

Biographies

Jeff has 16 years of clinical oncology programming experience with a focus on developing macros, creating graphics, and producing reports. Jeff is a Senior Manager of Statistical Programming in Innovation at Regeneron focusing on building macro tools and improving infrastructure.

Jeff has presented papers at multiple conferences including PharmaSUG, MidWest SUG, and SAS Global Forum.

P H U S E U S C o n n e c t 2 0 2 6

DV07: Combining Trigonometry, Functions, and the Polygon Plot to Create Complex Figures Not Natively Available in SAS Including Sankey, Sunburst, and CIRCOS Charts

Jeffrey Meyers

REGENERON®

Agenda

- The Polygon plot strategy
- CIRCOS chart application
- Sankey chart application
- Sunburst chart application

The Polygon Plot Strategy

- Polygon plot: marks a series of coordinates in a chart that are then connected in order and filled in to make a polygon shape
- Functions are combined with data step do loop to output coordinates to build specific shapes
- Strategy: combine these two components together to build figure shapes not available by default

The Polygon Plot Strategy

- **Trigonometry**
 - Simulate polar axes
 - Build circles, wedges, doughnuts
- **Curved rectangles**
 - Rectangles of controlled width with a specific arc
 - Rectangles that change width over time
- **Control location and size of shapes**
 - Useful for annotation: counts, labels, etc.

The Polygon Plot Strategy

Polygon Plot

- Three required inputs:
 - X: x-coordinate values
 - Y: y-coordinate values
 - ID: value to distinguish between shapes
- Frequent options:
 - FILL/OUTLINE: turn on/off the fill and outline
 - TRANSPARENCY: set transparency of fill color
- Example syntax:
 - POLYGON X=Xvar Y=Yvar ID=IDvar /
FILL NOOUTLINE TRANSPARENCY=0.7;

Data Step Loop

Draw a circle:

DATA circle;

radius=1;

ID=1;

do i=0 to 360 by 20;

x=radius*cos(i * CONSTANT("pi")/180);

y=radius*sin(i * CONSTANT("pi")/180);

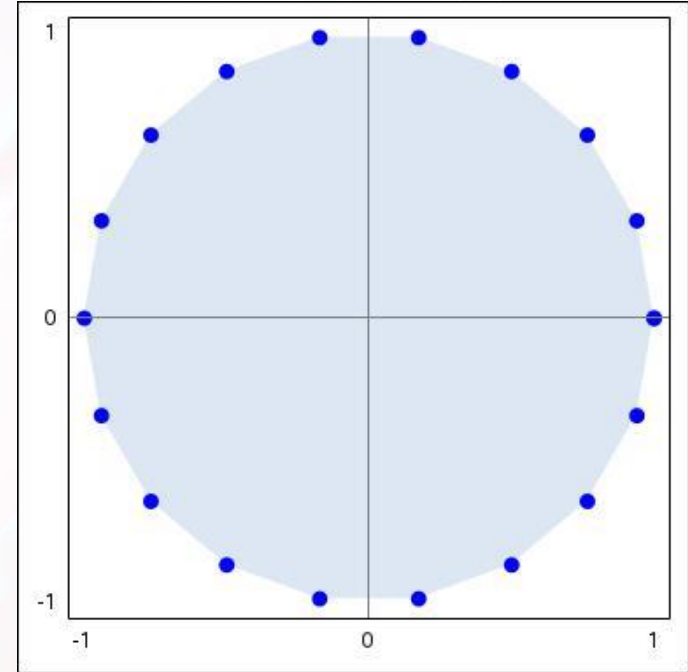
OUTPUT;

end;

Run;

The Polygon Plot Strategy

- Result is a “circle” (actually an icosagon)
- Impossible to create an actual curve, but with enough points it can be approximated
- SUBPIXEL option smooths out jagged lines

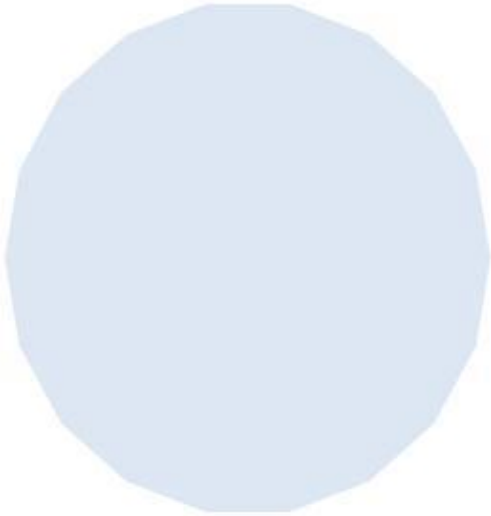


The Polygon Plot Strategy

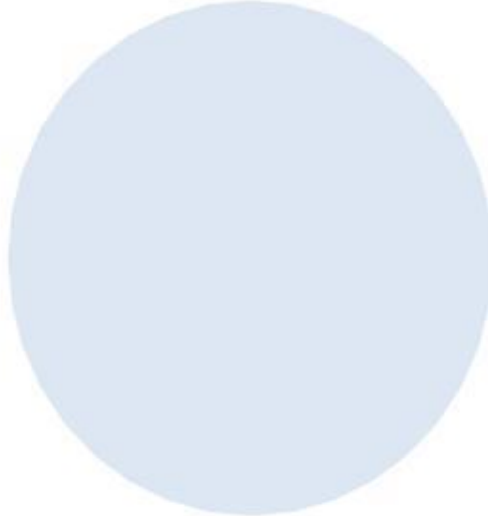
More points=smoother, but bigger file size

Diminishing returns thanks to the SUBPIXEL option smoothing objects

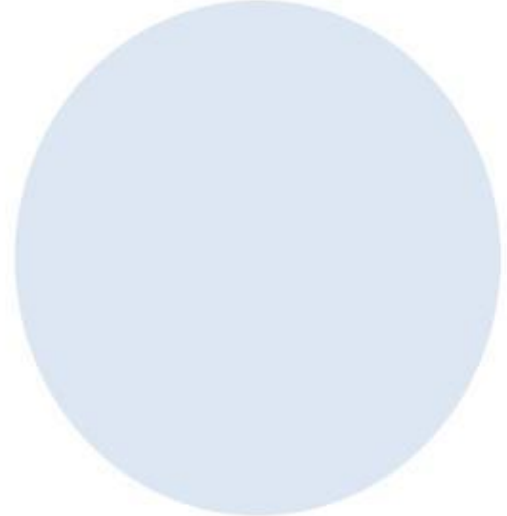
18 Points



36 Points

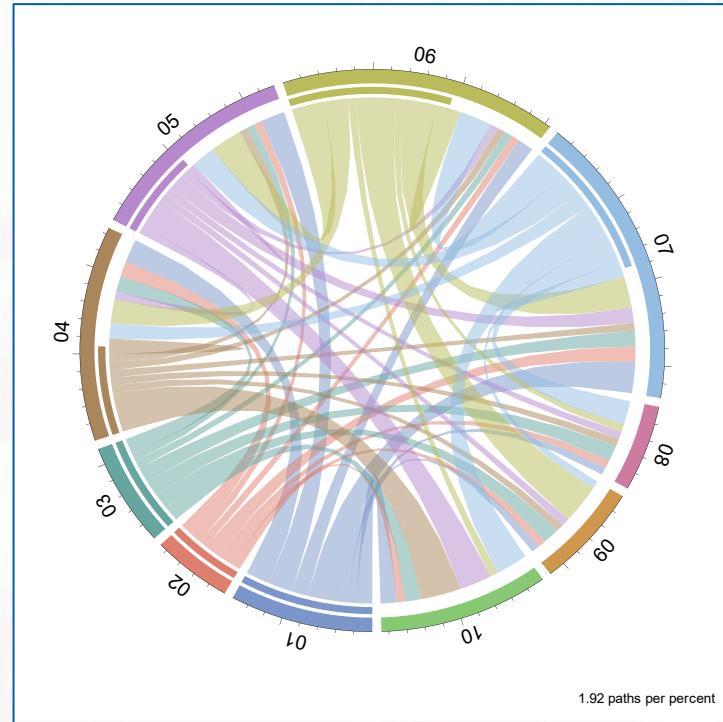


3600 Points



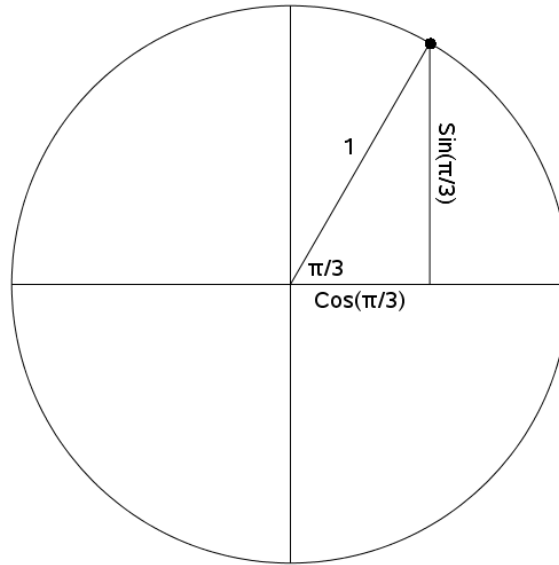
CIRCOS Chart Application

- Visual depiction of connections between groups
- Commonly used in genomics
- Four main components
 - Outer Circle (OC)
 - Inner Circle (IC)
 - Bezier Curves (BC)
 - Axes/Labels



CIRCOS Chart Application

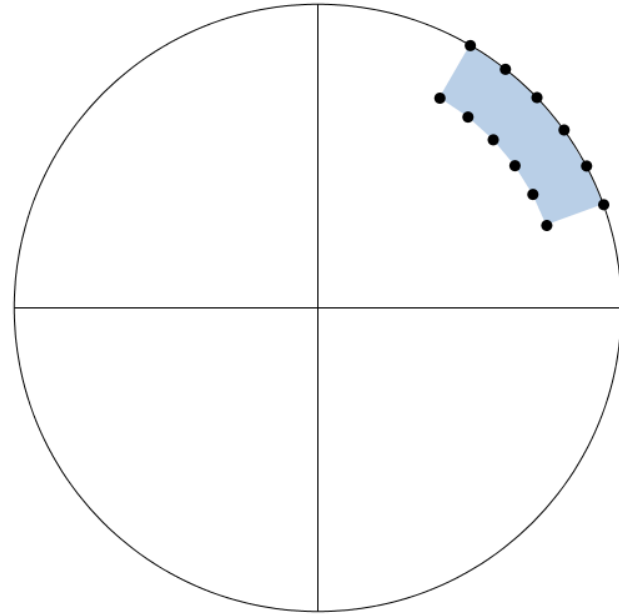
- Unit circle
 - Circle of Radius 1 centered on the (0, 0) coordinates
- Easy to calculate distances with SINE and COSINE functions
 - $X = d * \cos(\theta)$
 - $Y = d * \sin(\theta)$
- Full rotation around the circle is 360° or 2π radians
- Positive x-axis used as reference



CIRCOS Chart Application

Drawing a Curved Rectangle:

```
/*Draw Outside of Rectangle*/  
do i =  $\pi/3$  to  $\pi/9$  by  $(\pi/9 - \pi/3)/5$ ;  
    x =  $1 * \cos(i)$ ; y =  $1 * \sin(i)$ ; output;  
end;  
  
/*Draw Inside of Rectangle*/  
do i =  $\pi/9$  to  $\pi/3$  by  $(\pi/3 - \pi/9)/5$ ;  
    x =  $0.8 * \cos(i)$ ; y =  $0.8 * \sin(i)$ ; output;  
end;
```



CIRCOS Chart Application

Drawing a Bezier Curve:

Quadratic Bezier formula:

$$\mathbf{B}(t) = (1 - t)^2 \mathbf{P}_0 + 2(1 - t)t \mathbf{P}_1 + t^2 \mathbf{P}_2, 0 \leq t \leq 1$$

Connects three points: $\mathbf{P}_0$, $\mathbf{P}_1$, $\mathbf{P}_2$

$\mathbf{P}_1$ is always the center (0, 0) which reduces the equation to:

$$\mathbf{B}(t) = (1 - t)^2 \mathbf{P}_0 + t^2 \mathbf{P}_2, 0 \leq t \leq 1$$

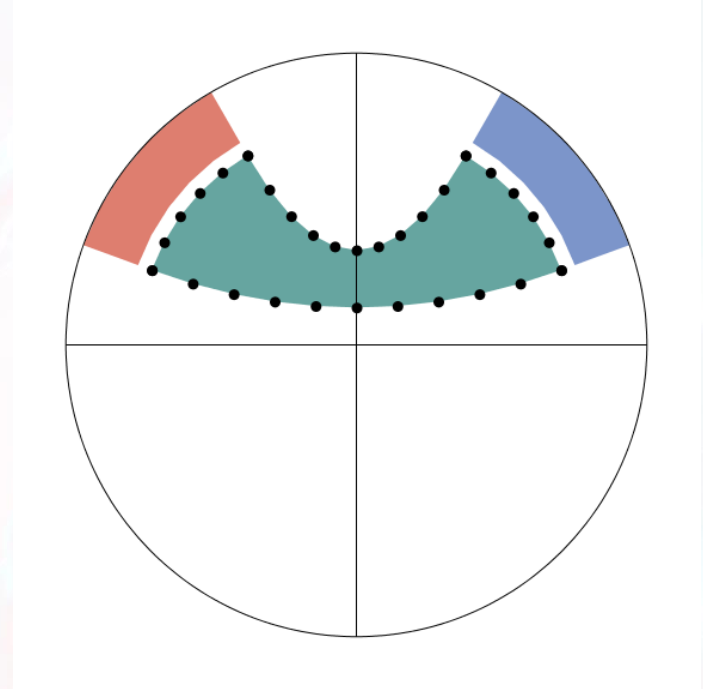
do t = 0 to 1 by 1/10;

 x=(1-t)**2*0.75*cos(π*2/3)+t**2*0.75*cos(π*1/3);

 y=(1-t)**2*0.75*sin(π*2/3)+ t**2*0.75*sin(π*1/3);

 output;

end;



