

Easy, Elegant, and Effective SAS® Graphs:

Inform and Influence
with Your Data

Let your data talk

Easy To Build

- Point example code at your data
- Make a few simple code changes

Elegance

the beauty of simplicity

Effective Communication

- Everything that needs
to be said/shown
- Only what needs
to be said/shown

Unneeded Complexity

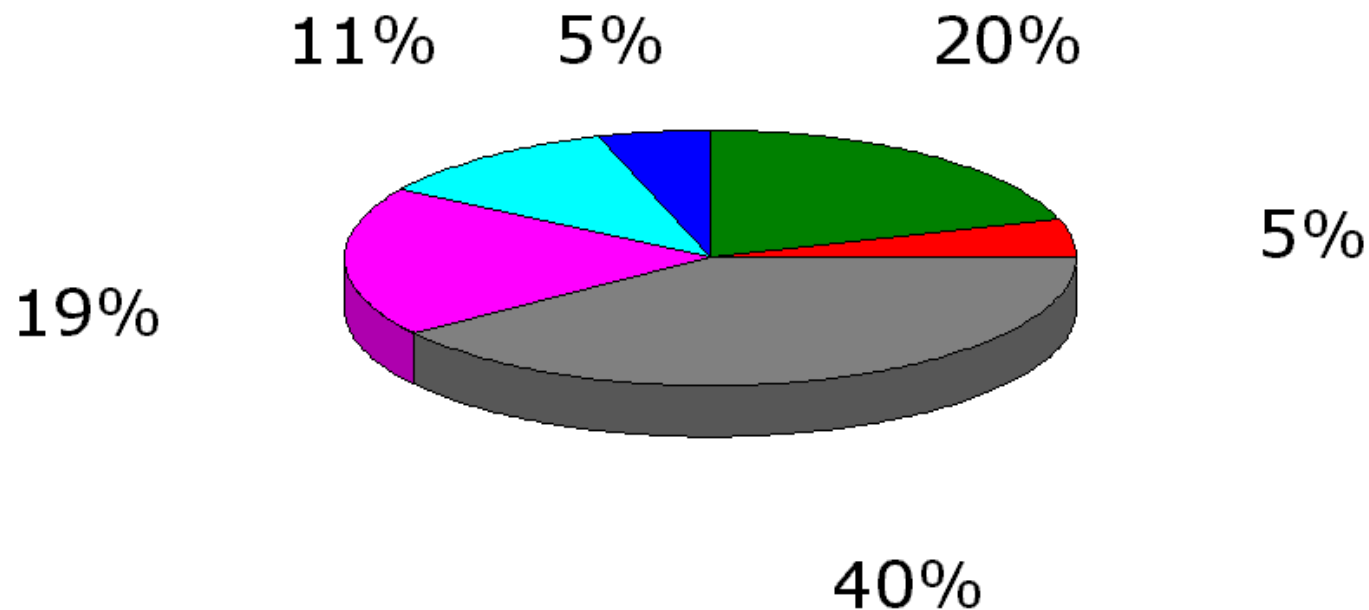
Distorts, Distracts, and Delays Communication

Sparse Image focuses attention

Use color to communicate,
not to decorate

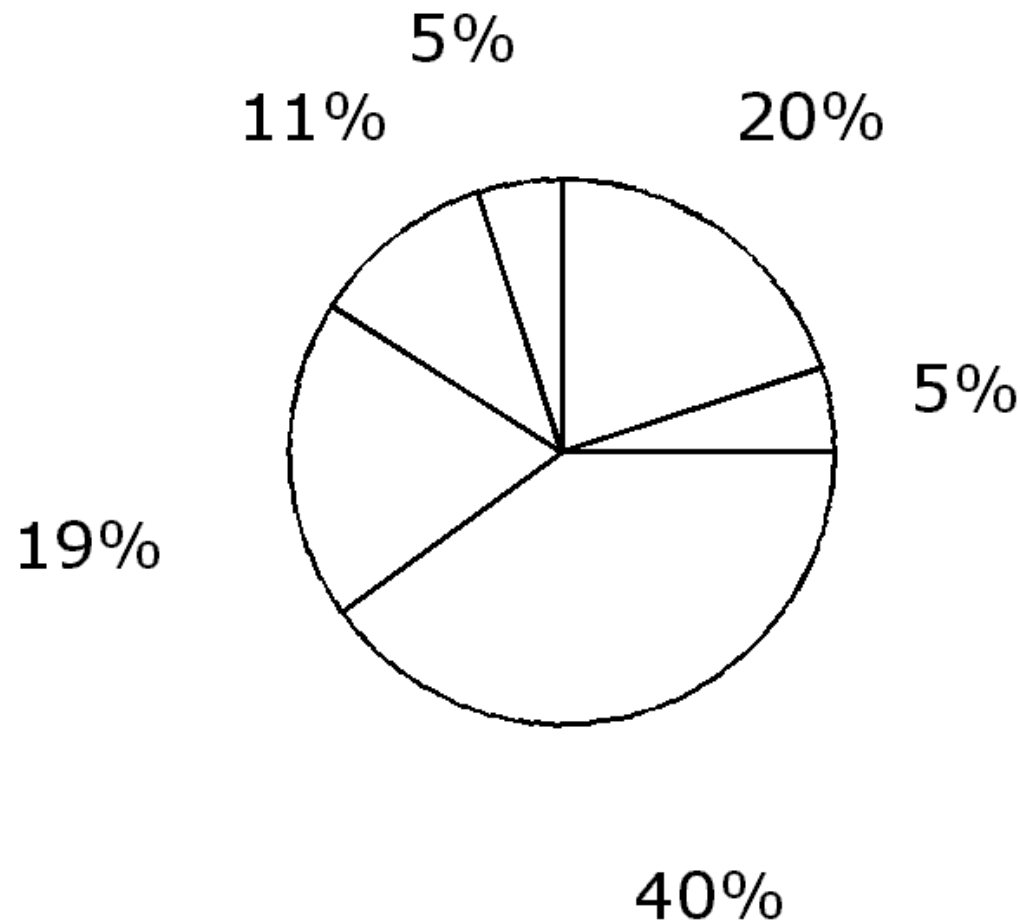
Figure 1: 3D Pie Chart.

Always Distorts Apparent Relative Size of Slices.



3D is for 3 variables

Figure 2: 2D Pie Chart. Simpler is Better.
Always Presents Accurate Relative Size of Slices.



```
pattern1 v=pempty r=6;  /* 6 empty slices */  
proc gchart data=DataForSimpleCharts;  
pie Name /              /* var for slice names */  
    sumvar=Value        /* var for measurements */  
    noheading /* suppress default heading */  
    coutline=CX000000  /* color of pie slice  
                        outline is Black */  
    woutline=2 /* thicken the pie outline */  
    slice=none /* no Name labels */  
    value=none /* no Value labels */  
    percent=outside; /* Percent labels */  
  
run; quit;
```

For All Figures in Tutorial

➤ `goptions device=PNG;`

Portable Network Graphic

good for MS Word or hardcopy

➤ `goptions`

`ftext='Verdana' /* default font */`

`htext=1.00; /* default height */`

For Figure 2:

```
goptions
vpos      = 16      /* rows for "cells" */
vsize     = 2.40 IN
ymax      = 2.40 IN
ypixels   = 720     /* 300 DotsPerInch */
hpos      = 34      /* columns for "cells" */
hsize     = 3.25 IN
xmax      = 3.25 IN
xpixels   = 975;
```

```
title2
/* no c= means take default colour */
h=1.00
f='Georgia'
j=L
' Figure 2: '
c=CX0000FF /* switch to blue */
'2D Pie Chart. Simpler is Better.'
j=L /* justify left again: start new line */
' Always Presents Accurate Relative Size.';
```

GOPTIONS

➤ `proc goptions; run;`

to see all options in effect

➤ `goptions device=YourPreferredDriver;`

to select driver with its default GOPTIONS

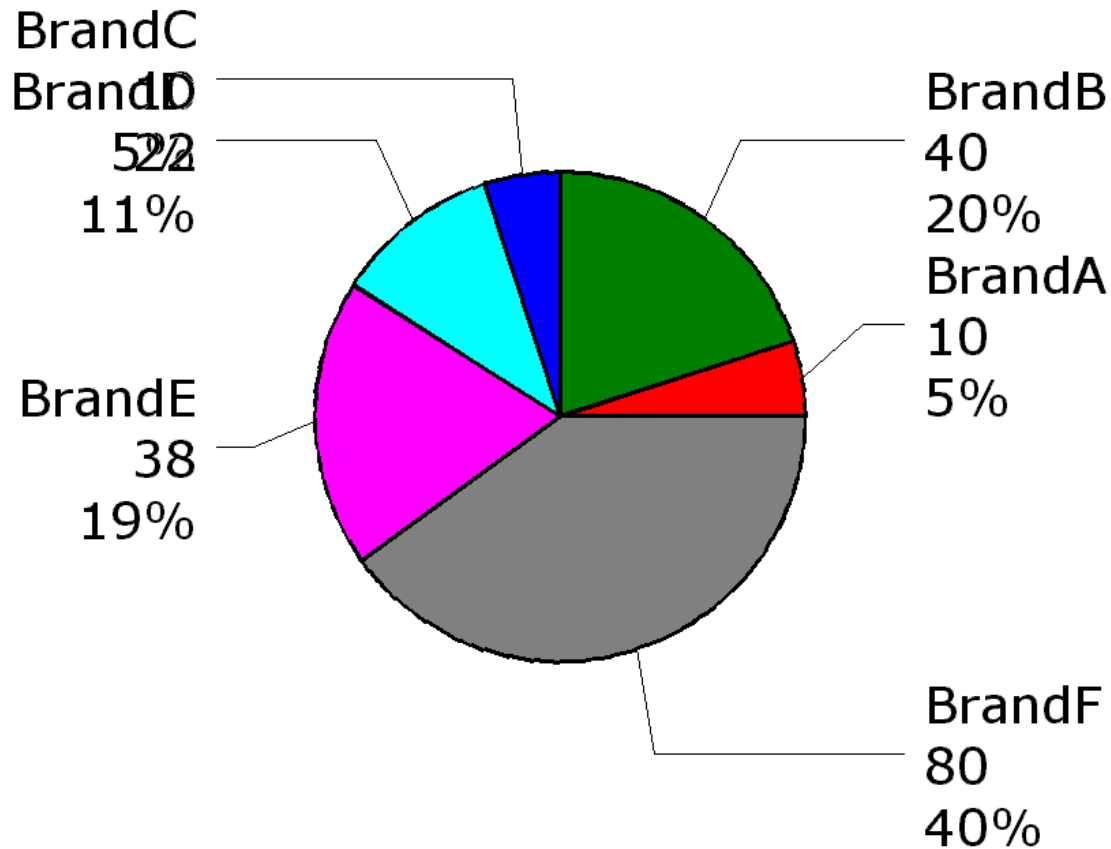
➤ For characteristics of any device driver,
look in `sashelp.devices`

