

Paper 71-26

## Data Analysing in SAS, Graphs Building in SigmaPlot, Outputs in PowerPoint

Sherry Zhang, American BCS, Mountainside, NJ

### Abstract

SAS provides you a lot of nice features to do statistic analysis. Sigmaplot gives you abilities to build high quality statistic graphs. But your presentation has to be done in Powerpoint. Wouldn't it be more helpful to integrate all those three parts into one handy macro. Then, you could re-use your mature codes on the desktop by making simple modifications to meet a new requirement and the particular task at hand.

This paper illustrates an easy way to transfer a SAS output to Excel, through Sigmaplot building graphs, getting output in Powerpoint in two easy steps. Even more advance, they are integrated in SAS.

### Introduction

Converting SAS graphs to PowerPoint is always a big issue and time consuming job. SAS is wonderful software to do statistic analysis but not in reporting graphics, nor on electronic presentation. SigmaPlot is a state-of-the-art technical graphing program designed for Windows and Microsoft Office. It is specifically designed to aid in documenting and publishing research, specializing in the graphical presentation of statistic results. PowerPoint as a tool make your presentation truly effective. As you can see each one has its strength and weakness. If you can't get all those nice features in one software, you can group them as a team to finish your task.

A common procedure is SAS produce data output and use SigmaPlot to build graphs and then copy to PowerPoint, in the meantime you have to manually adjust font

and color, do those tedious work. The new procedure is presented here to put all three software in one.

### How to do

There is a way to export SAS output as Excel data worksheet. The macro code is simple as below:

```
*****
**GOAL : Output sas data to
**      Excel worksheet
**PARAMETER: &SASDD, &EXCEL
**USAGE:
** %let sasdata=work.all
** %let excelout=
**      'c:\graph\forarm.xls'
** %include
**      't:\utilities\sas_excel.sas'
** %sasexcel
**
** by Sherry Zhang 8/00 **
*****;
%macro sasexcel;

    PROC DBLOAD DBMS=EXCEL
    DATA=&sasdata;
    PATH=&excelout;
    PUTNAMES YES;
    LIMIT=0;
    LOAD;
    RUN;

%mend;
```

Then open SigmaPlot to record a templet macro program. At the end of program, call PowerPoint build function. This function covert your graph to PowerPoint screen. Then turn off the macro recording. Go back to modify the VB code. It will be useful to change the code to interactive message box

to be able to enter the input data path after submit the program. If you want, you also can modify the output macro to put graph in one screen a file or multiple screens a file. If you want to print the PowerPoint file, you can record print function in your templet. Then save the macro as a VB file. Give it a shortcut key. You always can go back to do modification to that file to make it more dynamic. The Appendix is my example. It is Visual Basic code but it is recorded by software. I modified to be able to fit my task.

The code can be reused by press the short-cut key or call the macro name. I use it as template by modifying a little bit to fit another task. Example codes are shown in Appendix.

#### Run VB code in SAS

Use Dynamic Data Exchange (DDE) to link SAS to Window software SigmaPlot. Then using a DATA\_NULL to put FILENAME. There are two ways to link VB code to SAS. If the VB code is executable, the following code can be used in SAS program:

```
Options noxsync missing=0;
X 'the file path/filename';
Run;
```

If the macro code run in the particular system, the following code can be used in SAS program, for example:

```
Options noxsync missing=0;
X 'c:\program files\Excel\Excel.exe
mybook.xls';
X=sleep(15);
Run;
```

```
filename ex1 dde 'excel|system';
Data _Null_;
file ex1;
PUT '[RUN("mybook.xls!macroname")]';
run;
```

#### Integrate into SAS

Using OLE (Object Linking and Embedding) technique, the server information is stored in the linked object, but the data are not. Thus you are able to edit from either side of the

link. If you link SigmaPlot to a SAS/AF application, you can either edit the graph from within the SAS/AF application (by double-clicking on the picture) or by invoking SigmaPlot and opening the correct file. In either case, the changes you make are immediately reflected in the other side of the link by default. If someone has already created a SAS/AF application that uses OLE, you do not even need SAS/AF software installed on your machine. You can simply run the application.

As with choosing between OLE and DDE, choosing between object linking and object embedding depends on your preferences and application needs. If you need to be able to edit and manipulate graphs both from the server application and from the SAS System, you should use object linking. If you do not need to edit the data outside of the SAS System, you can use object embedding. A third choice is using static graphs, which are simply pictures and do not allow you to edit or manipulate the graph at all.

#### Conclusion

To create a SAS macro to export the analysis output is the first step; To create a SigmaPlot templet is the second step. Automated procedures one system is the major contribution of this paper. Taking different software strength is the idea to maximize utility of computer software. I group all three functional software as one project team. It expedite my work significantly.

#### Trademarks

SAS is a registered trademark of SAS Institute, Inc. in the USA and other countries. SigmaPlot is a registered trademark of SPSS, Inc. in the USA and other countries. PowerPoiit is a registered trademark of Microsoft, Inc. in the USA and other countries.

#### Contact Information

Sherry Zhang, American BCS,  
Mountainside, NJ

Zhangqing\_sherry@hotmail.com

### Appendix

Option Explicit

Function FlagOn(flag As Long)

FlagOn = flag Or FLAG\_SET\_BIT ' Use to set option flag bits on, leaving others unchanged

End Function

Function FlagOff(flag As Long)

FlagOff = flag Or FLAG\_CLEAR\_BIT ' Use to set option flag bits off, leaving others unchanged

End Function

Sub Main

This macro allows the user to record a macro that produces

a specific graph, insert the macro code below and then

'copy your recorded macro below here

'it must include adding a notebook and creating a graph

Notebooks.Add

ActiveDocument.CurrentItem.Import("C:\My Documents\VB\forearm.xls", 0, 0, 0, 0, 0, 0, ".XLS")

ActiveDocument.NotebookItems.Add(CT\_GRAPHICPAGE)

Dim ColumnsPerPlot()

ReDim ColumnsPerPlot(2, 1)

ColumnsPerPlot(0, 0) = 0

ColumnsPerPlot(1, 0) = 0

ColumnsPerPlot(2, 0) = 0

ColumnsPerPlot(0, 1) = 1

ColumnsPerPlot(1, 1) = 0

ColumnsPerPlot(2, 1) = 0

Dim PlotColumnCountArray()

ReDim PlotColumnCountArray(0)

PlotColumnCountArray(0) = 2

ActiveDocument.CurrentPageItem.CreateWizardGraph("Line Plot", "Simple Spline Curve", "XY Pair", ColumnsPerPlot, PlotColumnCountArray, "Worksheet Columns", "Standard Deviation", "Degrees", 0.000000, 360.000000)

'ActiveDocument.CurrentPageItem.Select(False, -1875, 1417, -1875, 1417)

'ActiveDocument.CurrentPageItem.Select(False, -1875, 1417, -1875, 1417)

'ActiveDocument.CurrentPageItem.Select(False, -1854, 1417, -1854, 1417)

'ActiveDocument.CurrentPageItem.Select(False, -1854, 1417, -1854, 1417)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETPLOTATTR, SLA\_SELECTDIM, 1)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SAA\_SELECTLINE, 4)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SAA\_SELECTLINE, 2)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETPLOTATTR, SLA\_SELECTDIM, 2)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETGRAPHATTR, SGA\_PLANECOLORXYBACK, &Hff000000)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETPLOTATTR, SLA\_SELECTDIM, 1)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SAA\_SELECTLINE, 4)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SEA\_LINETYPE, 1)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SEA\_THICKNESS, 10)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SAA\_OPTIONS, 12583688)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETAXISATTR, SAA\_SELECTLINE, 2)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETPLOTATTR, SLA\_SELECTDIM, 2)

ActiveDocument.CurrentPageItem.SetCurrentObjectAttribute(GPM\_SETGRAPHATTR, SGA\_FLAGS, 14340)

'end of recorded macro

'Copy Graph to PowerPoint

ActiveDocument.CurrentPageItem.SelectAll

'select everything on page

Clipboard "empty"

ActiveDocument.CurrentPageItem.Copy

'start PowerPoint

Dim PPApp As Object

Set

PPApp=CreateObject("PowerPoint.Application")

PPApp.Visible=True

'set up PowerPoint window and slide layout

PPApp.Presentations.Add WithWindow:=1

```
Dim LayoutNumber As Integer
LayoutNumber = 12      'blank slide
layout
PPApp.ActiveWindow.View.GotoSlide
Index:=PPApp.ActivePresentation.Slides.Add(In
dex:=PPApp.ActivePresentation.Slides.Count+1,
Layout:=LayoutNumber).SlideIndex

'add SigmaPlot graph to PowerPoint presentation
PPApp.ActiveWindow.ViewType = 1
'ppViewSlide=1, change from Normal View to
Slide View
PPApp.ActivePresentation.Slides(PPApp.Active
Presentation.Slides.Count).Shapes.Paste
PPApp.Activate      'may not be
required
PPApp.ActiveWindow.WindowState = 3
3=ppWindowMaximized, maximizes Slide View
window

End Sub
```