CIRCOS Chart Application

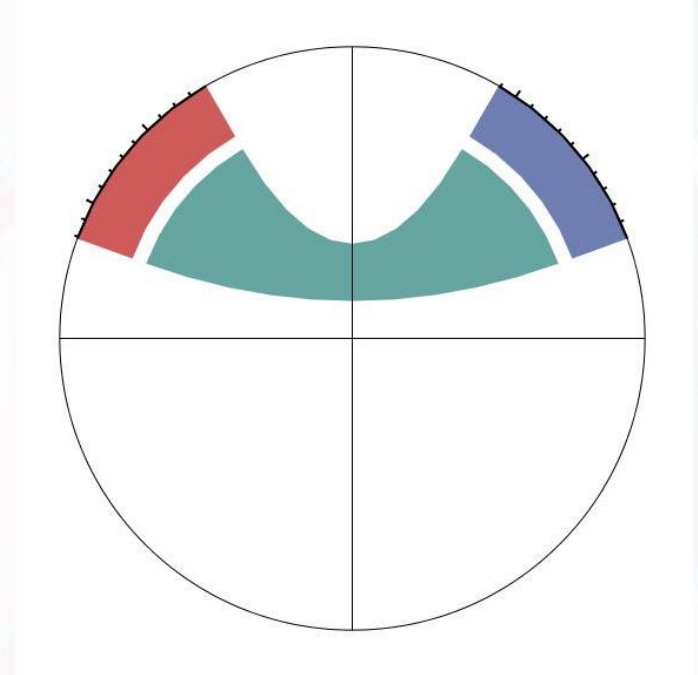
Drawing a Curved Axis:

Axis line

- Uses curved rectangles outer edge
- Drawn with a SERIESPLOT statement

Tick marks

- Drawn with SERIESPLOT statement
- GROUP option distinguishes between ticks



CIRCOS Chart Application

Drawing Rotated Labels:

Labels are drawn with TEXTPLOT

- Rotated to face the center of the circle

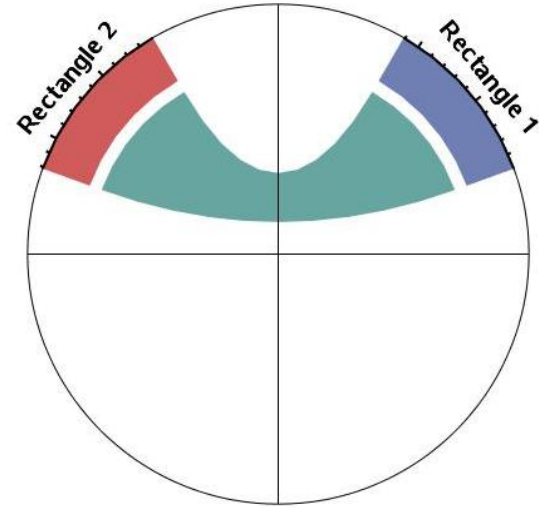
```
xlab=1.1*cos(π*2/9);
```

```
ylab=1.1*sin(π*2/9);
```

```
rotate=0-arcos(ylab/1.1)*360/(2π);
```

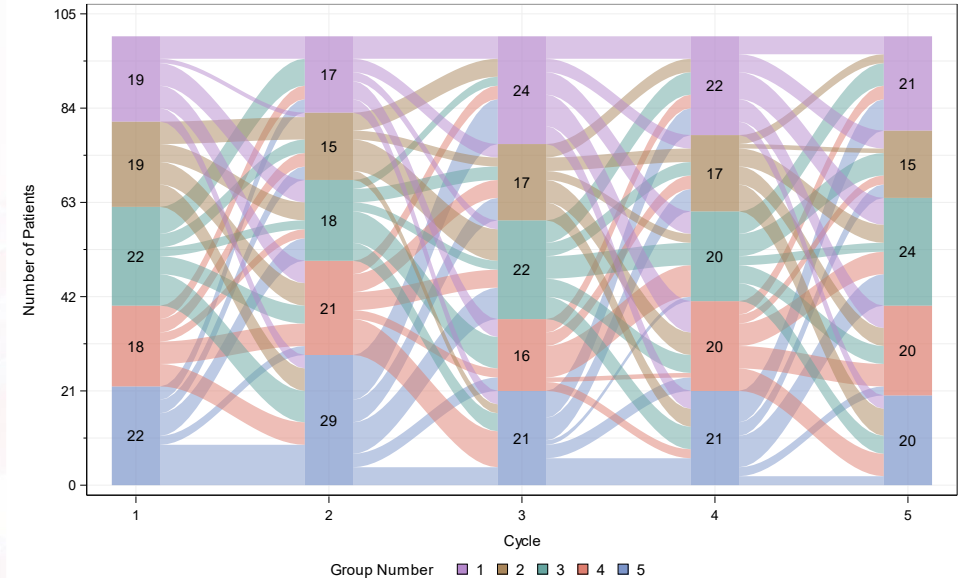
```
text='Rectangle 1';
```

```
textplot text=text x=xlab y=ylab / rotate=rotate;
```