For h=xx.xxx Units

- If no units listed,
value of **GOPTIONS GUNIT=** is used
- **GUNIT=**
CELLS (default
depends on hsize, hpos, vsize, vpos)
CM (centimeters) or **IN** (inches)
PCT (% of graphic output area)
PT (point size)

Standard Form of 2D Pie Chart. **Susceptible to Label Overlaps.**

Market Share by Brand, Sales in Billions of Units



Any SAS/GRAPH Problem Has a Solution

Non-solutions to this problem include:

- tweaking HPOS and VPOS
- desperately changing order of slices
- consolidating smaller slices into OTHER

“OTHER” does not communicate

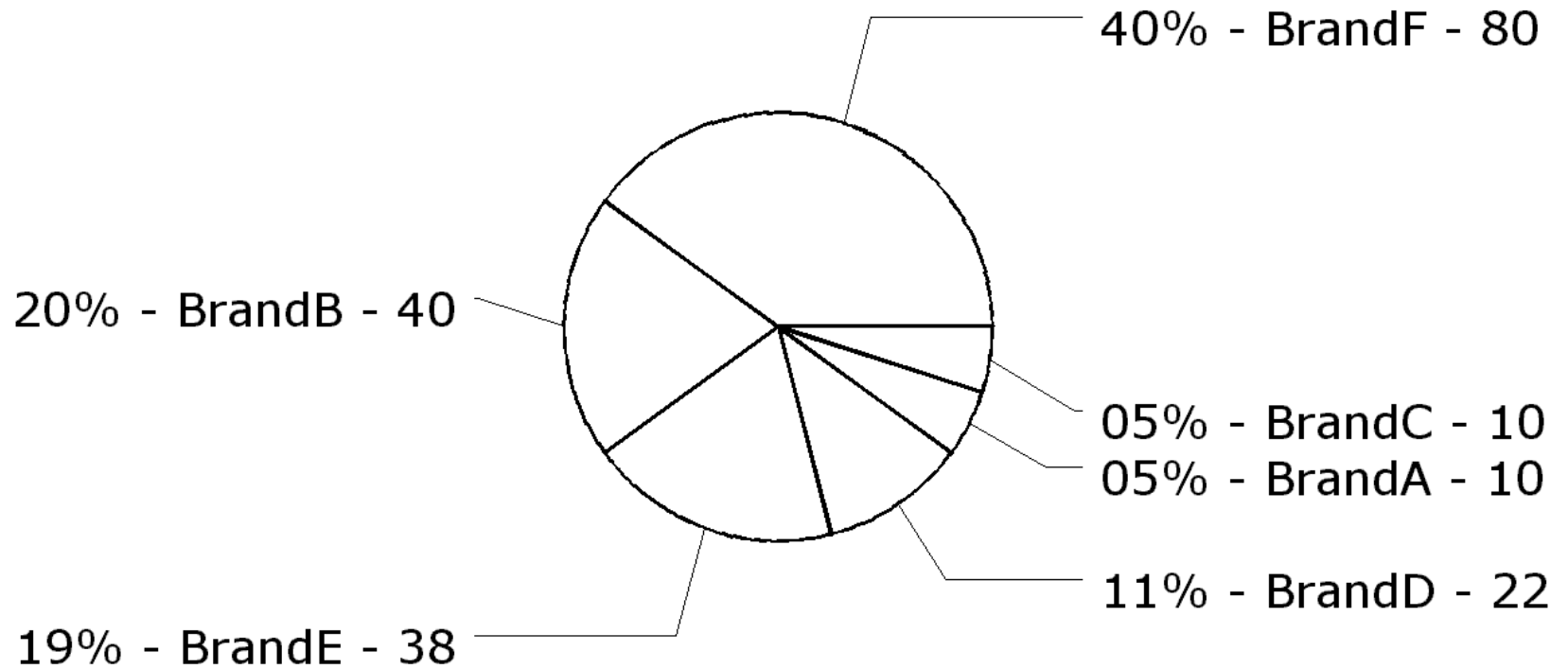
- Graphs should answer questions,
not cause questions
- such as “What’s in OTHER?”

SAS/GRAPH Solutions

- NO “tweaking”
- Get it right the first time, every time

Figure 4: Best Pie Chart Without a Legend.

Market Share by Brand, Sales in Billions of Units



```

proc means data=DataForSimpleCharts
    noprint sum; var Value;
output out=PieTotal sum=TotalValue;
run;

data SliceNameWithPercentAndValue;
length NameWithPercentAndValue $ 17;
set DataForSimpleCharts;
if _N_ eq 1 then set PieTotal;

Percent = (Value / TotalValue) * 100;

NameWithPercentAndValue =
trim(left(put(Percent,Z2.))) || '%' || ' - '
|| trim(left(Name)) || ' - '
|| trim(left(put(Value,2.)));

run;

```

```
proc gchart
data=SliceNameWithPercentAndValue;

pie NameWithPercentAndValue / sumvar=Value
noheading coutline=CX000000 woutline=2
descending /* order pie slices from
              large to small */
slice=arrow /* connect label outside pie
              to slice with arrow */
percent=none
value=none;

run; quit;
```

Let the graph talk

- **Communicate with order**
of pie slices, chart bars, etc.
- Unless a communication reason
not to do so, use:
 - **DESCENDING** (if bigger is better)
 - **ASCENDING** (if smaller is better)

Pie Chart With a Custom Legend

- If too many slices,
or too many very small slices,
even Figure 4 solution has overlaps
- A legend is always reliable
- Coding more complex & lengthy
- Deferring until Figure 19

