/M	*objectSpec* coordinates are in centimeters.
/R	*objectSpec* coordinates are in percent of printing part of the page.
/S	Stacks objects.
/T	Tiles objects.

See Also

The **Layout** and **AppendLayoutObject** operations for use with user-defined functions.

AppendToTable

```
AppendToTable [/W=winName] columnSpec [, columnSpec]…
```

The AppendToTable operation appends the specified columns to the top table. *columnSpec*s are the same as for the **Edit** operation; usually they are just the names of waves.

Flags

/W=*winName*	Appends columns to the named table window or subwindow. When omitted, action will affect the active window or subwindow. When identifying a subwindow with *winName*, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

See Also

Edit for details about *columnSpec*s, and **RemoveFromTable**.

AppendViolinPlot

```
AppendViolinPlot [ flags ] wave[, wave, ...] [vs xWave]
```

The AppendViolinPlot operation appends a violin plot trace to the target or named graph. A violin plot (also called a “bean” plot) is a way to display a summary of the distribution of data values using a kernel density estimator curve (see **StatsKDE**). Another way to display a summary of data distribution is via a box plot.

AppendViolinPlot was added in Igor Pro 8.00.

A violin plot trace is treated within Igor as a single graph trace, and many of the operations such as removing from a graph or reordering traces work the same with a violin plot trace as with any other graph trace.

There is no DisplayViolinPlot operation. Use Display followed by AppendViolinPlot.

Parameters

The data for a single violin plot in a violin plot trace comes from either one entire 1D wave or from a single column of a multi-column wave. The number of violin plots in the trace is determined by either the number of 1D waves in the list of waves, or by the number of columns in the single multi-column wave. It is not permitted to mix 1D and multi-column waves.

You can list up to 100 individual 1D waves. If you want more violin plots than 100 in a violin plot trace you must use a multi-column wave or add to the list using the **AddWavesToViolinPlot** operation.

If you do not provide *xWave*, each violin plot is positioned on a numeric axis at X=0, 1, etc. If the data is in a multi-column wave, positioning comes by default from the X scaling of the matrix wave. Providing a numeric *xWave* allows you to position each violin plot at an arbitrary position on the X axis. A text *xWave* displays the violin plots using a category axis. If you use the /CATL flag with a multi-column wave, the result is a category X axis using the dimension labels as the category labels. See **Category Plots** on page II-355 and **Dimension Labels** on page II-93.

xWave must have at least as many points as you have 1D Y waves or columns in a multi-column wave. Because a list of waves may become too long for the command line, you can use an *xWave* that is longer than the list. Use **AddWavesToViolinPlot** to complete the list.

Flags

/L/R/B/T
These axis flags are the same as used by **AppendToGraph**.

/CATL[=*doCatLabels*]
Use the column dimension labels to produce violin plots on a category axis using the column dimension labels as the category labels. *doCatLabels* is 0 or 1 and /CATL is equivalent to /CATL=1.

/TN=*traceName*
Allows you to provide a custom name for a trace. This is useful when displaying waves with the same name but from different data folders. See **User-defined Trace Names** on page IV-89 for details, except that the /TN flag for AppendViolinPlot comes in the normal position in the command, after the command name.

/VERT[=*doVert*]
Arranges the individual violin plots vertically along the Y axis. *doVert* is 0 or 1 and /VERT is equivalent to /VERT=1.

/VERT is similar to ModifyGraph SwapXY but on a trace-by-trace basis. To make a violin plot with horizontal violins, use either ModifyGraph SwapXY or AppendViolinPlot/VERT.

/W=*winName*
Appends to the named graph window or subwindow. When omitted, AppendViolinPlot is directed to the active window or subwindow. This must be the first flag specified when used in a Proc or Macro or on the command line.

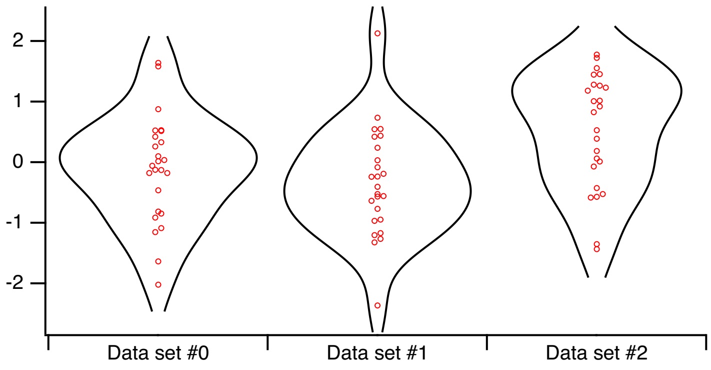
When identifying a subwindow with winName, see **Subwindow Syntax** on page III-92 for details on forming the window hierarchy.

Details

Some aspects of the overall appearance of a violin plot trace can be set using ModifyGraph. By default, the line color is black and markers for non-outlier data points are hollow circles half the size of the normal trace marker. Using the Modify Violin Plot dialog or the ModifyViolinPlot operation, you can choose to use trace color, line size, line dash style, marker and marker size as set by ModifyGraph just as for a regular trace. Detailed control of these characteristics for various parts of a violin plot trace is provided by **ModifyViolinPlot**.

Examples

```
// Demo vertical violin plot
Make/O/N=(25,3) multicol                // A three-column wave with 25 rows
SetRandomSeed(.5)
multicol = gnoise(1)                    // Three normally-distributed datasets
Make/O/N=3/T labels                     // A text wave to make a category plot
labels = "Dataset #"+num2str(p)         // Labels for the X axis of a category plot
Display; AppendViolinPlot multicol vs labels
```



```
// Demo horizontal violin plot
Make/O/N=50 ds1, ds2, ds3, ds4
Make/O/N=4 dsX
ds1 = gnoise(1)
ds2 = gnoise(2)
ds3 = logNormalNoise(0, 1)
```