Paper 180-2011

After SAS® Performs the Surgery, Excel® Applies the Make-Up: Making Healthcare Data Look Pretty

Andrea Scott, CareFirst BlueCross BlueShield

ABSTRACT

Sometimes we are asked to perform what is seemingly impossible: turn Healthcare data into a simple and understandable message. How is it possible to make Medical or Pharmacy claims, or enrollment data look pretty? What better way to do this than by breaking the data down to its simplest form (not necessarily a simple task) then transforming the numbers into colors. Shading can be used to denote intensity of dosage or addition of claims plus enrollment; contrasts can be used to compare populations. There are times when words hinder the message. The visuals speak for themselves. Numbers aren't always the optimal way to communicate analytic results.

INTRODUCTION

Working at a health insurance company, data takes many forms. Most of the time Medical (and/or Pharmacy) claims or Medical (and/or Pharmacy) insurance product enrollment, or eligibility, data is used; often they are used together. Some studies require continuous Medical (or Pharmacy) enrollment; others, simultaneous Medical and Pharmacy enrollment. For one particular study, I wanted to describe, visually, each patient's enrollment pattern while at the same time describing their claims patterns. The purpose was to be able to discern at one glance whether a patient was taking medication regularly; and, enrollment information was necessary to understand at what points during the study would there be claims data available for that patient. In my example, I describe the steps taken to transform each patient's enrollment and claims information from SAS data sets to a single, colorized, Excel spreadsheet. The small population size lent itself to this type of display, though is not typical in the health insurance setting; however, this particular condition has a small prevalence rate.

BACKGROUND

Both enrollment and claims records in a health insurance company contain numerous fields. However, to achieve our goal here, only a few are needed. The fields needed from the enrollment tables are member id and enrollment date (in month, year form). Member id can take different forms depending how the individual insurance company chooses to store the member's identification information. In this example, I will use two separate fields, contract holder id and member number, as this was the case in my original study. The fields needed from the Medical claims table are member id and service date. The same two separate fields are used as in the enrollment table. After pulling the data from the tables it looks like this:

Enrollment

contract_holder_id	member_number	enroll_date
100000001	232000000	012007
100000001	232000000	022007
100000001	232000000	032007
100000002	484000000	052007
100000002	484000000	062007

.

.

Medical Claims

contract_holder_id	member_number	service_date
100000001	232000000	01/18/2007
100000001	232000000	02/25/2007
100000002	484000000	06/17/2007

.
.

There are multiple rows for each patient with one column for either the enrollment date or service date of the claim. What I needed was one row per patient where the columns represented each month-year combination occurring during the study period. This would be accomplished with a PROC TRANSPOSE. However, I first needed to create a combined data set with a unique patient identifier and an indicator that will force the transposed data set to create month-year columns for each row in the original data set where the column headings would be created from the combined enrollment and service date values.

I then created formats for this indicator variable that would give my resulting Excel spreadsheet the display I desired. For the study I am using in my example, I wanted to portray a patient's enrollment and at what point during their enrollment they experienced these particular Medical claims. Each cell represents a patient's enrollment month throughout the entire study period. I decided to use shades of blue with a lighter shade representing enrollment only and a darker shade representing an occurrence of a claim during that enrollment month; lack of color indicates a month with neither enrollment nor claims.

METHODS

First I created a temporary data set using the following Base SAS code:

```
DATA temp_enr(DROP= contract_holder_id member_number);

       SET here.enrollment_file;

       id = contract_holder_id||member_number;

       FORMAT enroll_date monyy7.;


RUN;
```

Above, a unique patient identifier is created and the enrollment date field, enroll_date, has been reformatted so later the new columns will be easier to read. Now, the temporary enrollment file looks like this (except the id values may contain extra spaces depending on the length of the original individual member id fields):

id	enroll_date
100000001232000000	JAN2007
100000001232000000	FEB2007
100000001232000000	MAR2007
100000002484000000	MAY2007
100000002484000000	JUN2007

.
.

The equivalent step is completed for the Medical claims data set. Next, the data sets need to be sorted then merged. It is in the data step where the indicator variable will be created.

```
PROC SORT DATA=temp_enr OUT=temp_enr_srt;
       BY id enroll_date;
RUN;

PROC SORT DATA=temp_clm NODUPKEY OUT=temp_clm_srt;
       BY id svc_date;
RUN;

DATA temp_enr_clms;

       MERGE temp_enr_srt(IN=enr RENAME=(enroll_date=mth_yr))
             temp_clms_srt(IN=clm RENAME=(svc_date=mth_yr));

       BY id mth_yr;

        /* create indicator variable to be used for transpose and formats. */

       IF      enr and clm     THEN ind =  2;          /* both enrollment and claims */

       ELSE IF enr and not clm THEN ind =  1;          /* enrollment and no claims */

       ELSE IF NOT enr and clm THEN ind = -1;          /* claims and no enrollment */

RUN;
```

There should not be any months with a claim and without enrollment, an indicator value of -1. If there are, it is a good indication you should go back and check your data.

After this, the merged data set needs to be transposed. Before that, I want to confirm the data sets are sorted by id and date since the resulting columns need to be ordered; this is covered by the DATA step above.

Enrollment information, within each insurance product, appears in the form of one record per month indicating eligibility for that month. Medical (or Pharmacy) claims, on the other hand, could likely appear in the form of multiple records per month; these additional records per month need to be eliminated before transposing the data. Therefore, the PROC SORT performed on the claims data set should include the NODUPKEY statement only if you are interested in whether the patient incurred at least one claim for a particular month.