Figure 5: 3D Bar Chart. **Needless Complexity.**

Units Sold by Brand in 2002

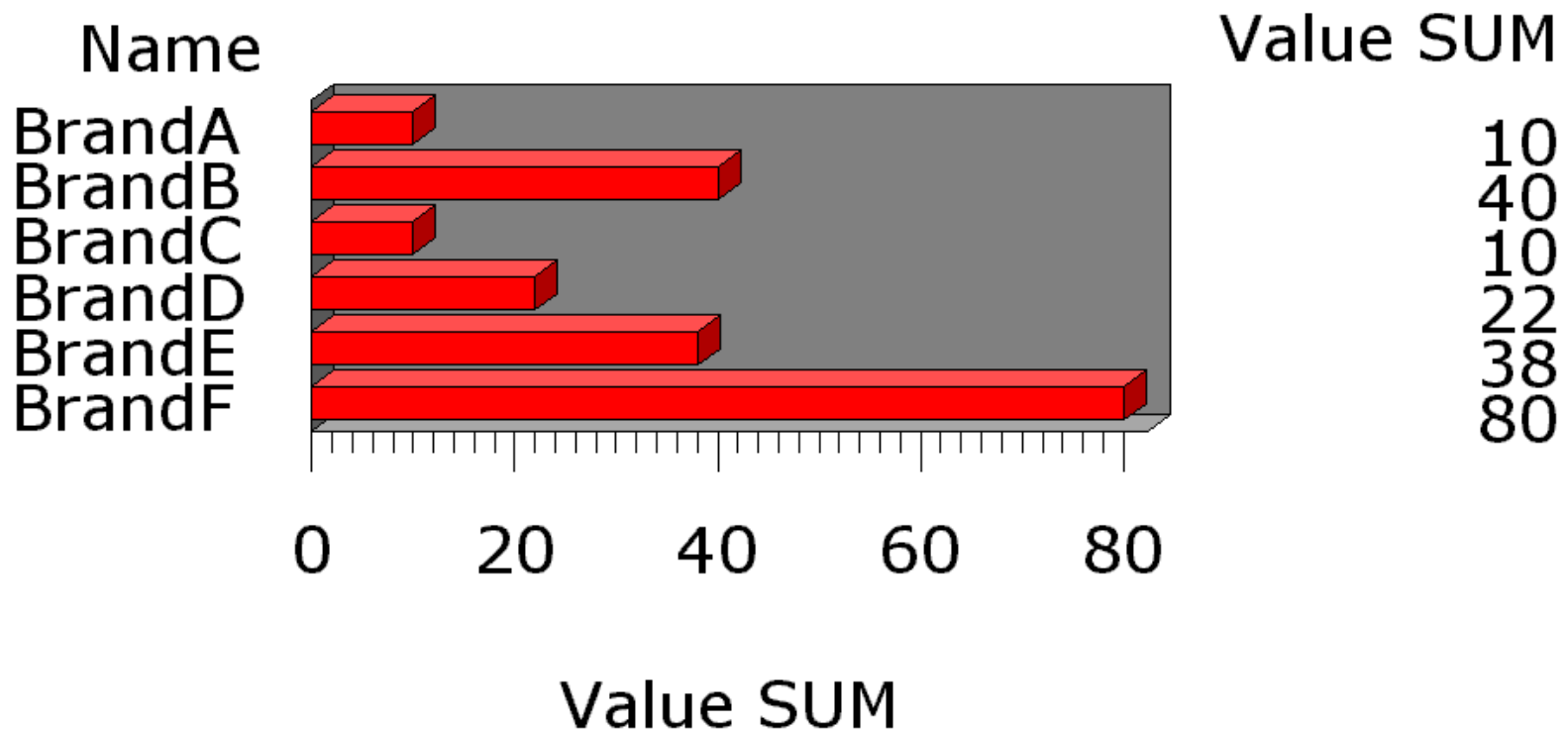
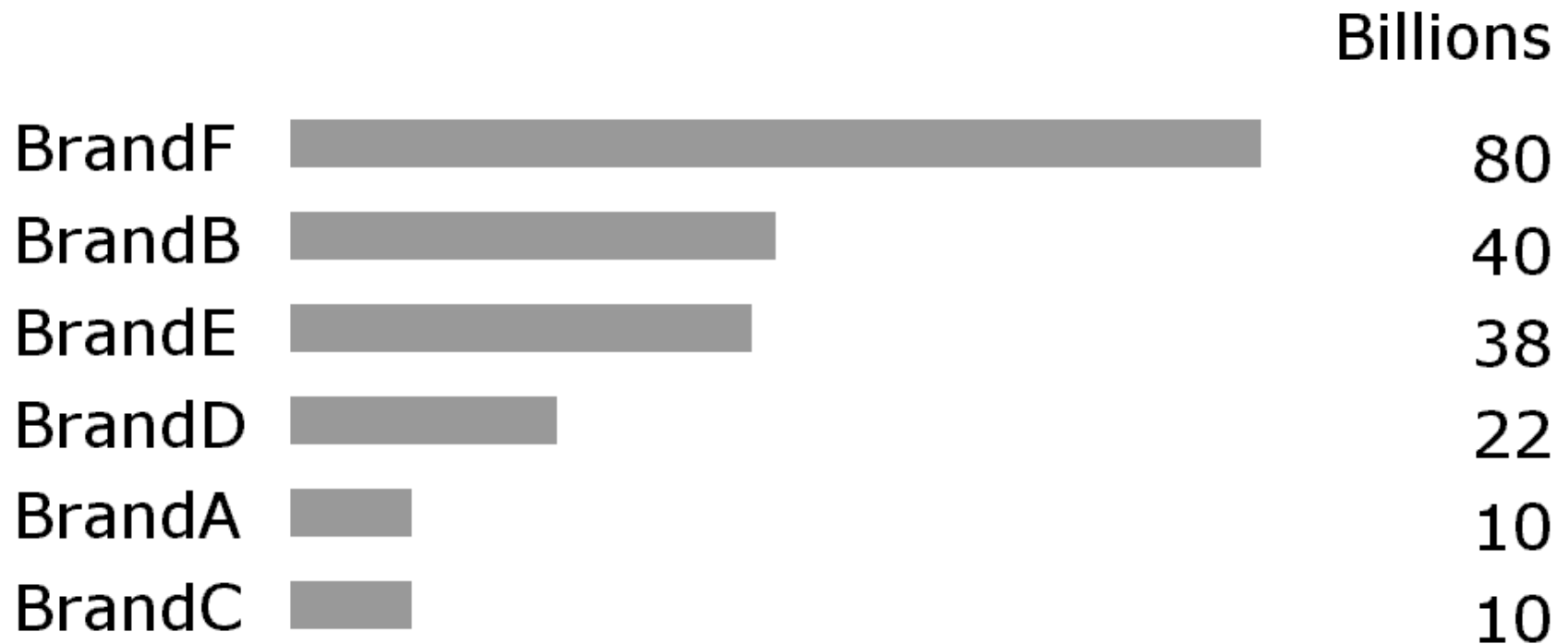


Figure 6: Best Horizontal Bar Chart.

Units Sold by Brand in 2002



```
pattern1 v=solid c=CX999999 r=6;
/* CX999999 light browser-safe gray */

axis1 /* vertical axis */
label=none /* no axis label */
major=none /* no ticks at value steps */
minor=none /* no ticks in between */
style=0; /* no axis line */

axis2 /* horizontal axis */
label=none major=none minor=none style=0
value=none; /* do not show values on axis
They are in column at right-hand margin */
```

```
proc gchart data=DataForSimpleCharts;  
hbar Name /      /* Name is "midpoint" var */  
    sumvar=Value /* Value is "response" var */  
    sumlabel='Billions' /* label for column  
                        of Values */  
width=0.6 /* width of bars */  
space=0.6 /* spacing between bars */  
maxis=axis1 /* axis1 statement  
            for midpoints */  
raxis=axis2 /* axis2 statement  
            for responses */  
  
    descending;  
run; quit;
```

Bar Chart Axis Nomenclature

- If both vars numeric, one axis is forced to be midpoints of subranges of values
- SAS Bar Charts must ALWAYS define
 - one axis as “maxis” (midpoint axis)
 - other axis as “raxis” (response axis)
- In most business bar charts, one axis is text (maxis), other is numeric (raxis)

Figure 7: Pie Chart Alternative.

Horizontal Bar Chart Can List Percent of Total.

Units Sold and Market Share by Brand in 2002

		Billions	Percent
BrandF		80	40
BrandB		40	20
BrandE		38	19
BrandD		22	11
BrandA		10	5
BrandC		10	5

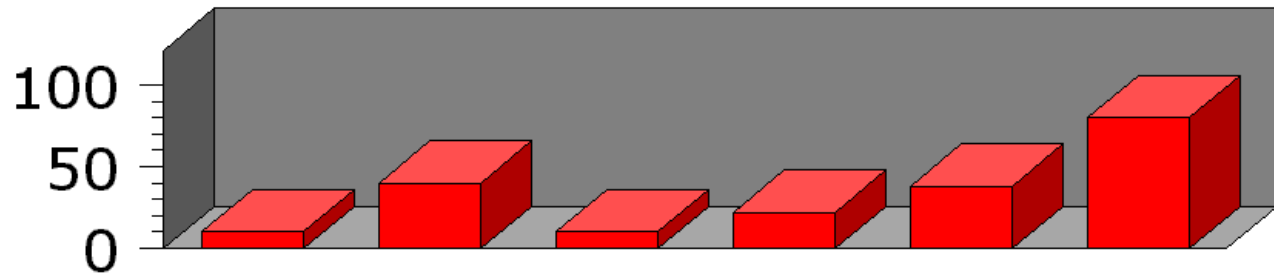
Useful when a pie chart is infeasible

```
proc gchart data=DataForSimpleCharts;  
  
hbar Name /  
    freq=Value /* pretend that Value is a  
                frequency in order to: */  
    freq /* get column of "frequencies" */  
    freqlabel='Billions'  
    percent /* get column of percents */  
    percentlabel='Percent'  
    width=0.5 space=0.5  
    maxis=axis1 raxis=axis2 descending;  
  
run; quit;
```

Figure 8: 3D Bar Chart. **Difficult to Use.**

Units Sold (in Billions) by Brand in 2002

Value SUM

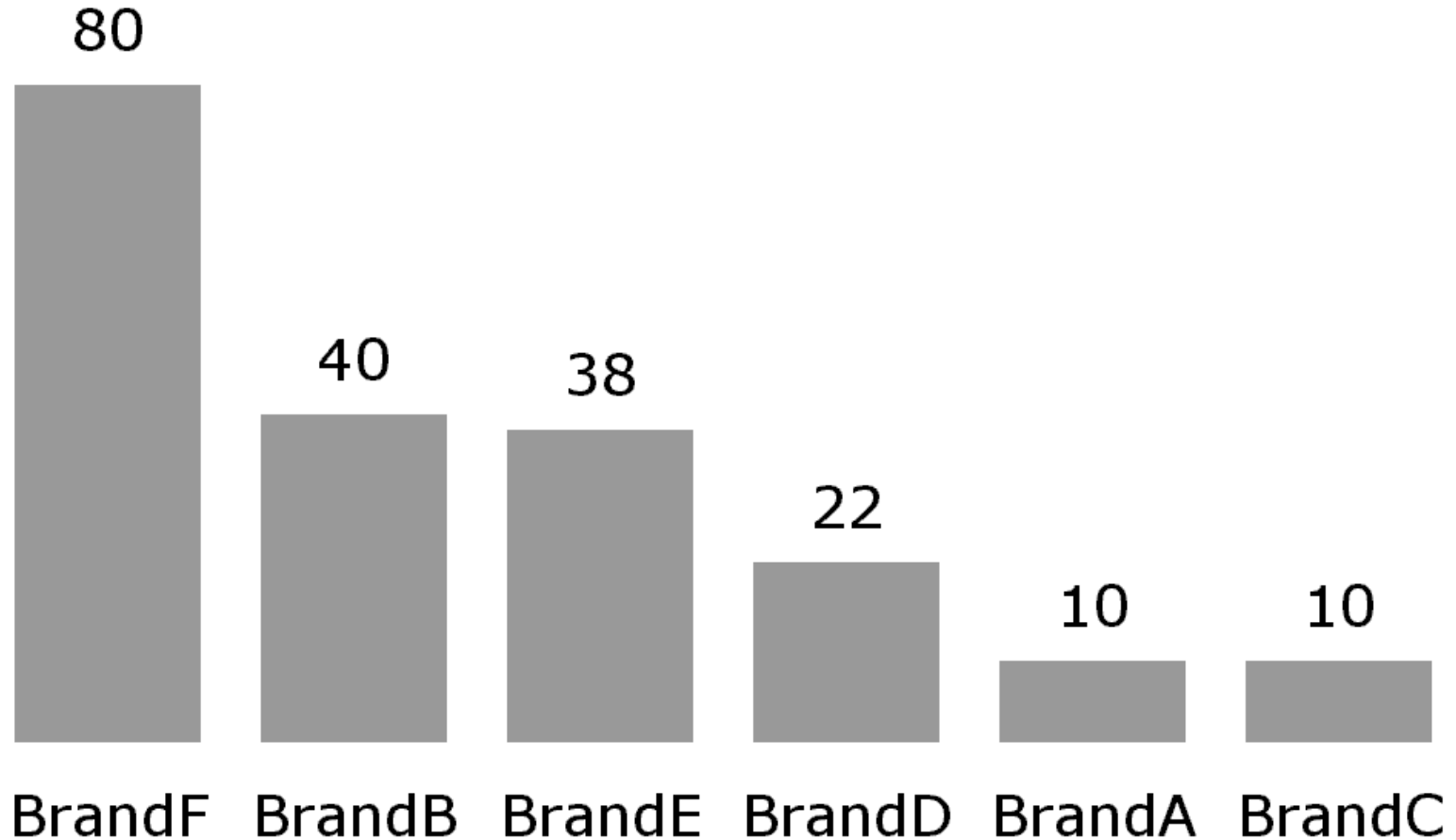


B	B	B	B	B	B
r	r	r	r	r	r
a	a	a	a	a	a
n	n	n	n	n	n
d	d	d	d	d	d
A	B	C	D	E	F

Name

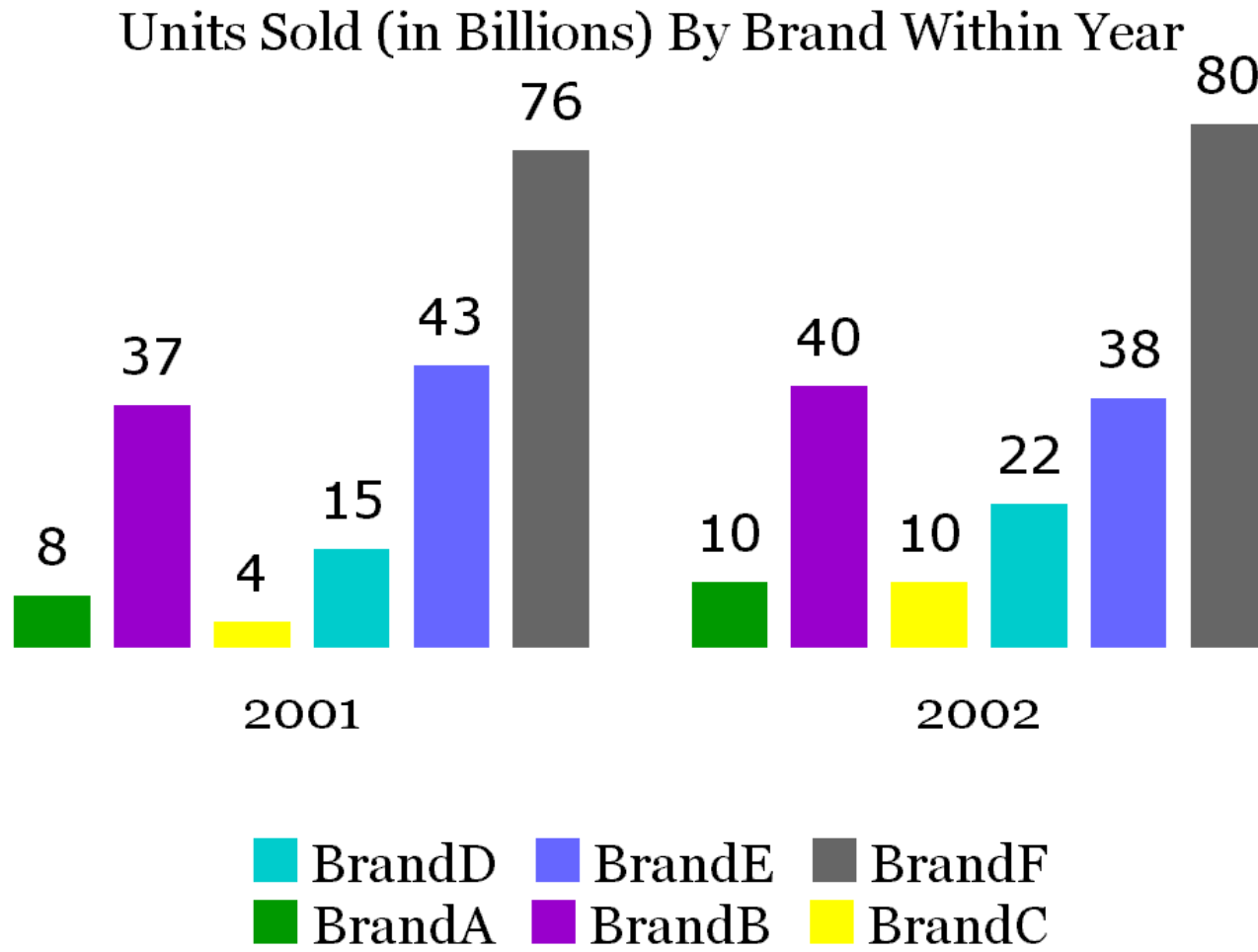
Figure 9: Best Vertical Bar Chart.

Units Sold (in Billions) by Brand in 2002



```
proc gchart data=DataForSimpleCharts;  
  
vbar Name /  
    sumvar=Value  
    sum /* display Value above each bar */  
    maxis=axis1  
    raxis=axis2  
    descending;  
  
run; quit;
```

Figure 10: Side-By-Side Bar Chart with Custom Legend.
Legend unavailable as SAS/GRAPH Option here.
You Need a Legend When the Bar Labels Are Too Long.



```
footnote1      /* create the legend */
h=1.00 j=C
f='Monotype Sorts' c=CX009900 '6E'X
/* '6E'X is solid square */
f='Georgia' c=CX000000 ' BrandA '

f='Monotype Sorts' c=CX9900CC '6E'X
f='Georgia' c=CX000000 ' BrandB '

f='Monotype Sorts' c=CXFFFF00 '6E'X
f='Georgia' c=CX000000 ' BrandC '

j=C /* now start new line */
f='Monotype Sorts' c=CX00CCCC '6E'X
f='Georgia' c=CX000000 ' BrandD '
...;
```

```
pattern1 v=solid c=CX009900; /* dark green */
pattern2 v=solid c=CX9900CC; /* purple */
pattern3 v=solid c=CXFFFF00; /* yellow */
pattern4 v=solid c=CX00CCCC; /* dark cyan */
pattern5 v=solid c=CX6666FF; /* light blue */
pattern6 v=solid c=CX666666; /* dark gray */
```

*SAS/GRAPH applies PATTERNx
statements in sort sequence
of the values of SUMVAR*

```
axis1 /* horizontal axis */  
label=None major=None minor=None style=0  
value=None; /* Names in legend instead */  
  
axis2 /* vertical axis */  
label=None major=None minor=None style=0  
value=None  
offset=(0.5,0); /* extra separation between  
                bottom of bars & the horizontal axis */
```

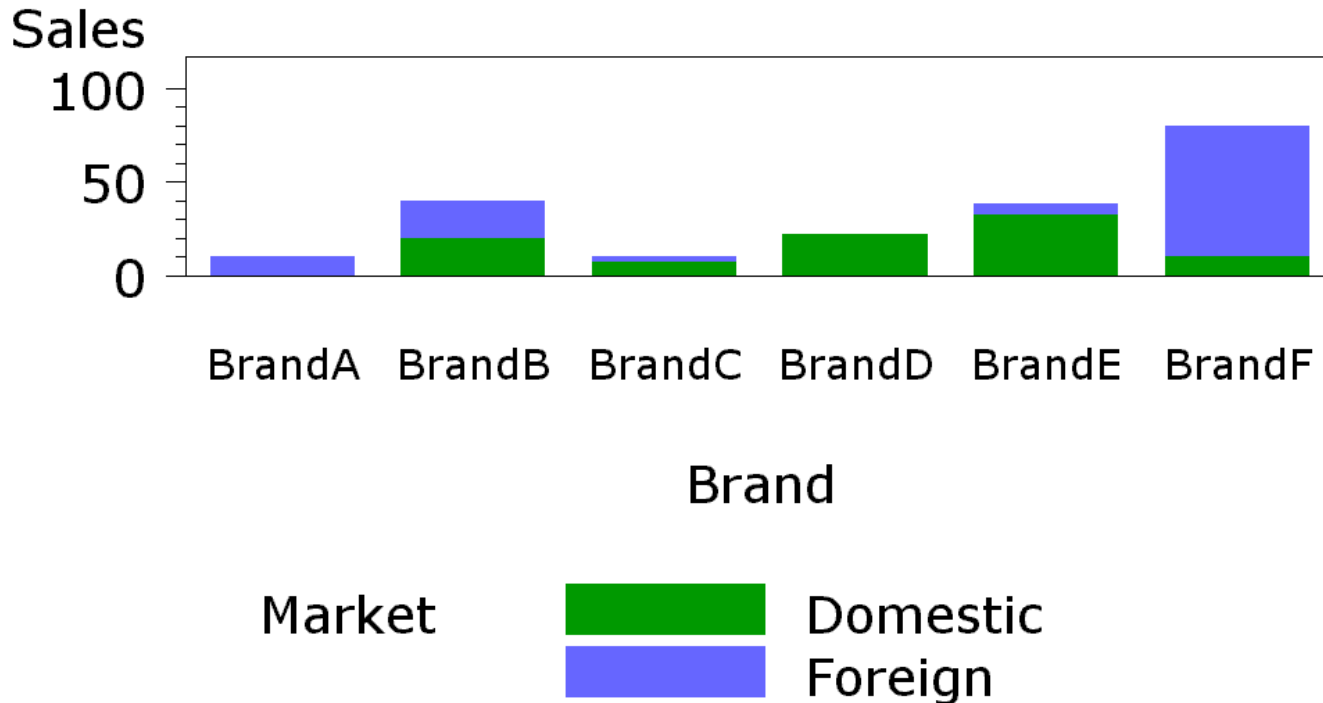
```
axis3 /* group axis */
      label=none
      value=(f='Georgia') /* override FTEXT= */
      nobrackets; /* run without nobrackets
                    option to see why it should be used */

proc gchart data=UnitsSoldByBrandAndYear;
vbar Brand / sumvar=Value sum
           maxis=axis1 raxis=axis2
           group=Year
           /* group midpoints (Brands) by Year */
           gaxis=axis3
           patternid=midpoint;
           /* vary pattern (color) by Brand */

run; quit;
```

Figure 11: **Stacked Bar Chart** Subdivides Comparison.
NOT an effective tool.*

Units Sold (in Billions) By Market Within Brand in 2002



***How easily and reliably estimate value for Upper Bar?**

Solution

- Preprocess the data
to compute totals
- Use a Side By Side Bar Chart

Figure 12: Best Bar Chart for Parts and Their Total.

Units Sold (in Billions) By Market Within Brand in 2002

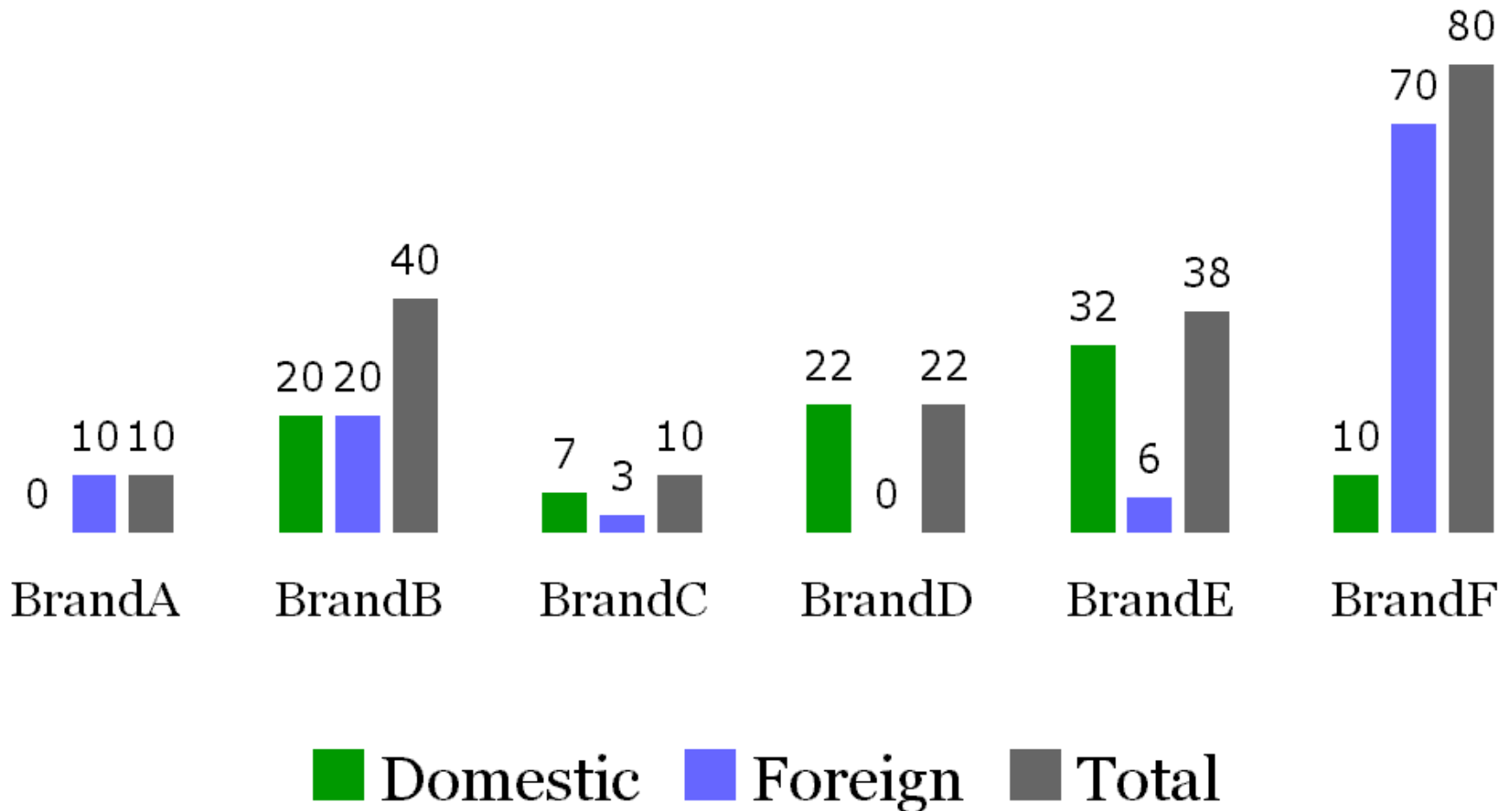
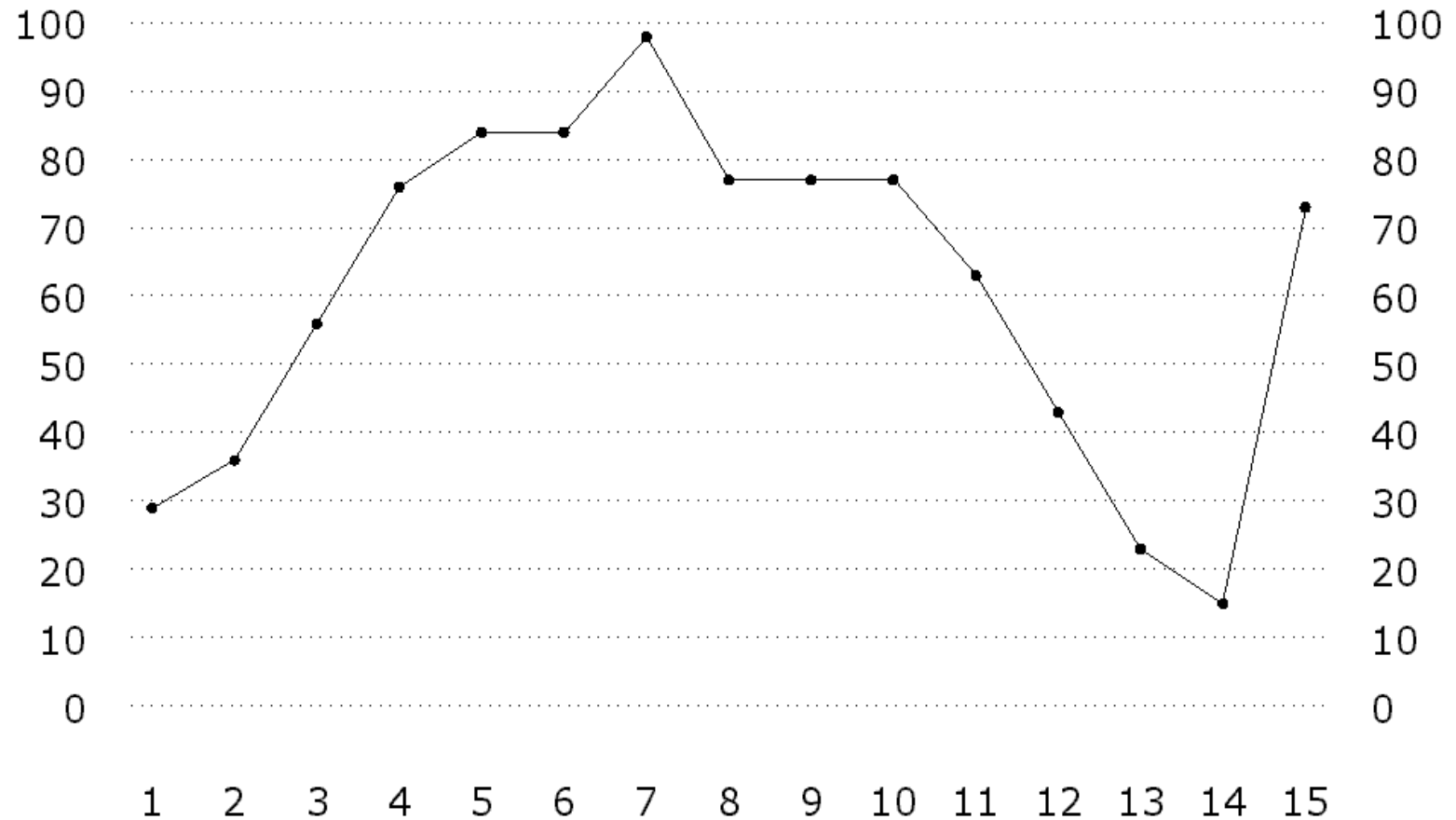


Figure 13: Fine lines to facilitate estimation of y-values.

Sales (in 1000's of \$) by Day During April 2003



```
axis1 /* horizontal axis */
label=none major=none minor=none style=0
value=(h=0.70); /* override HTEXT= */

axis2 /* left-hand vertical axis */
label=none minor=none style=0
major=(c=CXFFFFFFF) /* make ticks invisible */
value=(h=0.70);

axis3 /* right-hand vertical axis */
label=none minor=none style=0
major=(c=CXFFFFFFF)
value=(h=0.70 j=L); /* left justify values */

/* CXFFFFFFF is White,
   default background is White */
```

```
symbol1 v=dot    /* plot with dots */  
          h=0.25  /* shrink them */  
          i=join; /* connect them */
```

no **c=**

default symbol color is Black

to use a font character as your marker,
specify **f=YourChoiceOfFont**

v=' _ '

where **_** is the character you want

```
proc gplot data=DataForTrendPlot;

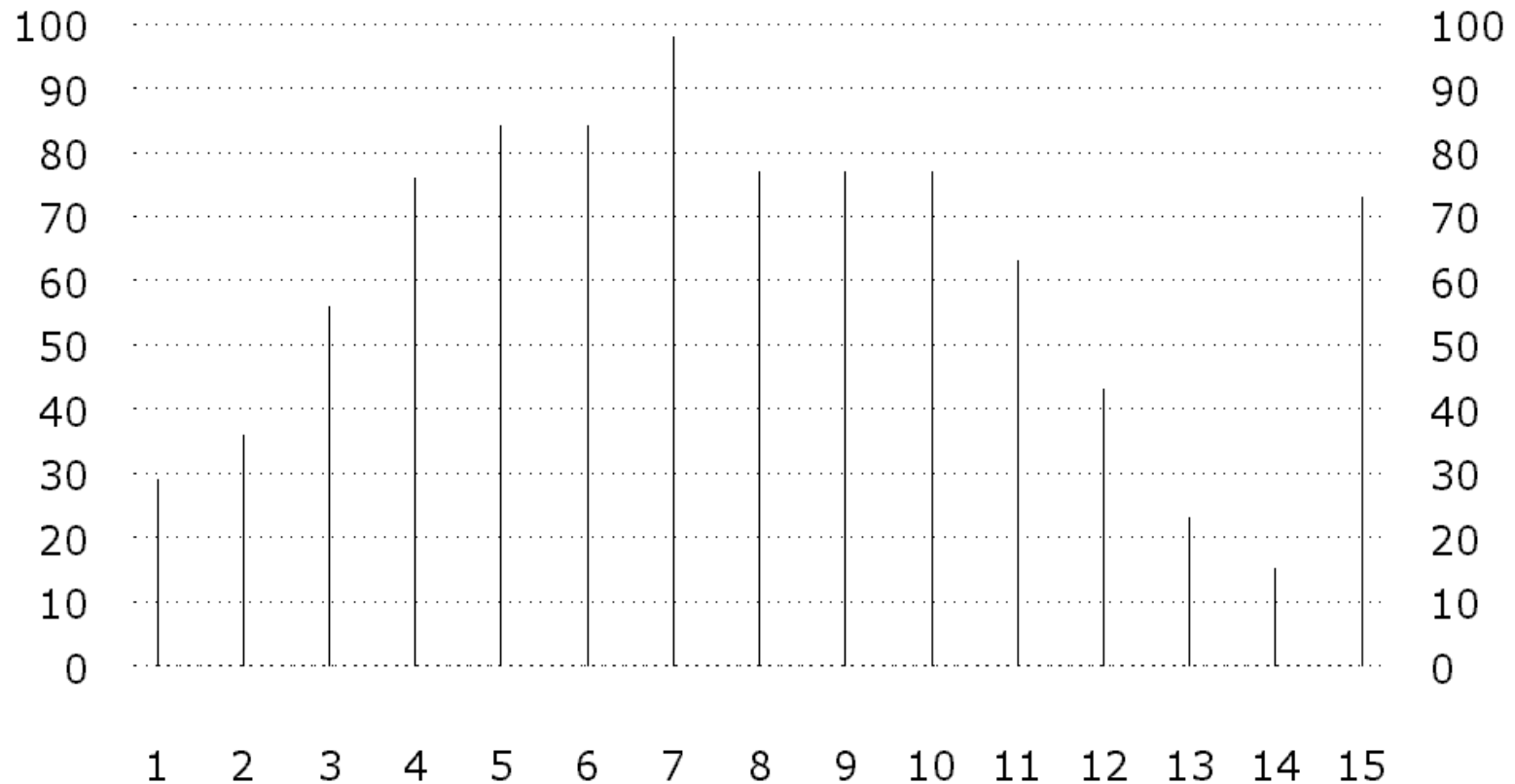
plot Sales*DayOfMonth /
    haxis=axis1 vaxis=axis2
    vzero      /* start vertical axis at zero */
    autovref   /* ref lines at major y ticks */
    lvref=33;  /* line type 33 for ref lines */

plot2 /* force right-hand vertical axis,
        but do the same plot */
    Sales*DayOfMonth=1 /* use symbol1 */
    /
    vaxis=axis3
    vzero;

run; quit;
```

Figure 14: Unambiguously identifies x-values,
and uses tiny dots for most precise y-value estimate.

Sales (1000's of \$) by Day During April 2003



Everything the same as for Figure 13,
except:

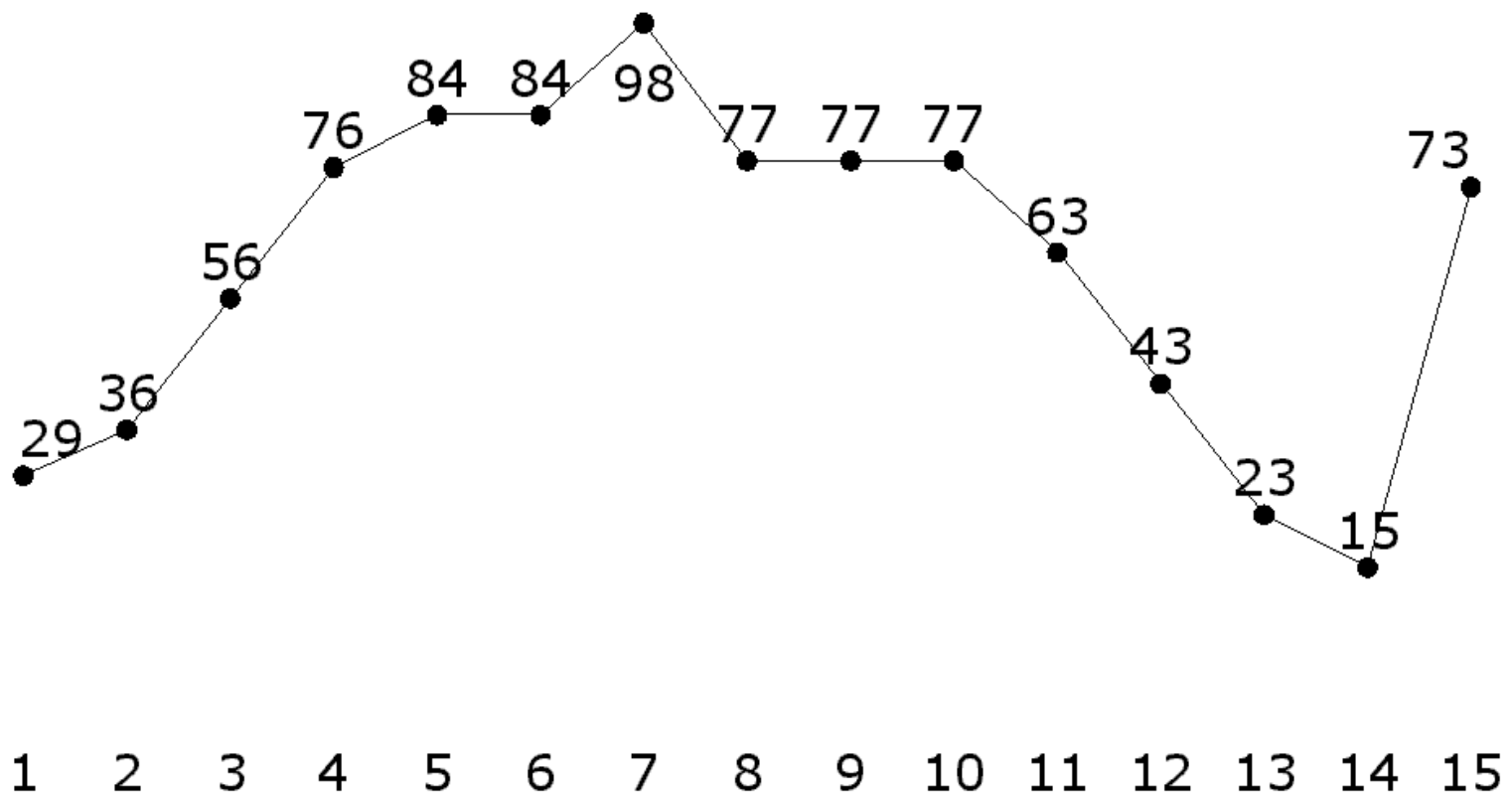
```
symbol1 v=point i=needle;
```

Better Than Estimation and Reference Lines

- Annotation of actual values
- But avoid collisions

Figure 15: Problem-prone POINTLABEL option.
Permits collisions of some labels with the plot line.

