

CHAPTER 5

Controlling Titles

Purpose: This chapter demonstrates how to control various characteristics of the titles in your graphs.

The Basics

For the first examples in this chapter, we'll once again use the `sashelp.class` dataset (which ships with SAS). Here's what the first few rows of the data look like:

Obs	Name	Sex	Age	Height	Weight
1	William	M	15	66.5	112.0
2	Thomas	M	11	57.5	85.0
3	Ronald	M	15	67.0	133.0
4	Robert	M	12	64.8	128.0
5	Philip	M	16	72.0	150.0

In SAS/GRAPH, you can use title statements to add a primary, and secondary title(s). Let's start with a simple graph that has a `title1` and `title2`, using the default title characteristics. Notice that the `title2` is much smaller than the `title1`, and uses the same font and size as the text in the axes of the graph. Also, the title is very close to the top edge of the graph (I have turned on a light gray border around the graph in this document so you can see where the 'edge' of the graph is.)

```
axis1 order=(50 to 75 by 5) minor=none offset=(4,4);  
axis2 order=(50 to 150 by 25) minor=none offset=(4,4);
```

```

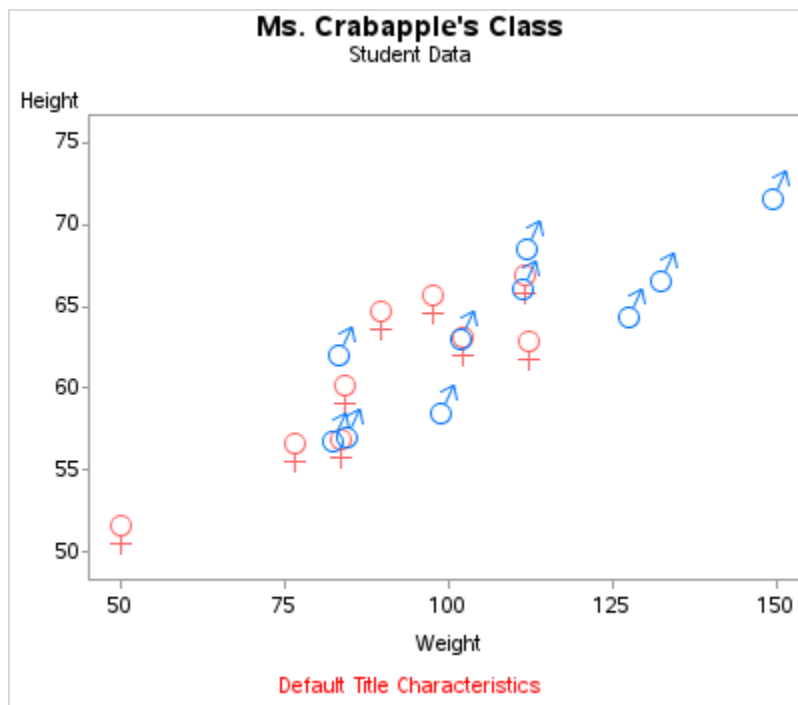
symbol1 font='Albany amt/unicode' value='2640'x h=6
      interpol=none color=cxFF6666; /* female */
symbol2 font='Albany amt/unicode' value='2642'x h=6
      interpol=none color=cx0276FD; /* male */

```

```

title1 "Ms. Crabapple's Class";
title2 "Student Data";
proc gplot data=sashelp.class;
plot height*weight=sex / nolegend
      vaxis=axis1 haxis=axis2;
run;

```



I almost always use the title statement's linespacing option (`ls=`) to add some white-space above the title. Also, I frequently make the `title2` a bit larger than the default text size, using the `height` option (`h=`), and change the font to bold (like the `title1`).

```

axis1 order=(50 to 75 by 5) minor=none offset=(4,4);
axis2 order=(50 to 150 by 25) minor=none offset=(4,4);

```

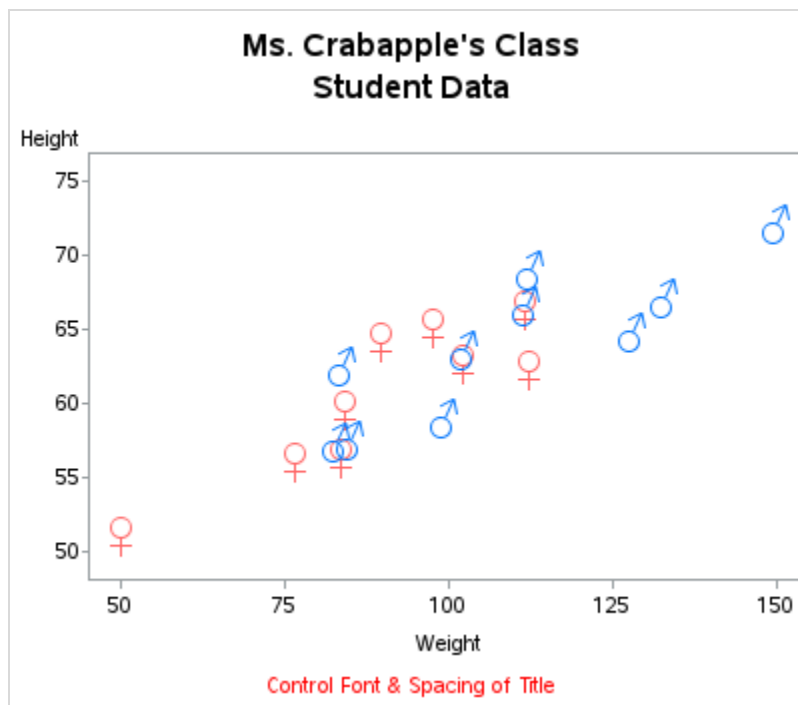
```

symbol1 font='Albany amt/unicode' value='2640'x h=6
      interpol=none color=cxFF6666; /* female */
symbol2 font='Albany amt/unicode' value='2642'x h=6
      interpol=none color=cx0276FD; /* male */

title1 ls=2.5 h=4.5 f='albany amt/bold'
      "Ms. Crabapple's Class";
title2 ls=1.0 h=4.5 f='albany amt/bold'
      "Student Data";

proc gplot data=sashelp.class;
plot height*weight=sex / nolegend
      vaxis=axis1 haxis=axis2;
run;

```



I usually go with centered titles (as is the default), but sometimes it works out better for the titles to be left-justified. You can easily control this using the title statement's justify (j=) option. We'll do this in two steps – this first graph shows you what just using the j=l option gets you (titles jammed up against the left border), and then I'll show you how to add some white-space in the next graph.

```

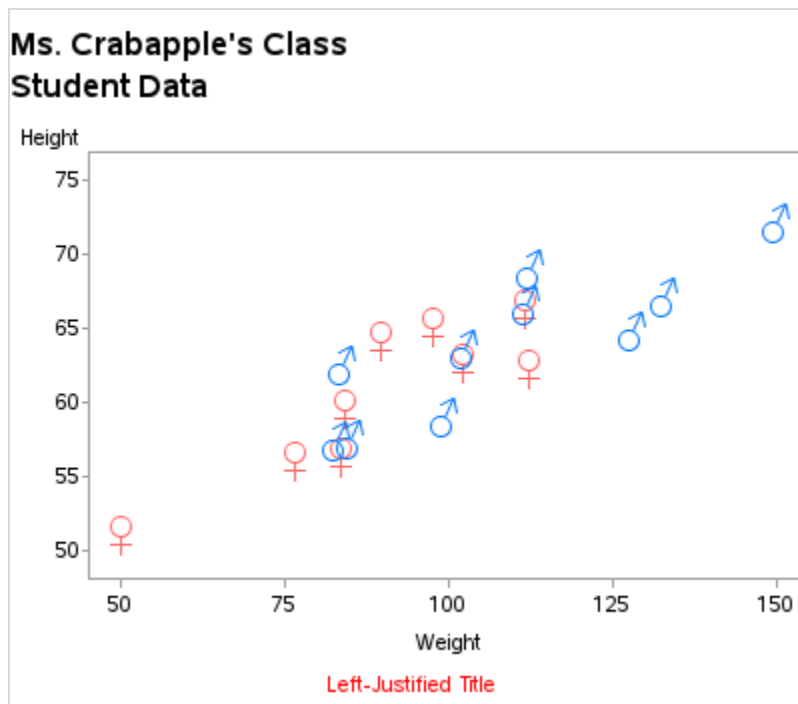
axis1 order=(50 to 75 by 5) minor=none offset=(4,4);
axis2 order=(50 to 150 by 25) minor=none offset=(4,4);

symbol1 font='Albany amt/unicode' value='2640'x h=6
  interpol=none color=cxFF6666; /* female */
symbol2 font='Albany amt/unicode' value='2642'x h=6
  interpol=none color=cx0276FD; /* male */

title1 ls=2.5 j=1 h=4.5 f='albany amt/bold'
  "Ms. Crabapple's Class";
title2 ls=1.0 j=1 h=4.5 f='albany amt/bold'
  "Student Data";

proc gplot data=sashelp.class;
plot height*weight=sex / nolegend
  vaxis=axis1 haxis=axis2;
run;

```



Now let's move the titles to the right a little, so they line up with the left axis. You can accomplish this using the title statement's `move=` option. It usually takes a bit of trial-

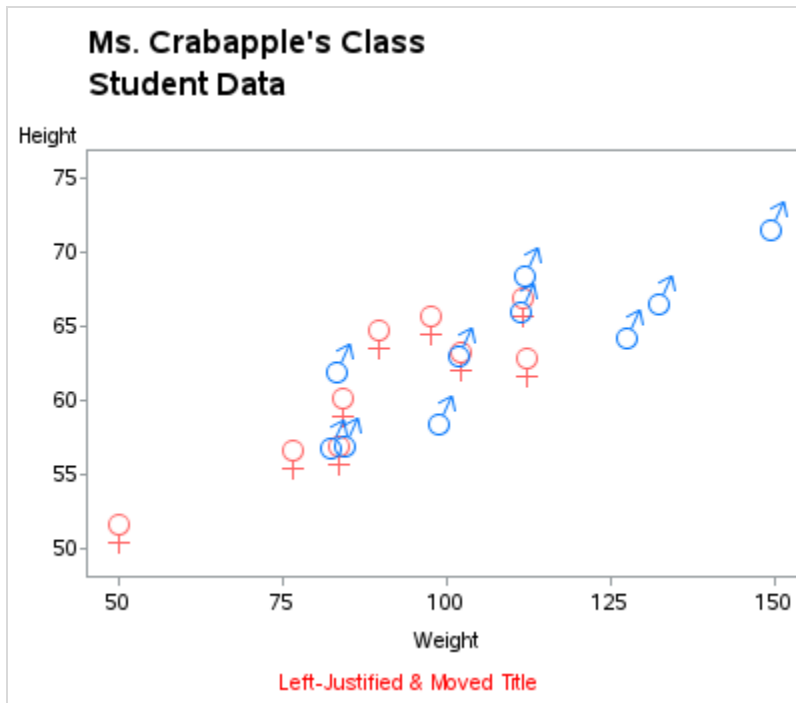
and-error to find the numeric value that positions the text exactly where you want it – in this case “+10” in the x direction worked out nicely.

```
axis1 order=(50 to 75 by 5) minor=none offset=(4,4);
axis2 order=(50 to 150 by 25) minor=none offset=(4,4);

symbol1 font='Albany amt/unicode' value='2640'x h=6
  interpol=none color=cxFF6666; /* female */
symbol2 font='Albany amt/unicode' value='2642'x h=6
  interpol=none color=cx0276FD; /* male */

title1 ls=2.5 j=1 move=(+10,+0) h=4.5 f='albany amt/bold'
  "Ms. Crabapple's Class";
title2 ls=1.0 j=1 move=(+10,+0) h=4.5 f='albany amt/bold'
  "Student Data";

proc gplot data=sashelp.class;
plot height*weight=sex / nolegend
  vaxis=axis1 haxis=axis2;
run;
```



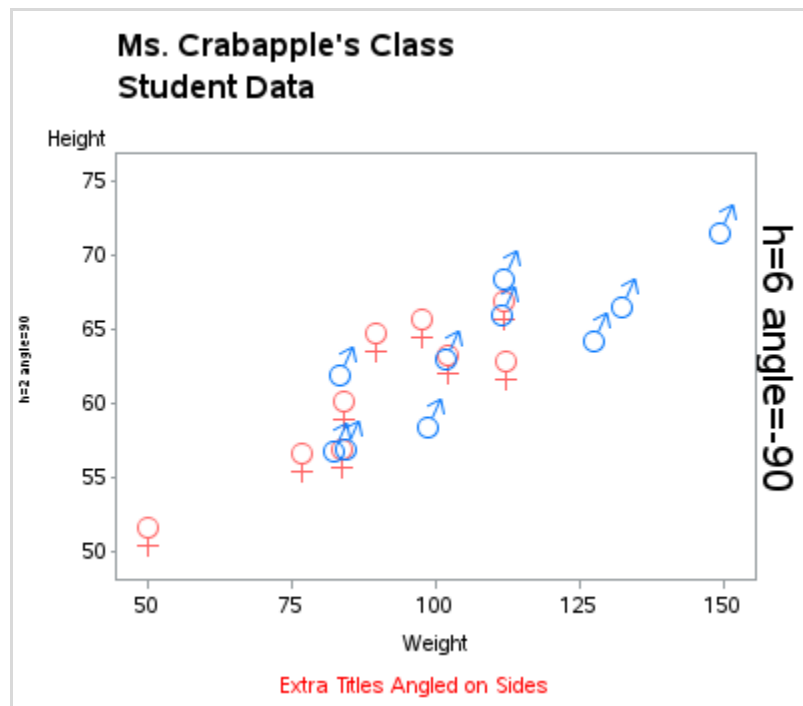
Now that we have the desired amount of white-space above the title, and to the left of the title, how about controlling the white-space around the rest of the graph? We can actually control that with some additional title statements! Titles can be angled so that they appear along the left and right sides of the graph, which will add more space between the right axis wall and the right edge of the graph space, for example.

```
axis1 order=(50 to 75 by 5) minor=none offset=(4,4);
axis2 order=(50 to 150 by 25) minor=none offset=(4,4);

symbol1 font='Albany amt/unicode' value='2640'x h=6
      interpol=none color=cxFF6666; /* female */
symbol2 font='Albany amt/unicode' value='2642'x h=6
      interpol=none color=cx0276FD; /* male */

title1 ls=2.5 j=1 move=(+13,+0) h=4.5 f='albany amt/bold'
      "Ms. Crabapple's Class";
title2 ls=1.0 j=1 move=(+13,+0) h=4.5 f='albany amt/bold'
      "Student Data";
title3 a=90 h=2 "h=2 angle=90";
title4 a=-90 h=6 "h=6 angle=-90";

proc gplot data=sashelp.class;
plot height*weight=sex / nolegend
      vaxis=axis1 haxis=axis2;
run;
```



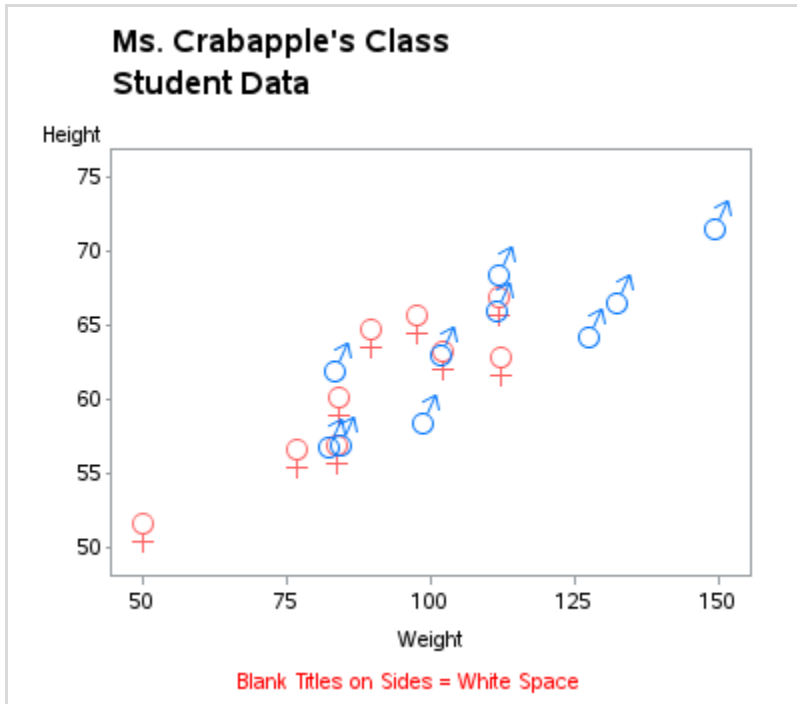
In the example above, I left some text in the title3 and title4 statements, so you could see where these titles showed up in the graph. That was “for demonstration purposes only.” In real life, you would make those titles ‘blank’ so that white-space is all that is visible on the left and right. Now the resulting graph is pleasantly spaced on all sides, earning a “check plus” ☺

```
axis1 order=(50 to 75 by 5) minor=none offset=(4,4);
axis2 order=(50 to 150 by 25) minor=none offset=(4,4);
```

```
symbol1 font='Albany amt/unicode' value='2640'x h=6
  interpol=none color=cxFF6666; /* female */
symbol2 font='Albany amt/unicode' value='2642'x h=6
  interpol=none color=cx0276FD; /* male */
```

```
title1 ls=2.5 j=1 move=(+13,+0) h=4.5 f='albany amt/bold'
  "Ms. Crabapple's Class";
title2 ls=1.0 j=1 move=(+13,+0) h=4.5 f='albany amt/bold'
  "Student Data";
title3 a=90 h=2 " ";
title4 a=-90 h=6 " ";
```

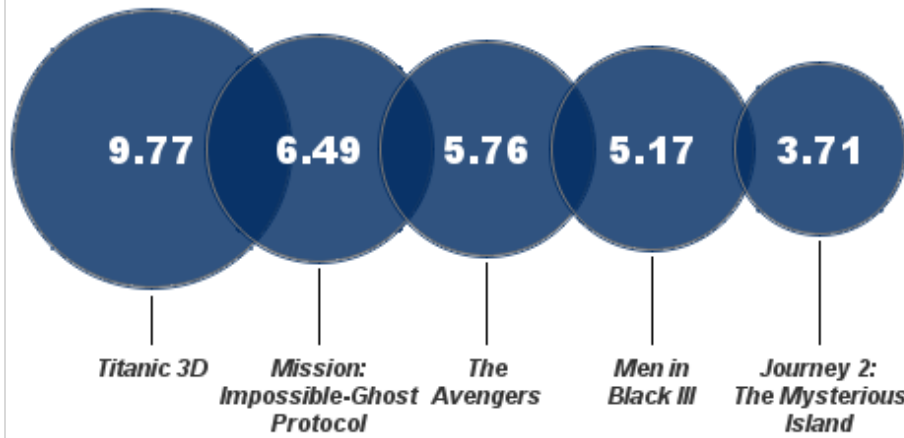
```
proc gplot data=sashelp.class;  
plot height*weight=sex / nolegend  
vaxis=axis1 haxis=axis2;  
run;
```



Below is a fancier example of titles that use the move= option for better placement:

TOP 5 MOVIES BY BOX OFFICE REVENUE IN THE FIRST HALF OF 2012

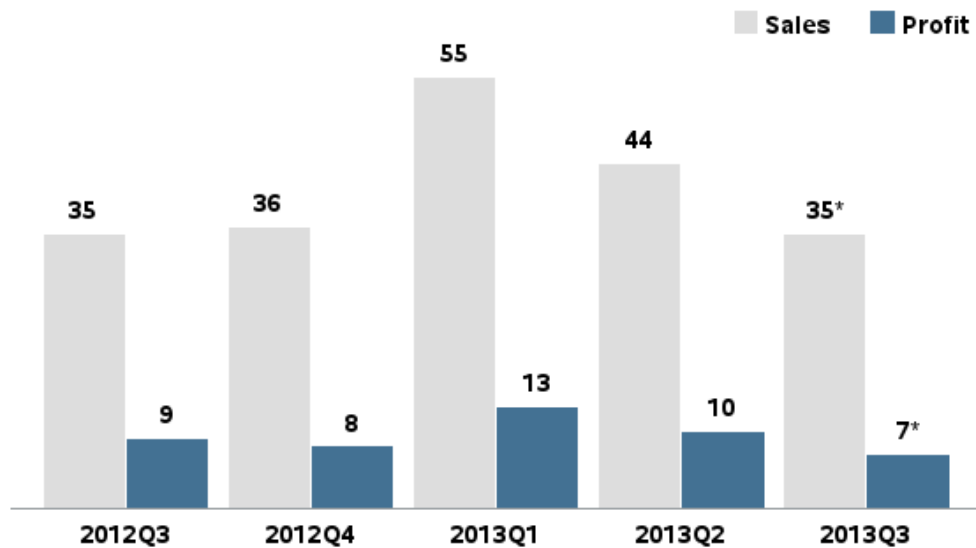
Unit: 100 million yuan



Source: EntGroup Consulting Group

ZHANG / CHINA DAILY

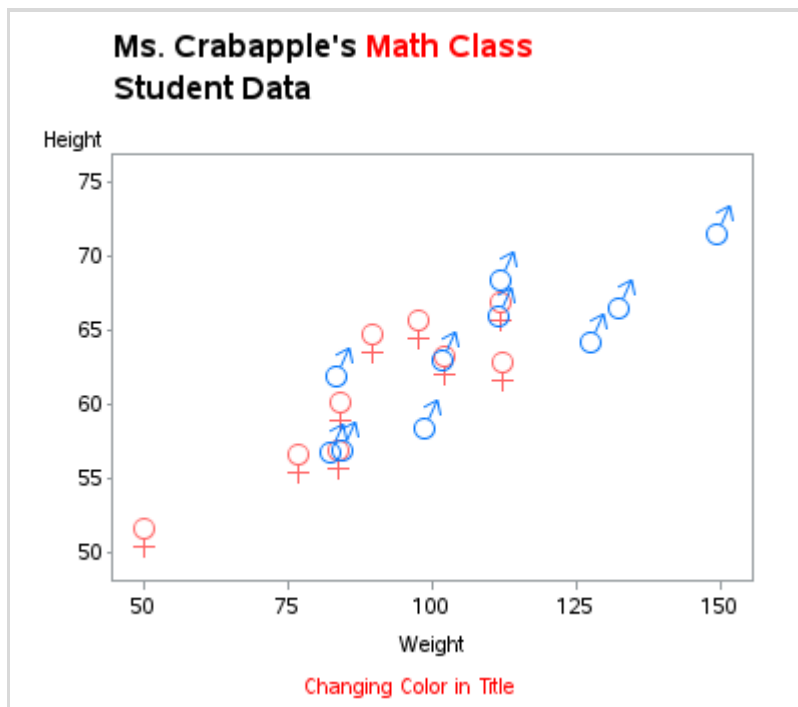
Apple sales and profit (in billion \$US)



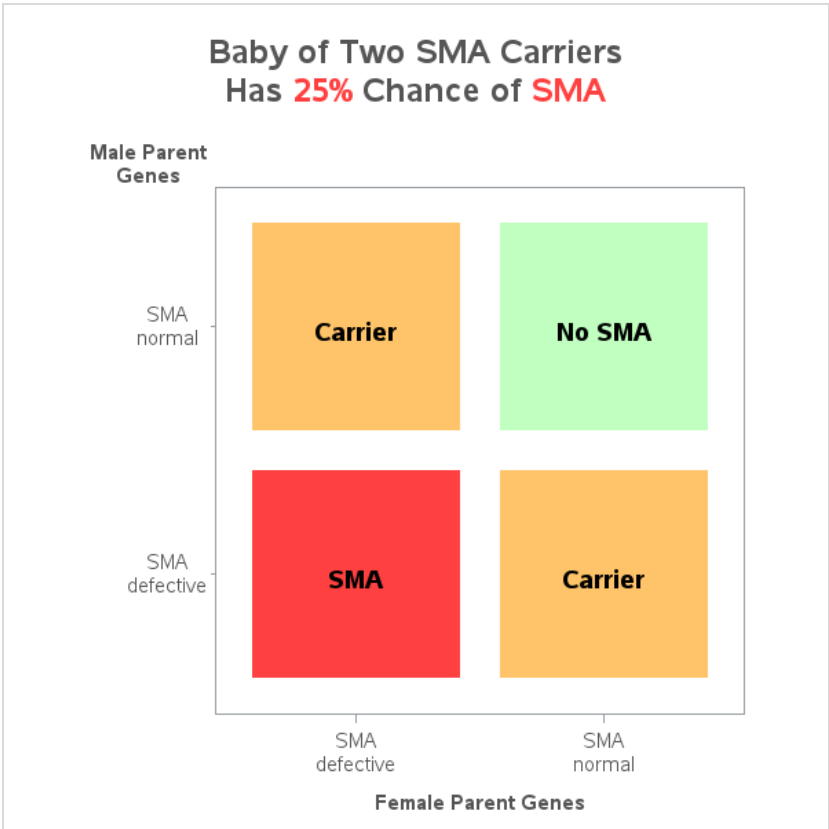
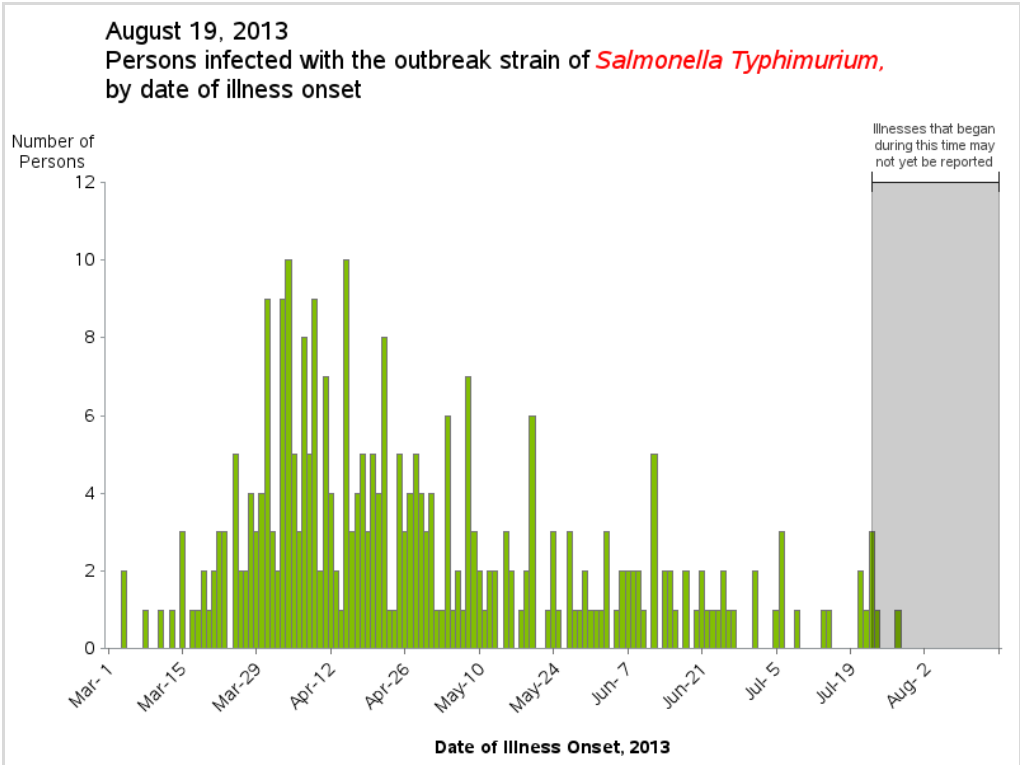
* Consensus forecast of analysis polled by Thomson Reuters; Source: Apple, Thomson Reuters

You might sometimes want some text to stand out in a title, and one way to do that is by using a different color. This is especially useful when you're creating a series of similar graphs, and you can highlight the thing that's changing in the title. All you have to do is break the title into separate pieces of text, and specify a `color=` between them. Here's how you could add color to the title of the previous graph:

```
title1 ls=2.5 j=1 move=(+13,+0) h=4.5 f='albany amt/bold'  
      "Ms. Crabapple's " c=red "Math Class";
```



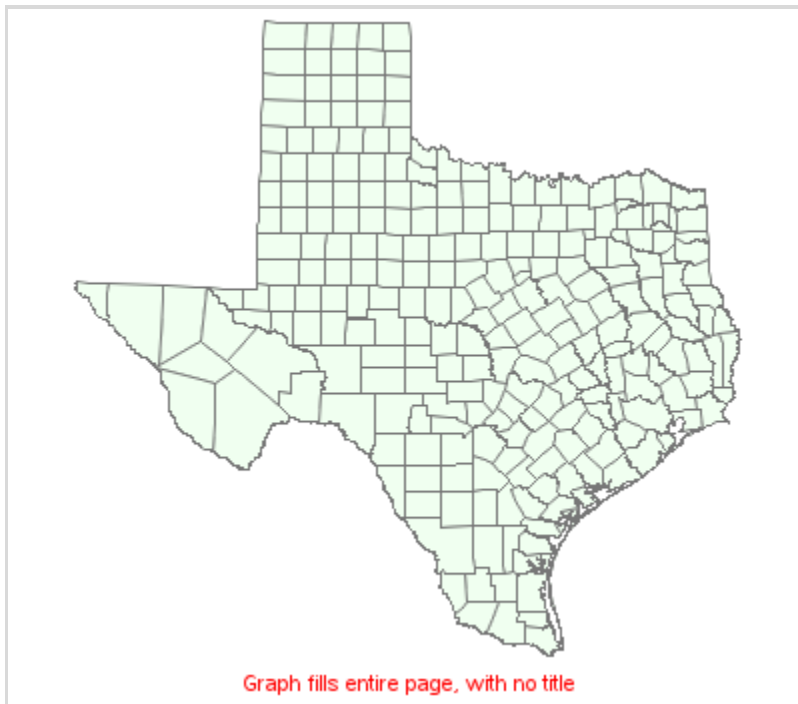
Below are some fancier examples of titles with multiple colors:



In order to utilize the space more efficiently, or provide an aesthetically balanced graph, you might sometimes want to have the title and the graph share the same space. These next few examples demonstrate some techniques you can use to do that.

We'll use a geographical map of Texas to demonstrate, but you can use the same techniques with any type of graph that has a nice chunk of unused space. Here's a map of Texas – notice that when there is no title, it sizes itself to utilize all available space:

```
title;
pattern1 v=s c=cxF0FFF0;
proc gmap data=my_map map=my_map;
id state county;
choro state / levels=1 nolegend outline=gray77;
run;
```



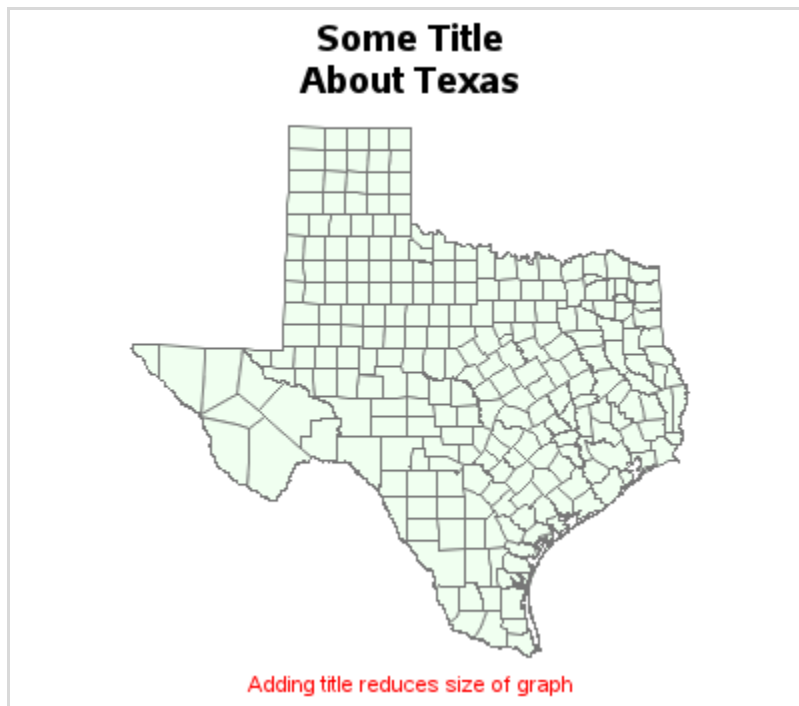
By default the titles are at the top of the page, and the graph is automatically resized to accommodate the title text. This example has two lines of title text at the top of the page, and the Texas map is now smaller than it was without the title. It's not a bad-looking map, but there is a good bit of wasted white-space:

```
title1 ls=1.5 "Some Title";
```

```

title2 f="albany amt/bold" h=5 "About Texas";
pattern1 v=s c=cxF0FFF0;
proc gmap data=my_map map=my_map;
id state county;
choro state / levels=1 nolegend coutline=gray77;
run;

```



Wouldn't it be great if you could move the titles into the white-space at the top-right area of the map, rather than above the map? If you're a "thinking person" you might be saying to yourself ... "Hmm, perhaps I can use the **move=** command that we just learned about, to move the title into that location!" OK – let's give that a try...

```

title1 ls=1.5 move=(+20,-4) "Some Title";
title2 f="albany amt/bold" h=5 move=(+20,+0) "About Texas";
pattern1 v=s c=cxF0FFF0;
proc gmap data=my_map map=my_map;
id state county;
choro state / levels=1 nolegend coutline=gray77;

```

```
run;
```

Well **that** didn't turn out so great, eh?... The titles are indeed moved to the right and down, but the map is still totally below the titles, and it has shrunk to an even smaller size, and there is now even more white-space! So basically, the titles and the graph do not mix – they do not share the same space. If you move the titles down, the graph gets smaller. Bummer ☹



Luckily there are always several different ways to do things in SAS/Graph ... so let's explore some alternate approaches.

One approach is to use a **note** statement, rather than a title statement. With a note statement, you can specify any location to place the text, and the text can even share space with the graph.

```
title;  
pattern1 v=s c=cxF0FFF0;  
proc gmap data=my_map map=my_map;  
note move=(62,90.5) f="albany amt/bold" h=5 "Some Title";  
note move=(60,84) f="albany amt/bold" h=5 "About Texas";
```

```

id state county;
choro state / levels=1 nolegend outline=gray77;
run;

```



Another approach is to use **annotate** to place the text in the desired area. Annotate is a little more work (you have to create an annotate dataset), but it provides a lot more flexibility and control than the note statement.

```

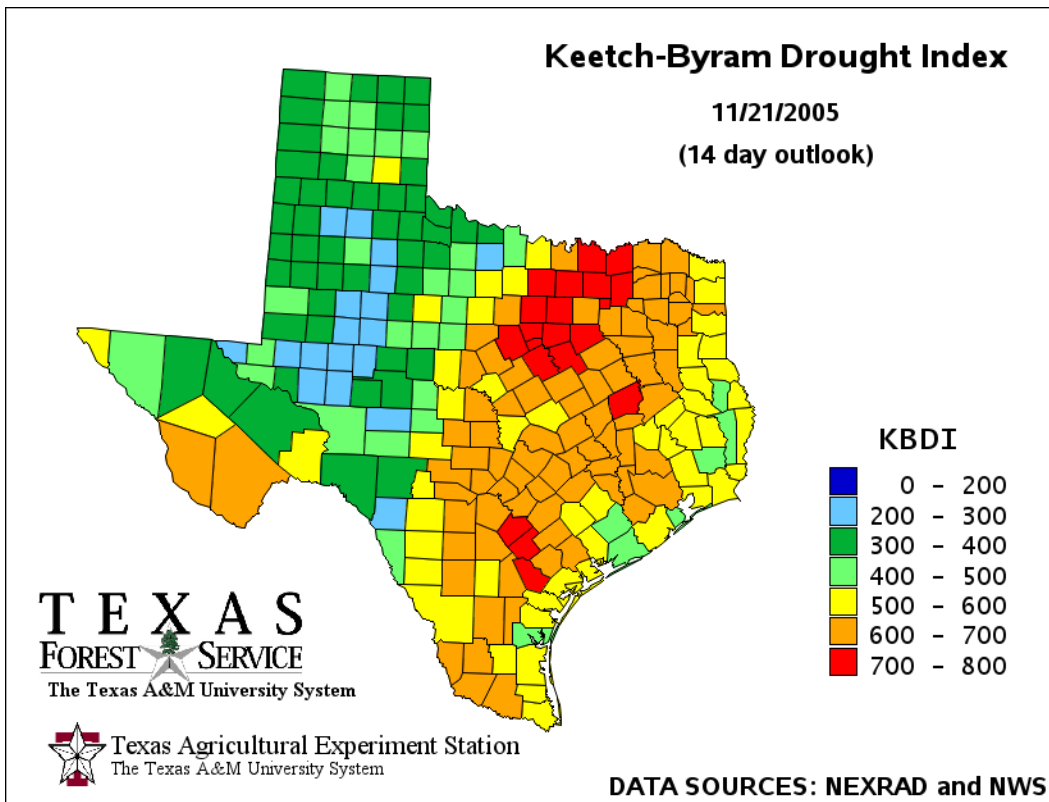
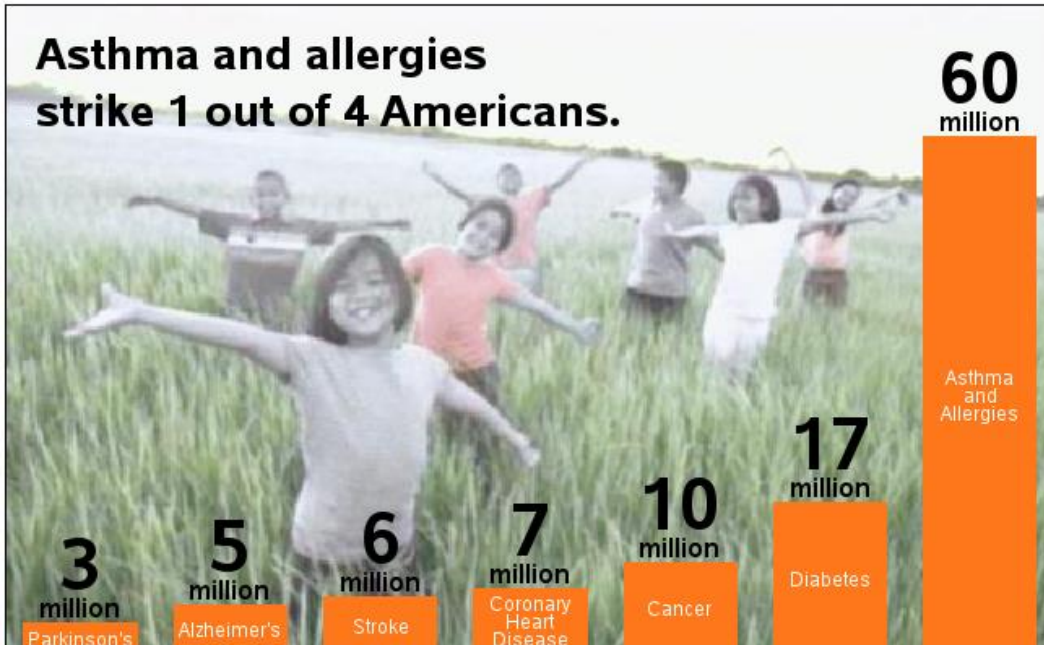
data anno_labels;
length text $100;
xsys='3'; ysys='3'; hsys='3'; when='a';
function='label'; style='ablany amt/bold';
size=5; position='5';
x=75;
y=92.0; text='Some Title'; output;
y=y-6.5; text='About Texas'; output;

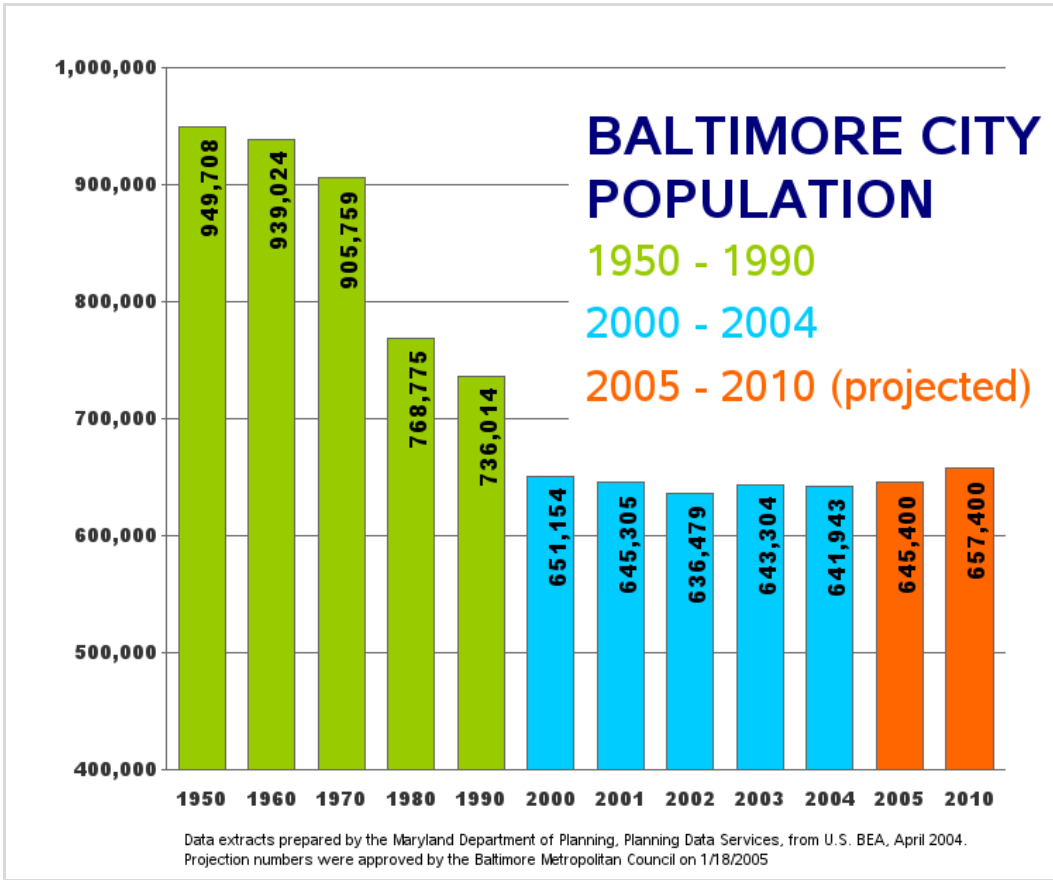
```

```
run;  
  
title;  
pattern1 v=s c=cxF0FFF0;  
proc gmap data=my_map map=my_map anno=anno labels;  
id state county;  
choro state / levels=1 nolegend outline=gray77;  
run;
```



Below are several fancier examples where the title shares the space with the graph:





Welcome to NESUG!

nesug
NorthEast SAS®
Users Group, Inc.