Sankey Chart Application

- Shows flows of patients from one group to the next across time points
- Need to know exact width of bar chart to attach connecting lines
- Connecting lines are a curved rectangle with different widths at the start and end



Sankey Chart Application

- Building a bar chart with the POLYGON plot:
 - Requires four cardinal points for the rectangle's corners
 - Separate ID for each segment of the bar chart

Setup ID value for each segment

Setup bottom two corners

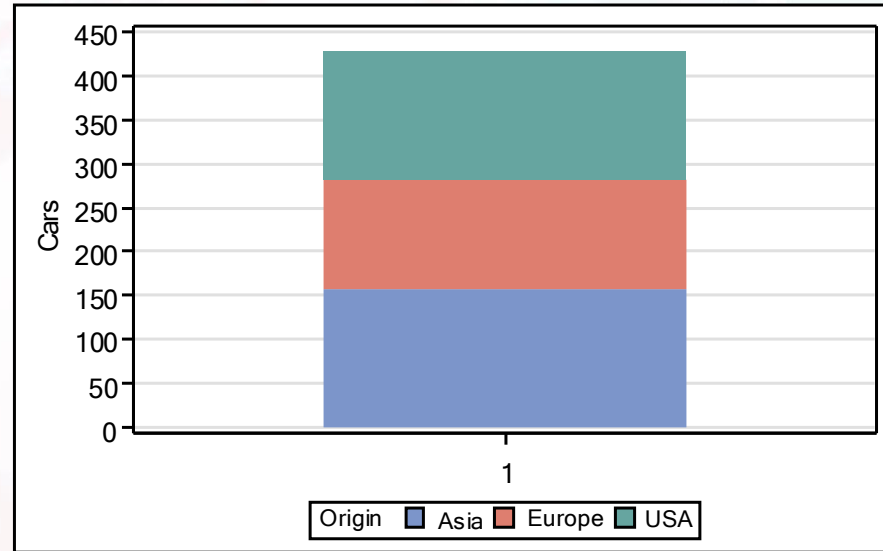
Setup top two corners

Example:

```
proc freq data=sashelp.cars noprint;
    table origin / out=freq;
run;
data barchart;
    set freq;
    if _n_=1 then start=0;
    id=_n_;
    x=0.5;y=start;output;
    x=1.5;y=start;output;
    x=1.5;y=start+count;output;
    x=0.5;y=start+count;output;
    start=start+count;
    retain start;
run;
```

Sankey Chart Application

```
proc sgplot data=barchart ;  
    polygon x=x y=y id=id / fill group=origin;  
    xaxis display=(nolabel) min=0 max=2  
        values=(1) valueshint;  
    yaxis label='Cars' min=0 max=450  
        values=(0 to 450 by 50) grid valueshint;  
run;
```



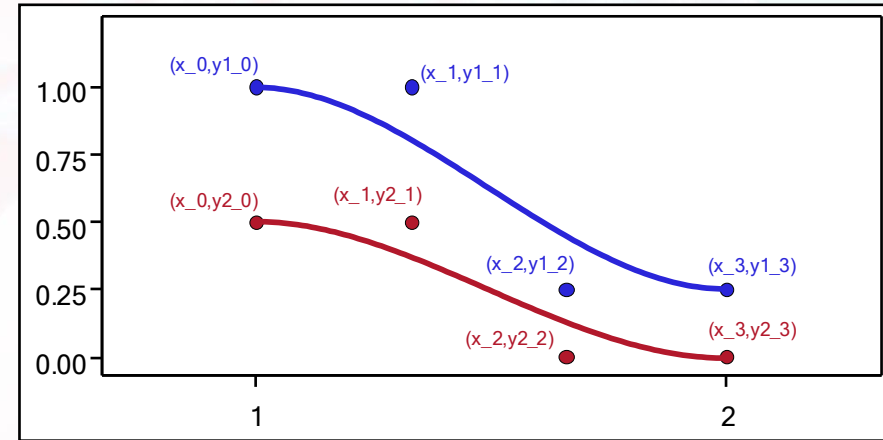
Sankey Chart Application

Connecting lines:

- Special line type called Bézier curve
- Line that has a start and end as well as at least one additional coordinate that pulls the line in that direction
- Multiple degrees of Bézier curve depending on number of additional points (quadratic, cubic, quartic)
- Used primarily for aesthetic appearance

Within the Sankey chart:

- Sankey diagram uses a cubic Bézier curve with two additional points
- Bézier lines are used to draw the top and bottom edges of the connecting lines between groups
- Allows the width of the connecting line to vary



Sankey Chart Application

Cubic Bézier function:

$$B(t) = (1 - t)^3 P_0 + 3(1 - t)^2 t P_1 + 3(1 - t) t^2 P_2 + t^3 P_3, 0 \leq t \leq 1$$

- P_0 and P_3 are start and ending coordinates
- P_1 and P_2 are coordinates to pull line
- t represents start to end of line as proportion

Within the data step:

- One side of connecting line must be drawn left to right and the return path drawn right to left
- POLYGON plot will automatically connect two Bézier curves

```
/**Bezier Curve 1: From Left group to Right Group **/
```

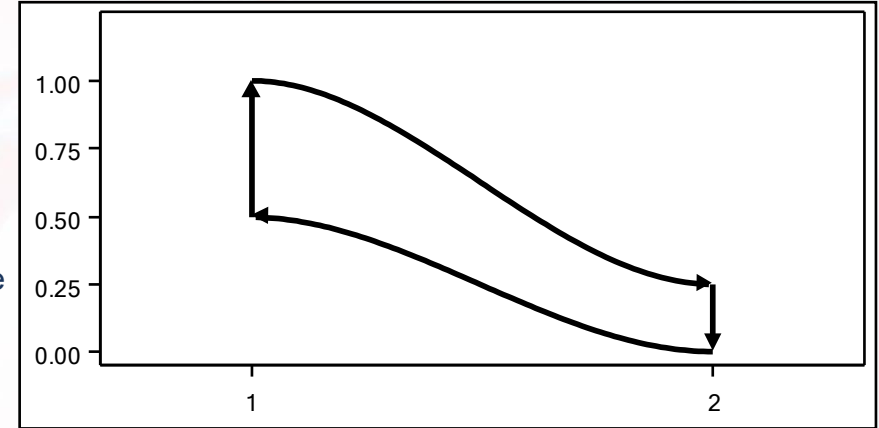
```
do t = 0 to 1 by 1/20;
```

```
x=((1-t)**3)*x_0+ 3*((1-t)**2)*t*x_1)+ 3*(1-t)*(t**2)*x_2+ (t**3)*x_3;
```

```
y=((1-t)**3)*y1_0+3*((1-t)**2)*t*y1_1+ 3*(1-t)*(t**2)*y1_3+(t**3)*y1_3;
```

