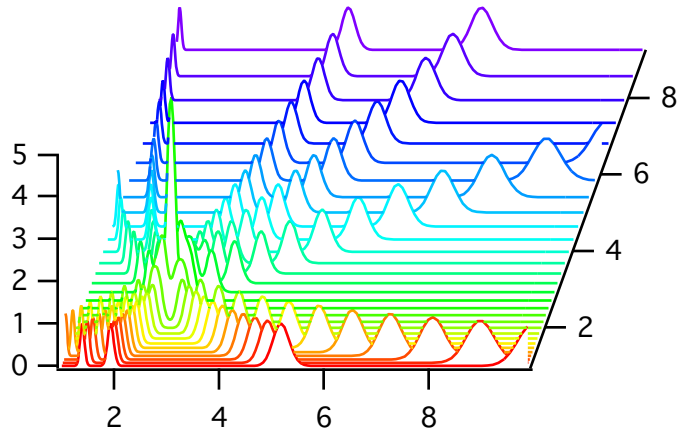
```
Function UnvenlySpacedWaterfallPlot()
   // Create matrix for waterfall plot
   Make/O/N=(200,30) mat2
   SetScale x,-3,4,mat2        // Scaling is needed only to generate
   SetScale y,-2,3,mat2        // the fake data
   mat2=exp(-((x-y)^2+(x+3+y)^2))
   mat2=exp(-60*(x-1*y)^2)+exp(-60*(x-0.5*y)^2)+exp(-60*(x-2*y)^2)
   mat2+=exp(-60*(x+1*y)^2)+exp(-60*(x+2*y)^2)
   SetScale x,0,0,mat2         // Scaling no longer needed because we will
   SetScale y,0,0,mat2         // use X and Y waves in waterfall plot

   // Make X and W waves
   Make/O/N=200 xWave = 10^(p/200)
   Make/O/N=30 yWave = 10^(p/30)

   // Create waterfall plot
   NewWaterfall /W=(21,118,434,510) mat2 vs {xWave,yWave}
   ModifyWaterfall angle=70, axlen= 0.6, hidden= 3

   // Apply color as a function of Z
   Duplicate mat2,mat2ColorIndex
   mat2ColorIndex=y
   ModifyGraph zColor(mat2)={mat2ColorIndex,*,*,Rainbow}
End
```



Fake Waterfall Plots

Creating a real waterfall plot requires a 2D wave. If your data is in the form of 1D waveforms or XY pairs, it may be simpler to create a "fake waterfall plot".

In a fake waterfall plot, you plot your waveform or XY data using a regular graph and then create the waterfall effect by offsetting the traces. Since fake waterfall plots use regular Igor traces, you can control their appearance the same as in a regular graph.

The result, with hidden line removal, looks like this: