The Armchair Quarterback:

Writing SAS® Code for the Perfect Pivot (Table, That Is)

Peter Eberhardt, Fernwood Consulting Group Inc., Toronto, ON, Canada

ABSTRACT

"Can I have that in Excel?" This is a request that makes many of us shudder. Now your boss has discovered Excel pivot tables. Unfortunately, he has not discovered how to make them. So you get to extract the data, massage the data, put the data into Excel, and then spend hours rebuilding pivot tables every time the corporate data are refreshed. In this workshop, you learn to be the armchair quarterback and build pivot tables without leaving the comfort of your SAS® environment. In this workshop, you learn the basics of Excel pivot tables and, through a series of exercises, you learn how to augment basic pivot tables first in Excel, and then using SAS. No prior knowledge of Excel pivot tables is required.

INTRODUCTION

In today's world the static report, although still required, is no longer enough; analysts and managers at all levels are looking deeper into interactions in the data. To help look at these interactions most people turn to Online Analytical Processing, commonly called OLAP cubes. For the more technical analysts, SAS provides exceptional OLAP capabilities using the OLAP server and SAS Enterprise Guide (EG). However, many business analysts, and business managers and executives prefer to work within the tool with which they are familiar – Microsoft Excel. Since Excel pivot tables offer much of the functionality of OLAP cubes, and managers and executives are demanding OLAP type 'reports' it is not surprising that more and more we are asked to provide data in the form of pivot tables.

In this paper we will look at the anatomy of an Excel Pivot table – first its components and then how a pivot table is created and accessed. This will allow us to see the terminology used so we can relate this back to our SAS emvironment. From there we will look at the SAS technology required to create pivot tables – and ODS markup destination called ***tableEditor***. The paper will then step through a series of examples first showing how to perfom a function in Excel (for example, change the default statistic from ***sum*** to ***average***) followed by the SAS code needed to create a pivot table the same way. There are a great number of ways a pivot table can be created and formatted; this paper will only touch on a few. After following through these examples you will be able to create your own pivot tables in a format that is consistent with your business requirements.

You do not need any prior experience with Excel pivot tables to follow the examples in the papers. In addition, anyone with an introductory level of SAS will be able to do the examples. The purpose of the paper is to demonstrate the use of the tableEditor tagset, so all the SAS examples are extremely simple. It practice, much more data manipulation would be required.

The examples in this paper closely follow Parker 2010; for more explanation on the how the tagset works refer to that paper.

DATA

The examples in the paper use data from the SAS supplied data set ***sashelp.shoes***; this will allow anyone with access to SAS the ability to follow these examples. Although this is a small dataset, it has enough variety to demonstrate.

TERMINOLOGY

In this paper there are a number of terms that mean the same thing and will often be used interchangeably. Some of the terms are used in SAS while others are used in Excel. In general we will try to use the term appropriate to the environment to which it applies; that is, in Excel we will use the Excel term and in SAS we will use the SAS term. The one term used most will be ***field*** (an Excel term), ***variable*** (the SAS DATA step term), and ***column*** (the SQL term).

EXCEL PIVOT TABLES – IN EXCEL

What is a Pivot Table

Pivot table is a generic term to describe data summarization in spreadsheets. In simple terms creating a pivot table is similar to creating a cross-tab report in SAS using PROC Tabulate or PROC Report – tabular data are summarized with one (or more) variable in the table becoming rows in the report, and one (or more) variables becoming columns in the report. For example, if our data had five variables – three classification variables Quarter, Product_Group and, Product_Category, and two analysis variables Sales and Quantity, we could create a cross-tab with values of Product_group on the rows, values of Quarter as the columns, and total Sales as the cell value. This type of cross-tab report is easily generated in PROC Tabulate. Like a PROC Tabulate report, a pivot table is built from a data table (in this case a spreadsheet table). Unlike a SAS PROC Tabulate report, a pivot table can dynamically change the variables used for rows, columns and cell values. To change the values, rows, or columns in the pivot table you drag and drop fields through the pivot table interface. To change the values, rows, or columns in the PROC Tabulate report you must change your code and re-run the PROC. Needless to say, when there are numerous classification variables, the process of changing the table is simpler in the spreadsheet. Pivot tables can be seen as a simplification of the more complete and complex OLAP concepts.

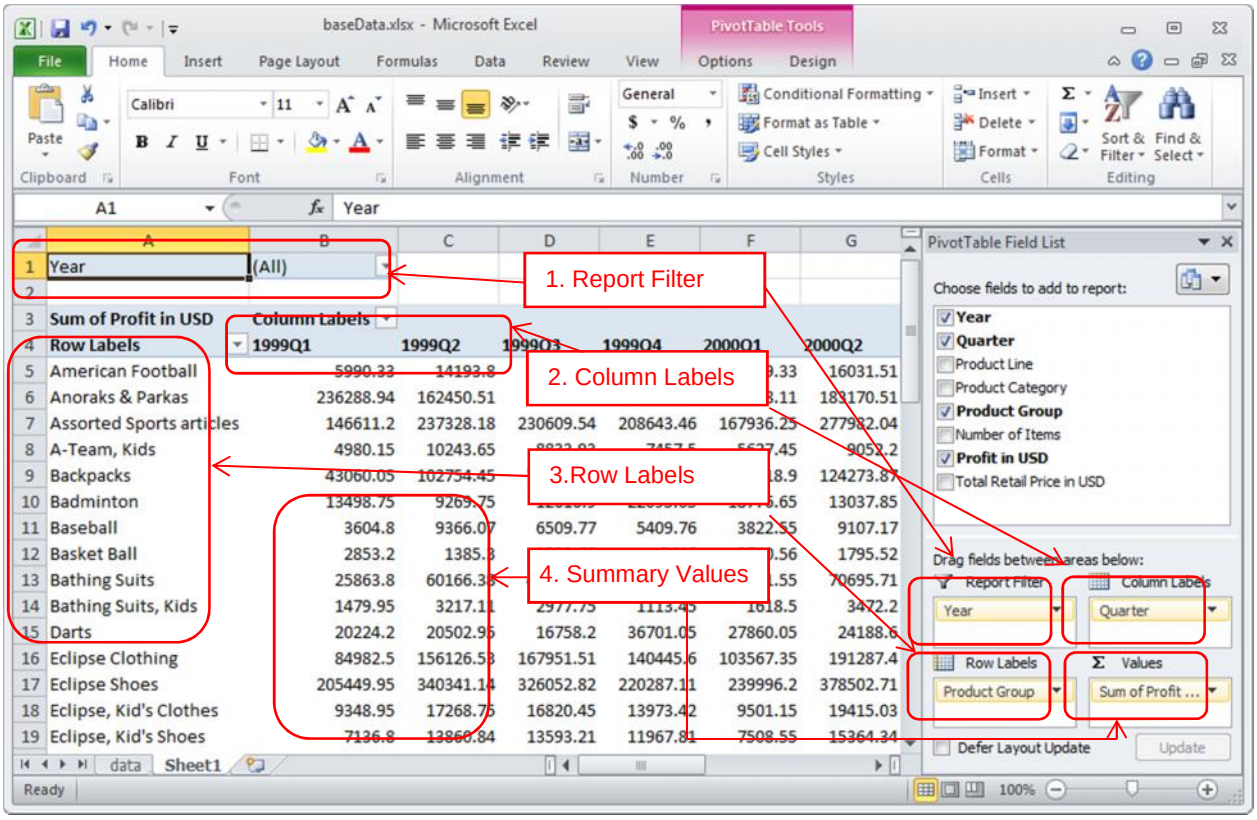
Whereas the term pivot table is generic, the term PivotTable is trademarked and specific to Microsoft Excel. In this paper all examples were shown using Excel 2010. Earlier versions of Excel (2003, 2007) have the similar functionality however some of the interface components look different.

What are the Components of a PivotTable

The four components of a PivotTable are:

1. Report Filter – to apply a filter to the entire PivotTable. In Figure 1, ***Year*** is the report filter. This is not required.
2. Column Labels – the variables (fields) that will be in the columns. In Display 1, ***Quarter*** is the column label.
3. Row Labels – the variables (fields) that will be in the row. In Display 1, ***Product_Group*** is the column label.
4. Summary values – the variables (fields) and summary statistic to be displayed in the cells. In Display 1, ***Profit*** is the summary field and ***sum*** is the statistic.

In Display 1, there is only one field in each component; it is possible to have multiple fields in each component.



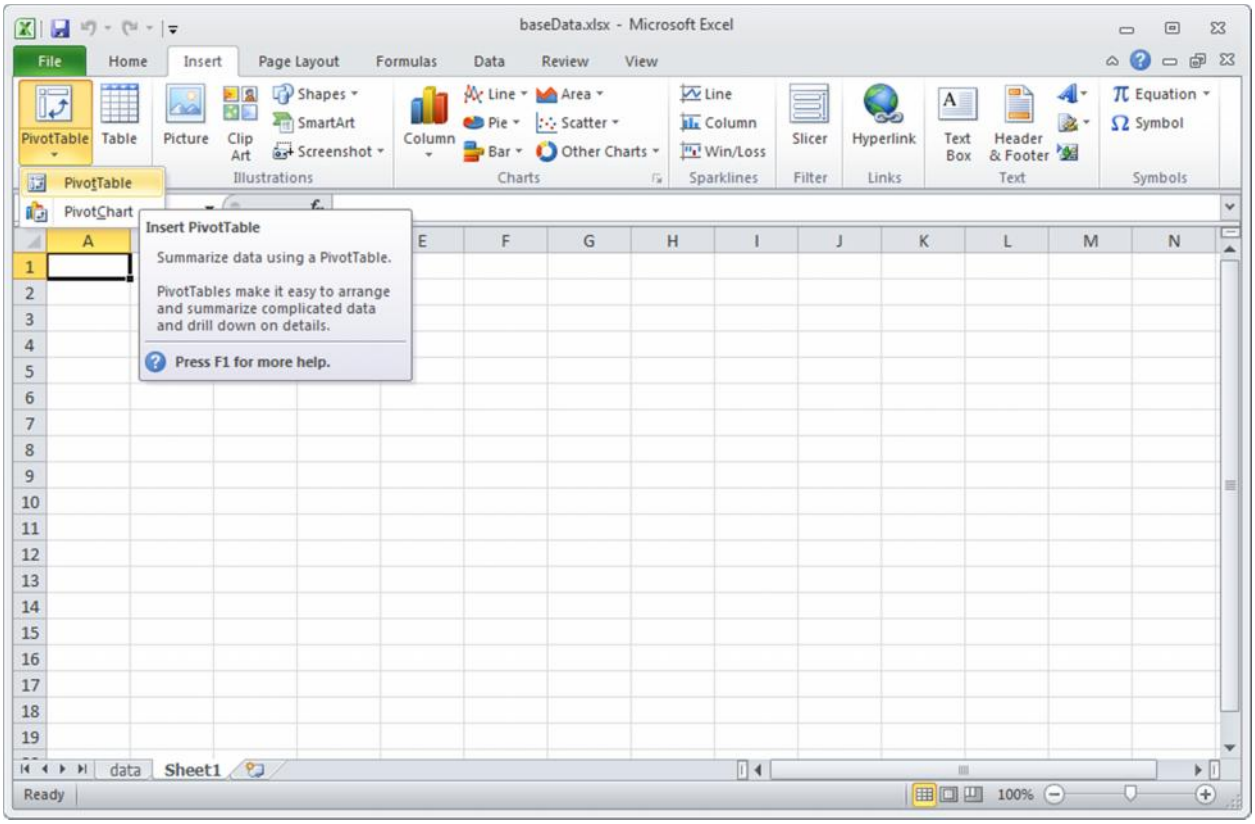
Display 1. A PivotTable with the four components highlighted

How is a Pivot Table Created

To create a PivotTable you first need to have a suitable data table in the spreadsheet. By suitable we mean it must have regular rows and columns in manner of a SAS data table. Normally at least one of the columns (variables) will be a numeric and several of the columns will be classification variables (commonly character). The classification variables are used in the Report Filter, Column Labels, and Row Labels components; the numeric variables are used in the Summary Values component. This is the same type of data that would be used in a PROC Tabulate report.

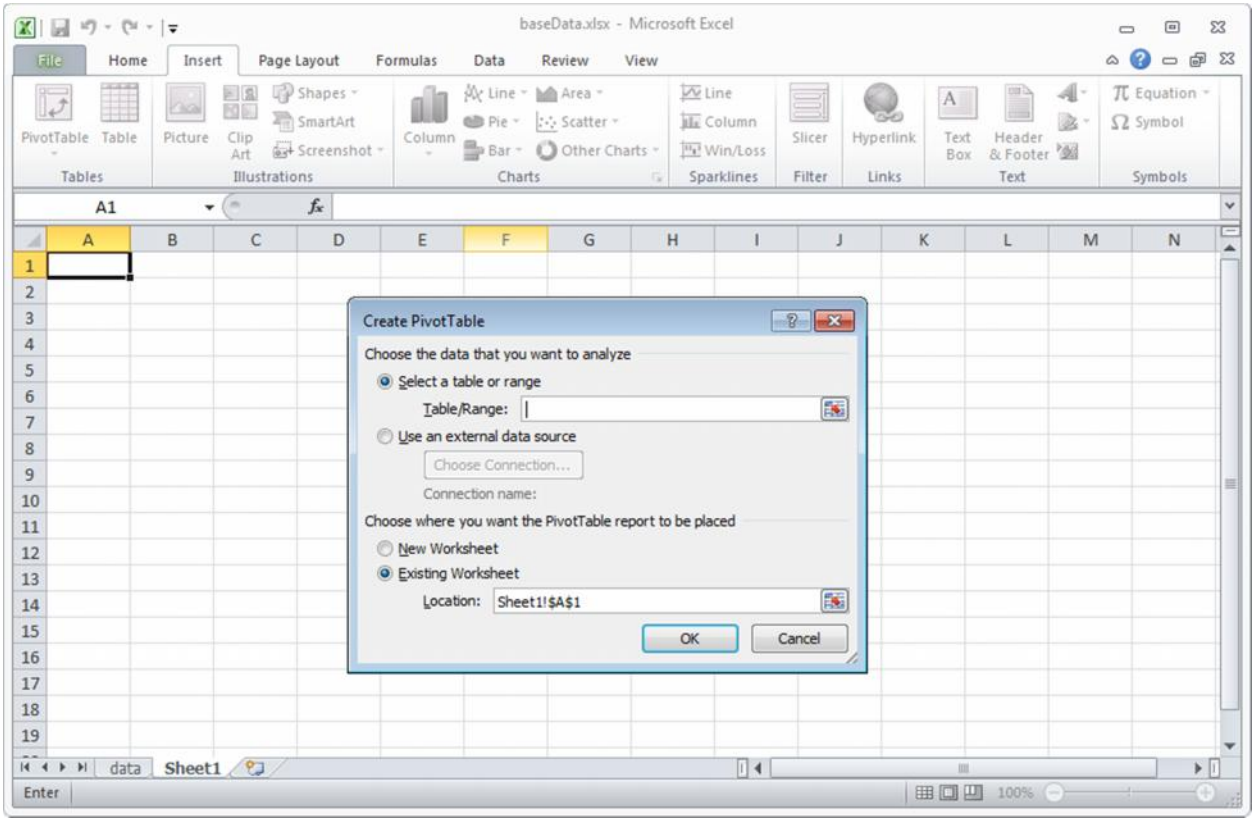
Once a suitable data table is loaded into Excel, the steps create a PivotTable are:

1. From the Excel menu select ***Insert*** tab and select ***PivotTable***. See Display 2



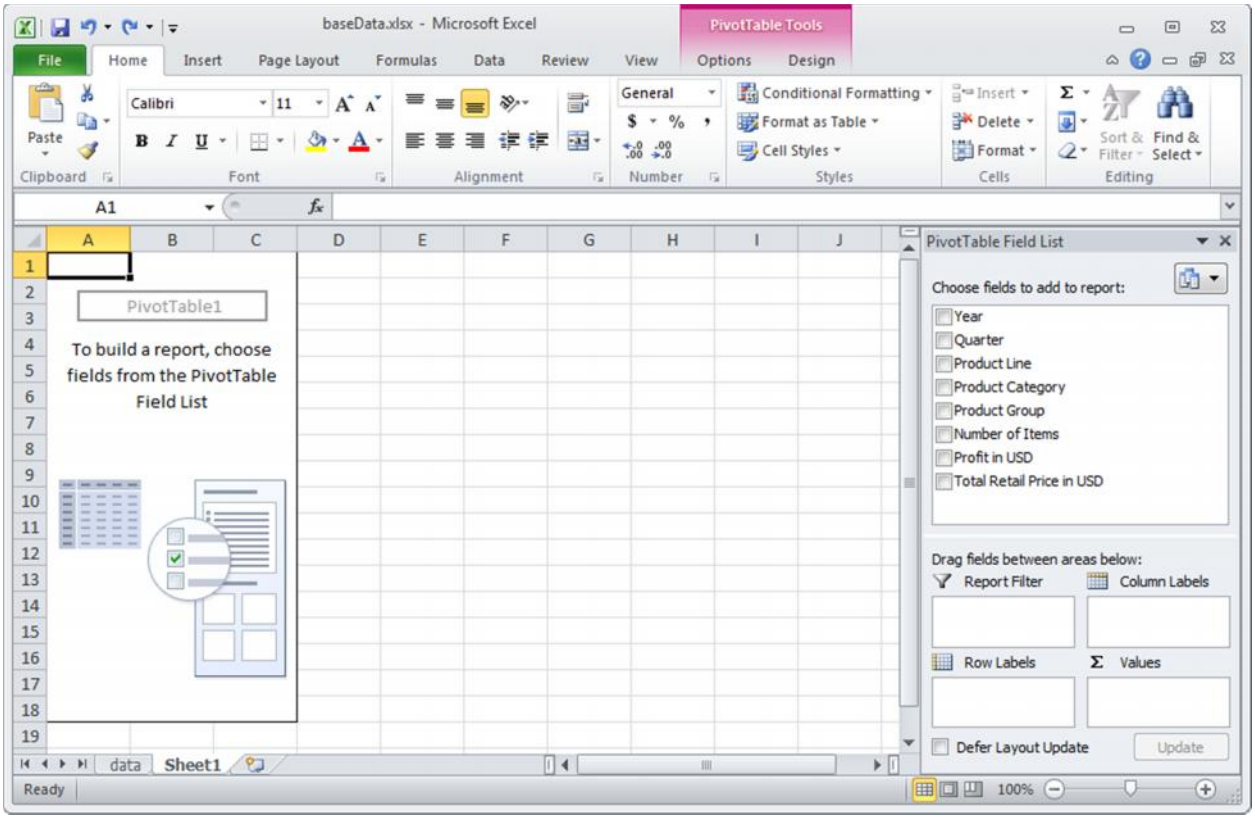
Display 2. Select PivotTable from Insert tab

2. In the Create PivotTable dialogue box select the data source (Select table or range) and the worksheet that will hold the PivotTable (Choose where you want the PivotTable to be placed). See Display 3.



Display 3. Select source data and the PivotTable destination

3. Once you select the data and destination Excel will present the interface to allow you to select the fields that go into each of the components. You can drag-and-drop fields from the PivotTable Field list into the component area. If you drop a field into the wrong area you can drag it to the correct area or drag it back to the field list. See Display 4



Display 4. The interface to drag-and-drop fields to the appropriate component area.

4. After creating a PivotTable you can easily change the layout/content by changing the fields in the component areas. For example, in Display 1 Product_Group is in the Row Labels. To change this from Product_Group to Product_Category simply drag Product_Category into the Row Labels area and drag Product_Group back to the Field List area. Note, the order in which you drag-and-drop is not important; that is you can first drag the field out of the area then drag the new field into the area.

Of course there are virtually unlimited amounts of formatting and customization that can be done in Excel. The best way to discover them is to open a workbook, create a PivotTable, and try out the options.

EXCEL PIVOT TABLES – FROM SAS

To create PivotTables from SAS you only need Base SAS and the ***TableEditor*** tagset ; since the TableEditor tagset is not included in the SAS distribution you will need to download it from the SAS website. At the time of writing the location is:

`support.sas.com/rnd/base/ods/odsmarkup/tableeditor/index.html`

Once the TableEditor file is downloaded you can open and submit it in SAS.

Note that at the time of writing some of the tagset options did not work properly in SAS 9.3; errors are generated although the PivotTable appears to be created correctly. All of the examples in this paper were written and tested in SAS 9.2 on a Windows 7 64 bit computer.

As noted, to create PivotTables SAS uses an ODS tagset destination. Since you have access to the tagset source code you can customize the tagset to meet your specific needs. For this paper no tagset customizations were made.

In order to create a PivotTable you wrap some ODS around code that will create the data table upon which the PivotTable is based. The ODS will cause SAS will create an HTML page with the data along with a command button. Through the use of JavaScript the code behind the button on the HTML page will generate the necessary objects to start Excel and create the PivotTable using the ODS options you supply. For more discussion on the TableEditor tagset refer to the papers by Parker cited in the references.

Since this paper and the supporting workshop are intended to show the various options that can be used to create PivotTables the examples are very simple – they have an ODS statement with the options being demonstrated followed by a simple PROC Print. All the examples can be run by using the ***sashelp.orasales*** table distributed with SAS. There are a number of SAS macro variables used to define paths; see Appendix 2 for the autoexec.sas code that sets these variables.

TableEditor Options

The TableEditor can take a variety of options; most of the option names are have self-evident meanings. For a list of all the options available and a brief description of the option see Appendix 1.

Example 1 – Creating a PivotTable

The first example will show how a PivotTable is created in SAS. We have already walked through creating a PivotTable in Excel.

```
ods tagsets.tableeditor file="&resultsHome\example1.html"
options(
        button_text = "Create PivotTable"
        auto_excel  = "yes"
        pivotrow    = "product_line"
        pivotcol    = "quarter"
        pivotdata   = "profit"
        pivotpage   = "year"
        excel_save_file="&JavaResultsHome\example1.xlsx"
        quit="NO"
       );
Title1 "Example 1 - Create the First Pivot Table";
proc print data=data.sales;
run;
ods tagsets.tableeditor close;
```

Example one is the template for all the other examples. As can be seen there is a simple PROC Print surrounded by ODS commands. In the example we specify the four main PivotTable components; notice that the tagset options have similar names to the Excel 'command'. The important mappings all the tables will have are:

Tagset Option	Excel Equivalent	Description
pivotpage	Report (page) filter	Subsets the data in the table. NOT REQUIRED
pivotcol	Column Labels	variable for the PivotTable columns
pivotrow	Row Labels	variable for the PivotTable rows
pivotdata	Summary Values	the cell variable

Let's look at the other options:

- button_text = "Create PivotTable"
  The text to put on the command button on the HTML form. The default test is "Exprort"
- auto_excel = "yes"
  Start Excel when the page is loaded
- excel_save_file="&JavaResultsHome\example1.xlsx"
  Tells Excel to save the file with the name supplied.
- quit="NO"
  Automatically quit Excel. Default is NO.

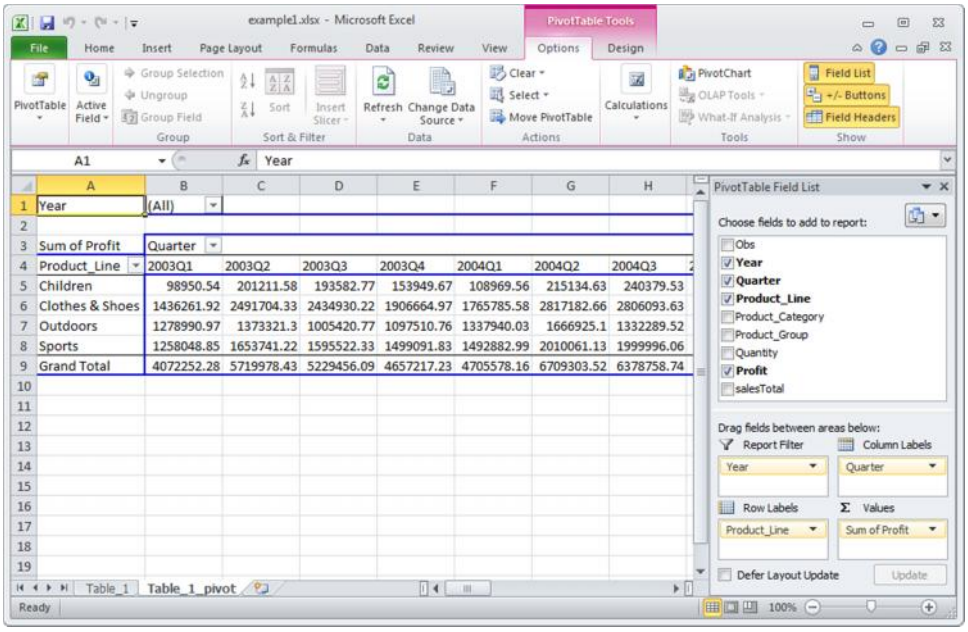
This example will automatically open Excel, create the PivotTable, and save the workbook as example1.xlsx in the folder specified by &JavaResultsHome. Excel will be automatically opened, but because quit="NO" was specified Excel stays open. See Display 5 for the HTML page generated by SAS, and Display 6 to see the Excel PivotTable created.


Create PivotTable

Example 1 - Create the First Pivot Table

Obs	Year	Quarter	Product_Line	Product_Category	Product_Group	Quan
1	2003	2003Q1	Children	Children Sports	A-Team, Kids	
2	2003	2003Q1	Children	Children Sports	Bathing Suits, Kids	
3	2003	2003Q1	Children	Children Sports	Eclipse, Kid's Clothes	
4	2003	2003Q1	Children	Children Sports	Eclipse, Kid's Shoes	
5	2003	2003Q1	Children	Children Sports	Lucky Guy, Kids	
6	2003	2003Q1	Children	Children Sports	Mad Dash	
7	2003	2003Q1	Children	Children Sports	N.D. Gear, Kids	
8	2003	2003Q1	Children	Children Sports	Olssons, Kids	
9	2003	2003Q1	Children	Children Sports	Orion Kid's Clothes	

Display 5. The HTML Page generated by SAS



Year	(All)							
Sum of Profit	Quarter							
Product_Line	2003Q1	2003Q2	2003Q3	2003Q4	2004Q1	2004Q2	2004Q3	
Children	98950.54	201211.58	193582.77	153949.67	108969.56	215134.63	240379.53	
Clothes & Shoes	1436261.92	2491704.33	2434930.22	1906664.97	1765785.58	2817182.66	2806093.63	
Outdoors	1278990.97	1373321.3	1005420.77	1097510.76	1337940.03	1666925.1	1332289.52	
Sports	1258048.85	1653741.22	1595522.33	1499091.83	1492882.99	2010061.13	1999996.06	
Grand Total	4072252.28	5719978.43	5229456.09	4657217.23	4705578.16	6709303.52	6378758.74	

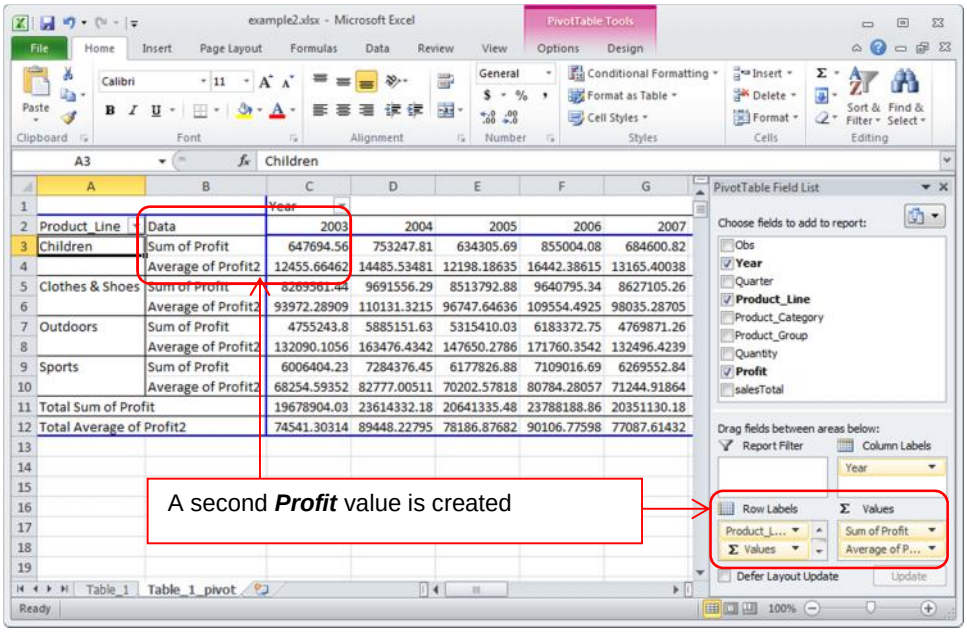
Display 6. The Excel PivotTable generated by SAS

Example 2 – Multiple Summary Values

In the discussion on PivotTables and in Example 1 we only had one Summary Value – Sum of Profit. Excel will allow us to have multiple Summary Values. We will first look at creating multiple Summary Values in the Excel interface, then create a PivotTable with multiple Summary Values through SAS.

In Excel. to add Average Profit as one of the Summary Values fields:

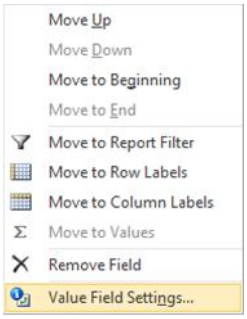
1. Drag Profit from the Field List to the Summary Values area. Since the default statistic is sum notice the new entry in the Summary Values area is listed as ***Sum of Profit 2***. Since I wanted to see the two summary statistics one under the other I moved statistic from Column Labels to Row Labels. See Display 7



		Year				
Product_Line	Data	2003	2004	2005	2006	2007
Children	Sum of Profit	647694.56	753247.81	634305.69	855004.08	684600.82
	Average of Profit2	12455.66462	14485.53481	12198.18635	16442.38615	13165.40038
Clothes & Shoes	Sum of Profit	8269561.44	9691556.29	8513792.88	9640795.34	8627105.26
	Average of Profit2	93972.28909	110131.3215	96747.64636	109554.4925	98035.28705
Outdoors	Sum of Profit	4755243.8	5885151.63	5315410.03	6183372.75	4769871.26
	Average of Profit2	132090.1056	163476.4342	147650.2786	171760.3542	132496.4239
Sports	Sum of Profit	6006404.23	7284376.45	6177826.88	7109016.69	6269552.84
	Average of Profit2	68254.59352	82777.00511	70202.57818	80784.28057	71244.91864
Total Sum of Profit		19678904.03	23614332.18	20641335.48	23788188.86	20351130.18
Total Average of Profit2		74541.30314	89448.22795	78186.87682	90106.77598	77087.61432

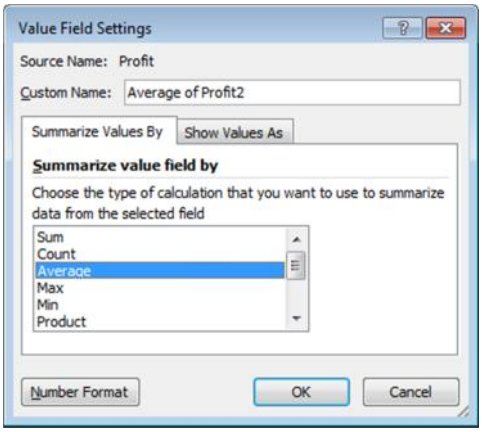
Display 7. A second field added to Summary Values

2. Right on the “more” arrow beside Sum of Profit 2. This will bring up a menu – select ValueField Settings. (see Display 8.



Display 8. “More” options for Sum of Profit2

3. Select Average from the list and click OK. See Display 9.



Display 9. Changing the summary to Average

4. The updated PivotTable is displayed. See Display 10.

To create a similar table using SAS we submit the following code. Note, to demonstrate that different fields in addition to different statistics for the same field can be created this example creates total profit and average quantity. See Display 11 for the PivotTable.

```
ods tagsets.tableeditor file="&resultsHome\example2.html"
options(
        button_text = "Excel"
        pivotrow  = "product_line"
        pivotcol  = "year"
        pivotdata = "profit,quantity"
        pivotdata_stats = "sum,average"       );
Title1 "Example 2 - Change the Statistics";
proc print data=data.sales;
run;
ods tagsets.tableeditor close;
```

In this code example we see one new option and a different usage of a previous option

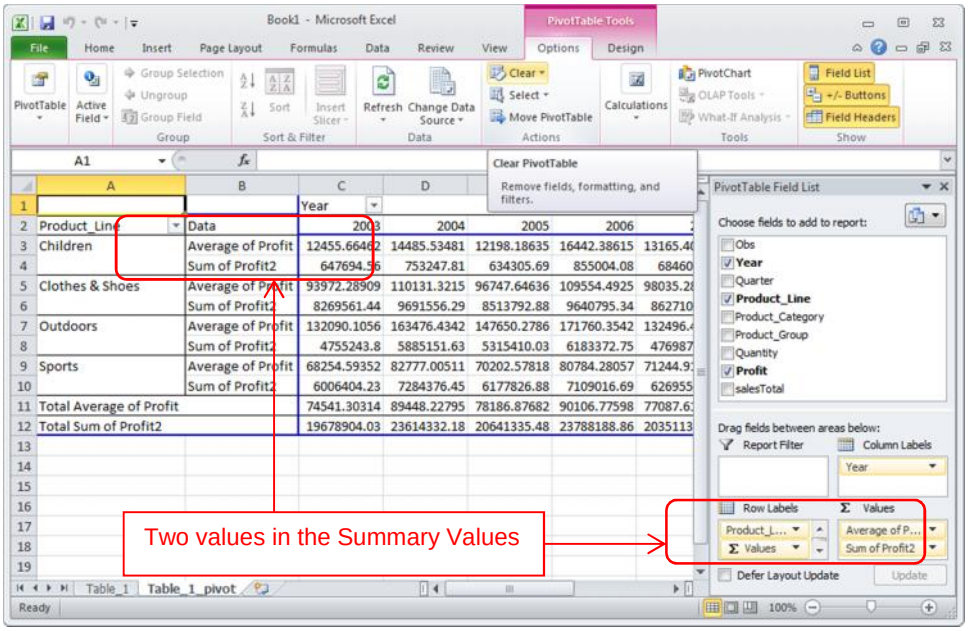
- pivotdata = "profit,quantity"

  This tells Excel we want multiple fields in the data area – profit and quantity. We separate the field with a comma

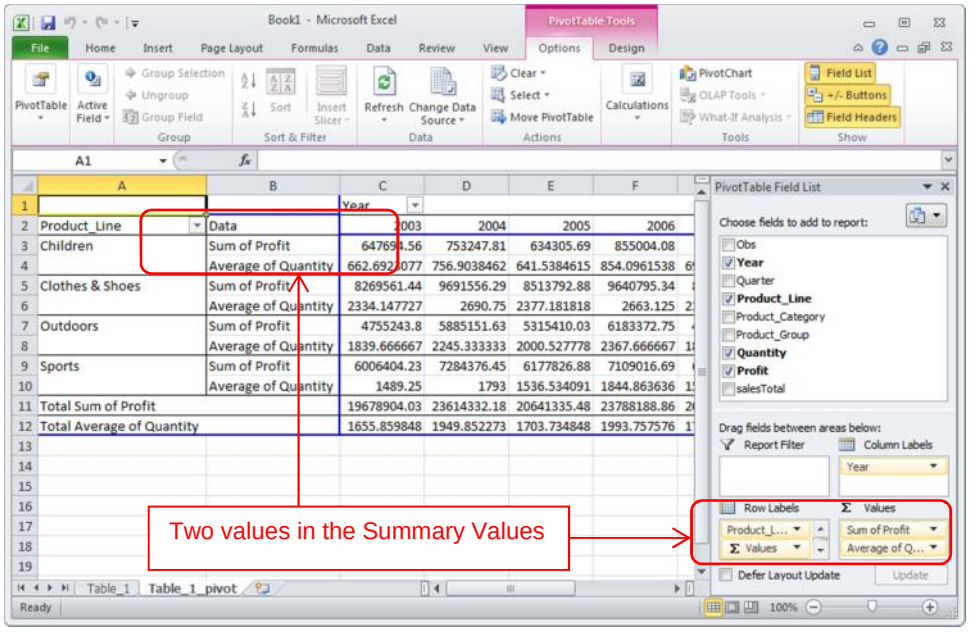
- pivotdata_stats = "sum,average"

  This tells Excel we want two statistics – sum and average. We separate the statistics with a comma. If no pivotdata_stats are specified, all the fields get the sum statistic. If only one statistic is specified, all fields get that statistic. It is a good practice to specify the statistic for all fields, even if they will all get the same statistic

As we specified multiple fields to be included in Summary Statistics we can also specify multiple fields in any of the other areas. In all cases we enter the list of fields separated by a comma.



Display 10. PivotTable after changing the summary statistic in Excel



Display 11. Example 2 PivotTable with two Summary Values from SAS

Example 3 – Multiple PivotTables

In Example 2 we looked at adding multiple fields into the Summary Values area, and by extension into the other areas of the PivotTable. What if we want each Summary Statistic on a different PivotTable. In Excel we would add a sheet for each PivotTable we want, then step through the process of creating the PivotTable; see the section "Creating PivitTables" above for this process.

In SAS we have a options to specify we are going to create multiple PivotTables - ***pivot_series***. Specifying pivot_series="yes" will tell Excel to create multiple PivotTables from the one data table. In addition to telling Excel we want multiple PivotTables we have to indicate what will be in the areas of each of the PivotTables. To do this we make another change to how we specify the areas:

```
ods tagsets.tableeditor file="&resultsHome\example3.html"
options(
        button_text = "Excel"
        pivot_series="yes"
        pivotrow="Product_Line | Product_Line | Product_Line"
        pivotcol=" Quarter | Quarter | Quarter"
        pivotdata="Profit | Profit | Profit "
        pivotdata_stats="Min | Max | Average"
);
Title1 "Example 3 - Multiple Sheets";
proc print data=data.sales;
run;
ods tagsets.tableeditor close;
```

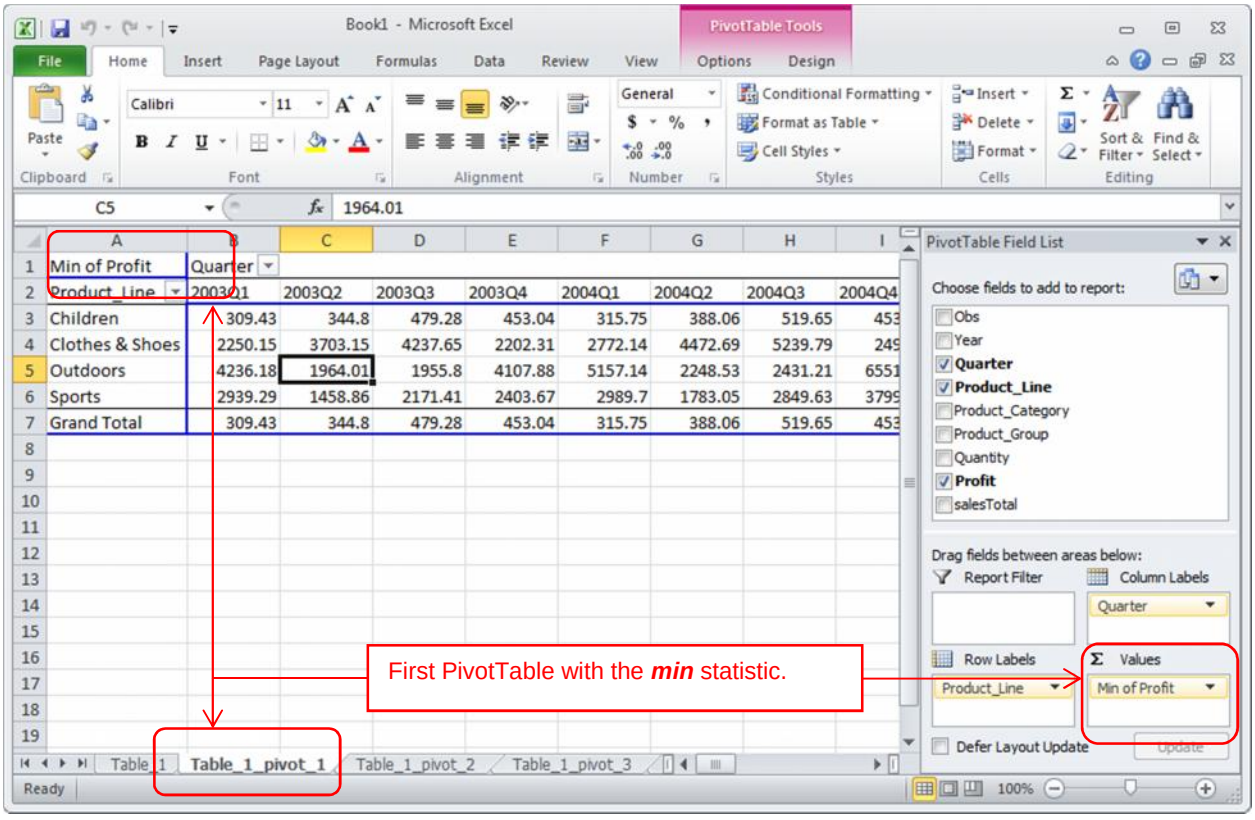
To specify we wanted multiple fields in one area of a PivotTable we separated the list of fields using a comma (,). To specify the fields we want in each area in each PivotTable we separate the fields with the vertical bar (|). In the example above we

- pivotrow="Product_Line | Product_Line | Product_Line"
  we will create three PivotTables, each will have Product_Line as the Row Label
- pivotcol=" Quarter | Quarter | Quarter"
  we will create three PivotTables, each will have Quarter as the Column Label
- pivotdata="Profit | Profit | Profit "

we will create three PivotTables, each will have Profit as the Summary Field

- pivotdata_stats="Min | Max | Average"
  we will create three PivotTables, the first Summary Field will have the Min statistic, the second the Max statistic, and the third the Average statistic.

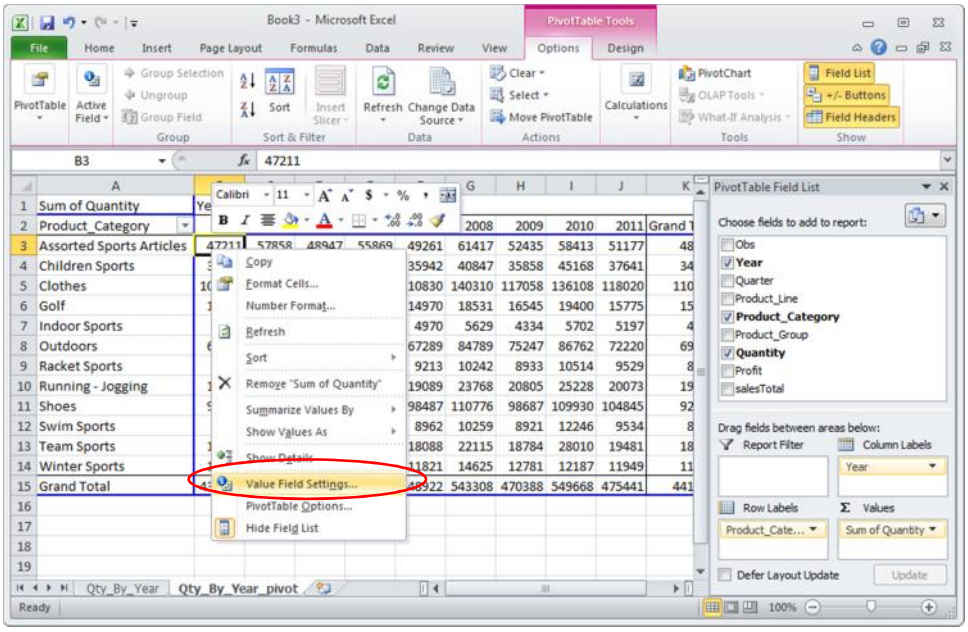
See Display 12 for the Excel workbook created by this code.



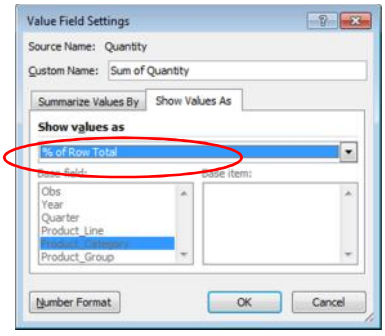
Display 12. The Excel Workbook with multiple PivotTables.

Example 4 – Percentages

Instead of showing values in the cells, we may want to calculate the percent of that cell of row total, column total, table total etc. In Excel we can do this by changing the Value Field Setting; we saw the use of the Value Field dialogue earlier. To change from a number to a percentage, right click on any value cell in the PivotTable and select Value Field Settings from the context menu; see Display 13. When the Value Field Settings dialogue appears, select the Show Values As tab and select the appropriate base for our percentage. In Display 14 the % of Row Total was selected. Display 14 shows the resulting PivotTable.



Display 13. Getting the Value Field Setting from the context menu

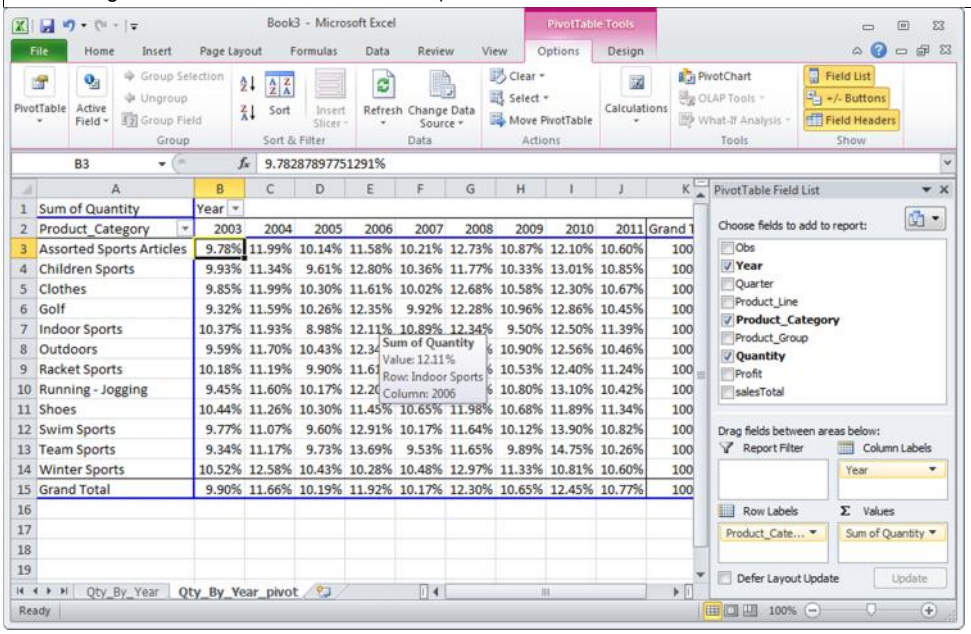


Display 14. Selecting % of Row Total

To perform the same type of calculation in SAS we use the ***pivotCalc*** option; the pivotCalc option tells Excel to create a perentage and the basis of the percentage. In this example we are creating a % of row total. Valid values are ***row***, ***column***, ***total***, ***percentOf***, ***index***, and ***runningTotal***. See Display 16 for the Excel PivotTable created from this code.

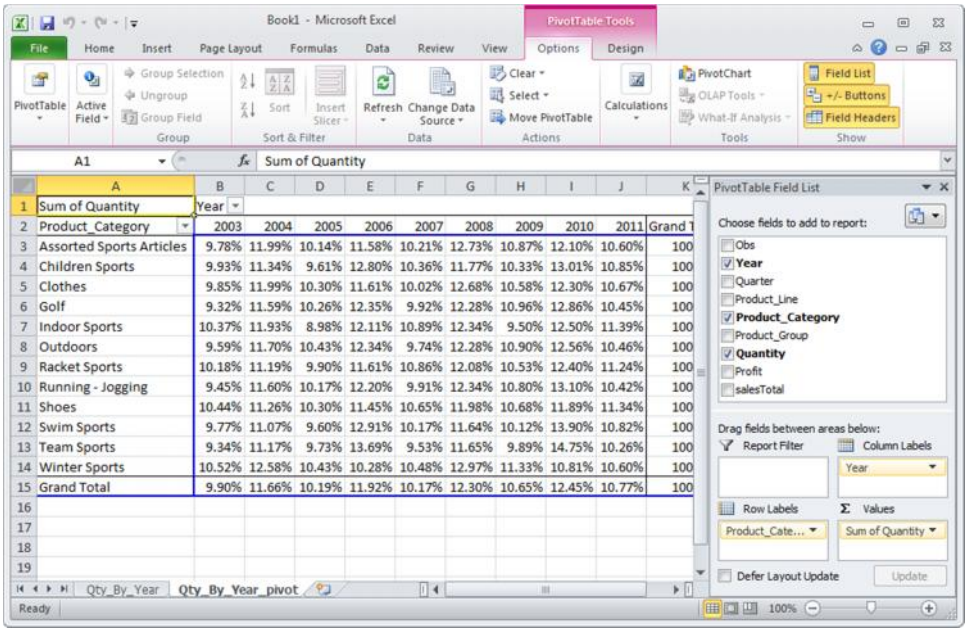
```
ods tagsets.tableeditor
file="&resultsHome\example4.html"
options(
        button_text = "Excel"
        sheet_name = "Qty_By_Year"
        pivotrow   = "product_category"
        pivotcol   = "year"
        pivotdata  = "quantity"
        pivotdata_stats = "sum"
        pivotcalc = "row"
        );

Title1 "Example 4 - Calculating Pecentages";
proc print data=data.sales;
run;
```

```
ods tagsets.tableeditor close;
```



Sum of Quantity	Year									
Product_Category	2003	2004	2005	2006	2007	2008	2009	2010	2011	Grand T
Assorted Sports Articles	9.78%	11.99%	10.14%	11.58%	10.21%	12.73%	10.87%	12.10%	10.60%	100
Children Sports	9.93%	11.34%	9.61%	12.80%	10.36%	11.77%	10.33%	13.01%	10.85%	100
Clothes	9.85%	11.99%	10.30%	11.61%	10.02%	12.68%	10.58%	12.30%	10.67%	100
Golf	9.32%	11.59%	10.26%	12.35%	9.92%	12.28%	10.96%	12.86%	10.45%	100
Indoor Sports	10.37%	11.93%	8.98%	12.11%	10.89%	12.34%	9.50%	12.50%	11.39%	100
Outdoors	9.59%	11.70%	10.43%	12.34			10.90%	12.56%	10.46%	100
Racket Sports	10.18%	11.19%	9.90%	11.61			10.53%	12.40%	11.24%	100
Running - Jogging	9.45%	11.60%	10.17%	12.2			10.80%	13.10%	10.42%	100
Shoes	10.44%	11.26%	10.30%	11.45%	10.65%	11.98%	10.68%	11.89%	11.34%	100
Swim Sports	9.77%	11.07%	9.60%	12.91%	10.17%	11.64%	10.12%	13.90%	10.82%	100
Team Sports	9.34%	11.17%	9.73%	13.69%	9.53%	11.65%	9.89%	14.75%	10.26%	100
Winter Sports	10.52%	12.58%	10.43%	10.28%	10.48%	12.97%	11.33%	10.81%	10.60%	100
Grand Total	9.90%	11.66%	10.19%	11.92%	10.17%	12.30%	10.65%	12.45%	10.77%	100

Display 15. PivotTable with Percentages



Sum of Quantity	Year									
Product_Category	2003	2004	2005	2006	2007	2008	2009	2010	2011	Grand T
Assorted Sports Articles	9.78%	11.99%	10.14%	11.58%	10.21%	12.73%	10.87%	12.10%	10.60%	100
Children Sports	9.93%	11.34%	9.61%	12.80%	10.36%	11.77%	10.33%	13.01%	10.85%	100
Clothes	9.85%	11.99%	10.30%	11.61%	10.02%	12.68%	10.58%	12.30%	10.67%	100
Golf	9.32%	11.59%	10.26%	12.35%	9.92%	12.28%	10.96%	12.86%	10.45%	100
Indoor Sports	10.37%	11.93%	8.98%	12.11%	10.89%	12.34%	9.50%	12.50%	11.39%	100
Outdoors	9.59%	11.70%	10.43%	12.34%	9.74%	12.28%	10.90%	12.56%	10.46%	100
Racket Sports	10.18%	11.19%	9.90%	11.61%	10.86%	12.08%	10.53%	12.40%	11.24%	100
Running - Jogging	9.45%	11.60%	10.17%	12.20%	9.91%	12.34%	10.80%	13.10%	10.42%	100
Shoes	10.44%	11.26%	10.30%	11.45%	10.65%	11.98%	10.68%	11.89%	11.34%	100
Swim Sports	9.77%	11.07%	9.60%	12.91%	10.17%	11.64%	10.12%	13.90%	10.82%	100
Team Sports	9.34%	11.17%	9.73%	13.69%	9.53%	11.65%	9.89%	14.75%	10.26%	100
Winter Sports	10.52%	12.58%	10.43%	10.28%	10.48%	12.97%	11.33%	10.81%	10.60%	100
Grand Total	9.90%	11.66%	10.19%	11.92%	10.17%	12.30%	10.65%	12.45%	10.77%	100

Display 16. PivotTable with Percentages from SAS

Example 5 – Formatting Numbers

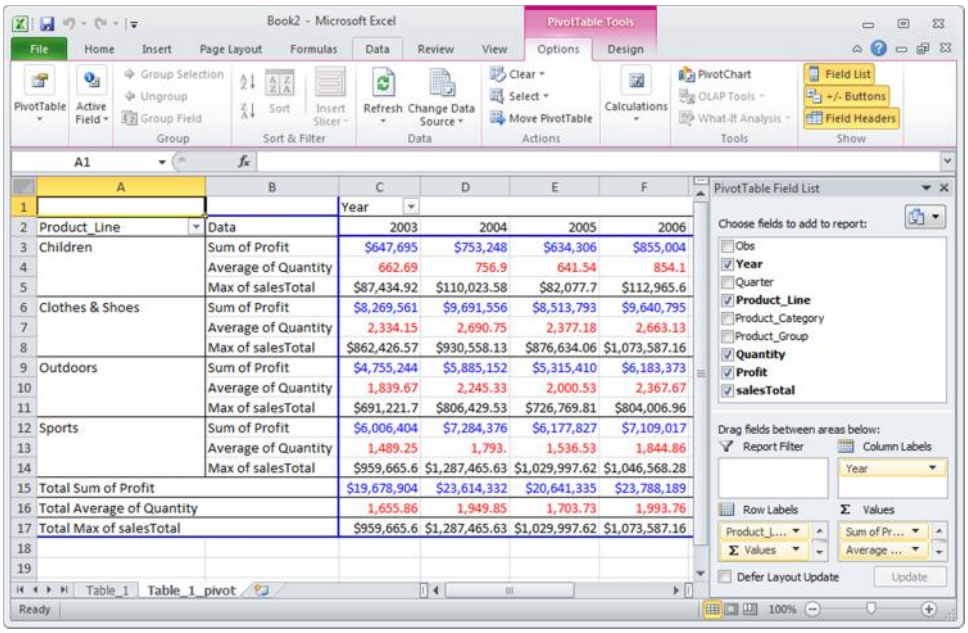
When Excel creates a PivotTable any formatting of the source data is not honoured. If you want the values to be formatted (say as currency) then you must apply the format after the PivotTable is created. As with any standard work sheet cell or range you can right click and format the cell; although this works you should go through the PivotTable interface to format the area. Once again you bring up the Value Field Settings dialogue; this time click on the ***Number Format*** button and apply the format that is appropriate. The dialogue box presented is the same dialogue used to format cells in any worksheet. After you click out of the dialogue the formats are applied.

When using the ***Custom Format*** option in Excel you can specify formats for the four main value types of a number – when positive, when negative, when zero, and when the cell contains text; different formats for each of these value types can be entered, sepatated by a semi-colon (;); if only one format is specifies all value types get this format. See the Excel help for all the of options available to format a number. In our examples we will apply a different colour to each Summary Field, that is for Profit the cell font will be blue, for Quantity the cell font will be red, and for salesTotal the cell font will be the default (black). In addition, the Profit and salesTotal fields will have a dollar sign ($). The PivotTable created by this code is in Display 17.

```
ods tagsets.tableeditor
    file="&resultsHome\example5.html"
    options(
            button_text="Excel"
            pivotrow="product_line"
            pivotcol="year"
            pivotdata="Profit,Quantity,salesTotal"
            pivotdata_stats="Sum,Average,Max"
            pivotdata_fmt="[blue] #,###~[red] $#,###.##~$#,###.##"
           );

Title1 "Example 5 - Formatting Numbers in the Pivot Table";
proc print data=data.sales;
run;

ods tagsets.tableeditor close;
```

The option to format the Summary Vallues is ***pivotdata_fmt***. We ***must*** provide one set of formats for each field in our pivotdata list. Also note that we have another delimiter here, the tilde (~) – each format set is separated with a tilde. In this example we are only providing one format for each Summay Field (eg [blue] #,### for Profit). If we want to provide formats for negative and zero values we would use the semi-colon to separate them as we would in Excel , for example to provide a format for positive, negative and zero values only for Profit we would use:

```
pivotdata_fmt="[blue] #,###;[red] (#,###); #,###~[green] $#,###.##~$#,###.##"
```

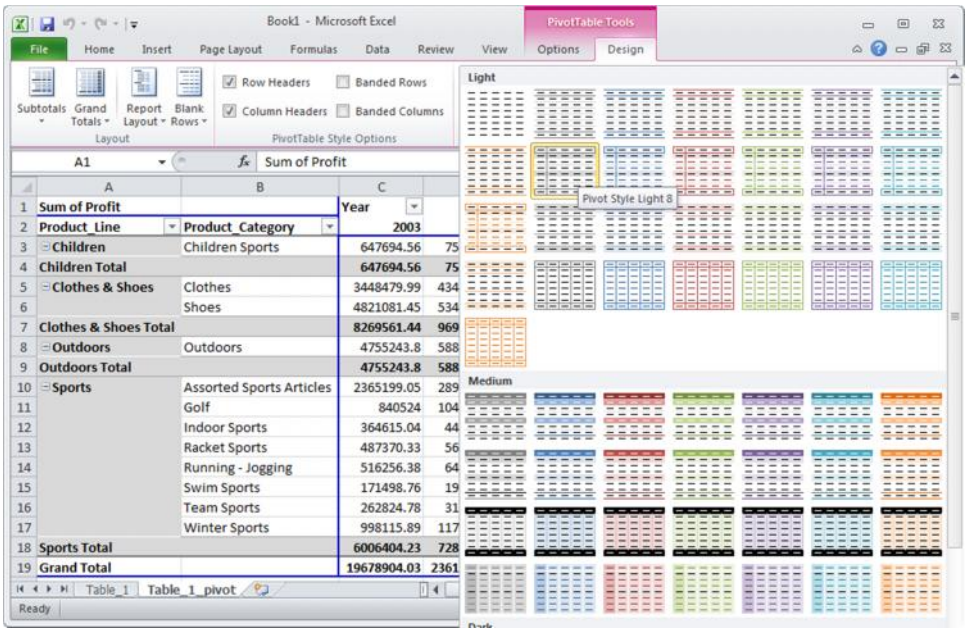


	A	B	C	D	E	F
1			Year			
2	Product_Line	Data	2003	2004	2005	2006
3	Children	Sum of Profit	$647,695	$753,248	$634,306	$855,004
4		Average of Quantity	662.69	756.9	641.54	854.1
5		Max of salesTotal	$87,434.92	$110,023.58	$82,077.7	$112,965.6
6	Clothes & Shoes	Sum of Profit	$8,269,561	$9,691,556	$8,513,793	$9,640,795
7		Average of Quantity	2,334.15	2,690.75	2,377.18	2,663.13
8		Max of salesTotal	$862,426.57	$930,558.13	$876,634.06	$1,073,587.16
9	Outdoors	Sum of Profit	$4,755,244	$5,885,152	$5,315,410	$6,183,373
10		Average of Quantity	1,839.67	2,245.33	2,000.53	2,367.67
11		Max of salesTotal	$691,221.7	$806,429.53	$726,769.81	$804,006.96
12	Sports	Sum of Profit	$6,006,404	$7,284,376	$6,177,827	$7,109,017
13		Average of Quantity	1,489.25	1,793.	1,536.53	1,844.86
14		Max of salesTotal	$959,665.6	$1,287,465.63	$1,029,997.62	$1,046,568.28
15	Total Sum of Profit		$19,678,904	$23,614,332	$20,641,335	$23,788,189
16	Total Average of Quantity		1,655.86	1,949.85	1,703.73	1,993.76
17	Total Max of salesTotal		$959,665.6	$1,287,465.63	$1,029,997.62	$1,073,587.16
18						
19						

Display 17. PivotTable with Formats from SAS

Example 6 – Applying Excel Styles

Excel provides a number of built-in styles that can be applied to the PivotTable; these styles are PivotTable design tab. In Excel 2010, if you run your mouse over any of the styles Excel will display your PivotTable in that style; this gives you an opportunity to see how the style would appear with your data without having to apply the style. Display 18 shows these styles. In Display 18, the mouse is over ***Pivot Style Light 8*** (note the pop-up window beside the style). You can use this style name to apply in your code.

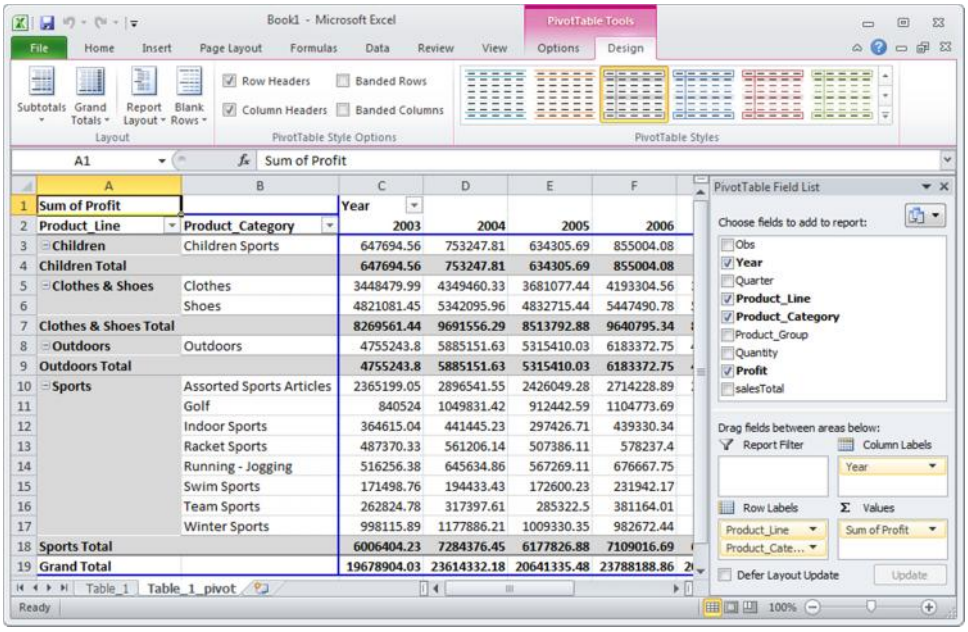


	A	B	C	
1	Sum of Profit		Year	
2	Product_Line	Product_Category	2003	
3	Children	Children Sports	647694.56	75
4	Children Total		647694.56	75
5	Clothes & Shoes	Clothes	3448479.99	434
6		Shoes	4821081.45	534
7	Clothes & Shoes Total		8269561.44	969
8	Outdoors	Outdoors	4755243.8	588
9	Outdoors Total		4755243.8	588
10	Sports	Assorted Sports Articles	2365199.05	289
11		Golf	840524	104
12		Indoor Sports	364615.04	44
13		Racket Sports	487370.33	56
14		Running - Jogging	516256.38	64
15		Swim Sports	171498.76	19
16		Team Sports	262824.78	31
17		Winter Sports	998115.89	117
18	Sports Total		6006404.23	728
19	Grand Total		19678904.03	2361

Display 18. Previewing a Style in Excel

To apply the same style in SAS we use the pivot_format option. You will need to use the interface and mouse to roll over the different styles to get the style name. In our experience, not all of the styles get applied correctly so you may have to experiment with different style names. In this example ***light8*** was applied. See Display 19 for the results.

```
ods tagsets.tableeditor
file="&resultsHome\example6.html"
options(
        button_text = "Excel"
        pivotrow="product_line,product_category"
           pivotcol="year"
           pivotdata="profit"
           pivot_format="light8"
     );

Title1 "Example 6 - Applying Excel Styles";
proc print data=data.sales;
run;

ods tagsets.tableeditor close;
```



	A	B	C	D	E	F
1	Sum of Profit		Year			
2	Product_Line	Product_Category	2003	2004	2005	2006
3	Children	Children Sports	647694.56	753247.81	634305.69	855004.08
4	Children Total		647694.56	753247.81	634305.69	855004.08
5	Clothes & Shoes	Clothes	3448479.99	4349460.33	3681077.44	4193304.56
6		Shoes	4821081.45	5342095.96	4832715.44	5447490.78
7	Clothes & Shoes Total		8269561.44	9691556.29	8513792.88	9640795.34
8	Outdoors	Outdoors	4755243.8	5885151.63	5315410.03	6183372.75
9	Outdoors Total		4755243.8	5885151.63	5315410.03	6183372.75
10	Sports	Assorted Sports Articles	2365199.05	2896541.55	2426049.28	2714228.89
11		Golf	840524	1049831.42	912442.59	1104773.69
12		Indoor Sports	364615.04	441445.23	297426.71	439330.34
13		Racket Sports	487370.33	561206.14	507386.11	578237.4
14		Running - Jogging	516256.38	645634.86	567269.11	676667.75
15		Swim Sports	171498.76	194433.43	172600.23	231942.17
16		Team Sports	262824.78	317397.61	285322.5	381164.01
17		Winter Sports	998115.89	1177886.21	1009330.35	982672.44
18	Sports Total		6006404.23	7284376.45	6177826.88	7109016.69
19	Grand Total		19678904.03	23614332.18	20641335.48	23788188.86

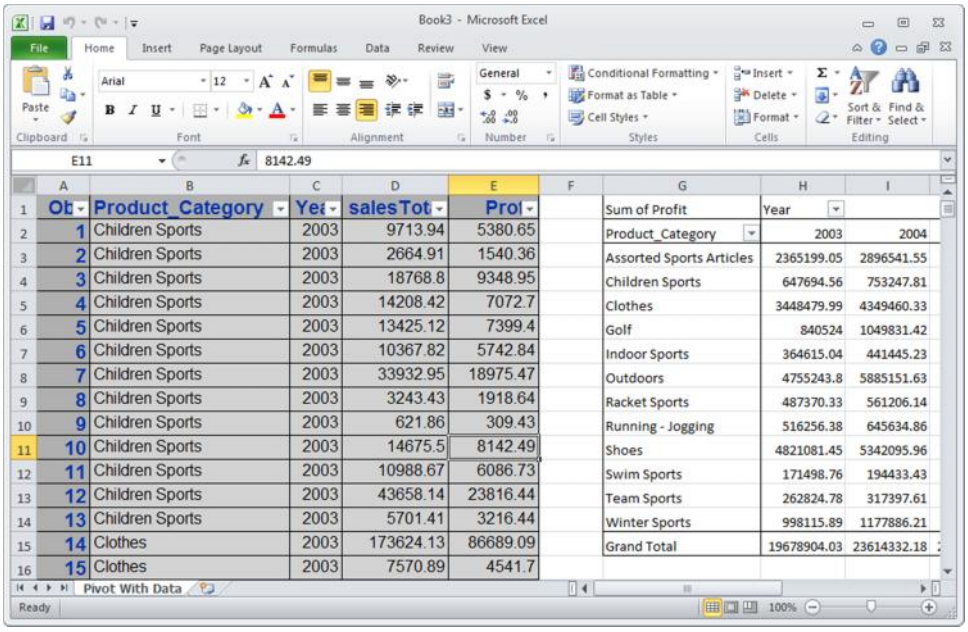
Display 19. A Style Applied by SAS

Example 7 – Placing the PivotTable on the same sheet as the data table

Normally we will be placing the PivotTable on a different sheet than the data table. First, even in Excel, the data tables can become very large and navigation of the sheet will be more difficult. Second, by separating the two we minimize the risk that the base table is inadvertently changed as the analyst is examining and changing the PivotTable. In order to place the PivotTable on the same sheet as the data table you simply select a PivotTable destination on the data sheet (see Display 3).

In SAS we use the ***ptdest*** option; this instructs Excel to write the PivotTable to the same sheet as the data and starting in the range specified; in the example the PivotTable will start in cell G1. Also note this example told Excel to apply an AutoFilter to the data table (***excel_autofilter***="yes"). The results are in Display 20.

```
ods tagsets.tableeditor
file="&resultsHome\example7.html"
options(
        button_text = "Excel"
        SHEET_NAME="Pivot With Data"
        excel_autofilter="yes"
        ptdest_range="g1"
        pivotrow="product_category"
        pivotcol="year"
        pivotdata="profit"
       );

Title1 "Example 7 - Placing the Pivot Table on the Data Sheet";
proc print data=data.sales;
var product_category year salesTotal profit;
run;

ods tagsets.tableeditor close;
```



Ob	Product_Category	Yea	salesTot	Prof
1	Children Sports	2003	9713.94	5380.65
2	Children Sports	2003	2664.91	1540.36
3	Children Sports	2003	18768.8	9348.95
4	Children Sports	2003	14208.42	7072.7
5	Children Sports	2003	13425.12	7399.4
6	Children Sports	2003	10367.82	5742.84
7	Children Sports	2003	33932.95	18975.47
8	Children Sports	2003	3243.43	1918.64
9	Children Sports	2003	621.86	309.43
10	Children Sports	2003	14675.5	8142.49
11	Children Sports	2003	10988.67	6086.73
12	Children Sports	2003	43658.14	23816.44
13	Children Sports	2003	5701.41	3216.44
14	Clothes	2003	173624.13	86689.09
15	Clothes	2003	7570.89	4541.7

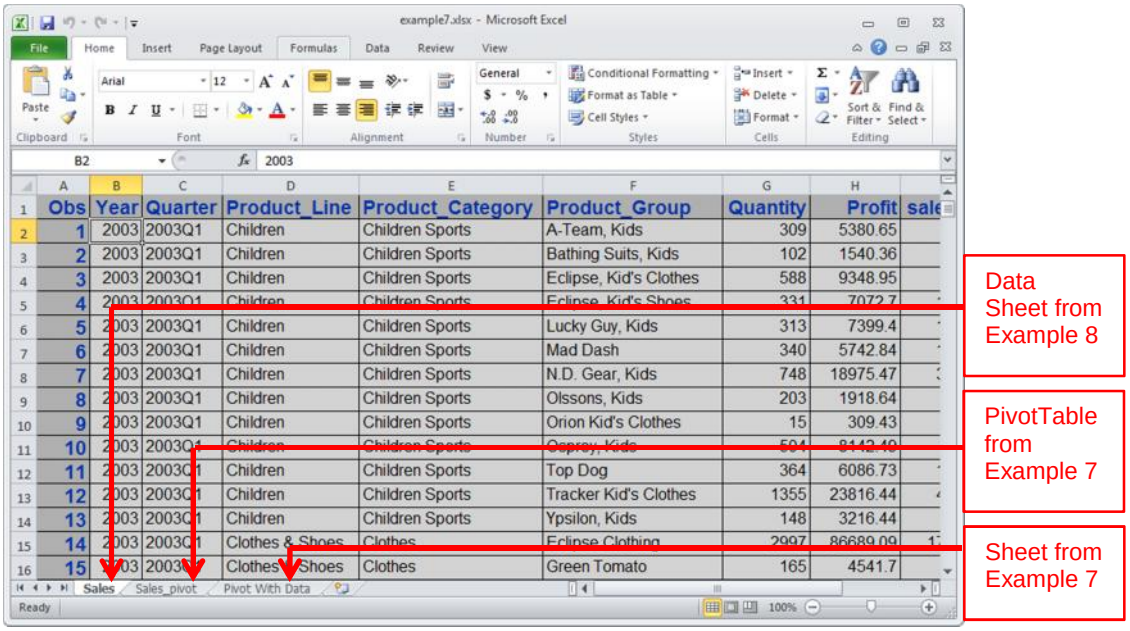
Sum of Profit	Year	
Product_Category	2003	2004
Assorted Sports Articles	2365199.05	2896541.55
Children Sports	647694.56	753247.81
Clothes	3448479.99	4349460.33
Golf	840524	1049831.42
Indoor Sports	364615.04	441445.23
Outdoors	4755243.8	5885151.63
Racket Sports	487370.33	561206.14
Running - Jogging	516256.38	645634.86
Shoes	4821081.45	5342095.96
Swim Sports	171498.76	194433.43
Team Sports	262824.78	317397.61
Winter Sports	998115.89	1177886.21
Grand Total	19678904.03	23614332.18

Display 20. Data and PivotTable on the same worksheet from SAS

Example 8 – Adding Sheets to an Existing Workbook

In Excel it is easy to add sheets to an existing workbook. In this example we will add a new sheet and PivotTable to the workbook created in Example 7. The option to specify we are writing to an existing workbook is ***update_target***; the value of the option is the fully qualified name of the Excel workbook to update. Display 21 has the results of this program.

```
ods tagsets.tableeditor
file="&resultsHome\example8.html"
options(
        button_text = "Excel"
        update_target="&JavaResultsHome.\example7.xlsx"
        sheet_name="Sales"
        pivotrow="product_line"
        pivotcol="year"
        pivotdata="profit"
        pivot_format="medium5"
       );
Title1 "Example 8 - Adding Sheets to an Existing Workbook";
proc print data=data.sales;
run;
ods tagsets.tableeditor close;
```



Obs	Year	Quarter	Product_Line	Product_Category	Product_Group	Quantity	Profit	sale
1	2003	2003Q1	Children	Children Sports	A-Team, Kids	309	5380.65	
2	2003	2003Q1	Children	Children Sports	Bathing Suits, Kids	102	1540.36	
3	2003	2003Q1	Children	Children Sports	Eclipse, Kid's Clothes	588	9348.95	
4	2003	2003Q1	Children	Children Sports	Eclipse, Kid's Shoes	331	7072.7	
5	2003	2003Q1	Children	Children Sports	Lucky Guy, Kids	313	7399.4	
6	2003	2003Q1	Children	Children Sports	Mad Dash	340	5742.84	
7	2003	2003Q1	Children	Children Sports	N.D. Gear, Kids	748	18975.47	
8	2003	2003Q1	Children	Children Sports	Olssons, Kids	203	1918.64	
9	2003	2003Q1	Children	Children Sports	Orion Kid's Clothes	15	309.43	
10	2003	2003Q1	Children	Children Sports	Osprey, Kids	504	8442.49	
11	2003	2003Q1	Children	Children Sports	Top Dog	364	6086.73	
12	2003	2003Q1	Children	Children Sports	Tracker Kid's Clothes	1355	23816.44	
13	2003	2003Q1	Children	Children Sports	Ypsilon, Kids	148	3216.44	
14	2003	2003Q1	Clothes & Shoes	Clothes	Eclipse Clothing	2997	86689.09	
15	2003	2003Q1	Clothes & Shoes	Clothes	Green Tomato	165	4541.7	

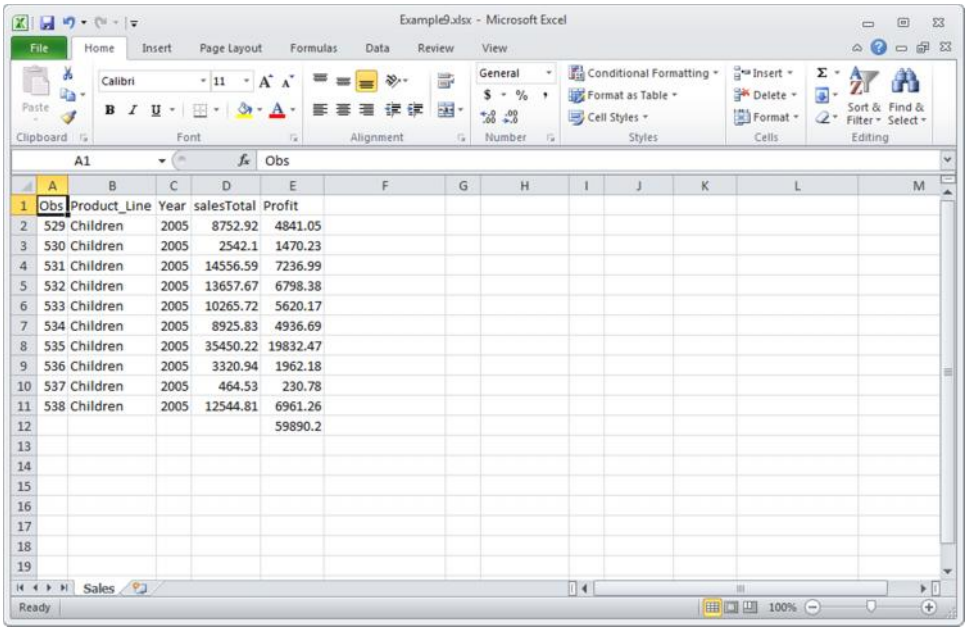
Display 21. Adding a Data Sheet and PivotTable to Existing Workbook from SAS

Example 9 – Updating a Range in an Existing Workbook

One common problem we often encounter is the need to update a range in an existing Excel workbook; perhaps there are reports and graphs in Excel that need to be refreshed on a daily basis. By only updating the data in the range you can keep the existing reports and graphs from having to be recreated every time the data are refreshed. If you have SAS/Access for PC File Formats you can read/write the named ranges directly, however if you do not have SAS/Access for PC File Formats you had to be more creative in order to update data in a range. Using the TableEditor tagset we can update a range. Display 22 shows a worksheet before we update it

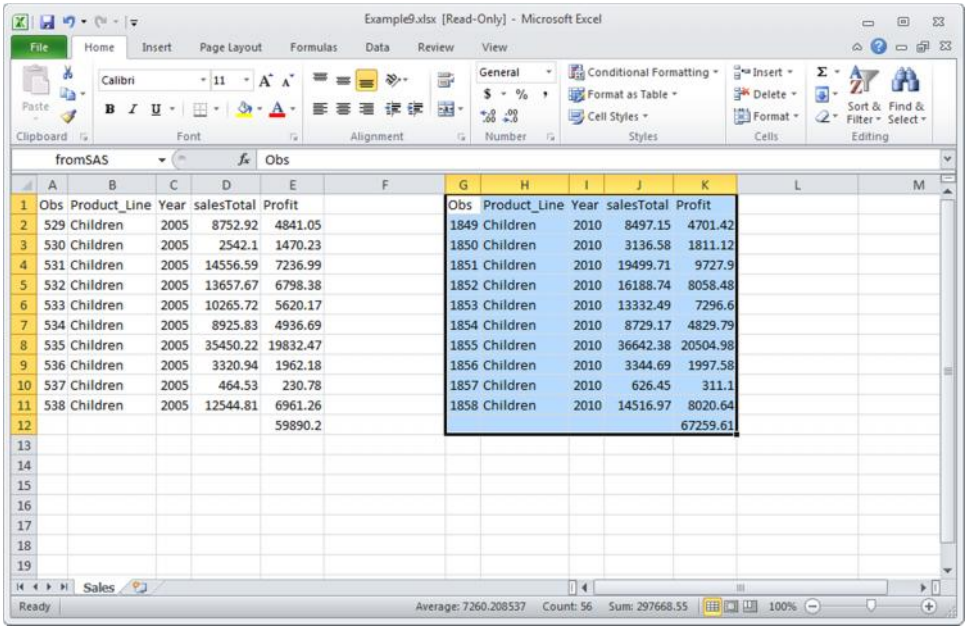
To tell Excel we want to update a range in an existing workbook we need to tell which workbook to update, which sheet in the workbook, and the starting location (row, col) to update. The options are update_target, update_sheet, and update_range respectively. In this example we are updating workbook Example9.xlsx, sheet Sales, and starting in row 1 column 11:

```
ods tagsets.tableeditor file="&resultsHome\example9.html"
options(
        button_text = "Excel"
        update_target="&javaDataHome\example9.xlsx"
        update_sheet="Sales"
        update_range="1,7"
       );

Title1 "Example 9 - Updating a Sheet in a Workbook";

proc print data=data.sales (obs=10 where=(year=2010));
var product_line year salesTotal profit;
sum profit;
run;

ods tagsets.tableeditor close;
```



Obs	Product_Line	Year	salesTotal	Profit
529	Children	2005	8752.92	4841.05
530	Children	2005	2542.1	1470.23
531	Children	2005	14556.59	7236.99
532	Children	2005	13657.67	6798.38
533	Children	2005	10265.72	5620.17
534	Children	2005	8925.83	4936.69
535	Children	2005	35450.22	19832.47
536	Children	2005	3320.94	1962.18
537	Children	2005	464.53	230.78
538	Children	2005	12544.81	6961.26
				59890.2

Display 22. An Existing Worksheet



Obs	Product_Line	Year	salesTotal	Profit
529	Children	2005	8752.92	4841.05
530	Children	2005	2542.1	1470.23
531	Children	2005	14556.59	7236.99
532	Children	2005	13657.67	6798.38
533	Children	2005	10265.72	5620.17
534	Children	2005	8925.83	4936.69
535	Children	2005	35450.22	19832.47
536	Children	2005	3320.94	1962.18
537	Children	2005	464.53	230.78
538	Children	2005	12544.81	6961.26
				59890.2
1849	Children	2010	8497.15	4701.42
1850	Children	2010	3136.58	1811.12
1851	Children	2010	19499.71	9727.9
1852	Children	2010	16188.74	8058.48
1853	Children	2010	13332.49	7296.6
1854	Children	2010	8729.17	4829.79
1855	Children	2010	36642.38	20504.98
1856	Children	2010	3344.69	1997.58
1857	Children	2010	626.45	311.1
1858	Children	2010	14516.97	8020.64
				67259.61

Display 23. An Existing Worksheet with updates starting in Cell G1 (1, 7)

Example 10 – Placing the PivotTable on the same sheet as the data table

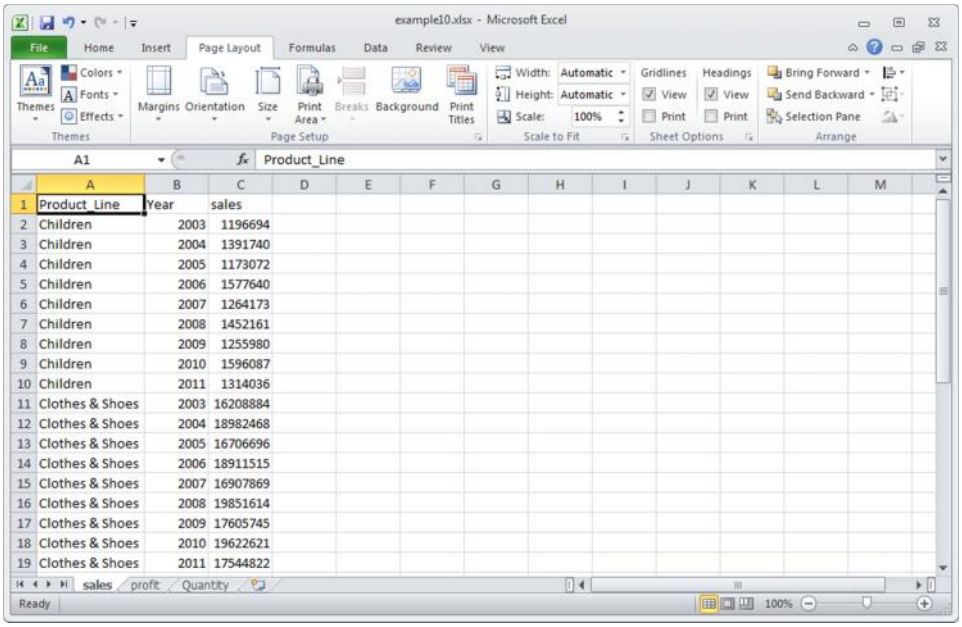
It is a sad reality that there are some that do not originate in SAS; sometimes spreadsheets are updated or created from other sources. Just because SAS did not generate the source data from the PivotTable does not mean we cannot use SAS to automate the creation of the PivotTable. In this example we will add PivotTables to a workbook with three existing data sheets. See Display 24.

Once again we need to identify the workbook (update_target) and sheet (sheet_name) with the data from which the PivotTable will be built; since we are also creating PivotTables we also need to supply the necessary options for the PivotTables. In this example we are creating three PivotTables, one for each of the sales, profit, and quantity sheet;

note the sheet names are separated by commas in the option. We are using the same field as the Row Label so it is specified three time separated by the vertical bar (|). Finally we specify the summary field for each of the PivotTables, again separated by the vertical bar.

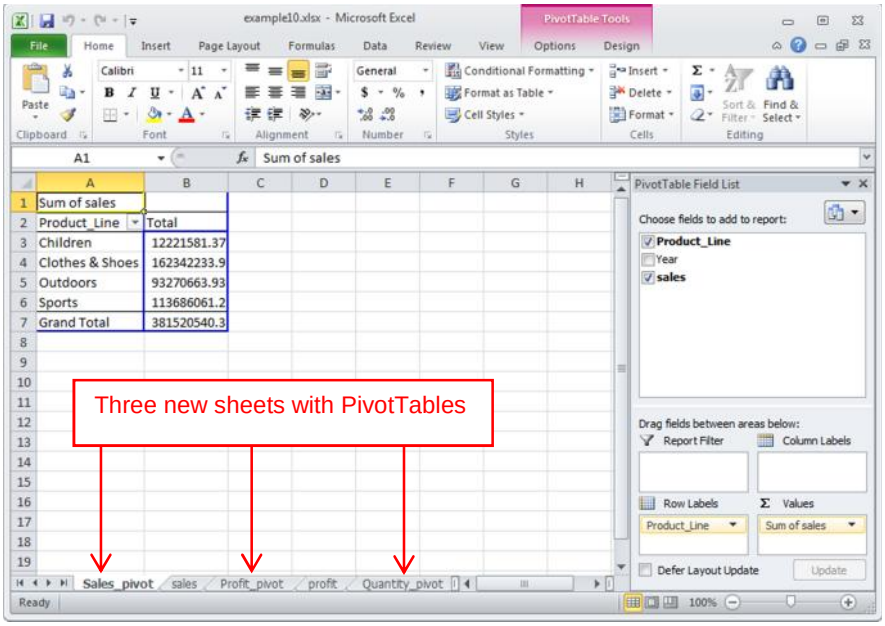
Since this example is not creating the data in an HTML page, we will create a “dummy” page with the DATA _NULL_. The results of this code are in Display 25.

```
ods tagsets.tableeditor file="&resultsHome\example10.html"
options(
        update_target="&javaDataHome\example10.xlsx"
        sheet_name="Profit, Quantity, Sales"
        pivotrow="Product_line |Product_line |Product_line"
        pivotdata="Profit |Quantity |Sales"
       );

Title1 "Example 10 - Using An Existing Worksheet As A Data Source";
data _null_;
file print;
put "Create Pivot Tables";
run;
ods tagsets.tableeditor close;
```



Product_Line	Year	sales
Children	2003	1196694
Children	2004	1391740
Children	2005	1173072
Children	2006	1577640
Children	2007	1264173
Children	2008	1452161
Children	2009	1255980
Children	2010	1596087
Children	2011	1314036
Clothes & Shoes	2003	16208884
Clothes & Shoes	2004	18982468
Clothes & Shoes	2005	16706696
Clothes & Shoes	2006	18911515
Clothes & Shoes	2007	16907869
Clothes & Shoes	2008	19851614
Clothes & Shoes	2009	17605745
Clothes & Shoes	2010	19622621
Clothes & Shoes	2011	17544822

Display 24. Data Sheets in an Existing Workbook

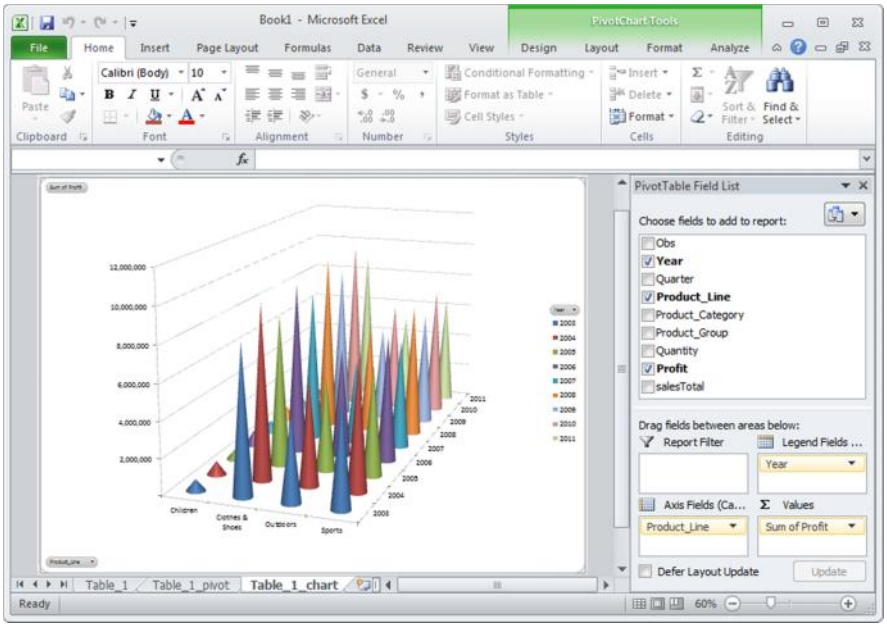


Display 25. New PivotTables in an Existing Workbook

Example 11 – Creating PivotCharts

It is almost inevitable that when data are added to Excel, charts and graphs are soon to be built. In this example we will use SAS to create our first PivotChart. To create PivotCharts we use the option ***pivotcharts="yes"*** ; this tells Excel to build a chart based on the PivotChart it creates. We will also need to tell Excel the type of chart to produce with the ***chart_type*** option. In this example we are creating a conecol chart. See Display 26 for the results of the program.

```
ods tagsets.tableeditor
   file="&resultsHome\example11.html"
options(
        button_text = "Excel"
        pivotrow="product_line"
        pivotcol="year"
        pivotdata="profit"
        pivotdata_fmt="#,###"
        pivotcharts="yes"
        chart_type="conecol"
);
Title1 "Example 11 - Creating a Pivot Chart";
proc print data=data.sales;
run;

ods tagsets.tableeditor close;
```

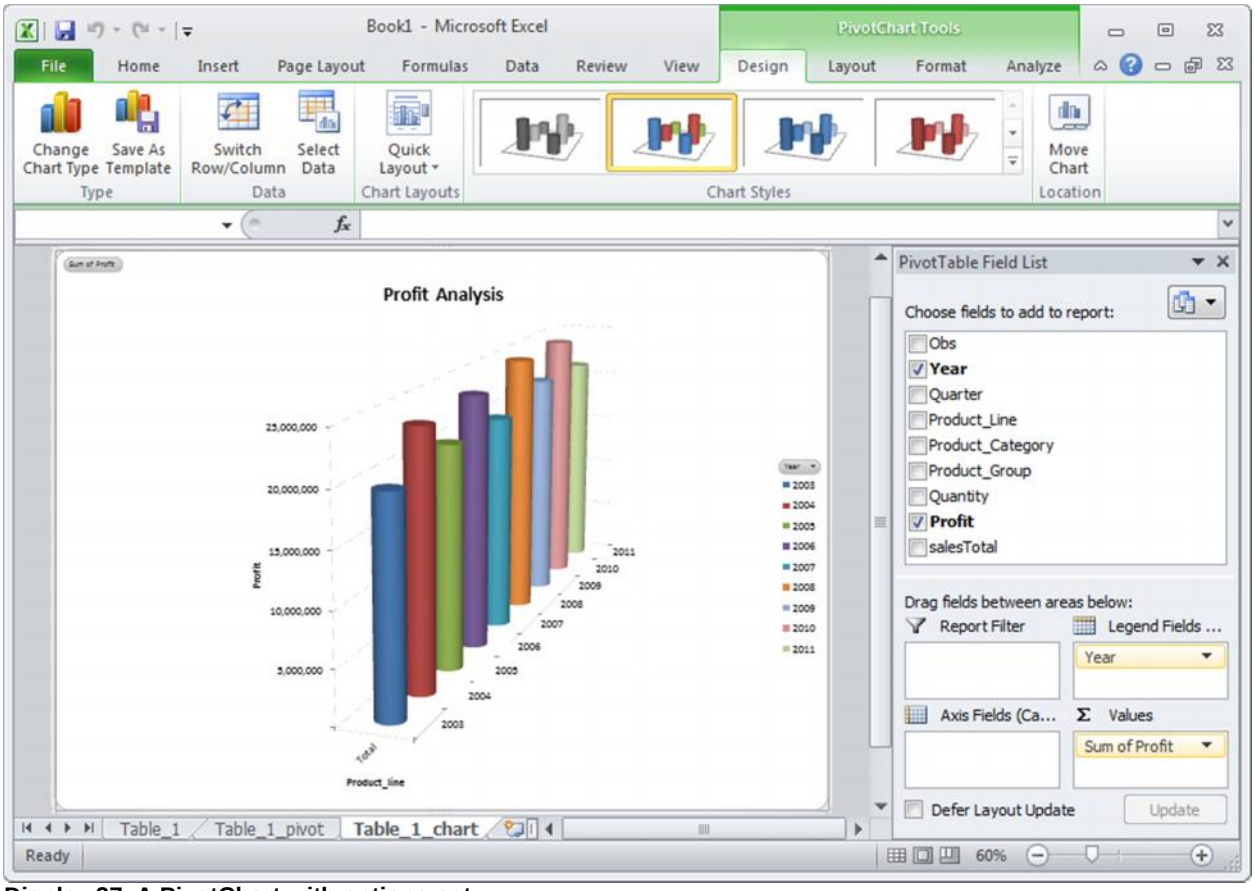


Display 26. A PivotChart

Example 12 – Enhancing PivotCharts

There are numerous chart options within the tagset; see Appendix 1 for the full list of options. In this example we will change the chart title (chart_title), the axis titles (xhart_yaxes_title, chart_xaxes_title) and the x-axis orientation (chart_xaxes_orientation). The results are in Display 27.

```
ods tagsets.tableeditor
   file="&resultsHome\example12.html"
options(
        button_text = "Excel"
        pivotcol="year"
        pivotdata="profit"
        pivotdata_fmt="#,###"
        pivotcharts="yes"
        chart_type="cylindercol"
        chart_title="Profit Analysis"
        chart_yaxes_title="Profit"
        chart_xaxes_title="Product_line"
        chart_xaxes_orientation="45"
       );
Title1 "Example 12 - Changing Pivot Chart Options";
proc print data=data.sales;
run;

ods tagsets.tableeditor close;
```



Display 27. A PivotChart with options set

There are many more chart options available. You are encouraged to examine the options in Appendix 1 and work through different combinations to see the variety of charts and chart effects that can be produced. Also keep in mind that examining and changing chart options through the Excel interface is relatively easy; once you produce basic PivotCharts, your Excel users can create the charts that best fit their wishes.

CONCLUSION

This paper has attempted to show how easy it is to create Excel PivotTables and PivotCharts from SAS. We only touched on a few of the options available yet saw how we can create a wide variety of PivotTables and PivotCharts. Since the purpose of a PivotTable is to allow analysts and managers to see and interactively change interactions of the data variables you will probably find you only need to provide basic PivotTables and the users will quickly learn to exploit them on their own.

It also became clear that it can be considerably easier to generate – and re-generate – PivotTables and PivotCharts from within SAS than using the Excel interface. Although the drag-and-drop interface is easy to use, creating SAS programs to pass in a few parameters will usually be faster, and certainly less error prone.

Now, when your boss says “Can I have that in Excel?” you can still shudder, but you can also quickly and easily provide the results.

REFERENCES

Parker, Chevell. 2008. “Creating That Perfect Data Grid Using the SAS® Output Delivery System.” *Proceedings of the SAS Global Forum 2008 Conference*. Cary, NC: SAS Institute Inc. www2.sas.com/proceedings/forum2008/258-2008.pdf.

Parker, Chevell. 2010. “Using SAS® Output Delivery System (ODS) Markup to Generate Custom PivotTable and PivotChart Reports.”

Proceedings of the SAS Global Forum 2008 Conference. Cary, NC: SAS Institute Inc. www2.sas.com/proceedings/forum2008/258-2008.pdf.

ACKNOWLEDGMENTS

I would like to acknowledge not only the good work of Chevell Parker, but also his willingness and ability to share his knowledge – PWE

The authors would like to thank the two sections chairs for Hands-on Workshops 2012 , Maribeth Johnson and Ann Stephan for inviting us to present; we know there are many quality papers from which to choose and we thank them for choosing our paper. It is always an honour and a privilege to present at SAS Global Forum.

CONTACT INFORMATION

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

Peter Eberhardt
Fernwood Consulting Group Inc.
288 Laird Drive
Toronto, ON, Canada M4G 3X5
(416)429-5705
peter@fernwood.ca
www.fernwood.ca

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.

APPENDIX 1 – TABLEEDITOR OPTIONS

The following tables have the options that are available for the TableEditor tagset. Tagset options are entered in name/value pairs. For example:

```
        pivotrow    = "product_line"
```

The options are grouped by the nature of the option, that is Style options, Positioning options, Display options etc.

Style Options	
banner_color_even="color"	creates alternate row colors
banner_color_odd="color"	creates alternate row colors
fbanner_color_even="color"	creates alternate even foreground colors
fbanner_color_odd="color"	creates alternate odd foreground colors
col_color_even="color	creates alternate even background column colo
col_color_odd="color"	creates alternate odd background column colo
gridline_color="color	color internal rules
gridline="yes|no"	removes internal gridline
background_color="color"	controls background color
background_image="path"	adds background image
scrollbar_color="color"	modifies scrollbar color
image_path="path"	adds logo or image
image_just="left|right|center"	justifies logo or image
highlight_color="color"	creates mouse-over color; generated when there is no background color such as styles.mystyle
highlight_cols="column#color"	modifies background color of columns
fontfamily="font"	modifies fonts overall if none specified
header_bgcolor="color"	modifies header background color
header_fgcolor="color"	modifies header foreground color
header_size="size"	modifies header font size
header_font="font"	modifies header font name
rowheader_bgcolor="color"	modifies header background color
rowheader_fgcolor="color"	modifies header foreground color
rowheader_size="size"	modifies header font size
rowheader_font="font"	modifies header font

Style Options	
data_bgcolor="color"	modifies data background color
data_fgcolor="color"	modifies data foreground color
data_size="size"	modifies data font size
data_font="font"	modifies data font
title_bgcolor="color"	modifies title background color
title_style="slant|roman|italic"	modifies data foreground color
title_size="size"	modifies data font size
title_fgcolor="color"	modifies foreground of title

Dynamic Positioning	
pageheight="value"	specifies height before scroll bars
pagewidth="value"	specifies width before scrollbars; percentage value works best
frozen_headers="yes|no"	freeze column headers; single table option.
frozen_rowheaders="yes|no|column#"	freeze row headers; single table option used for wide and not long tables; do not use for tables over 700 observations, which would be excessively slow

Page Setup Options	
print_header="string"	generates header text; uses ActiveX and is supported by I.E. and Windows only
print_footer="string"	generates footer text; uses ActiveX and is supported by I.E. and Windows only
print_zoom="value"	scale printed output

Page Setup Options	
orientation="landscape"	modifies orientation; uses ActiveX and is supported by I.E. and Windows only (To use this option you have to go to the below file and download the activex control http://www.meadroid.com/scriptx/)
pagebreak="yes|no|number"	deletes, add page breaks; also specifies the number of tables per page
left_margin="value"	changes the left margin
right_margin="value"	changes the right margin
top_margin="value"	changes top margin
bottom_margin="value"	changes bottom margin;
fit2page="yes|no"	allows output to be scaled to fit the printed page; you can also use this in conjunction with the orientation and the margin properties to reduce scaling

Table of Contents Options	
open_image_path="path"	adds open image
closed_image_path="path"	adds closed image
leaf_image_path="path"	adds leaf image
toc_background="path"	adds toc background color
toc_print="yes|no"	adds print button to the toc; this prints the individual body file
toc_expand="yes|no"	expand toc

Display Options	
drag="yes|no"	allow items on page to be dragged; dragged items are not persisted when the page is saved
zoom="percentage"	scale items on the screen; applies to all tables
zoom_toogle="yes|no"	adds dynamic selection list to page

Display Options	
table_zoom="100%,200%,300%"	scales list of tables separated by commas
sort="yes|no"	sort data by clicking on headers
sort_arrow_color="color"	color of arrows
sort_image="image path"	image indicating sort ; sample images located in zip file
sort_underline="yes"	underline column header; alerts reader that headers are sortable
header_underline="yes|no"	underline column header; underlines regardless sort
exclude_summary="yes"	allows filters and sorts to exclude summary; allows sort and filters to exclude grand total in report
data_type="value1,value2"	overrides the default data type for columns; valid types are Number,Numberx(formatted numbers), String, Date, None
describe="yes|no"	add color to identify data type
design_mode="yes|no"	allows the HTML to be edited; edited output is not persisted when the page is saved
pagebreak_toggle="yes|no"	interactive control of page breaks
zoom_toggle="yes|no"	interactive scaling control
autofilter="yes|no"	allows data to be filtered
autofilter_width="value"	allows the width of the filter to be modified; by default the width is the same as the width of the cell
autofilter_endcol="value"	column number to end filter
autofilter_table="value"	table to filter; the default is ALL
filter_cols="1,2,3"	column to add filter info
style_switch="yes"	allows styles to be switched on the fly; requires that you read existing CSS files to add to the selection

Display Options	
hide_cols="yes|no"	allows removal of columns; double click on column headers to remove
reorder_cols="yes|no"	provides the ability to dynamically reorder columns
web_tabs="Label1,Label2"	allows tabs on web page which names output
web_tabs_bgcolor="blue"	specifies background color for tabs
web_tabs_fgcolor="red"	specifies foreground color for web_tabs
panelcols="number"	number of columns to panel tables and graphs
header_display="yes|no"	allows removal of the column headers
header_vertical="yes|no"	allows the headers to be displayed vertically
nowrap="yes|no"	prevents wrapped text in the browser or when exported to Excel
align_cols="left,right"	Allows columns to be aligned based on the position in the list

Exporting Data	
excel_sheet_prompt="yes|no"	prompt to name sheet
excel_save_prompt="yes|no"	prompt to save file
excel_save_dialog="yes|no"	provide dialog to save file
excel_save_file="path"	save file by providing path
excel_autofilter="yes|no|number"	generate autofilters
excel_frozen_headers="yes|no|number"	freeze headers
excel_orientation="yes|no"	page orientation
sheet_name="name"	specify a sheet name
sheet_name="first"	
excel_table_move="1"	table from the page to move to Excel; this can a single value or a list of values separated by commas (This was depracated with release v2.30)
excel_table_move="1,2,3"	
file_format="ext"	supply format of excel file ; default is xls which is native Excel format; other formats are txt, csv, doc, xml, slk, html

Exporting Data	
auto_format="format name"	specifies Excel's format
auto_format_select="yes|no"	allows Excel's format to be chosen dynamically+B206
excel_macro="'file!macro'"	specify VBA macro to execute
excel_scale="number"	scale output printed output
excel_zoom="number"	specifies zoom for the worksheet
excel_default_width="number"	default width of each cell;
	helps prevent wrapping
excel_default_height="number"	default height of each cell;
	helps prevent wrapping
query_range="value"	location to begin writing data; query_ange="A11"
query_target="file-path"	file which is updated
query_file="file-path"	file which will provide the updating to
update_target="xls-file"	file to update; can be used with the sheet_name= and the excel_table_move option to update workbooks
update_range="value"	location to write the file update_range="11,1"
update_sheet="sheet name"	specifies the sheet to update
open_excel="yes|no"	determines if Excel is visible or not
quit="yes|no"	quits Excel and frees memory
pivotrow="name(s)|number(s)"	column names or numbers for pivot rows; to select more than one, separate list items with commas
pivotcol="names(s)|number(s)"	column names or number for pivot columns; to select more than one, separate list items with commas
pivotpage='names(s)|number(s)	column names or numbers for pivot page; to select more than one, separate list items with commas

Exporting Data	
pivotdata="name(s)|number(s)	column names or numbers for data; to select more than one, separate list items with commas
Pivotrow_fmt="@"	formats the row fields in the pivot table
Pivotcol_fmt="@"	formats the column fields in the pivottable
Pivotdata_fmt="#,###,$###"	formats one or more values specified inthe data area
Pivotdata_stats="max,min"	provides summary functions for one or more values specifed in the data area default is sum statistic; other summary functions are average, count, countnums, max,min, product,stddev,stddev,sum,var, varp
Pivotcalc="row"	performs calculation on the statistics such as percentage of row, column and total; valid values are row, column, total, percentOf, index, and runningTotal
Pivot_format="light1"	provides excel style for the pivot table;
Pivot_series="yes"	specifies that one or more pivot tables are created from the same worksheet
pivotcharts="yes"	indicates that you want to generate a pivotchart; used in conjunction with the chart_type= option
ptsource_range="range"	range of data to add to pivot table
ptdest_range="range"	range to write the pivot tables; if omitted, pivot table writes to a new worksheet
chart_type="type"	Excel's chart type to be used
chart_source="d:d"	data range to use for the chart
chart_title="string"	string sprecified for chart title
chart_title_size="size"	provides font size to the chart title
chart_title_color="color"	provides color for the chart title
chart_xaxes_title="string"	specifies a title for the X axes
chart_xaxes_size="size"	specifies size for the X axes title

Exporting Data	
chart_xaxes_orientation="45"	modifies the orientation of the axis
chart_yaxes_title="string"	specifies a title for the Y Axes
chart_yaxes_numberformat	modifies the format of the axis
Chart_yaxes_title="string"	specifies size for the Y axes title
chart_yaxes_orientation="45"	modifies the orientation of the axis
chart_yaxes_maxscale="4000"	modifies the axis scale
chart_yaxes_minscale="1000"	modifies the axis scale;
chart_area_color="color"	specifies color for chart area
chart_plotarea_color="color"	specifies color for the plot area
chart_datalabels="yes|no|value"	adds data labels to the chart; valid values are value, percent, labelandpercent, showbubblesizes
Chart_location="same_sheet|new_sheet| sheet_name"	specifies location to place charts; places charts on a new sheet by default
chart_location="0,100,300,400"	specifies top,left,height and width forembedded charts
chart_style="light1"	specifies the style for charts; values are light1-light13,medium1-medium13,dar1-dark13
chart_legend="bottom"	modifies the location of the legend
auto_excel="yes|no"	starts export to Excel after the page has been loaded
embedded_title="yes|no"	adds titles within worksheets for a single table
embedded_tables="yes|yes"	adds multiple tables within a worksheet
number_format="@|#,##"	applies excel formats to each column separated by a "|".

Informative and window Options	
include="file"	allows other files to be included to the page such as HTML, RTF, PDF
window_title="string"	name the window; this title is also used whensaved as favorite

Informative and window Options	
load_msg="yes|no"	generates a message while page is loading
load_image="image"	generates message and image while pageloading; see zip file for animated image which can be used
alert_text="string"	specifies string to display when the page is loaded
window_status="string"	text to display in the task bar
window_size="500,500|max"	allows window size to be specified; can specify coordinates or max to maximize window
fit2page_msg="yes|no"	adds information on the percentage output is scaled
caption_text="caption1,caption2"	text added to the caption of the table
caption_just="left|right|center"	justifies the caption of the table
caption_backround="color"	specifies background color for the caption
caption_color="color"	specifies foreground color for the caption
caption_style="normal|italic|oblique"	modifies style of the caption
caption_image="path"	provides image for the caption
doc="help"	display valid options and description
button_text="string"	replace text for export button
button_fgcolor="color"	foreground color of export button text
button_bgcolor="color"	background color of export button text
button_size="12pt"	size of export button text
powerpoint_master="primary#secondary"	specifies text to display on the masterslide; primary and sub title are separate by "#"
powerpoint_slides="a.html,b.html"	specifies HTML files to provide as individual slides
powerpoint_template="path"	supplies a PowerPoint to use for the formatting
powerpoint_saveas="path"	saves PowerPoint presentation to a slide
powerpoint_runs="yes|no"	runs PowerPoint presentation
auto_powerpoint="yes|no"	starts export to PowerPoint after the page has been loaded

APPENDIX 2 – THE AUTOEXEC TO CREATE THE MACRO VARIABLES

```
/* autoexec.sas                  */
/* the autoexec to start the workshop */
%let home         = C:\HoW\Eberhardt-Kong_13173;
%let solutionHome = &home.\solutions;
%let exerciseHome = &home.\exercises;
%let dataHome     = &home.\data;
%let resultsHome  = &home.\results;
%let javaHome     = C:\\HoW\\Eberhardt-Kong_13173\;
%let javaDataHome  = &javaHome.\data\;
%let javaResultsHome  = &javaHome.\results\;

/* assign the libraries */
libname data "&dataHome";
options fullstimer source source2;
```