

Paper 216-2010

ODS LAYOUT to Create Publication-Quality PDF Reports of STD Surveillance Data

Rob Nelson, Centers for Disease Control and Prevention, Atlanta, GA

ABSTRACT

CDC's Division of STD Prevention turned to ODS LAYOUT and ODS REGIONS to create high-quality reports of STD morbidity data. These reports allow for the combination of text and graphics into visually stunning output. The author demonstrates the SAS® code used to create this output and discusses some of nuances of this approach.

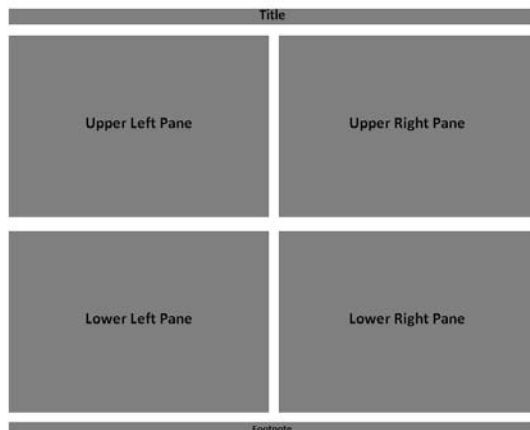
INTRODUCTION

The Division of STD Prevention at the Centers for Disease Control and Prevention (CDC) is responsible for managing and analyzing morbidity data on notifiable Sexually Transmitted Diseases, including chlamydia, gonorrhea, and syphilis, reported to CDC through state health departments. The Division needs to create updateable, publication quality reports to present these data to stakeholders in STD prevention. SAS ODS in version 9.2 provides the perfect tool for creating these products.

ODS LAYOUT and ODS REGION were implemented in version 9 and provide precise control over the placement of SAS output in PDF files. This output can be further enhanced with ODS PDF TEXT, which allows for placement of text and graphics on a PDF without running a SAS procedure, and inline formatting with ODS ESCAPECHAR, which can be used to control the formatting of ODS PDF TEXT text and graphics. In this paper, we will demonstrate how state-level profiles were created to display combined chlamydia, gonorrhea, and P&S syphilis data.

AN APPLICATION OF ODS LAYOUT AND ODS REGION

ODS LAYOUT and ODS REGION allow the user to define regions on a page and place SAS output into the regions. While different approaches can be used, the author suggests that that SAS programmers who plan to use ODS LAYOUT start by designing the workspace. The following image shows a conceptualization of a report layout.

**Figure 1**

The six regions shown in this report are the title, footnote, and four panes for the body of the document. ODS REGION statements should be preceded by an ODS LAYOUT statement that defines the page. Each of these regions is then defined with an ODS REGION statement, as shown below in Example 1.

```
ods layout start width=11in height=8.5in;
* title;
ods region width=11in height=.75in x=-.2in y=0in;
* footer;
ods region width=9in height=.5in x=.6in y=7.8in;
* top left pane;
ods region width=4.5in height=3in x=.5in y=1in;
*top right pane;
ods region width=4.5in height=3in x=5.25in y=1in;
*bottom left pane;
ods region width=4.5in height=3in x=.5in y=4.75in;
*bottom right pane;
ods region width=4.5in height=3in x=5.25in y=4.8in;
ods layout end;
```

Example 1

The ODS REGION statement provides the region's dimensions, e.g., `width=4.5in height=3in`, as well as the region's exact position on the page. This example creates a region 4.5 inches wide and 3 inches tall. Units of measure, such as inches, shown in the example, or percent can be used. The ODS REGION statement also provides the region's starting place. The origin, `x=0` and `y=0`, is the top left corner of the area defined in ODS LAYOUT. In the last region created in Example 1, these coordinates are `x=5.25in y=4.8in`, which positions the figure 5.25 inches from the left and 4.8 inches from the top. As with the region's dimensions, the region's `x` and `y` coordinate start point can be provided in either units of measure or percent. Note that the diagram displayed in Figure 1 can be easily drawn on paper in order to write the coordinates on the corresponding region.

The SAS Programmer can test these regions by running a simple SAS procedure, such as PROC FREQ, to see where the output is placed. In this case, the author decided to include four extra regions for pane titles, as well as SAS procedure code to produce output with the following code. This allows for the use of ODS PDF TEXT statements to place the main title and pane title text directly onto the PDF. Due to space considerations, only the code to generate the first map output pane (top left) of three is shown in Example 2.

```
ods pdf file="\\test\SGF\SGF STD Maps.pdf" startpage=never;
ods layout start width=11in height=8.5in;
ods region width=11in height=.75in x=-.2in y=0in;
ods pdf text="Rates of Reportable STDs among Young People 15 - 24 Years of Age, Alabama, 2007";

ods region width=9in height=.5in x=.6in y=7.8in;
ods pdf text="Data Source: National Electronic Telecommunications System for Surveillance,
Centers for Disease Control and Prevention. CDC/NCHS 2007 bridged population estimates, derived
from US Census data, used for rate denominators.";

ods region width=4.5in height=.22in x=.5in y=.78in;
ods pdf text="Chlamydia Rates by County";

ods region width=4.5in height=3in x=.5in y=1in;
legend label=(position=top j=center h=1 position=(top) color=black
"Rate (per 100,000 population)") position=(bottom center outside)
shape=bar (6,1) value=(c=black j=1 h=1)
across=2 cborder=white;
proc gmap data=datacnty map=map all;
id state county;
choro groupct / discrete legend=legend1 outline=black;
format groupct groupct.;
run;
quit;

ods layout end;
ods pdf close;
```

Example 2

The resulting output, shown in figure 2, has the correct layout. These maps were created with PROC GMAP and formatted with SAS/GRAPH. The maps are visually appealing and easy to interpret. The text, however, including the main title, maps titles, and footnote, is bland and lacks important formatting.



Figure 2

Fortunately, inline formatting allows excellent control over font and placement of ODS PDF TEXT output. This is invoked with a special character, defined in an ODS ESCAPECHAR statement. Example 3 demonstrates the use of inline formatting to change the font, font size, font weight, and justification of ODS PDF TEXT.

```
ods escapechar='^';
ods pdf text="^S={font=('Arial',13pt,bold) just=center}Chlamydia Rates by County";
```

Example 3

In addition to formatting text, the document can be further developed by populating the empty lower right pane using PROC TABULATE. Example 4 shows the code used to add a table to this fourth main pane. The example uses inline formatting within PROC TABULATE to customize font and table background color.

```
ods region width=4.8in height=.25in x=5.25in y=4.53in;
ods pdf text="^S={font=('Arial',13pt,bold) just=center}Cases and Rates per 100,000";
ods region width=4.5in height=3in x=5.25in y=4.8in;
proc tabulate data=mapdir.staterates style={font=('Arial',10pt,normal)};
  class disease / style=[background=white font=('Arial',10pt,bold)];
  var cases rate / style=[background=white font=('Arial',10pt,bold)];
  classlev disease / style=[background=white font=('Arial',10pt,bold)];
  table disease=" ", all=" "*(cases="Cases"*sum=" "*f=8.0
  rate="Rate"*sum=" "*f=8.1) / rts=15 box=[style=[background=white]];
  format disease dxnm.;
run;
```

Example 4

The output, shown below in Figure 3, is quite rich, and concisely presents that data for a given state. However, these data don't provide any context. The addition of data at the national level will add this missing piece. Similar contextual information could be provided at the regional level, for bordering states, or for any other appropriate slice of the data.

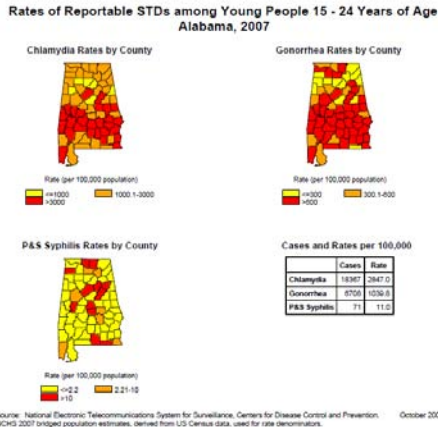


Figure 3

Rather than a table, macro programming can be used to develop automated narrative to create more complex output. The following example, shown in figures 8 and 9, demonstrates how to use macro variables to populate text with summary information from the national level. In this case, unshown data steps with CALL SYMPUT statements were used to read a data set with national rates and create the macro variable usctrate, with the national chlamydia rate, and to read state level data with rates, number of cases, and ranks, to create the macro variables ctcase, ctrate, ctcasek, and ctraterk.

```
ods pdf text="^S={font=('Arial',11pt,bold) just=left}Chlamydia: ^S={font=('Arial',11pt,normal) just=left}%trim(&statenm.) reported %trim(&ctcase.) case(s) of chlamydia, for a rate of %trim(%sysfunc(putn(&ctrate.,6.1))) per 100,000. The state rate ranked %trim(&ctraterk.) and number of cases ranked %trim(&ctcasek.) among 50 states and DC. The national rate for 2007 was %trim(%sysfunc(putn(&usctrate.,6.1))) per 100,000.";
ods pdf text="^S={font=('Arial',11pt,bold) just=left} ";
```

Example 5

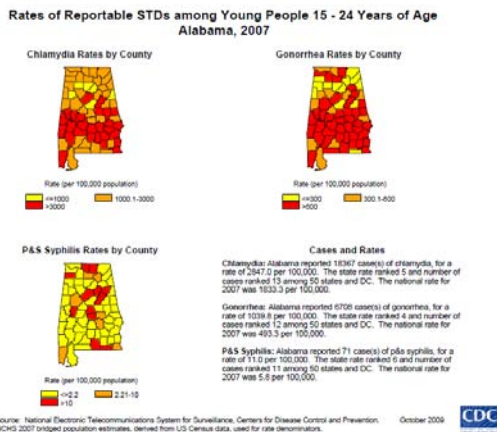


Figure 4

The last figure shown, Figure 5, is the Alabama page from the annual Syphilis State Profiles. The code to create the figure is beyond the scope of this article. The figure is shown to illustrate the use of complex conditional programming to tailor the narrative to different situations. In this case, different text is supplied depending on the number of cases. Similar methods could be used to provide different narrative for decreasing rates versus increasing rates.

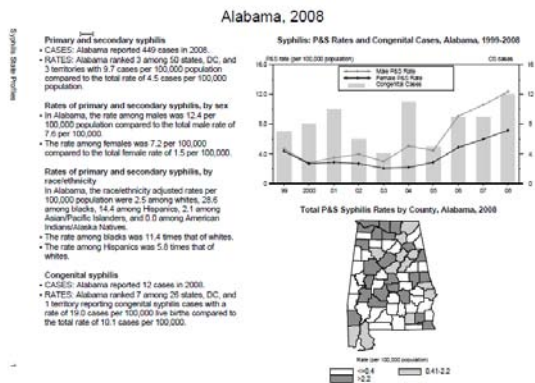


Figure 5

CONCLUSION

The profiles featured in this article were previously created through a tedious process of post processing images created with SAS into Word documents. The enhanced ODS features available in SAS version 9 have allowed the Division of STD Prevention to create SAS programs to handle the entire process, from summarization of the data to the create of polished, publication-quality PDF output. In particular, the latest updates to ODS LAYOUT and ODS REGIONS that were made available in SAS version 9.2 phase 2 provide SAS Users with an easy to use palette on which to combine text and graphics.

CONTACT INFORMATION

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

Name: Rob Nelson
 E-mail: RNelson1@cdc.gov

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.