enrollment, March 2014 through March 2015													
Medicaid Fee-For-Service (FFS)	2014-03	2014-04	2014-05	2014-06	2014-07	2014-08	2014-09	2014-10	2014-11	2014-12	2015-01	2015-02	2015-03
Total Fee-for-Service													
Total SSI or SSI Related Bns													
SSI or SSI Related Bns													
EPD Waiver													
DD Waiver													
MFP DD Waiver (874)													
MFP, EPD Physical Disab >65 (854)													
MFP, EPD Physical Disab <65 (814)													
Total Medicaid Duals													
Duals Non-Institutional													
Duals Institutional/LC													
EPD Waiver Duals (853Q+853S)													
DD Waiver Duals (873Q)													
MFP DD Waiver Duals (874Q)													
MFP, EPD Phy Disab > 65 QMB (854Q)													
MFP, EPD Phy Disab <65 QMB (814Q)													
Qualified Medicare Beneficiaries (QMB)													
ADAP and Elderly QMB													
ADAP and Disabled QMB													
Medicaid Managed Care	2014-03	2014-04	2014-05	2014-06	2014-07	2014-08	2014-09	2014-10	2014-11	2014-12	2015-01	2015-02	2015-03
Total Medicaid Managed Care													
Children and Families													
CHIP Children													
HSCSN													
Immigrant Kids Program													
0-133% FPL Expansion SPA													
Over 133-200% Waiver													
Alliance Managed Care	2014-03	2014-04	2014-05	2014-06	2014-07	2014-08	2014-09	2014-10	2014-11	2014-12	2015-01	2015-02	2015-03
Alliance													
Total DHCF Enrollment													
In-Process Spendown													
Key: SSI - Supplemental Security Income EPD - Elderly and Physically Disabled (95% HCBS Waiver) DD - Developmental Disabilities (95% HCBS Waiver) MFP - Money Follows the Person (Rebalancing Demonstration) SPA - State Plan Amendment DHCF Enrollment Growth: March 2014 - March 2015 SSI/SSI-Related Growth (FFS): Dual Eligible Growth (FFS): QMB Growth (FFS): Managed Care Growth (Medicaid): Total DHCF Enrollment Growth: Highlights * EPD Waiver Growth (Annual) * 0-133% SPA Growth (Annual) * 133-200 Waiver Growth (Annual) * HSCSN Growth (Annual)													

Figure 1. MCAC Enrollment Report Template

DC MCAC is a federally mandated committee consisting of stakeholders, including consumers, advocates, community members and providers, who review Medicaid program operations and provide recommendations. The purpose of the monthly report is to provide a high-level portrayal of enrollment figures for the previous thirteen months and to report trends such as enrollment growth for programs of interest.

Up until early 2015, DAPR analysts used Microsoft Excel to organize multiple outputs of monthly enrollment data and perform trend calculations. The lengthy and cumbersome process took several hours to retrieve raw enrollment data, manually organize and manipulate the format, group the program codes into four high-level program categories, and finally calculate enrollment growth.

To minimize time spent to manually produce recurring reports, DAPR sought to develop a SAS program to streamline and automate its monthly report.

SUPPLEMENTING DATA STEPS WITH MACRO STEPS

DEFINING MACRO VARIABLE USING %LET

After retrieving raw enrollment data for the Medicaid and Alliance programs, our goal is to output total number of individuals enrolled in each program code group for the past thirteen months. In enrollment data, FFS program enrollments are grouped separately from MCO and the Alliance programs.

We use a series of %LET statements that will allow us to use generic names for each of the thirteen months of interest for the MCO and Alliance dataset. Each %LET statement defines a macro variable (MonthA through MonthM), each of which is supposed to represent a month of enrollment:

```
%LET MonthA = MCO_BEGIN_DATE LE '31MAR2014'd and MCO_END_DATE GE '01MAR2014'd;
%LET MonthB = MCO_BEGIN_DATE LE '30APR2014'd and MCO_END_DATE GE '01APR2014'd;
```

We repeat this for each month variable until the thirteenth enrollment period of interest:

```
%LET MonthM = MCO_BEGIN_DATE LE '31MAR2015'd and MCO_END_DATE GE '01MAR2015'd;
```

Then we create a series of additional %LET statements to create generic names for each of the thirteen months of interest for the FFS enrollment dataset. Each %LET statement defines a macro variable (Month1 through Month 13), which is supposed to represent a month of enrollment:

```
%LET Month1 = FFS_BEGIN_DATE LE '31MAR2014'd and FFS_END_DATE GE '01MAR2014'd;
%LET Month2 = FFS_BEGIN_DATE LE '30APR2014'd and FFS_END_DATE GE '01APR2014'd;
```

We repeat this for each month variable until the thirteenth enrollment period of interest:

```
%LET Month13 = FFS_BEGIN_DATE LE '31MAR2015'd and FFS_END_DATE GE '01MAR2015'd;
```

The %LET statement allows us to define the dates of interest, which are the only variable values that change when we run this report each month. Any part of our code that required any manual manipulation by a DAPR analyst, we wanted to have up front in our code. Thus, once we defined our period of interest, we could run the full SAS program.

