

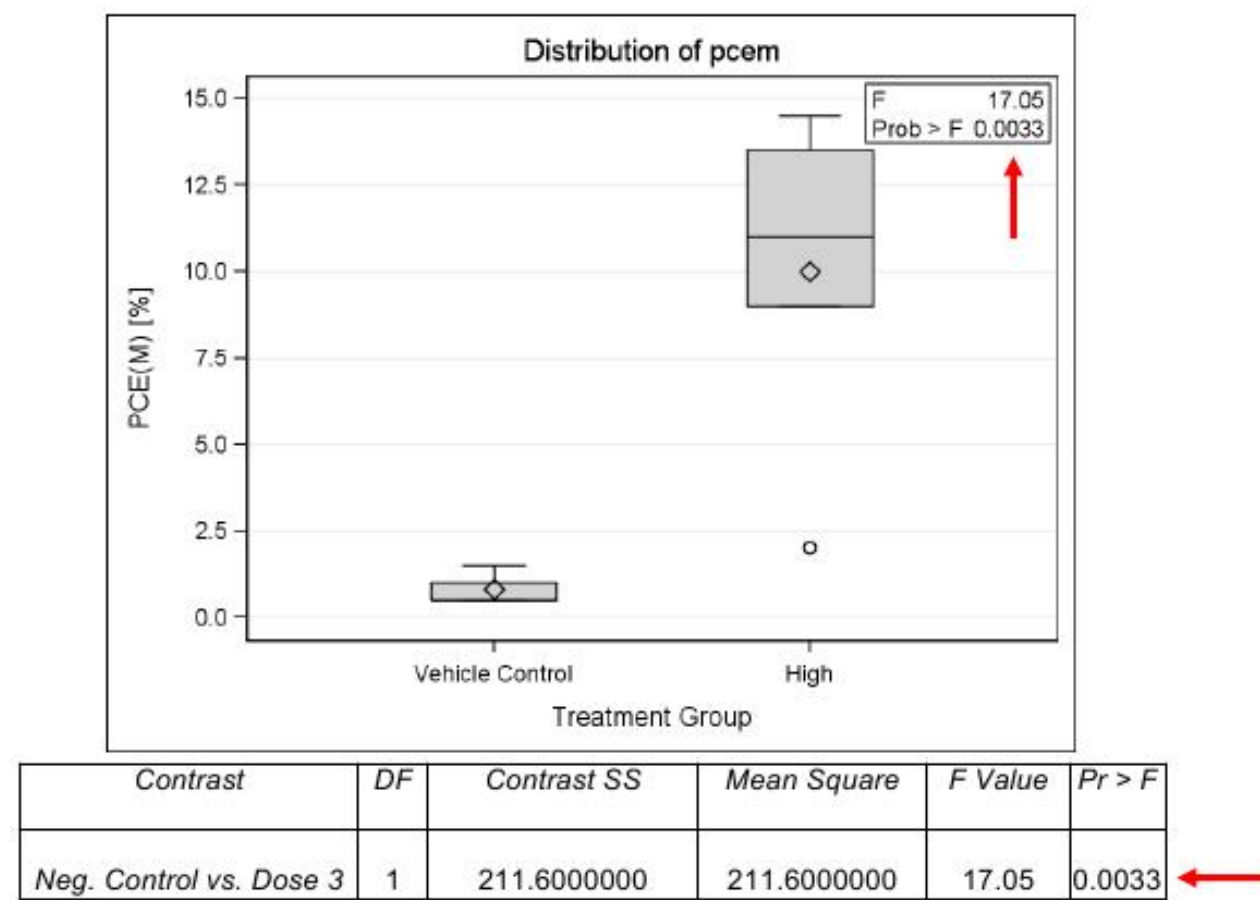
1. Introduction

Aim: Understand *SAS*® PROC TEMPLATE to customize the output of the reports.

