A Prompted Application to Easily Create Forest Plots with Inner Margin Tables SAS® Global Forum 2016

Ted Durie, SAS Institute Inc. ®

ABSTRACT

Two of the powerful features of ODS Graphics is the ability to create Forest Plots and add Inner-Margin tables to graphics output. The drawback, however, is the syntax required from proc template is complex and tedious. A prompted application or even a parameterized stored process that connects proc template code to a point and click application would make the task of generating the fore-mentioned output much easier.

INTRODUCTION

The graphic output featured is designed to compare favorable verses non-favorable results across treatments--remedies used in various kind of health care studies. The output is typically augmented with study information and statistics the left and to the right of the graphic output.

EXAMPLES OF FOREST PLOTS FROM GOOGLE
GENERAL FORM OF FOREST PLOTS

Figure 2

Forest Plots can vary greatly in terms of information and statistics displayed in the Left and Right Report. The graph in the middle is typically of constant form.

GRAPHIC OUTPUT FORM.

Figure 3.
The graphic output in the middle of the Forest Plots consists of two over-laid high-low plots with a logarithmic axis that frequently varies in overall minimum and maximum scale. The overall marker must be calculated from the statistics of the individual weighted studies, and the size of the overall marker will vary based off the range between the overall lower and higher odds-ratio calculated.

APPLICATION REQUIREMENTS

- SAS ®Version 9.4
- Base SAS® (ODS Graphics)
- Enterprise Guide version 6.1 or above.
- Business Analytics Environment.
- Office Analytics Add-in for Microsoft Office

KNOWLEDGE REQUIREMENTS

- ODS Graphics syntax SAS® Version9.4
- Data Step Code
- Proc Freq to calculate ODDs Ratio Statistics
- Macro Language concepts
- Creating Prompts in Enterprise Guide
- Generating and Using Stored Processes

FEATURES OF THE APPLICATION

- Select the data set to analyze
- Automatic resizing of output as number of studies in graph varies.
- Pick and choose statistics
- Control the Order of Statistics
- Apply Custom colors to each field
- Add Custom Titles and Footnotes
- Reposition text and markers
- Automatically Surface results in Microsoft Word
DEMO OF THE APPLICATION: SCREEN 1

Figure 4.

As the Stored Process opens the user 1.) Selects the study data set. 2.) Moves the statistics to include in the report from the left Available window to the right Selected 3.) Include the Statistics for the right report. The overall left to right order of statistical information can be modified with the up and down arrows at the right of the selected window.

DEMO OF THE APPLICATION: SCREEN 2

Figure 5

The color drop-downs shown in Figure 5 represent the individual colors which can be applied to any of the columns in the left or right report.
Up to two Titles and Footnotes can be added to the graphic output from the application. The size of the second title will be slightly smaller than the first.

Often as the number of studies in the overall analysis increases and decreases the horizontal location of the "Favor Treatment" and "Favors Placebo" will have to be adjusted. Often the overall logarithmic scale and range will change from one Forest Plot to another. The smaller the axis scale the wider the high-low plots on the graph will get.
**FINAL RESULT**

Figure 8.

**Figure 8**

**Forest Plot for Alzheimer Study is displayed.**

### GENERAL APPROACH USED TO DEVELOP APPLICATION

- Stabilize Graphic Model.
- Calculate ODDS Ratio Statistics with Proc Freq
- Program works with hardcoded values.
- Generalize solution with macro references.
- Create Macro Sub-routine
- Add decision logic and iterative code.
- Create Prompts in Enterprise Guide
- Connect Prompts to SAS Program
- Convert overall routine into a Stored Process
- Surface results across BA Tools

### ORIGINAL HARDCODED PROC TEMPLATE SYNTAX

**Figure 9**

```sas
proc template;
  define statgraph Forest_three;
  dynamic STUDY N Combo weight pertry pertrn perplN trN plY trY
  _PCORR_ U_RROR L_RROR X_High X_Low _RRC2_ L_RRC2 U_RRC2;
  mvar obsid group subgroup lstats rstats;
  begingraph / designheight=1000 designwidth=1400;
    entrytitle "Impact of Treatment on Mortality by Study";
    entrytitle textattrs=(size=10) 'Odds Ratio and 95% CL';
    layout lattice / columns=1 /columnweights=(0.35 0.35 0.30 0.25);
    layout overlay / walldisplay=none
    xaxisopts=(label='Favor Treatment Favorable')
    yaxisopts=(label='Favor Treatment Favorable')
  ;
  endgraph;
end;
```

---

**TABLE 1: Alzheimer Study Odds Ratio Medication Effectiveness**

<table>
<thead>
<tr>
<th>STUDY</th>
<th>Treatment Favourable</th>
<th>Treatment Adverse</th>
<th>Placebo Favourable</th>
<th>Weight</th>
<th>P-Value</th>
<th>Odds Ratio</th>
<th>Lower 95% CI</th>
<th>Upper 95% CI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nyugen (1991)</td>
<td>6.349</td>
<td>6.997</td>
<td>0.276</td>
<td>0.034</td>
<td>0.042</td>
<td>6.293</td>
<td>6.357</td>
<td>6.472</td>
</tr>
<tr>
<td>Molbis (1992)</td>
<td>6.749</td>
<td>9.25</td>
<td>0.276</td>
<td>0.036</td>
<td>0.048</td>
<td>6.528</td>
<td>6.955</td>
<td>6.995</td>
</tr>
<tr>
<td>Rivas (1993)</td>
<td>4.274</td>
<td>9.971</td>
<td>0.276</td>
<td>0.035</td>
<td>0.049</td>
<td>6.548</td>
<td>9.253</td>
<td>9.499</td>
</tr>
<tr>
<td>Lawrence (1993)</td>
<td>3.644</td>
<td>9.111</td>
<td>0.276</td>
<td>0.044</td>
<td>0.054</td>
<td>6.917</td>
<td>9.302</td>
<td>9.544</td>
</tr>
<tr>
<td>Armstrong (1994)</td>
<td>6.049</td>
<td>9.605</td>
<td>0.276</td>
<td>0.069</td>
<td>0.085</td>
<td>6.286</td>
<td>9.116</td>
<td>9.392</td>
</tr>
<tr>
<td>Champion (1997)</td>
<td>2.349</td>
<td>9.128</td>
<td>0.276</td>
<td>0.068</td>
<td>0.083</td>
<td>6.386</td>
<td>9.266</td>
<td>9.499</td>
</tr>
<tr>
<td>Parkinson (1997)</td>
<td>6.049</td>
<td>9.654</td>
<td>0.276</td>
<td>0.085</td>
<td>0.094</td>
<td>6.352</td>
<td>9.323</td>
<td>9.454</td>
</tr>
<tr>
<td>Beste (1998)</td>
<td>4.049</td>
<td>9.493</td>
<td>0.276</td>
<td>0.065</td>
<td>0.078</td>
<td>6.359</td>
<td>9.338</td>
<td>9.479</td>
</tr>
<tr>
<td>Wilhelm (1998)</td>
<td>4.549</td>
<td>9.911</td>
<td>0.276</td>
<td>0.077</td>
<td>0.092</td>
<td>6.359</td>
<td>9.338</td>
<td>9.479</td>
</tr>
<tr>
<td>U of M (1999)</td>
<td>2.349</td>
<td>9.685</td>
<td>0.276</td>
<td>0.066</td>
<td>0.083</td>
<td>6.386</td>
<td>9.266</td>
<td>9.499</td>
</tr>
<tr>
<td>Tamats (2000)</td>
<td>4.549</td>
<td>9.911</td>
<td>0.263</td>
<td>0.084</td>
<td>0.094</td>
<td>6.352</td>
<td>9.323</td>
<td>9.454</td>
</tr>
<tr>
<td>University of Kansas (2003)</td>
<td>3.049</td>
<td>9.911</td>
<td>0.263</td>
<td>0.084</td>
<td>0.094</td>
<td>6.352</td>
<td>9.323</td>
<td>9.454</td>
</tr>
<tr>
<td>Travis (2004)</td>
<td>2.349</td>
<td>9.685</td>
<td>0.276</td>
<td>0.066</td>
<td>0.083</td>
<td>6.386</td>
<td>9.266</td>
<td>9.499</td>
</tr>
<tr>
<td>O'Harahan (2008)</td>
<td>6.399</td>
<td>9.222</td>
<td>0.276</td>
<td>0.071</td>
<td>0.094</td>
<td>6.297</td>
<td>9.333</td>
<td>9.442</td>
</tr>
<tr>
<td>Oakley (2015)</td>
<td>5.748</td>
<td>9.25</td>
<td>0.276</td>
<td>0.071</td>
<td>0.094</td>
<td>6.297</td>
<td>9.333</td>
<td>9.442</td>
</tr>
<tr>
<td>Overall Effect</td>
<td>4.549</td>
<td>9.685</td>
<td>0.276</td>
<td>0.085</td>
<td>0.094</td>
<td>6.352</td>
<td>9.323</td>
<td>9.454</td>
</tr>
</tbody>
</table>
The original graphic syntax required from Proc template and sgrender is lengthy, layered and tedious to make individual edits to. As an application like this is developed, one has to make sure that the graphic model is stable and creates the basic graphic output required. The hardcoded solution will be connected to the prompted application, and automated thru use of the SAS Macro Language.
The above proc freq syntax is needed to calculate Odds Ratio’s. The code above in Fig 10. Was originally generated by the One-Way frequencies task in Enterprise Guide. Most importantly the “Measures option” had to be selected.

This output measures table was created by proc freq. The biggest challenge was determining what each of the cryptic field names meant and which one were required or might be optionally included in a Forest Plot.
FINDING A DESCRIPTION OF THE STATISTICS CALCULATED

Figure 12.

Looking at the column properties of the output table created by proc freq, a label or description appears which is very helpful in determining what is needed by the Forest Plot. Wouldn’t it be nice if these labels could automatically be loaded into the point and click application and selected as the reference to the columns to include in the left and right report.

CONVERTING FIELD NAMES INTO MACRO VARIABLE AND LIST OF VALUES IN PROMPTED APPLICATION

Figure 13.

These three coded steps use dictionary columns to dynamically get the field names created by proc freq. The information is converted into a macro list and user-defined format that is used later in the application.

```
proc sql noprint;
create table gfdata.statistics as
select name as start.label, ''$statfmt'' as fmtname
from dictionary.columns
where upcase(libname)='WORK' and upcase(memname)='COMBINE' and
upcase(type)='NUM' and upcase(name) not in ('ORDER');
quit;

proc sql noprint;
select name into &namelist separated by ','
from dictionary.columns
where upcase(libname)='WORK' and upcase(memname)='COMBINE' and
upcase(type)='NUM' and upcase(name) not in ('ORDER');
quit;

%put &namelist;
%let totstats=%sysfunc(countw(&namelist,#));
proc format cntlin=gfdata.statistics lib=gfdata;
select statfmt;
run;
options fmtsearch=(gfdata);
```
USING THE PROMPT MANAGER

Figure 14.

From the Resources pane in the lower left corner of Enterprise Guide the Prompt manager can be accessed and opened. Items or Objects in the point and click application, like list boxes, text entry fields, color selectors can be configured and added to the application.

CONFIGURING A PROMPT

Figure 15.

From Fig 15 on the previous page. 1.) The type of prompt, text, is selected. 2.) In the Business Analytics environment the List box can loaded statically, or dynamically. 3.) The Prompt Type and Values tab
allows for the creation of ordered multi-selection list boxes in EG Version 7.1. This latest feature allows statistics, therefore columns in the final graphic output to be selected in specific left to right order.

HOW CODE LOADS STATISTICS INTO A PROMPT

Figure 16.

From the Get values push button in Figure 16. The statistic field names created by proc freq can be loaded into a point and click prompt. And the formatted label/descriptions can displayed in the application that the user sees.

USING MACRO VARIABLE VIEWER TO FIND NAMES CREATED BY PROMPTS

Figure 17.

Once the prompt is configured going to Tools→Macro Variable Viewer and a developer can record the name of the macro variables that will have to be incorporated in the underlying program.
CONNECTING PROMPT NAMES TO CODE

Figure 18.

The ampersand references, (macro variable names), used in the underlying program, must match the names of the prompts generated from the prompt manager.

PROGRAM PROPERTIES (CONNECTING PROMPTS TO APPLICATION PROGRAM)

Figure 19.

Each of the Prompts in the Application must be also be manually connected to the properties of the Forest plot program. The up and down arrows at the right-hand side of the current window, control the relative top to bottom order of the object items in the application.
CONVERTING HARDCODED TEMPLATE SYNTAX INTO DYNAMIC MACRO SYNTAX

Figure 20. Original syntax

```latex
\begin{verbatim}
proc template;
   define statgraph Forest_two;
   dynamic STUDY N Combo weight pertry perTrn perPlY perpIlN trY plY trN plN
      _PCORR_ U_RROR L_RROR X_High x_Low _RRC2_ L_RRC2
      U_RRC2;
   mvar obsid group subgroup lstats rstats;
\end{verbatim}
\end{figure}
```