USING DATA STEPS TO GROUP ENROLLMENT DATA

Following our definition of macro variables using the %LET statement, we organized FFS and Medicaid Dual Eligible program codes into the program groups that are in the final output of the MCAC report. Table 1 displays the list of all FFS and Dual program groups:

Medicaid Fee-For-Service	Total Medicaid Duals
Total Fee-for-Service	Duals Non-Institutional
Total SSI or SSI Related Beneficiaries	Duals Institutional LTC
SSI or SSI Related Beneficiaries	EPD Waiver Duals
EPD Waiver	DD Waiver Duals
DD Waiver	MFP DD Waiver Duals
MFP DD Waiver	MFP, EPD Physical Disab Over 65 QMB
MFP, EPD Physical Disab Over 65	MFP, EPD Physical Disab Under 65 QMB
MFP, EPD Physical Disab Under 65	Qualified Medicare Beneficiaries (QMB)

Table 1. Medicaid Fee-for-Service and Dual Eligible Program Groups

Using a DATA step, we grouped program codes by program group using an IN statement embedded in an IF-THEN statement. Below is a partial of example of what the DATA step would resemble:

```
Data Enrollment_FFS2;
Set Enrollment_FFS;
If Program_Code in('PG_1', 'PG_2', 'PG_3', 'PG_4', 'PG_5') then Program_Type =
'Duals_Non_Institutional';
If Program_Code in('PG_100','PG_6') then Program_Type = "EPD_Waiver";
Run;
```

In the example above we only show two of the IF-THEN statements, but in reality these statements are repeated for each series of program code numbers (variable = Program_Code) until all relevant program code numbers are grouped into the appropriate program group type (variable = Program_Type). We grouped program codes by program group using an IN statement embedded in an IF-THEN statement.

REFERENCING MACRO VARIABLE USING %SCAN

The MCAC report requires us to group enrollment numbers by month using the following date format: yyyymm (i.e. March 2014 would be formatted as 201403). To this end, we need to refer back to the generic month variables that we previously created in our %LET statements.

In a DATA step, we would create a new variable called "yymm" whose values would be identified by extracting the month and year portion of the Month1 through Month13 variables created by the %LET statement. In the example below, we define the character variable "yymm" by using %SCAN and PUT functions. %SCAN extracts the desired year and month from each previously defined month1, month2, etc. variable. The PUT function turns them into the desired format. Notice the use of the macro variable we created in our %LET statement. The code string below depicts the process we outlined:

```
Data Enrollment_Month;
Set Enrollment2;
If &Month1 then do;
yymm=put(year(%scan(&month1.,-1)),4.)||put(month(%scan(&month1.,-1)),z2.);
output;
end;

If &Month2 then do;
yymm=put(year(%scan(&month2.,-1)),4.)||put(month(%scan(&month2.,-1)),z2.);
output;
end;
```

We repeat the IF-THEN statement for each macro variable through Month 13:

```
If &Month13 then do;
yymm=put(year(%scan(&month13.,-1)),4.)||put(month(%scan(&month13.,-1)),z2.);
output;
end;
Run;
```

The result of this code string is that for each Medicaid beneficiary we will output an observation for each month in which they were enrolled in the program. For example, in our output we see that there is a Medicaid beneficiary who has been enrolled from November 2013 to the present. As we can see, there is an observation for each month of interest and an appropriate value for each month of enrollment under the variable yymm.

	Program_Type	B_COE_SPN_BEG_DT	B_COE_SPN_END_DT	B_COE_CD	yymm
28	One_Thirty_Three...	11/01/2013	12/31/9999	774	201404
29	One_Thirty_Three...	11/01/2013	12/31/9999	774	201405
30	One_Thirty_Three...	11/01/2013	12/31/9999	774	201406
31	One_Thirty_Three...	11/01/2013	12/31/9999	774	201407
32	One_Thirty_Three...	11/01/2013	12/31/9999	774	201408
33	One_Thirty_Three...	11/01/2013	12/31/9999	774	201409
34	One_Thirty_Three...	11/01/2013	12/31/9999	774	201410
35	One_Thirty_Three...	11/01/2013	12/31/9999	774	201411
36	One_Thirty_Three...	11/01/2013	12/31/9999	774	201412
37	One_Thirty_Three...	11/01/2013	12/31/9999	774	201501
38	One_Thirty_Three...	11/01/2013	12/31/9999	774	201502
39	One_Thirty_Three...	11/01/2013	12/31/9999	774	201503
40	Other	02/01/2014	05/31/2015	120	201403