Example: Show a p-value with a "star" in the statistical results when the p-value is smaller than the significance level.

Standard Output of PROC GLM

```
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;
```



Method: Create a format for the p-value using the PICTURE statement and plug it into the default template for the graph and table.

2. Create a PICTURE Format for the p-value

Since the values for the p-values are not known in advance, it is necessary to use a global format for all possible values. PROC FORMAT with the PICTURE statement...

- ...creates a template that is used to display the values of numeric (only) variables.
- ...does not need to know all possible values of the numeric variable to be formatted.

```
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;
```

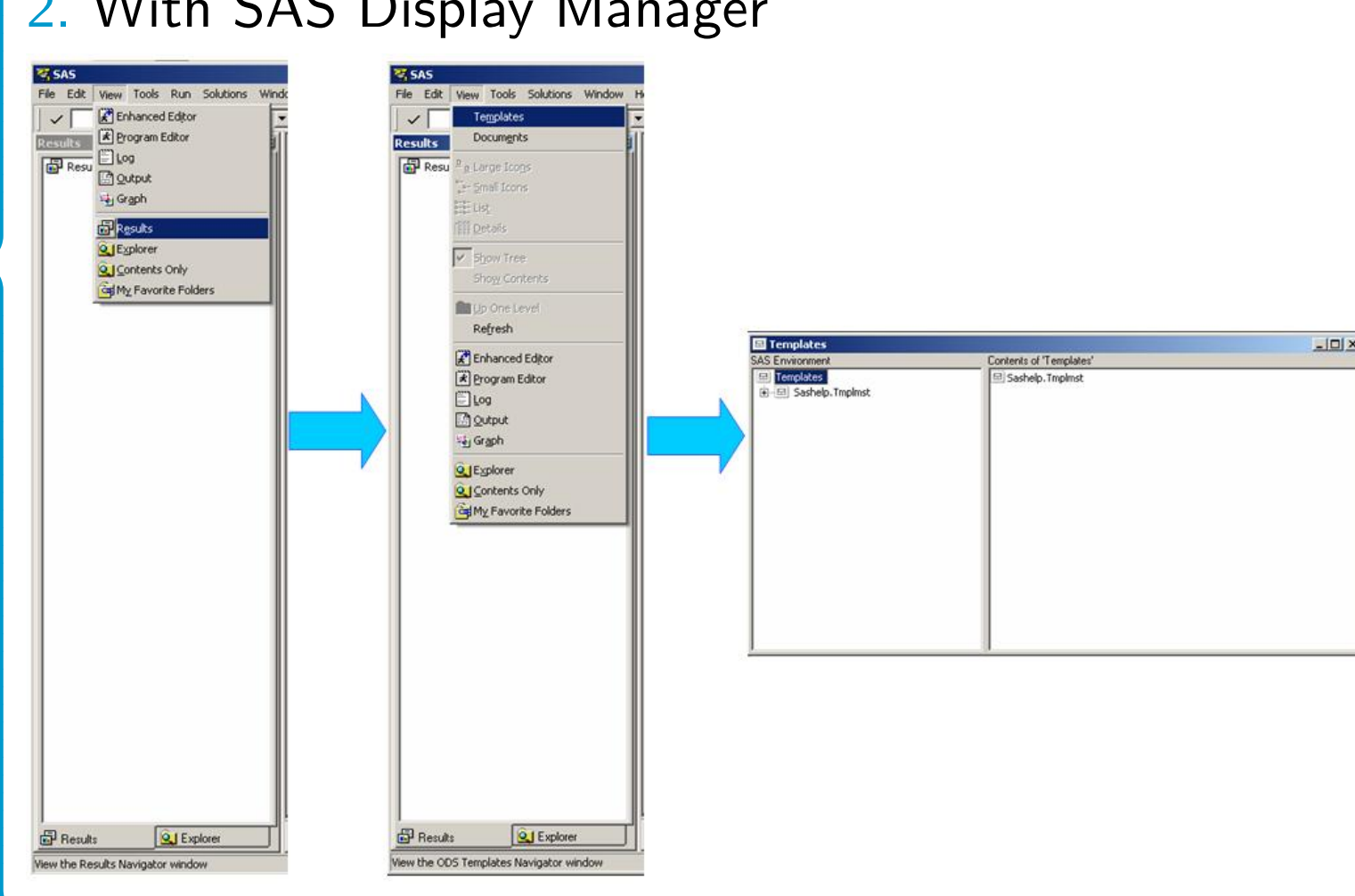
3. Find a Template for the p-value

Every output from ODS is defined by a template, so we have to find the specific template to alter.

1. TEMPLATE procedure with SOURCE statement + item path

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

2. With SAS Display Manager



4. Storing Templates

Before creating a template it is important to understand how SAS stores templates.

- Templates are stored in item stores (SAS files).

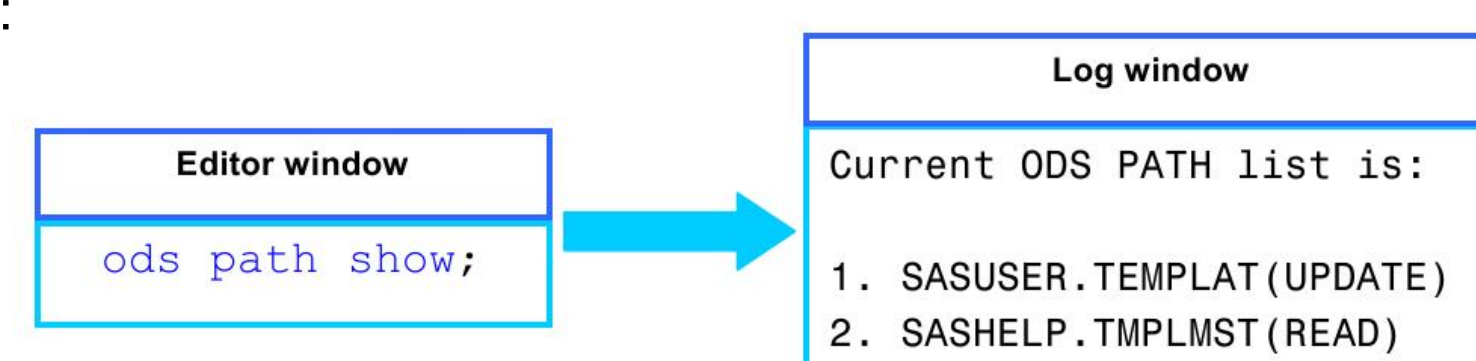
- SAS main stores are

SASUSER.TEMPLAT	{	stores user created templates.
		allows user update access.
SASHELP.TMPLMST	{	contains SAS compiled templates.
		allows user read access.

- Using ODS PATH allows to

{	identify SAS templates stores.
	read and write templates stores.
	define the search order of the templates stores.

To list the templates stores:



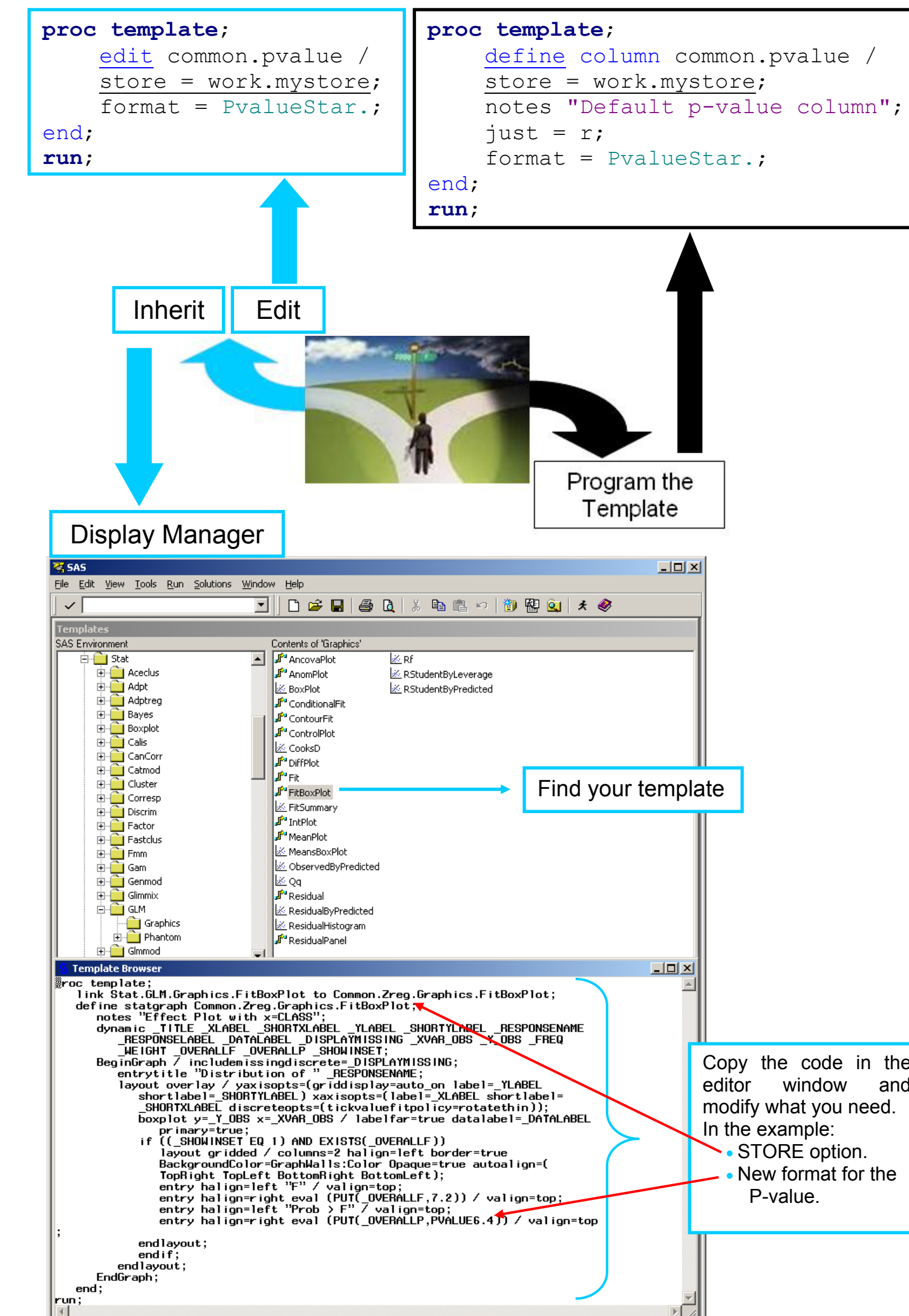
Working with your own user template store might have more advantages, because it is easier to eliminate the templates saved in your store. **It is highly recommended** to delete the customized templates once you have finished the SAS session in order to avoid that the templates are used unexpectedly. To create your template store in the work library use:

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

Now the first template store with update access is not SASUSER.TEMPLAT but work.mystore.

