

Modifying Results of PROC GLM with PROC TEMPLATE

Catalina Mejía Herrera, Bayer Pharma AG, Berlin, Germany
Volker Harm, Bayer Pharma AG, Berlin, Germany

ABSTRACT

This paper will present the graphical capabilities of PROC GLM, an overview of the PROC FORMAT with the PICTURE statement will be introduced and specific remarks and advice will be given when dealing with the powerful tool of PROC TEMPLATE. This is presented by way of an example of an adapted study report.

INTRODUCTION

Normally the researcher is satisfied with the aspect of SAS® output until the requirement of a specific task has to be modified. The motivation for this paper arose from the need to report understandable statistical results of a toxicology study (graphs and test outcomes) for clinical researchers with little a statistical background. In this case we want to show them a p-value with a “star” when the alternative hypothesis of a test was met. With the production release of ODS Graphics in SAS 9.2., one can produce graphical output for the statistical procedures this is a big advantage to avoid the creation of the graphics “manually” for the specific statistical model by using the graphical procedures. ODS templates are used to characterize the overall attributes of an output file such as formats, size, color and fonts. To achieve our purpose, it possible to create a format for the p-value when the p-value is smaller than the significance level and plug it into the default ODS template by using the procedure PROC TEMPLATE.

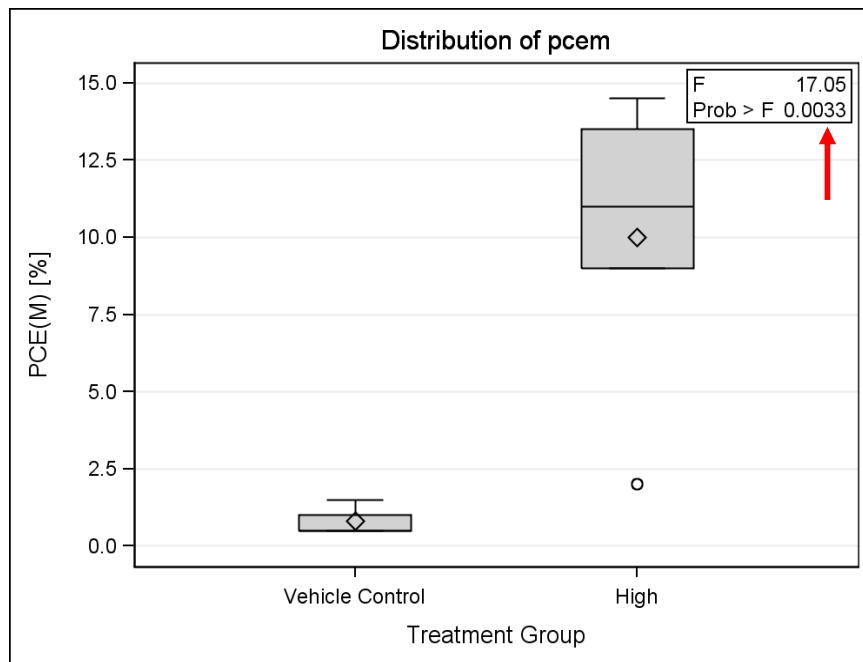
STANDARD OUTPUT OF PROC GLM

Having the output delivery system for the graphs enabled (ODS Graphics on) before submitting a SAS/STAT procedure one can automatically have the relevant graphs linked to the analysis. The graph shown below was created to give a representation of the linear model set out for a toxicology study by using ODS graphics and the PROC GLM . The name of pcem corresponds to the label of a toxicology end point.

```
ods graphics on / height = 9.5 cm;
proc glm data = work.AnalysisDataSet order = internal;
  class treat;
  model &EndPoint. = treat;
  means treat / lsd cldiff;
  contrast "Neg. Control vs. Dose 1"
    treat      -1 1;
  contrast "Neg. Control vs. Dose 2"
    treat      -1 0 1;
  contrast "Neg. Control vs. Dose 3"
    treat      -1 0 0 1;
run;
quit;
ods graphics off;
```

The output provides the overall statistical p-value of the test in the right upper corner of the graph as well as the usual table output. The aim is to have this value highlighted by a star when the test shows a significant result. This can be realized by creating a special format for this value.

PhUSE 2011



Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Neg. Control vs. Dose 3	1	211.6000000	211.6000000	17.05	0.0033

CREATE THE FORMAT

The appropriate format type for our goal is a PICTURE format. A PICTURE format creates a template that is used to display the values of numeric variables. The PICTURE format does not need to know all possible values of the numeric variable to be formatted. Since we will not have known the values for p-value in advance, it is necessary to use a global format for all possible values.

Using the following code produces the general format which includes a star when the p-value is below the significance level of 5%:

```
libname <name of the library> "<path destination>";
proc format library = <name of the library>;
    picture PvalueStar (round)
        low -< 0.05 = '9.9999*'
        0.05 -< 0.9999 = '9.9999'
        0.9999 -< 1 = '9';
run;
```

- ❶ PvalueStar refers to the name of the format. The round option ensures that the p-value is rounded and not truncated.
- ❷ The strings of "9's" are called digit selectors. They are used to tell PROC FORMAT how many "spaces" in the Picture format are needed to display the values of the numeric variable. A digit selector could be zero or an integer between 1 and 9. The most common digit selectors are the zero and nine, however there are slight differences between both of them.

If we want to change a default format of a graph, then the template has to be changed.

FINDING THE APPROPRIATE ODS TEMPLATE

Once the format is created, the next step is to find the appropriate ODS templates elements to alter. An easy way to find the templates is by exploring the contents of the Template store in the Display Manager.

To open the SAS Template store using the Display Manager, follow the steps:

- 1) Click on View in the top menu bar.
- 2) Click on Results in the drop-down menu.
- 3) Select View once again in the top menu bar.

PhUSE 2011

- 4) Click on Templates in the drop-down menu.
- 5) Expand the SasHELP.Tmplmst.
- 6) Select the SAS module you are interested in.
- 7) Double click on selected template. It will provide you with the code that defines the template in the Template Browser.