Figure 21. Automated Macro Equivalent

```latex
\begin{verbatim}
proc template;
   define statgraph Forest_four;
   dynamic studytext
      \%do I=1 %to &LSTATS0;
       &lstats&I&blank
      \%end;
   \%do J=1 %to &RSTATS0;
       &rstats&J&blank
      \%end;
   mvar
      \%do I=1 %to &LSTATS0;
       lstats&I&blank
      \%end;
   \%do J=1 %to &RSTATS0;
       rstats&J&blank
      \%end;
\end{verbatim}
```

Syntax Generated by Macro Program

```latex
\begin{verbatim}
MPRINT(FORREST2): proc template;
MPRINT(FORREST2): define statgraph Forest_five;
MPRINT(FORREST2): dynamic studytext N perTrF perTrN perPlF weight
   _PCORR__ _RRC2__ L_RROR U_RROR mvar lstats1 lstats2 lstats3
   stats4 rstats1 rstats2 rstats3 rstats4 rstats5 pcorr_color
   odcolor overr overu overl;
\end{verbatim}
```

The three sections of syntax listed in the figures above represent the original hard-coded proc template dynamic statement (figure 20). The macro version of the syntax (figure 21). And the final code generated by the macro program and displayed with the MPRINT option in the log (figure). If a user changes the columns used in the left or right report, the underlying program syntax is automatically modified and in this case the new dynamic statement is created with the latest required syntax.
SYNTAX WHICH GENERATES BASIC GRAPHIC OUTPUT

Figure 22.

```/*--Graph in the middle--*/
HighLowPlot Y='study'n High='U_RROR'n Low='L_RROR'n / primary=true Lineattrs=( Color= CX83838C thickness
HighLowPlot Y='study'n High='x_high'n Low='x_low'n / type=bar barwidth=.4 Display=( Fill ) FillAttr=(
ReferenceLine x=0.10 / clip=true Lineattrs=( Color= CX83838C Pattern=2 Thickness=2px);
ReferenceLine x=0.01 / clip=true Lineattrs=( Color= CX83838C Pattern=2 Thickness=2px);
ReferenceLine x=1 / Lineattrs=( Color=black Pattern=1 Thickness=3px);
ReferenceLine x=10 / Lineattrs=( Color= CX83838C Pattern=2 Thickness=2px);
ReferenceLine x=100 / Lineattrs=( Color= CX83838C Pattern=2 Thickness=2px);
```

The two different versions of the high-low statement will create a high-low line with a high low bar overlaid on top of it. The U_RROR and L_RROR field used in the syntax were calculated by the proc freq code shown previously. Reference line will be included for the logarithmic horizontal axis. Very little macro conversion was required for this syntax.

COLUMNS OF REPORT OUTPUT CREATED BY AXIS TABLE STATEMENTS

Figure 23.

```
Hardcoded Statements
axistable y='study'n value='study'n / display=(label) labellposition=max labelattrs=(size=10pt weight=bold)
axistable y='study'n value='try'n / display=(label) labellposition=max labelattrs=(size=10pt weight=bold)
axistable y='study'n value='trn'n / display=(label) labellposition=max labelattrs=(size=10pt weight=bold)
axistable y='study'n value='pertry'n / display=(label) labellposition=max labelattrs=(size=10pt weight=bold)
axistable y='study'n value='pertrn'n / display=(label) labellposition=max labelattrs=(size=10pt weight=bold)

%do L=1 %to &size; Macro Equivalent
axistable y='studytext'n value=&&LSTATS&L'n / display=(label) labellposition=max
labelattrs=(size=10pt weight=bold) valueattrs=(size=12pt color=&&Lstat_color&L);
%end;
```

The iterative macro %DO loop syntax will create a separate custom axistable statement for every left and right statistic that was selected from the application. The specific color for the column is incorporated here as well.
MACRO CONVERSION OF SGRENDER STATEMENTS NEEDED TO DISPLAY TEMPLATE

Figure 24.

Hardcoded Statements

```
proc sgrender data=work.combine template=forest_three spannames=makeanno;
  dynamic study='study' N='N'
  perTr='perTr
  perTrF='perTrF
  weight='weight
  PCORR='PCORR
  _RRCORR='_RRCORR
  RRC2='RRC2
  L_RCORR='L_RCORR
  U_RCORR='U_RCORR
  run;
```

Macro Equivalent

```
%let equal=칵스트();
%let double=cockstr('');%let rtext;
%let single=cockstr('%');%let blank=cockstr();
proc sgrender data=work.combine template=forest_five;
  dynamic studytext='studytext'
  perTr='perTrF'
  weight='weight'
  PCORR='PCORR'
  _RRCORR='_RRCORR'
  RRC2='RRC2'
  L_RCORR='L_RCORR'
  U_RCORR='U_RCORR'
  run;
```

The final syntax required in the program to display the graphic template is proc sgrender. Automated macro generated DYNAMIC statements had to be generated here as well.

CUSTOM VERBAGE ADDED TO GRAPH WITH DRAWTEXT

Figure 25.

```
drawtext textattrs=(color=Blue) "Favors Treatment" / x=adjustfactor1 y=10 anchor=center

drawtext textattrs=(color=Blue) "Favors Placebo" / x=adjustfactor2 y=10 anchor=center
```

For ODS Graphics SAS version 9.4 and later a new drawtable statement is available, which simplifies the old required SGANNOTATE syntax. The code in Figure adds and dynamically positions the “Favors” label near the bottom axis.

CUSTOM VERBAGE ADDED TO GRAPH WITH DRAWTEXT

Figure 26.

```
drawtext textattrs=(family='Times New Roman Uni' color=green size=45) \[unicode '258A']x
  / y='Overall Effect' anchor=center drawspace=datavalue layer=front width=250 widthunit=pixel;

drawtext textattrs=(family='Times New Roman Uni' color=green size=45) \[unicode '2554']x
  / y='Overall Effect' anchor=center drawspace=datavalue layer=front width=250 widthunit=pixel;
```

The other drawtext statements add a green leftward and rightward pointing arrow to depict the overall symbol normally included in Forest Plot output. The Unicode hexa-decimal value was retired from the character map shown in the next figure.
Any character from a Unicode font can be added to a graphic display at any x and y coordinate location via a drawtext statement. The character map can be accessed by going to Start ➔ Programs ➔ Accessories ➔ System Tools ➔ Character Map.

If an Enterprise Guide User whose organization has the Business Analytics Architecture installed, right-clicks on a program entry in either the process flow diagram or project tree, a menu will appear that brings up a point and click stored process wizard. This wizard will define the metadata, or information needed to allow the Forest Plot application to execute from any computer on an organization’s intranet and surface the results in Microsoft Tools like Word, PowerPoint and Excel.
In the first wizard window the developer defines a formal name for the stored process and a storage folder, which should ultimately allow the application to be accessed from any Business Analytics Tool.

Additional configurations, particularly in wizard windows 3 and 5 are required to define how the program code will be accessed and where it will be executed. In wizard 5, every prompt created in Enterprise
Guide is automatically registered to the stored process. Additional functionality can be added to the application here like cascading prompts, and nested prompts not available in Enterprise Guide.

**SETTING ACCESS PERMISSIONS FOR A STORED PROCESS**

Figure 31.

Once the Stored Process has been configured from the wizard, a user should go over to the SAS Management Console and insure the routine has the right permission for different uses and groups to access. The application developer or SAS Business Analytics administrator will navigate to the folder selected in the previous stored process wizard 1 and set the permissions.

**PERMISSIONS FOR A STORED PROCESS**

Figure 32

Comprehensive security can be defined from the SAS Management Console to the users and groups obtaining access to the application. The console is part of the office analytics environment.
Once the stored process has been registered in the SAS Management Console, a user can go over to Microsoft Word, PowerPoint, or Excel access the SAS Add-In and select the Report Icon from the SAS Ribbon. Then a SAS Folder Window will appear, and the application user can navigate to the location where the stored process was stored, click on the entry and run the point-and-click application.

SAME APPLICATION SHOWN EARLIER IN THE PAPER WILL APPEAR. THE USER CAN SELECT AND DEFINE THE PARTICULAR ATTRIBUTES OF THE FOREST PLOT, HIT THE RUN BUTTON, AND DISPLAY THE OUTPUT.
The output from the application is automatically inserted into a word document which is being used as documentation for health care studies being submitted to the FDA.

CONCLUSION

As the paper demonstrates creating Forest Plots using a point and click prompted stored process is definitely feasible. The basic application presented can be modified further to incorporate additional customization features. Knowledge from several SAS disciplines and products is required to build and deploy an application like this one. If your organization frequently creates graphic output of this form, then in addition to base SAS software, the SAS Office Analytics environment will have to be licensed.

REFERENCES


CONTACT INFORMATION

If you would like a copy of the Enterprise Guide Application and SAS Program used to generate the output contact the writer via the information below. In addition another solution to the Forrest Plot problem was explored and somewhat developed using proc report, proc sgplot, and ODS Layout. This application showed real promise to extend the capabilities of the current application, but was not specifically related to the abstract.

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

Ted Durie Ted.Durie@sas.com
SAS Institute
Austin, TX

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.