

SAS ODS

Greg Jenkins

Overview

- ODS stands for the Output Delivery System
- ODS allows output from the Data Step & SAS procedures to be presented in a more “useful” way.
- ODS also allows for some of the output of SAS procedures to be stored in SAS data sets.
- Although this is an improvement over the “regular” SAS output it still has its limitations.

Uses of ODS

- ODS can arrange output in a “prettier” way.
- It also can create output in a variety of formats, such as: html, pdf, rtf, etc.
- As stated earlier it can also create output datasets that generally can also be created within most of the SAS procedures.

General Idea of ODS

- Prior to version 7 of SAS output was the results of a SAS procedure and was stored in a basic output file (or output window).
- ODS now creates output objects from procedures that have basically three components: data component, table definition (order of columns, rows, etc.), and an output destination (.html, .rtf, etc.)

General Use of ODS

- To start output being delivered to ODS the general syntax is:

```
ods output-format <options>;
```

- To end output being delivered to ODS:

```
ods output-format close;
```

Example

- To send the output of a procedure to ODS and create an .html file:

```
ods html file = "body.htm"  
      contents = "contents.htm"  
      frame = "frame.htm";
```

```
proc print data =new;run;
```

```
ods html close;
```

Other .html File Options

- Other options that are provided are to change the style of the html output, just add a *style=style-type* to the opening ods statement.
- To see the SAS styles available use the following code: *proc template; list styles; run;*
- Style sheets can also be used by adding a *stylesheet=filename(url=url-name)* line to the opening ods statement.

RTF Files

- This creates “prettier” output that can be read by MS Word and other word processing programs.
- The general syntax is:

```
ods rtf file=“rtf-filename.rtf”;  
    ... sas-code ...  
ods rtf close;
```

PDF Files

- This creates “prettier” output that can be read by *Adobe Acrobat Reader*, one caveat of using this is that you need the *Adobe Acrobat Distiller*.
- The general syntax is:

```
ods pdf file="pdf-filename.pdf";  
    ... sas-code ...  
ods pdf close;
```

Enough with the Regular Output

What About Graphics?

- Works the same way as any other procedure although, the sizing of your graph probably won't be what you want, ODS fits output to 8½ x 11 (use goptions to change this).
- For html output you need to specify a path for your graphics, generally the same path as where you put your other html files, add the following line to your opening ods line:

`gpath = "path-name" (i.e. gpath="C:/sasclass/")`

Enough with the Regular Output What About Graphics?

RTF Example

```
ods rtf file="U:/rtffile.rtf";  
proc gplot data = data1;  
  plot y*x;  
run;  
ods rtf close;
```

Enough with the Regular Output What About Graphics?

HTML Example

```
ods html file="U:/web/body.htm"  
  gpath="U:/web/";  
proc gplot data = data1;  
  plot y*x;  
run;  
ods html close;
```

Some Other Options for HTML Graphics

- Another thing you can do with HTML output is create “drill down” graphs.
- This is done using the actually gplot or gchart procedure.
- A variable must be created in the SAS dataset that the gplot/gchart procedure is using, indicating the link requested.

Drill-Down Chart Example

```
Data temp; length mylink $40.;
```

```
if z = 1 then mylink = 'href="mylink1.htm"';
```

```
else if z = 2 then mylink = 'href ....
```

```
ods html file="U:/body.htm" gpath="U:/";
```

```
proc gchart;
```

```
hbar z/html=mylink;
```

```
run;
```

```
ods html close;
```

What if I Don't Want All the ODS Tables Created by a Procedure

- Use the select or exclude statements in the different ods opening statements.
- In order to exclude and select tables you need to know their names.
- The help generally gives you names of the tables but you can also use the ods trace command:

ods trace output;

Example of Selecting a Table

```
ods listing select "Moments";
```

```
proc univariate;
```

```
  var age;
```

```
run;
```

How About those Output Datasets?

- Just as in the case of using select and exclude statements you'll need to know the tables that you want to create datasets from, but the general syntax is:

ods output *table-name=dataset-name*;

Example of Output Dataset

```
ods output Moments=mom;
```

```
proc univariate;
```

```
  var age;
```

```
run;
```

```
ods output close;
```

The Output is “Nice” But How Can I Change It?

- To create your own ODS schemes, you need to create what is called a template (don't confuse this with the SAS/GRAPH templates).
- To create and modify templates you need to use the procedure `proc template`.

Proc Template

- Two parts to ODS output: an overall style, and individual table definitions.
- The overall style consists of general formats like: colors for backgrounds, default fonts, etc.
- Individual table definitions are things like: the order of columns in a table, SAS format to use for column headings, etc.

Proc Template

- Templates are stored like other SAS files, but can have multiple level names.
- The main “unit” of a template “group” is a template store.
- Within the template store may be many templates, where styles, table elements, etc. are stored. It’s basically like a directory for templates.

Proc Template

- The basic syntax used to create a template is the following:

```
proc template;
```

```
  define definition-type definition-path  
  </store=libname-template-store>;
```

Example of Style Definition

```
libname in 'U:/mytemplates';  
Proc template;  
  define style mystyle.testout  
    store=in.mystore;  
  style celldatasimple / font_face=arial  
    background=very light vivid blue  
    foreground=white;  
run;
```

Proc Template

- There are a lot of options using this procedure, it appears that you can have a lot of control over how things are formatted.
- However, it's not that simple to figure out, SAS online doc has a lot of information on this or look at Heffner 1998 *The Complete Guide to the SAS Output Delivery System*.
- Also code of default styles and table definitions can be looked at using proc template and the source command:

```
proc template; source template-name;
```