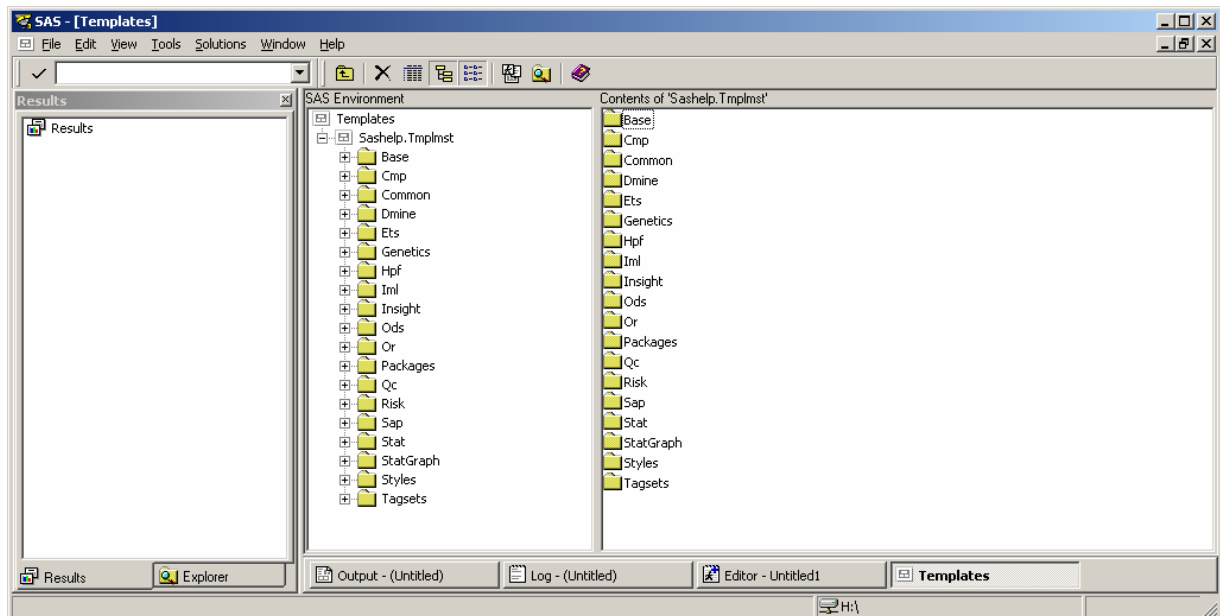


Figure 1: Shows the SAS Template store window using the Display Manager.

One can also find templates by using PROC TEMPLATE with the SOURCE statement and the location of the templates.

```
proc template;
  source stat.glm;
run;
```

Displays all templates elements that compose the GLM procedure in the STAT folder in the log window.

The interest is to modify the format of the p-value of the ANOVA table, thus in the list of all templates produced by the preceding code in the log window will be searched for the item path of the p-value. Item path are SAS folders separated with dots.

```
proc template;
  source stat.glm.probf;
run;
```

Provides the item path common.anova.probf. in this case this means that the p-value used in the anova table was defined in the template folder anova which can be found in folder common

Searching in the item path common.anova.probf, in the same way, will provide the item path of the p-value, common.pvalue, after the option PARENT. The PARENT option in the PROC TEMPLATE indicates that the template definition is in part based on another template. Finally seeking in the last item path common.pvalue gives the code of template definition in the log window:

```
proc template
  source common.pvalue;
run;
```

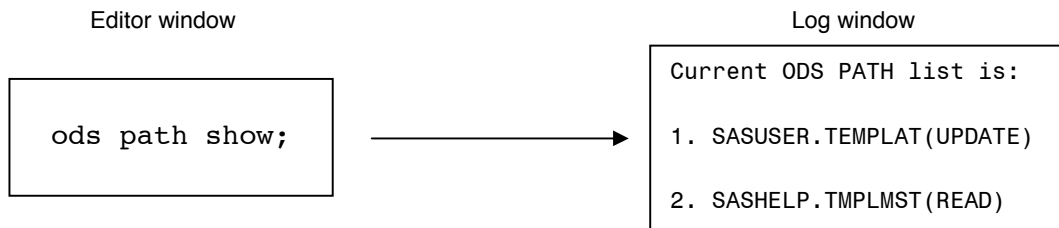
```
NOTE: Path 'Stat.GLM.ProbF' is in: SASHELP.TMPLMST.
3619   SOURCE COMMON.PValue;
define column Common.PValue;
  notes "Default p-value column";
  just = r;
  format = pvalue6.4;
end;
NOTE: Path 'Common.PValue' is in: SASHELP.TMPLMST.
3620   RUN;
NOTE: PROCEDURE TEMPLATE used (Total process time):
      real time           0.00 seconds
      cpu time            0.00 seconds
```

PhUSE 2011

Now it is known that the p-value has a format called pvalue6.4. Our aim is to swap the pvalue6.4 format with the new PvalueStar format.

