PharmaSUG 2026 - Paper DV-409

Be a Multi-Media Wizard - Make Your Output Dance and Sing

LeRoy Bessler, Bessler Consulting and Research

ABSTRACT

Static and simple presentation of information always adequately communicates, but SAS does offer capabilities to take advantage of more communication channels and methods. Among possibles are audio, video, animation, a marquee for your web page (the analogue for that traveling text at the bottom of a television news program—where it is called a chyron, not a marquee), things that you can do with images (different ways you can put them on a web page), examples of 3D graphics that actually communicate versus the bad examples that ALWAYS distort the information, and even the odd phenomenon of making a line of text twinkle.

INTRODUCTION

I am showing you the 2026 updates and enhancements of a presentation for which the audience filled a large room to Standing Room Only. For a paper, no live demo of audio, video, or animation is possible. Here you will see static screen captures, accompanied by description of what you could see and hear if you attended the reprise in 2026. For the examples, code is included. A zip file of ALL code and needed inputs (data, audio, animated GIF, or static image) are available upon email request. The specific video used is too large for email, but you should be able to substitute any MP4 that you have.

After sections on working with audio, video, animated GIFs, RTF Animation, and the web-page hosted marquee (both horizontal and vertical . . . for proof of concept, a vertical marquee with an imbedded animated GIF), the paper takes up more prosaic concerns as 3D (USABLE, communication-effective 3D) and a variety of ways to imbed and display images in a web page (not data graphics, but logos, photos, diagrams, etc., anything visual to add communication value).

FOUR INTERLINKED WEB PAGES OF MULTI-MEDIA WIZARDRY

Depending on the case, the web page may include also:

a hyperlink to one or more other places (not just to the next web page in the demonstration)

mouseover text for any of the images included

one or more animated GIF (Graphic Interchange Format) image files

an audio clip

a video clip (which includes its own audio)

a marquee (also called a chyron, which is the usual name for the traveling text at the bottom of a TV screen), either a horizontal marquee like that on TV, or a vertical marquee (which here happens to include an animated GIF, which could have been displayed in the web page outside the marquee).

Excerpts from the four-part live show follow . . .

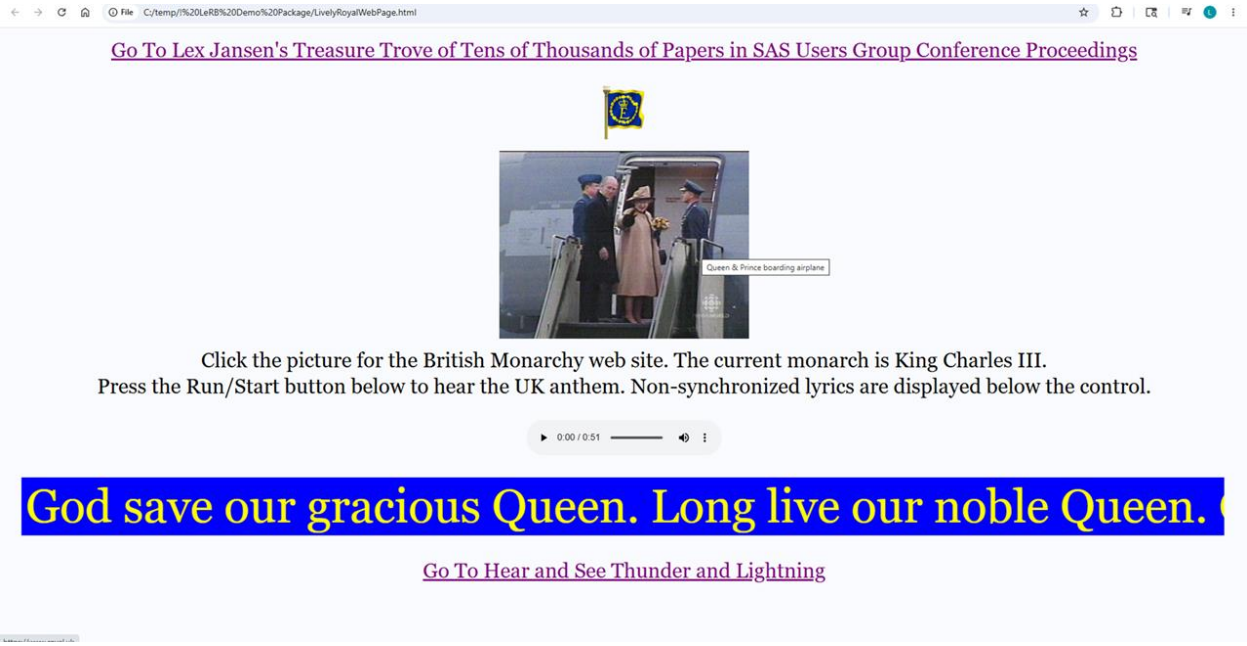

Figure 1. A Lively Melodious Royal Web Page: Queen and Prince neither dance, nor sing. There is mouseover text displayed near the middle right of the main picture. Resting the mouse on the picture caused a display at the bottom left of the web browser of the destination link where a click on the picture will take you. The picture of the Queen's Flag (so identified as such if the mouse is rested there) actually shows the flag waving due the picture being an animated GIF (Graphics Interchange Format image). In 2026 (unlike 2006), the auto-run feature does not work for audio files. So the run button is available for user initiation of the UK anthem. The marquee (or chyron) to display the lyrics to the UK anthem runs indefinitely in a closed loop.

```
proc template;
define style styles.NoPageBreaksWithinWebPage;
  parent=styles.htmlencore; /* default style for ODS HTML5 */
  class pagebreak / display=none;
end;
run;

%let path = C:\temp\! LeRB Demo;

ods results off;
ods _all_ close;
title; footnote;
options center;
ods escapechar='^';
ods html5
  style=styles.NoPageBreaksWithinWebPage
  path="&path" (url=none)
  body='LivelyRoyalWebPage.html'
  (title='HRH Elizabeth II and Husband Returning Home');

data _null_;
dcl odsout obj();
obj.format_text(
  data: "Go To Lex Jansen's Treasure Trove of Tens of Thousands of Papers in SAS Users
Group Conference Proceedings",
  style_attr: 'fontfamily=Georgia fontsize=24pt just=C',
  /* in style_attr, color=SomeValidColor would be disregarded */
  url: "https://www.lexjansen.com/");
obj.image(
  file: "WavingFlagOfQueenElizabethII.gif",
```

```
  description: "The Flag of Queen Elizabeth II",
  /* DESCRIPTION is for Screen Reader, NOT for mouseover text */
  style_attr: 'flyover="The Flag of Queen Elizabeth II"');
run;

proc odstext;
p ""/style={preimage="RoyalCoupleBoardingAirplane.jpg?width=4in&height=3in"
                      url='https://www.royal.uk/'
                      flyover='Queen & Prince boarding airplane'
                      just=C};
p "Click the picture for the British Monarchy web site. The current monarch is King Charles
III. ^n Press the Run/Start button below to hear the UK anthem. Non-synchronized lyrics are
displayed below the control."
  / style={just=C fontfamily=Georgia fontsize=24pt};
p "<audio controls=""controls"" preload=""auto"">
  <source src=""UnitedKingdom_Anthem.wav"" type=""audio/wav""/></audio>" /
  style={just=C};
run;
/* width=1460 hspace=25 vspace=13 */
ods text=
"<font face=Georgia size=7 color=Yellow>
 <marquee bgcolor=blue width=1850 hspace=25>God save our gracious Queen. Long live our
noble Queen. God save the Queen! Send her victorious, happy, and glorious, long to reign
over us. God save the Queen!</marquee>
 </font>";
/* NOTE: hspace=25 is meant to try to center the Marquee display box with width=1850 and
when displayed on my laptop's 17-inch diagonal screen.
Vspace=13 is meant to, on MY laptop's screen, assure approximately the same height of white
space above and below the Marquee display box. */

data _null_;
dcl odsout obj();
obj.format_text(
  data: "Go To Hear and See Thunder and Lightning",
  style_attr: 'fontfamily=Georgia fontsize=24pt
               just=C',
  url: "./ThunderAndLightning.html");
run;

ods html5 close;
```

Code for Figure1.

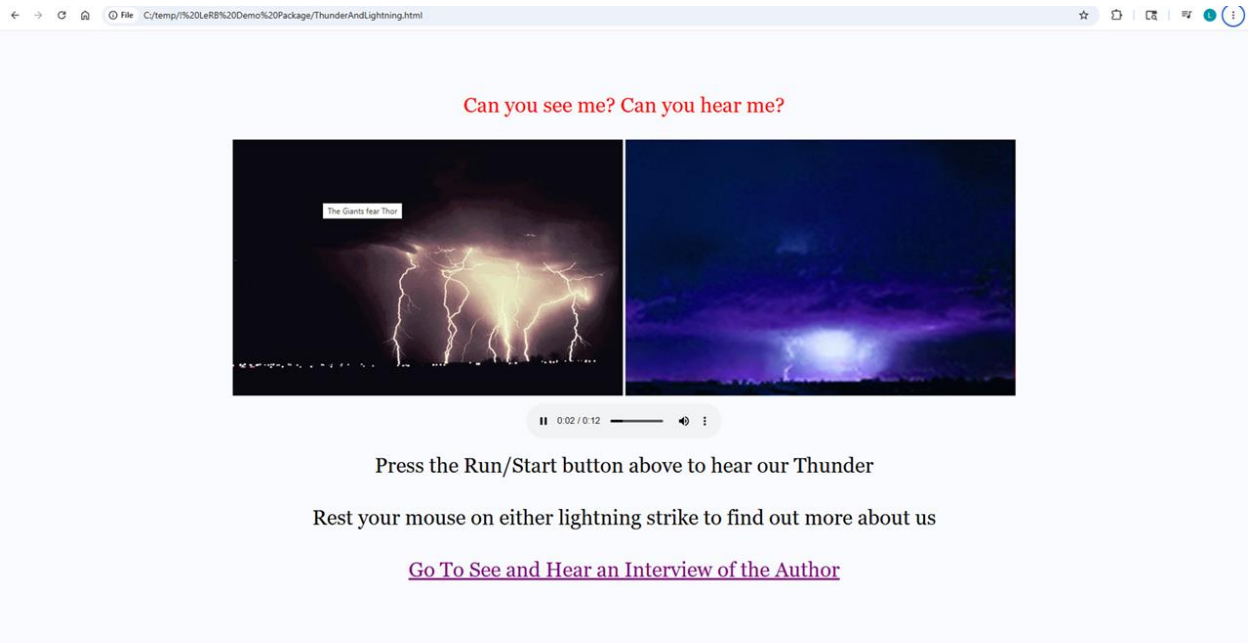

Figure 2. Thunder and Lightning: The most exciting multi-media demo on show today, even the faint of hearing (like your presenter) are likely to hear it. One incessant crashing rumbling closed loop audio track (associated with neither storm), two jittering animated GIFs, each with its own mouseover text descriptor (not shown: “Thor is the warder for Midgard”), and a link to the next demo.

```
proc template;
define style styles.NoPageBreaksWithinWebPage;
  parent=styles.htmlencore; /* default style for ODS HTML5 */
  class pagebreak / display=none;
end;
run;

%let path = C:\temp\! LeRB Demo;

ods results off;
ods _all_ close;
title; footnote;
options center;

ods html5
  style=styles.NoPageBreaksWithinWebPage
  path="&path" (url=none)
  body='ThunderAndLightning.html'
  (title='Thunder and Lightning');

ods graphics / reset=all imagemap;

data _null_;
dcl odsout obj();

obj.format_text(
  data: "Can you see me? Can you hear me?",
style_attr: 'fontfamily=Georgia fontsize=24pt color=Red just=C');

ods layout gridded columns=2 column_widths=(600px 600px);

ods region column=1;
obj.image(
  file: "walking_AKA_jellyfishlight1_FromThundrune.gif",
```

```
  description: "The Giants fear Thor",
  style_attr: 'flyover="The Giants fear Thor"
               width=600 height=393'); /* 440 288 */

ods region column=2;
obj.image(
  file: "tempest_AKA_thorstorm_FromThundrune.gif",
  description: 'Thor is the Warder for Midgarde',
  style_attr: 'flyover="Thor is the Warder for Midgard"
               width=600 height=393'); /* 440 288 */

ods layout end;

obj.format_text(
  data: "White Space to force Audio control into an area below the two-column array",
  style_attr: 'fontfamily=Chiller fontweight=bold fontsize=1pt color=white');

obj.audio(
   file: "Thunder.wav",
   type: "wav",
   preload: "auto",
   autoplay: "yes",
   loop: "on" );

run;

data _null_;
dcl odsout obj();
obj.format_text(
  data: "Press the Run/Start button above to hear my Thunder",
  style_attr: 'fontfamily=Georgia fontsize=24pt just=C color=Black');
run;

data _null_;
dcl odsout obj();
obj.format_text(
  data: "Rest your mouse on either lightning strike to find out more about us",
  style_attr: 'fontfamily=Georgia fontsize=24pt just=C color=Black');
run;

data _null_;
dcl odsout obj();
obj.format_text(
  data: "Go To the Video of an Interview at the MWSUG 2010 Conference",
  style_attr: 'fontfamily=Georgia fontsize=24pt just=C',
  url: "./ShellyGoodinInterviewingLeRoyBesslerAtMWSUG2010.html");
run;

ods html5 close;
```

Code for Figure 2.

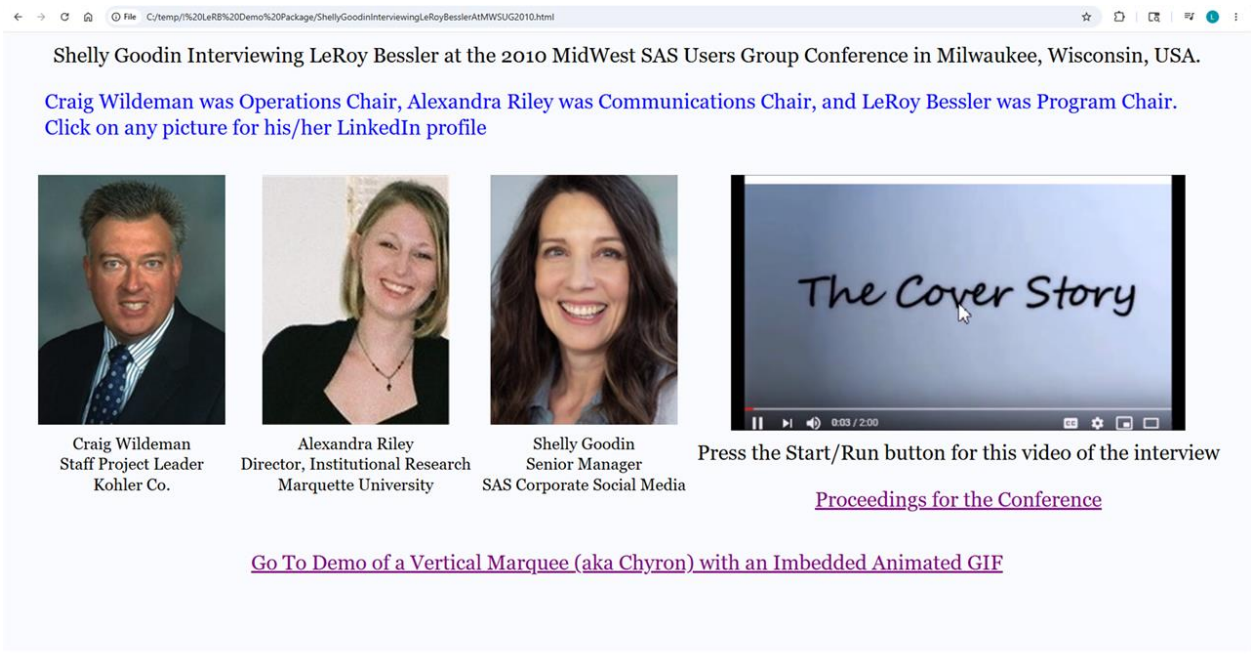

Figure 3. Video of Live Interview at MWSUG 2010: Video with audio, photos with mouseover text and hyperlinks, and link to the MWSUG 2010 Conference Proceedings.

```
proc template;
define style styles.NoPageBreaksWithinWebPage;
  parent=styles.htmlencore; /* default style for ODS HTML5 */
  class pagebreak / display=none;
end;
run;

%let path = C:\temp\! LeRB Demo;

ods results off;
ods _all_ close;
title; footnote;
options center;
ods escapechar='^';

ods html5
  style=styles.NoPageBreaksWithinWebPage
  path="&path" (url=none)
  body="ShellyGoodinInterviewingLeRoyBesslerAtMWSUG2010.html"
  (title="Shelly Goodin Interviewing LeRoy Bessler At MWSUG 2010");

proc odstext;
p "Shelly Goodin Interviewing LeRoy Bessler at the 2010 MidWest SAS Users Group Conference
in Milwaukee, Wisconsin, USA."
  / style={just=C fontfamily=Georgia fontsize=24pt color=black};
p "Craig Wildeman was Operations Chair, Alexandra Riley was Communications Chair, and LeRoy
Bessler was Program Chair. ^n Click on any picture for his/her LinkedIn profile"
  / style={leftmargin=56px just=L fontfamily=Georgia fontsize=24pt color=blue};
run;

ods layout gridded columns=4 column_widths=(300px 300px 300px 570px) column_gutter=5px;

ods region column=1;
proc odstext;
p ""/style={preimage="CraigWildeman3inWide.PNG"
                      url='https://www.linkedin.com/in/craig-wildeman-59171512/'
```

```
                      flyover='Craig Wildeman, Staff Project Leader, Kohler Co.'
                      just=C};
p "Craig Wildeman ^n Staff Project Leader ^n Kohler Co."
  / style={just=C fontfamily=Georgia fontsize=19pt color=black};
run;

ods region column=2;
proc odstext;
p ""/style={preimage="AlexandraRiley3inWide.PNG"
                      url='https://www.linkedin.com/in/alixriley/'
                      flyover='Alexandra Riley, Director of Institutional Research,
Marquette University'
                      just=C};
p "Alexandra Riley ^n Director, Institutional Research ^n Marquette University"
  / style={just=C fontfamily=Georgia fontsize=19pt color=black};
run;

ods region column=3;
proc odstext;
p ""/style={preimage="ShellyGoodin3inWide.PNG"
                      url='https://www.linkedin.com/in/shellygoodin/'
                      flyover='Shelly Goodin, Senior Manager, SAS Corporate Social Media'
                      just=C};
p "Shelly Goodin ^n Senior Manager ^n SAS Corporate Social Media"
  / style={just=C fontfamily=Georgia fontsize=19pt color=black};
run;

ods region column=4;

data _null_;
dcl odsout obj();

obj.video(
 file: "TheCoverStory_11October2010_MilwaukeeWisconsinUSA.mp4",
 type: "mp4",
 width: "700",
 height: "393",
 preload: "auto",
 /* autoplay: "on", AUTOPLAY does not really work. */
 loop: "on"
);

obj.format_text(
  data: "White space gap between audio control and instruction below. Adjust gap height as
desired.",
  style_attr: 'fontsize=1pt color=white' );

obj.format_text(
  data: "Press the Start/Run button for this video of the interview",
  just: "C",
  style_attr: "color=blue fontfamily=Georgia font_size=24pt");

obj.format_text(
  data: "Proceedings for the Conference",
  just: "C",
  style_attr: "color=blue fontfamily=Georgia font_size=24pt",
  url: "https://www.lexjansen.com/cgi-bin/xsl_transform.php?x=mwsug2010");

run;

ods layout end;

data _null_;
dcl odsout obj();
obj.format_text(
  data: "Go To Demo of a Vertical Marquee (aka Chyron) with an Imbedded Animated GIF",
  style_attr: 'fontfamily=Georgia fontsize=24pt just=C',
```

```
  url: "./VerticalMarqueeWithAnimatedGIF.html");
run;

ods html5 close;
```

Code for Figure 3.

Figure 4. Clipping of the Top of a Web Page That Includes Vertical (Scrolling) Marquee at the Top and Which Includes an Image (Here, an Animated GIF). The GIF cycles once and then becomes a stationary image. A horizontal marquee also may include an image file.

```
%let path = C:\temp\! LeRB Demo;

ods noresults;
ods _all_ close;
title; footnote;
ods html5
  path="&path" (url=none)
  body='VerticalMarqueeWithAnimatedGIF.html'
  (title='Animated SAS Logo Inside of this Vertical Marquee');

/* HEIGHT=32 WIDTH=92 These are the "native" dimensions of the GIF. */

ods text=
"<font face=Georgia size=7 color=yellow>
<marquee bgcolor=blue width=1852 height=300 hspace=21 vspace=13 direction=up>
<img src='PoweredBySASanimated.gif' HEIGHT=128 WIDTH=368>
<br>In a marquee, whether horizontal or vertical,
<br>you can include an image,
<br>including an animated GIF
<br>(like the one here).
<br>
<br>The animated GIF here cycles two times,
<br>and then becomes a stationary image.
<br>
<br>This vertical marquee cycles indefinitely.
 </marquee>
 </font>";
/* NOTE: hspace is meant to try to center the Marquee display box when width=1852
   and the web page is displayed on my laptop 17-icnh diagonal screen monitor.
   vspace affects the height of white space above and below the Marquee display box.
   The marquee is wider than my web browser window when the web page is displayed
   on my 31-inch diagonal auxiliary monitor.
   Both monitors are set to 1920 X 1080 resolution
   and scale for "size of text, apps, and other items" set to 100%.
   The browser is set to 100% zoom for both.*/
```

```
data _null_;
dcl odsout obj();
obj.format_text(
  data: "Go To a Lively Royal Web Page",
  style_attr: 'fontfamily=Georgia fontsize=24pt just=C',
  url: "./LivelyRoyalWebPage.html");
run;

ods html5 close;
```

Code for Figure 4.

ANIMATION

RTF ANIMATION

RTF stands for Rich Text Format. In the earlier days of Microsoft Word, there was interest in this document format. MS Word can open and display an RTF document.

I first used RTF animation in 2006. For me, it still works in 2026, when the RTF file is opened with my Microsoft Word 2010. In some (all?) later versions of MS Word, animated text in an opened RTF file is not supported (i.e., NO animation is apparent).

GIF ANIMATION

GIF stands for Graphics Interchange Format. GIF images are not inherently animated. You can create a data graphic as a GIF. A GIF image is likely to be the smallest file that you can create for a data graphic.

An animated GIF can be built to loop once through its animation, and then stop. It can be built to loop N times, where N is any number you desire, and then stop. It can be built to loop forever.

You can find an enormous universe of pre-built animated GIFs on the internet. Some are intended for use as EMOJIs. Some are free, some are for-fee.

The stand-up presentation for this paper includes several examples that I can't usefully show you in this static paper. The Queen's flag included on the Lively Royal Web Page at the start of this paper is an animated GIF.

I show you how to use SAS to build animated GIFs that tell a story. Their story is always hi**story**.

ANIMATED GIFS

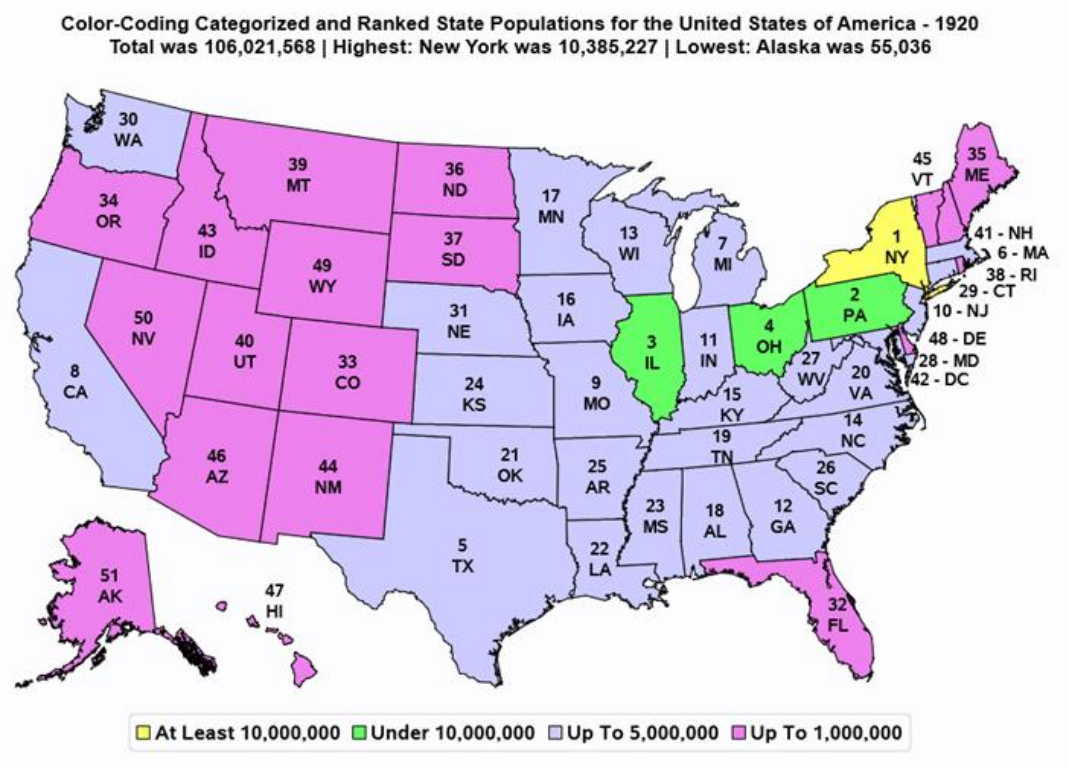

Figure 5A. Static Screen Capture of First Image in Animated GIF: USA Popuiation by State 1920.

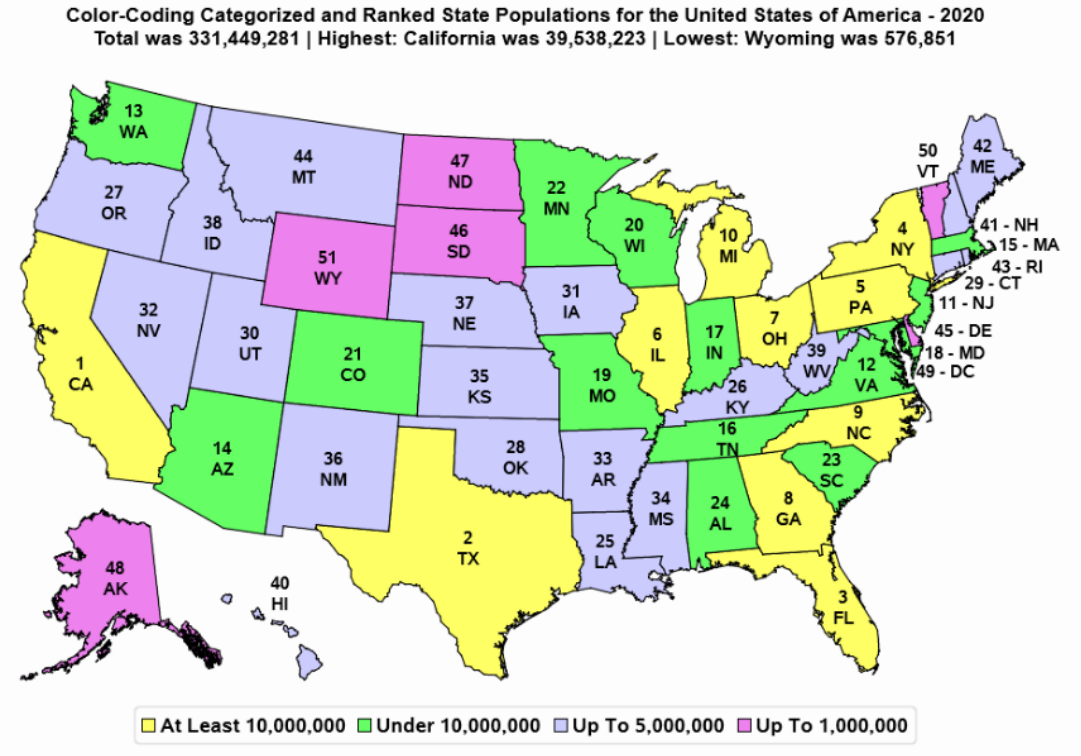

Figure 5B. Static Screen Capture of Last Image in Animated GIF: USA Population by State 2020.

```
proc template;
define style styles.LeRB_FourColorMap;
 parent=styles.htmlblue;
class graphcolors /
  'gdata4'=violet
  'gdata3'=CXCCCCFF
  'gdata2'=CX66FF66
  'gdata1'=CXFFFF66;
run;

data ToFormat;
retain fmtname 'popfmt_ForFourColorMap' type 'N' Start 0;
length Start End 8 Label $ 64;
Start = 0;
End   = 1000000;
Label = 'Up To 1,000,000';
output;
Start = 1000001;
End   = 5000000;
Label = 'Up To 5,000,000';
output;
Start = 5000001;
End   = 9999999;
Label = 'Under 10,000,000';
output;
Start = 10000000;
End   = 99999999;
Label = 'At Least 10,000,000';
output;
run;

proc format cntlin=ToFormat library=work;
run;

%macro MapByYear(Start=,End=);

%do Year = &Start %to &End %by 10;

data work.PopData(keep=statecode statename Population);
set sashelp.us_data;
where statecode NE 'PR';
Population = population_&Year;
run;

proc sort data=work.PopData;
by descending Population;
run;

data work.Ranked;
length TotalPop 8;
retain TotalPop 0;
length Rank $ 2;
set work.PopData end=LastOne;
Rank = left(put(_N_,2.));
TotalPop + Population;
if _N_ EQ 1 then do;
  call symput('HighestState',trim(left(statename)));
  call symput('HighestPop',trim(left(put(Population,comma11.))));
  call symput('Highest',Population);
end;
else
if LastOne then do;
  call symput('LowestState',trim(left(statename)));
  call symput('LowestPop',trim(left(put(Population,comma11.))));
  call symput('TotalPop',trim(left(put(TotalPop,comma11.))));
end;
output;
```

```
run;

proc sort data=work.Ranked out=work.RankedToMerge;
by statecode;
run;

proc sort data=mapsgfk.uscenter(keep=x y statecode ocean) out=work.StateCentersToMerge;
where statecode NE 'PR';
by statecode;
run;

data work.population_plus_statecenter;
merge work.StateCentersToMerge(in=StateCenter) work.RankedToMerge(in=Ranked);
by statecode;
if StateCenter and Ranked;
if ocean EQ 'Y' AND statecode EQ 'VT'
then delete;
else
if ocean EQ 'N' AND statecode IN ( 'NH' 'MA' 'CT' 'RI' 'NJ' 'DE' 'MD' 'DC' )
then delete;
run;

data work.StateAnnoLabels;
set work.population_plus_statecenter;
length label $15;
label=statecode;
if label='FL'
then x = x + 0.004;
if label='HI'
then y = y + 0.025;
if label='ME'
then y = y + 0.005;
if statecode EQ 'TN'
then y = y - 0.002;
if statecode EQ 'MI'
then y = y - 0.005;
if statecode EQ 'SC'
then y = y - 0.0022;
if statecode EQ 'WV'
then y = y - 0.001;
if ocean EQ 'Y'
then do;
  if statecode IN ('NH')
  then x = x + 0.03;
  else
  if statecode IN ('MA' 'CT' 'RI')
  then x = x + 0.04;
  else
  if statecode EQ 'MD'
  then x = x + 0.02;
  else
  if statecode EQ 'DC'
  then x = x + 0.01;
  else
  if statecode IN ('NJ' 'DE')
  then x = x + 0.03;
  x = x - 0.02;
  OceanLabel = trim(Rank) || ' - ' || StateCode;
  output;
end;
else do;
  if statecode EQ 'VT'
  then do;
    y = y + 0.033;
    x = x - 0.003;
  end;
  RankLabel = Rank;
```

```
  StateCodeLabel = StateCode;
  output;
end;
run;

title1 height=13pt f=Arial Bold
  "Color-Coding Categorized and Ranked State Populations for the United States of America -
&Year";
title2 height=13pt f=Arial Bold
  "Total was &TotalPop | Highest: &HighestState was &HighestPop | Lowest: &LowestState was
&LowestPop";
footnote justify=right height=8pt color=blue 'LeRB';
proc sgmap
  maprespdata=work.PopData
  mapdata=mapsgfk.us
  plotdata=work.StateAnnoLabels;
format Population popfmt_ForFourColorMap.;
/* This WARNING message, with no practical significance,
   will appear in the SAS log:
WARNING: Variable Population not found in data set
         MAPSGFK.US. */
choromap Population / mapid=statecode discrete
  lineattrs=(thickness=1 color=black)
  name='map';
text x=x y=y text=RankLabel / position=top
  textattrs=(color=black family=Arial size=11pt weight=bold);
text x=x y=y text=StateCodeLabel / position=bottom
  textattrs=(color=black family=Arial size=11pt weight=bold);
text x=x y=y text=OceanLabel /
  textattrs=(color=black family=Arial size=11pt weight=bold);
keylegend 'map' / title=''
  valueattrs=(family=Arial size=13pt weight=bold) autoitemsize;
run;

%end;

%mend MapByYear;

options mprint;
ods _all_ close;
ods graphics / noscale imagefmt=GIF width=13.33in height=6.8in noborder;
options papersize=('13.33 in', '6.8 in')
  nodate nonumberSAScodeGeneratedAnimatedGIFs
  printerpath=gif animation=start /* start recording images to GIF */
  animduration=1 /* the time unit is actually shorter than one second */
  animloop=no noanimoverlay;
ods noresults;
ods printer style=styles.LeRB_FourColorMap
  file='C:\temp\! Demo
Elements\SAScodeGeneratedAnimatedGIFs\15May2026_AnimatedFourColorUSApopulationMapAnnotatedW
ithRankAndStateCodeFor1920to2020.gif';
%MapByYear(Start=1920,End=2020);
options printerpath=gif animation=stop; /* stop recording images */
ods printer close; /* close the animated GIF file */;
```

Code for Figure 5.

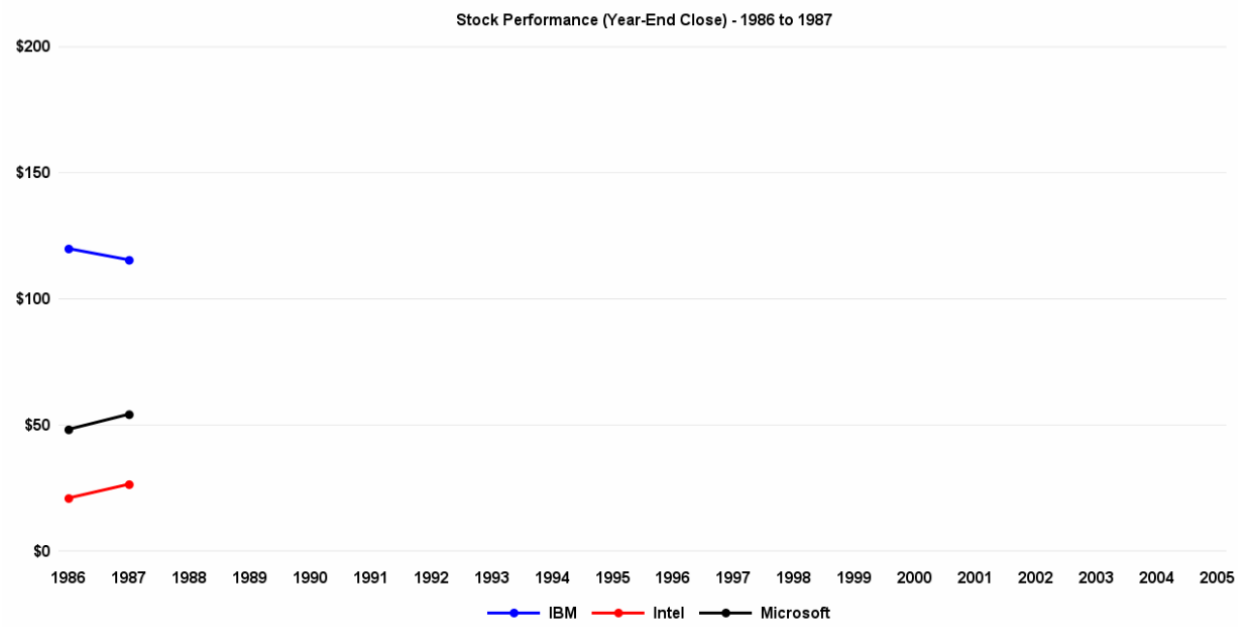

Figure 6A. Static Screen Capture of First Image in Animated GIF: Stock Performance 1986-1987.

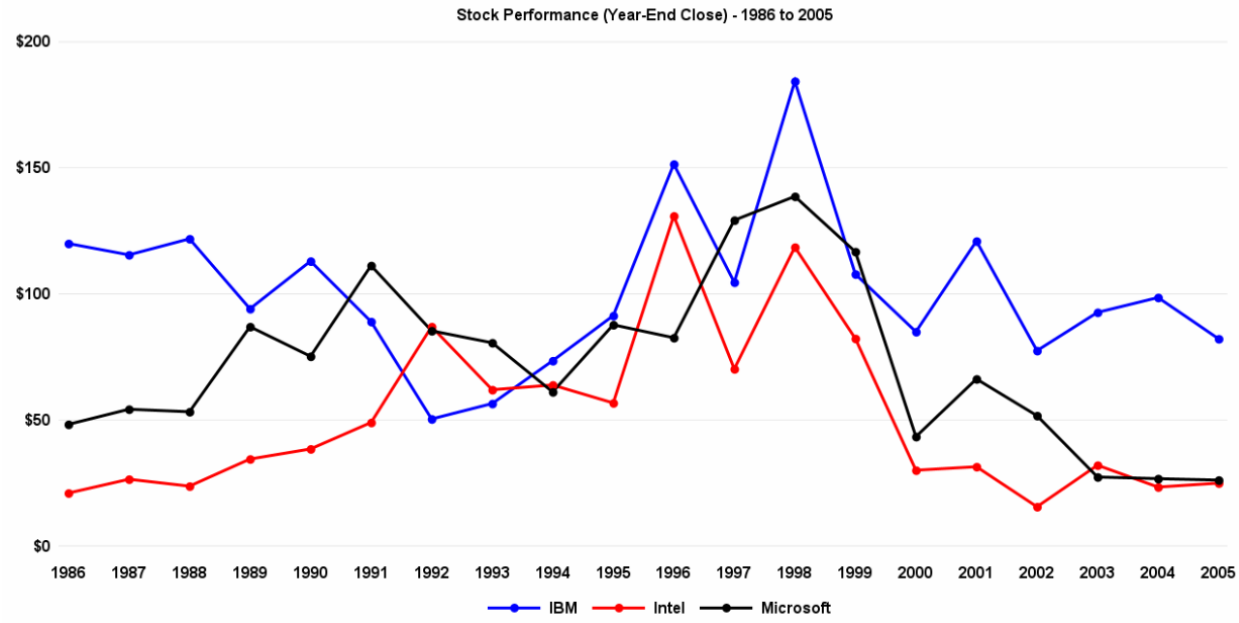

Figure 6B. Static Screen Capture of Last Image in Animated GIF: Stock Performance 1986-2006.

```
/* NOTE: this is an adaptation of code from a blog by Rick Wicklin, found at
https://blogs.sas.com/content/iml/2016/08/22/animation-by-statement-proc-sgplot.html */

data work.stocks;
set sashelp.stocks;
Month = month(date);
Year = year(date);
run;

proc sort data=stocks;
```

```
by date;
run;

%macro PlotByYear;
%do endyear = 1987 %to 2005 %by 1;
proc sgplot data=work.stocks noborder;
StyleAttrs datacontrastcolors=(Blue Red Black);
where year LE &endyear AND month EQ 12;
title
  "Stock Performance (Year-End Close) - 1986 to &endyear";
  /* create 19 images, one for each ending year */
   series x=year y=close /
     group=stock
     lineattrs=(pattern=Solid thickness=3px)
     markers markerattrs=(symbol=CircleFilled size=9px);
   xaxis display=(noline noticks nolabel)
     values=(1986 to 2005 by 1);
   yaxis display=(noline noticks nolabel)
     min=0 max=200 grid;
keylegend / title=' ' noborder  valueattrs=(family=Arial size=11 weight=bold);
run;
%end;
%mend PlotByYear;

%macro AllGraphTextSetup(Size,Family=Arial,Weight=Bold);
%let FamilyForStyle = %sysfunc(compress(&Family,' '));
proc template;
  define style FromPearlStyleGraphFont&FamilyForStyle.&Size.pt&Weight;
  parent=styles.Pearl;
  class GraphFonts /
     'GraphValueFont' = ("&Family",&Size.pt,&Weight)
     'GraphLabelFont' = ("&Family",&Size.pt,&Weight)
     'GraphDataFont'  = ("&Family",&Size.pt,&Weight)
     'GraphTitleFont' = ("&Family",&Size.pt,&Weight)
     'GraphFootnoteFont' = ("&Family",&Size.pt,&Weight);
end;
run;
%mend AllGraphTextSetup;
%AllGraphTextSetup(24);

%macro PlotByYear;
%do endyear = 1987 %to 2005 %by 1;
proc sgplot data=work.stocks noborder;
StyleAttrs datacontrastcolors=(Blue Red Black);
where year LE &endyear AND month EQ 12;
title
  "Stock Performance (Year-End Close) - 1986 to &endyear";
  /* create 19 images, one for each ending year */
   series x=year y=close /
     group=stock
     lineattrs=(pattern=Solid thickness=3px)
     markers markerattrs=(symbol=CircleFilled size=9px);
   xaxis display=(noline noticks nolabel)
     fitpolicy=stagger
     values=(1986 to 2005 by 1);
   yaxis display=(noline noticks nolabel)
     min=0 max=200 grid
     minorgrid minorcount=4;
keylegend / title=' ' noborder;
run;
%end;
%mend PlotByYear;

/* NOTE: In the SAS log,
   this code will produce this WARNING message of NO Real Consequence:
   WARNING: WIDTH exceeds available space for PRINTER destination.
            Setting WIDTH=12.82292in. */
```

```
%let path = C:\temp\! Demo Elements\SAScodeGeneratedAnimatedGIFs;

options mprint;
title; footnote;
ods noresults;
ods _all_ close;
ods graphics / imagefmt=GIF width=13.33in height=6.8in noborder;
options papersize=('13.33 in', '6.8 in')
  nodate nonumber
  animduration=0.5
  animloop=yes noanimoverlay
  printerpath=gif animation=start; /* start recording images to GIF */
ods printer style=FromPearlStyleGraphFontArial24ptBold
  file="&path\15May2026_AnimatedPictureOfStockYearEndClose_1986to2005.gif"; /* create the
animated GIF file */
%PlotByYear;
options printerpath=gif animation=stop; /* stop recording images */
ods printer close; /* close the animated GIF file */
```

Code for Figure 6.

ANIMATION IN RTF (RICH TEXT FORMAT) DOCUMENTS OPENED WITH MICROSOFT WORD

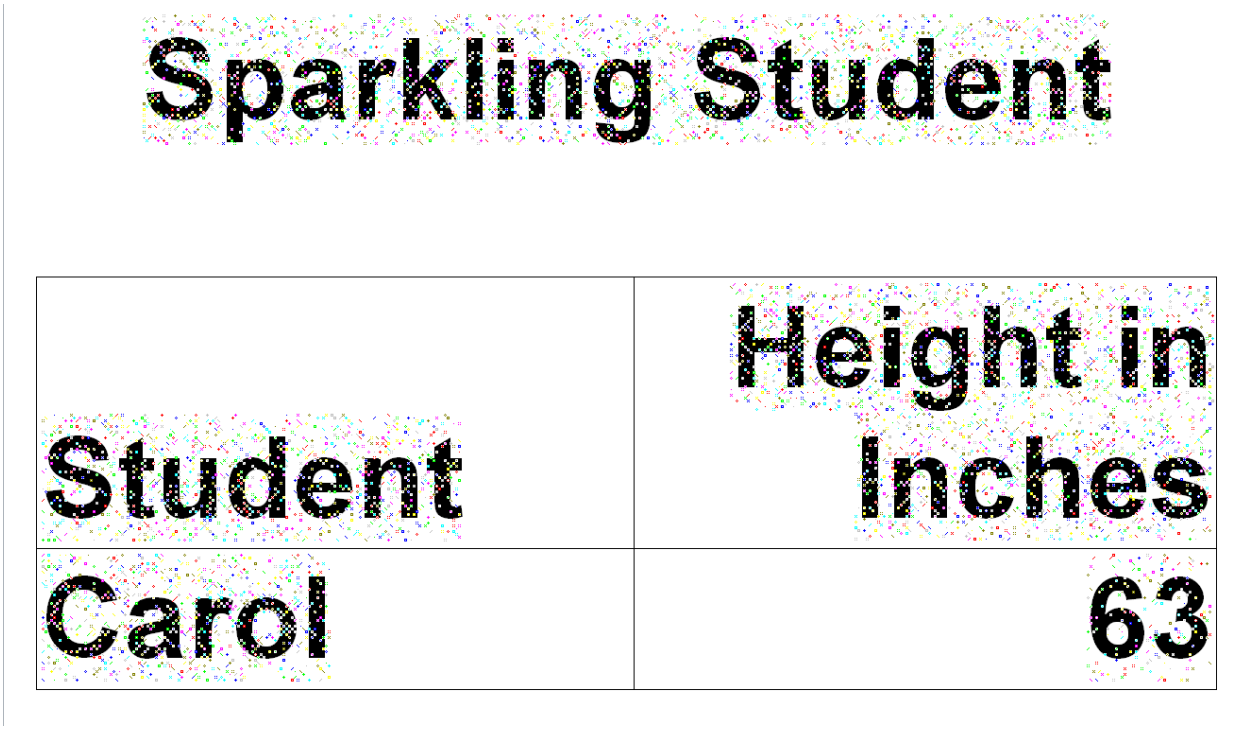

Figure 7. Sparkling Text Using RTF Animation: Animtext Option 3. This static screen capture was made with the RTF file opened in Microsoft Word 2010. The later versions of Microsoft Word might show NO effect.

```
proc format;
picture rtfAnim
0 -< 63 = " "
63    = "63}" (prefix="{\animtext3 ")
63 - high   = " ";
value $namefmt
 'Carol'="{\animtext3 Carol}";
run;

data work.ToPrint;
label Height='Height in Inches';
set sashelp.class(keep=Name Height);
if Name EQ 'Carol';
Height = ceil(Height); /* round up to next higher integer */
run;

%let path = C:\temp\! Demo Elements\RTFanimation;

ods results off;
ods _all_ close;
options nodate nonumber;
options orientation=landscape
  leftmargin=0.25in rightmargin=0.25in;
title; footnote;
ods escapechar='^';

title1 height=24pt font=Georgia color=black '{\animtext3 My Sparkling Wife}';
/* Regardless of the STYLE used, or if taking the default style, As Here,
```

```
   to get black titles, BODYTITLE is required. */
ods rtf body="&path\15May2026_SparklingTableInRTFdocument.rtf" bodytitle;

proc print data=work.ToPrint noobs label;
var name height /
  style(header) = {fontsize=24pt fontfamily=Georgia fontweight=medium background=white}
  style(data)   = {fontsize=24pt fontfamily=Georgia};
format height rtfAnim. name $namefmt.;
label name=  '{\animtext3 My Wife}';
label height='{\animtext3 Height in Inches}';
run;

ods text="^S={JUST=CENTER FONT_FACE=Georgia FONT_SIZE=24pt COLOR=BLUE
  URL=""15May2026_SelfDescribedRTFAnimation.rtf""}
Go To a Demo of SelfDescribed RTF Animation";

/* Another link to make available by UNCOMMENTing this code
ods text="^S={JUST=CENTER FONT_FACE=Georgia FONT_SIZE=24pt COLOR=BLUE
  URL=""https://www.sas.com""}Go To SAS.com"; */

ods rtf close;
```

Code for Figure 7.

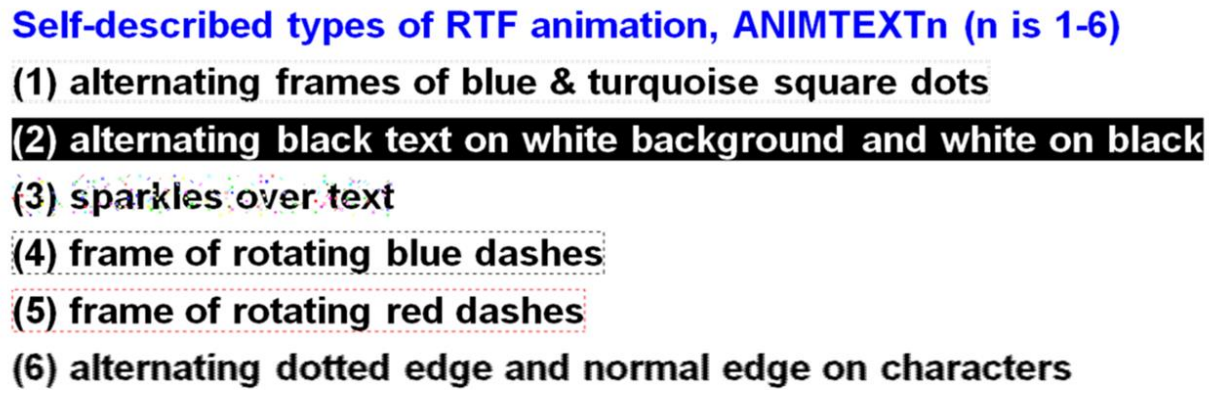

Figure 8. Current Demonstration of All Six RTF Animation Options, Self-Described. The timing of this screen capture failed to capture the dotted edge mode of the pulsating cycle for ANIMTEXT6 above.

```
%let path = C:\temp\! Demo Elements\RTFAnimation;

ods results off;
ods _all_ close;
options nodate nonumber;
options orientation=landscape
  leftmargin=0.25in rightmargin=0.25in;
title; footnote;
ods escapechar='^';

ods rtf body="&path\15May2026_SelfDescribedRTFAnimation.rtf"
    bodytitle
    style=styles.minimal;
title1 font='Arial/Bold'
   justify=left h=24pt color=blue
'Self-described types of RTF animation, ANIMTEXTn (n is 1-6)'
  justify=left height=8pt color=white 'white space'
  justify=left height=24pt color=black
'{\animtext1 (1) alternating frames of blue & turquoise square dots}'
  justify=left height=8pt color=white 'white space'
  justify=left height=24pt color=black
'{\animtext2 (2) alternating black text on white background and white on black}'
```

```
  justify=left height=8pt color=white 'white space'
  justify=left height=24pt color=black
'{\animtext3 (3) sparkles over text}'
  justify=left height=8pt color=white 'white space'
  justify=left height=24pt color=black
'{\animtext4 (4) frame of rotating blue dashes}'
  justify=left height=8pt color=white 'white space'
  justify=left height=24pt color=black
'{\animtext5 (5) frame of rotating red dashes}'
  justify=left height=8pt color=white 'white space'
  justify=left height=24pt color=black
'{\animtext6 (6) alternating dotted edge and normal edge on characters}';
proc print data=sashelp.class noobs label
 /* Make the invisible table as small as possible.
    Its size isn't really important since it is invisible.
    Passing the mouse over it does show that something is really lurking there. */
style(table)={bordercolor=white borderwidth=1px}
style(header)={color=white fontsize=1pt cellpadding=0 cellspacing=0}
style(data)={color=white fontsize=1pt cellpadding=0 cellspacing=0};
where name EQ 'Carol';
var sex;
run;

ods text="^S={JUST=CENTER FONT_FACE='Arial' FONT_WEIGHT=BOLD FONT_SIZE=24pt COLOR=BLUE
  URL=""5May2026_SparklingTableInRTFdocument.rtf""}
See Sparkling Title and Sparkling Table in an RTF Document";

ods rtf close;
```

Code for Figure 8.

3D MADE SAFE FOR DATA GRAPHIC VISUAL COMMUNICATION

This section is devoted to pie charts, donut charts, and bar charts. There are lots of examples. The only code furnished will be for a subset, namely, for recommended examples.

3D can make a data graphic more “interesting”. It can add what likers call “pzazz”. My ongoing mission using SAS graphics software since 1981 has been successful visual communication with data graphics, not to amaze the viewer. However, THIS paper DOES want to help you make your output “dance and sing”. 3D won’t really make your data graphic dance or sing, but let’s see what it can usefully do for you.

Can a 3D data graphic be “Safe and Effective”?

Figure 9. This is an eminently true statement. SAS/GRAPH **can** enable you to plot three numeric variables, if you need to. The result is **not** really easy to accurately and precisely interpret.

For a real disaster, consider the next Figure.

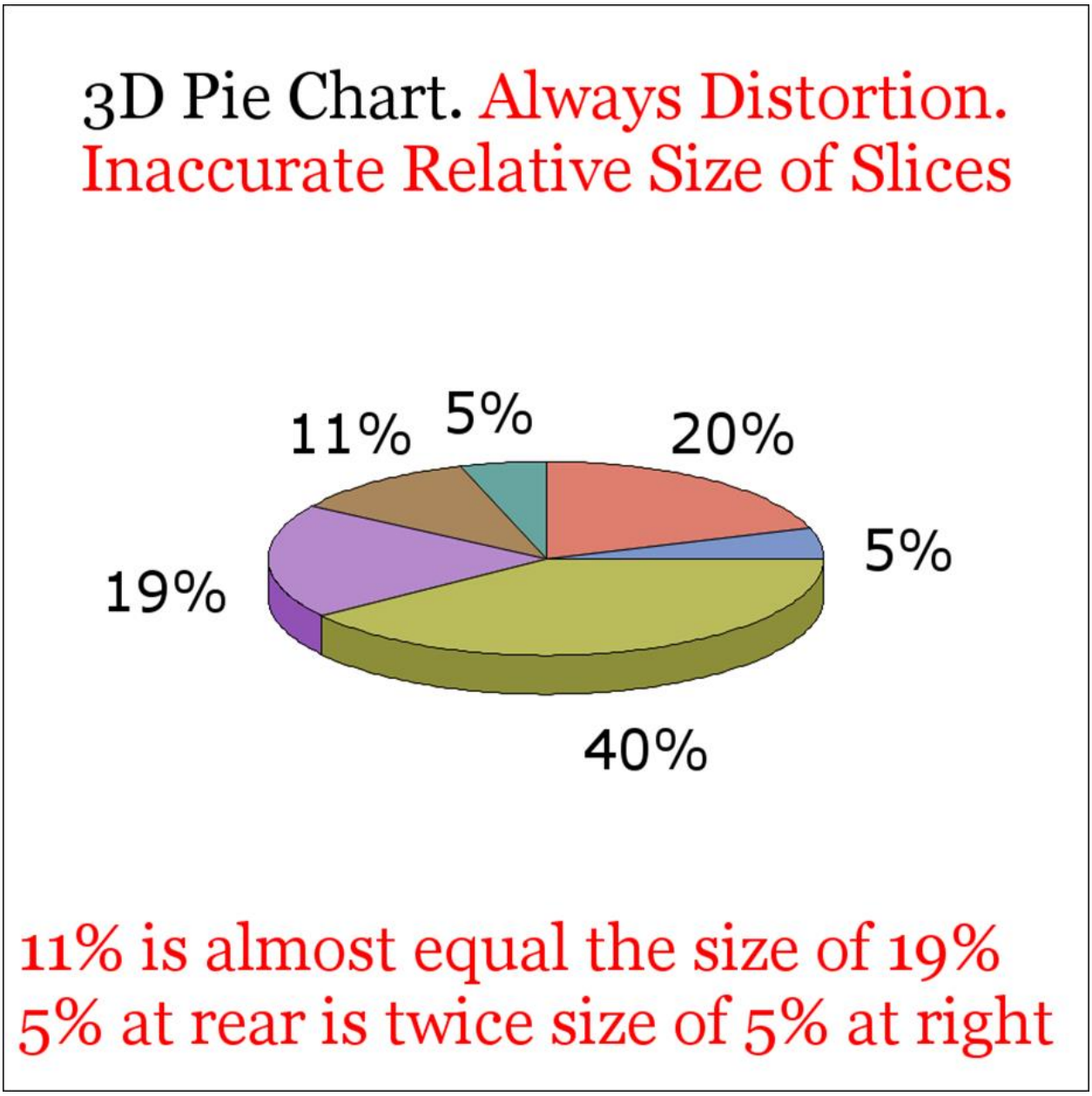

Figure 10. 3D Pie Charts (unless two slices) ALWAYS Distort Comparative Size of Slices. Regrettably, SAS/GRAPH **can** be used to do this. (Before the capability was added, SAS users would contrive various ways to create the Should Not Have Been.)

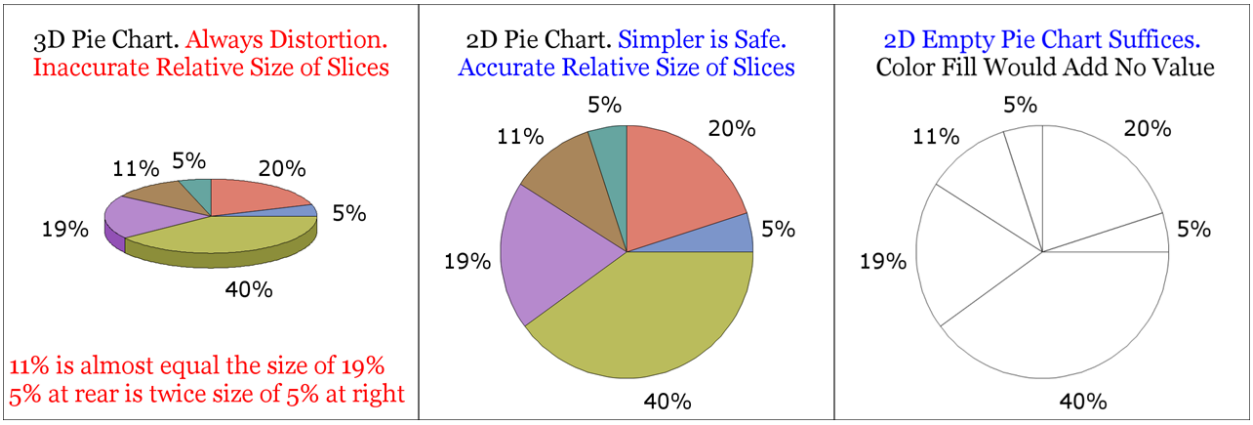

Figure 11. A Triptych of SAS Pie Charts. Bad, Good, Simplest Still Good

```
data work.CategoriesAndValues;
Category = 'A'; Value = 10; output;
Category = 'B'; Value = 40; output;
Category = 'C'; Value = 10; output;
Category = 'D'; Value = 22; output;
Category = 'E'; Value = 38; output;
Category = 'F'; Value = 80; output;
run;

ods results=off;
ods _all_ close;
ods listing style=listing dpi=300;
goptions reset=all;
goptions border;
goptions device=png gsfname=anyname;
goptions hsize=2.17in vsize=2.17in
  htext=8pt ftext='Verdana' ctext=Black;
title2 h=10pt f='Georgia'
c=Blue j=C  "2D Empty Pie Chart Suffices."
c=Black j=C  "Color Fill Would Add No Value";
filename anyname "C:\temp\Figure11_Empty2DpieChart.png";
proc gchart data=work.CategoriesAndValues;
pie  Category / sumvar=Value NoHeading
  slice=none value=none
  percent=outside
  cfill=White
  coutline=Black;
run; quit;
filename anyname clear;

title2 h=10pt f='Georgia'
j=C
c=Black '2D Pie Chart. '
c=Blue 'Simpler is Safe.'
j=C 'Accurate Relative Size of Slices';
filename anyname "C:\temp\Figure11_2DpieChartWithSliceColorFill.png";
proc gchart data=work.CategoriesAndValues;
pie  Category / sumvar=Value NoHeading
  slice=none value=none
  percent=outside
  /* cfill=White take the default color palette */
  coutline=Black;
```

```
run; quit;
filename anyname clear;
ods listing close;
```

Code for Figure 11 2D Pie Charts.

NOT ALL HOPE IS LOST: ODS GRAPHICS COMES TO THE RESCUE FOR 3D

For a 3D-ish effect, demonstrated below for bar charts, pie charts, and donut charts, there is a DATASKIN option, with five not-NONE choices. It can be used on the following PROC SGPLOT statements: DOT, HBAR, HBOX, HEATMAP, HIGHLOW, HISTOGRAM, HLINE, LOESS, NEEDLE, PBSPLINE, POLYGON, REFLINE, REG, SCATTER (to make 3D-ish markers), SERIES, SPLINE, STEP, VBAR, VBOX, VECTOR, VLINE, and WATERFALL. (Even various kinds of lines, unfilled bubbles, unfilled dots, unfilled plot markers, etc. can experience a DATASKIN!) For PROC SGPIE, it can be used on PIE and DONUT statements.

NOTE: Some of the following figures that include pie charts and bar charts are composites built as slides, using ODS PowerPoint. Such composites include their own descriptive headings. Those descriptions may be redundant with part of the figures' captions. Any code that is provided will be for individual graphs within the composite, not code to build the composite.

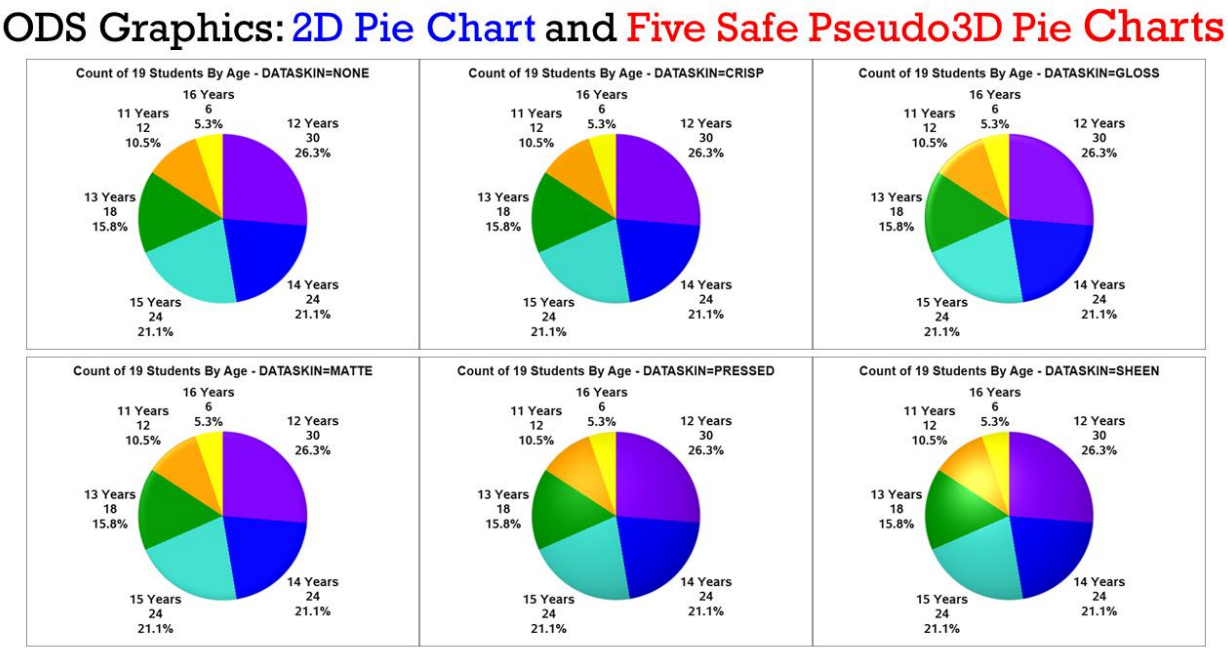

Figure 12. Not all viewers will interpret these as equally approximating bona fide 3D. My preferences are enlarged below.

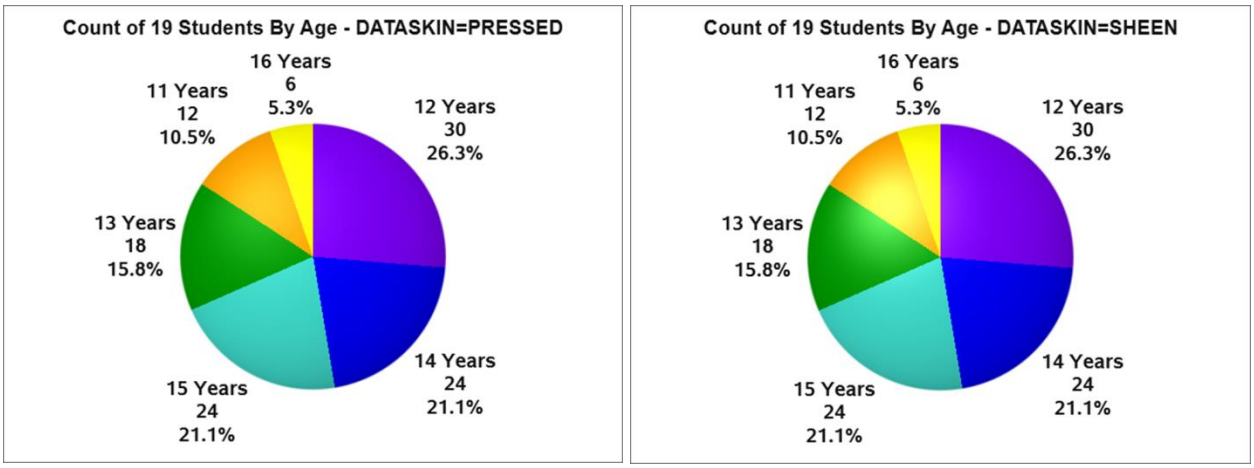

Figure 13. Bessler's preferred choices. Yours?

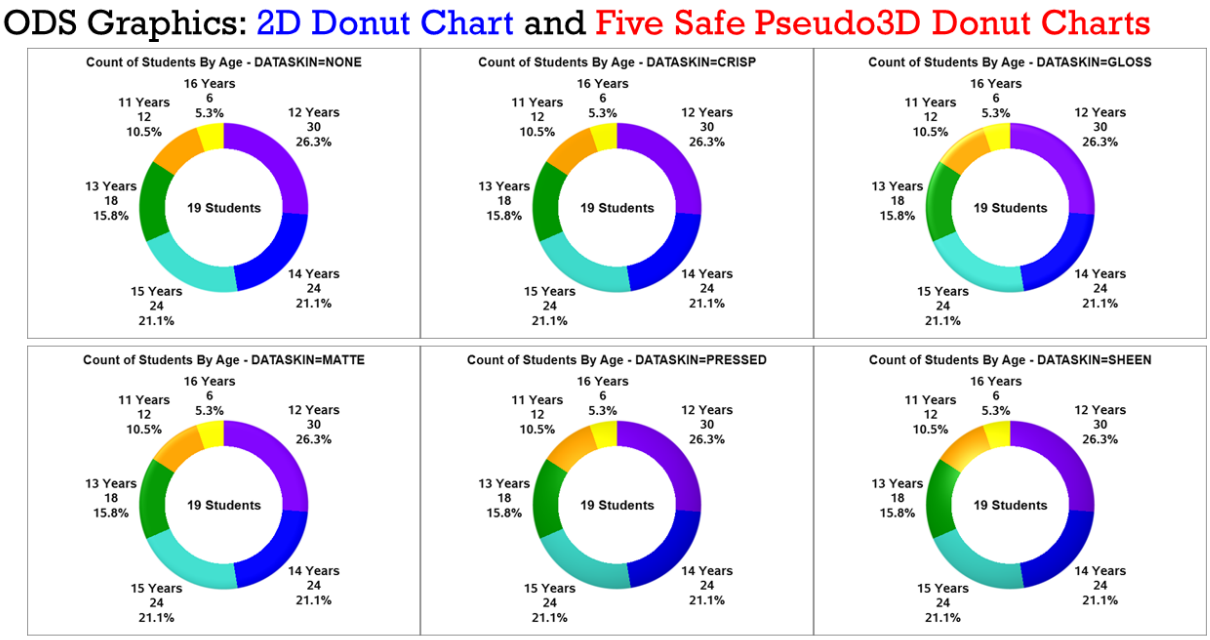

Figure 14. Not all viewers will interpret these as equally approximating bona fide 3D. Which looks most like a real donut?

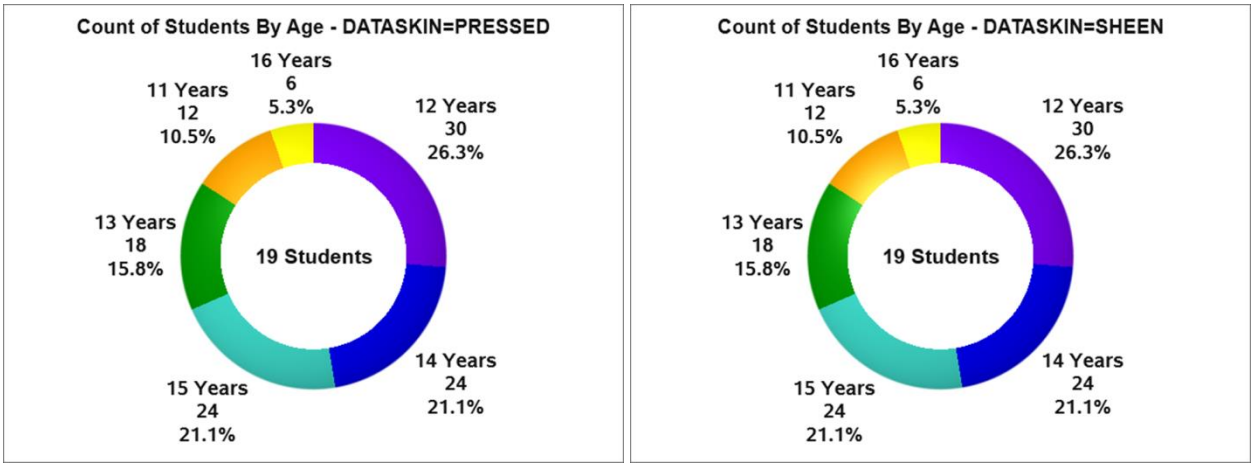

Figure 15. Which looks most like a real donut?

Which do YOU prefer? Pie OR Donut?

A Donut Chart's hole label information ("19 Students") can instead be displayed in a Pie Chart title/subtitle.

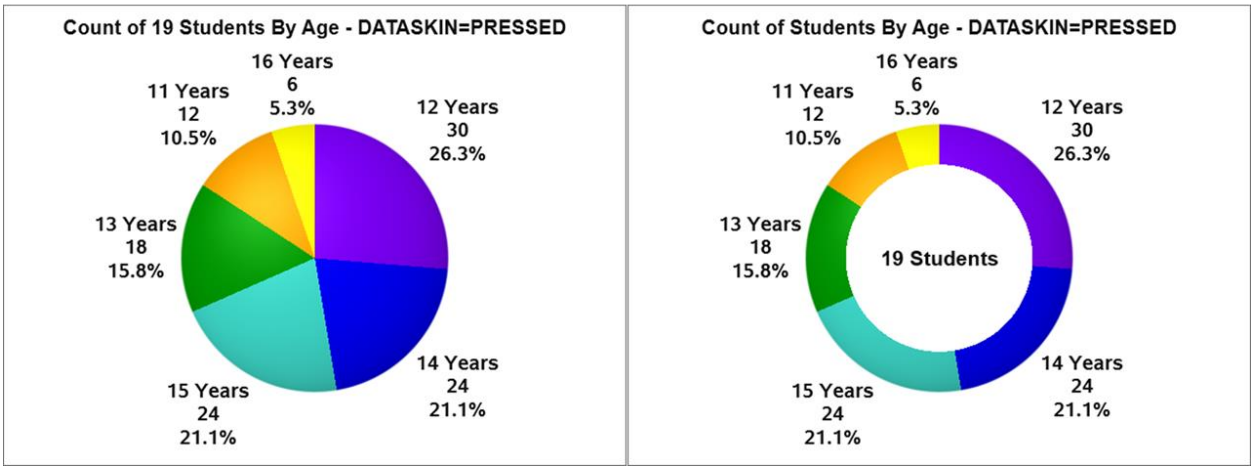

Figure 16. The pie has a diameter here equal to that of the donut, but it certainly would contain more sugar and more dietary calories. Regardless of that not-real-world consideration, the donut chart is a more efficient use of color. Without a legend involved, there is no actual need for color coding. There is no dancing or singing, but the image, whether pie or donut, IS more interesting with color than it would be if empty.

```
DATA work.ToChart(KEEP=AgeInYears);
LENGTH AgeInYears $ 8;
SET sashelp.class(keep=Age) nobs=HowMany;
if _N_ EQ 1 then call symput('TotalCount',trim(left(HowMany)));
AgeInYears = TRIM(LEFT(Age)) || ' Years';
RUN;

TITLE; FOOTNOTE;
ODS RESULTS OFF;
ODS _ALL_ CLOSE;

ODS LISTING GPATH="C:\temp" DPI=300;
```

```
%MACRO PieOrDonutLabelsOutAndDataSkin(PieOrDonut,DataSkin);

ODS GRAPHICS / RESET=ALL NOSCALE WIDTH=4.44in HEIGHT=3.33in
%IF &DataSkin NE NONE %THEN %DO;

IMAGENAME="19Apr2026_Figure16_3D&PieOrDonut.ChartWithOutsideLabelsDataSkinEqual&DataSkin";
%END;
%ELSE %DO;
  IMAGENAME="19Apr2026_Figure16_2D&PieOrDonut.ChartWithOutsideLabels";
%END;
TITLE1 FONT='Arial/Bold' HEIGHT=8pt
  "Count of "
%IF &PieOrDonut EQ Pie %THEN %DO;
  "&TotalCount"
%END;
  " Students By Age - DATASKIN=&DataSkin";
PROC SGPIE DATA=work.ToChart;
STYLEATTRS
  DATACOLORS=(CX7F00FF Blue Turquoise CX009900 Orange Yellow);
/* omit the STYLEATTRS statement to use SAS default color palette */
/* CX009900 is 40% of the way to Black */
&PieOrDonut AgeInYears /
/* omitting RESPONSE= forces use
   of frequency of each CATEGORY value */
  DATASKIN=&DataSkin /* for 3D-ish effect
     options are NONE (the default),
     CRISP, GLOSS, MATTE, PRESSED, SHEEN */
%IF &PieOrDonut EQ Donut %THEN %DO;
  HOLELABEL="&TotalCount Students"
  HOLELABELATTRS=(FAMILY='Arial/Bold')
%END;
  OTHERPERCENT=0
  SLICEORDER=RESPDESC /* largest to smallest */
  DIRECTION=CLOCKWISE /* COUNTERCLOCKWISE is default */
  STARTANGLE=90 /* Twelve O'Clock, Three O'Clock is default */
  STARTPOS=EDGE /* Default is CENTER */
  DATALABELATTRS=(FAMILY=Arial WEIGHT=Bold SIZE=8pt)
  /* SIZE adjusted so that the label for smallest slice
     fits within extended boundaries of the slice */
  DATALABELDISPLAY=(CATEGORY RESPONSE PERCENT)
  /* This is always the order top to bottom */
  /* But the default omits PERCENT */
  DATALABELLOC=OUTSIDE;
RUN;

%MEND PieOrDonutLabelsOutAndDataSkin;

OPTIONS MPRINT;
%PieOrDonutLabelsOutAndDataSkin(Pie,PRESSED);
%PieOrDonutLabelsOutAndDataSkin(Donut,PRESSED);

ODS LISTING CLOSE;
```

Code for Figure 16.

Any Bessler-recommended pie chart or donut chart must have labels or a legend so that slices or bites can be identified and informed with Category, Value, and Percent of Total. That information must be provided for ANY chart for categorical data. Let's look at 3D for an equally popular type of chart used for categorical data, the bar chart.

Not shown, but available for PROC SGPLOT PIE and DONUT charts is the option of callout labels. Like in a legend, all of the label elements are in a horizontal string. If interested, see, for example, Figure 5-12 and Figure 5-19 (and code) in my book. Callout labels can reduce the likelihood of collisions between labels for adjacent pie slices or donut bites. ("bite" is the Bessler-recommended more apt name for donut

segments than “slice”, which is unnecessarily chosen to be used in the software documentation.) The absolutely always failsafe slice/bite label collisions is a legend. If interested, see, for example, Figure 5-15 and Figure 5-20 (and code) in my book. Finally, putting all three parts of the labels inside of a pie chart is an alternative that can be unnecessarily undertaken. If REALLY interested, see the communication-effective example Figure 5-18 (and code) in my book. Oh, let me just show it here. It’s really a big delicious donut. Note that the problem of collisions of labels for possible independent slices for Sport Shoe and Sandal is cunningly overcome by consolidating them into what I call a maximally informative alternative to an “Other” segment. An “Other” segment always invites or prompts the obvious viewer question, “What IS in ‘OTHER’?”

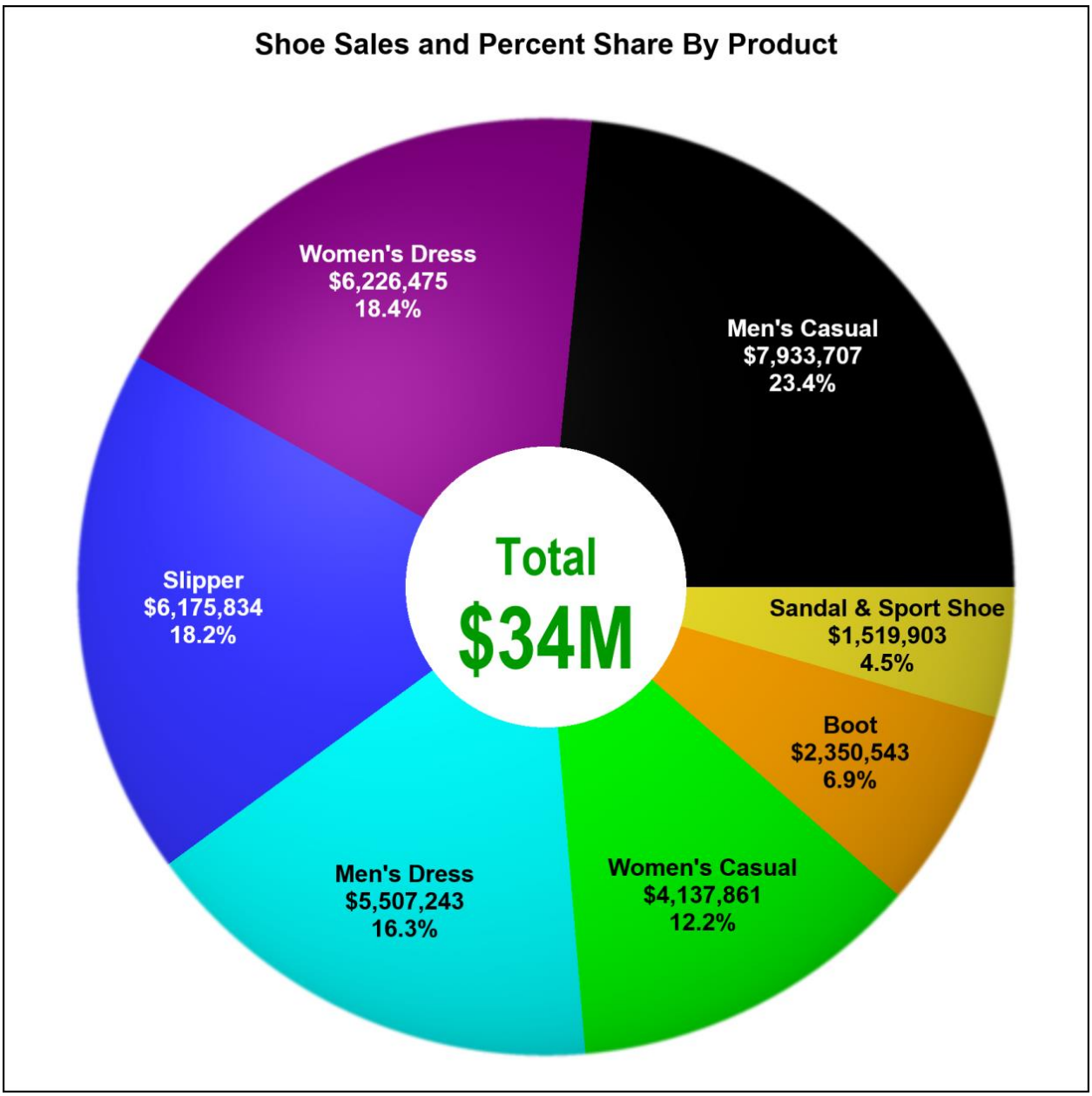

Figure 17. A deliciously informative donut that is definitely too big to eat. If Sandal and Sport Shoe were separate slices, they could not be labeled. The standard solution, consolidation into an “Other” slice, is a needlessly underinformative alternative.

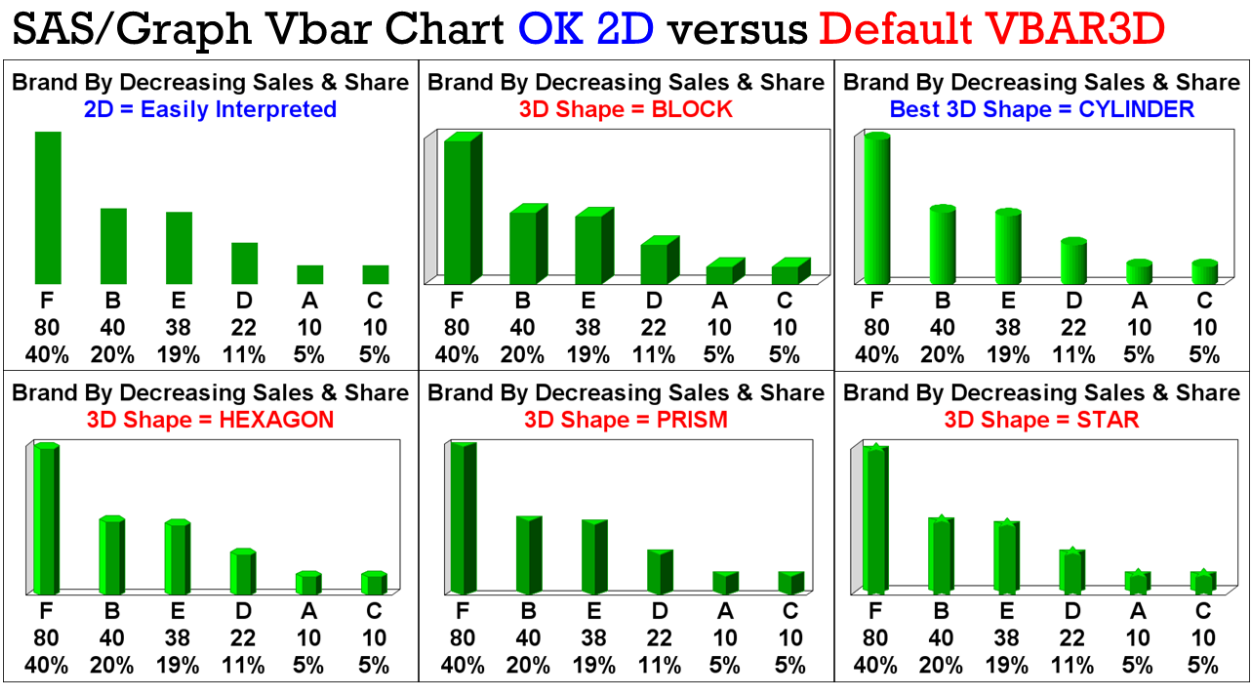

Figure 18. 2D is ALWAYS a good choice. I see CYLINDER as realistically 3D, but NO value added by what are called the frame and the planes.

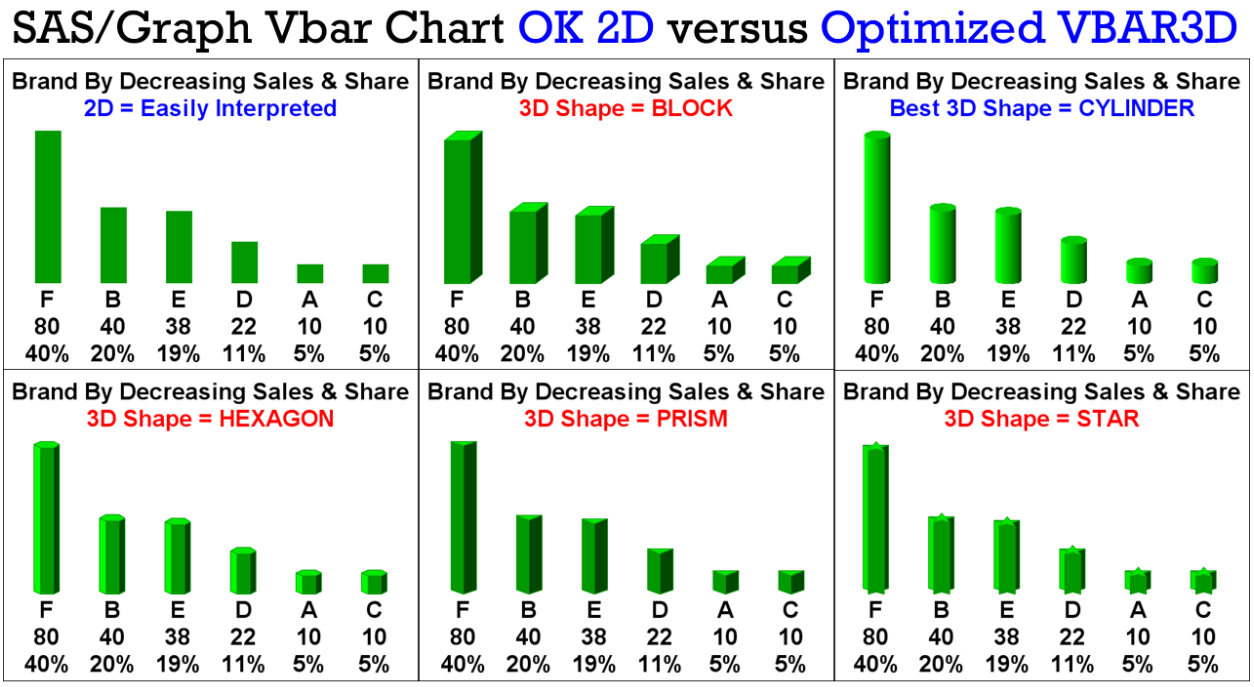

Figure 19. 2D is ALWAYS a good choice. I see CYLINDER as realistically 3D and pleasing to the eye.

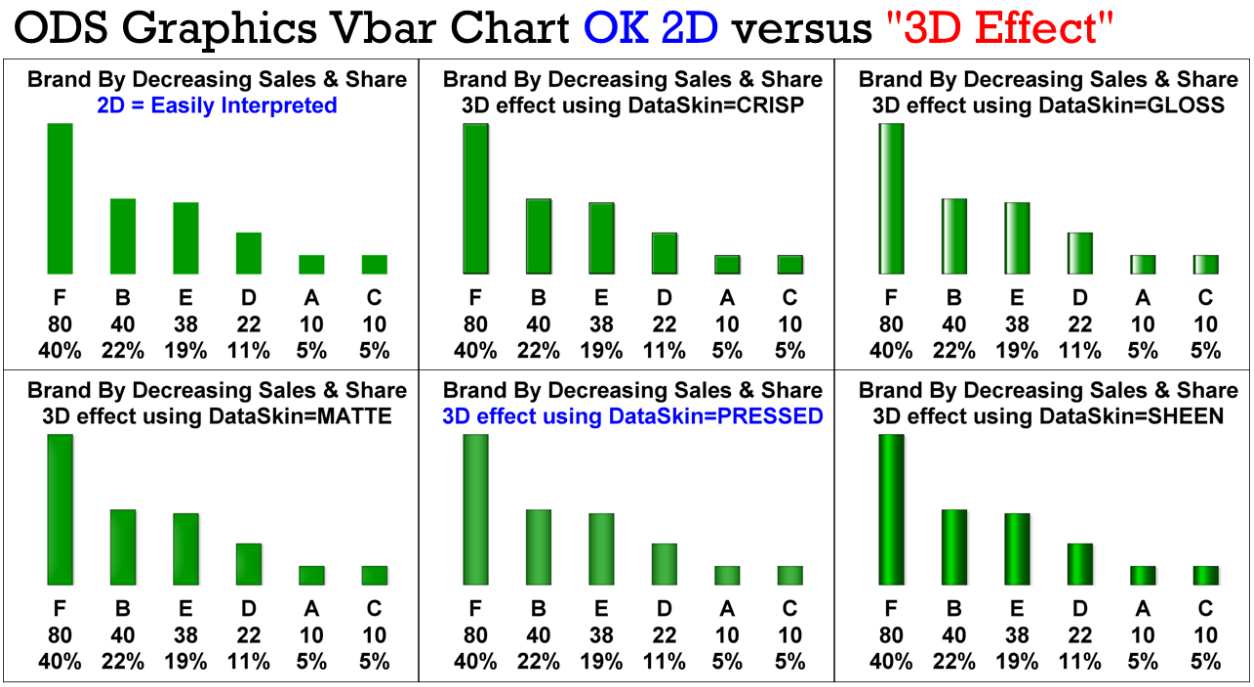

Figure 20. Different Solution, and Adequately Better Than Most of the SAS/GRAPH Bar Shapes.

The best bar chart is always horizontal. Fitting the labels is never a problem. AND, it's how we read text! Notice that for all three of the vertical bar chart alternatives, the bar label components had to be stacked. Fitting the labels for a vertical bar chart is often a challenge. Tilted labels (or worse, vertical) is an unattractive act of desperation. We read horizontally.

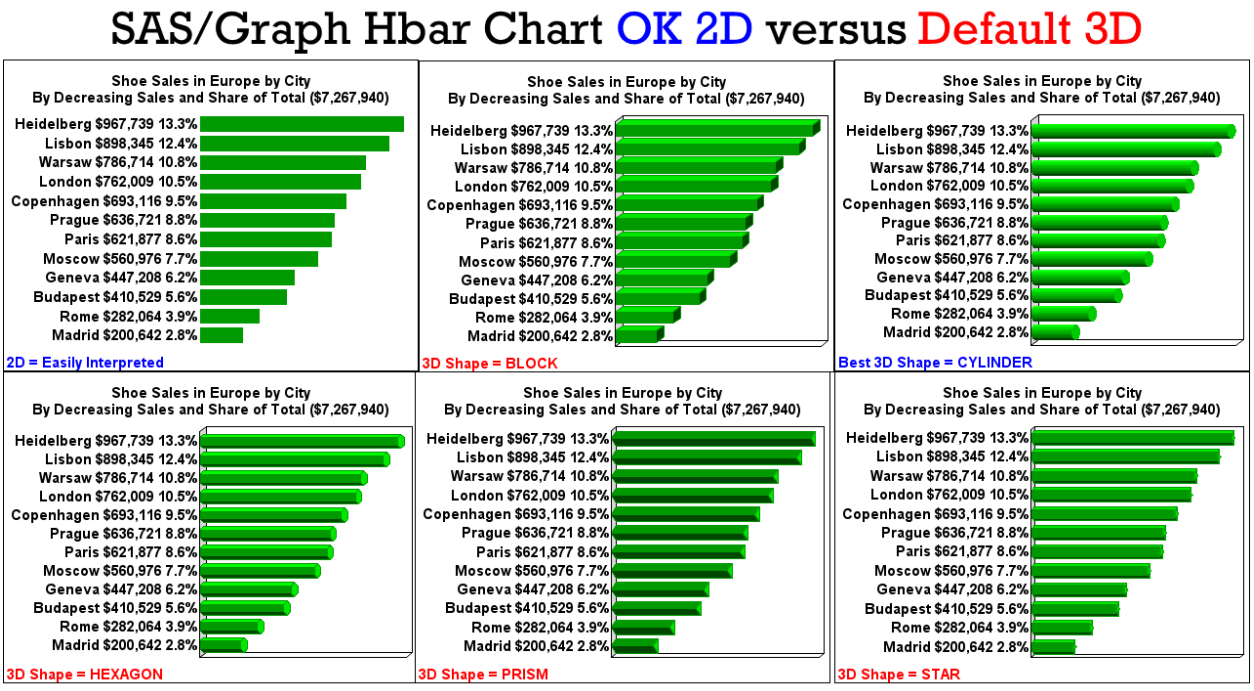

Figure 21. The SAS/GRAPH HBAR3D default options to provide PLANEs along the vertical and horizontal axes and a FRAME to close the top and right sides of the drawing area add no communication value, just needless complexity.