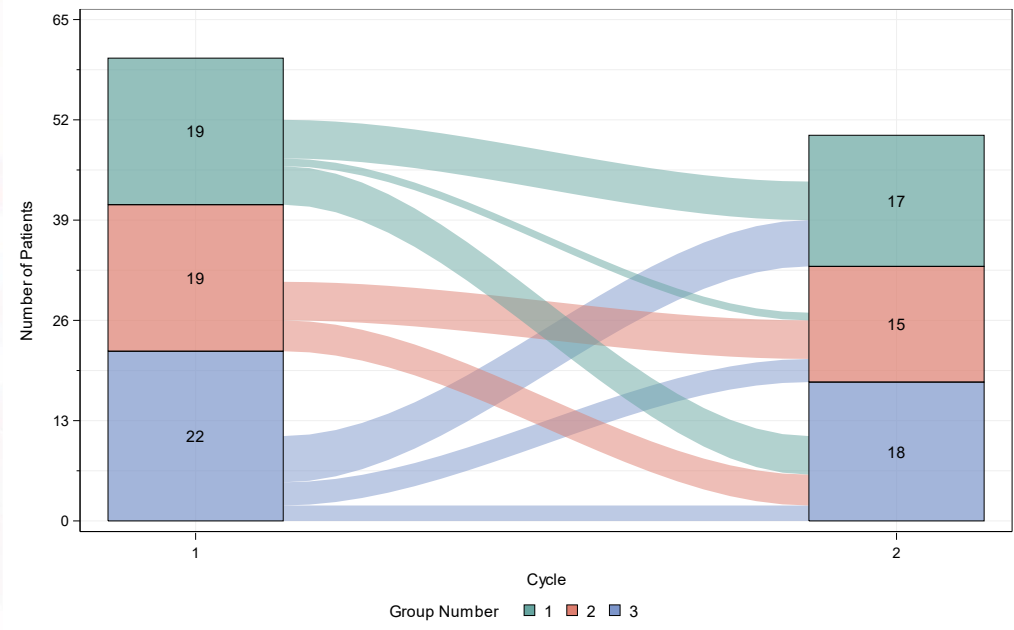
```
output;
```

```
19end;
```



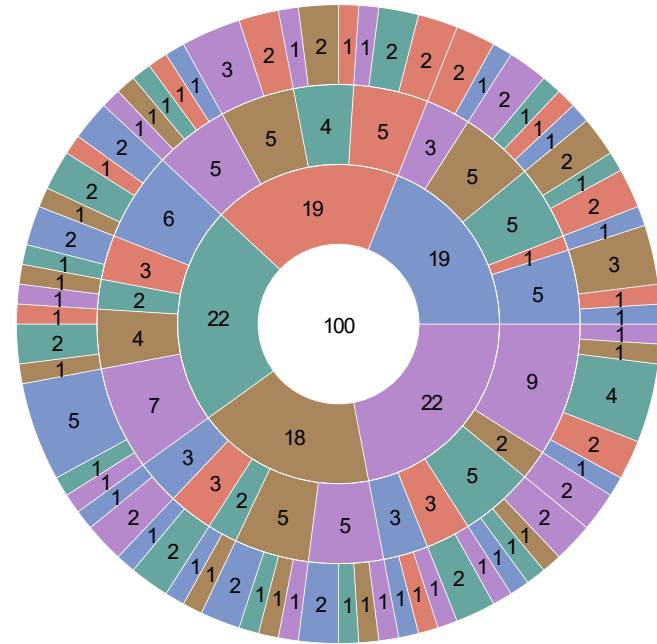
Sankey Chart Application

- Determine starting/ending points for all four corners of each Bézier curve
- Sort to avoid crisscrossing
- Separate ID value for each curve
- Color by group variable
- More transparency for Bézier curves



Sunburst Chart Application

- Shows flows of patients from one group to the next across time points (rings)
 - Easier to track specific patients vs Sankey chart
- Series of doughnut plots stacked and aligned
- Uses trigonometric equations

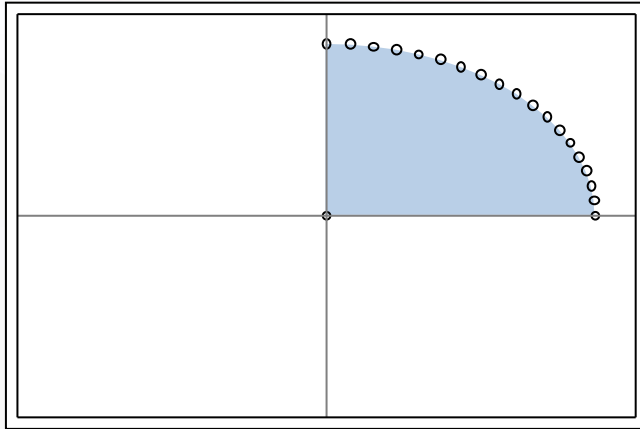


Group Number 1 2 3 4 5

Sunburst Chart Application

Building a wedge with the POLYGON plot:

1. Requires origin point, distance, and an arc
2. Using $x=0$, $y=0$ as the origin point the unit circle can be used for distance and arc
3. Trigonometric equations SIN and COS can be used to plot points



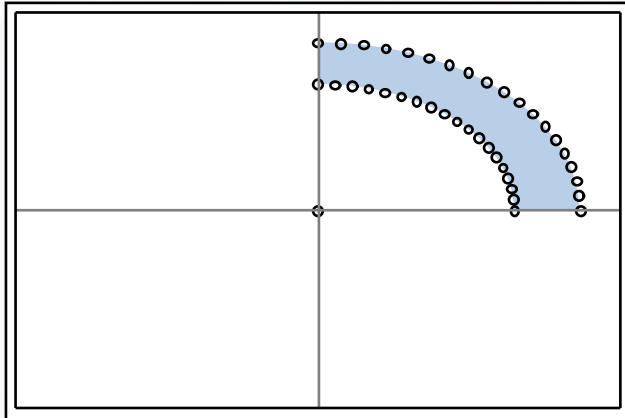
Example:

```
Data arc;  
ID=1;  
D=1; /**Distance of 1**/  
X_O=0; /**X-coordinate of origin**/  
Y_O=0; /**Y-coordinate of origin**/  
/**Output x/y-coordinates at origin**/  
X=0;Y=0;OUTPUT;  
  
/**Run DO loop from start to end of arc**/  
DO t = 0 to 90 by 5;  
    X = X_O + D * COS((t/360)*2*CONSTANT('PI'));  
    Y = Y_O + D * SIN((t/360)*2*CONSTANT('PI'));  
    OUTPUT;  
end;  
run;
```

Sunburst Chart Application

Building a doughnut with the POLYGON plot:

- Take the wedge and add in another arc closer to the origin



Example:

Data arc;

...

```
/**Add inner arc going in reverse**/
```

```
DO t = 90 to 0 by -5;
```

```
  X = X_O + 0.75 * D * COS((t/360)*2*CONSTANT('PI'));
```

```
  Y = Y_O + 0.75 * D * SIN((t/360)*2*CONSTANT('PI'));
```

```
  OUTPUT;
```

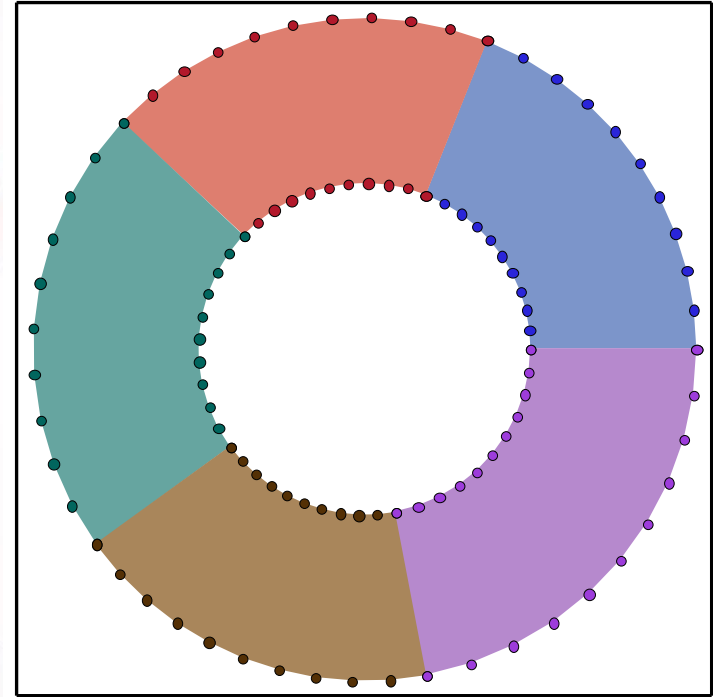
```
end;
```

```
run;
```

Sunburst Chart Application

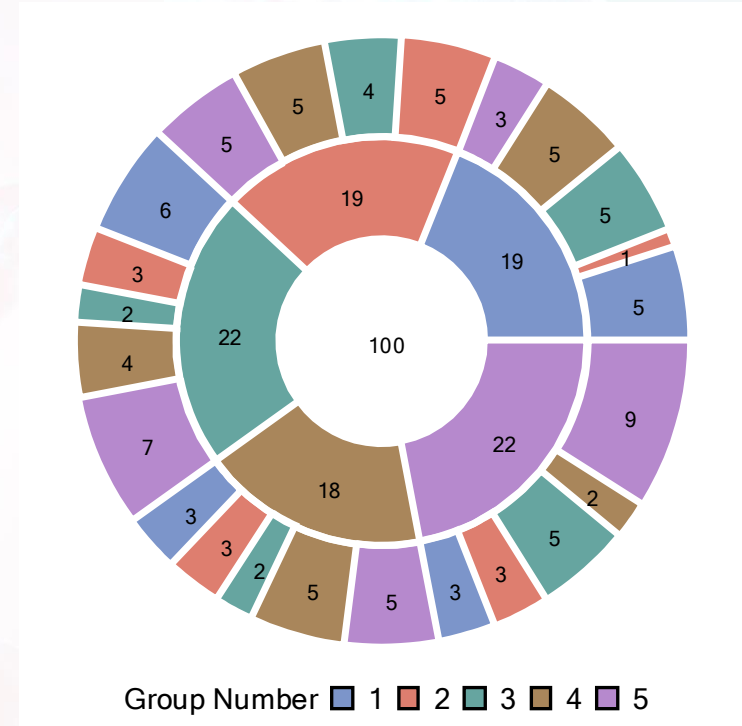
Building a doughnut with the POLYGON plot:

- Repeat for each group around the doughnut
- Proportion of population equals the proportion of the doughnut covered



Sunburst Chart Application

- Each sequential outer ring lines up with the inner ring's population
- Proportion is still based on original population
- Spacing and width of bars set by distance from the origin in trigonometry equations
- Series plot is used to add spacing between rectangles



Conclusion

- The POLYGON plot is a versatile tool to fill in the gaps in the SAS graphics library
- Functions can be applied to create the coordinates for the POLYGON plot with do loops in a DATA step
- Other figures such as pie charts and donut charts can be enhanced with these methods

Contact Information

Name: Jeffrey Meyers

Affiliation: Regeneron Pharmaceuticals

E-mail: Jeffrey.Meyers@regeneron.com

Website: [SAS Communities Page](#)