style=*n*

n is a bitwise parameter with each bit controlling one aspect of the column's font style as follows:

Bit 0:	Bold
Bit 1:	Italic
Bit 2:	Underline
Bit 4:	Strikethrough

For example, bold underlined is $2^0 + 2^2 = 1 + 4 = 5$. See **Setting Bit Parameters** on page IV-12 for details about bit settings.

title="*title*"

Sets the title of a column to *title*.

topLeftCell=(*row*, *column*)

Scrolls the table contents so that the cell identified by (*row*, *column*) is the top left visible data cell, or as close as possible.

If *row* is -1 then the table's vertical scrolling is not changed. If *column* is -1 then the table's horizontal scrolling is not changed.

If they are positive, *row* and *column* are zero-based numbers which are clipped to valid values before being used. *row*=0 refers to the first row of data in the table, *column*=0 refers to the first column of data.

The Point column can not be scrolled horizontally.

trailingZeros=*t*

t=1 shows trailing zeros. This affects the general numeric format only.

viewSelection

Scrolls the table contents so that the target cell and selection are maximally in view. The target cell will always be visible. The selection may overflow the visible area.

See also the topLeftCell and selection keywords.

viewSelection was added in Igor Pro 9.00.

width=*w*

Sets column width to *w* points.

You will not always get the exact number of points that you request. This is because a column must have an even number of screen pixels, so that grid lines look good. Igor will modify your requested number of points to meet this requirement.

Flags

/W= *winName*

Modifies 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.

/Z

No errors generated if the indexed or specified column does not exist in a style macro.

Examples

```
ModifyTable size(myWave)=14       // change font size of myWave column
ModifyTable width(Point)=0                  // hide Point column
ModifyTable style(cmplxWave.imag)=32        // condensed= bit 5 = 2^5 = 32
```

See Also

See **Column Names** on page II-241 and **ModifyTable Elements Command** on page II-263.

ModifyViolinPlot

ModifyViolinPlot [/W=*winName*] [keyword=*value*, keyword=*value*, ...]

The ModifyViolinPlot operation modifies a violin plot trace in the target or named graph.

ModifyViolinPlot was added in Igor Pro 8.00.

For a detailed discussion of violin plots and the parts of a violin plot, see **Violin Plots** on page II-337.

Parameters

ModifyViolinPlot parameters consist of keyword=value pairs. The trace keyword specifies the trace targeted by the subsequent keywords. For example, the command:

```
ModifyViolinPlot trace=trace0, markerSize=5
```

sets the marker size for all datasets of the trace0 trace.

As of Igor Pro 9.00, you can modify a setting for a specific dataset of a specific trace by adding a zero-based dataset index in square brackets after the keyword. For example:

```
ModifyViolinPlot trace=trace0, markerSize[1]=7
```

This sets the marker size for the second dataset (index=1) to 7 leaving the marker size for other datasets unchanged.

General Parameters

trace=*traceName*

Specifies the name of a violin plot trace to be modified. An error results if the named trace is not a violin plot trace. Without the trace keyword, ModifyViolinPlot uses the first trace in the graph, whether it is a violin plot trace or not but see the instance keyword for an exception.

instance=*instanceNum*

The combination of trace and instance works the same as (*traceName#instanceNum*) for a ModifyGraph keyword.

The instance keyword without trace keyword accesses the *instanceNum*'th trace in the graph, just like [*traceNum*] used with a ModifyGraph keyword. See **Trace Names** on page II-282 and **Object Indexing** on page IV-20.

bandwidth[(*ds*)]=*bw*

Sets the bandwidth used in computing the KDE curves. If you include *ds*, the bandwidth is set to *bw* for dataset *ds*. If you omit *ds*, the bandwidth is set for all datasets. *bw* must be greater than or equal to 0. 0 uses the bandwidth estimated by the method set by the bandwidthMethod keyword.

bandwidthMethod=*m*

Sets the method used to estimate the appropriate bandwidth used in computing the KDE curves. The methods are

m=0:	Silverman
m=1:	Scott (default)
m=2:	Bowman and Azzalini

See StatsKDE for details.

You can override the estimated bandwidth using the bandwidth keyword.

boxWidth=*w*

For a non-category X axis, boxWidth sets the width of the space reserved for displaying the KDE curves. If *w* is between zero and one, it is taken to be a fraction of the width of the plot rectangle. If *w* is greater than one, it is in points.

All the KDE curves are normalized such that the maximum density value of the KDEs over all datasets fills the box width so only one of the violin plots will be as wide as *w*. You can control the normalization using the maxDensity keyword.

If the violin plot is displayed using a category X axis, boxWidth is ignored. The space reserved for each dataset is controlled by the category axis and is affected by ModifyGraph barGap and catGap.

A new non-category violin plot has box width set as boxWidth=1/(2**n*) where *n* is the number of datasets (the number of violin plots) on the trace.

closeOutline[=*close*]

If *close* is 1 or omitted, the ends of the KDE curves are connected by a straight line. Closed curves are more visible if side=1 or side=2 in which case the KDE curves are closed by a straight line at the midline between the curves.

curveExtension=*ext*

Sets the range of the KDE curves. The KDE curves are computed for a range of y=minData-*ext* to y=maxData+*ext*, where *ext* is in units of bandwidth. *ext* defaults to 1.

dataMarker=marker

Sets the marker number used to display the raw data. If marker is -1, the marker as set by ModifyGraph marker is used. The default is 8 (hollow circle).

jitter=*j*

In order to separate closely-spaced raw data points, the data points can be displaced horizontally. *j* controls the maximum offset applied to any data point, expressed as a fraction of the box width. The value of *j* may be greater than 1, but in general values less than 1 look better. The default value is 0.5.

kernel=*k*

Sets the kernel function to be used in computing the KDE curves. Supported values are

k=1:	Epanechnikov
k=2:	Bi-weight
k=3:	Tri-weight
k=4:	Triangular
k=5:	Gaussian (default)
k=6:	Rectangular

It is unusual for a violin plot to use anything other than Gaussian, which is the default. See StatsKDE for details.

lineStyle=*style*

Sets the line style used to draw the KDE curves. See **Dashed Lines** on page III-496 for a description of line styles.

A line style of -1 uses the line style set by ModifyGraph lstyle.

style defaults to -1.

lineThickness=*thick*

Sets the thickness of the line used to plot the KDE curves. A thickness less than zero uses the line width set by the ModifyGraph lsize keyword. *thick*=0 hides the KDE curve. The default value is -1.

markerSize=*size*

Sets the size of the marker used to display the raw data. A marker size of zero uses one-half of the marker size as set by ModifyGraph msize. *size* defaults to zero.

markerThick=*t*

Sets the thickness in points of the strokes of markers used to display the raw data.

The markerThick keyword was added in Igor Pro 9.00.

maxDensity=*d*

Sets the normalization for the width of the KDE curves to *d*. All the KDE curves are normalized such that the maximum density value of all KDEs fills the box width. The width of each curve gives an indication of the relative maximum density. maxDensity allows you to set the same normalization for multiple violin plots so that the width for all is relative to the same value.

plotSide=*side*

By default, a violin plot is symmetrical with the KDE curve plotted twice in mirror image. You can control which side is plotted:

side=0:	Both sides in mirror image (default)
side=1:	The left side
side=2:	The right side

Using 1 or 2 allows you to combine two violin plots in an asymmetric plot.

showData[=*sd*]

Shows or hides the raw data points. *sd* is 0 or 1. showData by itself is equivalent to showData=1. The default is 1.

Mean and Median Parameters

You can display markers showing the mean and the median of each dataset using the keywords documented in this section. See **Markers** on page II-291 for a list of available markers. For both mean and median, a marker size of zero uses the marker size set by ModifyGraph msize.

showMean[=*show*]
Shows or hides the marker representing the mean value. *show* is 0 or 1. showMean by itself is equivalent to showMean=1. The default is 0.

meanMarker=*marker*
Sets the marker to use for the mean value. The default for the mean marker is 27 (hollow horizontal diamond with dot).

meanMarkerSize=*size*
Sets the size of the marker used to display the mean value. A value of zero uses the trace marker size as set by ModifyGraph msize, or the normal trace default marker size. The default is zero.

meanMarkerThick=*t*
Sets the thickness in points of the stroke of markers used to display the mean value if showMean is enabled.

The meanMarkerThick keyword was added in Igor Pro 9.00.

showMedian[=*show*]
Shows or hides the marker representing the median value. *show* is 0 or 1. showMedian by itself is equivalent to showMedian=1. The default is 0.

medianMarker=*marker*
Sets the marker to use for the median value. The default for the median marker is 26 (filled horizontal diamond).

medianMarkerSize=*size*
Sets the size of the marker used to display the median value. A value of zero uses the trace marker size as set by ModifyGraph msize, or the normal trace default marker size. The default is zero.

medianMarkerThick=*t*
Sets the thickness in points of the stroke of markers used to display the median value if showMedian is enabled.

The medianMarkerThick keyword was added in Igor Pro 9.00.

Color Parameters

All colors are specified as (*r*,*g*,*b*[,*a*]) **RGBA Values**.

fillColor=(*r*,*g*,*b*[,*a*])
Fills the area between the distribution curves with the specified color. Specifying fillColor=(0,0,0,0) removes the fill. If side=1 or side=2, the fill is between the curve and the mid line.

lineColor=(*r*,*g*,*b*[,*a*])
Sets the color of the KDE curves.

Specify lineColor=(0,0,0,0) to use the trace color set using ModifyGraph rgb as the line color. The default is black.

markerColor=(*r*,*g*,*b*[,*a*])
Sets the color of markers used to display the raw data.

Specify markerColor=(0,0,0,0) to use the trace color set using ModifyGraph rgb as the marker color. This is the default behavior.

markerStrokeColor=(*r*,*g*,*b*[,*a*])
Sets the color of the strokes of markers used to display the raw data.

Specify markerStrokeColor=(0,0,0,0) to use the to use the trace color set using ModifyGraph rgb as the marker color. This is the default behavior.

The markerStrokeColor keyword was added in Igor Pro 9.00.

markerFilled=*f*
If *f* is non-zero, markers are filled with color. Normally, hollow markers have transparent fill. Default fill color is white. Applies only to hollow markers such as marker 8, the hollow circle marker.

The markerFilled keyword was added in Igor Pro 9.00.

markerFillColor=(*r*,*g*,*b*[,*a*])
Sets the fill color of markers used to display the raw data. Applies only to hollow markers such as marker 8, the hollow circle marker.

The markerFillColor keyword was added in Igor Pro 9.00.

meanMarkerColor=(*r*,*g*,*b*[,*a*])

Sets the color of markers used to display the mean value.

Specify meanMarkerColor=(0,0,0,0) to use the trace color set using ModifyGraph rgb as the mean marker color. This is the default behavior.

meanMarkerStrokeColor=(*r*,*g*,*b*[,*a*])

Sets the color of the strokes of markers used to display the mean of the data if showMean is enabled.

Specify meanMarkerStrokeColor=(0,0,0,0) to use the to use the trace color set using ModifyGraph rgb as the marker color. This is the default behavior.

The meanMarkerStrokeColor keyword was added in Igor Pro 9.00.

meanMarkerFilled=*f*

If *f* is non-zero, the mean value marker is filled with color. Normally, hollow markers have transparent fill. Default fill color is white. Applies only to hollow markers such as marker 8, the hollow circle marker.

The meanMarkerFilled keyword was added in Igor Pro 9.00.

meanMarkerFillColor=(*r*,*g*,*b*[,*a*])

Sets the fill color of markers used to display the mean value if showMean is enabled. Applies only to hollow markers such as marker 8, the hollow circle marker.

The meanMarkerFillColor keyword was added in Igor Pro 9.00.

medianMarkerColor=(*r*,*g*,*b*[,*a*])

Sets the color of markers used to display the median value.

Specify medianMarkerColor=(0,0,0,0) to use the trace color set using ModifyGraph rgb as the median marker color. This is the default behavior.

medianMarkerStrokeColor=(*r*,*g*,*b*[,*a*])

Sets the color of the strokes of markers used to display the mean of the data if showMedian is enabled.

Specify medianMarkerStrokeColor=(0,0,0,0) to use the to use the trace color set using ModifyGraph rgb as the marker color. This is the default behavior.

The medianMarkerStrokeColor keyword was added in Igor Pro 9.00.

medianMarkerFilled=*f*

If *f* is non-zero, the median value marker is filled with color. Normally, hollow markers have transparent fill. Default fill color is white. Applies only to hollow markers such as marker 8, the hollow circle marker.

The medianMarkerFilled keyword was added in Igor Pro 9.00.

medianMarkerFillColor=(*r*,*g*,*b*[,*a*])

Sets the fill color of markers used to display the median value if showMedian is enabled. Applies only to hollow markers such as marker 8, the hollow circle marker.

The medianMarkerFillColor keyword was added in Igor Pro 9.00.

Violin Plot Per-Data-Point Marker Settings

You can override the basic settings for data point marker color, marker style and marker size using marker settings waves containing per-data-point settings.

You can apply per-data-point marker settings to an entire trace (to all datasets comprising a trace) or to a specific dataset of a trace. For example:

```
Function DemoViolinPlotPerPointMarkerSettings()
   Make/O violin0={1,2,3,4,5}, violin1={2,3,4,5,6}, violin2={3,4,5,6,7}
   String title = "Violin Plot Per Point Marker Settings"
   Display/W=(557,99,948,310)/N=ViolinPlotPerPointPlot as title

   // Create a trace named trace0 with three datasets: violin0, violin1, violin2
   AppendViolinPlot/TN=trace0 violin0,violin1,violin2
   // Set the marker and marker size for all datasets of trace trace0
   ModifyViolinPlot trace=trace0, dataMarker=18       // 18=Diamond
   ModifyViolinPlot trace=trace0, markerSize=5

   // Set the per-data-point marker for all datasets of trace trace0
   Make/O violinMarkers = {15,16,17,18,19}
   ModifyViolinPlot trace=trace0, dataMarkerWave=violinMarkers

   // Set the per-data-point marker for dataset violin1 only
   Make/O violinMarkersForViolin1 = {32,33,34,35,36}
   ModifyViolinPlot trace=trace0, dataMarkerWave[1]=violinMarkersForViolin1
End
```

Usually a marker settings wave will have the same number of rows as there are data points in a given dataset, but that is not required. If there are fewer rows in the settings wave than in the dataset, the extra data points retain their basic settings. If there are more rows in the settings wave than in the dataset, the extra settings wave points are not used.

See **Making Each Data Point Look Different** on page II-343 for more information and examples.

dataColorWave [=*colorWave*]

Sets *colorWave* to override data point marker color on a point-by-point basis. The wave must be a 3 or 4 column wave containing red, green, blue and optionally alpha values.

If you omit "=*colorWave*", any previous marker color wave setting is cleared.

The dataColorWave keyword was added in Igor Pro 9.00.

dataMarkerWave [=*markerWave*]

Sets *markerWave* to override data point markers on a point-by-point basis. The values in *markerWave* are standard graph marker numbers. See **Markers** on page II-291 for a table of the markers and the associated marker numbers. The marker numbers are clipped to a valid range.

If you omit "=*markerWave*", any previous marker wave setting is cleared.

The dataMarkerWave keyword was added in Igor Pro 9.00.

dataSizeWave [=*markerSizeWave*]

Sets *markerSizeWave* to override data point marker size on a point-by-point basis. The values in *markerSizeWave* are clipped to the range [0,200].

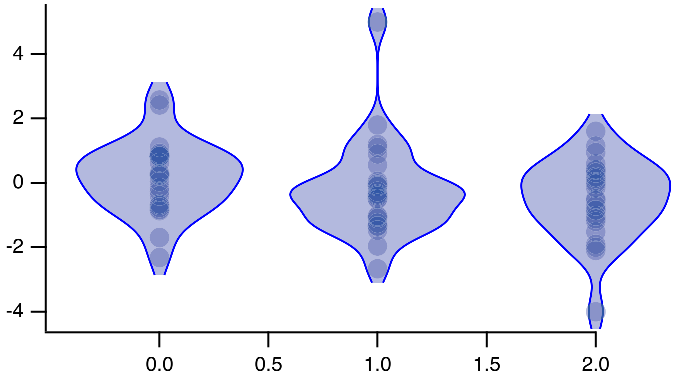
If you omit "=*markerSizeWave*", any previous marker size wave setting is cleared.

The dataSizeWave keyword was added in Igor Pro 9.00.

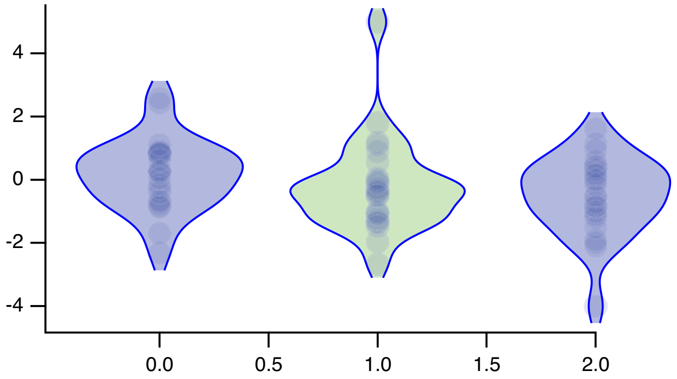
Examples

A violin plot with transparent marker color used to give a visual indication of the density of data points.

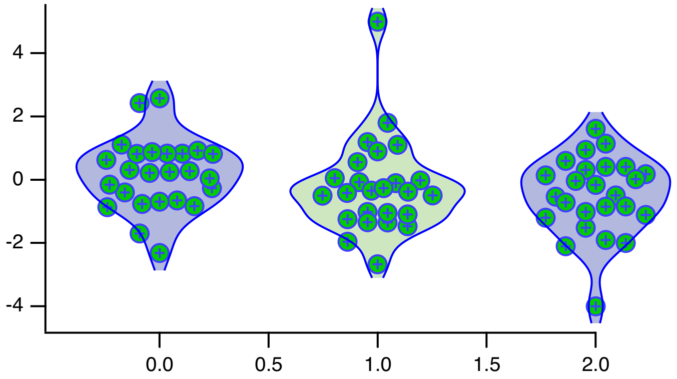
```
// Create a violin plot with transparent marker color used to give a visual
// indication of the density of data points
Make/O/N=(25,3) multicol                       // A three-column wave with 25 rows
SetRandomSeed(.4)                              // For reproducible "randomness"
multicol = gnoise(1)                           // Three normally-distributed datasets
multicol[20][1] = 5                            // A "far" outlier
multicol[13][2] = -4                           // An outlier
Display; AppendViolinPlot multicol
ModifyViolinPlot lineColor=(0,0,65535)
ModifyViolinPlot fillColor=(0,0,65535,19661)
ModifyViolinPlot jitter=0
ModifyViolinPlot showData=1
ModifyViolinPlot dataMarker=19
ModifyViolinPlot markerSize=6
ModifyViolinPlot markerColor=(0,0,65535,6554)
```



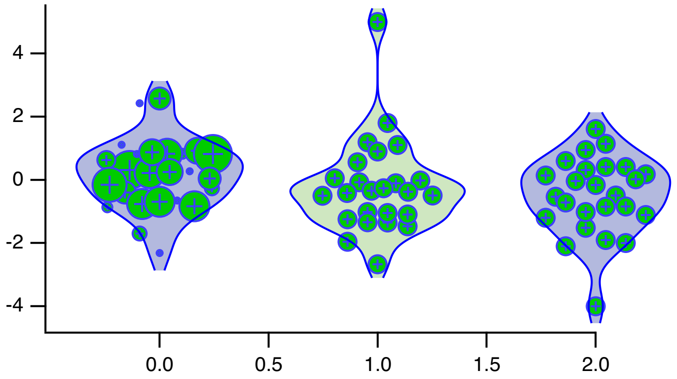
```
ModifyViolinPlot fillColor[1]=(0,65535,0,19661)    // Second dataset transparent green
```



```
// Add jitter, choose the circle-with-plus marker
ModifyViolinPlot trace=multicol,Jitter=0.7
ModifyViolinPlot trace=multicol,DataMarker=42,MarkerSize=5
// Set marker color to blue with a one-point outline
ModifyViolinPlot trace=multicol,MarkerColor=(16385,16388,65535)
ModifyViolinPlot trace=multicol,MarkerThick=1
// Set the marker to filled with dark green fill color
ModifyViolinPlot trace=multicol,MarkerFilled=1,MarkerFillColor=(3,52428,1)
```



```
// Set a marker size wave for dataset 0 (the first dataset)
Make/N=25/O sizeWave = enoise(5)+7
ModifyViolinPlot DataSizeWave[0] = sizeWave
```



```
// Create a violin plot with asymmetric curves plotted on a category axis
Make/O/N=(25,3) ds1, ds2
SetRandomSeed(.4)                                 // For reproducible "randomness"
ds1 = gnoise(1)
ds2 = gnoise(2)+q

// We need a text wave to make a category plot
Make/N=3/T/O labels={"treatment 1", "treatment 2", "treatment 3"}
Display
AppendViolinPlot ds1 vs labels
AppendViolinPlot ds2 vs labels

// Keep plots together in a single category space
ModifyGraph toMode(ds1)=-1

// Display ds1 on the left, ds2 on the right
ModifyViolinPlot trace=ds1,plotSide=1
ModifyViolinPlot trace=ds2,plotSide=2

// Extend the KDE curves
ModifyViolinPlot trace=ds1,CurveExtension=2
ModifyViolinPlot trace=ds2,CurveExtension=2

// Close the curves with line at the midline
ModifyViolinPlot trace=ds1,closeOutline=1
ModifyViolinPlot trace=ds2,closeOutline=1

// Use the same normalization for both curves
ModifyViolinPlot trace=ds1,maxDensity=0.36
ModifyViolinPlot trace=ds2,maxDensity=0.36

// Apply jitter to the data points
ModifyViolinPlot trace=ds1,jitter=0.5
ModifyViolinPlot trace=ds2,jitter=0.5

// Set markers and colors
ModifyViolinPlot trace=ds1,showData=1,dataMarker=16,markerColor=(2,39321,1),lineColor=(2,39321,1)
ModifyViolinPlot trace=ds2,showData=1,dataMarker=19,markerColor=(0,0,65535),lineColor=(0,0,65535)
ModifyViolinPlot trace=ds1,fillColor=(2,39321,1,19661)
ModifyViolinPlot trace=ds2,fillColor=(0,0,65535,19661)
```