STORING TEMPLATES

Before modifying ODS templates it is important to understand how SAS stores them. Templates are stored in an item store, which is a type of SAS file. There are two main stores: SASUSER.TEMPLAT and SASHELP.TMPLMST. Template stores can have read or update access. SASHELP.TMPLMST has only read access and it contains the SAS compiled templates. SASUSER.TEMPLAT has update access, thus user created templates will be stored there. An ODS PATH statement identifies SAS templates stores, allows the possibility to read and write templates, and defines the search order. To list the template stores, submit the following code:



By default, personally created templates are stored in SASUSER.TEMPLAT, because this is the first template store with update access in the path. It is possible to create your own store and store created templates there instead. It is highly recommended to create your own template store, since it is easier to delete the individual elements. Elimination of the customized template is also a good habit, so that the edited template is not accidentally used later. To set a template path in order to create a user template store, use the follow example code:

```
ods path work.mystore(update) sashelp.tmplmst;
```

CREATE A TEMPLATE

Creating a simple template with a few options is not difficult:

```
proc template;  
  define column common.pvalue /store=work.mystore; ❶  
  notes "Default p-value column";  
  just = r;  
  format = PvalueStar.; ❷  
end;  
run;
```

- ❶ The Define statement begins the element definition. The element to be created is a column of a table with common.pvalue as item path. To save all edited templates in work.mystore it is necessary to use the STORE option in the TEMPLATE procedure.
- ❷ The new format PvalueStar is applied to the created template.

More complicated templates can be challenging, and it is therefore convenient to edit or inherit code.

The p-value template for the ANOVA table with the new format will be edited as follows:

```
proc template;  
  edit common.pvalue / store = work.mystore;  
  format = PvalueStar.;  
end;  
run;
```

The easiest way to inherit a template is using the Display Manager, by clicking in the desired template the code will appear in the template browser, copy it into your editor and modify the appropriate statements to achieve your purpose.

