

## Advanced Figures using SAS Graph Template Language (GTL)

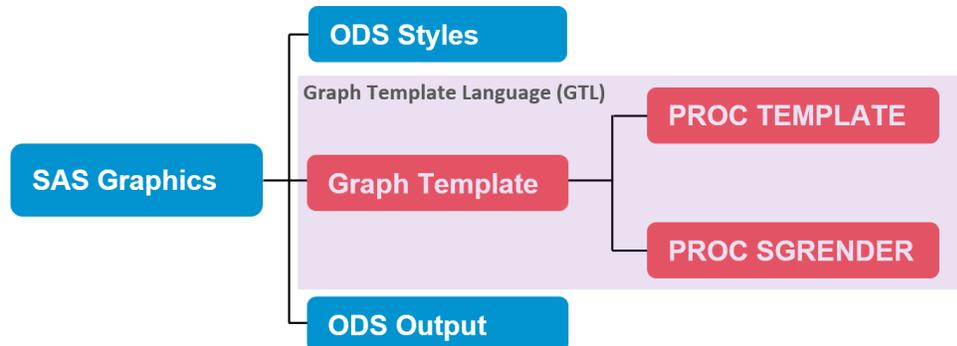
Wei (Tony) Zhang, Pfizer (China) Research and Development Co.,Ltd. China

### ABSTRACT

High quality graphs are essential for analysis of data in the clinical trials. The SAS Graph Template Language (GTL) was introduced into SAS Version 9.2 and improved in SAS Version 9.4. It is a different way of creating graphs in SAS compared with traditional SAS/Graph procedures, which starts a new life into graphs within SAS. This paper demonstrates how SAS Graph Template Language can be effectively and easily used to create plots like waterfall plot, swimmer plot, forest plot using Graph Template Language and the ODS Graphics procedure; new functions in SAS 9.4 GTL; ODS template modification, and other tips to use in GTL.

### INTRODUCTION

With the development of technology, many languages such as SAS, R, Python can generate figures for data analysis and visualization. Graph Template Language (GTL) is a useful tool for programmers to improve graphics generation using SAS. The following chart shows the current SAS ODS and the connection with GTL. Learning ODS and GTL together is an efficient way as they seem brothers in SAS. This paper and the related workshop will give an easy hands-on practice for users to implement these elements for advanced figures generation. As there are a lot of elements in SAS ODS and GTL, audiences will benefit from this practice for their knowledge knowing graphics generation improvement during the SAS maintenance release by versions and gain the learning skills for SAS GTL.



### WATERFALL PLOT

A waterfall plot is graphic illustrations of data that can vary from audio frequencies to clinical trial subject information and results. In oncology, for example, a waterfall plot may be used to present each individual patient's response. Vertical bars are drawn for each patient and the vertical (y) axis may be used to measure best percent change from baseline.

In GTL, if you are using SAS 9.2, the **BARCHART** is the key plot statement to generate the waterfall plot. If you are using SAS 9.4, in addition to **BARCHART**, the **WATERFALLCHART** can also be used to generate this plot. A sample data was generated for the Hands-On Workshop (HOW) and the code below shows an example using **BARCHART**.

## PROC TEMPLATE for Waterfall Plot:

```
PROC TEMPLATE;
  Define statgraph waterfall;
    Begingraph; /*1*/
    Layout Overlay/ /*2*/
      xaxisopts=(label="Subjects"
        labelattrs=(weight=bold size=10pt)
        griddisplay=on display=(label)
        tickvalueattrs=(weight=bold size=10pt)
        type=discrete)
      yaxisopts=(label="Best Percentage Change from Baseline"
        labelattrs=(weight=bold size=10pt)
        griddisplay=on
        tickvalueattrs=(weight=bold size=10pt)
        linearopts=( thresholdmax=1 thresholdmin=1));
    Barchart x=subjid y=aval/ /*3*/
      group=trt01a
      name="barname"
      barwidth=0.7;
    Scatterplot x=subjid y=bestorp/ /*4*/
      MARKERCHARACTER=bestors;
    Referenceline y=-30/ /*5*/
      lineattrs=(thickness=1px pattern=34 color=cx000000 ) ;
    Referenceline y=20/
      lineattrs=(thickness=1px pattern=34 color=cx000000 ) ;
    Discretelegend "barname"/ /*6*/
      location=inside valign=top halign=right across=1
      valueattrs=(weight=bold size=8pt);
    Endlayout;
    Endgraph;
  END;
RUN;

PROC SGRENDER data=plot template=waterfall;
RUN;
```

1. The entire GTL code is inside the **Begingraph - Endgraph** code block.
2. **Layout Overlay - Endlayout** is the block to generate the waterfall plot.
3. **Barchart** is to draw the vertical bars for each patient.
4. **Scatterplot** is to display the best overall response label at each bar.
5. **Referenceline** creates the horizontal reference line
6. **Discretelegend** displays a legend inside the plot with the upper right location.

The above example code creates the Figure 1 below.

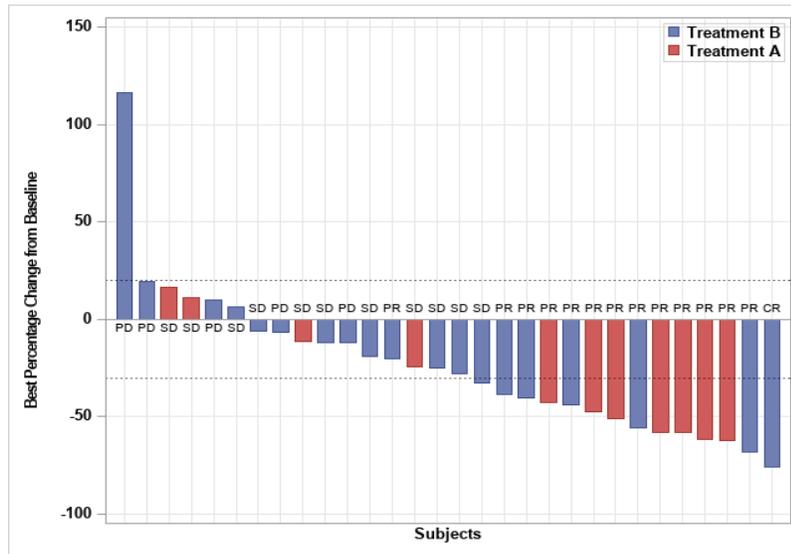


Figure 1: Waterfall Plot using BARCHART

- **First Adjustment – Group Color:**

From Figure 1 example, different colors by treatment are displayed in the plot and legend. This is the result of **GROUP** in the **Barchart** option. But how can we customize the color for each group? **PROC TEMPLATE** is used to define the individualized template such as color, pattern, font, etc. as the following example based on **Styles.Statistical** as parent.

PROC TEMPLATE for ODS Style:

```

PROC TEMPLATE;
  define style water;
  parent = Styles.Statistical;
  class GraphData1/          /* 1st Group */
    contrastcolor=cx0000ff /* Outline Color - Blue */
    color=cx0000ff;        /* Filled Color - Blue */
  class GraphData2/          /* 2nd Group */
    contrastcolor=cxff0000 /* Outline Color - Red */
    color=cxff0000;        /* Filled Color - Red */
  class GraphData3/          /* 3rd Group */
    contrastcolor=cx00ff00 /* Outline Color - Green */
    color=cx00ff00;        /* Filled Color - Green */
  class GraphData4/          /* 4th Group */
    contrastcolor=cxff00ff /* Outline Color - Pink */
    color=cxff00ff;        /* Filled Color - Pink */
  END;
RUN;

```

Then the figure will be created in PDF or RTF output with option **style=water** in ODS PDF or ODS RTF statement. As there are only two groups in this sample data, Treatment B will be blue as the definition in **GraphData1** (group value in the first bar from the left is Treatment B in Figure 1), then Treatment A will be red as the definition in **GraphData2** (group value in the third bar from the left is Treatment A). If other GraphData style applies for the group, an **INDEX** option can be added in the **Barchart** statement. Then Treatment A subjects' bars will be green if variable COLOR value is 3 for treatment A subjects in dataset and Treatment B subjects' bars will be pink if variable COLOR value is 4 for Treatment B subjects in

dataset. This **INDEX** option is very helpful if the same categorization such as treatment across all different figures needs to share the same color group.

```
Barchart x=subjid y=aval/
  group=trt01a
  INDEX=color
  name="barname"
  barwidth=0.7;
```

- **Second Adjustment – Order Display in Legend:**

However in Figure 1 legend, the treatment group is not the order as we expect. The reason is that the first bar from left is the patient for treatment B, so the first treatment value in the legend is not treatment A, even if we use GROUP option. In **SAS 9.4**, it is very easy to fix the order issue in the legend by using option **SORTORDER = ASCENDINGFORMATTED** in the Discretelegend statement.

```
Discretelegend "barname"/
  title="Treatment Group:"
  location=inside valign=top halign=right across=2
  titleattrs=(weight=bold size=10pt)
  valueattrs=(weight=bold size=10pt)
  SORTORDER=ASCENDINGFORMATTED;
```

- **Third Adjustment – Simple Annotation:**

Acronym in each bar (CR, PR, SD, PD) indicates the best overall response of each subject. If we want to annotate these acronym in the bottom left of this plot, a nest **Gridded Layout** can be easily added.

```
Layout Overlay/.....;
  Barchart .....;
  Scatterplot .....;
  Referenceline .....;
  Referenceline .....;
  Discretelegend .....;
```

```
Layout Gridded/
  rows=4 order=columnmajor border=false
  autoalign=(bottomleft);
  entry textattrs=(size=10pt) halign=left
  "CR = COMPLETE RESPONSE";
  entry textattrs=(size=10pt) halign=left
  "PR = PARTIAL RESPONSE";
  entry textattrs=(size=10pt) halign=left
  "SD = STABLE DISEASE";
  entry textattrs=(size=10pt) halign=left
  "PD = PROGRESSION DISEASE";
Endlayout;
```

```
Endlayout;
```

Each entry statement annotates the acronym in the bottom left of the plot line by line. Anyway, macro variables are suggested to do the repeat entry as the code below.

```
Layout Gridded/
  rows=&num order=columnmajor border=false
  autoalign=(bottomright bottomleft);
  &do i=1 %to &num;
  entry textattrs=(size=10) halign=left "&a&i = &v&i";
  &end;
Endlayout;
```

Then Figure 2 below is the updated one after above three adjustments.

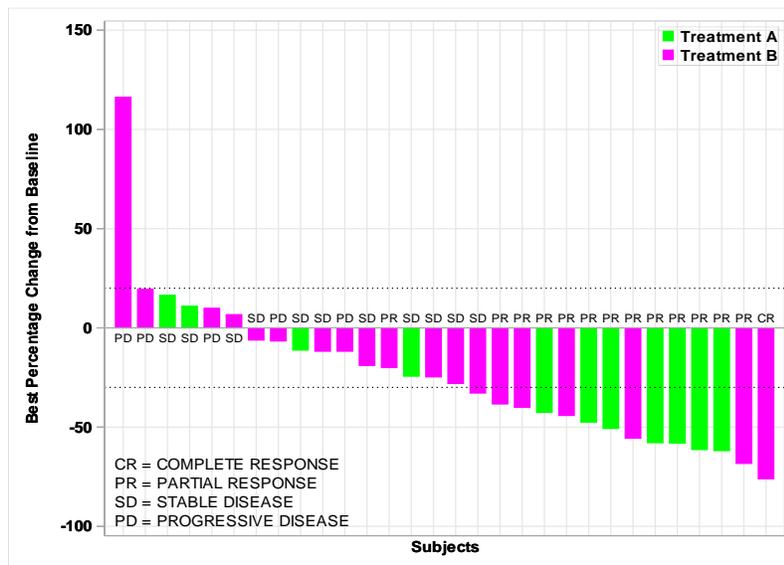


Figure 2: Updated Waterfall Plot using BARCHART

## SWIMMER PLOT

A swimmer plot is a graphical presentation involving horizontal bars that can be used to show multiple pieces of information. In **SAS 9.2**, **BARCHART** can be used to generate the swimmer plot as Figure 3 example. Some users also use **VECTORPLOT** to generate the swimmer plot in **SAS 9.2** if there are multiply durations for the same patient.

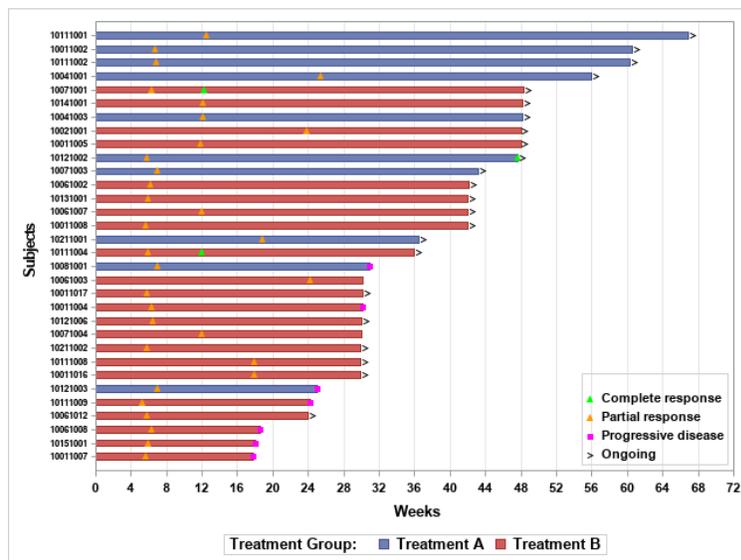


Figure 3: Swimmer Plot using BARCHART (SAS 9.2)

In **SAS 9.4**, plot statement **HIGHLOWPLOT** can be used and preferred to generate the swimmer plot or even mirror plot (prior-treatment duration and on-treatment duration on the same plot), which creates a display of floating vertical or horizontal lines or bars that connect the minimum and maximum response values for each value of a categorical variable. This new plot statement also makes GTL possible to generate the swimmer bar with different style (eg: arrow bar for ongoing subjects in Figure 4). In addition to the arrow bar, the label of duration of weeks for each subject is also displayed at the end of bar, which can be easily implemented using **HIGHLABEL** option in the **HIGHLOWPLOT** statement.

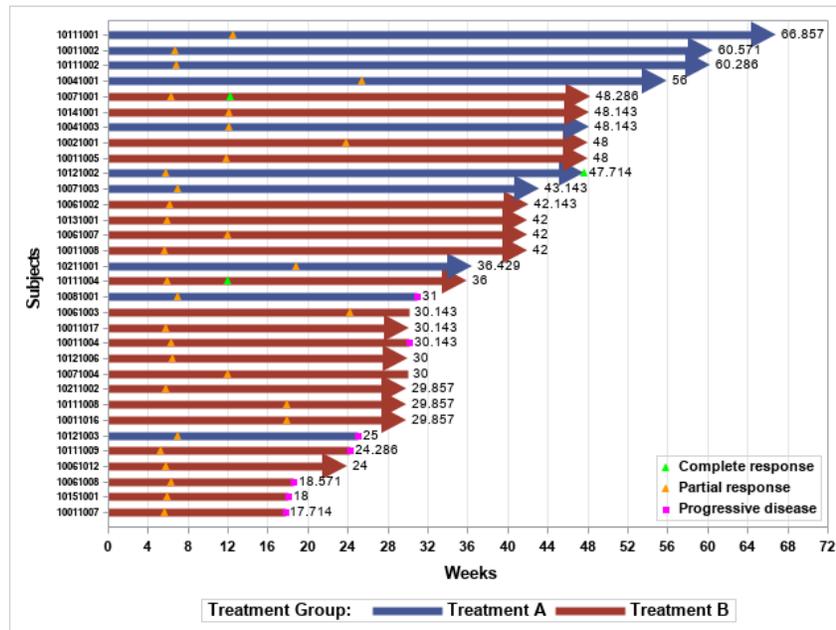


Figure 4: Swimmer Plot using HIGHLOWPLOT (SAS 9.4)

PROC TEMPLATE for Figure 4 using SAS 9.4:

```

PROC TEMPLATE;
  Define statgraph swimmer2;
  Begingraph;
  Layout Overlay/
    xaxisopts=(label="Weeks" labelattrs=(weight=bold size=10pt)
      linearopts=(viewmin=0 viewmax=72
        TICKVALUESEQUENCE=(start=0 end=72 increment=4))
      griddisplay=on
      offsetmin=0 offsetmax=0
      tickvalueattrs=(weight=bold size=8pt))
    yaxisopts=(label="Subjects" labelattrs=(weight=bold
      size=10pt)
      tickvalueattrs=(weight=bold size=6pt)
      type=discrete
      offsetmin=0.03 offsetmax=0.03
      display=all);
  /*1*/
  HIGHLOWPLOT y=subjid low=low high=duration/
    group=TRT01A index=trt01an
    TYPE=LINE LINEATTRS=(THICKNESS=5PT)
    HIGHCAP=CAP
    HIGHLABEL=DURATION LABELATTRS=(COLOR=CX000000)
    name="trt";

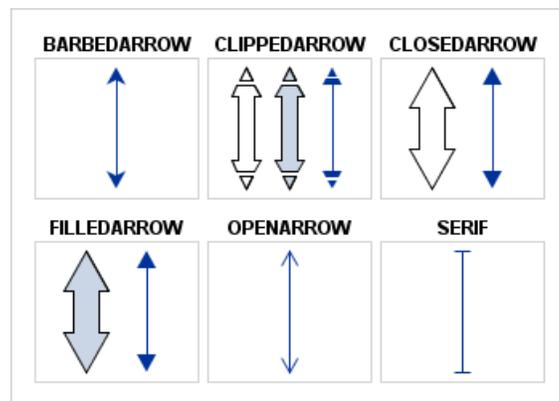
```

```

/*2*/
SCATTERPLOT x=ttcr y=subjid/MARKERATTRS=(SIZE=4pt color=cx00ff00
SYMBOL=trianglefilled) name="cr" LEGENDLABEL="Complete response";
SCATTERPLOT x=ttpr y=subjid/MARKERATTRS=(SIZE=4pt color=cxff9900
SYMBOL=trianglefilled) name="pr" LEGENDLABEL="Partial response";
SCATTERPLOT x=ttpd y=subjid/MARKERATTRS=(SIZE=4pt color=cxff00ff
SYMBOL=squarefilled) name="pd" LEGENDLABEL="Progressive disease";
SCATTERPLOT x=ttdeath y=subjid/MARKERATTRS=(SIZE=6pt
color=cx000000 SYMBOL=star) name="death" LEGENDLABEL="Death";
/*3*/
DISCRETELEGEND "cr" "pr" "pd" "death"/ location=inside
valign=bottom halight=right titleattrs=(weight=bold size=8pt)
valueattrs=(weight=bold size=8pt) across=1 ;
DISCRETELEGEND "trt"/title="Treatment Group:"
SORTORDER=ASCENDINGFORMATTED titleattrs=(weight=bold size=10pt)
valueattrs=(weight=bold size=10pt);
Endlayout;
Endgraph;
End;
Run;

```

1. **HIGHLOWPLOT** is the statement to draw the bar. Option **HIGHCAP=CAP** is used to display the arrow bar for ongoing patient if it is required. CAP is a variable in the dataset which contains the value of "FILLEDARROW" for ongoing patient. All the values of HIGHCAP is shown below for each arrow style.



**TYPE=LINE LINEATTRS=(THICKNESS=5PT)** is set to display the bar since the thickness is 5PT. The reason why TYPE=LINE, not TYPE=BAR is that there are many subjects in the plot. If the type is bar, there will be a risk that the plot is not able to display bar due to narrow horizontal space between each subject. If there are many subjects in the plot and bar is required for display, TYPE=LINE and LINEATTRS for the width of line setting is the approach to drawing the bar. **HIGHLABEL=DURATION LABELATTRS=(COLOR=CX000000)** is set to automatically display the duration of weeks at the end of each bar. DURATION is the variable containing the weeks value in the dataset. As group=TRT01A is set to categorize the treatment by color, the default color of label value is consistent with bar color. LABELATTRS=(COLOR=CX000000) purpose is to display label value as black at the end of all bars.

2. Four **SCATTERPLOT** statements are used to generate each assessment symbol inside the bar to present the responses of each patient in the duration.

- Two **DISCRETELEGEND** statements are used to generate the response legend inside the bottom right of the plot and treatment group legend outside the plot at the bottom.

## FOREST PLOT

The forest plot is a complex graph with many presentations. An example of a forest plot is shown in Figure 5. If we generate this plot using **SAS 9.2** in GTL, a three-celled lattice layout is need in the implementation including the left subgroup description, the center plot, and the right risk difference with CI presentation with long code in GTL. Thanks to **SAS 9.4**, GTL provides a new statement **AXISTABLE**, which enables the user to write simple and short code within only one overlay layout.

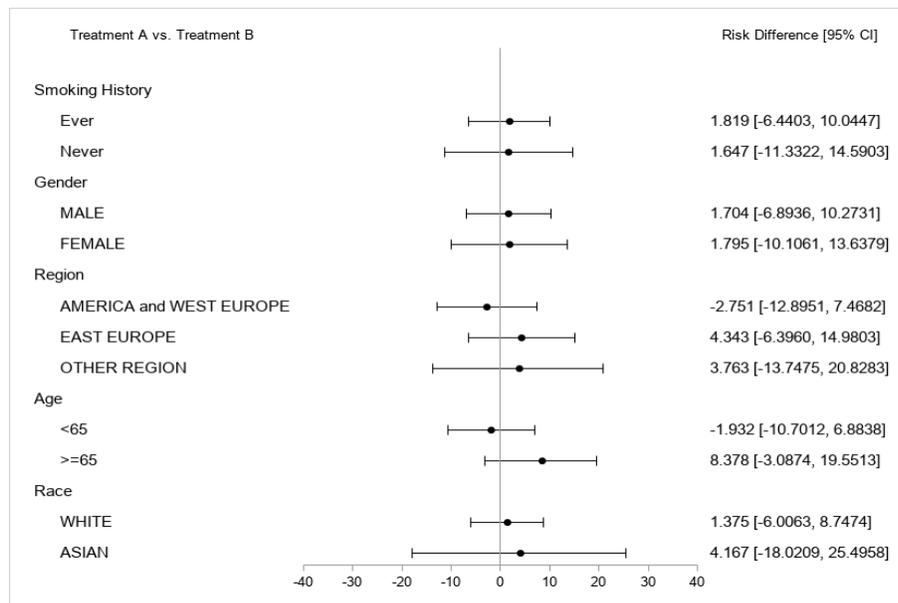


Figure 5: Example of Forest Plot

PROC TEMPLATE for Figure 5 using SAS 9.4M3 or later version:

```

PROC TEMPLATE;
  Define statgraph forest;
    Begingraph;
      Layout Overlay/ /*1*/
        yaxisopts=(reverse=true display=none)
        xaxisopts=(display=(ticks tickvalues line)
          LINEAROPTS=(viewmin=-40 viewmax=40
            TICKVALUESEQUENCE=(start=-40 end=40 increment=10)))
        WALLDISPLAY=none
        LINEEXTENT=data; /*2*/
        /* Left-side table */
        Innermargin/align=left;
          Axistable y=obnum value=subgroup/ /*3*/
            INDENTWEIGHT=indent
            LABEL="Treatment A vs. Treatment B" /*4*/
            VALUEATTRS=(size=10pt);
        Endinnermargin;
        /* Risk Difference */
        Scatterplot Y=obnum X=stat/ /*5*/

```

```

XErrorUpper=lower XErrorLower=upper
MARKERATTRS=(symbol=circlefilled color=cx000000)
ERRORBARATTRS=(color=cx000000);
Referenceline x=0 / lineattrs=(pattern=1);

/* Right-side table */
Innermargin/align=right;
Axistable y=obnum value=value/ /*3*/
LABEL="Risk Difference [95% CI]" /*4*/
VALUEATTRS=(size=10pt);
Endinnermargin;
Endlayout;
Endgraph;
End;
Run;

```

1. **Layout Overlay - Endlayout** is the block to generate the forest plot.
2. **LINEEXTENT** is a new option for axis setting from third maintenance release of SAS 9.4 (**SAS 9.4M3**), which can control the extents of an axis line to justify the data area with the option **LINEEXTENT = data**. This means the x-axis line will not display in the left and right margin areas.
3. **Axistable** statement can be used to present the information inside the overlay layout's inner margin, which is also a new statement in **SAS 9.4** compared with **9.2**. This statement is very helpful to generate the forest plot or other similar plots that need the presentation or annotation besides X or Y axis.
4. **LABEL** option in the Axistable statement is used to display the description on the top of each Axistable area.
5. **Scatterplot** statement is for the risk difference plot with bars for representing the confidence intervals.

## TITLE, FOOTNOTE, PAGE X OF Y IN FIGURE OUTPUT

We usually present "Page X of Y" in the Graph output. An easy way to display the page number in the title or footnote of the PDF or RTF output file is **^{thispage}** and **^{lastpage}**, especially after **SAS 9.3**, **NOGTITLE** and **NOGFOOT** options are released to improve titles and footnotes display for Graphics Output using PDF or RTF file.

However, users need to check your SAS version, as **^{lastpage}** may have issue using ODS for graphics file output in SAS old version. The optional method is to use a macro variable to store the number of total page to replace **^{lastpage}**. Fortunately, this issue has been fixed in **SAS 9.4M5** release. If you use 9.4M5 or later version, there will be no any issues for **^{lastpage}**. More detailed explanation is in the link: <http://support.sas.com/kb/34/573.html>.

**Title** and **Footnote** statements can create the titles and footnotes of the output. **PROC TEMPLATE** is also used to define the titles and footnotes font and size to meet the customized requirement. The code below shows an example of PDF output file with titles, footnotes, and page x of y on each page.

```

proc sort data = sashelp.gas out = gas;
  by fuel;
run;
/* Output page setup */
OPTIONS orientation=LANDSCAPE /*1*/

```

```

        leftmargin=1.2in
        rightmargin=1in
        topmargin=1in
        bottommargin=1in
        nodate nonumber nobyline;
ods escapechar="^";
/* SAS old version using below title1 statement */
Title1 j=r "Page ^{thispage} of &totpage"; /*2*/
/* SAS 9.4M5 or later version using below title1 statement */
Title1 j=r "Page ^{thispage} of ^{lastpage}";
Title2 "Plot of example";
Footnote1 "This figure is just for the title and footnote example";
/* Titles and footnotes font and size setup using PROC TEMPLATE */
PROC TEMPLATE;
    define style setup;
        parent = Styles.journal;
        /* Output file titles font and size setup */
        STYLE SystemTitle / /*3*/
        FONT_FACE = "Courier"
        FONT_SIZE = 0.2in
        FONT_WEIGHT = light
        FONT_STYLE = roman
        BACKGROUND_COLOR = white
        JUSTIFY = LEFT;
        /* Output file footnotes font and size setup */
        STYLE SystemFooter /
        FONT_FACE = "Courier"
        FONT_SIZE = 0.2in
        FONT_WEIGHT = light
        FONT_STYLE = roman
        BACKGROUND_COLOR = white
        JUSTIFY = LEFT
    ;
End;
Run;
/* Output PDF file */
ODS PDF file = "C:\zhang\output.pdf"
    NOGTITLE NOGFOOT style=setup /*4*/
    StartPage=yes
    PDFTOC=1;
    ODS GRAPHICS ON/height=16cm width=22cm; /*5*/
proc sgplot data = gas;
    by fuel;
    styleattrs datalinepatterns=(solid);
    loess y=nox x=eqratio;
run;
ODS GRAPHICS OFF;
ODS PDF close;

```

1. **OPTIONS** is for the PDF layout setup.
2. **Title** and **Footnote** statements can be used to present the titles and footnotes in the PDF output file.
3. **STYLE SystemTitle** and **STYLE SystemFooter** in PROC TEMPLATE can be used to define the titles and footnotes font, size, and other styles.
4. In **ODS PDF** statement, it is suggested to add **NOGTITLE** and **NOGFOOT** options, which tells SAS that the titles and footnotes are outside the plot and presented as file titles and footnotes (No Graph Title or No Graph Footnote). **NOGTITLE** and **NOGFOOT** are not

available for the PDF destination until **SAS 9.3**. If you are still using **SAS 9.2**, these two options are not available.

5. **ODS GRAPHICS ON** is for the plot size setup.

The PDF output file (1<sup>st</sup> Page) using above example code is shown in Figure 6 below.

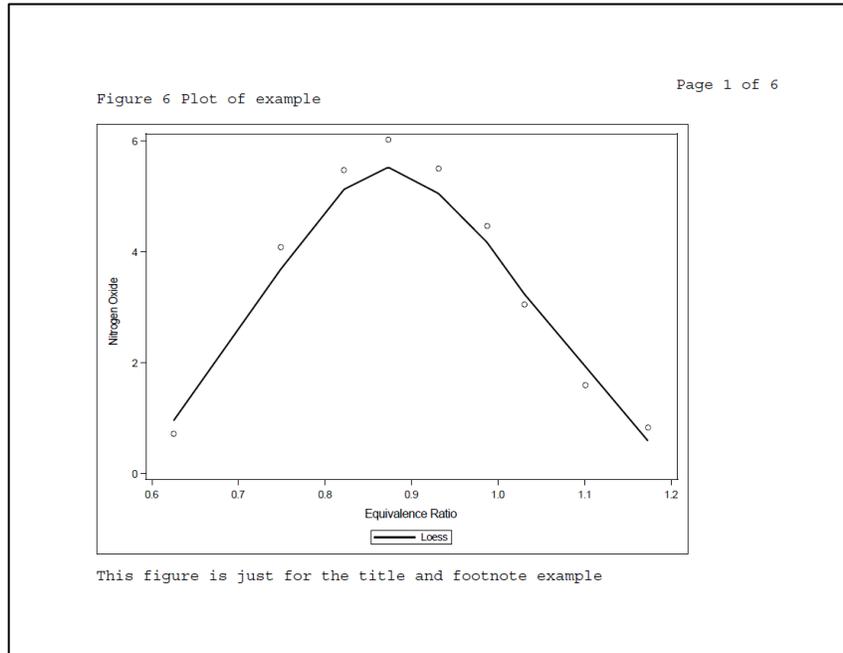
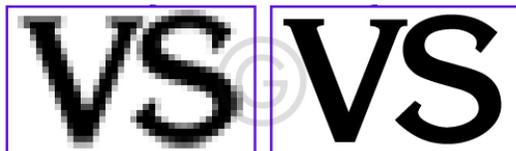


Figure 6: PDF Output Example with Titles and Footnotes

## VECTOR GRAPHICS FORMAT

**SAS 9.4** can output vector graphics format file, which is often required for the poster due to high quality figures presentation. User can output two kinds of vector graphics format in SAS 9.4. One is **.EPSI** file, the other is **.PS** file. But please note that **SAS 9.2** cannot output vector format plot even though the output file is **.EPSI** or **.PS** as the default graph output in 9.2 is bitmap format.

How can we distinguish between bitmap graphics and vector graphics? An easy way is that if we enlarge the figure as shown below, bitmap graphics (left) have discrete pixels that represent color. Vector graphics (right) are composed of lines and shapes.



Bitmap Graphics

Vector Graphics

The code below shows an example of vector graphics format output in **SAS 9.4**. Please note that **SAS 9.2** cannot output vector graphics using this example code.

```

/* EPSI file vector format figure using SAS 9.4 */
Ods Latex Path="C:\zhang";                               /*1*/
Ods Graphics on /Imagefmt=epsi Imagename="figure";       /*2*/
/* Plot Statements SAS Codings nested by ODS Graphics on and off */
.....
.....
.....
Ods Graphics off;
Ods Latex close;

/* PS file vector format figure using SAS 9.4 */
Ods Latex Path="C:\zhang";                               /*1*/
Ods Graphics on /Imagefmt=ps Imagename="figure";         /*2*/
/* Plot Statements SAS Codings nested by ODS Graphics on and off */
.....
.....
.....
Ods Graphics off;
Ods Latex close;

```

1. **Ods Latex Path** defines the output file directory or location in your Local PC or Server.
2. **Imagefmt** defines the vector graphics format and **Imagename** defines the output file name.

## CONCLUSION

There are many ways to generate high quality figures in SAS 9.4 combined with ODS styles and output such as: traditional procedures (eg: GPLOT, GCHART, GREPLAY, etc.), statistical procedures nested by ODS GRAPHICS, statistical graph procedures (eg: SGPLOT, SGSCATTER, SGPANEL), and Graph Template Language (GTL). GTL enables user to generate complex, advanced and customized figures, though it is difficult and even impossible for you to remember all statements and options of SAS GTL. The efficient way for programmer to enhance the graphics generation ability is more reading the guidance from SAS Institute Inc., more sharing and more hands-on practice.

## REFERENCES

SAS Institute Inc. 2016. SAS® 9.4 Graph Template Language: Reference, Fifth Edition. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2014. SAS® 9.4 Graph Template Language: Reference, Third Edition. Cary, NC: SAS Institute Inc.

SAS Institute Inc. 2010. SAS/GRAPH® 9.2: Graph Template Language Reference, Second Edition. Cary, NC: SAS Institute Inc.

SAS® 9.4 Graph Template Language: Reference, Fifth Edition  
<https://documentation.sas.com/?docsetId=grstatgraph&docsetTarget=titlepage.htm&docsetVersion=9.4&ocale=en>

SAS/GRAPH®  
<http://support.sas.com/software/products/graph/index.html>

## CONTACT INFORMATION

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

Wei (Tony) Zhang  
Pfizer (China) Research and Development Co.,Ltd. China  
+86-21-28935629  
wei.zhang18@pfizer.com