Medical versus Pharmacy Insurance: Which is More Cost-Effective for Providing the Prescription? Solving the Problem Via SAS® Enterprise Guide®

Amber Schmitz, Prime Therapeutics

ABSTRACT

Data-driven decisions that provide strategic solutions. These are buzzwords and phrases we have all heard before, but actually applying those words to deliver actionable data is less commonplace than it should be. This paper explores how to use both pharmacy and medical insurance claims data in order to assess drug utilization behavior across the medical and pharmacy insurance benefits. The final result provides actionable data to assess moving drug fills to the more cost-effective insurance benefit. This paper explores: 1) Utilizing macro variables for efficient data pulls, 2) Using Tasks to create data sets, merge tables, and calculate variables, 3) Demonstrate business intelligence via graph options, and 4) Exploiting built-in tools for template organization and version control.

INTRODUCTION

Many companies have transitioned to SAS Enterprise Guide as their preferred SAS interface. While SAS Enterprise Guide may seem like a simplified version of BASE SAS, it is a truly a unique tool that allows for organized and documented analysis. The most basic setup of SAS Enterprise Guide maintains the base functionality of SAS while also offering a broad suite of built-in functionality with drag and drop capabilities. This paper will assess the most cost-effective insurance benefit for drug fills while utilizing both program code & SAS Enterprise Guide’s built-in capabilities.

SAS Enterprise Guide has simplified the complexity of creating an organized analysis project by allowing the user to create multiple Process Flows within a SAS Enterprise Guide project. The incorporation of multiple Process Flows into the layout of the project allows analysts to build easily interpretable analysis projects. Display 1 below demonstrates a typical SAS Enterprise Guide Session layout. The Project Tree on the left-hand side of the graphic shows three Process Flows housed within the project: General Analysis, Reporting Output, Documentation/Version Control. The titles of each provide a large amount of information as to the contents of each and can be essential in building a master project template should the analysis need to be repeated by others.

TIP: Moving between Process Flows is incredibly simple. Switch between Process Flows via any of the following methods:

- Click on a Process Flow from the Project Tree on the Left
- Select a Process Flow from the Process Flow drop-down list
- Select another Process Flow from the general drop-down list

Display 1. Multiple Process Flows Example
TIP: Another organization tool that every SAS Enterprise Guide user needs to be aware of is Notes. Notes are useful when the need arises to call out specific analysis criteria within the project flow. For example, Display 1 shows a Note that directly specifies the analysis step where a filter is applied in the project along with noting exactly what the restriction is.

Notes can either be stand-alone notes or they can be linked to an analysis step. To link a note, use either of the following methods:

- Right-click on the note and select the ‘Link to’ option & appropriate item
- Hover on the edge of the note to get crosshairs. Click & drag the crosshairs to the appropriate item

CREATING PROCESS FLOW 1: GENERAL ANALYSIS

The General Analysis Process Flow will be used to house the basic analysis and data manipulation pieces of the project. For this analysis, the process flow will extract the applicable medical claims data, re-price the claims to the pharmacy benefit price, and then finally calculate the savings opportunity for the pharmacy benefit.

First, the medical claims data extract is completed in order to quantify the actual drug spend and volume. This is done via program code that utilizes macro variables. The top pane, or Process Flow pane, in Display 1 demonstrates the process flow used for the data extraction. The local MACRO VARIABLES_CLIENTS program has an INCLUDE statement that brings in the standard stored program files along with detailing the necessary macro variables that must be defined for the medical data extract. The med_datapull.sas program builds the MED_EXTRACT macro by incorporating the defined macro variables and looping through the applicable client libraries to complete the data extraction. Included in the MED_EXTRACT macro is a KEEP statement that contains a WHERE statement that applies the project specifications defined by the macro variables. These restrictions are all applied to the SET portion of the DATA step in order to process only the data that meets the report specifications. This is a great way to improve the efficiency of data pulls from the source data set!

TIP: Programs within a SAS Enterprise Guide project can either be saved only as a piece of the .egp project file (local) or they can be saved as a stand-alone .sas program file that can be referenced in multiple projects (stored). In order to save the program as a stand-alone program, select the ‘Save Program As’ option from the drop down menu and select the appropriate storage location and program name.