PhUSE 2011

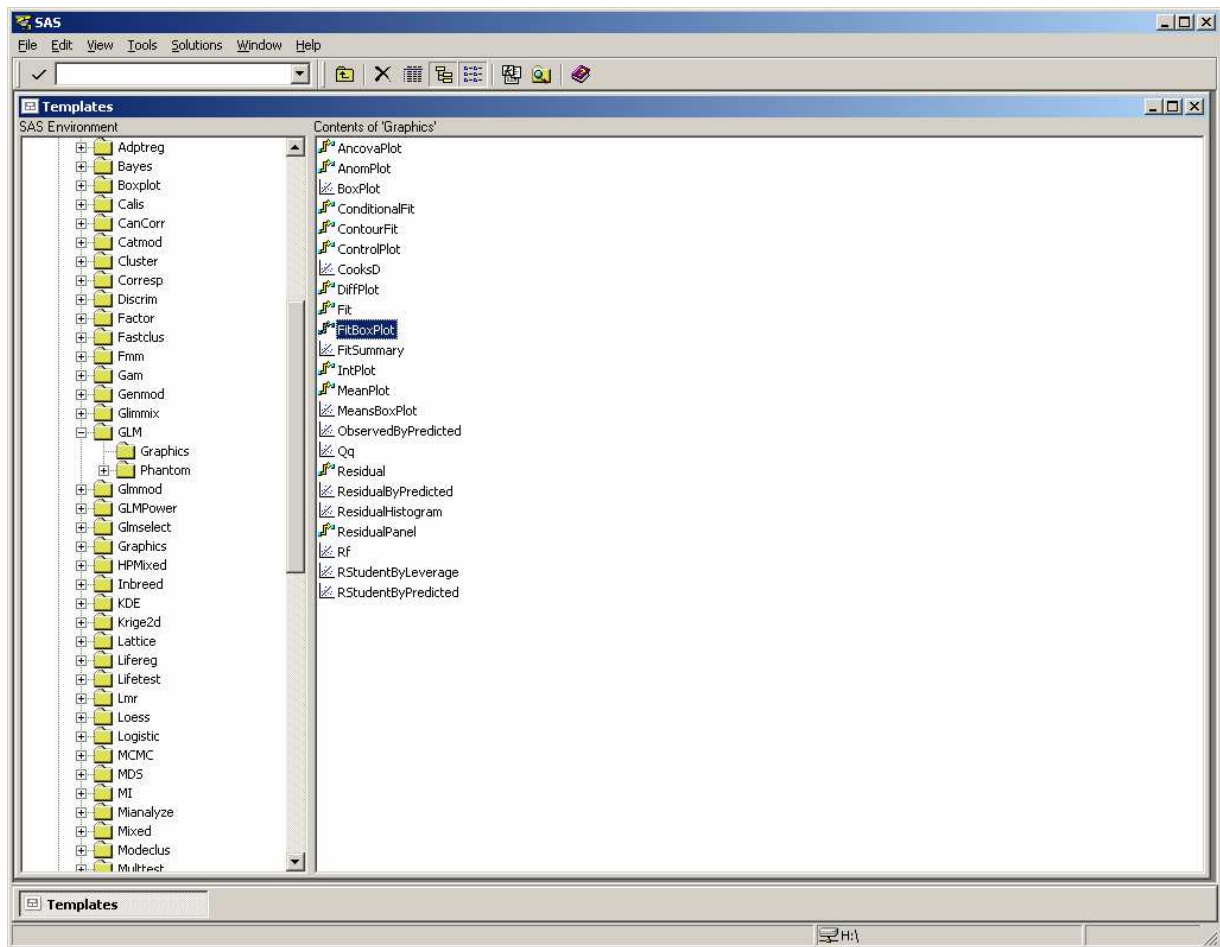


Figure 2: Shows the Template Store where the default ODS template of the box plot (FitBoxPlot) is defined.

The SAS code of the box plot template, that appears in the template browser, is as follows. The highlighted edits are to apply the new format for the p-value and to save the modified template in our personal template store using the STORE statement.

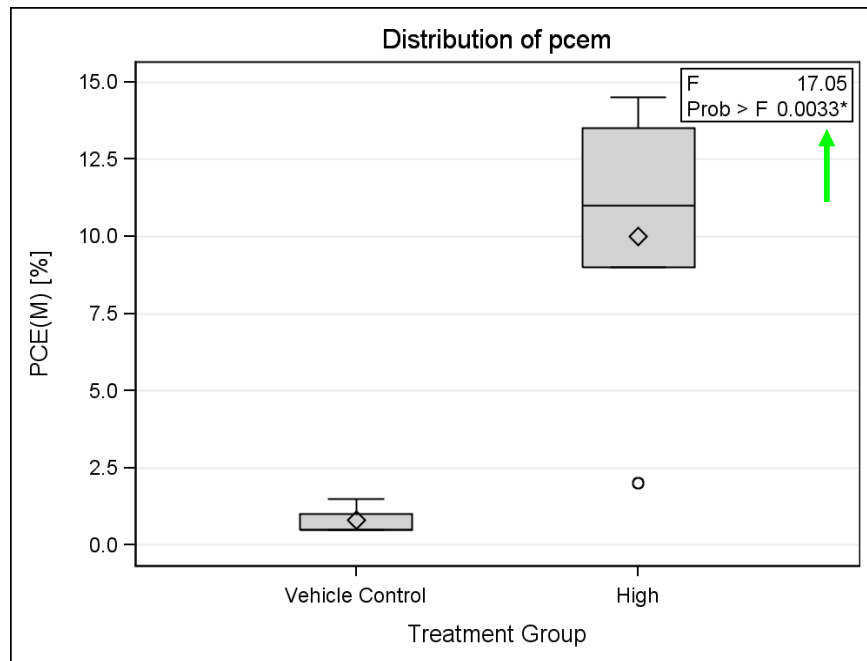
```
proc template;
  define statgraph Common.Zreg.Graphics.FitBoxPlot/ store=work.mystore;
    notes "Effect Plot with x=CLASS";
    dynamic _TITLE _XLABEL _SHORTXLABEL _YLABEL _SHORTYLABEL _RESPONSENAME
      _RESPONSELABEL _DATALABEL _DISPLAYMISSING _XVAR_OBS _Y_OBS _FREQ
      _WEIGHT _OVERALLF _OVERALLP _SHOWINSET;
    begingraph / includemissingdiscrete= _DISPLAYMISSING;
    entrytitle "Distribution of " _RESPONSENAME;
    layout overlay / yaxisopts=(griddisplay=auto_on label=_YLABEL
      shortlabel=_SHORTYLABEL) xaxisopts=(label=_XLABEL shortlabel=
      _SHORTXLABEL discreteopts=(tickvaluefitpolicy=rotatethin));
    boxplot y=_Y_OBS x=_XVAR_OBS / labelfar=true datalabel=_DATALABEL
      primary=true;
    if ((_SHOWINSET eq 1) and EXISTS(_OVERALLF))
      layout gridded / columns=2 halign=left border=true
        BackgroundColor=GraphWalls:Color Opaque=true autoalign=(
        TopRight TopLeft BottomRight BottomLeft);
      entry halign=left "F" / valign=top;
      entry halign=right eval (PUT(_OVERALLF,7.2)) / valign=top;
      entry halign=left "Prob > F" / valign=top;
      entry halign=right eval (PUT(_OVERALLP,PvalueStar.)) / valign=top;
    endlayout;
  endif;
```