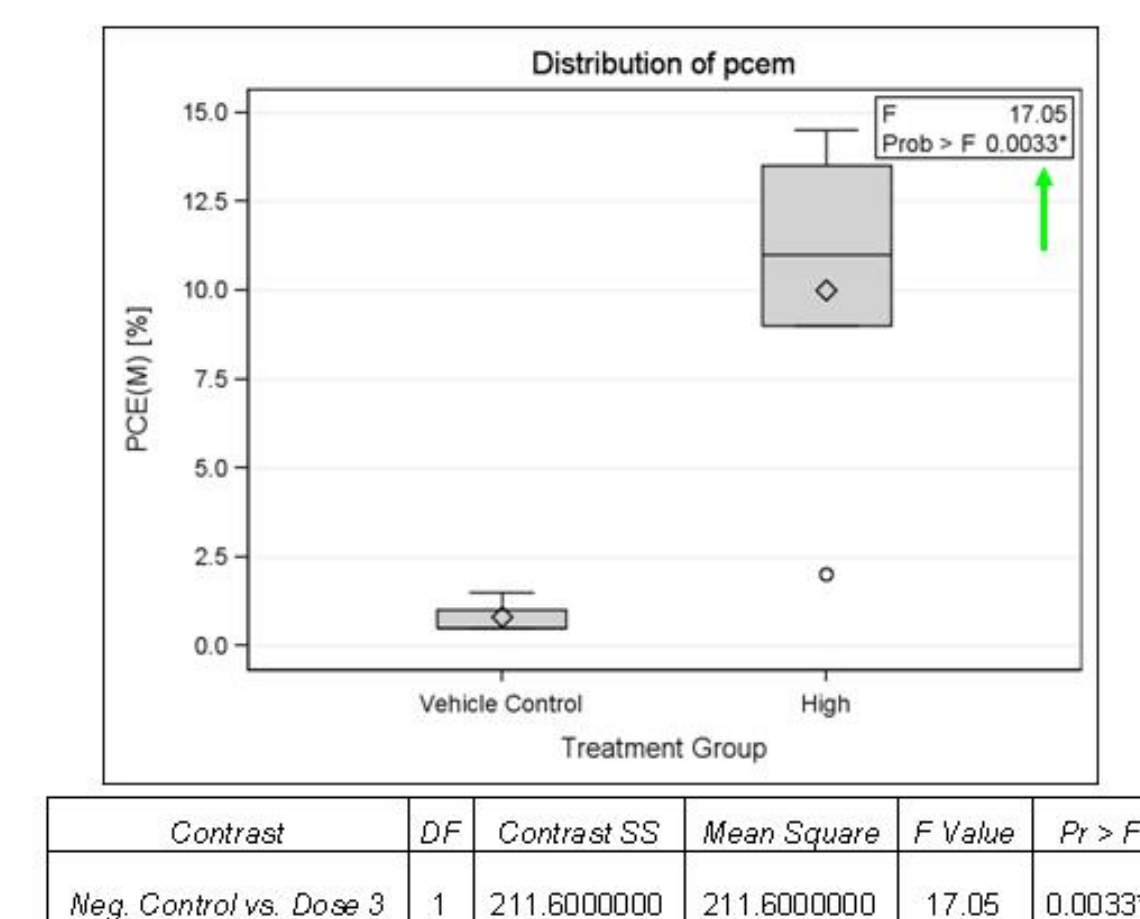
5. Create a Template

One can program, edit or inherit a template.



6. Final Output

The final output with the new format for the p-value looks like this:



7. Deleting templates

- PROC TEMPLATE eliminates templates specifying the DELETE statement and the location of the template (item path).
- The template procedure will delete elements of a store with update access, thus it will never delete a template of the SASHELP.TMPLMST store.

```
1 proc template;
  delete common;
run;

2 proc datasets library = work nolist;
  delete mystore (memtype = itemstore);
run;

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

- Deletes the folder common (which contains the customized template for the p-value) from the customized template store work.mystore, since this store is listed as the first store with update access.
- Eliminates the created template store work.mystore.
- Sets the default ODS path.

8. Conclusion

PROC TEMPLATE is a powerful tool that enables you to characterize the overall attributes of an output file such as formats, size, color and fonts. It could seem a bit challenging, but we encourage you to customize your reports by using it!

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", SUGI 31. Cary, NC. SAS Institute Inc.
- Karp, Andrew H. (2006), "Getting in to the Picture (Format)", WUSS. Irvine, California.