Display 2. Med_Extract Macro and Macro Variables

Once the data extract is complete, SAS Enterprise Guide’s built-in data set creation options will be used to re-price the medical claims and determine the pharmacy benefit savings opportunity. In order to assess the cost of the drug on the pharmacy benefit, the medical claim must be re-priced according to the pharmacy benefit price structure. This project takes medical claims paid at a contracted rate based on Average Sales Price (ASP) and re-prices them to an Average Wholesale Price (AWP) equivalent. This allows the medical benefit price (ASP) to be compared to the pharmacy benefit price (AWP).

To link the pharmacy benefit price to the medical claim, a reference table must be incorporated that crosswalks the medical procedure code to the associated pharmacy price. Pharmacy insurance pricing also incorporates a discount to the AWP price, so the client’s applicable discount rate is also maintained in the reference table. The Filter & Sort task is the most basic method of creating a new data set. This task is only available for use when there is no need to merge tables or create calculated variables. The pricing reference table will be filtered to reflect the appropriate pricing information for the selected therapeutic classes (Core) of drugs as demonstrated in Display 3 below.
Display 3. Filter Example

The pricing reference table is now ready to be joined to the medical claims data for the re-pricing exercise. The Query Builder option allows for more complex data sets to be built by 1) selecting the necessary variables, 2) joining data tables, 3) calculating variables. Query Builder also allows for complex joining of data sets. For this analysis, the medical claims data set (t1) will be inner joined to the reference price data set (t2) on procedure code and using a substring of the client name as shown in Display 4. In order to utilize the substring function (or any other function), the Join Properties menu must be used in conjunction with an established join.

Display 4. Complex Join Example

Once the data sets are joined, Query Builder is used to create calculated variables to determine the pharmacy benefit savings opportunity for each claim. For this analysis, calculated variables must be created to reflect the Best (Rx_Price_Best) and Worst (Rx_Price_Worst) pharmacy price available for the medical procedure code because medical procedure codes can encompass multiple National Drug Codes (NDCs) each with a client specific AWP discount rate. To create a calculated variable, select the blue calculator in the Query Builder box and specify the calculation. As Display 5 shows, the Best and Worst price variables are calculated by taking the number of drug services from the medical claim and multiplying by the AWP code price less the client's discount rate (repeated for
both the Best and the Worst applicable discount rate for the procedure code). The pharmacy benefit savings opportunity is then calculated as the difference of the actual drug spend on the claim versus the newly calculated pharmacy benefit price (repeated for both the Best and the Worst prices). This creates a savings estimate range based on any NDC that may be billable for the procedure code on the claim since medical claims data does not typically reflect the distinct NDC billed for reimbursement.

Display 5. Calculated Variables Example

TIP: The calculated variable can be referenced in other calculations or filters in the same Query Builder interface by specifying CALCULATED 'variable-name' in the referencing calculation or filter.

CREATING PROCESS FLOW 2: REPORTING OUTPUT

Once the general analysis flow is complete, the Reporting Output process flow can be built to meet business needs for report content. A good rule of thumb is to create a list report that demonstrates the numerical values along with visual representations of the data in order to appease all business users. The final report in this project is built using the Report functionality of SAS Enterprise Guide which allows the user to drag and drop each component of the report (listing reports, graphs, etc.) into a coordinated order prior to generating the final report.

For this analysis, it is best to start at a client level to demonstrate the overall savings opportunity range and then the report should drill-down into the data layer-by-layer to uncover each component of the story. The List Report Wizard can be used to construct the basic tabular output needed for the savings report. The List Report Wizard allows you to designate the data source, order the columns as needed, format output, and add in subtotals as appropriate.

Output 1. List Report Example
After the list report is generated, graphical output can be constructed to give a visual representation of the pharmacy savings opportunity according to each layer of the data. The drag and drop interface of SAS Enterprise Guide’s Graph suite allows analysts to quickly build various types of graphs that best demonstrate the stories told by the data. Every aspect of the output can be modified such as chart grouping, colors, axis options, and titles. Things to consider when building the final report with respect to this analysis: 1) Is there a therapeutic class that is not profitable on the pharmacy benefit?, 2) Is there a specific procedure code that has a much better price on the pharmacy benefit?, 3) Can the current business processes support the prospective volume?  While the listing output (Output 1) demonstrates an overall positive savings opportunity for the pharmacy benefit, there are components within the estimate where the medical benefit is the better choice.  This becomes apparent once each layer of data is dissected.

For this analysis, a Bar-Line Chart gives useful insight with respect to the most profitable therapeutic classes at each client. The bar is helpful to demonstrate the pharmacy benefit savings opportunity while the line builds in the prospective number of claims that may be transferred across insurance benefits. In Output 2 below, the Autoimmune therapeutic class has a positive savings opportunity of almost $6 million dollars on 4,000 claims while the IVIG & Other Serums therapeutic class has a negative savings opportunity of around $1 million on only 1,000 claims. This indicates that the Autoimmune therapeutic class offers a savings opportunity on the pharmacy benefit with respect to drug cost while IVIG & Other Serums does not unless the drug cost loss could be recouped through ancillary pharmacy benefit services (such as controlled patient management services offered through the pharmacy) which are not calculated in this analysis. Targeting the Autoimmune therapeutic class would offer financial savings, but it could also strain business processes due to the increased claims volume. Thus, the business operations aspect can also be analyzed from this prospective impact report.

Output 2.  Bar-Line Chart Example

Once the profitable therapeutic classes have been identified, a deeper dive into the actual drugs and associated savings that make up the therapeutic class is necessary. A bubble plot provides constructive insight as it is able to demonstrate the number of services available to transfer to the pharmacy benefit (most comparable to quantity in the pharmacy world) along with the associated pharmacy benefit savings at the procedure code level. Output 3 demonstrates that there is a negative savings opportunity for procedure code J0129 with a high amount of services while J3357 offers a large amount of savings with a small number of services.
Output 3. Bubble Plot Example

The final report provides guidance on the strategy that the business should take with respect to each client. In this example, the overall pharmacy benefit savings opportunity is positive for both clients (see Output 1), but to maximize the pharmacy savings opportunity, therapeutic classes should be targeted for movement from the medical to the pharmacy benefit. The amount of claims and respective volume of drug are also quantified in the report output to allow the business to estimate its capabilities for handling this prospective business. The operational volumes are especially useful if the pharmacy benefit provider also has an in-house pharmacy that would be responsible for distributing a majority of the transferred business’s drug volume.

CREATING PROCESS FLOW 3: DOCUMENTATION/VERSION CONTROL

If this project is to be maintained as a master template for future analyses, the most crucial component of the project will be the documentation of usage and version change control information. SAS Enterprise Guide allows for this documentation to be easily implemented. Display 6 demonstrates the Documentation/Version Control Process Flow that utilizes Notes to incorporate pertinent information. Documentation of changes to the template is maintained in a consistent manner in the Version Documentation note. The release notes are structured in a way that many programmers already incorporate into program code, but now the change documentation applies to the entire project template. This is especially useful when you have multiple users accessing a master template for report generation.
TIP: Two other beneficial pieces of information that can be maintained in the Documentation/Version Control process flow are Instructions for Use and Interpreting Results. Each of these topics can be contained within its own Note in the Process Flow. By incorporating both of these items, the need to maintain this information in another document is removed along with the information always being readily accessible whenever an analyst is working in the project.

CONCLUSION

SAS Enterprise Guide is used widely throughout the business world. Becoming more familiar with built-in tasks and organizational tools can greatly benefit many reporting analysts. This paper has explored creating a SAS Enterprise Guide project that determines the most cost-effective insurance benefit for drug fills. The assessment utilizes both program code and SAS Enterprise Guide’s built-in capabilities to create a master project that allows for consistent and timely reporting in order to deliver actionable data that drives business strategy.

ACKNOWLEDGMENTS

I would like to thank my family, coworkers, and boss for challenging me to share my knowledge.

I would also like to thank Scott Leslie for his thoughtful review & comments.

RECOMMENDED READING

- The Little SAS® Book
- The Little SAS® Book for Enterprise Guide®

CONTACT INFORMATION

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

Name: Amber Schmitz  
Enterprise: Prime Therapeutics  
Address: 5600 West 84th Street  
City, State ZIP: Bloomington, MN 55437  
E-mail: ASchmitz@PrimeTherapeutics.com or Akroeger@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.