Output 1. Creating an Observation for Each Month of Enrollment

PROC FREQ AND PROC SQL IN ENROLLMENT COUNTS

Since we now have observations for each month of enrollment for each Medicaid beneficiary, we can now count the total number of beneficiaries enrolled in each month for total enrollment numbers by using PROC FREQ. The PROC FREQ output will then be followed by a DATA step creating a new variable called "Program_Type" and the output will look like this:

	Program_Type	yymm	COUNT
1	Total_Enrollment_Numbers	201403	252350
2	Total_Enrollment_Numbers	201404	254929
3	Total_Enrollment_Numbers	201405	256396
4	Total_Enrollment_Numbers	201406	257681
5	Total_Enrollment_Numbers	201407	259260
6	Total_Enrollment_Numbers	201408	260733
7	Total_Enrollment_Numbers	201409	262271
8	Total_Enrollment_Numbers	201410	263480
9	Total_Enrollment_Numbers	201411	263784
10	Total_Enrollment_Numbers	201412	264647
11	Total_Enrollment_Numbers	201501	265408
12	Total_Enrollment_Numbers	201502	265565
13	Total_Enrollment_Numbers	201503	264862

Output 2. PROC FREQ Output of Total Enrollment Numbers

Similarly, we use PROC FREQ to count the number of beneficiaries within each program type for each month of interest. Here is a sample of our partial output to give you a sense of what this looks like:

	Program_Type	yymm	COUNT
25	ADAP_and_Elderly_QMB	201502	41
26	ADAP_and_Elderly_QMB	201503	39
27	CHIP_Children	201403	6705
28	CHIP_Children	201404	6786
29	CHIP_Children	201405	6833
30	CHIP_Children	201406	6870
31	CHIP_Children	201407	6943
32	CHIP_Children	201408	6995
33	CHIP_Children	201409	7051
34	CHIP_Children	201410	7117
35	CHIP_Children	201411	7139
36	CHIP_Children	201412	7160
37	CHIP_Children	201501	7235
38	CHIP_Children	201502	7391
39	CHIP_Children	201503	7722
40	Duals_Institutional_LTC	201403	2411
41	Duals_Institutional_LTC	201404	2422
42	Duals_Institutional_LTC	201405	2392
43	Duals_Institutional_LTC	201406	2397
44	Duals_Institutional_LTC	201407	2384
45	Duals_Institutional_LTC	201408	2352
46	Duals_Institutional_LTC	201409	2358
47	Duals_Institutional_LTC	201410	2373
48	Duals_Institutional_LTC	201411	2369
49	Duals_Institutional_LTC	201412	2391
50	Duals_Institutional_LTC	201501	2412
51	Duals_Institutional_LTC	201502	2373
52	Duals_Institutional_LTC	201503	2371
53	Duals_Non_Institutional	201403	11982

Output 3. PROC FREQ Output of Enrollment Count by Program Type

From here, we combined the two previous outputs into one dataset, and realized we had a small problem. Since the "yymm" variable is a character variable and not a date variable, we had to develop a method to keep each month of interest in ascending order for our report. In other words, it would not be acceptable to report enrollment numbers for each month out of order.

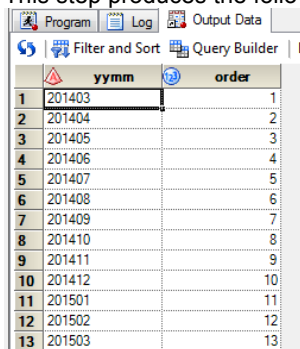
First, we created a brand new temporary table called "tmp" using PROC SQL. The SELECT statement retrieves each "distinct" output of the variable "yymm" from our previous data set. In our case, each distinct output would be each month of interest. We also place each month in order:

```
Proc Sql;
  create table tmp as
  select distinct yymm
  from libname.enrollmentcount;
  order by yymm;
quit;
```

Using a DATA step, we take the output we created in the PROC SQL step and create a new variable called “order,” which adds a value of 1 through 13 for each month:

```
data libname.order;
  set tmp;
  order = _N_;
run;
```

This step produces the following output:



	yymm	order
1	201403	1
2	201404	2
3	201405	3
4	201406	4
5	201407	5
6	201408	6
7	201409	7
8	201410	8
9	201411	9
10	201412	10
11	201501	11
12	201502	12
13	201503	13