PhUSE 2011

```
endlayout;
endgraph;
end;
run;
```

FINAL OUTPUT

Once the template is enabled in the SAS session, submit the code for the model. The appearance of the p-value now will contain the desirable “star” in the graphics and in the tables produced by the statistical procedure when the p-value is below the determined significance level, in this case set out at 5%.



Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Neg. Control vs. Dose 3	1	211.6000000	211.6000000	17.05	0.0033*

DELETE THE TEMPLATE

As discussed, it is good practice to delete used templates to prevent accidental unintended use. Using the TEMPLATE procedure one can delete an adjusted template by specifying the item path name. The template procedure will delete elements of the first store set with update access. Since the default template search path was changed by using the ODS PATH statement, PROC TEMPLATE will then delete templates from the store work.mystore. The template procedure never deletes a template in SASHELP.TMPLMST, so the SAS provided templates are safe and cannot be modified by the user.

```
proc template;
  delete common; ❶
run;
```

- ❶ The common option refers to the item path (folder) in the template store where the p-value template definition remains.

To delete the customized template store submit the following code:

```
proc datasets library = work nolist;
  delete mystore (memtype = itemstor);
run;
```

and set the default path afterwards typing:

PhUSE 2011

```
ods path sasuser.templat(update) sashelp.tmplmst(read);
```

CONCLUSION

The TEMPLATE procedure is a powerful tool that allows the user to customize the appearance of reports, although the way to program and to understand them without previous knowledge could seem unusual. The example presented is provided to guide the user as to the steps to start using the procedure and to manipulate the templates, in order to create new personal templates that are stored and deleted correctly. We encourage you to start using this procedure in order to create perfect reports.

REFERENCES

SAS/STAT 9.2 User's Guide, Second Edition (2009), "*The Default Template Stores and the Template Search Path*", SAS Institute Inc. SAS 9.2.

Cartier, J. (2008), "A Programmer's Introduction to the Graphics Template Language", Proceedings of the Thirty-first Annual SAS Users Group International Conference. Cary, NC. SAS Institute Inc.

Karp, Andrew H. (2006), "*Getting in to the Picture (Format)*", Proceedings of the Western Users of SAS Software Conference. Irvine, California.

ACKNOWLEDGMENTS

The authors would like to thank the department of Global Drug Discovery Statistics for their help and assistance and special thanks to Nick Downie for his review and suggestions.

CONTACT INFORMATION

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

Catalina Mejía Herrera & Volker Harm
Bayer Pharma AG.
Müllerstraße 178 S110/ 02/ 729
Berlin, 13353
Work Phone: +49 30 468 11208
Email: catalina.mejiaherrera@bayer.com, volker.harm@bayer.com