Sales (1000's of \$) by Day During April 2003



Collision Avoidance Needed

- We want Graph and Detail
 - Graph = quick easy general inference
 - Detail = reliable precise inference
- On web, ALT text for temporary display, or hyperlink to static printable table
- We want non-web solution

Collisions Avoided

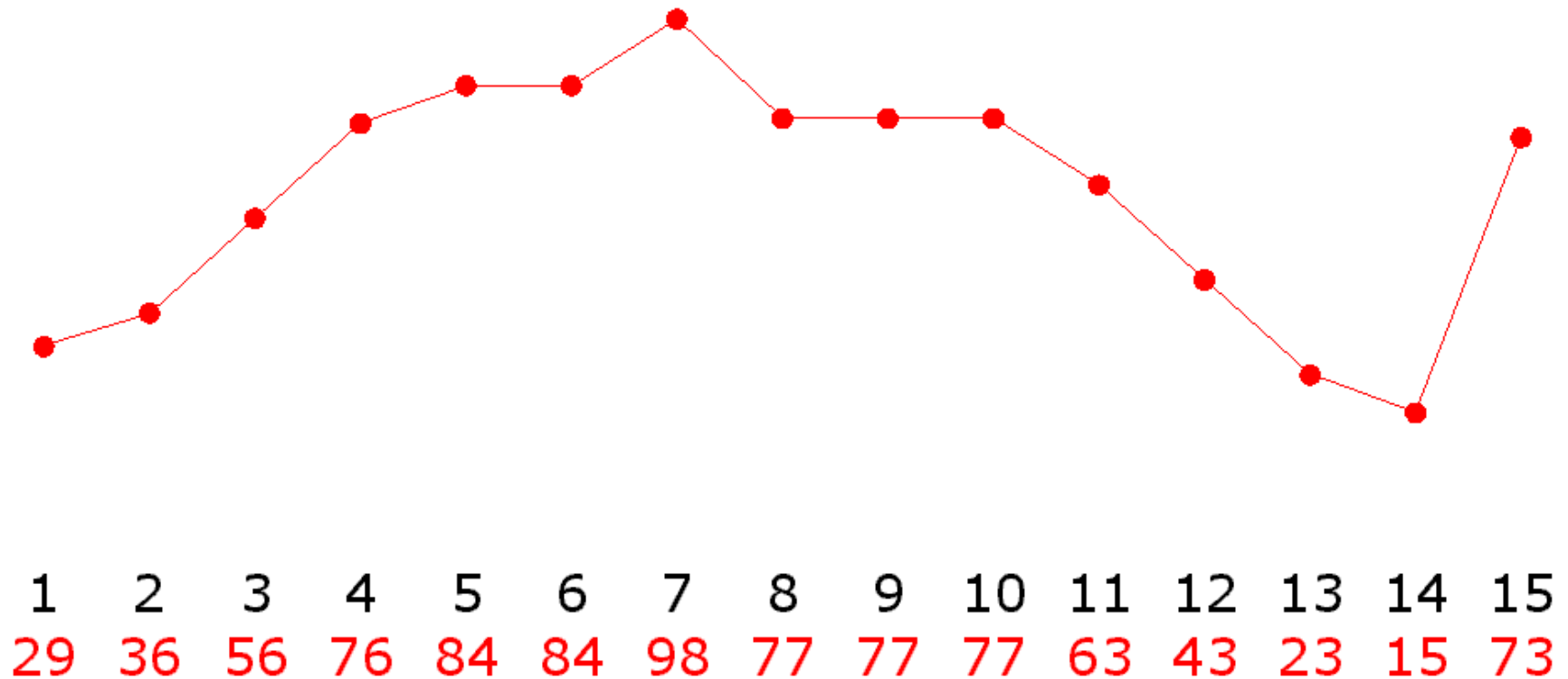
- **%TREND macro** (not discussed here)
collision-resistant annotation
- Today, “Annotate” with y-values
below their corresponding x-values
along horizontal axis

Tick Mark Value Table

- Uses “Software Intelligence”
- Preprocess the data
- x-axis tick mark labels dynamically created from macro variables
- Extendable to multi-line trends

Figure 16: Color-Coded Line & Tick Mark Value Table.
Visual Image of Trend PLUS Precise Detail Values.

Sales (in 1000's of \$) by Day During April 2003



```
data _null_;  
set DataForTrendPlot end=lastobs;  
call symput('y_value' || trim(left(_N_)),  
            trim(left(Sales)));  
call symput('x_value' || trim(left(_N_)),  
            trim(left(DayOfMonth)));  
if lastobs; call symput('ValCount', _N_);  
run;
```

```
%macro maketks;  
%do i = 1 %to &ValCount;  
tick=&i "&&x_value&i"  
      j=C  
      c=CXFF0000  
      "&&y_value&i"  
%end;  
%mend maketks;
```

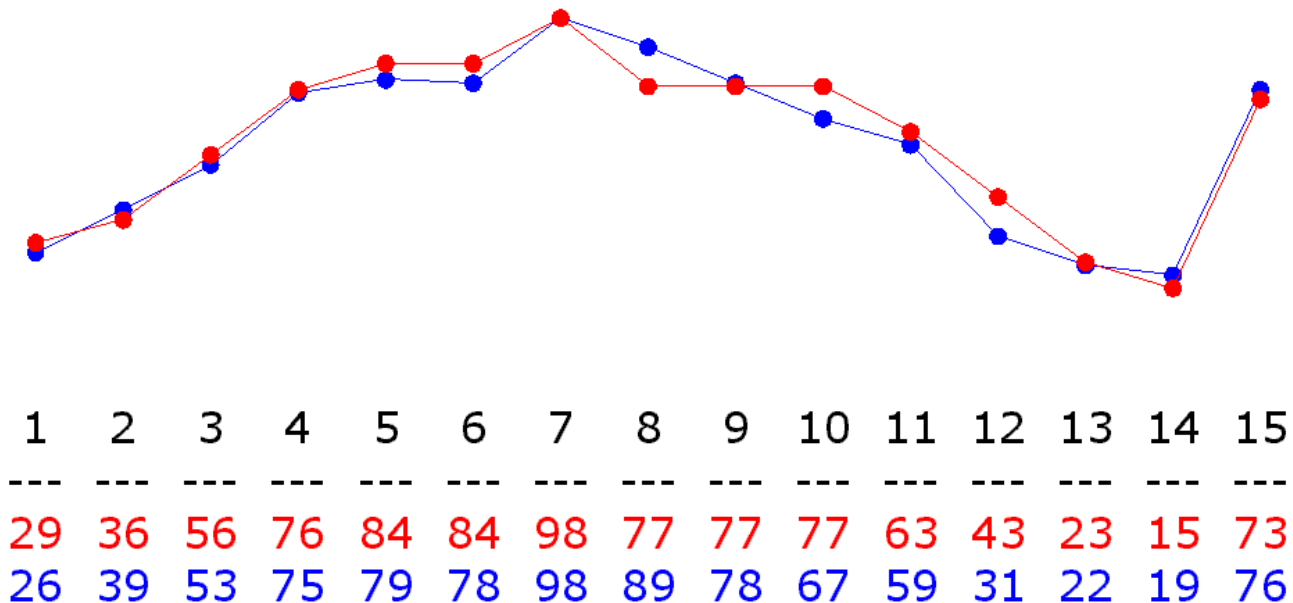
```
axis1 label=none major=none minor=none style=0  
      value=(h=0.85  
            %maketks); /* supply values dynamically */  
axis2 label=none major=none minor=none style=0  
      value=none; /* no values on y-axis */  
symbol1 v=dot h=0.50 i=join  
        c=CXFF0000; /* use red dots to match  
                    display of y-values  
                    below x-axis tick values */  
proc gplot data=DataForTrendPlot;  
plot Sales*DayOfMonth /  
      haxis=axis1 vaxis=axis2 vzero;  
run; quit;
```

If you use OPTIONS MPRINT,
resolved code in SAS log:

```
axis1 label=none major=none minor=none style=0  
value=(h=0.50  
      tick=1    "1"  j=C  c=CXFF0000  "29"  
      tick=2    "2"  j=C  c=CXFF0000  "36"  
      tick=3    "3"  j=C  c=CXFF0000  "56"  
      . . .  
      tick=15   "15" j=C  c=CXFF0000  "73");
```

Figure 17: Color-Coded Trends & Tick Mark Value Table.
Multi-Line Multi-Class Plot.

Sales (in 1000's of \$) by Day During April 2003
and Tabular Comparison with Previous Month



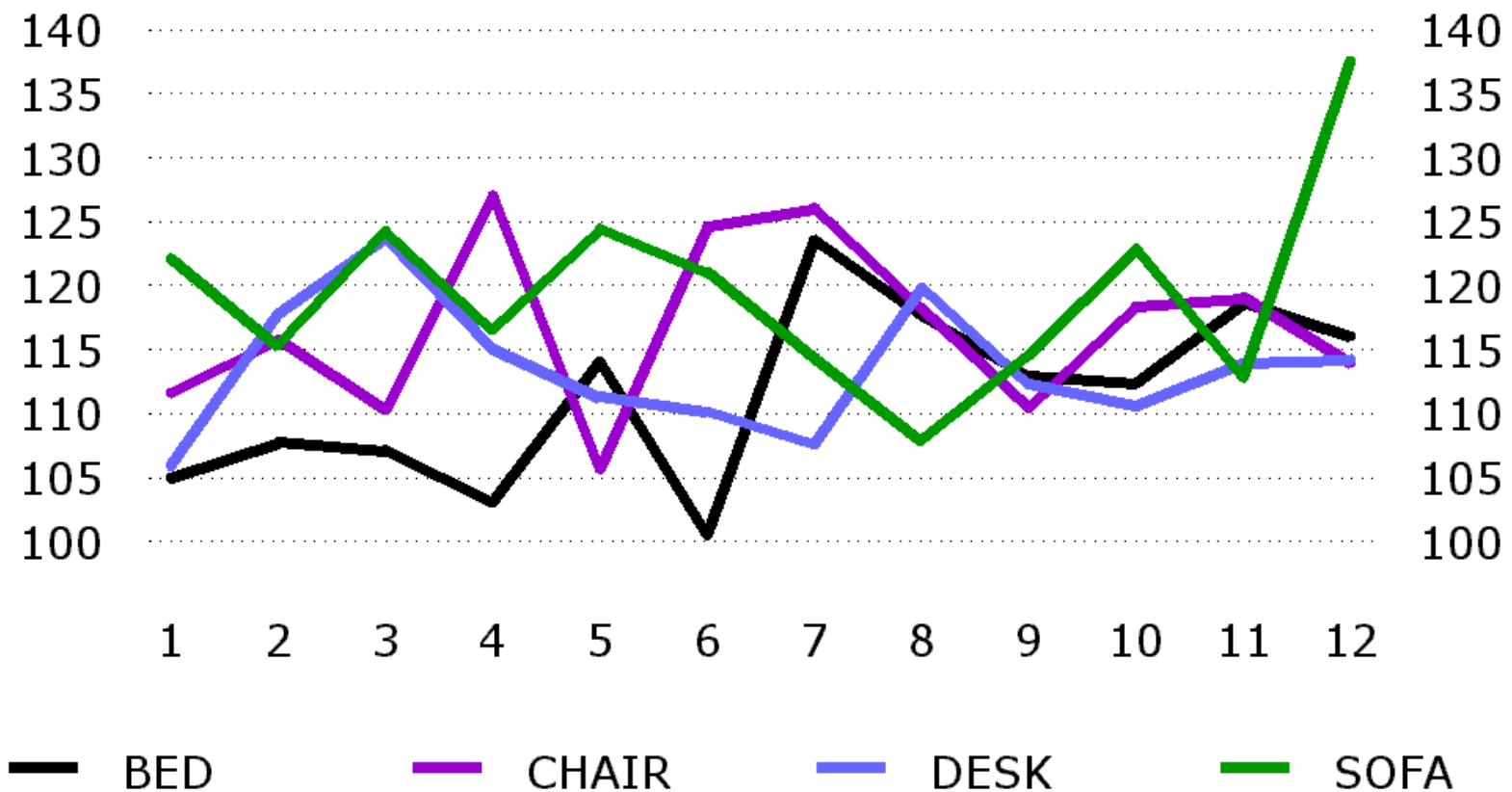
Day of Month

Sales During Report Month

Sales During Previous Month

Figure 18: Multi-Line Multi-Class Plot With Legend.
Too many lines for tick mark value table to be feasible.

Sales by Product Category by Month During 1998



Colored lines must be **thick**.
(Also, colored text.
In these graphs, all text is black.)

```
axis2 /* left-hand vertical axis */
label=none
major=(c=CXFFFFFFF) /* invisible ticks */
minor=none style=0 value=(h=0.75)
order =      /* specify the tick marks */
        100 to 140 /* start to end */
        by 5; /* with this increment */

axis3 /* right-hand vertical axis */
label=none major=(c=CXFFFFFFF)
minor=none style=0 value=(h=0.75 j=L)
order = 100 to 140 by 5;
```

```
/* no markers, but solid lines */
/* very thick lines : w=6 */
symbol1 v=none i=join w=6 c=CX000000; Black
symbol2 v=none i=join w=6 c=CX9900CC; Violet
symbol3 v=none i=join w=6 c=CX6666FF; Blue
symbol4 v=none i=join w=6 c=CX009900; Green

/* for PLOT2 statement: */
symbol5 v=none i=none r=4; /* use 4 times */
/* no markers and no lines */
```

*For multi-line plot, as in plot $y*x=z$,
SYMBOLn statements used in sort sequence
of values of z variable,
as in: Bed, Chair, Desk, Sofa.*

When PLOT2 statement is present,
next-numbered SYMBOLn statements are used.

legend1

```
label=None /* no title */
```

```
/* controls for legend entry text: */
```

```
value=(h=0.75) /* override HTEXT= */
```

```
/* controls for line samples: */
```

```
shape=line(1.5); /* shorter line segments */
```

```
/* format is line(length,thickness) */
```

```
/* default is line(5,1) */
```

Default action--

*legend entries displayed in sort sequence
of values of z variable,*

as in: Bed, Chair, Desk, Sofa.

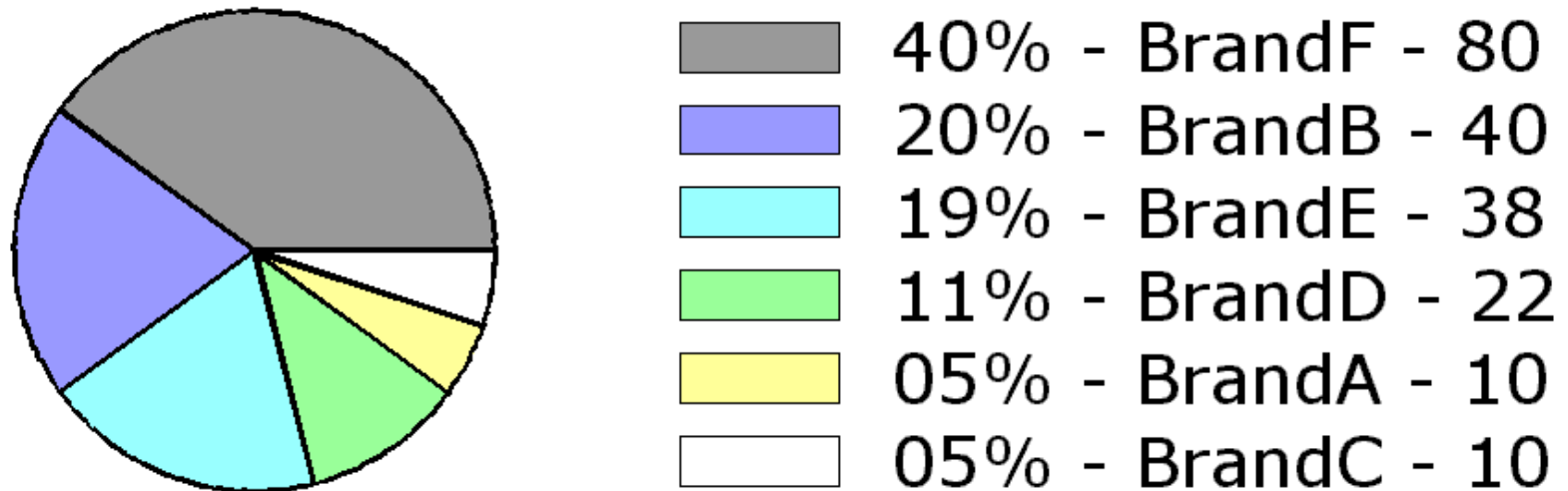
Can override sequence with order=

Can override legend text with value=

```
proc gplot data=DataForFourLineTrendPlot;  
plot Sales*Month=Product /  
/* one plot line for each value of Product */  
    haxis=axis1 vaxis=axis2  
    autovref lvref=33  
    legend=legend1;  
  
plot2          /* right-hand vertical axis */  
    Sales*Month=Product /  
    vaxis=axis3  
    nolegend; /* another legend not needed */  
run; quit;
```

Figure 19: Best Pie Chart With Legend.

Market Share, Brand, and Sales in Billions of Units



Solution for Best Pie Chart

- Too lengthy to present here
- Uses “Software Intelligence”
 - Data is preprocessed
 - Macro variables are created
 - Then used to dynamically build
LEGEND and **SYMBOLn** statements

Tip: Persistence of Options

- If you create multiple graphs in one session, statements for PATTERN, AXIS, SYMBOL, LEGEND, etc., and all GOPTIONS, remain in effect, unless replaced or reset
- GOPTIONS RESET=
 - RESET=PATTERN (selective example)
 - RESET=ALL (everything)

Notice

SAS & SAS/GRAPH are trademarks or registered trademarks of SAS Institute Inc. in the USA and other countries. ® denotes USA registration.

Your Feedback:

- Your questions, requests
- Which of Bessler's Best
can/should be made Better
- Other Problems
- Other Solutions

Email & Telephone

LeRoy Bessler PhD

bessler@execpc.com

+1 414 351 6748

(USA PM / weekend

6 hours earlier than GMT)