

CHAPTER 3

Splitting Axis Text

Purpose: This chapter describes techniques you can use to split long text into two (or more) lines in your axes. Two techniques are described – one to split text in axis tick mark values, and another to split text in axis labels.

Splitting Text in Axis Tick Mark Values

This first example shows how to split the text for long axis tick mark values into multiple lines, using the axis **split=** option. In this example we'll use a bar chart, but you can similarly use this technique with other SAS/Graph charts and graphs as well.

To make things simple, we'll use some custom data for this example. Here's the data step code to create the data:

```
data my_data;
input widgets name $ 4-50;
datalines;
5 Robert Allison
8 Ed Odom
12 Mark Fletcher
10 Jeff Phillips
6 John Davis
13 Armistead Sapp
;
run;
```

And a snapshot of the data it creates:

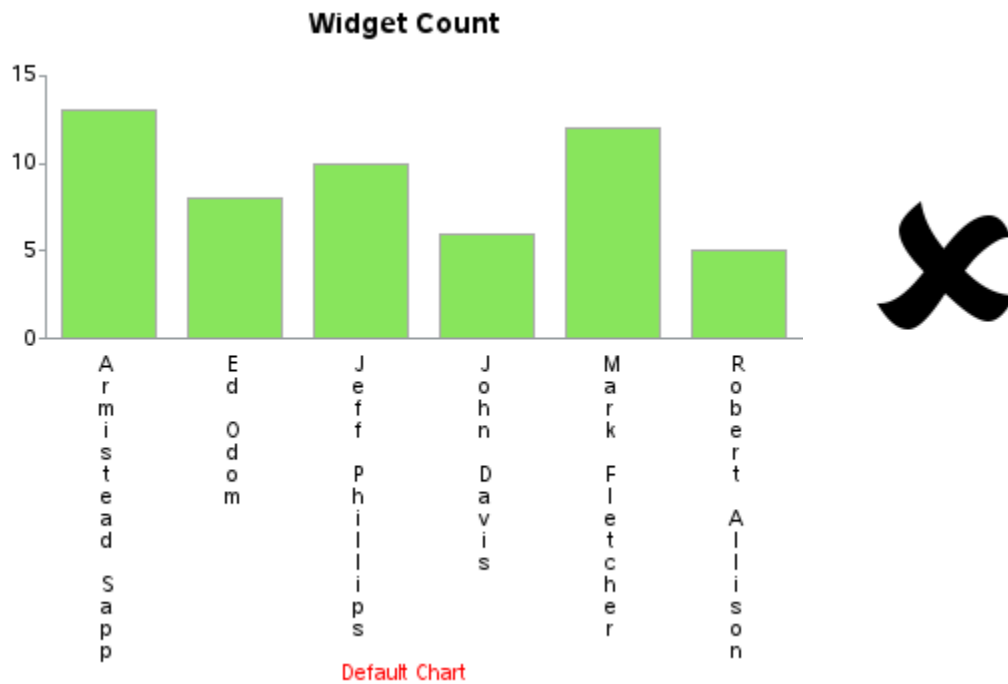
widgets	name
5	Robert Allison
8	Ed Odom
12	Mark Fletcher
10	Jeff Phillips
6	John Davis
13	Armistead Sapp

Here's some simple/minimal code to plot the data as a bar chart:

```
axis1 label=none order=(0 to 15 by 5) minor=none
      offset=(0,0);
axis2 label=none;

title1 ls=1.5 "Widget Count";
pattern1 v=s color=bilg;
proc gchart data=my data;
vbar name / type=sum sumvar=widgets
      raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

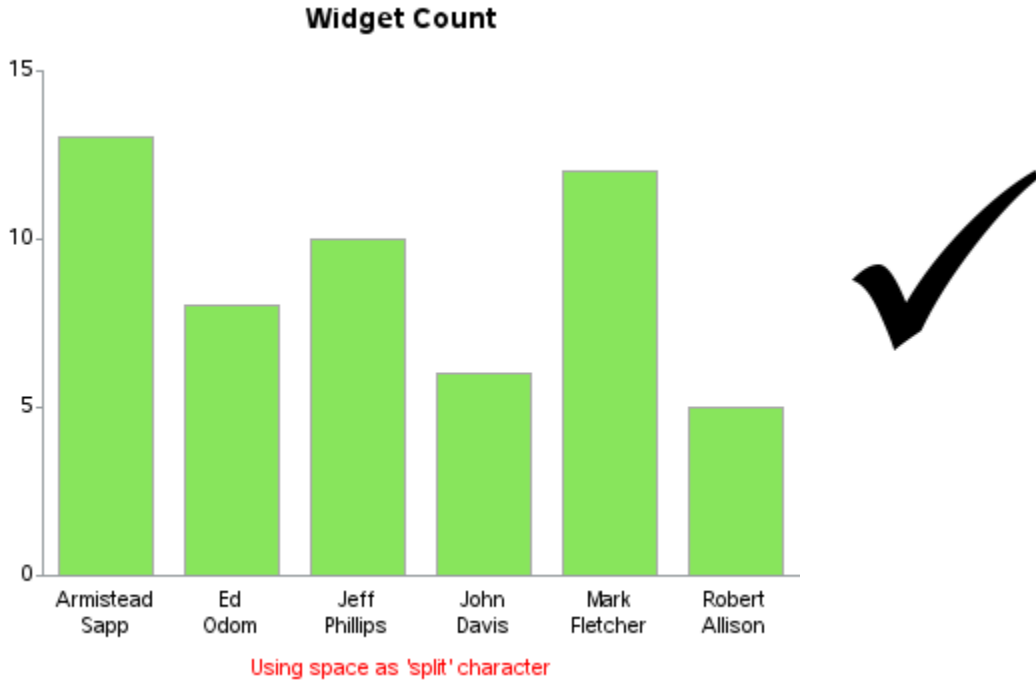
Using the default characteristics for the midpoint axis tick marks (bar labels), GChart makes them fit using the safe/easy technique of stacking the characters up/down below the bars. This is better than GChart simply saying the values are too long, but it's not really an aesthetically pleasing bar chart.



By simply adding the 'split=' option to the midpoint axis, you can create a bar chart that looks much better. The split option lets you specify a character (such as a space), and then any time that character is encountered in the text, the text will be split onto another line.

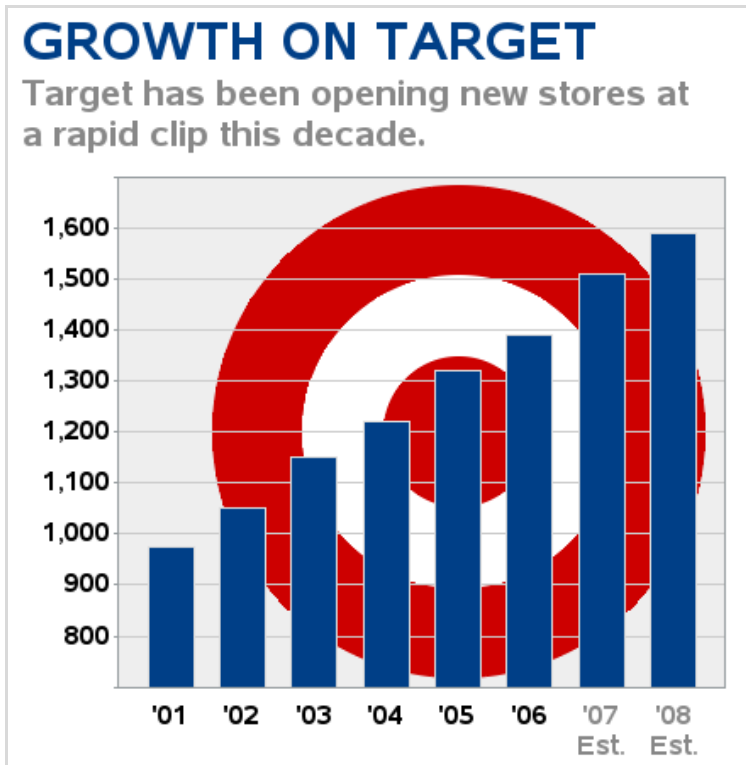
```
axis1 label=none order=(0 to 15 by 5) minor=none
offset=(0,0);
axis2 label=none split=' ';

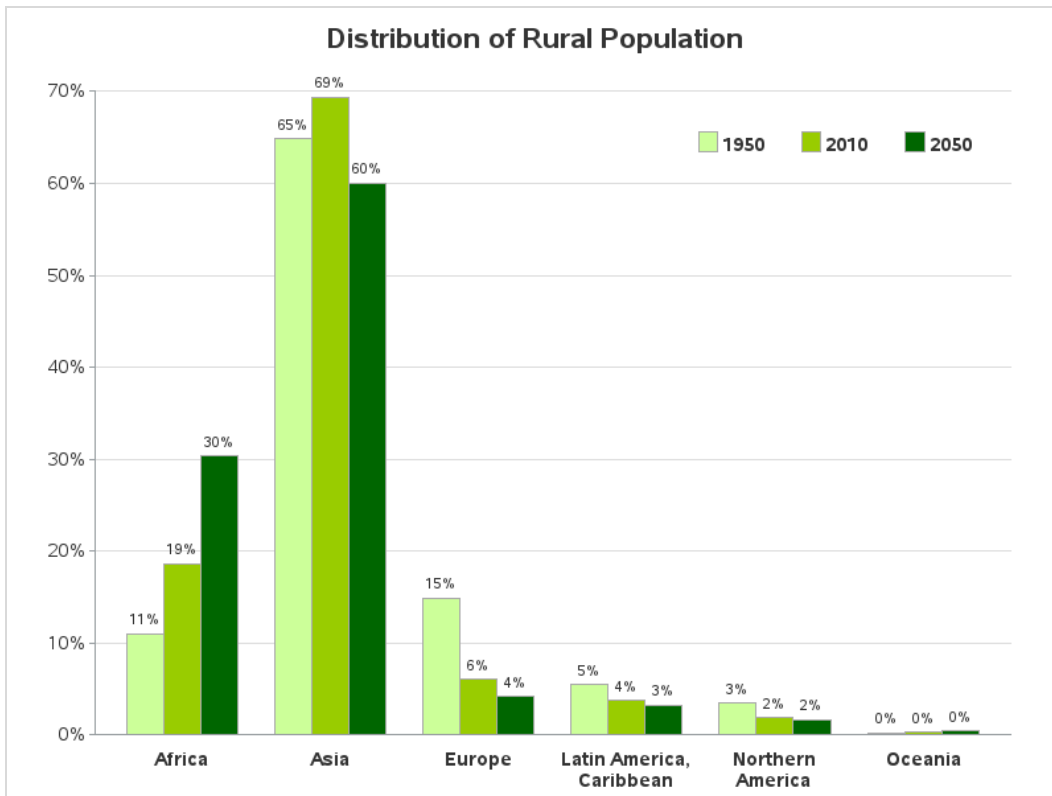
title1 ls=1.5 "Widget Count";
pattern1 v=s color=bilg;
proc gchart data=my_data;
vbar name / type=sum sumvar=widgets
  raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

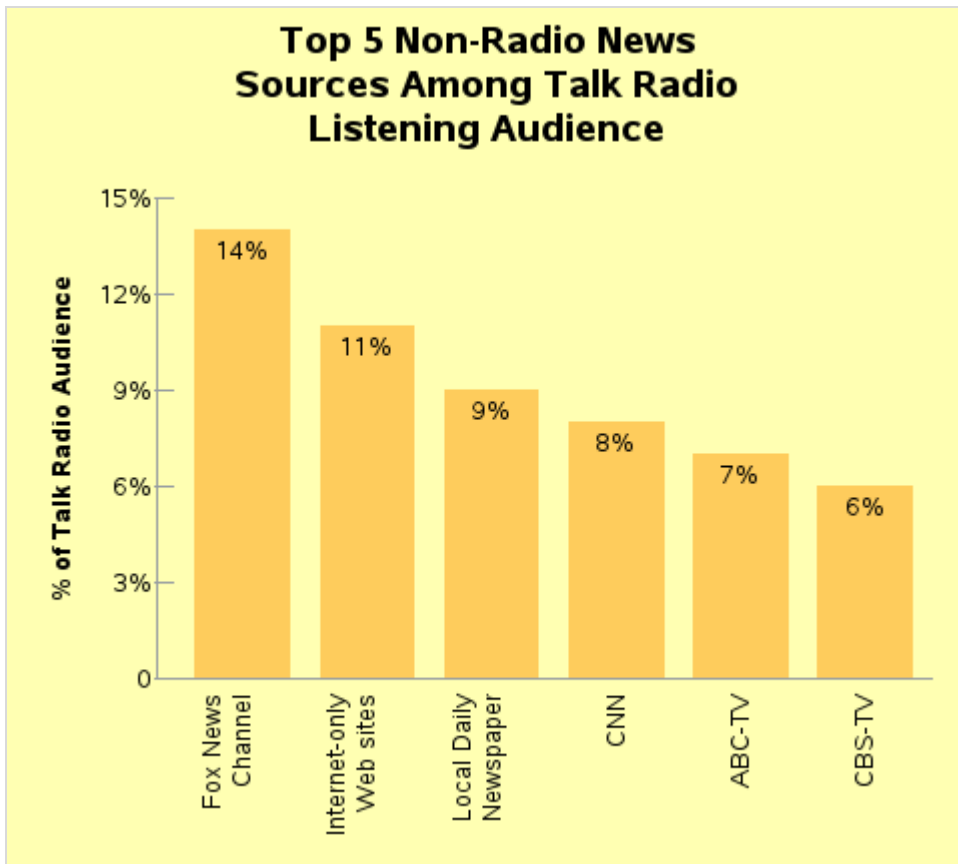
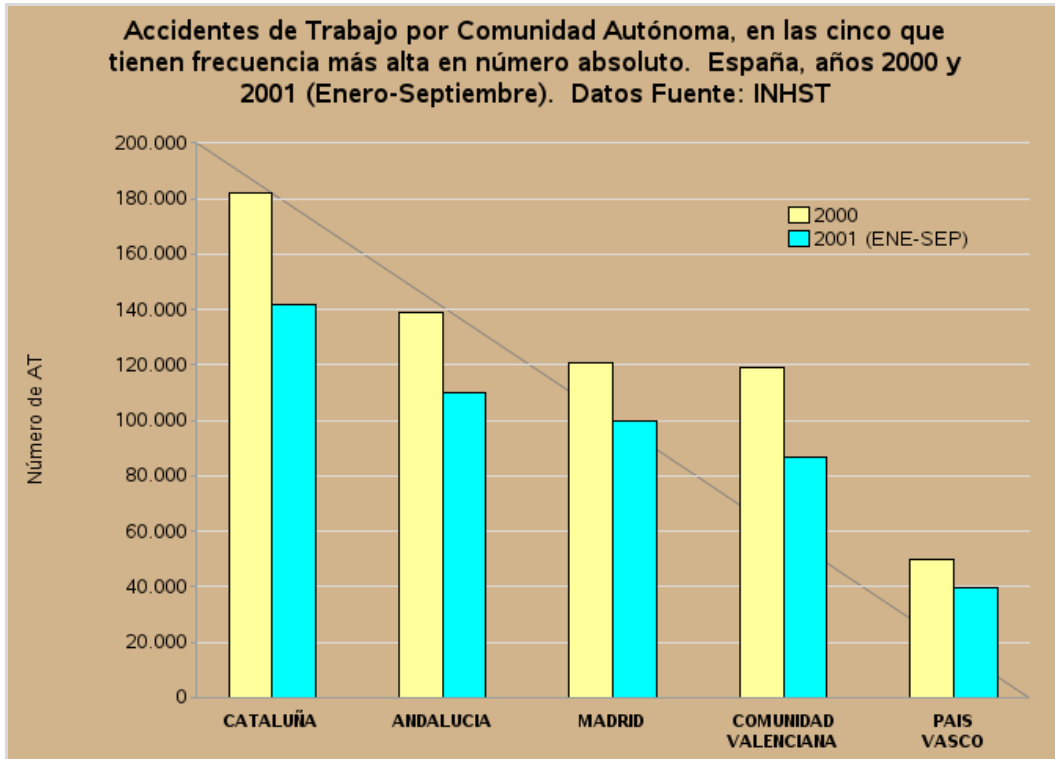


In this simple case with a first & last name, it is natural to split the text at the space. If you have more words, and want to control exactly where the text is split, you can insert a special character into the text (such as a '\')

Below are several examples that use split text in the axis tick mark values:







Splitting Text in Axis Labels

Sometimes it's not the text in the axis tick mark values that is too long, but rather the text of the axis label. You can't use the `split=` option for this, but there is a trick you can use to easily split the axis label into multiple lines! We'll use the following simple data to create the chart to demonstrate this trick...

```
data my_data;
  length lot $1;
  input lot widgets;
  datalines;
A 7
B 5
C 8
D 4
E 3
;
run;
```

lot	widgets
A	7
B	5
C	8
D	4
E	3

Now, let's say we want the axis label for the numeric axis to be a very long/descriptive block of text, such as "Number of Widgets Produced (in boxes of 12)."

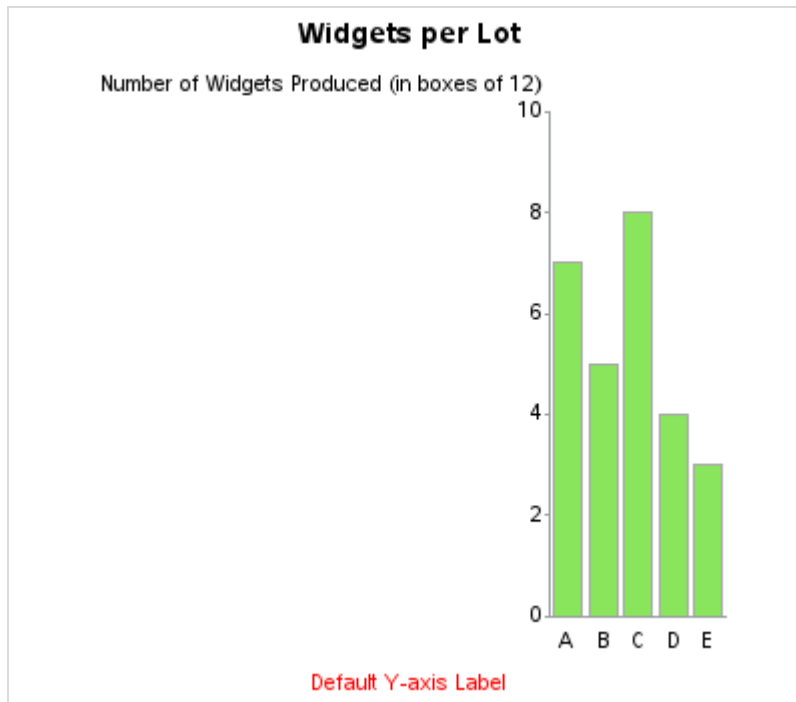
```
axis1 label=('Number of Widgets Produced (in boxes of 12)')
      order=(0 to 10 by 2) minor=none offset=(0,0);
axis2 label=none;
```

```

title1 ls=1.5 "Widgets per Lot";
pattern1 v=s color=bilg;
proc gchart data=my_data;
vbar lot / type=sum sumvar=widgets
  raxis=axis1 maxis=axis2 outline=grayaa noframe;
run;

```

Proc GChart will easily allow this, but the resulting chart is not aesthetically pleasing, and doesn't have a good balance of white-space and graphics.