If the number of claims per month is important, instead of a PROC SORT above, a PROC SQL can be used to count the number of claims per month. That count field can be considered when creating the indicator variable above such that after the data is transposed, the cells contain the number of claims per month. These steps will not be discussed here.

The data is then transposed as follows:

```
PROC TRANSPOSE DATA=temp_enr_clms NAME=id OUT=here.enr_clm_transp;

       ID mth_yr;
       VAR ind;
       BY id;

RUN;
```

The above allows the following to occur: the data set temp_enr_clms containing the column id to be transposed using the NAME statement creating the observations to be contained in the resulting data set here.enr_clm_transp which has columns labeled with the corresponding mth_yr values using the ID statement having the value of ind populating the rows using the VAR statement all for each patient separately using the BY statement. In other words, the resulting data set looks like this:

id	JAN2007	FEB2007	MAR2007	MAY2007	JUN2007

```
100000001232000000      2          2              1              .            .
100000002484000000      .          .              .              1            2
.
.
```

One PROC TRANSPOSE option I did not need here, but want to mention anyway, is the LET statement. It allows for duplicates of the ID variable. I have found it very helpful to transform observations having multiple occurrences within a service date, e.g. procedure codes.

Before the new Excel spreadsheet is created, the merged data set needs to be merged back with a patient list containing the original id fields, or other identifiers such as first and last name, and the new concatenated id so the patients can be properly identified. Alternatively, dummy id's can be created so the final results in Excel will be de-identified to follow HIPAA regulations. These steps will not be described here. They can be performed in Excel as well as Base SAS.

Something to keep in mind is, for the resulting Excel worksheet to be complete, all months occurring during the study need to be represented. Most studies involving health outcomes or prescribing patterns require continuous Medical and/or Pharmacy enrollment during the entire study period. For these cases, completeness of data will not be an issue. However, for those exceptional cases, to ensure complete results, it may be necessary to create a dummy patient with enrollment and/or claims for all months and add that to the relevant data sets before transposing so that all months are represented in the transposed data set.

Now, two formats are created using PROC FORMAT, "color" and "noshow", such that when the spreadsheet is created it displays the visual I wanted: colors, not numbers, used to simultaneously represent enrollment and claims information for each patient.

```
PROC FORMAT;

      VALUE color  /* to indicate enrollment and claims */

                    2   = 'Dark Blue'         /* both enrollment and claims */
                    1   = 'Blue'              /* enrollment and no claims */
                    0,. = 'White'             /* neither enrollment nor claims */
                   -1   = 'Green'             /* claims and no enrollment */
      ;

      VALUE noshow /* suppress display of the indicator value in the spreadsheet */

                   .,1,2,-1 = ' '             /* include all possible values */
      ;

RUN;
```

It appears that PROC TRANSPOSE creates the columns in the order it encounters them in the data sets inputted into the statement. Meaning, if the first observation for the first patient in the data set does not have an observation for the first month of the study period, i.e. JAN2007, that column will not be created until it is encountered through a later patient. This can be achieved listing them in the proper order below. The resulting worksheet should be carefully checked to ensure the columns are in chronological order.

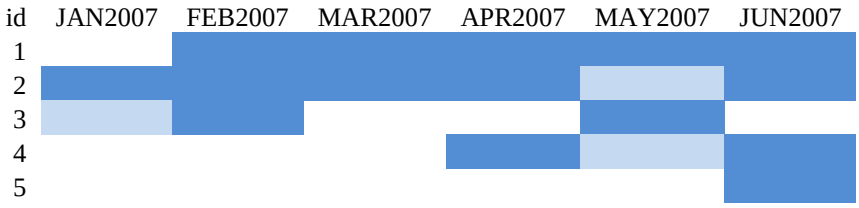
The transposed enrollment and claim data set is now used to create an Excel spreadsheet using the ExcelXP tagset.

```
ODS tagsets.excelxp FILE='C:\NESUG\Enrollment and Claims Summary.xls" STYLE=minimal;

PROC PRINT DATA=here.enr_clm_transp NOOBS LABEL;

      VAR id;

      VAR JAN2007--DEC2008 / STYLE=[background=color.] ;

      FORMAT JAN2007--DEC2008 noshow. ;    /* suppress values in spreadsheet cells */

RUN;
```

```
ODS tagsets.excelxp CLOSE ;
```

The above allows the PROC PRINT statements to create the Excel spreadsheet, "Enrollment and Claims Summary.xls". The ODS tagsets.excelxp FILE and the ODS tagsets.excelxp CLOSE statements are required before and after the PROC PRINT, respectively. The FILE statement tells SAS where to put the Excel spreadsheet and what to name it. The separate VAR statements allow the fill of the cells corresponding to the month-yr columns values to be printed using the format "color". The FORMAT statement ensures the actual values will not be passed to the spreadsheet. The result is just what I wanted, all without even opening Excel.

The following is an example of a similar resulting spreadsheet using different data from above.



CONCLUSION

The amount of data available to someone analyzing health insurance data can be overwhelming and frustrating, at times. Sometimes it seems as if the only way to convey the information you want is by endless rows of numbers. My hope was that this paper would provide a small respite from the depressing depths of numbers crunching and allow the reader to see that there is hope for displaying results without using "too many numbers" as a customer once disappointingly said to me.

ACKNOWLEDGEMENTS

CONTACT INFORMATION

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

Andrea Scott
CareFirst BlueCross BlueShield CT-07-11
1501 S. Clinton St.
Baltimore, MD 21224
Work Phone: 410-528-5003
Fax: 410-720-5313
Email: andrea.scott@carefirst.com