Output 4. PROC SQL Output Specifying the Correct Order of Each Month

PROC TRANSPOSE AND A MACRO CALLED %TRANSPOSE

The ordered logic that we created using the PROC SQL and DATA step combination in the previous section, needs to be merged back into our dataset containing the enrollment counts for each program type by month. In addition, we also wanted to transpose the variable “yymm,” so that each month of interest is converted from an observation associated with a program type into its own variable. In order to allow us to do this, we used the following macro called %transpose developed by Arthur Tabachneck, Xia Ke Shan, Robert Virgile, and Joe Whitehurst:

```
%transpose(data=libname.Enrollment_Merge,
           out=libname.Enrollment_Merge2,
           by=Program_Type,
           id=yymm,
           delimiter=_,
           var=Count,
           use_varname=,
           preloadfmt=libname.order)
```

Here’s how this macro works:

- The “data=” line refers to our dataset containing the enrollment counts for each program type.
- The “out=” line refers to the dataset name of our macro output.
- The “by=” line refers to the variable by which we want to group the data, in our case “Program_Type”
- The “id” line refers to the variable that we want to transpose, which in our case is “yymm” – our months of interest
- The “delimiter=” line allows us to place an underscore “_” before the variable name, which is specified in the next line
- The “var=” line allows us to place “Count” or other text before each month. We need to do this because no SAS variable name that we ever want to reference in future DATA steps or PROC steps can start with a number. This way, we allow “Count_” to precede each “yymm” value.
- The “preloadfmt=” line refers to the macro to the dataset containing the ordered month variables.

Our output from this macro is a dataset called "libname.Enrollment_Merge2" and contains each Program_Type and each month of interest in order:

Program_Type	COUNT_201403	COUNT_201404	COUNT_201405	COUNT_201406	COUNT_201407	COUNT_201408	COUNT_201409	COUNT_201410	COUNT_201411	COUNT_201412	COUNT_201501	COUNT_201502	COUNT_201503
ADAP_and_Disabled_GMB	191	192	192	193	197	175	161	164	167	171	171	170	166
ADAP_and_Elderly_GMB	41	41	41	44	46	40	39	39	39	38	39	41	39
CHIP_Children	6705	6786	6833	6870	6943	6995	7051	7117	7139	7160	7235	7391	7722
Duals_Institutional_LTC	2411	2422	2392	2397	2384	2352	2358	2373	2369	2391	2412	2373	2371
Duals_Non_Institutional	11982	12025	11998	11967	12004	12010	12031	12003	12042	12031	12017	11965	11903
EPD_Waiver_Duals	2246	2256	2239	2263	2300	2326	2336	2337	2295	2274	2223	2206	2202
Immigrant_Kids_Program	2391	2411	2440	2505	2550	2650	2729	2767	2800	2782	2758	2767	2782
In_Process_Spenddown	526	539	523	544	535	545	551	549	553	539	557	548	541
One_Thirty_Three_FPL_SPA	49425	50304	50746	51294	51829	52235	52732	53134	53310	53882	54304	55069	55596

Output 5. Monthly Enrollment Count for Each Program Type in Chronological Order

In our final output we do not want to have "Count_" preceding each of our months of interest. In order to correct the variable names, we must transpose the data using the simple PROC TRANSPOSE step:

```
Proc Transpose data=libname.Enrollment_Merge2 out=libname.Enrollment_Merge3;
ID Program_Type;
Run;
```

Here is the output from this step:

NAME	COUNT_201403	COUNT_201404	COUNT_201405	COUNT_201406	COUNT_201407	COUNT_201408	COUNT_201409	COUNT_201410	COUNT_201411	COUNT_201412	COUNT_201501	COUNT_201502	COUNT_201503
ADAP_and_Disabled_GMB	191	192	192	193	197	175	161	164	167	171	171	170	166
ADAP_and_Elderly_GMB	41	41	41	44	46	40	39	39	39	38	39	41	39
CHIP_Children	6705	6786	6833	6870	6943	6995	7051	7117	7139	7160	7235	7391	7722
Duals_Institutional_LTC	2411	2422	2392	2397	2384	2352	2358	2373	2369	2391	2412	2373	2371
Duals_Non_Institutional	11982	12025	11998	11967	12004	12010	12031	12003	12042	12031	12017	11965	11903
EPD_Waiver_Duals	2246	2256	2239	2263	2300	2326	2336	2337	2295	2274	2223	2206	2202
Immigrant_Kids_Program	2391	2411	2440	2505	2550	2650	2729	2767	2800	2782	2758	2767	2782
In_Process_Spenddown	526	539	523	544	535	545	551	549	553	539	557	548	541
One_Thirty_Three_FPL_SPA	49425	50304	50746	51294	51829	52235	52732	53134	53310	53882	54304	55069	55596

Output 6. PROC TRANSPOSE Enrollment Months of Interest

This output shows that all program types have been converted from observations into variables, while a new variable called "_NAME_" is automatically created containing each month of interest. We use a DATA step featuring a COMPRESS function to remove "Count_" preceding each month of interest. Essentially, this step allows us to redefine the variable "_NAME_" by setting it equal to the COMPRESS function, which removes the text "Count_" from any value under the variable "_NAME_."

```
Data libname.Enrollment_Merge4;
Set libname.Enrollment_Merge3;
_NAME_ = Compress(_NAME_,'COUNT_');
Run;
```

At this point, we want to convert our months of interest from observations into variables once again. We use PROC TRANSPOSE to transpose our dataset by the variable "_NAME_":

```
Proc Transpose data=libname.Enrollment_Merge4 out=libname.Enrollment_Merge5;
ID _NAME_;
Run;
```

Here is the output from this step:

NAME	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412	201501	201502	201503
ADAP_and_Disabled_GMB	191	192	192	193	197	175	161	164	167	171	171	170	166
ADAP_and_Elderly_GMB	41	41	41	44	46	40	39	39	39	38	39	41	39
CHIP_Children	6705	6786	6833	6870	6943	6995	7051	7117	7139	7160	7235	7391	7722
Duals_Institutional_LTC	2411	2422	2392	2397	2384	2352	2358	2373	2369	2391	2412	2373	2371
Duals_Non_Institutional	11982	12025	11998	11967	12004	12010	12031	12003	12042	12031	12017	11965	11903
EPD_Waiver_Duals	2246	2256	2239	2263	2300	2326	2336	2337	2295	2274	2223	2206	2202
Immigrant_Kids_Program	2391	2411	2440	2505	2550	2650	2729	2767	2800	2782	2758	2767	2782
In_Process_Spenddown	526	539	523	544	535	545	551	549	553	539	557	548	541
One_Thirty_Three_FPL_SPA	49425	50304	50746	51294	51829	52235	52732	53134	53310	53882	54304	55069	55596

Output 7. PROC TRANSPOSE Output with Finalized Month Values

We will use a DATA step containing the RENAME function to convert the variable _NAME_ to the variable "Program_Type" for our final enrollment count output:

Program_Type	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412	201501	201502	201503
1 ADAP_and_Disab...	191	192	192	193	197	175	161	164	167	171	171	170	16
2 ADAP_and_Elderl...	41	41	41	44	46	40	39	39	39	38	39	41	1
3 CHIP_Children	6705	6786	6833	6870	6943	6995	7051	7117	7139	7160	7235	7391	77
4 Duals_Institutional...	2411	2422	2392	2397	2384	2352	2368	2373	2369	2391	2412	2373	23
5 Duals_Non_Institu...	11982	12025	11998	11967	12004	12010	12031	12003	12042	12031	12017	11965	119
6 EPD_Waiver_Dua...	2246	2256	2239	2263	2300	2326	2336	2337	2295	2274	2223	2206	22
7 Immigrant_Kids_P...	2391	2411	2440	2505	2550	2650	2729	2767	2800	2782	2758	2767	27
8 In_Process_Spen...	526	539	523	544	535	545	551	549	553	539	557	548	5
9 One_Thirty_Three...	49425	50304	50746	51294	51829	52235	52732	53134	53310	53882	54304	55069	558

Output 8. Use of RENAME function to convert variable name

If we remember back to our original data sources, the FFS raw enrollment data was separate from the MCO and Alliance raw enrollment data. Table 2 displays the list of all Managed Care and Alliance program groups:

Medicaid Managed Care and Alliance
Total Medicaid Managed Care (TMC)
Children and Families
CHIP Children
HSCSN
Immigrant Kids Program
0-133% FPL Expansion SPA
Over 133-200% Waiver
Alliance

Table 2. Medicaid Managed Care and Alliance Program Groups

Enrollment associated with Medicaid Managed Care and Alliance program is organized into these groups

Our next step is to revisit our MCO and Alliance raw dataset and repeat each of the steps that we applied to the FFS dataset, such as referencing the macro variables MonthA through MonthM, PROC FREQ to count the enrollment by MCO and Alliance program type, using PROC SQL and a data step to place the months of interest in ascending order, and using PROC TRANSPOSE steps to put our enrollment numbers in the correct format. Upon completion of these steps, our final enrollment output for the MCO and Alliance enrollment resembles the format of our final FFS enrollment dataset.

FINALIZING THE MCAC REPORT

ENROLLMENT DATA OUTPUT

Once our MCO and FFS enrollment outputs are combined, we use a final DATA step containing a RENAME function to convert the variable _NAME_ to the variable "Program_Type" for our final enrollment count output:

Program_Type	201403	201404	201405	201406	201407	201408	201409	201410	201411	201412	201501	201502	201503
1 Total_FFS	72224	72147	70594	68583	68329	67885	67706	67234	66654	66460	66535	67255	67919
2 TSSI	46278	46102	44550	42569	42246	41682	41331	40833	40219	39977	40063	40862	41620
3 SSI	44815	44638	43085	41095	40748	40182	39837	39341	38700	38486	38559	39374	40143
4 P853	554	556	555	557	564	560	557	557	546	545	548	529	519
5 P873	904	903	905	912	930	935	931	930	937	941	951	954	953
6 P874	3	3	3	3	2	2	3	3	3	2	2	2	2
7 P854	2	2	2	2	2	2	2	1	2	2	2	2	2
8 P814						1		1	1	1	1	1	1
9 Total_Medicaid_Duals	17336	17401	17328	17328	17390	17391	17438	17432	17426	17423	17379	17271	17207
10 Duals_Non_Institutional	11982	12025	11998	11967	12004	12010	12031	12003	12042	12031	12017	11965	11903
11 Duals_Institutional_LTC	2411	2422	2392	2397	2384	2352	2368	2373	2369	2391	2412	2373	2371
12 EPD_Waiver_Duals	2246	2256	2239	2263	2300	2326	2336	2337	2295	2274	2223	2206	2202
13 P873Q	688	691	692	693	692	692	700	704	704	707	707	709	711
14 P874Q										1	1	1	1
15 P854Q	2	2	3	3	4	5	5	6	6	7	6	5	6
16 P814Q	7	5	4	5	6	6	8	9	10	12	13	12	13
17 QMB	8610	8644	8716	8686	8693	8813	8937	8969	9009	9060	9093	9122	9092
18 ADAP_and_Elderly_QMB	41	41	41	44	46	40	39	39	38	39	39	41	39
19 ADAP_and_Disabled_QMB	191	192	192	193	197	175	161	164	167	171	171	170	166
20 TMC	162688	165145	167877	171078	172963	174625	176276	177849	178851	180098	180763	180130	178769
21 Children_and_Families	91863	92927	94943	97312	98406	99387	100305	101235	101881	102401	102425	100693	98239
22 CHIP_Children	6705	6786	6833	6870	6943	6995	7051	7117	7139	7160	7235	7391	7722
23 HSCSN	5699	5749	5787	5792	5828	5831	5822	5830	5851	5843	5884	5875	5848
24 Immigrant_Kids_Program	2391	2411	2440	2505	2550	2650	2729	2767	2800	2782	2758	2767	2782
25 One_Thirty_Three_FPL_S...	49425	50304	50746	51294	51829	52235	52732	53134	53310	53882	54304	55069	55596
26 Over_Thirty_Three_Waiver	5605	5668	5728	5795	5867	5937	5997	6066	6135	6204	6273	6342	6411
27 Alliance	14465	14597	14722	14895	15002	15175	15276	15432	15544	15706	15868	16030	16192
28 Total_DHCF_Enrollment	249377	251889	253243	254556	256294	257686	259257	260515	262049	263647	265281	266945	268609
29 In_Process_Spendown	526	539	523	544	535	545	551	549	553	539	557	548	541

Output 9. Final MCAC Enrollment Count Output

ENROLLMENT TREND REPORTING

In addition to reporting monthly enrollment numbers, the MCAC report also requires that DAPR analysts report enrollment trends. Specifically, MCAC members are interested in enrollment growth in select programs over the past 13 months. We can use a couple of DATA and PROC TRANSPOSE steps to output these figures.

First, we will replace each month value with a numerical assignment. Each month will be assigned a number in chronological order of the month's occurrence. To do this, we will create a new variable called "yymm." In addition, since the MCAC members are interested in specific programs, we have to use a keep statement to restrict our calculations to the variables of interest. I will only list a few of the variables of interest for this example, instead of providing the full list of variables of interest.

```
Data libname.calculation
(keep = yymm Total_Medicaid_Duals TMC Total_DHCF_Enrollment);
Set libname.FinalMCAC;
yymm = _n_;
Run;
```

	TMC	Total_DHCF_Enrollment	Total_Medicaid_Duals	yymm
1	162688	249377	17336	1
2	165145	251889	17401	2
3	167877	253243	17328	3
4	171078	254556	17328	4
5	172963	256294	17390	5
6	174625	257686	17391	6
7	176276	259257	17438	7
8	177849	260515	17432	8
9	178851	260849	17426	9
10	180098	261834	17423	10
11	180763	262467	17379	11
12	180130	262581	17271	12
13	178769	262045	17207	13