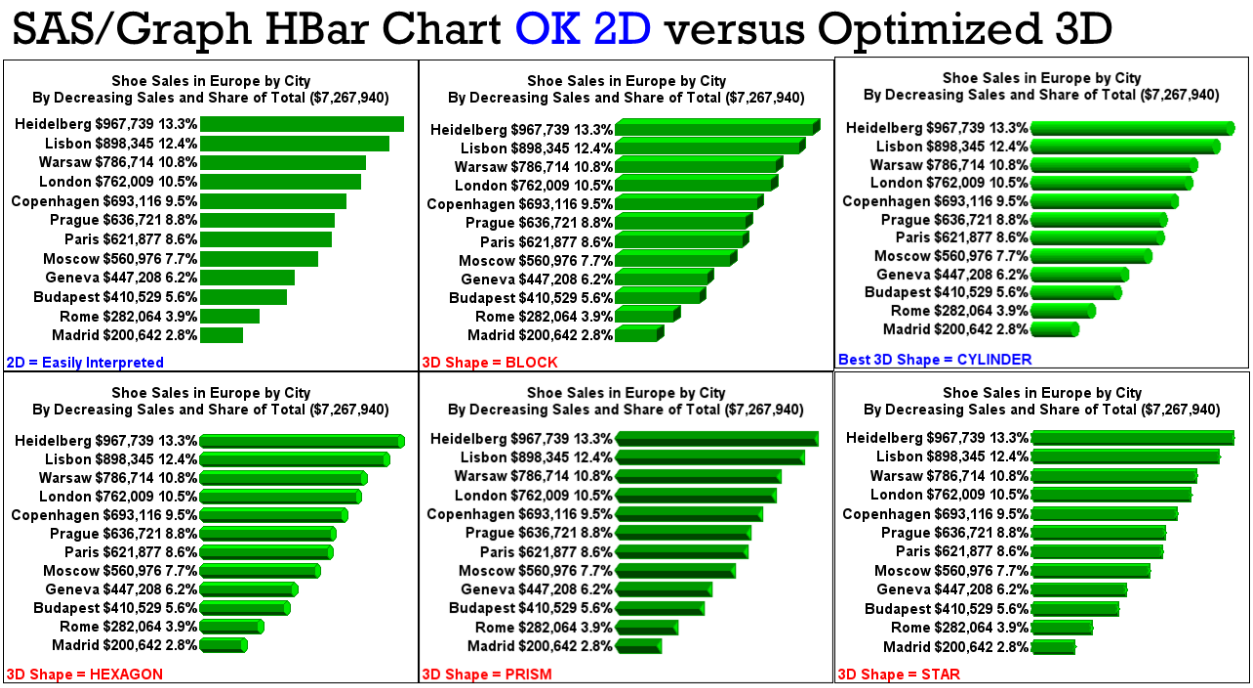

Figure 22. The HBAR3D statement options NOFRAME NOPLANE usefully simplify the images and make the shapes appear as if they are hanging out in 3D space.

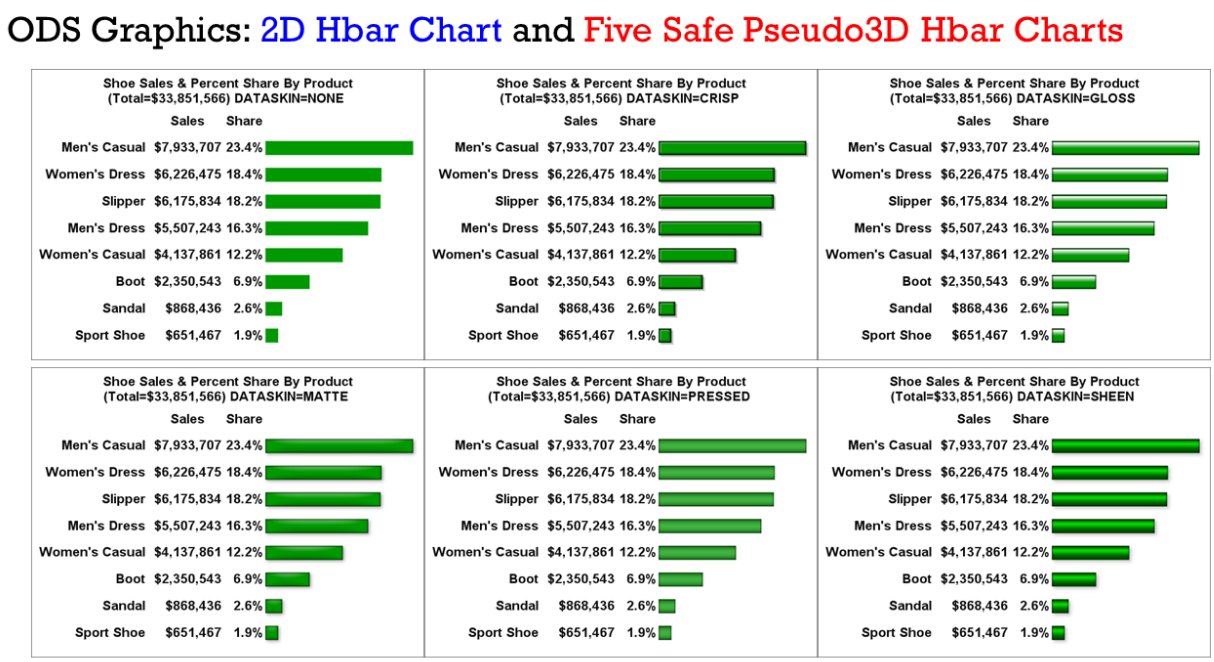

Figure 23. Using the PROC SGPLOT HBAR statement DATASKIN option. It can perhaps create a visual impression of rectangles bulging out of the 2D, but I regard it as a 3D-ish effect, Pseudo3D, with no added value.

GETTING PAST THE OPTIONS OVER-CHOICE

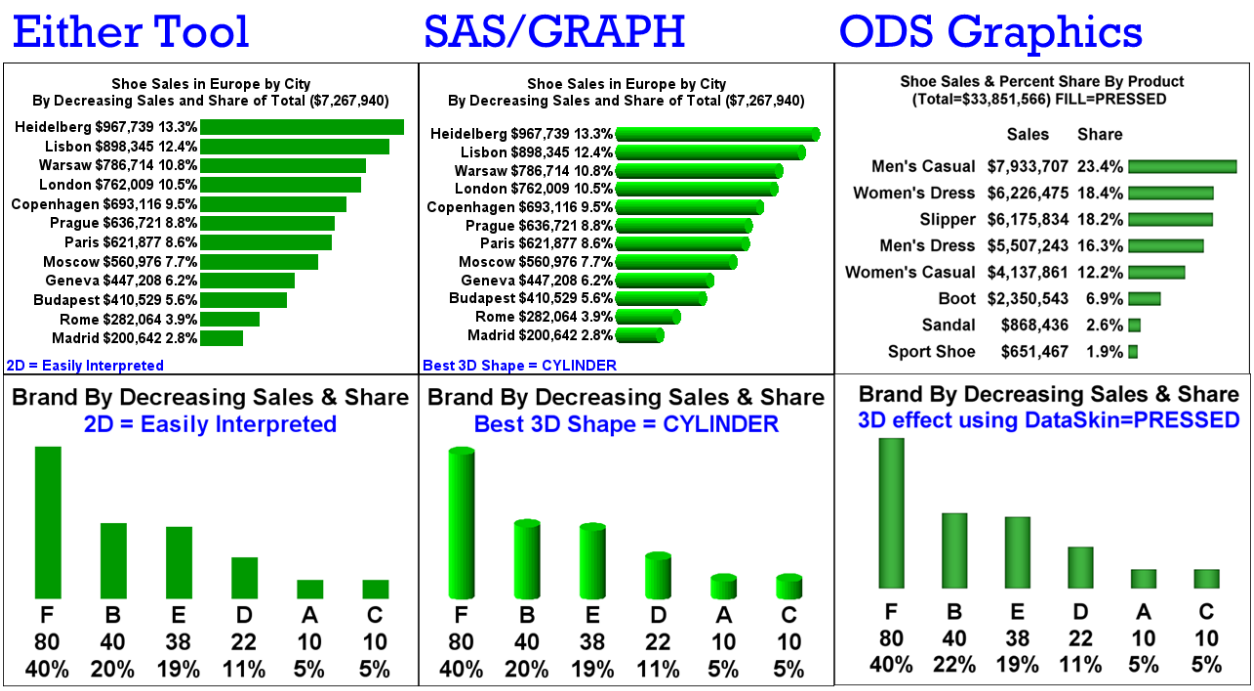

Figure 24. Six Communication-Effective Bar Chart Alternatives.

See four blocks of code below that can create 2D and 3D or pseudo3D hbar charts and vbar charts with SAS/GRAPH and ODS Graphics. I regard the SAS/GRAPH CYLINDER as more truly 3D-like than the ODS Graphics PRESSED dataskin.

```
proc summary data=sashelp.shoes;
WHERE INDEX(Region, 'Europe') NE 0;
class Subsidiary;
var Sales;
output out=work.ToChart sum=Sales;
run;

data work.ToBarCharts;
keep CategoryWIthSTATSimbedded Sales;
retain GrandTotal 0;
length CategoryWIthSTATSimbedded $ 64;
set work.ToChart(keep=Subsidiary Sales);
if _N_ EQ 1 then do;
  GrandTotal = Sales;
  call symput('GrandTotalSales',trim(left(put(Sales,dollar10.))));
  delete;
end;
CategoryWIthSTATSimbedded = Subsidiary;
CategoryWIthSTATSimbedded =
  trim(left(Subsidiary)) || ' ' ||
  trim(left(put(Sales,dollar8.)))  || ' ' ||
  trim(left(put((Sales / GrandTotal) * 100,4.1))) || '%';
run;

%let path = C:\temp;

ods noresults;
ods _all_ close;
ods listing;
```

```
%macro HbarChart(Shape);
/* available shapes are
   NONE, CYLINDER (best), BLOCK, HEXAGON, PRISM, STAR */
goptions reset=all gsfname=anyname device=png
  border
  htitle=11pt ftitle="Arial/Bold" ctitle=black
  htext=11pt ftext="Arial/Bold" ctext=black
  hsize=4.44in vsize=3.33in
  ;
pattern1 c=CX009900;
axis1 /* vertical axis */
  value=(justify=right)
  label=none major=none minor=none style=0;
axis2 /* horizontal axis */
  value=none
  label=none major=none minor=none style=0;
title1 height=8pt color=white 'White Space';
title2 'Shoe Sales in Europe by City';
title3 "By Decreasing Sales and Share of Total (&GrandTotalSales)";
%if &Shape NE 2D %then %do;
footnote1 justify=left
  %if &Shape EQ CYLINDER %then %do;
  color=blue " Best 3D Shape = CYLINDER";
  %end;
  %else %do;
  color=red " 3D Shape = &Shape";
  %end;
filename anyname
"&path\20May2026_SASgraphResponseAndPercentImbeddedInCategoryValueForOptimizedHbar3D_SHAPEe
q&Shape..png";
%end;
%else %do;
footnote1 justify=left color=blue " 2D = Easily Interpreted";
filename anyname
"&path\20May2026_SASgraphWithResponseAndPercentImbeddedInCategoryValueForHbar2Dchart.png";
%end;
proc gchart data=work.ToBarCharts;
%if &Shape NE 2D %then %do;
hbar3d CategoryWithSTATSimbedded /
  noplane /* essential for optimized */
  noframe /* essential for optimized */
  shape=&Shape
%end;
%else %do;
hbar CategoryWithSTATSimbedded /
%end;
  sumvar=Sales
  descending
  nostats
  coutline=CX009900
/* accept defaults for width and space */
  maxis=axis1 raxis=axis2;
run; quit;
filename anyname clear;
%mend HbarChart;

options mprint;
%HbarChart(2D);
%HbarChart(CYLINDER);
/* available shapes are
   2D, CYLINDER (best), BLOCK, HEXAGON, PRISM, STAR */
```

Code for Figure 24 SAS/GRAPH 2D and 3D Horizontal Bar Chart.

```
data _null_;
retain TotalSales 0;
set sashelp.shoes(keep=Sales) end=LastOne;
```

```
TotalSales + Sales;
if LastOne then call symput('TotalSales',trim(left(put(TotalSales,dollar13.))));
run;

%macro HbarChartYaxisTablesAndDataSkin(DataSkin);

ods graphics / reset=all noscale width=4.44in height=3.33in
%if &DataSkin NE NONE %then %do;
  imagename="19May2026_Figure24_3DHbarChartWithYaxisTablesDataSkinEqual&DataSkin";
%end;
%else %do;
  imagename="19May2026_Figure24_2DHbarChartWithYaxisTablesUsingDataSkinEqual&DataSkin";
%end;
title1 font='Arial/Bold' height=11pt
  "Shoe Sales & Percent Share By Product (Total=&TotalSales) DATASKIN=&DataSkin";
title2 height=4pt COLOR=white 'White Space'; /* my preference */
proc sgplot data=sashelp.shoes noborder;
hbar Product / displaybaseline=off
  response=Sales
  dataskin=&DataSkin /* for 3D-ish effect
     options are NONE (the 2D default),
     CRISP, GLOSS, MATTE, PRESSED, SHEEN */
  categoryorder=respdesc
  barwidth=0.5
  nooutline
  fillattrs=(color=CX009900);
yaxistable Sales /
  stat=sum
  location=inside
  position=left
  valueattrs=(family=Arial size=12pt weight=Bold)
  label='Sales'
  labelattrs=(family=Arial size=12pt weight=Bold)
  labelpos=top;
yaxistable Sales /
  stat=percent
  location=inside
  position=left
  valueattrs=(family=Arial size=12pt weight=Bold)
  label='Share'
  labelattrs=(family=Arial size=12pt weight=Bold)
  labelpos=top;
yaxis display=(nolabel noline noticks)
  fitpolicy=none /* needed to prevent thinning the axis values */
  valueattrs=(family=Arial size=12pt weight=Bold);
xaxis display=none;
run;

%mend HbarChartYaxisTablesAndDataSkin;

title; footnote;
ods results off;
ods _all_ close;

ods listing gpath="C:\temp" dpi=300;

options mprint;
%HbarChartYaxisTablesAndDataSkin(NONE);
%HbarChartYaxisTablesAndDataSkin(PRESSED);

ods listing close;
```

Code for Figure 24 ODS Graphics 2D and Pseudo3D Horizontal Bar Chart with Y Axis Tables.

```
data work.ToBarCharts;
length Brand $ 1 Sales PctOfMkt 8;
```

```
Brand='F'; Sales=80; PctOfMkt=40; output;
Brand='B'; Sales=40; PctOfMkt=20; output;
Brand='E'; Sales=38; PctOfMkt=19; output;
Brand='D'; Sales=22; PctOfMkt=11; output;
Brand='A'; Sales=10; PctOfMkt=5;  output;
Brand='C'; Sales=10; PctOfMkt=5;  output;
run;

data work.ToBarCharts;
keep BrandAndSalesAndPercent Sales;
length BrandAndSalesAndPercent $ 16;
set work.ToBarCharts;
BrandAndSalesAndPercent =
  trim(left(Brand)) || ' ' ||
  trim(left(Sales)) || ' ' ||
  trim(left(put(PctOfMkt,2.))) || '%';
run;

%let path = C:\temp;

title; footnote;
ods noresults;
ods _all_ close;
ods listing;

%macro VbarChart(Shape,Width,Space);
/* available shapes are
   2D, CYLINDER (best), BLOCK, HEXAGON, PRISM, STAR */

goptions reset=all gsfname=anyname device=png
  border
  htitle=24pt ftitle="Arial/Bold" ctitle=black
  htext=24pt ftext="Arial/Bold" ctext=black
  hsize=5.95in vsize=4.46in /* the image of this size, in this paper, is shrunk to fit */
  ;
pattern1 c=CX009900;
axis1 /* vertical axis */
  split=' '
  label=none major=none minor=none style=0;
axis2 /* horizontal axis */
  value=none
  label=none major=none minor=none style=0;
title1 height=8pt color=white 'White Space';
%if &Shape NE 2D %then %do;
filename anyname
"&path\20May2026_Figure24_SASgraphStackedSplitXaxisLabelsForOptimizedVBAR3D_SHAPEeq&Shape._
WithPreferredBarWidthAndSpace.png";
title2 "Brand By Decreasing Sales & Share";
title3
  %if &Shape EQ CYLINDER %then %do;
    color=blue
    "Best 3D Shape = CYLINDER";
  %end;
  %else %do;
    color=red
    "3D Shape = &Shape";
  %end;
%end;
%else %do;
filename anyname
"&path\20May2026_Figure24_SASgraphStackedSplitXaxisLabelsForVBAR2Dchart_WithPreferredBarWid
thAndSpace.png";
title2 "Brand By Decreasing Sales & Share";
title3 color=blue "2D = Easily Interpreted";
%end;
proc gchart data=work.ToBarCharts;
%if &Shape NE 2D %then %do;
```

```
vbar3d BrandAndSalesAndPercent / sumvar=Sales
  descending
  coutline=CX009900
  noplane /* essential for acceptable 3D */
  noframe /* essential for acceptable 3D */
  shape=&Shape
  Width=&Width
  Space=&Space
%end;
%else %do;
vbar BrandAndSalesAndPercent / sumvar=Sales
  descending
  coutline=CX009900
  width=&Width
  space=&Space
%end;
  maxis=axis1 raxis=axis2;
run; quit;
filename anyname clear;
%mend VbarChart;

%VbarChart(2D,7,11);
%VbarChart(CYLINDER,7,11);
/* available shapes are
   2D, CYLINDER (best), BLOCK, HEXAGON, PRISM, STAR */
```

Code for Figure 24 SAS/GRAPH 2D and 3D Vertical Bar Chart with Stacked Split X Axis Values.

```
data work.ToChart;
length Brand $ 1 Sales 8;
Brand='F'; Sales=80; output;
Brand='B'; Sales=40; output;
Brand='E'; Sales=38; output;
Brand='D'; Sales=22; output;
Brand='A'; Sales=10; output;
Brand='C'; Sales=10; output;
run;

%macro VbarChartXaxisTablesAndDataSkin(
DataSkin,Width=,Height=,FontSize=);
ods graphics / reset=all noscale width=&Width height=&Height
%if &DataSkin NE NONE %then %do;
  imagename="20May2026_Figure24_3DVbarChartWithXaxisTablesDataSkinEqual&DataSkin";
%end;
%else %do;
  imagename="20May2026_Figure24_2DVbarChartWithXaxisTablesUsingDataSkinEqual&DataSkin";
%end;
title1 height=&FontSize font='Arial/Bold'
  "Brand By Decreasing Sales & Share";
title2 height=&FontSize font='Arial/Bold'
%if &DataSkin NE NONE %then %do;
  %if &DataSkin EQ PRESSED %then %do;
  color=blue
  %end;
  "3D effect using DataSkin=&DataSkin";
%end;
%else %do;
  color=blue
  "2D = Easily Interpreted";
%end;
proc sgplot data=work.ToChart noborder
  pctndec=0; /* roundXaxisTable values for stat=percent to integers */
vbar Brand /
  response=Sales
  dataskin=&DataSkin /* for 3D-ish effect
     options are NONE (the 2D default),
```

```
     CRISP, GLOSS, MATTE, PRESSED, SHEEN */
  categoryorder=respdesc
  displaybaseline=off
  barwidth=0.4
  nooutline
  fillattrs=(color=CX009900);
xaxis display=(nolabel noline noticks)
  fitpolicy=none /* needed to prevent thinning the axis values */
  valueattrs=(family=Arial size=&FontSize weight=Bold);
xaxistable Sales /
  location=outside
  position=bottom
  valueattrs=(family=Arial size=&FontSize weight=Bold)
  label=' ';
xaxistable Sales /
  stat=percent
  location=outside
  position=bottom
  valueattrs=(family=Arial size=&FontSize weight=Bold)
  label=' ';
yaxis display=none;
run;

%mend VbarChartXaxisTablesAndDataSkin;

title; footnote;
ods results off;
ods _all_ close;
ods listing gpath="C:\temp" dpi=300;

/* 16pt is the maximum FontSize to fit titles with wrapping
   at image width=4.44in  */
options mprint;
%VbarChartXaxisTablesAndDataSkin(
NONE,Width=4.44in,Height=3.33in,FontSize=16pt);
%VbarChartXaxisTablesAndDataSkin(
PRESSED,Width=4.44in,Height=3.33in,FontSize=16pt);

ods listing close;
```

Code for Figure 24 ODS Graphics 2D and Pseudo3D Vertical Bar Chart with X Axis Tables.

CLOSING THE BAR CHARTS SECTION WITH FOCUS ON THE BEST

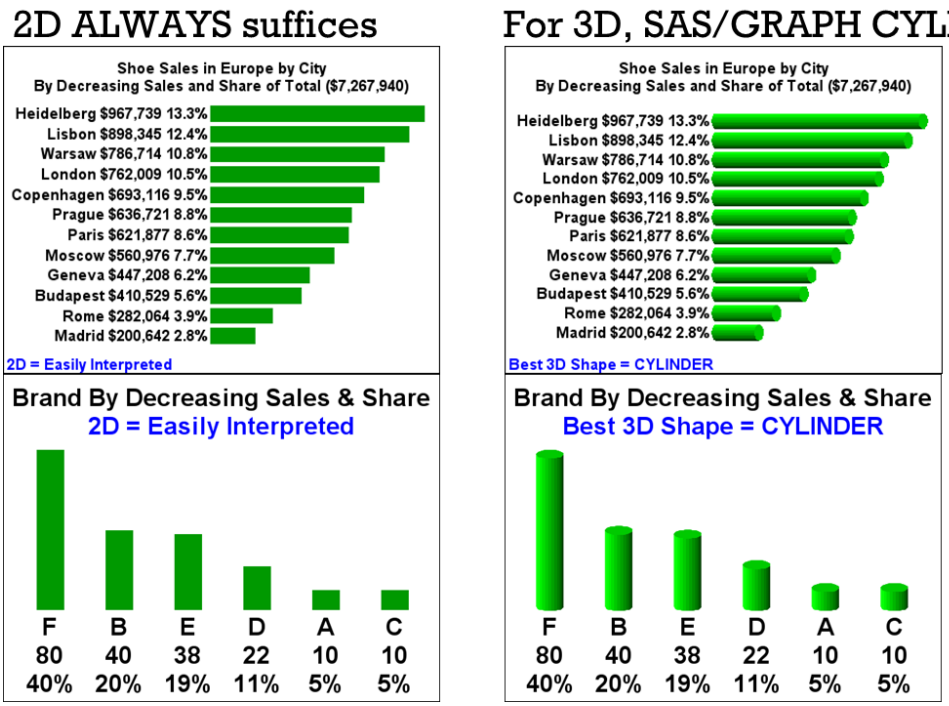

Figure 25. The Absolutely Sufficient Set of Communication-Effective Bar Chart Alternatives.

Horizontal Bar Charts accommodate more information. That's why the vertical bar chart examples above use fake skimpy category values instead of realistic ones.

SAS/GRAPH defaults for 3D are awful. For the PROC GCHART HBAR3D and VBAR3D statements, options NOPLANE NOFRAME are essential anti-defaults.

The Optimized SAS/GRAPH Cylinder Chart is the Best SAS 3D Chart. Use SHAPE=CYLINDER on the HBAR3D and VBAR3D statements.

ODS Graphics 3D-ish Effect DATASKIN options "PRESSED" & "SHEEN" are OK for pies and donuts,

If you do not have SAS/GRAPH, on the ODS Graphics HBAR or VBAR statement, use option DATASKIN=PRESSED. Alternatively, consider DATASKIN=CRISP for a "button bar look" or DATASKIN=MATTE for "bulging buttons".

No Additional Charge ODS Graphics comes with the Base SAS license.

DISPLAYING IMAGES IN A WEB PAGE WITH TABLES

Figure 26. A standalone table in the clipped top of an ODS HTML5 web page with the default ODS style. The web page has a light blue background color, with no communication benefit, but which does reduce contrast with text. The default color for text is unexpectedly (at least for me) NOT Black. Except on dark background colors, Black is ALWAYS the most readable color. On dark backgrounds, White text is always most readable.

```
title;footnote;
ods results off;
ods _all_ close;
ods html5
/* UNcomment this if the web page contains an image
   that can be imbedded
   to make the web page relocatable/portable as only one file
   options(bitmap_mode='inline') */
    path="C:\temp"
    body = "19May2026_Figure26.html"
    (title="Web Page Table Using Default Style For ODS HTML5");
title1 "Web Page Table Using Using Default Style For ODS HTML5";
proc print data=sashelp.class(where=(name =: 'A')) label;
id name;
var age;
run;
ods html5 close;
```

Code for Figure 26.

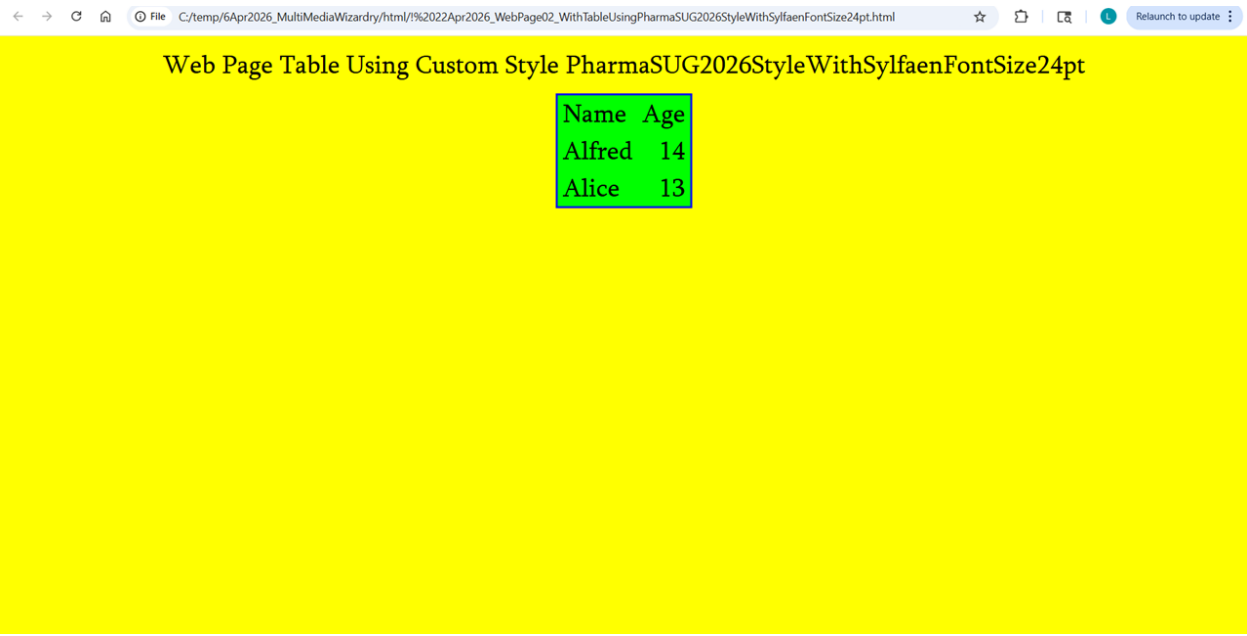

Figure 27. Custom ODS Style Web Page with a Table, But No Images. Use of Yellow for the web page background, Green for the table background, and Blue for table border is not really necessary. The custom colors have no communication purpose, but for presentation purposes they add interest. The default style has a table grid, and all table cells except the data cells have a needless color fill. Here, all table cells have color, needless except for added visual interest. All parts of a table are equally important. If colored, all cells should be the same “equal” color.

```
%macro MakeStdStyle(
TextColor=black,
StyleName=,
Font =,
TableTitleFootnoteSize =,
TableHeadingSize       =,
TableDataSize          =,
ConspicuousColor       = CXFF00FF, /* magenta or fuschia */
WebPageBackgroundColor = CXFFFF00, /* yellow */
TableFrameColor        = CX0000FF, /* blue */
TableBackgroundColor   = CX00FF00, /* green */
TableGrid              = NO,
TableFrame             = BOX,
TableSpacing           = 9);

proc template;

  edit styles.Default as styles.&StyleName; /* START of the EDIT */
  /* Create a modified style based on the ODS STYLES.DEFAULT.
     Anything not referenced or overridden here
     will be inherited from the ODS Default Style. */

style fonts
  "Some of the fonts in STYLES.DEFAULT, but customized by LeRB" /

  'TitleFont'      = ("&Font, Times New Roman, Times", &TableTitleFootnoteSize)
   /* "system" titles and footnotes                    */

  'HeadingFont'    = ("&Font, Times New Roman, Times", &TableHeadingSize)
   /* col & row hdgs (including obs number and id var value) */

  'DataFont'       = ("&Font, Sylfaen, Helvetica", &TableDataSize)
   /* table data. This font added by LeRB. */

  'DocFont'        = ("Comic Sans MS, Courier",1);
   /* default for unassigned fonts.
      Unusual font chosen for recognizability
      if it should actually be used by ODS.
      Ddue to a minor bug,
      although 'DocFont' has no effect on font faces elsewhere,
      its font size DOES affect space above TITLE and FOOTNOTE line.
      Assignment of 1 makes that space the minimum possible.
      When I say "has no effect on font faces elsewhere",
      that is true only if this macro also includes
      "style Data from Cell / font = fonts('DataFont');"
      which appears later in this macro.
      See the comment there. */

style colors

  "Some colors used in STYLES.DEFAULT, but customized by LeRB,
  plus new ones added by LeRB. " /

  'systitlefg'  = &TextColor
  /* "system" titles & footnotes */

  'systitlebg'  = &WebPageBackgroundColor
  /* background for "system" title/footnote areas */
```

```
  'headerfg'    = &TextColor
  /* override fgA2, table row & column labels */

  'headerbg'    = &TableBackgroundColor
  /* background for table row & column labels */

  'datafg'      = &TextColor
  /* table data */

  'databg'      = &TableBackgroundColor
  /* background for table data */

  'docfg'       = &ConspicuousColor
  /* foreground (i.e., text color) where not overridden.
     Bright color chosen for noticeability
     if it should actually be used by ODS. */

  'docbg'       = &WebPageBackgroundColor;
  /* background for web page and ??? */

style SysTitleAndFooterContainer from Container /

  cellpadding = 0
  /* override 1 to compact the title/footnote area */
  cellspacing = 0;
  /* override 1 to remove the grid in title/footnote area */

style Output from Container /

  /* next three lines control table grid and table border */
  rules = NONE /* override GROUPS.
                  NO line between table labels & data */
%if %upcase(&TableGrid) eq NO
%then %do;
  frame = &TableFrame /* BOX for on, VOID for off */
  cellspacing = 0 /* override 1. Space between cells:
                     if zero, grid is not visible. */
%end;
%else %do;
  frame = VOID
%end;

  cellpadding = &TableSpacing /* override 7 */
  /* separation of rows and columns */

  bordercolor = &TableFrameColor
  /* color of table frame and table rules/grid */

  borderwidth = 3; /* default is 1 */

style Data from Cell /
  font = fonts('DataFont');
/*  Added to override default use of DocFont */

style body /
  pagebreakhtml=_undef_; /* suppress rule between
  successive PROC outputs to get them on the same web page */

  end; /* END of the EDIT */

run;

%mend MakeStdStyle;

options mprint;
```

```
%MakeStdStyle(StyleName=PharmaSUG2026StyleWithSylfaenFontSize24pt,
Font=Sylfaen,
TableTitleFootnoteSize = 24pt,
TableHeadingSize       = 24pt,
TableDataSize          = 24pt);

title;footnote;
ods results off;
ods _all_ close;
ods html5
    path="C:\temp"
    body = "19May2026_Figure27.html"
    (title="Web Page Table Using Custom Style PharmaSUG2026StyleWithSylfaenFontSize24pt")
    style=styles.PharmaSUG2026StyleWithSylfaenFontSize24pt;
title1 "Web Page Table Using Custom Style PharmaSUG2026StyleWithSylfaenFontSize24pt";
proc print data=sashelp.class(where=(name =: 'A')) label;
id name;
var age;
run;
ods html5 close;
```

Code for Figure 27. The custom ODS style created here will be used as the style (or as PARENT style for a derivative style) in later examples.

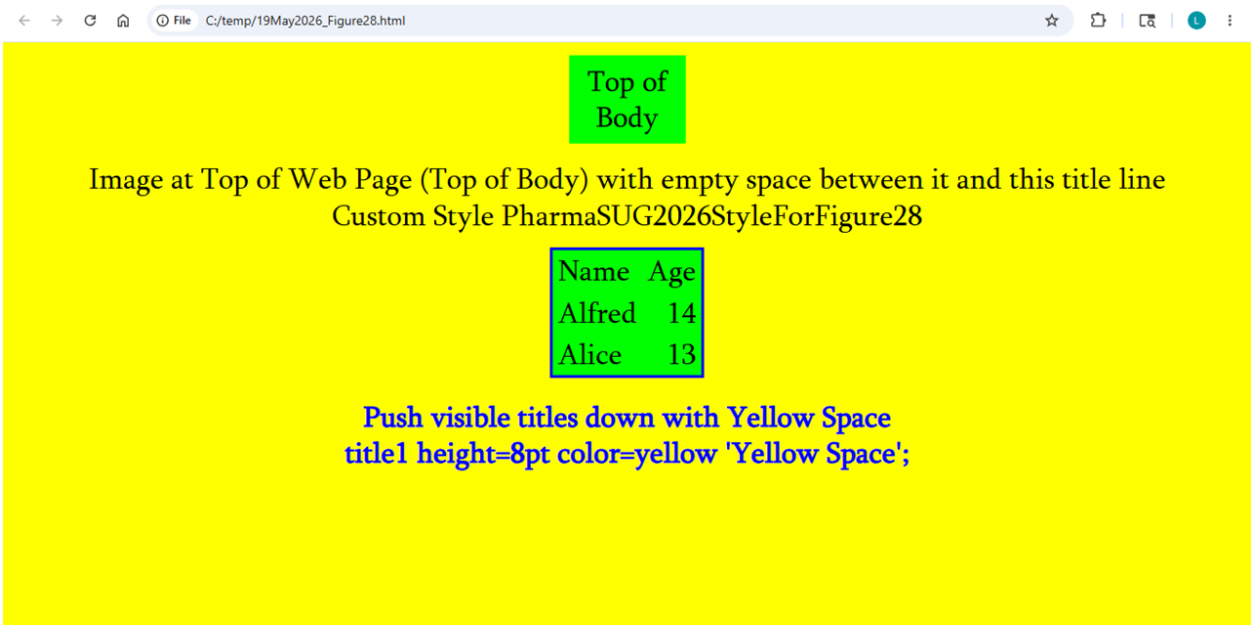

Figure 28. Footnote explains the code.

```
%let path = C:\temp\;
%let imagepath = &path\ImbeddedImagesForWebPagesWithTables;

proc template;
edit styles.PharmaSUG2026StyleWithSylfaenFontSize24pt
  as styles.PharmaSUG2026StyleForFigure28;
style body / preimage = "&imagepath\TopOfBody.gif";
end;
run;

title;footnote;
ods _all_ close;
ods noresults;
ods html5 options(bitmap_mode='inline')
  /* make the web page relocatable / portable */
```

```
    path="&path"
    body = "19May2026_Figure28.html"
    (title="Image at Top of Web Page")
    style=styles.PharmaSUG2026StyleForFigure28;
title1 height=8pt color=yellow 'Yellow Space';
title2 "Image at Top of Web Page (Top of Body) with empty space between it and this title
line";
title3 "Custom Style PharmaSUG2026StyleForFigure28";
footnote1 color=blue bold "Push visible titles down with Yellow Space";
footnote2 color=blue bold "title1 height=8pt color=yellow 'Yellow Space';";
proc print data=sashelp.class(where=(name =: 'A')) label;
id name;
var age;
run;
ods html5 close;
```

Code for Figure 28. To see the adverse consequences of no Yellow space, delete TITLE1, and change TITLE2 to TITLE1, and TITLE3 to TITLE2.

