```
Make/N=(50,3) run1      // 50 points per dataset, three datasets
Make/N=(50,3) run2      // Another with three datasets with 50 points
run1 = gnoise(q+1)      // Gaussian data with standard deviation
run2 = gnoise(1) + q    // Gaussian data with constant width and location
Make/N=3/T categories="Dataset "+num2str(p+1)    // Text wave for category plot
```

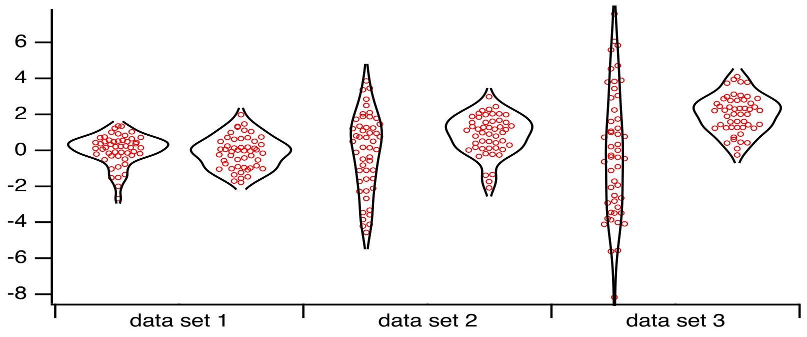
At this point you could select Windows→New Violin Plot, turn on the Multicolumn checkbox, select run1 as the data and categories as the X wave. That would result in these commands:

```
Display;AppendViolinPlot run1 vs categories
```

Now select Graph→Append Violin Plot, turn on the Multicolumn checkbox, select run2 as the data and accept categories as the X wave. That would result in this command:

```
AppendViolinPlot run2
```

The resulting graph looks like this:



Appearance Options for Violin Plots

To change the appearance of a violin plot, select Graph→Modify Violin Plot or right-click and select Modify Violin Plot.

Igor's violin plot has six kernel shapes to choose from. The default is Gaussian and we don't anticipate that other shapes will be used much.

There are three methods for automatically estimating the best bandwidth to use. If you don't like the results of the automatic bandwidths, you can set your own, with a separate bandwidth for each dataset. The automatic estimate assumes a Gaussian kernel.

Each plot of a violin trace occupies a horizontal space equivalent to the box width of a box plot, and that box width can be set in the same way as a box plot—fractions give a width that is a fraction of the plot area, values larger than one are absolute sizes in points. If the X axis is a category axis, Igor sets the box width, overriding whatever you may have chosen.

When Igor makes a violin trace containing more than one dataset, the curves are all normalized to the largest peak amongst all the KDE curves so only one of the violin plots occupies the full box width. You can see this in the plot above, where the third dataset in the Run1 trace (left plots in each category) represents a dataset with broader distribution, which means that to achieve an area of one it must have a smaller amplitude. Compare it with the short, fat distributions in the other categories.

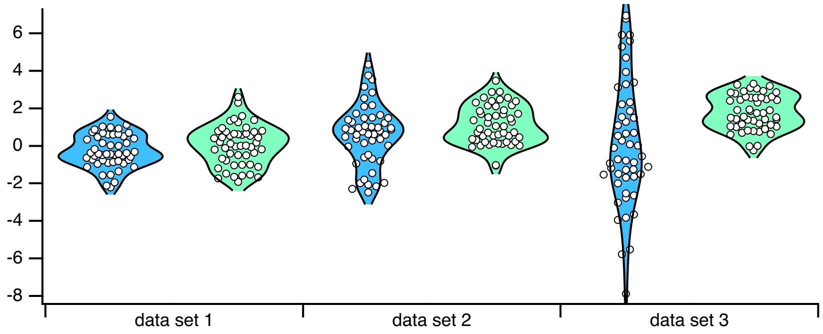
If you have more than one set of violin plots, as in this example where there are two violin traces—one from run1 and one from run2, by default the widest plot in each trace fills the box. The two traces are normalized separately. If you want all the plots in both traces to use the same normalization, you can set the normalization yourself so that the relative amplitudes of all the curves represent the same relative values. This is done using the Distribution Max setting in the General tab of the Modify Violin Plot dialog. The value computed by Igor is shown next to the edit box.

It is common to show the raw data with markers. By default, the raw data points are shown using hollow circles drawn half the size of the default markers for a normal trace. You can choose a marker for the raw data points, plus one for the mean and the median if desired. Violin plots do not distinguish outliers. Jitter can be applied to the data point markers so that they don't overlap.

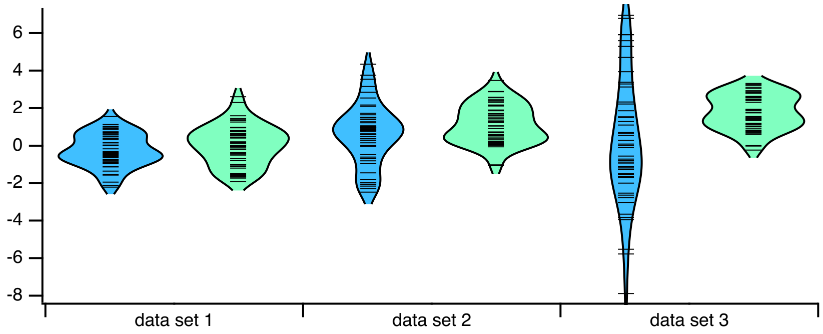
To make a rug plot of the raw data, use the appropriate line marker and set the jitter to 0. Most violin plots are plotted vertically, and the horizontal line marker would be appropriate.

The space between the KDE curves can be filled with color. The fill color is drawn behind the markers so that special effects can be achieved with marker color, marker stroke color and marker fill color.

Here is the plot above, re-worked with fill color and hollow round markers with white fill:



And here it is as a rug plot:



You may notice that some of the curves appear to be a bit truncated. That is particularly true of the dataset 3 plot in the right-hand trace. By default, Igor plots the KDE curves one bandwidth beyond the last data point. In this case, it appears that it might be nice to have the curves extend a bit farther. To achieve that, right-click the trace and select Modify Violin Plot. On the General tab, find the Curve Extension edit box and enter a more pleasing amount of extension. The units are kernel bandwidths. Here we have set 2 for the plot above: