Implementing Excel Pivot Functionality in SAS® Software

Shane TRAHAN, Mai NGUYEN, Inga ALLRED, Nick KINSEY
RTI International
Research Triangle Park, NC 27709

ABSTRACT

The enormity of data used and collected in all levels of research is overwhelming; to many data analysts this deluge poses not only opportunities but can be a significant hindrance to figuring out "what does my data tell me?" Tools abound but many analysts just need something to get them started and many turn to one simple yet effective tool, Microsoft Excel's PivotTable® utility. The Excel's PivotTable is a versatile function allowing users to view data in a variety of different ways. Large datasets can be easily manipulated by filtering, transforming and aggregating information providing valuable insights including difficult to detect trend identification. Our paper will provide codes and illustrate a method of using this data mining technique and firmly places this unique and simple-to- understand tool into the hands of SAS® developers. We will step users through the use of PROC SQL and PROC TRANSPOSE procedures to create a robust pivot table utility easily applied to a variety of SAS® based applications. Our goal is to give users tools that not only can be used across many types of data but also help identify important information to begin analysis.

Keywords: pivot table, transpose, SQL, SAS®.

INTRODUCTION

Managing large data sets can be a challenge and SAS® offers many robust analytical tools. However many analysts prefer to see data in simple terms and in a format familiar to others outside of the field of analytics. The knowledge worker of today has grown accustomed to the power and ease of use of software packages such as Microsoft Excel which offers many mid-level analytical tools. With relatively few clicks of a pointing device, Excel users can whip up data summaries with relative ease and results can be copied and pasted into Microsoft Word documents and email messages. One of the most powerful data models available to analyze patterns of data is the Pivot Table. The Pivot Table is a data summarization tool or technique available in spreadsheet programs and in business intelligence software. This paper will introduce how to build PivotTable summaries using PROC SQL and PROC TRANSPOSE using SAS® software.

SAS®, Statistical Analysis System, is an integrated system of software products widely used for statistical analysis, report writing, data integration and data warehousing among many other features and functionality. Its rich and powerful libraries of built-in procedures allow the users to prepare and analyze large dataset quickly and with ease. In this paper, two SAS® procedures will be used for the creation of a pivot table. PROC SQL is used for data summarization and aggregation while PROC TRANSPOSE is used for converting data from rows to columns or vice versa. Both of these procedures are feature rich and provide effective analytical capabilities on their own, however when used together they provide analysts valuable insights into the discovery of information and data patterns.

Introductory descriptions of PROC SQL and PROC TRANSPOSE procedures are provided in the next two sections to help those unfamiliar with them. For the complete reference of these two procedures, consult SAS® online documentation at http://support.sas.com/documentation/92/index.html.

PROC SQL

The PROC SQL procedure is SAS® software's way of implementing Structured Query Language (SQL). The procedure is equipped with a variety of data manipulation tools. PROC SQL can generate reports, summary statistics, retrieve data from tables or views, combine data from tables or views, create tables, views, and indexes, update data values, and retrieve or update data from database management system (DBMS) tables. A PROC SQL table is the same as a SAS® dataset; rows in the table are equivalent to SAS dataset observations and columns are equivalent to SAS dataset variables. To create a PROC SQL data view, the following code can be used.

```
PROC SQL;
   SELECT Category
   FROM SampleData;
RUN;
```

This view will return the variable Category for all observations in the SampleData table. To create a data table, only one extra line of code is needed.

```
PROC SQL;
   CREATE TABLE CategorySales as
   SELECT Category
   FROM SampleData
RUN;
```

This code snippet will create a table called CategorySales containing the variable Category for all observations in the SampleData table. A WHERE condition may also be used to subset the incoming data to only those rows that meet specified criteria.

There are several other useful features that can be used with PROC SQL. UPDATE is used for updating existing rows in a table or view. INSERT is used for inserting new rows and DELETE is used for deleting rows from a table or view. ALTER TABLE can be used to add, modify, and even delete columns within existing tables. These are just a few statements that are available for use within the PROC SQL procedure. The most useful PROC SQL data mining statement is the universal SELECT statement. This statement can be used to perform data aggregation by groups and together with aggregate functions such as SUM can help to provide valuable insight into data. Some of these features are shown in the code snippet below.

```
PROC SQL;
   CREATE TABLE CategorySales AS
   SELECT Category, Product, Sum(Sales) as Sales
   FROM SampleData
   GROUP BY Category, Product
   ORDER BY Category, Sales Desc;
RUN;

PROC SQL;
   ALTER TABLE CategorySales
   ADD SalesPerson Character(25)
                   label = “Sales Person”;

   UPDATE CategorySales SET
        SalesPerson = “JOHN DOE”;

   TITLE “Adding SalesPerson Column”;

   SELECT Category, Product, Sales, SalesPerson
   FROM CategorySales;
RUN;
```

PROC TRANSPOSE

The PROC TRANSPOSE procedure reads in a dataset and rotates/pivots the specified rows and columns to form a new named dataset. What were once rows or observations in the input dataset are now columns and therefore become variables and what were once columns become the rows or observations of the new transposed SAS® dataset. This is illustrated in the sample code below where the input dataset ProductSales is transposed to produce the output ProductSalesByQuarters dataset.

```
* Pivot dataset by Quarter variable;
PROC TRANSPOSE DATA=ProductSales
   OUT=ProductSalesByQuarters NAME=Sales;
   BY Category Product;

   VAR Sales;
   ID Quarter;
RUN;
```

Category	Product	Sales	Quarter
Beverages	Chai	$705.60	Qtr1
Beverages	Chai	$878.40	Qtr2
Beverages	Chai	$1,174.50	Qtr3
Beverages	Chai	$2,128.50	Qtr4
Beverages	Chang	$2,720.80	Qtr1
Beverages	Chang	$228.00	Qtr2
Beverages	Chang	$2,061.50	Qtr3
Beverages	Chang	$2,028.25	Qtr4
Beverages	Chartreuse vert	$590.40	Qtr1
Beverages	Chartreuse vert	$360.00	Qtr2
Beverages	Chartreuse vert	$1,100.70	Qtr3
Beverages	Chartreuse vert	$2,424.60	Qtr4

Figure 1 - Input Dataset before PROC TRANSPOSE

The procedure knows the number of columns needed based on number of observations in the data, doing this with data steps would be highly inefficient. The BY statement in the code will create an observation for each variable value. The VAR statement tells SAS which variable to transpose and it should be noted that without the VAR statement, numeric variables not specified in the ID or BY statements will also be transposed.

Category	Product	Sales	Qtr1	Qtr2	Qtr3	Qtr4
Bevera…	Chai	Sales	$706	$878	$1,175	$2,129
Bevera…	Chang	Sales	$2,721	$228	$2,062	$2,028
Bevera…	Chartr…	Sales	$590	$360	$1,101	$2,425

Figure 2 - Output Dataset after PROC TRANSPOSE

IMPLEMENTING PIVOT FUNCTIONS IN SAS®

Our sample codes start with an example using simple PROC SQL statement and will wrap up with a more complex one using both PROC SQL and PROC TRANSPOSE together to summarize data by particular variables. Due to the page limit, only partial sample codes are presented in this paper. Likewise, sample data [1] and output data are incomplete and shown for demonstration purpose only. Complete sample codes, sample data and output data are available upon request.

A partial listing of the sample data is shown in Figure 3 below.

Category	Product	Sales	Quarter
Beverages	Chai	$705.60	Qtr 1
Beverages	Chai	$878.40	Qtr 2
Beverages	Chang	$2,061.50	Qtr 3
Beverages	Chang	$2,028.25	Qtr 4
Beverages	Chartreuse verte	$590.40	Qtr 1
Beverages	Chartreuse verte	$360.00	Qtr 2
Condiments	Aniseed Syrup	$544.00	Qtr 1
Condiments	Aniseed Syrup	$600.00	Qtr 2
Condiments	Chef Anton's Cajun Seasoning	$1,337.60	Qtr 3
Condiments	Chef Anton's Cajun Seasoning	$682.00	Qtr 4
Condiments	Chef Anton's Gumbo Mix	$288.22	Qtr 3
Condiments	Chef Anton's Gumbo Mix	$85.40	Qtr 4
Confections	Chocolade	$162.56	Qtr 2
Confections	Chocolade	$68.85	Qtr 3
Confections	Maxilaku	$1,605.60	Qtr 1
Confections	Maxilaku	$620.00	Qtr 2
Dairy Products	Camembert Pierrot	$3,329.28	Qtr 1
Dairy Products	Camembert Pierrot	$3,060.00	Qtr 4
Dairy Products	Fløtemysost	$4,454.80	Qtr 1
Dairy Products	Fløtemysost	$174.15	Qtr 2
Dairy Products	Fløtemysost	$2,541.29	Qtr 3
Grains/Cereals	Filo Mix	$226.80	Qtr 3
Grains/Cereals	Filo Mix	$911.75	Qtr 4
Grains/Cereals	Gnocchi di nonna Alice	$6,931.20	Qtr 1
Grains/Cereals	Gnocchi di nonna Alice	$9,868.60	Qtr 2
Grains/Cereals	Gnocchi di nonna Alice	$6,771.60	Qtr 3
Grains/Cereals	Gnocchi di nonna Alice	$9,032.60	Qtr 4
Grains/Cereals	Gustaf's Knäckebröd	$201.60	Qtr 1
Grains/Cereals	Gustaf's Knäckebröd	$504.00	Qtr 2
Meat/Poultry	Alice Mutton	$2,667.60	Qtr 1
Meat/Poultry	Alice Mutton	$4,013.10	Qtr 2
Meat/Poultry	Alice Mutton	$3,900.00	Qtr 3
Meat/Poultry	Alice Mutton	$6,000.15	Qtr 4
Meat/Poultry	Mishi Kobe Niku	$1,396.80	Qtr 1
Meat/Poultry	Mishi Kobe Niku	$1,319.20	Qtr 2
Meat/Poultry	Mishi Kobe	$4,219.50	Qtr 4

	Niku		
Produce	Longlife Tofu	$360.00	Qtr 1
Produce	Longlife Tofu	$128.00	Qtr 2
Produce	Longlife Tofu	$400.00	Qtr 4
Produce	Manjimup Dried Apples	$1,411.92	Qtr 1
Produce	Manjimup Dried Apples	$8,384.60	Qtr 2
Seafood	Boston Crab Meat	$1,474.41	Qtr 1
Seafood	Boston Crab Meat	$2,272.00	Qtr 2
Seafood	Boston Crab Meat	$3,887.92	Qtr 3
Seafood	Boston Crab Meat	$2,162.00	Qtr 4
Seafood	Carnarvon Tigers	$1,500.00	Qtr 1
Seafood	Carnarvon Tigers	$2,362.50	Qtr 2
Seafood	Carnarvon Tigers	$7,100.00	Qtr 3

Figure 3 – Partial Listing of Test Data

EXAMPLE 1 - SALES BY CATEGORIES

The sample code for the first example shown below emphasize the usage of the PROC SQL procedure to calculate the sum of sales grouped by category. It includes three distinct steps described below:

1. Read input data into a working SAS® dataset.
2. Using PROC SQL with the GROUP BY clause to generate Sales by Categories data. SQL allows grouping by any number of variables, however variables in the GROUP BY clause must be contained in the SELECT list or an error will occur.
3. Output the results to Excel. Once in Excel, the values can be formatted or summarized as desired.

```
/**
 ** Excel Pivot Example 1 -- Category Sales
 ** Total Sales by Categories
 **/

* (1) Read sample data into the working dataset
      SampleData;
%include 'C:\WMSCI_2012\ReadSampleData.SAS';

* (2) Generate Total Sales by Categories data
*     using PROC SQL;
PROC SQL;
   CREATE TABLE CategorySales AS
   SELECT
       Category,
       SUM(Sales) as Sales format=DOLLAR20.0
    FROM SampleData
    GROUP BY Category;
QUIT;

* (3) Output result to Excel;
PROC EXPORT DATA=CategorySales
   File='C:\WMSCI_2012\PivotOutputBySAS.xls'
   dbms=excel
   replace;
RUN;
```

The Excel output is shown in Figure 4.

Category	**Sales**
Beverages	102,074.29
Condiments	55,277.56
Confections	80,894.11
Dairy Products	114,749.75
Grains/Cereals	55,948.82
Meat/Poultry	81,338.06
Produce	53,019.98
Seafood	65,544.19

Figure 4 – Total Sales by Categories

EXAMPLE 2 - SALES BY PRODUCTS

Taking the first example a step further, we add an additional GROUP BY variable to our SELECT statement. With that, we can break out categories by products as shown in Figure 5. Now we know sales by category and by product. The sample code for this example is shown below

```
/**
 ** Excel Pivot Example 2 -- Product Sales
 ** Total Sales by Products
 **/

* Read sample data into the working dataset
  SampleData;
%include 'C: WMSCI_2012\ReadSampleData.SAS';

* Generate Total Sales by Categories and
  Products using PROC SQL;
PROC SQL;
   CREATE TABLE ProductSales AS
   SELECT
        Category,
        Product,
        SUM(Sales) as Sales format=DOLLAR20.0
   FROM SampleData
   GROUP BY Category, Product;
QUIT;

* Output result to Excel;
PROC EXPORT DATA=ProductSales
   File='C:\WMSCI_2012\PivotOutputBySAS.xls'
   dbms=excel
   replacel;
RUN;
```

The Excel output of this program is shown below in Figure 5.

Category	**Product**	**Sales**
Beverages	Chai	4,887.00
Beverages	Chang	7,038.55
Beverages	Chartreuse verte	4,475.70
Condiments	Aniseed Syrup	1,724.00
Condiments	Chef Anton's Cajun Seasoning	5,214.88
Condiments	Chef Anton's Gumbo Mix	373.62
Confections	Chocolade	1,282.01
Confections	Maxilaku	3,060.60
Dairy Products	Camembert Pierrot	20,652.28
Dairy Products	Fløtemysost	9,642.74
Dairy Products	Geitost	786.00
Grains/Cereals	Filo Mix	2,068.15
Grains/Cereals	Gnocchi di nonna Alice	32,604.00
Grains/Cereals	Gustaf's Knäckebröd	4,233.60
Grains/Cereals	Ravioli Angelo	2,156.70
Meat/Poultry	Alice Mutton	16,580.85
Meat/Poultry	Mishi Kobe Niku	6,935.50
Meat/Poultry	Perth Pasties	12,784.13
Meat/Poultry	Pâté chinois	8,826.48

Meat/Poultry	Thüringer Rostbratwurst	33,109.51
Meat/Poultry	Tourtière	3,101.59
Produce	Longlife Tofu	888.00
Produce	Manjimup Dried Apples	23,550.02
Produce	Rössle Sauerkraut	12,854.28
Produce	Tofu	6,541.38
Produce	Uncle Bob's Organic Dried Pears	9,186.30
Seafood	Boston Crab Meat	9,796.33
Seafood	Carnarvon Tigers	15,950.00
Seafood	Escargots de Bourgogne	2,076.28

Figure 5 – Total Sales by Products and Categories

EXAMPLE 3 - TOP THREE PRODUCTS

With very large datasets, we can still be overwhelmed even when using the summarization toolsets that have been previously discussed. In this case, we can add additional criteria to return less data and allow a more focused analysis.

This can be done by simply adding a WHERE clause to our SQL statement but what if we want to look at the top three products by sales in each category? Using a SAS® macro as shown in this example we can evaluate each category and with the OUTOBS option we can pull out the top three products by sales in each category. Figure 6 below shows the Excel output produced by the sample code.

Category	**Product**	**Sales**
Beverages	Côte de Blaye	46,563.08
Beverages	Ipoh Coffee	11,069.90
Beverages	Lakkalikööri	7,883.10
Condiments	Louisiana Fiery Hot Pepper Sauce	9,331.08
Condiments	Sirop d'érable	9,091.50
Condiments	Vegie-spread	6,899.25
Confections	Tarte au sucre	20,762.82
Confections	Gumbär Gummibärchen	11,225.65
Confections	Schoggi Schokolade	10,974.00
Dairy Products	Raclette Courdavault	33,616.55
Dairy Products	Camembert Pierrot	20,652.28
Dairy Products	Gudbrandsdalsost	14,041.80
Grains/Cereals	Gnocchi di nonna Alice	32,604.00
Grains/Cereals	Wimmers gute Semmelknödel	7,125.47
Grains/Cereals	Singaporean Hokkien Fried Mee	5,184.20
Meat/Poultry	Thüringer Rostbratwurst	33,109.51
Meat/Poultry	Alice Mutton	16,580.85
Meat/Poultry	Perth Pasties	12,784.13
Produce	Manjimup Dried Apples	23,550.02
Produce	Rössle Sauerkraut	12,854.28
Produce	Uncle Bob's Organic Dried Pears	9,186.30
Seafood	Carnarvon Tigers	15,950.00
Seafood	Boston Crab Meat	9,796.33
Seafood	Ikura	8,819.50

Figure 6 - Top Three Products in Each Category by Sales

The sample code below steps us through a series of PROC SQL statements and the creation of a SAS® macro that will give our desired criteria of top three products by sales for each category. There are four distinct steps as described below:

1. Obtain total sales by products in each category using multiple variables in the GROUP BY statement.
2. Select number of categories and the category names into macro variables to be used later in the Top_N_By_Group macro created in step 3.
3. The Top_N_By_Group macro loops through the dataset and select top three products by total sales for each category using the OUTOBS option.
4. Output the results to Excel. Once in Excel, the values can be formatted or summarized.

```
/**
 ** Excel Pivot Example 3 -- Top 3 Product Sales
 ** by Categories
 ** Top 3 Product Sales by Categories
 **/

%let TopN = 3;

* Read sample data into the working dataset
  SampleData;
%include 'C:\WMSCI_2012\ReadSampleData.SAS';

* (1) Sort Total Sales by Products data;
PROC SQL;
   CREATE TABLE ProductSales AS
   SELECT
        Category,
        Product,
        SUM(Sales) as Sales format=DOLLAR20.0
   FROM SampleData
   GROUP BY Category, Product
   ORDER BY Category, Sales Desc;
QUIT;

* (2) Select year into macro variables for
      processing;
PROC SQL NOPRINT;
   SELECT COUNT(DISTINCT Category) INTO
      :n_Categories
   FROM ProductSales;

   SELECT DISTINCT "'" || Category || "'" INTO
      :Category1 - Category%left(&n_Categories)
   FROM ProductSales;
QUIT;

* (3) select TopN rows into the result dataset;
%macro Top_N_By_Group;
   PROC SQL;
      CREATE TABLE Top3Products LIKE
         ProductSales;
   QUIT;

   %do i=1 %to &n_Categories;
      PROC SQL OUTOBS = &TopN;
        INSERT INTO Top3Products
        SELECT * FROM ProductSales
        WHERE Category = &&Category&i;
      QUIT;
   %end;
%mend Top_N_By_Group;

%Top_N_By_Group;

* (4) Output result to Excel;
PROC EXPORT DATA=Top3Products
   File='C:\WMSCI_2012\PivotOutputBySAS.xls'
   dbms=excel
   replace;
RUN;
```

EXAMPLE 4 – PRODUCT SALES BY QUARTERS

Using the same dataset we can break the sale data out by Quarter. Listed below are the sales figures by product and category organized by quarter. Notice that the total sales data has been “morphed” from a single column into four columns, one for each quarter.

Category	Product	Qtr1	Qtr2	Qtr3	Qtr4
Beverages	Chai	705	878	1,174	2,128
Beverages	Chang	2,720	228	2,061	2,028
Beverages	Chartreuse verte	590	360	1,100	2,424
Condiments	Aniseed Syrup	544	600	140	440
Condiments	Chef Anton's Cajun Seasoning	225	2,970	1,337	682
Condiments	Chef Anton's Gumbo Mix			288	85
Condiments	Genen Shouyu		176	1,298	
Condiments	Grandma's Boysenberry Spread			1,750	750
Condiments	Gula Malacca	1,994	1,753	1,093	1,701
Condiments	Louisiana Fiery Hot Pepper Sauce	1,347	2,150	1,975	3,857
Confections	Gumbär	5,079	1,249	2,061	2,835
Confections	Maxilaku	1,605	620	835	
Confections	NuNuCa Nougat	193	865		493
Dairy Products	Camembert Pierrot	3,329	3,989	10,273	3,060
Dairy Products	Fløtemysost	4,454	174	2,541	2,472
Dairy Products	Geitost	294	242	99	150
Meat/ Poultry	Alice Mutton	2,667	4,013	3,900	6,000
Meat/ Poultry	Mishi Kobe Niku	1,396	1,319		4,219
Meat/ Poultry	Perth Pasties	5,154	2,099	1,500	4,029
Produce	Longlife Tofu	360	128		400
Produce	Manjimup Dried Apples	1,411	8,384	1,855	11,898
Produce	Rössle Sauerkraut	4,105	3,310	1,881	3,556
Seafood	Boston Crab Meat	1,474	2,272	3,887	2,162
Seafood	Carnarvon Tigers	1,500	2,362	7,100	4,987
Seafood	Gravad lax	208	421		

Figure 7 - Quarterly Sales by Products and Categories

We produced the output result in Figure 7 with the SAS® program below, using a combination of PROC SQL and PROC TRANSPOSE.

```
/**
 ** Excel Pivot Example 4 -- Product Sales by
 ** Quarters
 ** Product Sales by Quarters
 **/

* Read sample data into the working dataset
  SampleData;
%include 'C:\WMSCI_2012\ReadSampleData.SAS';

* (1) Generate Total Sales by Products data;
PROC SQL;
   CREATE TABLE ProductSales AS
   SELECT
        Category,
        Product,
        Quarter,
        SUM(Sales) as Sales format=DOLLAR20.0
   FROM SampleData
   GROUP BY Category, Product, Quarter;
QUIT;

* (2) Pivot dataset by Quarter variable;
PROC TRANSPOSE DATA=ProductSales
   OUT=ProductSalesByQuarters NAME=Sales;

   BY Category Product;
   VAR Sales;
   ID Quarter;
RUN;

* (3) Output result to Excel;
PROC SQL;
   ALTER TABLE ProductSalesByQuarters
            DROP Sales;
QUIT;

PROC EXPORT DATA=ProductSalesByQuarters
   File='C:\WMSCI_2012\PivotOutputBySAS.xls'
   dbms=excel
   replace;
RUN;
```

The three main steps in this program are:

1. First aggregate total sales by product by quarter.
2. Use PROC TRANSPOSE to pivot data with the Quarter variable.
3. Drop the Sales field from the ProductSalesByQuarters table for a cleaner summary data.

CONCLUSION

Excel’s PivotTable summary functions are useful for finding pattern matches and assist data analysts when mining data for information. We have shown that the same pivot table functionality commonly utilized by many Excel users can also be done in SAS® with very common SAS® procedures. The four examples discussed here have shown a very simple pivot table to a more robust and complex summary. Each one of these examples can be easily expanded on and enhanced to use more advanced SQL, potentially reaching across multiple data sources including Oracle databases, Microsoft Excel files or even comma delimited text files. Using PROC TRANSPOSE and PROC SQL together will allow users to view data in many different ways and the code presented here demonstrates how easy it is to customize output of data analysis summaries.

AUTHOR CONTACT

Shane Trahan
RTI International
3040 Cornwallis Rd
Durham NC 27709
srt@rti.org

Mai Nguyen
RTI International
3040 Cornwallis Rd
Durham NC 27709
mnguyen@rti.org

Inga Allred
RTI International
3040 Cornwallis Rd
Durham NC 27709
irb@rti.org

Nick Kinsey
RTI International
3040 Cornwallis Rd
Durham NC 27709
nlk@rti.org

REFERENCES

[1]. The sample data titled “Excel 2002 Sample: PivotTable Reports” was used in this paper and can be downloaded directly from Microsoft website at the following address: http://www.microsoft.com/download/en/confirmation.aspx ?id=14738.