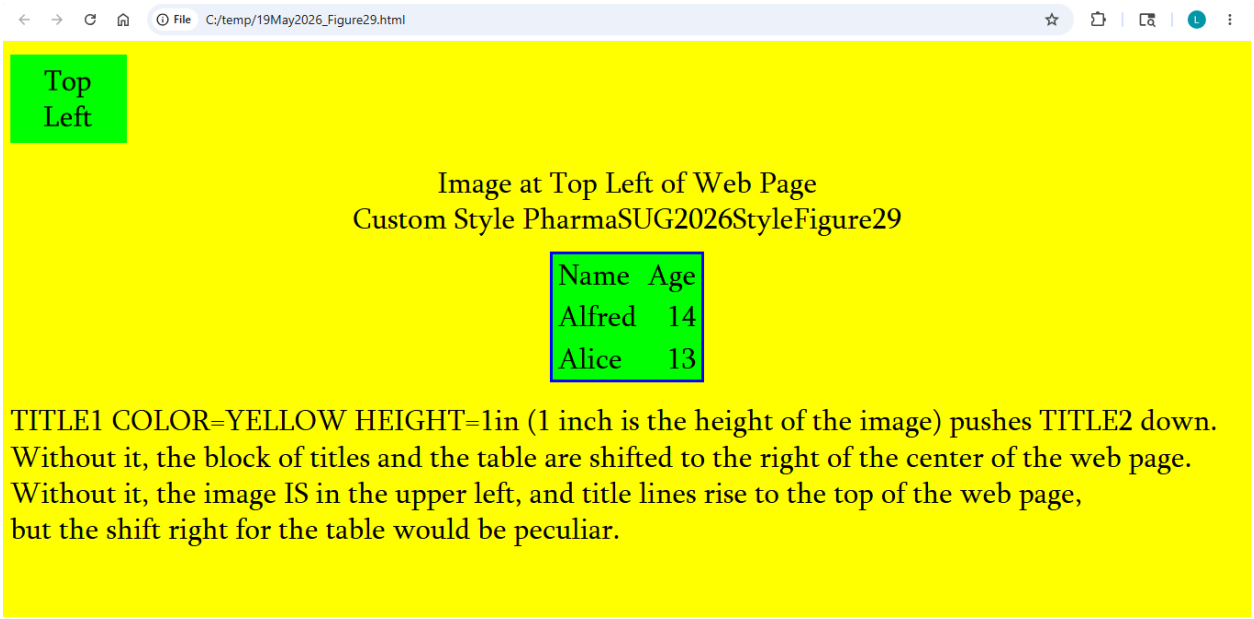

Figure 29. Image Shifted to Upper Left Corner.

```
%let path = C:\temp\;
%let imagepath = &path\ImbeddedImagesForWebPagesWithTables;

proc template;
edit styles.PharmaSUG2026StyleWithSylfaenFontSize24pt
  as styles.PharmaSUG2026StyleFigure29;
style body /
  prehtml = "<img src='&imagepath\TopLeft.gif' align='left'>";
  /* Use PREHTML to apply location controls.
     PREIMAGE would simply place the image at the middle
     at the top of the web page body. */
end;
run;

title;footnote;
ods _all_ close;
ods noresults;
ods html5 options(bitmap_mode='inline')
     path="&path"
     body = "19May2026_Figure29.html" (title="Image at Top Left of Web Page")
     style=styles.PharmaSUG2026StyleFigure29;
```

```
title1 color=yellow height=1in "yellow space";
title2 "Image at Top Left of Web Page";
title3 "Custom Style PharmaSUG2026StyleFigure29";
footnote1 justify=left 'TITLE1 COLOR=YELLOW HEIGHT=1in (1 inch is the height of the image)
pushes TITLE2 down.';
footnote2 justify=left  'Without it, the block of titles and the table are shifted to the
right of the center of the web page.';
footnote3 justify=left  'Without it, the image IS in the upper left, and title lines rise
to the top of the web page,';
footnote4 justify=left  'but the shift right for the table would be peculiar.';
proc print data=sashelp.class(where=(name =: 'A')) label;
id name;
var age;
run;
ods html5 close;
```

Code for Figure 29. To see the adverse consequences of no Yellow space, delete TITLE1, and change TITLE2 to TITLE1, and TITLE3 to TITLE2.

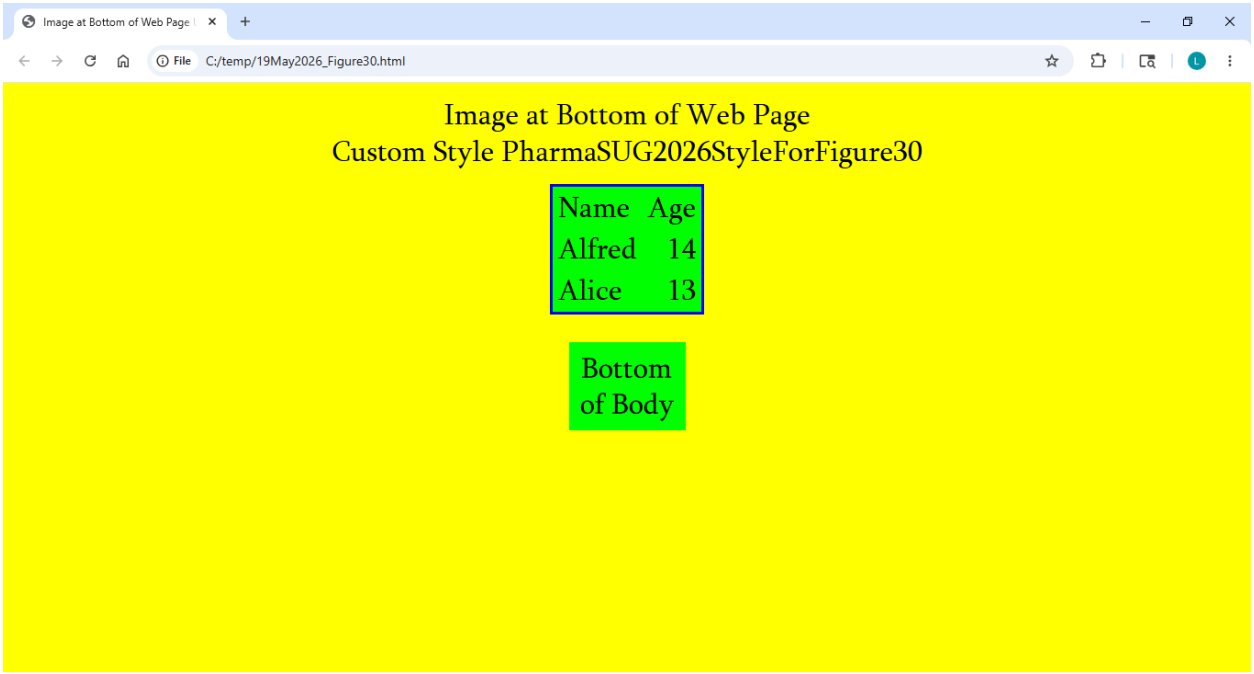

Figure 30. For this placement nothing special needed to be done other than create the ODS style.

```
%let path = C:\temp\;
%let imagepath = &path\ImbeddedImagesForWebPagesWithTables;

proc template;
edit Styles.PharmaSUG2026StyleWithSylfaenFontSize24pt as
Styles.PharmaSUG2026StyleForFigure30;
  style body / postimage="&imagepath\BottomOfBody.gif";
end;
run;

title; footnote;
ods noresults;
ods _all_ close;
ods html5 options(bitmap_mode='inline')
  path = "&path"
  body = "19May2026_Figure30.html" (title="Image at Bottom of Web Page Using POSTIMAGE
Option")
  style=Styles.PharmaSUG2026StyleForFigure30_UsingPOSTIMAGE;
```

```
title1 "Image at Bottom of Web Page";
title2 "Custom Style PharmaSUG2026StyleForFigure30";
proc print data=sashelp.class(where=(name =: 'A')) label;
id name;
var age;
run;
ods html5 close;
```

Code for Figure 30. Nothing remarkable here.

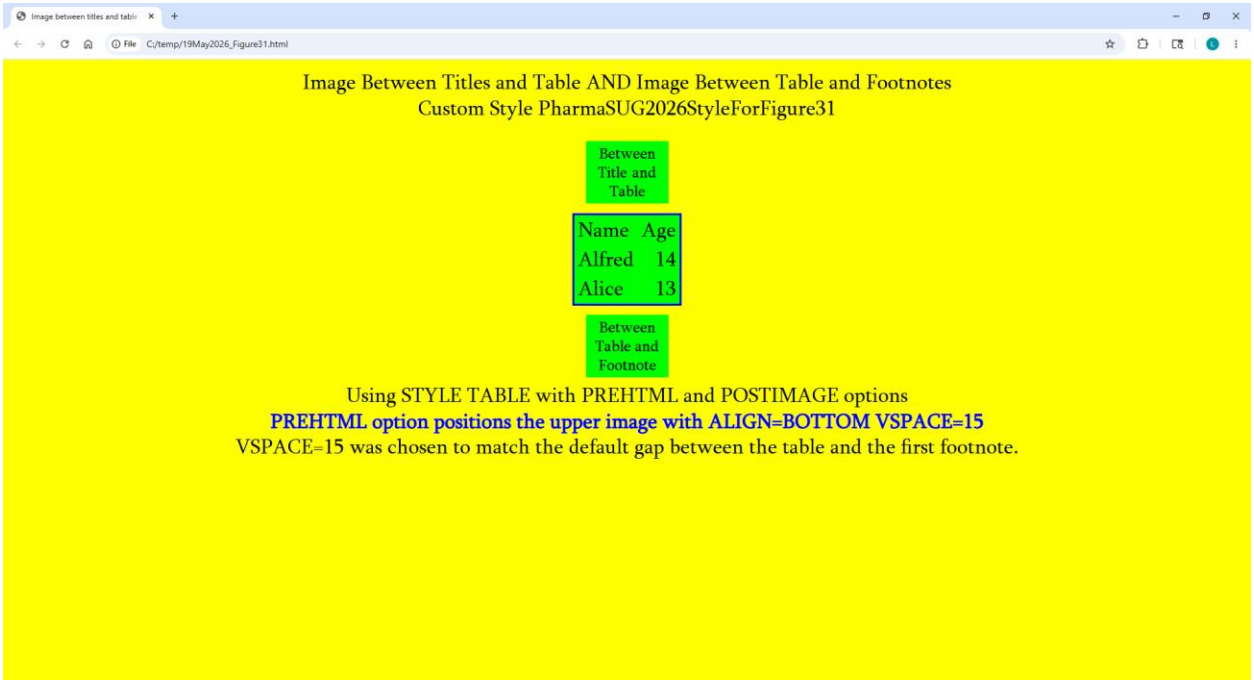

Figure 31. Here, too, the only placement and adjustment work is in the ODS style.

```
%let path = C:\temp\;
%let imagepath = &path\ImbeddedImagesForWebPagesWithTables;

proc template;
edit Styles.PharmaSUG2026StyleWithSylfaenFontSize24pt
  as Styles.PharmaSUG2026StyleForFigure31;
style table /
  prehtml = "<img src='&imagepath\BetweenTitleAndTable.gif' align='bottom' vspace=15>"
  postimage =  "&imagepath\BetweenTableAndFootnote.gif";
end;
run;

title; footnote;
ods noresults;
ods _all_ close;
ods html5 options(bitmap_mode='inline')
  path = "&path"
  body = "19May2026_Figure31.html"
  (title="Image between titles and table and Image between table and footnotes")
  style=Styles.PharmaSUG2026StyleForFigure31;
title1 "Image Between Titles and Table AND Image Between Table and Footnotes";
title2 "Custom Style PharmaSUG2026StyleForFigure31";
footnote1 "Using STYLE TABLE with PREHTML and POSTIMAGE options";
footnote2 color=blue bold
  "PREHTML option positions the upper image with ALIGN=BOTTOM VSPACE=15";
footnote3 "VSPACE=15 was chosen to match the default gap between the table and the first
footnote.";
```

```
proc print data=sashelp.class(where=(name =: 'A')) label;
id name;
var age;
run;
ods html5 close;
```

Code for Figure 31. PREHTML provides positional control that PREIMAGE would not.

Figure 32A. Watermark Background Image At the Open of the Web Page.

Figure 32B. Watermark Background Image At the Bottom of the Table Scroll.

```
proc template;
  define style styles.PharmaSUG2026logoAsWebPageWatermark;
```

```
  parent=styles.HTMLencore;
  style body from document /
  backgroundcolor=white
  backgroundrepeat=no_repeat
  htmlstyle="background-image: url('C:\temp/PharmaSUG 2026 Logo.gif');
             background-attachment: fixed;
             background-size: contain;
             background-position: top left;";
  style SysTitleAndFooterContainer / background = _undef_;
  style systemtitle                / background = _undef_;
  style systemfooter               / background = _undef_;
  end;
run;

title1; footnote1;
options center;
ods noresults;
ods _all_ close;
ods html options(bitmap_mode='inline') /* make the web page relocatable / portable */
  path="C:\temp"
  body="20May2026C_HTML5tableOnWebPageWithGIFwatermark.html"
  (title="HTML5 Web Page with GIF Watermark")
  style=styles.styles.PharmaSUG2026logoAsWebPageWatermark;

title1 justify=right font=Sylfaen height=16pt bold color=Blue
  "This is an HTML5 Web Page with a GIF Logo Watermark";
title2 justify=right font=Sylfaen height=16pt bold color=red
  "As you scroll, the Watermark is stationary";

proc print data=sashelp.class noobs
  style(header) = [foreground=Blue background=Yellow fontsize=24pt fontfamily=Sylfaen
font_weight=bold]
  style(data)   = [foreground=Blue background=Yellow fontsize=24pt fontfamily=Sylfaen
font_weight=bold];
var Name;
run;

ods html close;
```

Code for Figure 32. Watermark Background Image Locked In Place During Web Page Scroll.

EVEN IN MULTI-MEDIA, COMMUNICATION EFFECTIVENESS MATTERS

ACCESSIBLE COLOR

About 25 years ago, some SAS users/authors began a fad of so-called "Traffic Lighting" to color code Bad & Good with Red & Green. Since then, I have been informing SAS users that the commonest form of color blindness cannot distinguish Red from Green.

READABLE TEXT

Provide sufficient contrast. Light backgrounds need dark (i.e., Black) text. Dark backgrounds need light (i.e., White) text. Red and Blue are dark colors.

COMMUNICATION-EFFECTIVE COLOR USE

For color to be distinguishable (i.e., to answer the question “What color is it?”), color requires **mass**. Markers and legend color swatches must be big enough. Text and lines must be thick enough.

COUNTERPOINT TO COMPLEXITY AND FEATURE RICHNESS, IF YOUR INFORMATION PACKAGE INCLUDES TIME SERIES . . .

BEST USE OF THE AXES FOR THE TIME SERIES PLOTS

When tracking performance versus standard, limit the Y axis range to the actual minimum through maximum. Use an X axis range that starts and ends with the same month (day) of the years (weeks) span. It may reveal cyclic behavior.

UNIVERSAL KEY QUESTION FOR TIME SERIES PLOTS

What ARE the X & Y values? Explicitly display them. But HOW?

Do not require viewers to run their eyes from plot points to the Y axis, and then to interpolate a precise Y value based on its embracing axis values.

You might be able to annotate the plot points. Including both the Y value and the X value might be impractical due to collisions between adjacent values, or (less likely) between the annotation and the plot line. Annotating the Y value only leaves it potentially distant from its X value. An X Axis Table displays Y values in immediate proximity to (i.e., above) their X values.

If both annotation and an X axis table are infeasible, use Sparse Line Annotation. It Can Suffice.

SPARSE LINE ANNOTATION

On a line with no axes, annotate the Y value and the X value for Start, End, Minimum Y, Maximum Y, and Amount and Direction of the Last Change. In a time series plot, knowing the values for EVERY point is unlikely to be of interest. One is normally only interested in the last point. This concept is not to be confused with the Spark Line (or Sparkline) of Peter Zelchenko, which appeared several years after I first demonstrated Sparse Line Annotation. A spark line is a line only, uninforming as compared to Sparse Line Annotation.

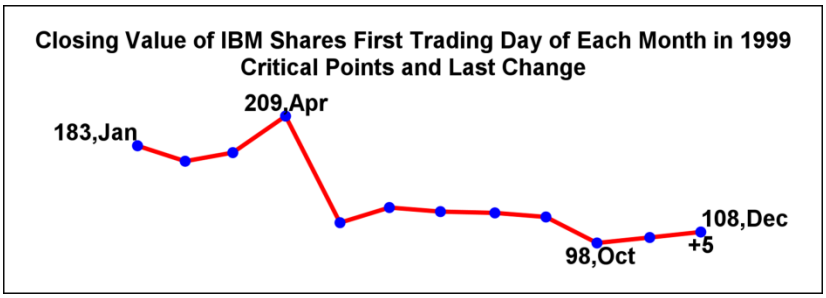

Figure 32. Sparse Line Annotation. The simplest, importantly informative, plot that you can make. The Sparse Line visual puts the attention on the four always most interesting data points, and the annotation delivers those most important values precisely, as well as what is the direction and precise value of the always most interesting change, the latest one. You can also do a table of years of monthly plots, months of weekly or daily plots, or whatever best presents the tracking cycle over the longer period.

CONCLUSION

The really key concern for data graphics is to make them communication-effective: quick easy understanding from a visual, but correct understanding from precise numbers. The pie charts, donut charts, bar charts, and sparse line for time series shown here all meet that standard, that objective.

However, the main purpose of this paper is to show how to add more communication channels: video, audio, marquee, animation, and safe 3D.

Safe 3D really only adds superfluous dimension to the visual impression, but, if done well, it remains harmless (unlike the always dangerous 3D pie chart that is doable with SAS/GRAPH).

Animation, when not simply a decoration like a waving flag, can add the dimension of time (“the fourth dimension” of physics) to the display process, not to the image itself. It is a bona fide communication tool.

RTF animation, given that it is not viewable with current versions of Microsoft Word, is only an interesting antiquity enjoyable if you happen to have an old enough version of MS Word, but it is still creatable with even the latest version of SAS software.

Data tips and flyover text for web pages and flyover text for PDF pages are helpful dynamic communication accessories that can only be activated by your mouse. For web page data tips, your code can specify variables, their labels, and their SAS format. The data tips are activated by resting your mouse on some graphic element (bar, pie slice, donut bite, plot point, etc. Web page flyover text, static or dynamic (i.e., delivered as a macro variable value) can be specified for any imbedded images—if you create the Lively Royal Web Page, you can raise flyover text for the waving flag and for the picture of the Royal Couple.

The best use of a marquee is on your TV screen during a live broadcast, but you CAN add one to a web page. The less familiar vertical version would be more likely to be usefully applied NOT at the full width of the web page. Unlike the example provided, the animated GIF could, if desired at all, perhaps more usefully be displayed outside of the marquee. However, there might be special significance for imbedding it with the marquee's text message.

Though static, rather than dynamic, flashy, or exciting, an eminently valuable enhancement that came with ODS Graphics is the ability to easily deploy data labels for almost any, if not all, graphic elements. The use of cumbersome, but helpful, and extremely valuable SAS annotation with SAS/GRAPH to do such labelling is, in ODS Graphics applications, only occasionally needed now.

So many of the SG procedures' statements to create charts and plots have the DATALABEL feature, and other options to control the text characteristics (font, size, color, etc.), SAS format of the datalabel, its position, whether multi-element content can be split, and how the split parts can be justified. Where the datalabel cannot be defined as an accessory to the graphic element, the TEXT statement may be available to display and control a datalabel. In addition, X axis tables and Y axis tables are another tool to deliver precise numbers associated to graphic elements. In summary, there is an astonishing tool box to provide the numbers for the data as an essential informative complement to the visual instrument of the graphic element. See my book for a full treatment of what is possible for all of the charts, plots, and maps that you can build with ODS Graphics, The SuperPower Tool for data graphics.

AKNOWLEDGEMENTS

At SAS Technical Support, I am grateful for the help of Martin Mincey, Marcia Surrat, Chevell Parker, and Kathryn McLawhorn. Any errors in the code or the comments therein are mine.

RELATED READING

For information about design and construction of data graphics, even if they don't dance or sing:
Bessler, LeRoy. January 2023. Visual Data Insights Using SAS ODS Graphics: A Guide to Communication-Effective Data Visualization. New York, NY: Apress.
Chapters 1 and 2 for best practices for graphic design & use of color are applicable for any software. Remaining chapters implement the best practices. The content can apply to any application.

CONTACT INFORMATION

Your comments, questions, and requests for code are valued and encouraged. Contact the author at:

LeRoy Bessler PhD
Bessler Consulting and Research
Visual Data Insights™
Strong Smart Systems™
BesslerLeRoy2024@gmail.com
Valparaiso, Indiana, USA