Output 10. Creation of 'yymm' Variable Replacing Each Month Value with Numerical Assignment

Now, we need to transpose our data so that we can calculate the difference in enrollment between month 1 and 13. Using PROC TRANSPOSE, we will transpose while keeping the months in ascending order. In addition, we will also use the PREFIX function to place "month" in front of "yymm" observation. In other words yymm = "1" will get converted to yymm = "months1."

```
Proc Transpose data=libname.calculation Out= libname.calculation2
Prefix = months;
ID yymm;
Run;
```

Finally, using a DATA step, we can create a new variable called "Growth," which will represent the percent change from month 1 to month 13 for each program type. We will use the RENAME option to change the automatically assigned name of the variable "_NAME_" to be "Program_Type." Using FORMAT option, we will format the "Growth" variable to be a percent rounded to 2 decimal places. In this output, we are only interested in seeing the enrollment growth for each program type of interest, so we can remove the variables months1 through months13 once we have used them to calculate "Growth."

```
Data libname.calculation_final (rename=(_NAME_ = Program_Type));
Set libname.calculation2;
Growth = (months13 - months1)/months1;
Format Growth percent8.2;
Drop months1-months13;
Run;
```

	Program_Type	Growth
1	TMC	9.88%
2	Total_DHCF_Enrollment	5.08%
3	Total_Medicaid_Duals	(0.74%)

Output 11. Final MCAC Enrollment Growth Calculations

Our last step is to output these values to Microsoft Excel, and paste them into our standard MCAC template. Here is the final version of the MCAC report as it is provided to the MCAC committee members and the public:

MCAC Enrollment Report - July 2015 - DC Medicaid and Alliance Enrollment, March 2014 through March 2015													
Medicaid Fee-For-Service (FFS)	2014-03	2014-04	2014-05	2014-06	2014-07	2014-08	2014-09	2014-10	2014-11	2014-12	2015-01	2015-02	2015-03
Total Fee-for-Service	72,184	72,102	70,542	68,520	68,253	67,796	67,602	67,104	66,508	66,256	66,269	66,852	67,346
Total SSI or SSI Related Bns	46,246	46,065	44,507	42,516	42,179	41,601	41,239	40,715	40,088	39,791	39,819	40,491	41,103
SSI or SSI Related Bns	44,783	44,601	43,042	41,042	40,681	40,101	39,745	39,224	38,600	38,301	38,317	39,005	39,628
EPD Waiver	554	556	555	557	564	560	557	556	545	544	546	527	518
DD Waiver	904	903	905	912	930	935	931	930	937	941	951	954	952
MFP DD Waiver (874)	3	3	3	3	2	2	3	3	3	2	2	2	2
MFP, EPD Physical Disab >65 (854)	2	2	2	2	2	2	2	1	2	2	2	2	2
MFP, EPD Physical Disab <65 (814)	-	-	-	-	-	1	1	1	1	1	1	1	1
Total Medicaid Duals	17,329	17,394	17,321	17,320	17,381	17,382	17,426	17,420	17,412	17,406	17,356	17,240	17,162
Duals Non-Institutional	11,974	12,018	11,991	11,959	11,995	12,001	12,020	11,993	12,032	12,018	12,001	11,949	11,880
Duals Institutional/LTC	2,412	2,422	2,392	2,397	2,384	2,352	2,357	2,372	2,367	2,389	2,406	2,359	2,352
EPD Waiver Duals (853Q+853S)	2,246	2,256	2,239	2,263	2,300	2,326	2,336	2,336	2,293	2,272	2,222	2,205	2,199
DD Waiver Duals (873Q)	688	691	692	693	692	692	700	704	704	707	707	709	711
MFP DD Waiver Duals (874Q)	-	-	-	-	-	-	-	-	-	1	1	1	1
MFP, EPD Phy Disab > 65 QMB (854Q)	2	2	3	3	4	5	5	6	6	7	6	5	6
MFP, EPD Phy Disab <65 QMB (814Q)	7	5	4	5	6	6	8	9	10	12	13	12	13
Qualified Medicare Beneficiaries (QMB)	8,609	8,643	8,714	8,684	8,693	8,813	8,937	8,969	9,008	9,059	9,094	9,121	9,081
ADAP and Elderly QMB	41	41	41	44	46	40	39	39	39	38	39	41	39
ADAP and Disabled QMB	191	192	192	193	197	175	161	164	167	171	171	170	166
Medicaid Managed Care	2014-03	2014-04	2014-05	2014-06	2014-07	2014-08	2014-09	2014-10	2014-11	2014-12	2015-01	2015-02	2015-03
Total Medicaid Managed Care	162,685	165,143	167,875	171,078	172,962	174,624	176,275	177,847	178,850	180,097	180,763	180,130	178,770
Children and Families	91,884	92,950	94,970	97,347	98,455	99,447	100,376	101,324	101,985	102,534	102,594	100,938	98,528
CHIP Children	6,706	6,787	6,833	6,870	6,942	6,993	7,049	7,116	7,138	7,157	7,228	7,380	7,708
HSCSN	5,699	5,749	5,787	5,792	5,828	5,831	5,822	5,830	5,851	5,843	5,884	5,875	5,848
Immigrant Kids Program	2,391	2,411	2,440	2,505	2,550	2,649	2,728	2,766	2,799	2,781	2,755	2,764	2,773
0-133% FPL Expansion SPA	49,404	50,281	50,723	51,263	51,787	52,187	52,878	53,064	53,231	53,785	54,184	54,910	55,411
Over 133-200% Waiver	6,601	6,965	7,122	7,299	7,400	7,517	7,622	7,747	7,846	7,997	8,118	8,263	8,502
Alliance Managed Care	2014-03	2014-04	2014-05	2014-06	2014-07	2014-08	2014-09	2014-10	2014-11	2014-12	2015-01	2015-02	2015-03
Alliance	14,465	14,597	14,773	14,895	15,001	15,174	15,274	15,431	15,343	15,275	15,169	15,197	15,357
Total DHCF Enrollment	249,334	251,842	253,190	254,491	256,216	257,594	259,151	260,382	260,701	261,628	262,201	262,179	261,473
In-Process Spendown	526	539	523	544	535	545	551	549	553	539	557	549	543
Key:	DHCF Enrollment Growth: March 2014 - March 2015							Highlights					
SSI - Supplemental Security Income	SSI/SSI-Related Growth (FFS):							* EPD Waiver Growth					
EPD - Elderly and Physically Disabled (815c HCBS Waiver)	Dual Eligible Growth (FFS):							* 0-133% SPA Growth					
DD - Developmental Disabilities (815c HCBS Waiver)	QMB Growth (FFS):							* 133-200 Waiver Grow					
MFP - Money Follows the Person (Rebalancing Demonstration)	Managed Care Growth (Medicaid):							* HSCSN Growth					
SPA - State Plan Amendment	Total DHCF Enrollment Growth:												

Figure 2. Finalized MCAC Enrollment Report

REVIEW OF METHODOLOGY

Data steps were combined with macro and PROC statements to develop a SAS program for reporting. %LET statements created macro variables to represent each monthly enrollment span, which allowed for the formatting and transposing steps, occurring later in the program, to automatically reference each month. PUT and %SCAN functions were used to reference and redefine variable formats. PROC FREQ was used to obtain total enrollment counts, by month, for each Medicaid program. PROC TRANSPOSE was used to convert each month from an observation into a variable type, which allowed for each row to display enrollment for each Medicaid program type, while the columns served to classify enrollment by month. Finally, we used a series of DATA steps to calculate enrollment growth for Medicaid and Alliance programs of interest.

Using essential macro and PROC statements in SAS, DAPR was able to organize data, calculate trends and output a finalized report related to monthly Medicaid program enrollment, thereby resulting in near-automation in its reporting process.

CONCLUSION

Adopting the use of macro statements alongside data steps and PROC statements in SAS enables greater automation and accuracy in periodic reporting. DAPR has incorporated the combined use of macro and PROC statements in an effort to streamline other recurring reports.

REFERENCES

Cody Ronald P. 2012. *Learning SAS by Example: A Programmer's Guide*. Cary, NC: SAS Institute Inc.

Tabachneck, Arthur S. Shan, Xia Ke. Virgile, Robert. Whitehurst, Joe. 2013. "A Better Way to Flip (Transpose) a SAS Dataset. *SAS Global Forum 2013*. Available at: <http://support.sas.com/resources/papers/proceedings13/538-2013.pdf>

ACKNOWLEDGMENTS

Many thanks to Yiping Zhao and John Wedeles for their contributions in reviewing this paper. Their suggestions and feedback were invaluable.

CONTACT INFORMATION

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

Name: Deniz Soyer

Enterprise: District of Columbia Department of Health Care Finance, Division of Analytics and Policy Research

Address: 441 4th Street NW, Suite 900S

City, State ZIP: Washington, DC 20001

Work Phone: 202-442-4625

E-mail: deniz.soyer@gmail.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.