Subject: Re: cgGallery with function graphics
Posted by Matthew Argall on Sun, 02 Mar 2014 01:39:28 GMT
View Forum Message <> Reply to Message

Filled Area By Height Plot
http://www.idlcoyote.com/gallery/height_filled_area_plot.pro

```
PRO Filled_Area_By_Height_Plot_FG, WINDOW=awindow
   compile_opt strictarr

   ; Set up variables for the plot. Normally, these values would be
   ; passed into the program as positional and keyword parameters.
   x = Findgen(101)
   y = 4 * Sin(x * !DtoR) / Exp( (x-15) / 25.)

   ; Set up the low and high x indices of the area under the curve
   ; you want to fill.
   low = 10
   high = 45

   ; Find the y indices associated with the low and high indices.
   lowY  = 4 * Sin(low * !DtoR) / Exp( (low-15) / 25.)
   highY = 4 * Sin(high * !DtoR) / Exp( (high-15) / 25.)
   indices = Value_Locate(x, [low, high])
   lowIndex = indices[0]
   highIndex = indices[1]

   ; Make sure the indices you find correspond the the right X indices.
   IF x[lowIndex] LT low THEN lowIndex = lowIndex + 1
   IF x[highIndex] GT high THEN highIndex = highIndex - 1

   ; Open a window and return its reference to the user.
   aWindow = Window(WINDOW_TITLE="Filled Area by Height Plot")

   ; Turn refresh off until we are finished adding all of the graphics
   aWindow -> Refresh, /Disable

   ; Draw the plot axes.
   fgPlot = Plot(x, y, /Current, XTitle='X Axis', YTitle='Y Axis', Color='Navy', $
            Name='4*Sin(x) $\slash$ e^(x-15) $\slash$ 25')

   ;APPROACH 1
   ;  - Use a bunch of polygons to fill the area under the curve. Necessary for
   ;      versions of IDL < 8.2.1
   IF (!Version.Release LE 8.2) THEN BEGIN
      ; Scale the y data for colors.
```

```
   cgLoadCT, 4, /Brewer, Clip=[50, 230], RGB_Table=RGB_Table

   ; Draw the area under the curve with scaled colors.
   min_y = Min(y[lowIndex:highIndex], Max=max_y)
   colors = BytScl(y, MIN=min_y, MAX=max_y)

   ; Number of polygons to make
   nPoly = highIndex-lowIndex

   ; Create the polygons
   fgPolygons = objarr(nPoly)
   FOR j=lowIndex,highIndex-1 DO BEGIN
      ;Each little area has to be its own color/polygon. Create the vertices.
      xpoly = [x[j],        x[j], x[j+1],       x[j+1],        x[j]]
      ypoly = [!Y.CRange[0], y[j], y[j+1], !Y.CRange[0], !Y.CRange[0]]

      ;Create the polygon
      fgPolygons[j-lowIndex] = Polygon(xpoly, ypoly, /Data, Target=fgPlot, $
                               LineStyle=6, /Fill_Background, $
                               Fill_Color=reform(RGB_Table[colors[j],*]), $
                               Name='Filled Area' + strtrim(j-lowIndex))
   ENDFOR

;APPROACH 2
;   - Use the Vert_Colors and RGB_Table keywords (introduced in IDL 8.2.1)
;   - Cannot test because I have IDL 8.2
ENDIF ELSE BEGIN
   ; Create closed polygons to color fill.
   yMin = fgPlot.YRange[0]
   xpoly = [ low,  low, x[lowIndex:highIndex],  high, high]
   ypoly = [yMin, lowY, y[lowIndex:highIndex], highY, yMin]

   ;Get the color table
   cgLoadCT, 4, /Brewer, RGB_Table=RGB_Table

   ;Create an array of indices between 50 and 230, scaled by height.
   ;   - The RGB_Table keyword in Polygon takes a full palette [256x3], so I
   ;       presume giving a smaller palette will not work.
   colors = BytScl(y, MIN=min_y, MAX=max_y, Top=256-76) + 50B

   ;Create a filled polygon and keep it in the data space.
   ;   - I assume the Vert_Colors keyword will color the area between each
   ;       vertice the correct color...
   fgPoly = Polygon(xpoly, ypoly, /Data, Target=fgPlot, /Fill_Background, $
                Fill_Color='Dodger Blue', Name='Area Under Plot', $
                RGB_Table=RGB_Table, Vert_Color=colors, LineStyle=6)
ENDELSE
```

```
    ;Add lines at the edges of the filled region
    yrange = fgPlot.YRange
    fgLine1 = PolyLine([low, low], [yrange[0], lowY], /Data, Target=fgPlot, Color='Grey', $
                Name='Low Line')
    fgLine2 = PolyLine([high, high], [yrange[0], highY], /Data, Target=fgPlot, Color='Grey', $
                Name='High Line')

    ; Bring the plot to the front
    fgPlot -> Order, /Bring_To_Front

    ;Refresh the plot
    aWindow -> Refresh

END  ;****************************************************** ******

; This main program shows how to call the program and produce
; various types of output.

  ; Display the plot in a  resizeable graphics window.
  Filled_Area_By_Height_Plot_FG, Window=window

  ; Create a PostScript file. Linestyles are not preserved in IDL 8.2.3 due
  ; to a bug. Only encapsulated PostScript files can be created.
  window.save, 'filled_area_by_height_plot_fg.eps'

  ; Create a PNG file with a width of 600 pixels. Resolution of this
  ; PNG file is not very good.
  window.save, 'filled_area_by_height_plot_fg.png', WIDTH=600

  ; For better resolution PNG files, make the PNG full-size, then resize it
  ; with ImageMagick. Requires ImageMagick to be installed.
  window.save, 'filled_area_by_height_plot_fg_fullsize.png'
  Spawn, 'convert filled_area_by_height_plot_fg_fullsize.png -resize 600
filled_area_by_height_plot_fg_resized.png'

END
```