Paper 10180-2016

Creating and Sharing SAS® ODS Graphics with a Code Playground Based on Microsoft Office

Ted Conway, Chicago, IL; Zeke Torres, Chicago, IL

ABSTRACT

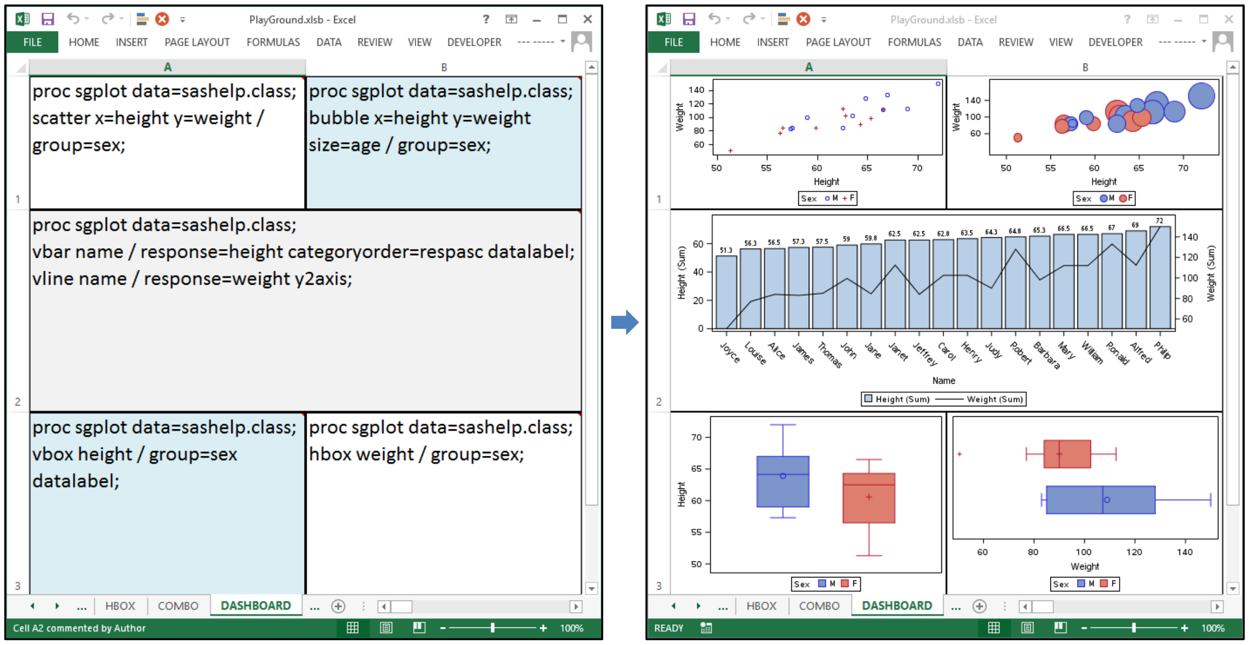
You've heard that SAS® Output Delivery System (ODS) Graphics provides a powerful and detailed syntax for creating custom graphs, but for whatever reason you still haven't added them to your bag of SAS® tricks. Let's change that! We will also present a code playground based on Microsoft Office that will enable you to quickly try out a variety of prepared SAS ODS Graphics examples, tweak the code, and see the results—all directly from Microsoft Excel. More experienced users will also find the code playground (which is similar in spirit to Google Code Playground or JSFiddle) useful for compiling SAS ODS Graphics code snippets for themselves and for sharing with colleagues, as well as for creating dashboards hosted by Microsoft Excel or Microsoft PowerPoint that contain precisely sized and placed SAS graphics.

INTRODUCTION

With its robust and easy-to use framework, SAS ODS Graphics is just what the doctor ordered if you're in need of flexibility, consistency, scalability, preciseness, and reproducibility in a data visualization solution.

In this paper, we present a "Code Playground" based on Microsoft Office to make it easier and more enticing to get started with SAS ODS Graphics by allowing you to:

- Review and share a variety of SAS code examples organized in Excel or PowerPoint documents
- Run SAS ODS Graphics code and view the results directly from within Office
- Create good-looking Office "dashboards" in a jiffy with automatically-sized and positioned SAS ODS Graphics-generated images



```
proc sgplot data=sashelp.class;
scatter x=height y=weight /
group=sex;
```

```
proc sgplot data=sashelp.class;
bubble x=height y=weight
size=age / group=sex;
```

```
proc sgplot data=sashelp.class;
vbar name / response=height categoryorder=respasc datalabel;
vline name / response=weight y2axis;
```

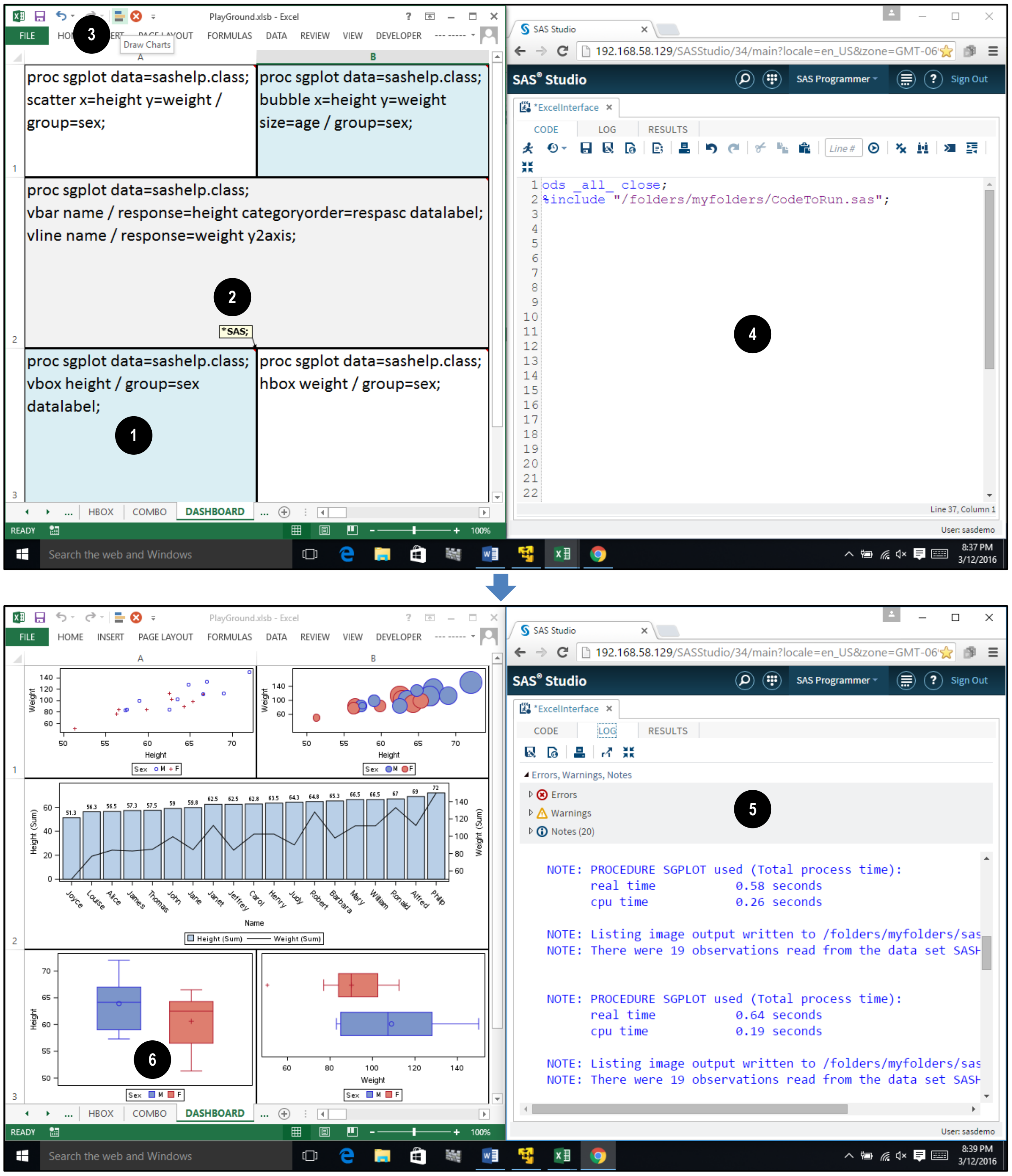
```
proc sgplot data=sashelp.class;
vbox height / group=sex
datalabel;
```

```
proc sgplot data=sashelp.class;
hbox weight / group=sex;
```

Figure 1: Go from SAS code in Excel cells to cell-sized images with the click of a button!

And, as the next page shows, creating SAS ODS Graphics images from Office is easy as 1-2-3(-4-5-6).

AN EXCEL-BASED CODE PLAYGROUND FOR CREATING SAS ODS GRAPHICS IMAGES



```
proc sgplot data=sashelp.class;
scatter x=height y=weight /
group=sex;
```

```
proc sgplot data=sashelp.class;
bubble x=height y=weight
size=age / group=sex;
```

```
proc sgplot data=sashelp.class;
vbar name / response=height categoryorder=respasc datalabel;
vline name / response=weight y2axis;
```

*SAS;

```
proc sgplot data=sashelp.class;
vbox height / group=sex
datalabel;
```

```
proc sgplot data=sashelp.class;
hbox weight / group=sex;
```

```
ods _all_ close;
%include "/folders/myfolders/CodeToRun.sas";
```

```
NOTE: PROCEDURE SGPLOT used (Total process time):
      real time           0.58 seconds
      cpu time            0.26 seconds

NOTE: Listing image output written to /folders/myfolders/sas
NOTE: There were 19 observations read from the data set SASH


NOTE: PROCEDURE SGPLOT used (Total process time):
      real time           0.64 seconds
      cpu time            0.19 seconds

NOTE: Listing image output written to /folders/myfolders/sas
NOTE: There were 19 observations read from the data set SASH
```

Figure 2: Excel-based Code Playground in split-screen with SAS Studio

First, you add SAS code that generates SAS ODS Graphics images ❶ into Excel cells tagged with comments that start with "*SAS;" ❷. When you click the Quick Access Toolbar charts icon ❸, an Excel VBA macro will run and write the SAS code to a file together with statements that set image dimensions (cell height/width), invoke SAS Studio (which %include's the code) ❹, wait for SAS to complete ❺, and insert the SAS-created images over the cells that contained ODS Graphics code. ❻

SO, WHERE CAN I GET MY HANDS ON THIS?

Shortly after SGF 2016, look for a link to an Excel workbook-based Code Playground containing the SAS ODS Graphics code examples presented at the conference on the Windy City SAS Users Group site at http://www.wcsug.com/presentations/tedconway/presentation.html. A link to a PowerPoint-based Code Playground that uses Office shapes instead of Excel cells for SAS code "containers" will also be provided.

If you just can't wait and want to roll your own Excel-based Code Playground, the *Appendix* contains the full Excel VBA macro (< 60 lines of code) and 2-line SAS Studio code snippet you'll need. You DIY'ers will also need to add Quick Access Toolbar buttons for the *DrawCharts* and *DeleteCharts* macros.

TROUBLESHOOTING TIPS

Ah, glitches – ones you're likely to run into will occur when:

- Macros aren't enabled in Office, or a cell's in insert/edit mode when *Draw Charts* is clicked
- SAS Studio isn't open, is minimized, or its browser tab doesn't have "focus"
- SAS syntax errors

Not to worry – once you've correct the situation, just try again. Also, the VBA macro will attempt to give you an opportunity to cancel out of an unresponsive SAS request after 30 seconds.

CONCLUSION

An Excel workbook provides a convenient vehicle for organizing a variety of SAS code examples for those learning – or teaching – ODS Graphics. When paired with SAS software (the free SAS University Edition here, but that's not a requirement), an Excel workbook can also be turned into a "Code Playground" from which the code examples can also be launched and output images viewed. And because images are automatically sized and positioned, both new and experienced ODS Graphics users can quickly knock out MS-Office "dashboards."

Try it, you'll like it!

ACKNOWLEDGMENTS AND RECOMMENDED READING

SAS. *Base SAS® Output Delivery System (ODS) Graphics Suite*
http://support.sas.com/documentation/prod-p/grstat/index.html

SAS. *Graphically Speaking: Data Visualization with a focus on SAS ODS Graphics*
http://blogs.sas.com/content/graphicallyspeaking/

SAS. *SAS® University Edition*
http://www.sas.com/en_us/software/university-edition.html

CONTACT INFORMATION

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

Ted Conway
ted.j.conway@gmail.com

Zeke Torres
MPA Healthcare Solutions / Chicago Area SUG Leader
ztorres@consultmpa.com - zt@wcsug.com

APPENDIX

Below is the complete VBA and SAS code for an Excel workbook-based SAS ODS Graphics “Code Playground.” If necessary, change the PCdir and VMdir directory paths in the VBA macro to match your system.

```
#If VBA7 Then
    Public Declare PtrSafe Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As LongPtr) ' Sleep function (msec), 64 bit
#Else
    Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)            ' Sleep function (msec), 32 bit
#End If

'==> Create Excel charts by calling SAS with cell width/height & code from each tagged comment/cell

Sub DrawCharts()

DeleteCharts                                                      ' Get rid of any existing charts
                                                                  ' PC and VMware folder names (same physically)
PCdir = "C:\Users\" & Environ("Username") & "\Documents\SASUniversityEdition\myfolders\"
VMdir = "/folders/myfolders/"

Open PCdir & "CodeToRun.sas" For Output As #1                     ' Create SAS program
i = 0                                                             ' Image counter
For Each cmt In ActiveSheet.Comments                              ' Get SAS code from comment and/or cell
  If UCase(Left(cmt.Text, 5)) = "*SAS;" Then                      ' SAS code tagged cell?
    i = i + 1
    Print #1, "%let XLwidth=" & cmt.Parent.MergeArea.Width & ";" & _
              "%let XLheight=" & cmt.Parent.MergeArea.Height & "; " & _
              "ods listing gpath='" & VMdir & "'; " & _
              "ods graphics on / reset=index  imagename='saschart" & i & "' width=&XLwidth.pt height=&XLheight.pt;" & _
              cmt.Parent.Value & cmt.Text & "; run;"
  End If
Next cmt
Print #1, "data _null_; file '" & VMdir & "CodeToRunDone.txt'; put 'Done'; run;"
Close #1

On Error Resume Next                                              ' Delete trigger file that indicates SAS is done
Kill PCdir & "CodeToRunDone.txt"
On Error GoTo 0                                                   ' Activate SAS and run code

AppActivate "SAS Studio", True                                    ' Activate SAS and submit code via SendKeys
function
SendKeys "{F3}", True                                             ' SAS %INCLUDE’s /folders/myfolders/CodeToRun.sas

i = 0                                                             ' Wait for SAS to finish (trigger file created)
While (Dir$(PCdir & "CodeToRunDone.txt") = "")
  Sleep (1000)                                                    ' Sleep for 1 second
  i = i + 1
  If i > 30 Then
    w = MsgBox("Still unfinished, wait?", vbYesNo)                ' After 30 seconds, ask to continue or bail
    If w = vbNo Then Exit Sub
    i = 0
  End If
Wend

i = 0
For Each cmt In ActiveSheet.Comments                              ' Retrieve and position SAS-generated images
  If UCase(Left(cmt.Text, 5)) = "*SAS;" Then                      ' SAS code tagged cell?
    i = i + 1
    ActiveSheet.Shapes.AddPicture PCdir & "saschart" & i & ".png", False, True, _
                       cmt.Parent.Left, cmt.Parent.Top, cmt.Parent.MergeArea.Width, cmt.Parent.MergeArea.Height
  End If
Next cmt
AppActivate "Excel"

End Sub

'==> Delete any existing charts

Sub DeleteCharts()
c = ActiveSheet.Shapes.Count
For i = c To 1 Step -1
  If ActiveSheet.Shapes(i).Type = 13 Then ActiveSheet.Shapes(i).Delete
Next
End Sub
```

Figure 3: Playground.xlsb Excel VBA Macro Code

```
ods _all_ close;
%include "/folders/myfolders/CodeToRun.sas";
```

Figure 4: ExcelInterface.sas SAS Studio Snippet (%include's SAS code written by Excel VBA macro)