When we define the axis label, we can break it into multiple text fragments and specify the `justify=` option between the fragments, and GChart will automatically split the text onto separate lines. As the following code shows, the syntax is very simple – nothing tricky or difficult. And instead of typing out the entire “`justify=center,`” most programmers use the `j=c` abbreviation so the code will be more compact (and therefore easier to read):

```

axis1 label=(j=c 'Number of Widgets Produced'
j=c '(in boxes of 12)') order=(0 to 10 by 2)
minor=none offset=(0,0);

```



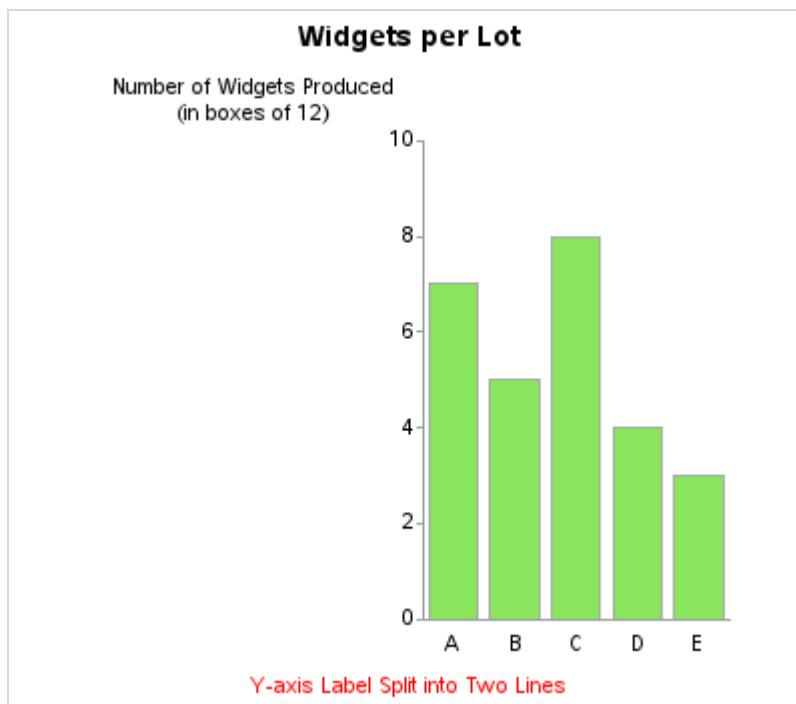
```

axis2 label=none;

title1 ls=1.5 "Widgets per Lot";
pattern1 v=s color=blg;
proc gchart data=my_data;
vbar lot / type=sum sumvar=widgets width=7
raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;

```

The resulting chart looks somewhat better, but the text is still a bit long even when split onto two lines, and therefore the graph still has quite a bit of wasted white-space.



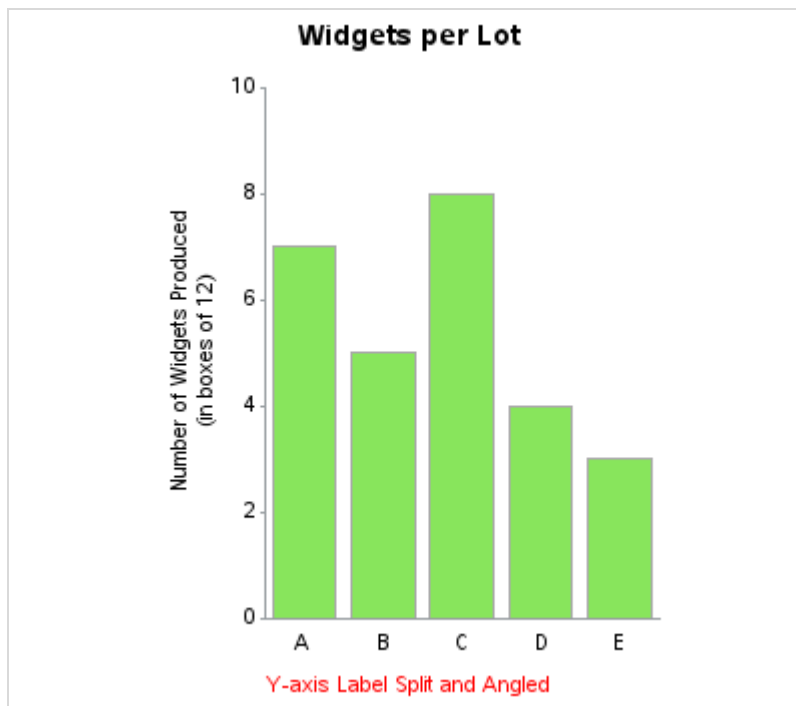
In situations like this, with very long axis labels, you might consider angling the axis label vertically. The label will then fit better (making better use of the white-space), but the text will be slightly more difficult to read (imagine the user having to turn their head sideways to read the label). I will leave you to be the judge of which side of the trade-off you prefer ... my goal here is to teach you both techniques, so you will have the choice.

```

axis1 label=(angle=90 j=c 'Number of Widgets Produced'
             j=c '(in boxes of 12)')
             order=(0 to 10 by 2) minor=none offset=(0,0);
axis2 label=none;

title1 ls=1.5 "Widgets per Lot";
pattern1 v=s color=bilg;
proc gchart data=my_data;
vbar lot / type=sum sumvar=widgets width=9
raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;

```

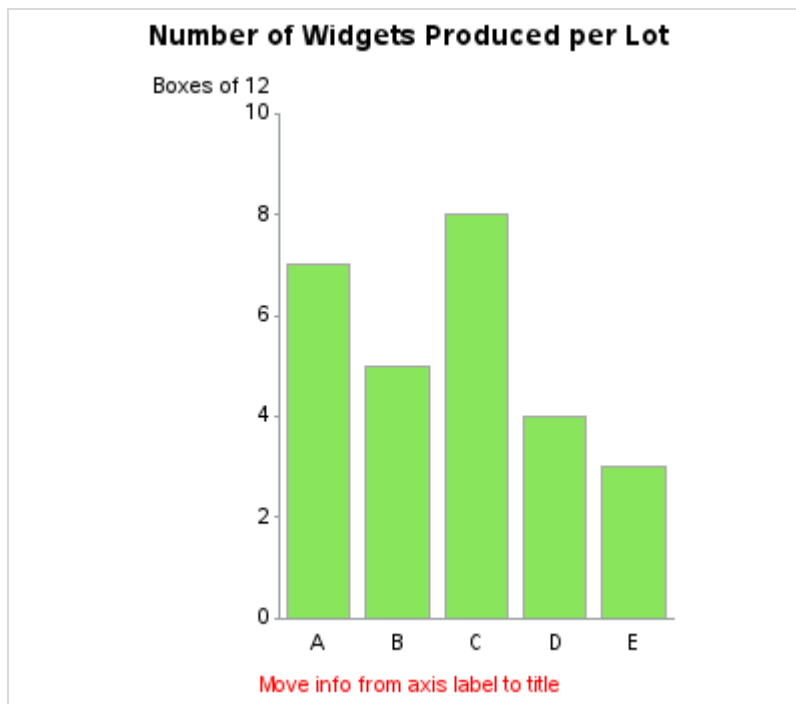


In cases like the one above, where the axis label is so long that it's awkward any way you split or angle the text, you might consider taking a different approach. Perhaps we're trying to put too much information in the axis label, and maybe some of it could be displayed somewhere else in the chart, such as the title or footnote. It's great to have tricks such as splitting and angling axis labels when we need them – but remember that the simplest solution is often the best!

```
axis1 label=('Boxes of 12')
      order=(0 to 10 by 2) minor=none offset=(0,0);
axis2 label=none;

title1 ls=1.5 "Number of Widgets Produced per Lot";
pattern1 v=s color=bilg;
proc gchart data=my_data;
vbar lot / type=sum sumvar=widgets width=9
      raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

Now the axis label fits easily:



Below are several examples that use split text in the axis label:

