CHAPTER 2

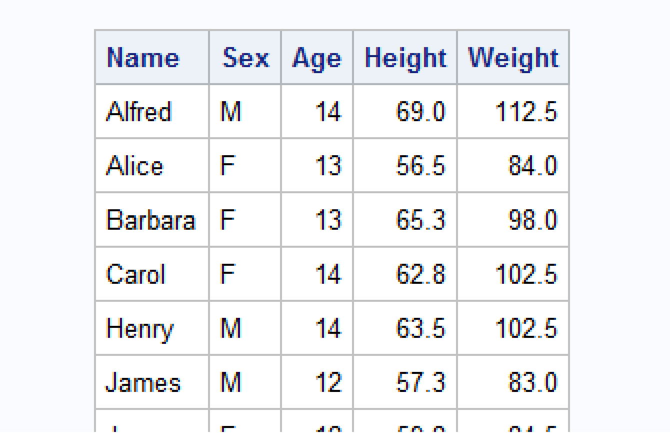
Custom Labels on Bars

Purpose: This chapter describes techniques you can use to label the bars in Proc GChart bar charts, in cases where the labeling cannot be done using the built-in options.

Labels at Top of VBar

This first example shows how to place a custom label at the top of each bar, in GChart VBar.

We'll be using the SASHELP.CLASS data set for the next few examples. Here's what the first few lines of that data look like:

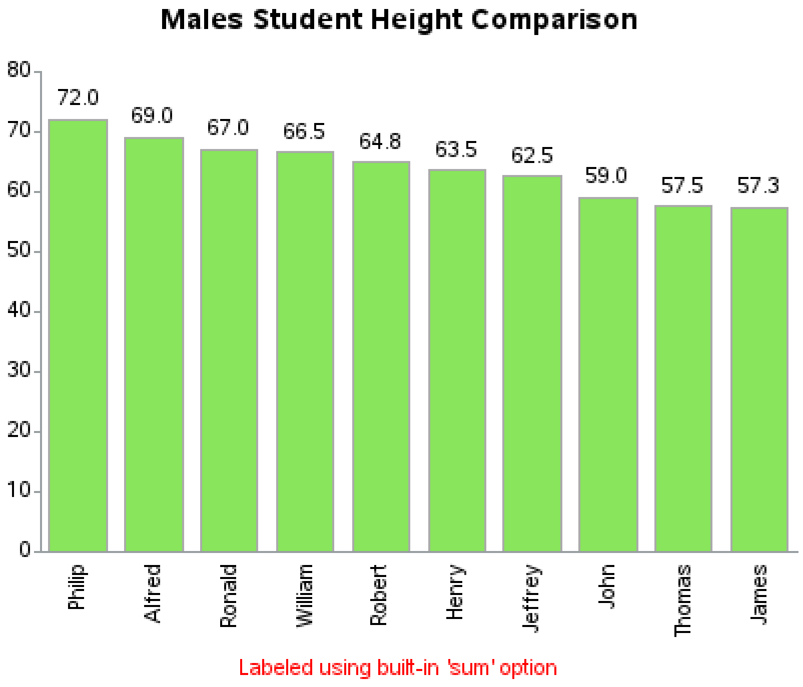


Name	Sex	Age	Height	Weight
Alfred	M	14	69.0	112.5
Alice	F	13	56.5	84.0
Barbara	F	13	65.3	98.0
Carol	F	14	62.8	102.5
Henry	M	14	63.5	102.5
James	M	12	57.3	83.0

You can easily generate a simple bar chart, with values printed at the top of each bar, using code like the following with the 'sum' option (or using 'outside=sum'):

```
axis1 axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(angle=90);

title1 ls=1.5 "Males Student Height Comparison";
pattern1 v=s color=bilg;
```

```
proc gchart data=sashelp.class (where=(sex='M'));
vbar name / type=sum sumvar=height descending sum
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

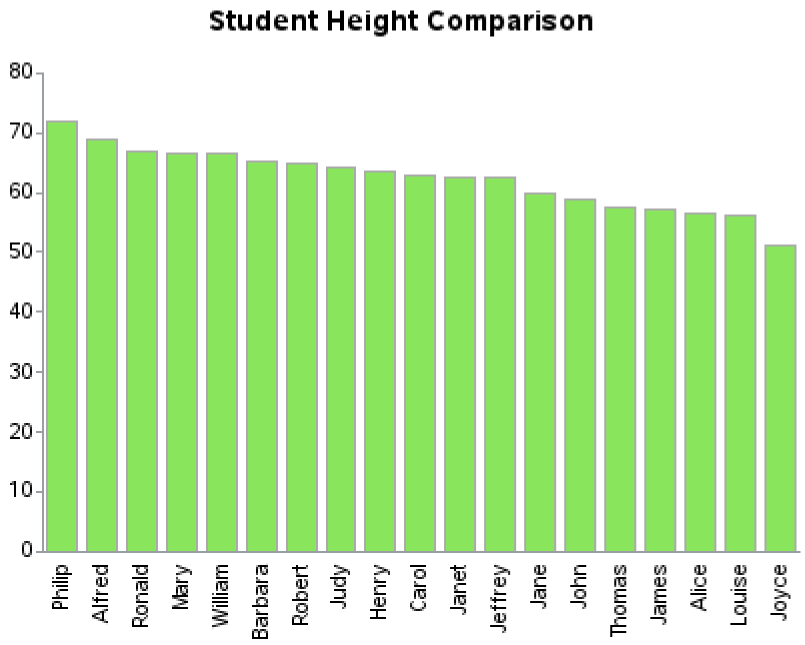


Males Student Height Comparison

Labeled using built-in 'sum' option

But the astute observer will notice that this chart only shows the male students. If we wanted to show all the students (both male & female), then the chart becomes much more crowded, and GChart tells us that it cannot honor the 'sum' option, and suppresses the labels at the top of the bars:

```
axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(angle=90);

title1 ls=1.5 "Student Height Comparison";
pattern1 v=s color=bilg;
proc gchart data=sashelp.class;
vbar name / type=sum sumvar=height descending sum
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

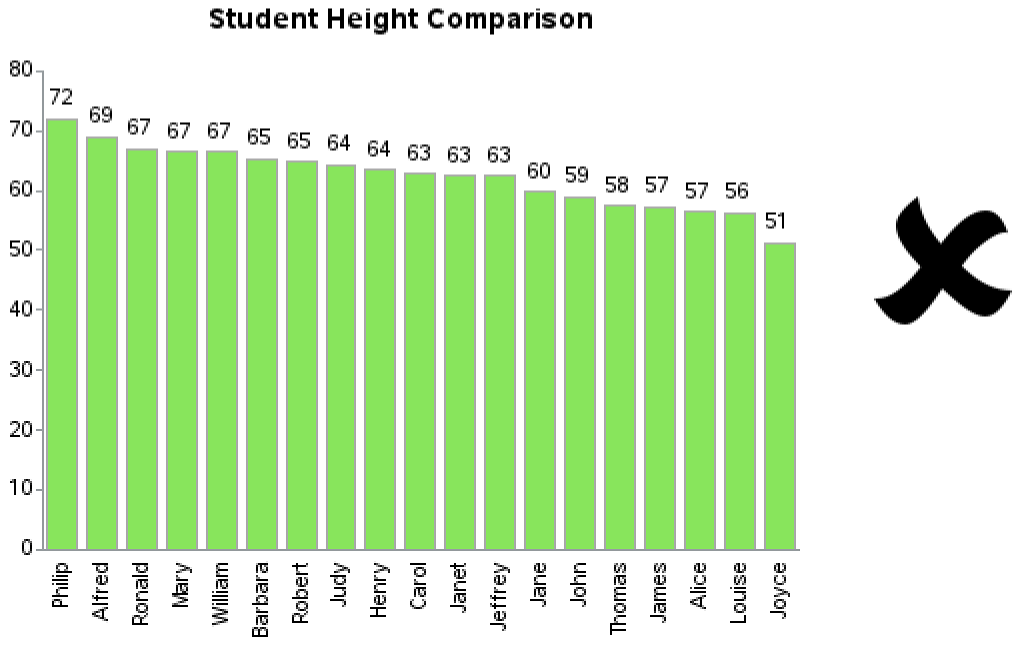


Built-in 'sum' option can't fit labels

As a compromise, let's make the labels shorter by using a format with 0 decimal places, to round the heights off to integer values. We can do that by adding a simple format statement:

```
axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(angle=90);

title1 ls=1.5 "Student Height Comparison";
pattern1 v=s color=bilg;
proc gchart data=sashelp.class;
format height comma2.0;
vbar name / type=sum sumvar=height descending sum
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

But was that a good idea? If you scrutinize the chart below, you'll notice that some of the bars with the same height value printed above them are noticeably different heights. And when it's obvious to the 'naked eye' that the bars and labels do not match, that makes the user (rightfully) call into question the accuracy of the chart. The rounded values are simply not high enough precision to go with the bars, and this compromises the integrity of the chart.



'sum' option can fit labels if rounded values

In cases like this when the simple built-in options will not produce the desired chart, you'll need to customize the chart with your own text labels – and you can do that in a data-driven way using 'annotate'. Annotate lets you control the position, text, and other properties of your custom labels by using values in the data itself. The annotate commands are even stored in a SAS dataset!

Here's the code to create an annotate dataset from the SASHELP.CLASS dataset, to place labels at the top of each bar. Following the code, I explain what each piece does:

```
data bar_anno; set sashelp.class;
xsys='2'; ysys='2'; hsys='3'; when='a';
midpoint=name; y=height;
function='label'; position='2';
text=put(height,comma4.1);
run;
```

xsys='2'; ysys='2'; -- this tells SAS to use the coordinate system of the data

hsys='3'; -- if I specify a size for the text, that size will be in %

when='a'; -- draw the annotate 'after' (ie, on top of) the chart

midpoint=name; -- annotate usually has an x-value, but with vbar it's called midpoint

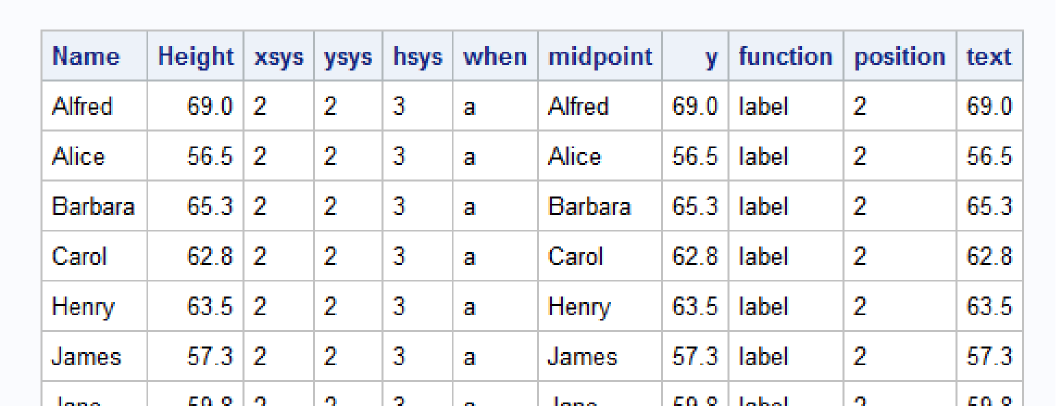
y=height; -- this is the y-value of the text (ie, at the top of the bar)

function='label'; -- we're annotating a label (as opposed to a pie, line, etc)

position='2'; -- place the label slightly above the top of the bar ('8' would be below)

text=put(height,comma4.1); -- and this is the text value to annotate

The resulting annotate dataset looks like this …

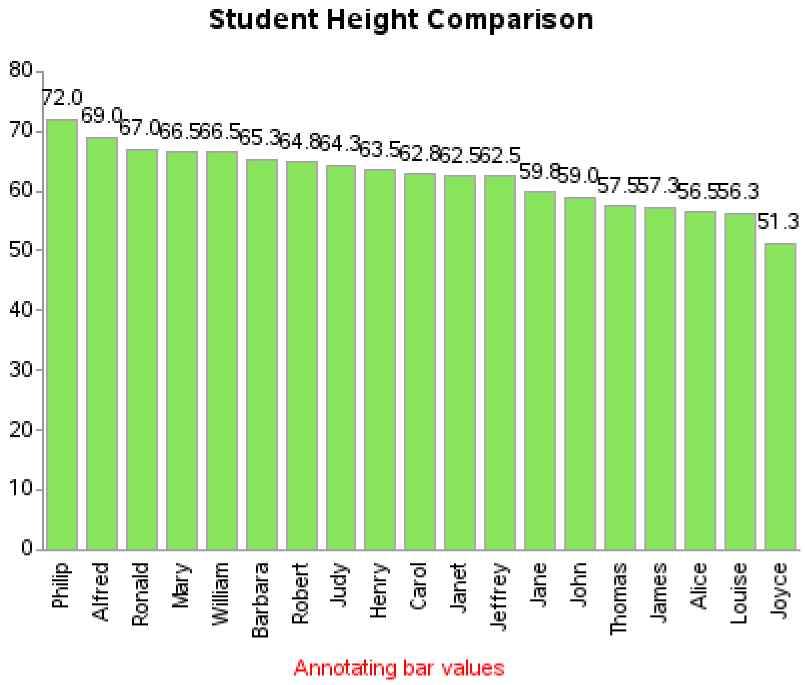


Name	Height	xsys	ysys	hsys	when	midpoint	y	function	position	text
Alfred	69.0	2	2	3	a	Alfred	69.0	label	2	69.0
Alice	56.5	2	2	3	a	Alice	56.5	label	2	56.5
Barbara	65.3	2	2	3	a	Barbara	65.3	label	2	65.3
Carol	62.8	2	2	3	a	Carol	62.8	label	2	62.8
Henry	63.5	2	2	3	a	Henry	63.5	label	2	63.5
James	57.3	2	2	3	a	James	57.3	label	2	57.3
Jane	59.8	2	2	3	a	Jane	59.8	label	2	59.8

And we can add one simple option (anno=bar_anno) to our bar chart code to have is display the annotated labels on the bars, as follows:

```
axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(angle=90);

title1 ls=1.5 "Student Height Comparison";
pattern1 v=s color=bilg;
proc gchart data=sashelp.class;
format height comma2.0;
vbar name / type=sum sumvar=height descending anno=bar_anno
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

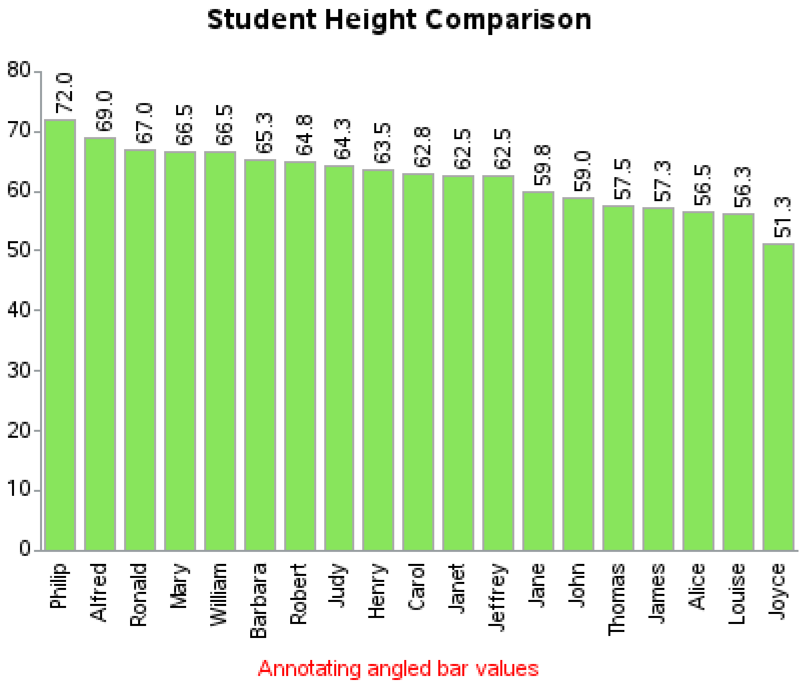
The results are exactly what we asked for (each bar is labeled with the value to 1 decimal place) … but it doesn't look all that great. The labels are too wide, and overlap. It isn't that the chart is bad – it's just that we're not finished yet!



To get around the overlapping problem, you could make the bars wider (assuming you have room), or you could make the annotated text labels smaller … or you could rotate the labels 90 degrees. Rotating the text does make it a bit more difficult to read, but if you want a vertical bar chart with somewhat long labels on the bars, that's often the only practical choice. And in this case, the student names at the bottom of the bars are already rotated, therefore having the values at the top of the bars similarly rotated is at least a consistent thing to do – the user can read both values while they have their head turned sideways. Note that when you angle the text, you also have to pick a 'position' that works well with that angle.

```
data bar_anno; set sashelp.class;
xsys='2'; ysys='2'; hsys='3'; when='a';
midpoint=name; y=height+1;
function='label'; position='6'; angle=90;
text=put(height,comma4.1);
run;

axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(angle=90);
```

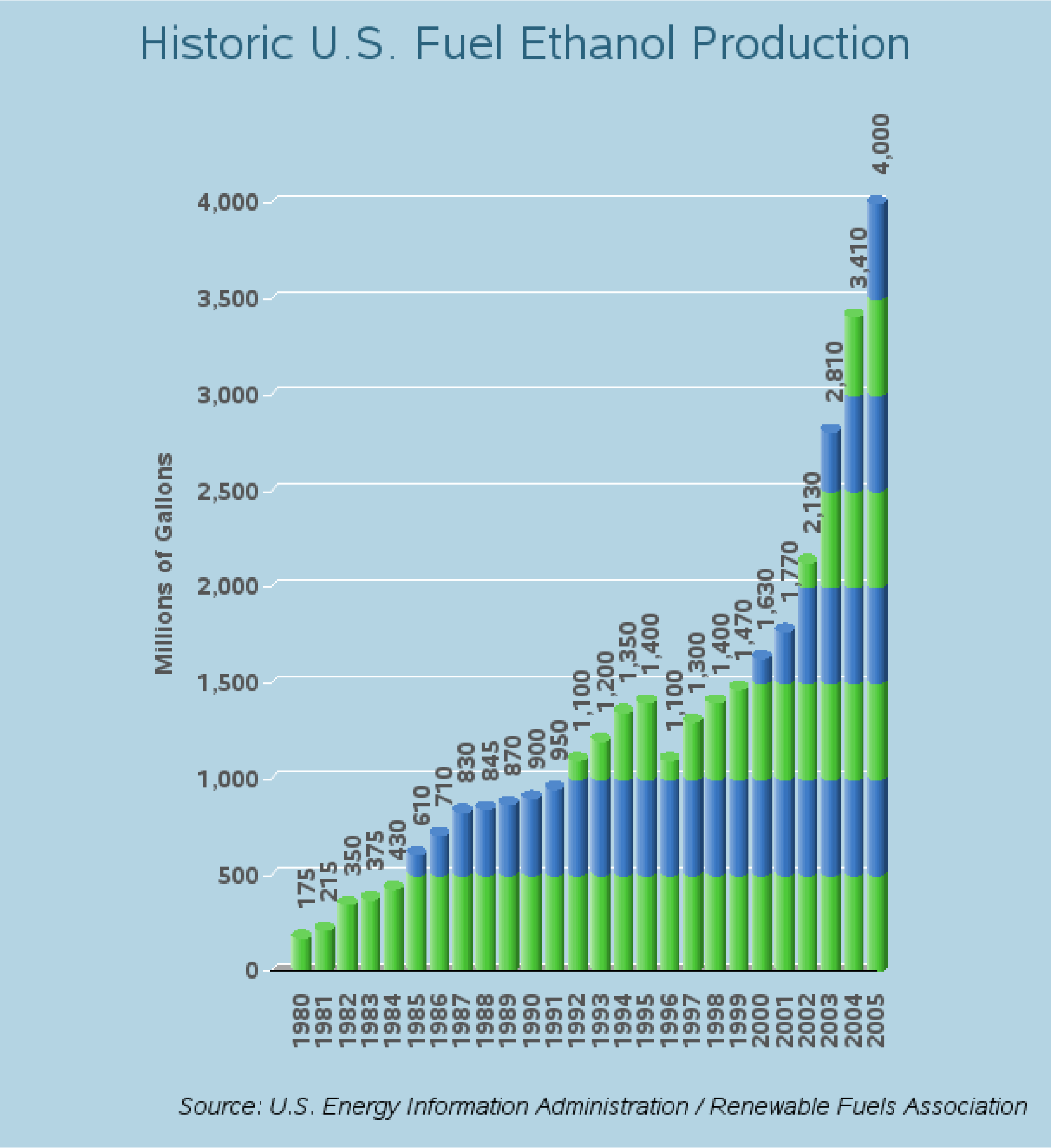
```
title1 ls=1.5 "Student Height Comparison";
pattern1 v=s color=bilg;
proc gchart data=sashelp.class;
format height comma2.0;
vbar name / type=sum sumvar=height descending anno=bar_anno
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```



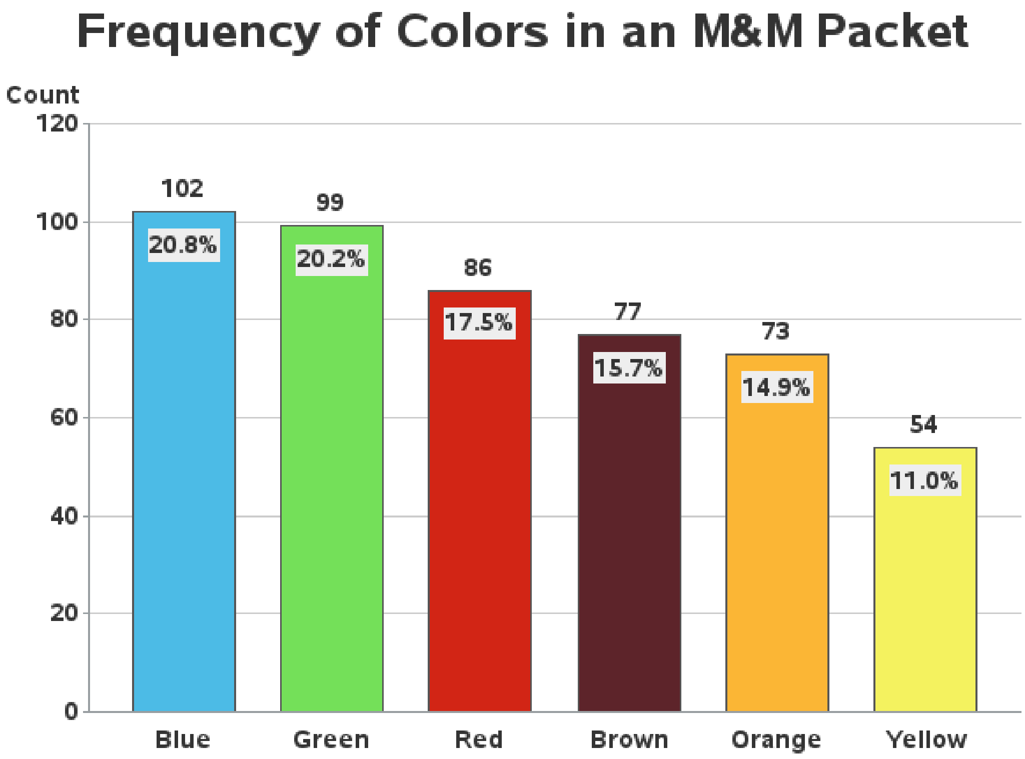
✓

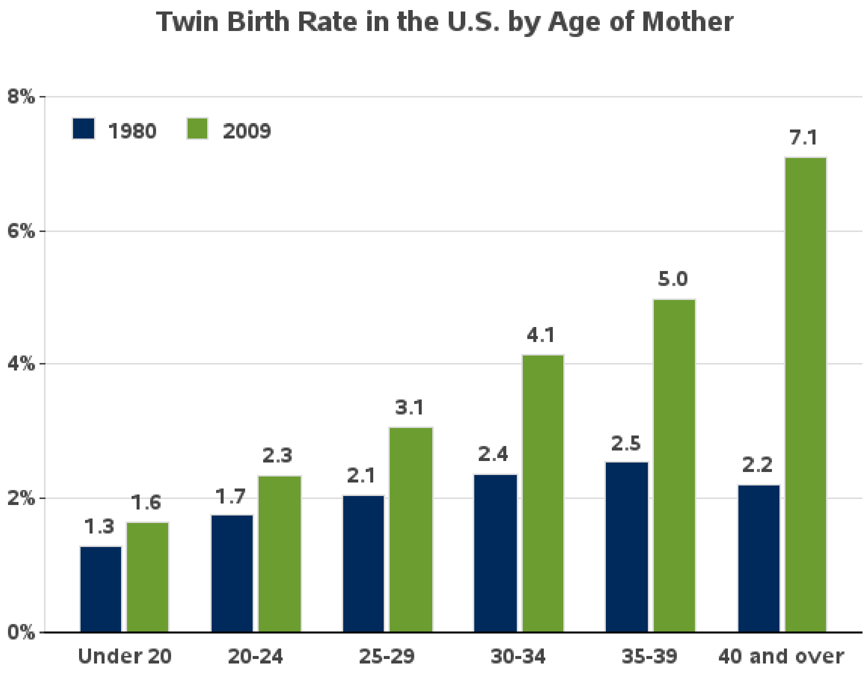
Let’s Talk: Although it’s cool to know how to place custom text labels on your bar charts, it does add a bit of *visual clutter* to the chart. Therefore be sure to determine that the people reading your chart actually need to know the exact values, before going to the trouble of adding them. Also, if your users are viewing the graphs on the Web, perhaps it is a better alternative to let them view the values in html hover-text instead.

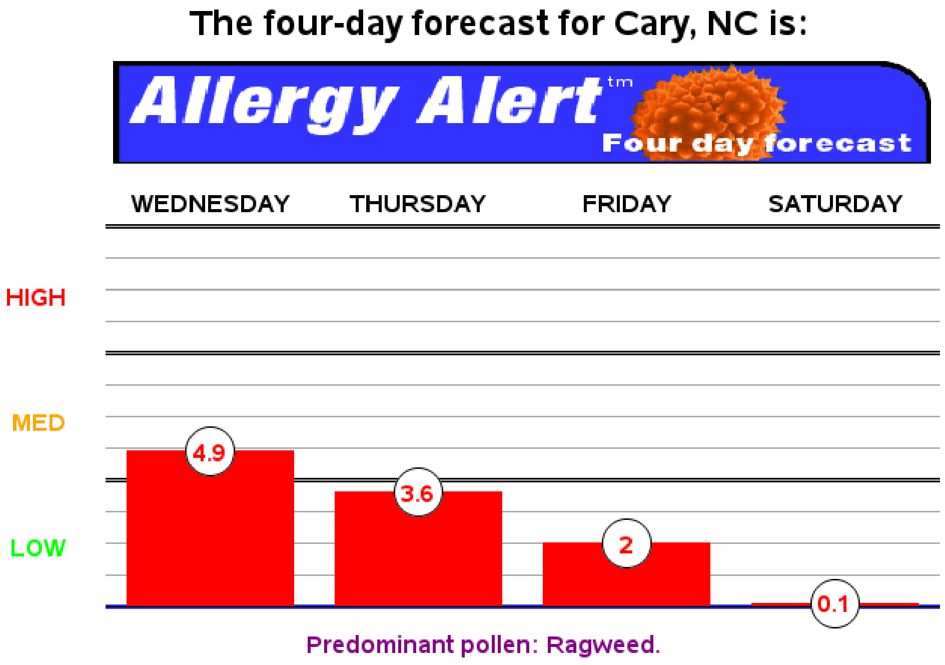
Here are a few other examples that demonstrate how this trick (annotating custom text values at the top of a vbar chart), and slight variations of it, can be used with real data:



Source: U.S. Energy Information Administration / Renewable Fuels Association


Frequency of Colors in an M&M Packet
Count
120
100
80
60
40
20
0
102
20.8%
99
20.2%
86
17.5%
77
15.7%
73
14.9%
54
11.0%
Blue
Green
Red
Brown
Orange
Yellow


Twin Birth Rate in the U.S. by Age of Mother
8%
6%
4%
2%
0%
1980
2009
1.3
1.6
1.7
2.3
2.1
3.1
2.4
4.1
2.5
5.0
2.2
7.1
Under 20
20-24
25-29
30-34
35-39
40 and over



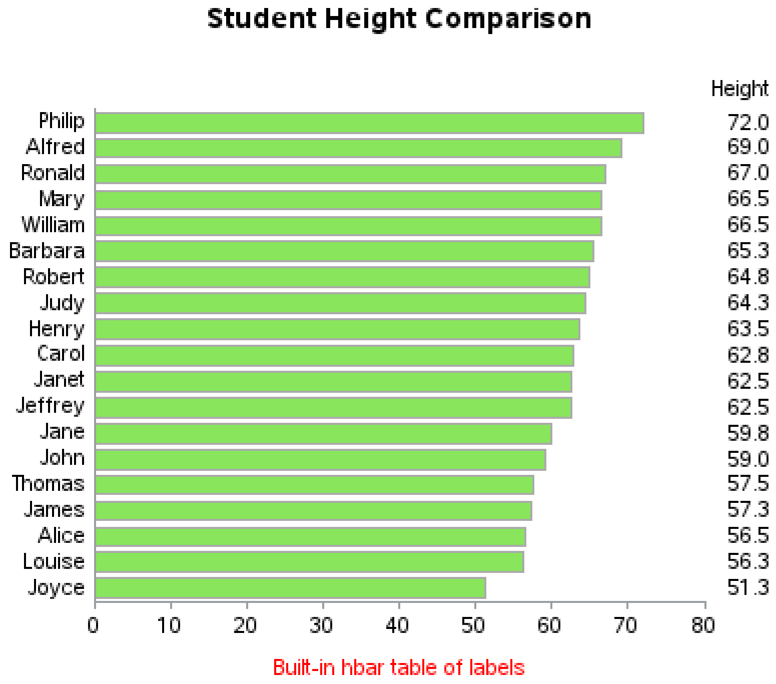
Labels at End of HBar

This next example shows how to place a custom label at the end of each bar, in GChart HBar.

Similar to the vertical bar chart (vbar), the horizontal bar chart (hbar) also has the built-in capability to print the numeric values for each bar. But in the case of hbar, these values go in a column to the right of the graph.

```
axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(justify=right);

title1 ls=1.5 "Student Height Comparison";
pattern1 v=s color=bilg;
proc gchart data=sashelp.class;
hbar name / type=sum sumvar=height descending
 sum sumlabel='Height'
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

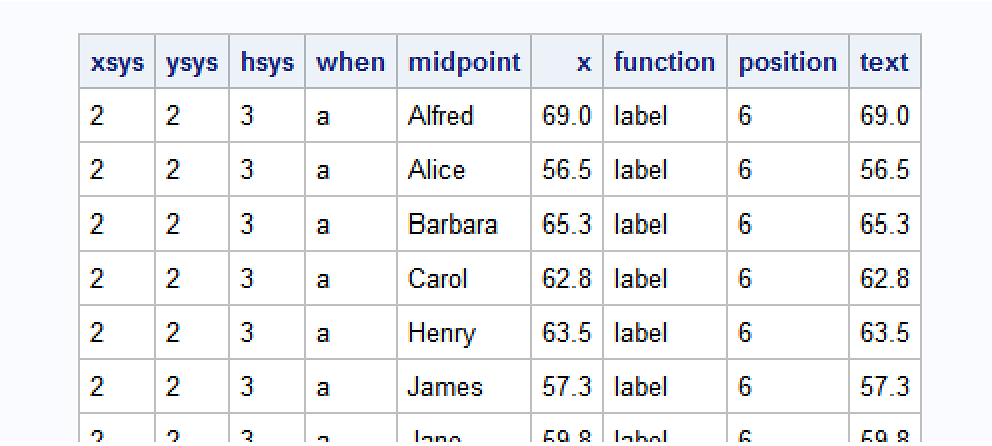
The resulting graph looks nice & neat, with the values in a column such that the decimals line up, and it is easy to compare the values. But it is sometimes difficult to determine exactly which labels go with which bars (especially the shorter bars).



It would be easier to know exactly which value is associated with which bar, if the values were printed right at the end of the bars. You can do this using annotate, with code very similar to the previous example – instead of using midpoint & y, you will now be using midpoint & x. Notice that we’re using position 6, which is to the right of the end of the bar.

```
data bar_anno; set sashelp.class;
xsys='2'; ysys='2'; hsys='3'; when='a';
midpoint=name; x=height;
function='label'; position='6';
text=put(height,comma4.1);
run;
```

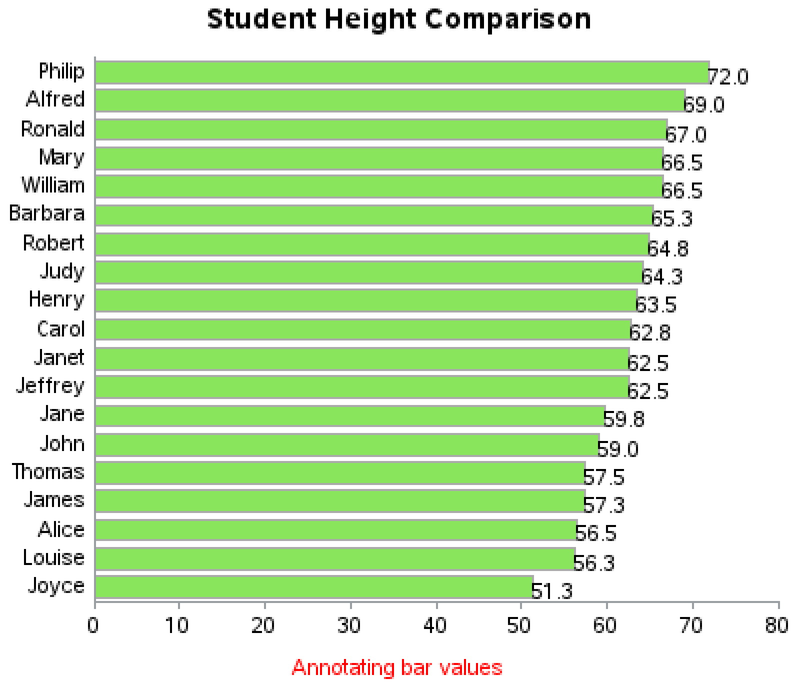
Here are the important variables in the annotate dataset:



xsys	ysys	hsys	when	midpoint	x	function	position	text
2	2	3	a	Alfred	69.0	label	6	69.0
2	2	3	a	Alice	56.5	label	6	56.5
2	2	3	a	Barbara	65.3	label	6	65.3
2	2	3	a	Carol	62.8	label	6	62.8
2	2	3	a	Henry	63.5	label	6	63.5
2	2	3	a	James	57.3	label	6	57.3
2	2	3	a	Jane	59.8	label	6	59.8

```
axis1 label=none minor=none offset=(0,0);
axis2 label=none value=(justify=right);

title1 ls=1.5 "Student Height Comparison";
pattern1 v=s color=bilg;
proc gchart data=sashelp.class anno=bar_anno;
hbar name / type=sum sumvar=height descending sum nostats
 raxis=axis1 maxis=axis2 coutline=grayaa noframe;
run;
```

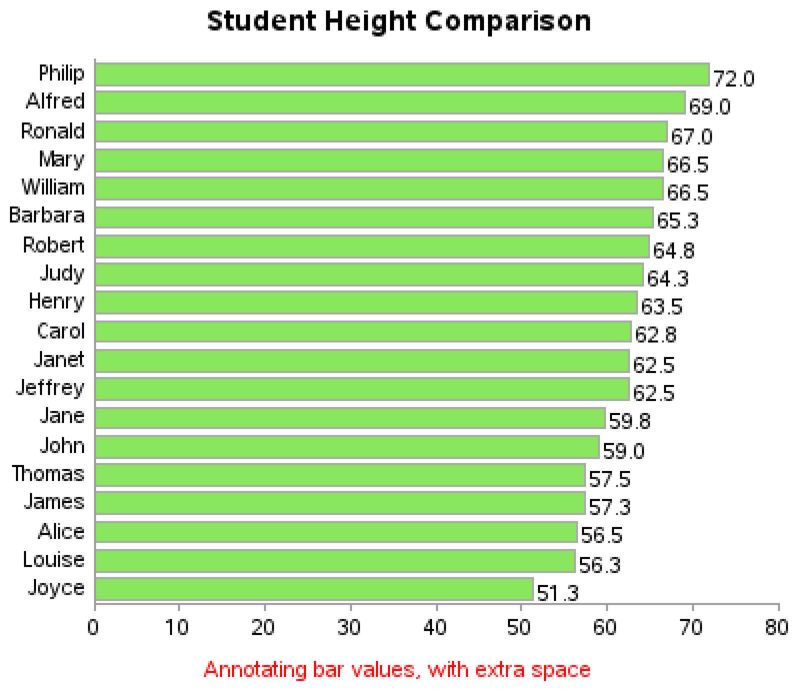
The resulting chart is almost what we want … but the values are a little too close to the bars, which makes it look a little crowded.



You can pad a little extra blank space on the left of the text variable to add a little extra space between the bar and the annotated label.

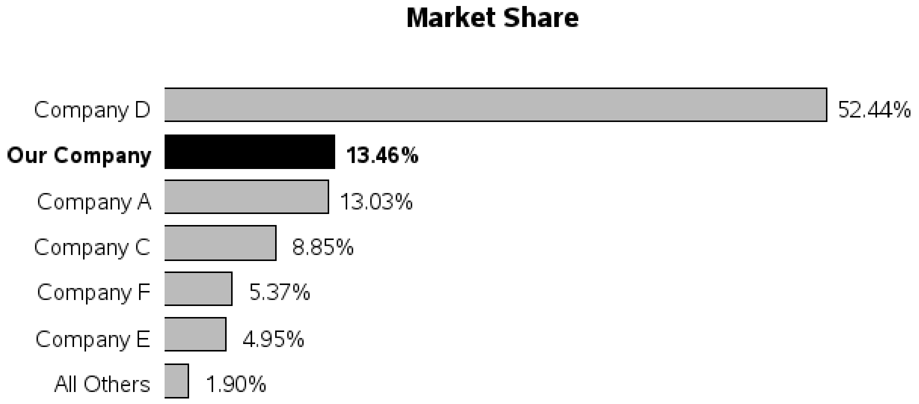
```
data bar_anno; set sashelp.class;

xsys='2'; ysys='2'; hsys='3'; when='a';

midpoint=name; x=height;

function='label'; position='6';

text='  '||put(height,comma4.1);
run;
```

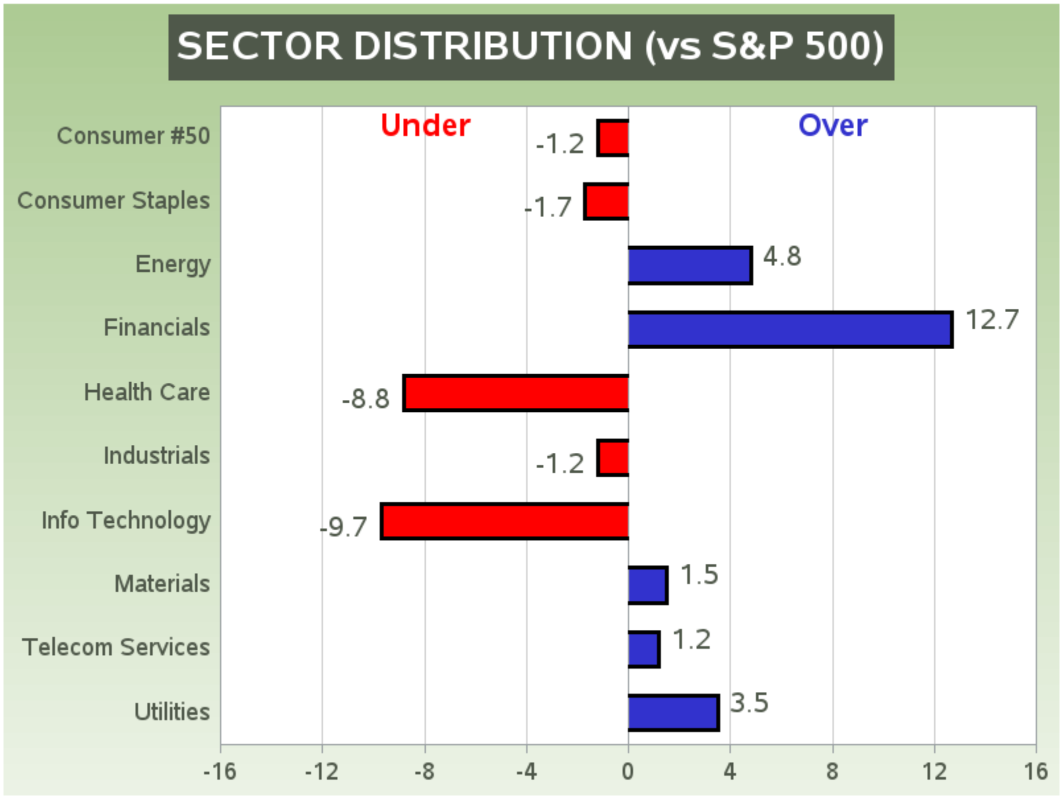
The resulting chart looks much better, with aesthetic spacing between the bar and the value:

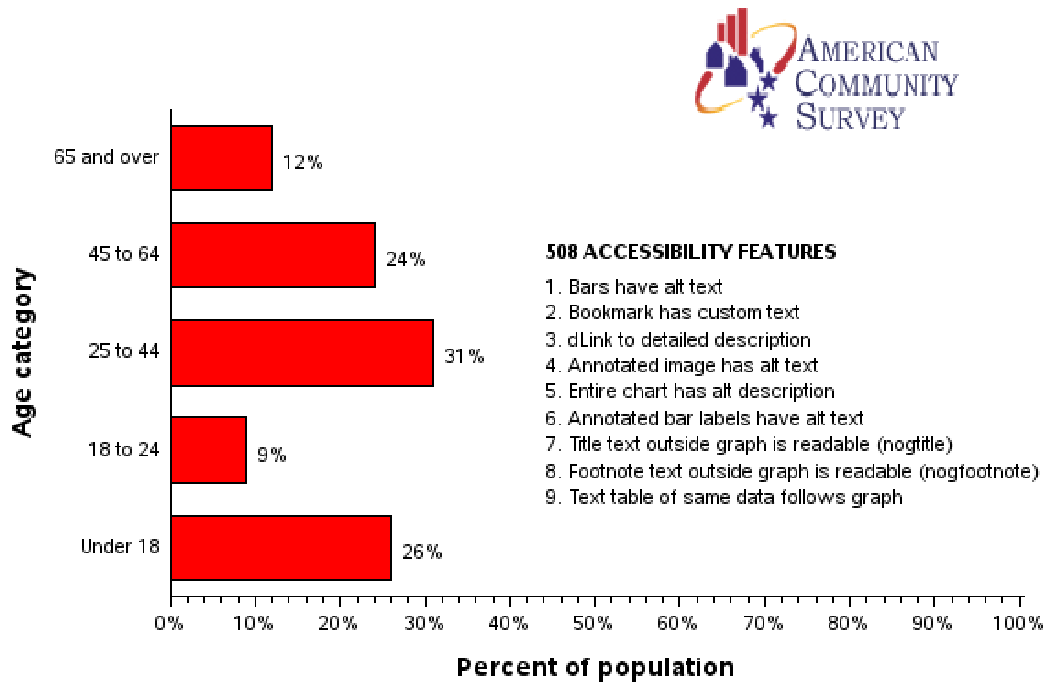

Student Height Comparison

Annotating bar values, with extra space

Here are a few other examples that demonstrate how this trick (annotating custom text values at the end of an hbar chart) can be used with real data:

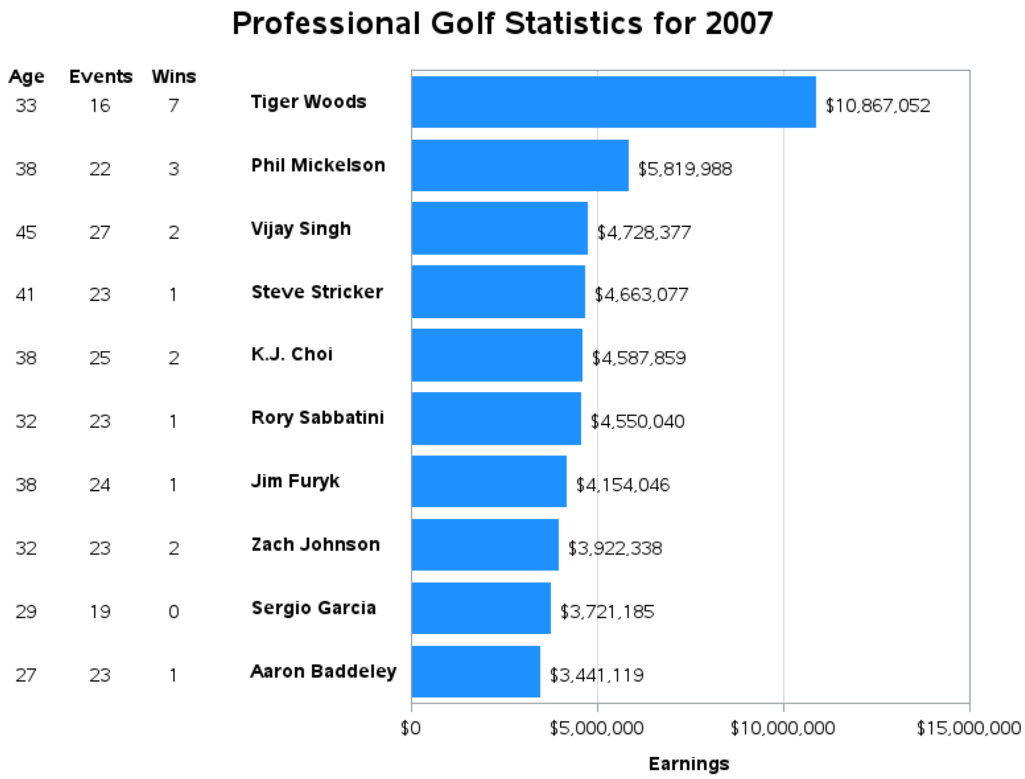






508 ACCESSIBILITY FEATURES

1. Bars have alt text
2. Bookmark has custom text
3. dLink to detailed description
4. Annotated image has alt text
5. Entire chart has alt description
6. Annotated bar labels have alt text
7. Title text outside graph is readable (nogtitle)
8. Footnote text outside graph is readable (nogfootnote)
9. Text table of same data follows graph

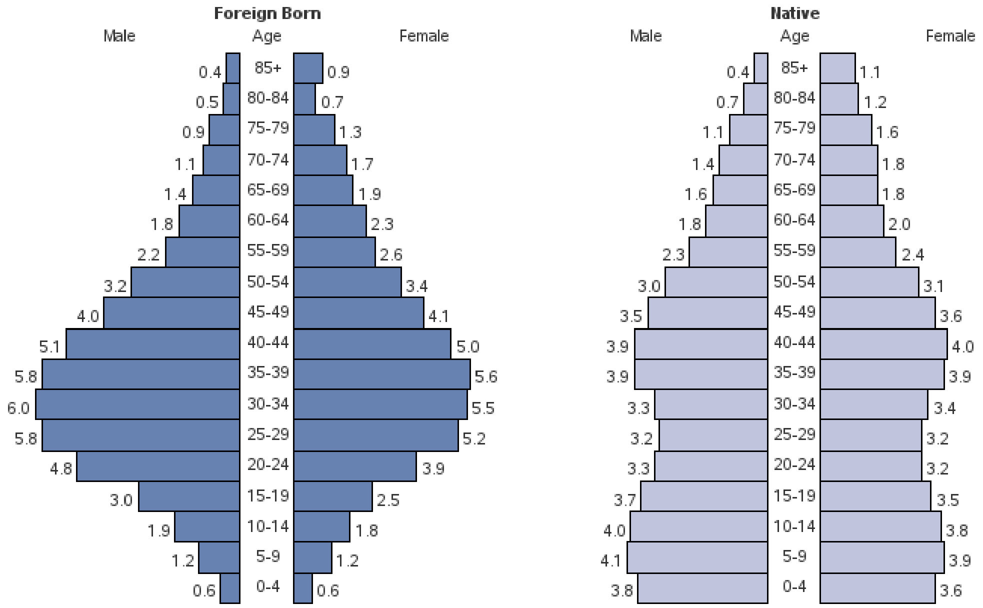


Professional Golf Statistics for 2007

Age	Events	Wins		
33	16	7	**Tiger Woods**	$10,867,052
38	22	3	**Phil Mickelson**	$5,819,988
45	27	2	**Vijay Singh**	$4,728,377
41	23	1	**Steve Stricker**	$4,663,077
38	25	2	**K.J. Choi**	$4,587,859
32	23	1	**Rory Sabbatini**	$4,550,040
38	24	1	**Jim Furyk**	$4,154,046
32	23	2	**Zach Johnson**	$3,922,338
29	19	0	**Sergio Garcia**	$3,721,185
27	23	1	**Aaron Baddeley**	$3,441,119

Age and Sex by Nativity: 2000

(Data based on sample. For information on confidentiality protection, sampling error, nonsampling error, and definitions, see *www.census.gov/prod/cen2000/doc/sf3.pdf*)



Each bar represents the percent of the population (foreign-born or native) who were in the specified age-sex group.
Source: U.S. Census Bureau, Census 2000, special tabulations.