# Histogram

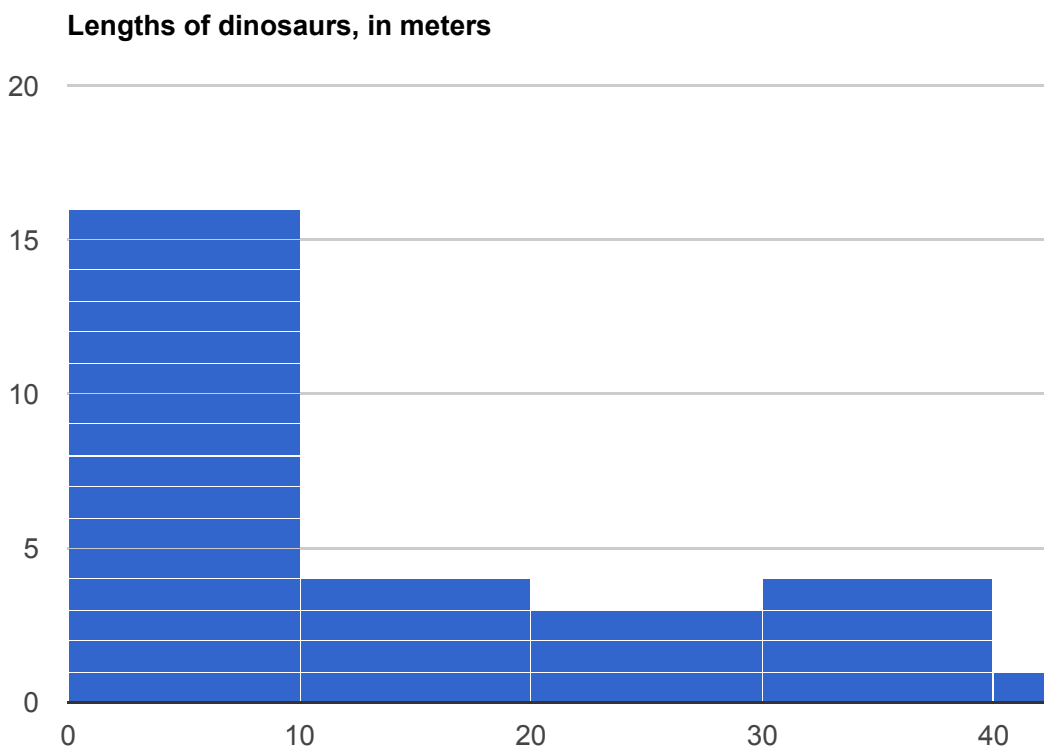
## Overview

A *histogram* is a chart that groups numeric data into bins, displaying the bins as segmented columns. They're used to depict the distribution of a dataset: how often values fall into ranges.

Google Charts automatically chooses the number of bins for you. All bins are equal width and have a height proportional to the number of data points in the bin. In other respects, histograms are similar to column charts (https://developers.google.com/chart/interactive/docs/gallery/columnchart).

## Example

Here's a histogram of dinosaur lengths:



The histogram tells us that the most common bin is < 10 meters, and that there's only one dinosaur over 40 meters. We can hover over the bar to discover that it's the Seismosaurus (which might be just a very big Diplodocus; paleontologists aren't sure (http://blogs.smithsonianmag.com/dinosaur/2010/08/whatever-happened-to-seismosaurus/)).

The code to generate this histogram is shown below. After defining the data (here, with `google.visualization.arrayToDataTable`), the chart is defined with a call to `google.visualization.Histogram` and drawn with the `draw` method.

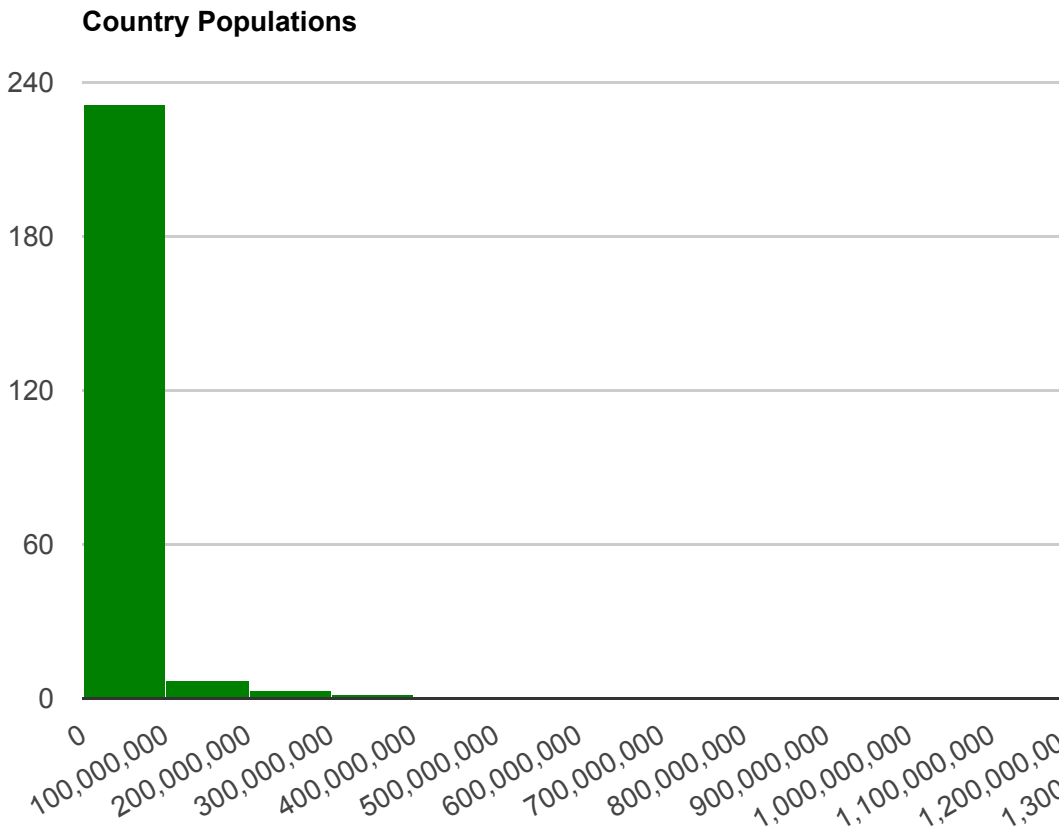
```
<html>
  <head>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader
    <script type="text/javascript">
      google.charts.load("current", {packages:["corechart"]});
      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['Dinosaur', 'Length'],
          ['Acrocanthosaurus (top-spined lizard)', 12.2],
          ['Albertosaurus (Alberta lizard)', 9.1],
          ['Allosaurus (other lizard)', 12.2],
          ['Apatosaurus (deceptive lizard)', 22.9],
          ['Archaeopteryx (ancient wing)', 0.9],
          ['Argentinosaurus (Argentina lizard)', 36.6],
          ['Baryonyx (heavy claws)', 9.1],
          ['Brachiosaurus (arm lizard)', 30.5],
          ['Ceratosaurus (horned lizard)', 6.1],
          ['Coelophysis (hollow form)', 2.7],
          ['Compsognathus (elegant jaw)', 0.9],
          ['Deinonychus (terrible claw)', 2.7],
          ['Diplodocus (double beam)', 27.1],
          ['Dromicelomimus (emu mimic)', 3.4],
          ['Gallimimus (fowl mimic)', 5.5],
          ['Mamenchisaurus (Mamenchi lizard)', 21.0],
          ['Megalosaurus (big lizard)', 7.9],
          ['Microvenator (small hunter)', 1.2],
          ['Ornithomimus (bird mimic)', 4.6],
          ['Oviraptor (egg robber)', 1.5],
          ['Plateosaurus (flat lizard)', 7.9],
          ['Sauronithoides (narrow-clawed lizard)', 2.0],
          ['Seismosaurus (tremor lizard)', 45.7],
          ['Spinosaurus (spiny lizard)', 12.2],
          ['Supersaurus (super lizard)', 30.5],
          ['Tyrannosaurus (tyrant lizard)', 15.2],
          ['Ultrasaurus (ultra lizard)', 30.5],
          ['Velociraptor (swift robber)', 1.8]]);
```

```
        var options = {
          title: 'Lengths of dinosaurs, in meters',
          legend: { position: 'none' },
        };

        var chart = new google.visualization.Histogram(document.getElementByI
        chart.draw(data, options);
      }
    </script>
  </head>
  <body>
    <div id="chart_div" style="width: 900px; height: 500px;"></div>
  </body>
</html>
```

The labels (here, the dinosaur names) can be omitted, in which case the tooltips will show only the numeric value.

## Controlling Colors

Here's a histogram of national populations:




There are over two hundred countries with populations less than a hundred million, and a severe tailing off after that.

This histogram uses the `colors` option to draw the data in green:

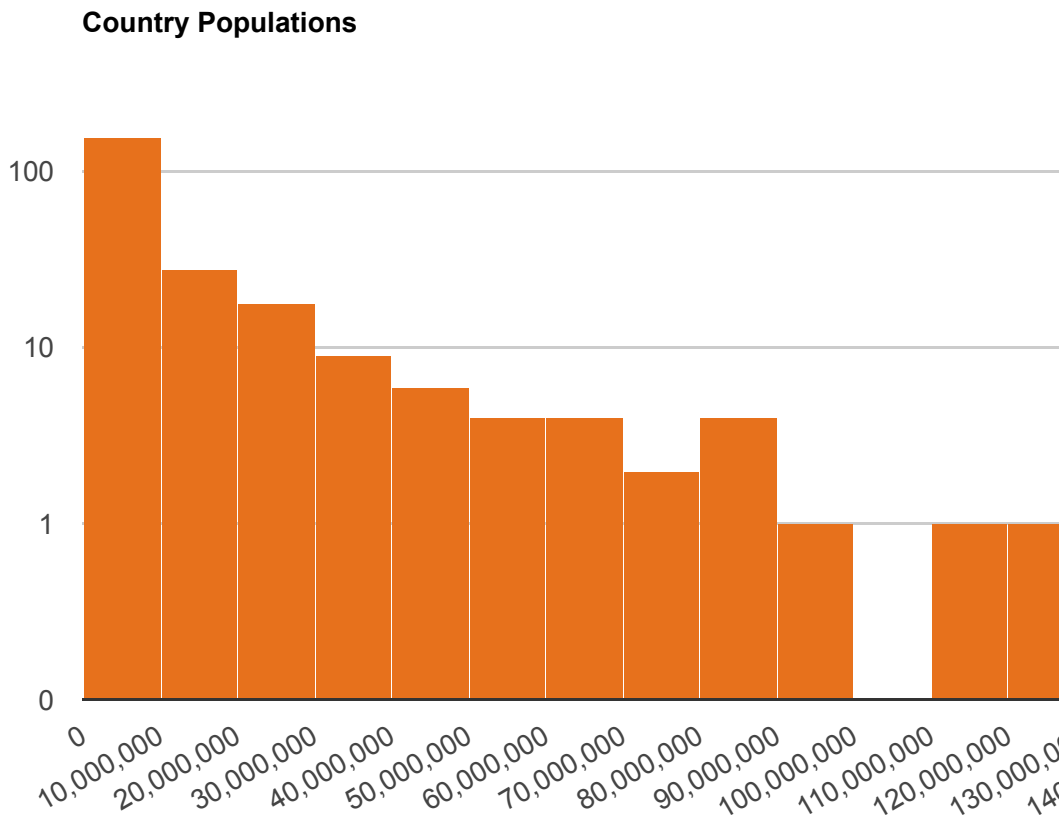
```
var options = {
  title: 'Country Populations',
  legend: { position: 'none' },
  colors: ['green'],
};
```

As with all Google Charts, colors can be specified either as English names or as hex values.

## Controlling Buckets

By default, Google Charts will choose the bucket size automatically, using a well-known algorithm for histograms. However, sometimes you'll want to override that, and the chart above is an example. With so many countries in the first bucket, it's hard to examine those in others.

For situations like this, the Histogram chart provides two options: `histogram.bucketSize`, which overrides the algorithm and hardcodes the bucket size; and `histogram.lastBucketPercentile`. The second option needs more explanation: it changes the computation of bucket sizes to ignore the values that are higher or lower than the remaining values by the percentage you specify. The values are still included in the histogram, but do not affect how they're bucketed. This is useful when you don't want outliers to land in their own buckets; they will be grouped with the first or last buckets instead.




In the above chart, we ignored the top five and bottom five percent of values when calculating bucket size. The values are still charted; the only thing that's changed is the bucket size, but it makes for a more readable histogram.

This example also shows how we can change the scale of the vertical axis to use "mirror log" scale, which also helps when charting data that has a long tail of small values.

```
var options = {
  title: 'Country Populations',
  legend: { position: 'none' },
  colors: ['#e7711c'],
```

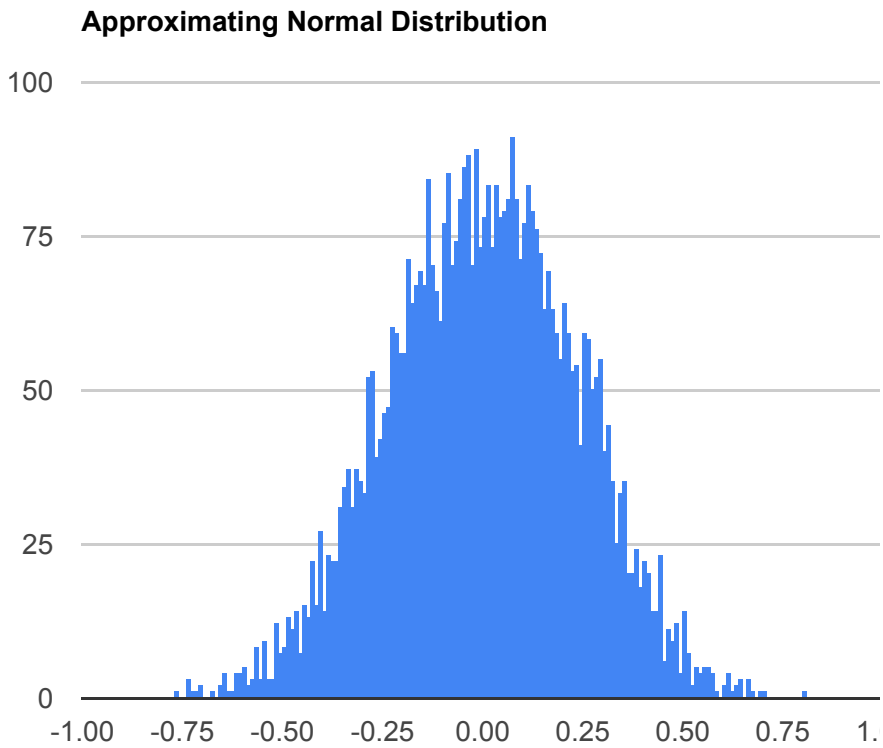
```
    histogram: { lastBucketPercentile: 5 },
    vAxis: { scaleType: 'mirrorLog' }
  };
```

As you can see, removing the top and bottom five percent from the calculation led to a bucket size of 10,000,000 rather than the 100,000,000 it would have been otherwise. If you knew all along that a bucket size of 10,000,000 was what you wanted, you could have used **`histogram.bucketSize`** to do that:

```
  var options = {
    title: 'Country Populations',
    legend: { position: 'none' },
    colors: ['#e7711c'],
    histogram: { bucketSize: 10000000 }
  };
```

In the following example, we show how to expand the range of the buckets and display many more buckets with no gap between them. The **`maxNumBuckets`** option can be used to increase the default number of buckets. The **`histogram.minValue`** and **`histogram.maxValue`** options will expand the range of the buckets, but note that if there is data outside this range, these options will not shrink the range.

This example also shows that you can specify the ticks to display for each of the buckets using the explicit **`ticks`** option for the **`hAxis`**. This does not affect the buckets themselves, but only how the ticks are displayed.

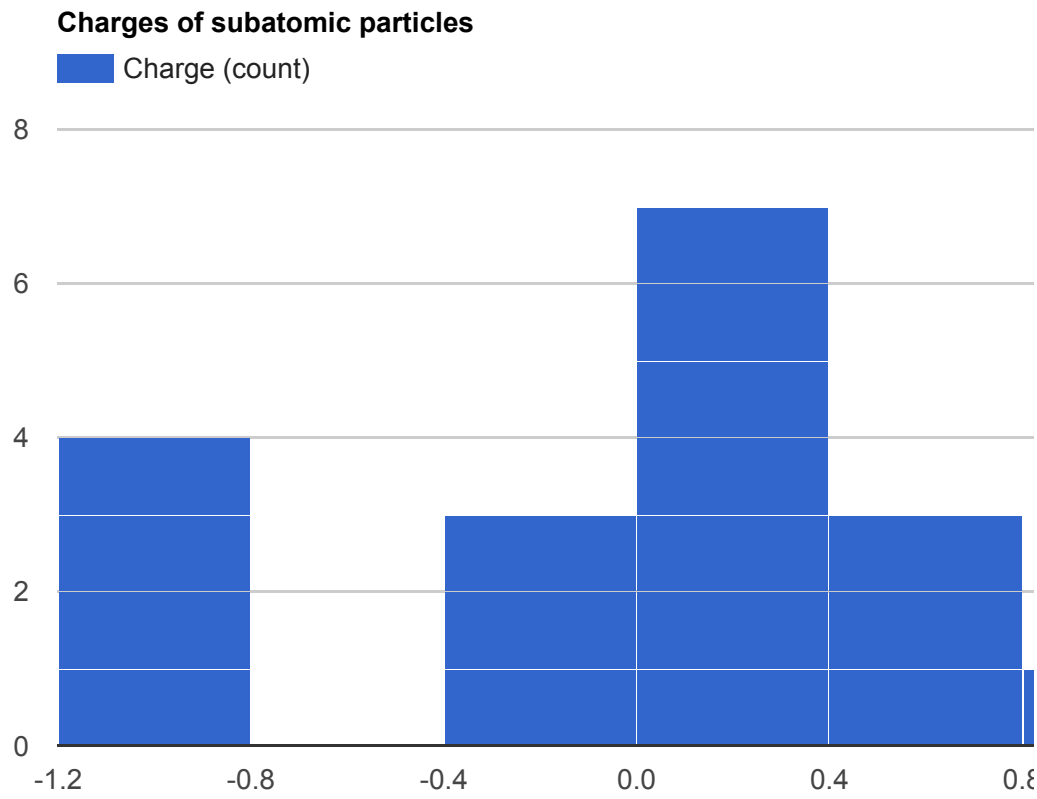



Note also that we specify the `chartArea.width` such that the number of buckets will fit more precisely without visual artifacts. Here are the options for this example.

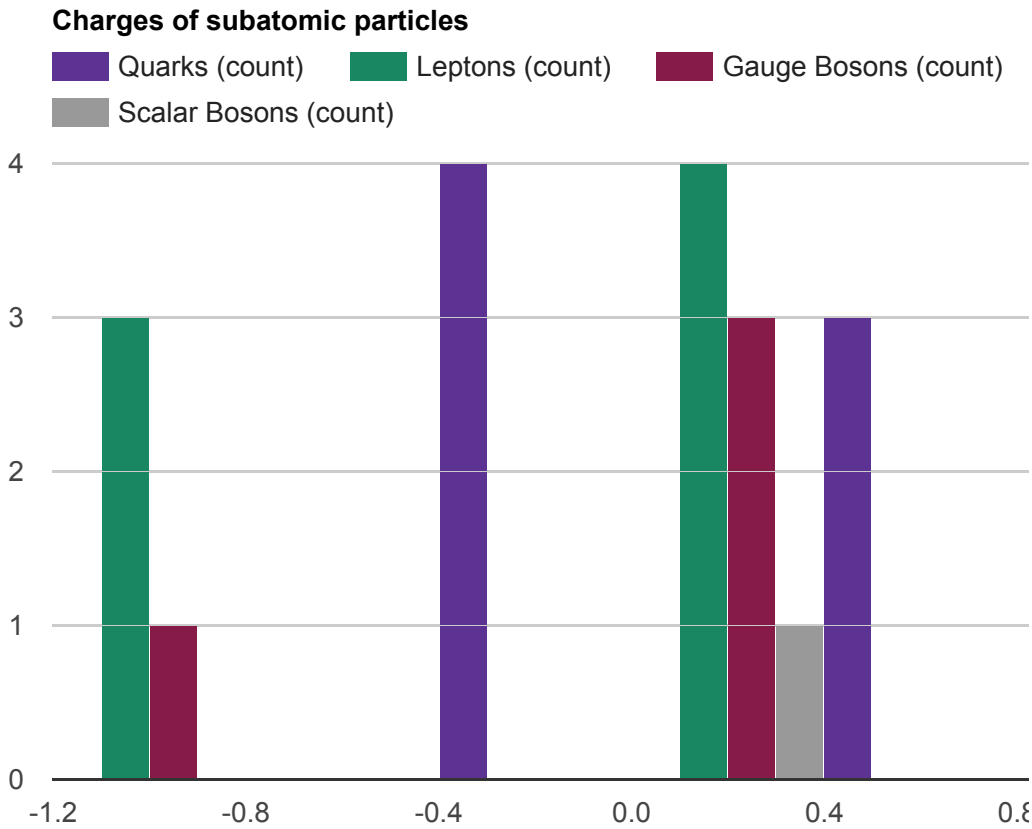
```
var options = {
  title: 'Approximating Normal Distribution',
  legend: { position: 'none' },
  colors: ['#4285F4'],

  chartArea: { width: 401 },
  hAxis: {
    ticks: [-1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1]
  },
  bar: { gap: 0 },

  histogram: {
    bucketSize: 0.02,
    maxNumBuckets: 200,
    minValue: -1,
    maxValue: 1
  }
};
```

## Multiple Series

Here's a histogram of the charges of subatomic particles, according to the Standard Model (http://en.wikipedia.org/wiki/Standard_Model):




The above chart has one series containing all the particles. Subatomic particles can be divided into four groups: quarks, leptons, and bosons. Let's treat each as its own series:




In this chart, we use a different series (and therefore color) for each of the four types of subatomic particle. We explicitly set `interpolateNulls` to `false` to ensure that the null values (needed because the series are of unequal length) aren't plotted. We also set `legend.maxLines` to add another line to the legend:

```
var data = google.visualization.arrayToDataTable([
  ['Quarks', 'Leptons', 'Gauge Bosons', 'Scalar Bosons'],
  [2/3, -1, 0, 0],
  [2/3, -1, 0, null],
  [2/3, -1, 0, null],
  [-1/3, 0, 1, null],
  [-1/3, 0, -1, null],
  [-1/3, 0, null, null],
  [-1/3, 0, null, null]
]);

var options = {
  title: 'Charges of subatomic particles',
  legend: { position: 'top', maxLines: 2 },
  colors: ['#5C3292', '#1A8763', '#871B47', '#999999'],
```

```
  interpolateNulls: false,
};
```

## Loading

The `google.charts.load` package name is `"corechart"`.

```
google.charts.load("current", {packages: ["corechart"]});
```

The visualization's class name is `google.visualization.Histogram`:

```
var visualization = new google.visualization.Histogram(container);
```

## Data Format

There are two ways to populate a histogram datatable. When there's only one series:

```
var data = google.visualization.arrayToDataTable([
  ['Name', 'Number'],
  ['Name 1', number1],
  ['Name 2', number2],
  ['Name 3', number3],
  ...
]);
```

...and when there are multiple series:

```
var data = google.visualization.arrayToDataTable([
  ['Series Name 1', 'Series Name 2', 'Series Name 3', ...],
  [series1_number1, series2_number1, series3_number1, ...],
  [series1_number2, series2_number2, series3_number2, ...],
  [series1_number3, series2_number3, series3_number3, ...],
  ...
]);
```

No optional column roles are supported for histograms at the moment.

## Configuration Options

| Name | |
|---|---|
| animation.duration | The duration of the animation, in milliseconds. For details, see t (https://developers.google.com/chart/interactive/docs/animat<br><br>**Type:** number<br>**Default:** 0 |
| animation.easing | The easing function applied to the animation. The following opt<br><br>• 'linear' - Constant speed.<br>• 'in' - Ease in - Start slow and speed up.<br>• 'out' - Ease out - Start fast and slow down.<br>• 'inAndOut' - Ease in and out - Start slow, speed up, then slow<br><br>**Type:** string<br>**Default:** 'linear' |
| animation.startup | Determines if the chart will animate on the initial draw. If `true`, animate to its final state.<br><br>**Type:** boolean<br>**Default** false |
| axisTitlesPosition | Where to place the axis titles, compared to the chart area. Supp<br><br>• in - Draw the axis titles inside the chart area.<br>• out - Draw the axis titles outside the chart area.<br>• none - Omit the axis titles.<br><br>**Type:** string<br>**Default:** 'out' |
| backgroundColor | The background color for the main area of the chart. Can be eit example: `'red'` or `'#00cc00'`, or an object with the following<br><br>**Type:** string or object<br>**Default:** 'white' |
| backgroundColor.stroke | The color of the chart border, as an HTML color string.<br><br>**Type:** string<br>**Default:** '#666' |
| backgroundColor.strokeWidth | The border width, in pixels.<br><br>**Type:** number<br>**Default:** 0 |
| backgroundColor.fill | The chart fill color, as an HTML color string. |

| | |
|---|---|
| | **Type:** string<br>**Default:** 'white' |
| bar.groupWidth | The width of a group of bars, specified in either of these format<br>• Pixels (e.g. 50).<br>• Percentage of the available width for each group (e.g. '20%'), no space between them.<br>**Type:** number or string<br>**Default:** The golden ratio (http://en.wikipedia.org/wiki/Golden_r |
| chartArea | An object with members to configure the placement and size of drawn, excluding axis and legends). Two formats are supported A simple number is a value in pixels; a number followed by % is `{left:20,top:0,width:'50%',height:'75%'}`<br>**Type:** object<br>**Default:** null |
| chartArea.backgroundColor | Chart area background color. When a string is used, it can be eit English color name. When an object is used, the following prope<br>• `stroke`: the color, provided as a hex string or English color r<br>• `strokeWidth`: if provided, draws a border around the chart color of `stroke`).<br>**Type:** string or object<br>**Default:** 'white' |
| chartArea.left | How far to draw the chart from the left border.<br>**Type:** number or string<br>**Default:** auto |
| chartArea.top | How far to draw the chart from the top border.<br>**Type:** number or string<br>**Default:** auto |
| chartArea.width | Chart area width.<br>**Type:** number or string<br>**Default:** auto |
| chartArea.height | Chart area height.<br>**Type:** number or string<br>**Default:** auto |
| colors | The colors to use for the chart elements. An array of strings, wh string, for example: `colors:['red','#004411']`. |

| | |
|---|---|
| | **Type:** Array of strings<br>**Default:** default colors |
| dataOpacity | The transparency of data points, with 1.0 being completely opac<br>histogram, bar, and column charts, this refers to the visible data<br>rectangles in the others. In charts where *selecting data* creates<br>this refers to the circles that appear upon hover or selection. Th<br>and this option has no effect on other charts. (To change the op<br>opacity (https://developers.google.com/chart/interactive/docs<br><br>**Type:** number<br>**Default:** 1.0 |
| enableInteractivity | Whether the chart throws user-based events or reacts to user in<br>throw 'select' or other interaction-based events (but *will* throw re<br>hovertext or otherwise change depending on user input.<br><br>**Type:** boolean<br>**Default:** true |
| focusTarget | The type of the entity that receives focus on mouse hover. Also<br>mouse click, and which data table element is associated with ev<br><br>• 'datum' - Focus on a single data point. Correlates to a cell in<br>• 'category' - Focus on a grouping of all data points along the r<br>data table.<br><br>In focusTarget 'category' the tooltip displays all the category val<br>values of different series.<br><br>**Type:** string<br>**Default:** 'datum' |
| fontSize | The default font size, in pixels, of all text in the chart. You can ov<br>chart elements.<br><br>**Type:** number<br>**Default:** automatic |
| fontName | The default font face for all text in the chart. You can override th<br>elements.<br><br>**Type:** string<br>**Default:** 'Arial' |
| forceIFrame | Draws the chart inside an inline frame. (Note that on IE8, this op<br>in i-frames.)<br><br>**Type:** boolean<br>**Default:** false |
| | |

| | |
|---|---|
| hAxis | An object with members to configure various horizontal axis ele object, you can use object literal notation, as shown here:<br><br>`{ title: 'Hello', titleTextStyle: { color: '#FF0000' } }`<br><br>**Type:** object<br>**Default:** null |
| hAxis.gridlines | An object with members to configure the gridlines on the horizo object, you can use object literal notation, as shown here:<br><br>`{color: '#333', count: 4}`<br><br>**Type:** object<br>**Default:** null |
| hAxis.gridlines.color | The color of the horizontal gridlines inside the chart area. Speci<br><br>**Type:** string<br>**Default:** '#CCC' |
| hAxis.gridlines.count | The number of horizontal gridlines inside the chart area. Minimu automatically compute the number of gridlines.<br><br>**Type:** number<br>**Default:** 5 |
| hAxis.gridlines.units | Overrides the default format for various aspects of date/datetin with chart computed gridlines. Allows formatting for years, mor milliseconds.<br><br>General format is:<br><br>`gridlines: { units: { years: {format: [/*format strings here*/]}, months: {format: [/*format strings here*/]} days: {format: [/*format strings here*/]} hours: {format: [/*format strings here*/]} minutes: {format: [/*format strings here*/] seconds: {format: [/*format strings here*/] milliseconds: {format: [/*format strings he } }` |

| | |
|---|---|
| | }<br><br>Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesa<br><br>**Type:** object<br>**Default:** null |
| hAxis.minorGridlines | An object with members to configure the minor gridlines on the hAxis.gridlines option.<br><br>**Type:** object<br>**Default:** null |
| hAxis.minorGridlines.color | The color of the horizontal minor gridlines inside the chart area.<br><br>**Type:** string<br>**Default:** A blend of the gridline and background colors |
| hAxis.minorGridlines.count | The number of horizontal minor gridlines between two regular g<br><br>**Type:** number<br>**Default:** 0 |
| hAxis.minorGridlines.units | Overrides the default format for various aspects of date/datetin with chart computed minorGridlines. Allows formatting for year seconds, and milliseconds.<br><br>General format is:<br><br>`gridlines: {`<br>`  units: {`<br>`  years: {format: [/*format strings here*/]},`<br>`  months: {format: [/*format strings here*/]}`<br>`  days: {format: [/*format strings here*/]}`<br>`  hours: {format: [/*format strings here*/]}`<br>`  minutes: {format: [/*format strings here*/]`<br>`  seconds: {format: [/*format strings here*/]`<br>`  milliseconds: {format: [/*format strings he`<br>`  }`<br>`}`<br><br>Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesa<br><br>**Type:** object<br>**Default:** null |
| | |

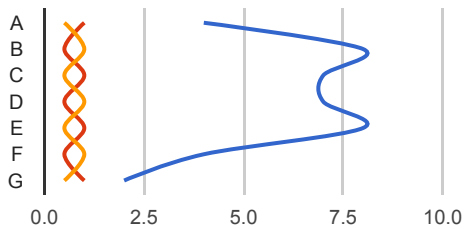
| | |
|---|---|
| hAxis.textPosition | Position of the horizontal axis text, relative to the chart area. Su<br><br>**Type:** string<br>**Default:** 'out' |
| hAxis.textStyle | An object that specifies the horizontal axis text style. The objec<br><br>`{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }`<br><br>The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontSize`**.<br><br>**Type:** object<br>**Default:** **`{color: 'black', fontName: <global-font-r size>}`** |
| hAxis.title | **`hAxis`** property that specifies the title of the horizontal axis.<br><br>**Type:** string<br>**Default:** null |
| hAxis.titleTextStyle | An object that specifies the horizontal axis title text style. The o<br><br>`{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }`<br><br>The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontSize`**.<br><br>**Type:** object<br>**Default:** **`{color: 'black', fontName: <global-font-r size>}`** |
| hAxis.allowContainerBoundaryTextCufoff | If false, will hide outermost labels rather than allow them to be c will allow label cropping.<br><br>**Type:** boolean<br>**Default:** false |
| hAxis.slantedText | If true, draw the horizontal axis text at an angle, to help fit more horizontal axis text upright. Default behavior is to slant text if it |

| | |
|---|---|
| | that this option is available only when the **`hAxis.textPositi`**<br><br>**Type:** boolean<br>**Default:** automatic |
| hAxis.slantedTextAngle | The angle of the horizontal axis text, if it's drawn slanted. Ignore is in auto mode, and the chart decided to draw the text horizont<br><br>**Type:** number, 1—90<br>**Default:** 30 |
| hAxis.maxAlternation | Maximum number of levels of horizontal axis text. If axis text la might shift neighboring labels up or down in order to fit labels c most number of levels to use; the server can use fewer levels, if<br><br>**Type:** number<br>**Default:** 2 |
| hAxis.maxTextLines | Maximum number of lines allowed for the text labels. Labels ca and the number of lines is, by default, limited by the height of th<br><br>**Type:** number<br>**Default:** auto |
| hAxis.minTextSpacing | Minimum horizontal spacing, in pixels, allowed between two adj spaced too densely, or they are too long, the spacing can drop b of the label-unclutter measures will be applied (e.g, truncating t<br><br>**Type:** number<br>**Default:** The value of **`hAxis.textStyle.fontSize`** |
| hAxis.showTextEvery | How many horizontal axis labels to show, where 1 means show label, and so on. Default is to try to show as many labels as pos<br><br>**Type:** number<br>**Default:** automatic |
| hAxis.viewWindowMode | Specifies how to scale the horizontal axis to render the values w string values are supported:<br><br>• 'pretty' - Scale the horizontal values so that the maximum an bit inside the left and right of the chart area. This will cause **`haxis.viewWindow.max`** to be ignored.<br>• 'maximized' - Scale the horizontal values so that the maximu left and right of the chart area. This will cause **`haxis.view`** **`haxis.viewWindow.max`** to be ignored.<br>• 'explicit' - A deprecated option for specifying the left and righ (Deprecated because it's redundant with **`haxis.viewWind`** **`haxis.viewWindow.max`**.) Data values outside these valu **`hAxis.viewWindow`** object describing the maximum and m |

| | |
|---|---|
| | **Type:** string<br>**Default:** Equivalent to 'pretty', but `haxis.viewWindow.min` an<br>precedence if used. |
| hAxis.viewWindow | Specifies the cropping range of the horizontal axis.<br><br>**Type:** object<br>**Default:** null |
| hAxis.viewWindow.max | The zero-based row index where the cropping window ends. Dat<br>cropped out. In conjunction with `vAxis.viewWindowMode.mi`<br>max) that denotes the element indices to display. In other words<br>`< max` will be displayed.<br><br>Ignored when `hAxis.viewWindowMode` is 'pretty' or 'maximize<br><br>**Type:** number<br>**Default:** auto |
| hAxis.viewWindow.min | The zero-based row index where the cropping window begins. D<br>be cropped out. In conjunction with `vAxis.viewWindowMode`<br>[min, max) that denotes the element indices to display. In other<br>`index < max` will be displayed.<br><br>Ignored when `hAxis.viewWindowMode` is 'pretty' or 'maximize<br><br>**Type:** number<br>**Default:** auto |
| histogram.bucketSize | Hardcode the size of each histogram bar, rather than letting it be<br><br>**Type:** number<br>**Default:** auto |
| histogram.hideBucketItems | Omit the thin divisions between the blocks of the histogram, ma<br><br>**Type:** boolean<br>**Default:** false |
| histogram.lastBucketPercentile | When calculating the histogram's bucket size, ignore the top and<br>percent. The values are still included in the histogram, but do no<br><br>**Type:** number<br>**Default:** 0 |
| histogram.minValue | Expand the range of buckets to include this value.<br><br>**Type:** number<br>**Default:** auto - use data min |
| histogram.maxValue | Expand the range of buckets to include this value. |

| | |
|---|---|
| | **Type:** number<br>**Default:** auto - use data max |
| height | Height of the chart, in pixels.<br><br>**Type:** number<br>**Default:** height of the containing element |
| interpolateNulls | Whether to guess the value of missing points. If true, it will gues<br>on neighboring points. If false, it will leave a break in the line at t<br><br>This **is not** supported by Area<br>(https://developers.google.com/chart/interactive/docs/gallery/<br>`isStacked: true/'percent'/'relative'/'absolute'`<br><br>**Type:** boolean<br>**Default:** false |
| isStacked | If set to true, stacks the elements for all series at each domain v<br>(https://developers.google.com/chart/interactive/docs/gallery/<br>(https://developers.google.com/chart/interactive/docs/gallery/<br>(https://developers.google.com/chart/interactive/docs/gallery/<br>Charts reverses the order of legend items to better correspond v<br>(E.g. series 0 will be the bottom-most legend item). This **does n**<br>(https://developers.google.com/chart/interactive/docs/gallery/<br><br>The `isStacked` option also supports 100% stacking, where the<br>value are rescaled to add up to 100%.<br><br>The options for `isStacked` are:<br><br>• `false` — elements will not stack. This is the default option.<br>• `true` — stacks elements for all series at each domain value.<br>• `'percent'` — stacks elements for all series at each domain<br>add up to 100%, with each element's value calculated as a pe<br>• `'relative'` — stacks elements for all series at each doma<br>they add up to 1, with each element's value calculated as a fr<br>• `'absolute'` — functions the same as `isStacked: true`.<br><br>For 100% stacking, the calculated value for each element will ap<br><br>The target axis will default to tick values based on the relative 0<br>`'relative'`, and 0-100% for `'percent'` (**Note:** when using th<br>values are displayed as percentages, however the actual values<br>because the percentage axis ticks are the result of applying a fo<br>values. When using `isStacked: 'percent'`, be sure to spec<br>0-1 scale values). You can customize the gridlines/tick values a<br>`hAxis/vAxis` options. |

| | |
|---|---|
| | 100% stacking only supports data values of type **`number`**, and n<br><br>**Type:** boolean/string<br>**Default:** false |
| legend | An object with members to configure various aspects of the leg<br>you can use object literal notation, as shown here:<br><br>`{position: 'top', textStyle: {color: 'blue', fo`<br><br>**Type:** object<br>**Default:** null |
| legend.alignment | Alignment of the legend. Can be one of the following:<br><br>• 'start' - Aligned to the start of the area allocated for the leger<br>• 'center' - Centered in the area allocated for the legend.<br>• 'end' - Aligned to the end of the area allocated for the legend.<br><br>Start, center, and end are relative to the style -- vertical or horizo<br>'right' legend, 'start' and 'end' are at the top and bottom, respecti<br>would be at the left and right of the area, respectively.<br><br>The default value depends on the legend's position. For 'bottom<br>legends default to 'start'.<br><br>**Type:** string<br>**Default:** automatic |
| legend.maxLines | Maximum number of lines in the legend. Set this to a number gr<br>legend. Note: The exact logic used to determine the actual num<br><br>This option currently works only when legend.position is 'top'.<br><br>**Type:** number<br>**Default:** 1 |
| legend.position | Position of the legend. Can be one of the following:<br><br>• 'bottom' - Below the chart.<br>• 'left' - To the left of the chart, provided the left axis has no se<br>the legend on the left, use the option **`targetAxisIndex:`**<br>• 'in' - Inside the chart, by the top left corner.<br>• 'none' - No legend is displayed.<br>• 'right' - To the right of the chart. Incompatible with the **`vAxes`**<br>• 'top' - Above the chart.<br><br>**Type:** string |

| | |
|---|---|
| | **Default:** 'right' |
| legend.textStyle | An object that specifies the legend text style. The object has thi<br><br>`{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }`<br><br>The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontSize`**.<br><br>**Type:** object<br>**Default:** **`{color: 'black', fontName: <global-font-r size>}`** |
| orientation | The orientation of the chart. When set to **`'vertical'`**, rotates instance) a column chart becomes a bar chart, and an area cha<br><br>  <br><br><br>**Type:** string<br>**Default:** 'horizontal' |
| reverseCategories | If set to true, will draw series from right to left. The default is to<br><br>**Type:** boolean<br>**Default:** false |
| series | An array of objects, each describing the format of the correspor values for a series, specify an empty object {}. If a series or a va be used. Each object supports the following properties:<br><br>• **`color`** - The color to use for this series. Specify a valid HTM<br>• **`labelInLegend`** - The description of the series to appear in<br>• **`targetAxisIndex`** - Which axis to assign this series to, wh opposite axis. Default value is 0; set to 1 to define a chart wh against different axes. At least one series much be allocated different scale for different axes.<br>• **`visibleInLegend`** - A boolean value, where true means tha and false means that it should not. Default is true. |

| | |
|---|---|
| | You can specify either an array of objects, each of which applies can specify an object where each child has a numeric key indica example, the following two declarations are identical, and decla from the legend, and the fourth as red and absent from the lege<br><br>`series: [`<br>`  {color: 'black', visibleInLegend: false}, {},`<br>`  {color: 'red', visibleInLegend: false}`<br>`]`<br>`series: {`<br>`  0:{color: 'black', visibleInLegend: false},`<br>`  3:{color: 'red', visibleInLegend: false}`<br>`}`<br><br>**Type:** Array of objects, or object with nested objects<br>**Default:** {} |
| theme | A theme is a set of predefined option values that work together visual effect. Currently only one theme is available:<br><br>• 'maximized' - Maximizes the area of the chart, and draws the chart area. Sets the following options:<br><br>`chartArea: {width: '100%', height: '100%'},`<br>`legend: {position: 'in'},`<br>`titlePosition: 'in', axisTitlesPosition: 'in`<br>`hAxis: {textPosition: 'in'}, vAxis: {textPos`<br><br>**Type:** string<br>**Default:** null |
| title | Text to display above the chart.<br><br>**Type:** string<br>**Default:** no title |
| titlePosition | Where to place the chart title, compared to the chart area. Supp<br><br>• in - Draw the title inside the chart area.<br>• out - Draw the title outside the chart area.<br>• none - Omit the title.<br><br>**Type:** string<br>**Default:** 'out' |
| titleTextStyle | An object that specifies the title text style. The object has this fo |

| | |
|---|---|
| | `{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }`<br><br>The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontSize`**.<br><br>**Type:** object<br>**Default:** **`{color: 'black', fontName: <global-font-r size>}`** |
| tooltip | An object with members to configure various tooltip elements. T can use object literal notation, as shown here:<br><br>`{textStyle: {color: '#FF0000'}, showColorCode:`<br><br>**Type:** object<br>**Default:** null |
| tooltip.isHtml | If set to true, use HTML-rendered (rather than SVG-rendered) toc (https://developers.google.com/chart/interactive/docs/custom<br><br>★ **Note:** customization of the HTML tooltip content via the tooltip (https://developers.google.com/chart/interactive/docs/roles#t Bubble Chart (https://developers.google.com/chart/interactive/<br><br>**Type:** boolean<br>**Default:** false |
| tooltip.showColorCode | If true, show colored squares next to the series information in th **`focusTarget`** is set to 'category', otherwise the default is false<br><br>**Type:** boolean<br>**Default:** automatic |
| tooltip.textStyle | An object that specifies the tooltip text style. The object has this<br><br>`{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }` |

| | |
|---|---|
| | The `color` can be any HTML color string, for example: `'red'` c<br>`fontSize`.<br><br>**Type:** object<br>**Default:** `{color: 'black', fontName: <global-font-r`<br>`size>}` |
| tooltip.trigger | The user interaction that causes the tooltip to be displayed:<br><br>• 'focus' - The tooltip will be displayed when the user hovers ov<br>• 'none' - The tooltip will not be displayed.<br><br>**Type:** string<br>**Default:** 'focus' |
| vAxes | Specifies properties for individual vertical axes, if the chart has<br>is a `vAxis` object, and can contain all the properties supported<br>override any global settings for the same property.<br><br>To specify a chart with multiple vertical axes, first define a new<br>`series.targetAxisIndex`, then configure the axis using `vA`<br>series 2 to the right axis and specifies a custom title and text st<br><br>`{ series: { 2: { targetAxisIndex:1 } }, vAxes: { 1: { title:'Losses', textStyle: {color: 'red'} } } }`<br><br>This property can be either an object or an array: the object is a<br>numeric label that specifies the axis that it defines--this is the fo<br>of objects, one per axis. For example, the following array-style n<br>shown above:<br><br>`vAxes: [ {}, // Nothing specified for axis 0 { title:'Losses', textStyle: {color: 'red'} // Axis 1 }` |

| | |
|---|---|
| | ]<br><br>**Type:** Array of object, or object with child objects<br>**Default:** null |
| vAxis | An object with members to configure various vertical axis eleme<br>you can use object literal notation, as shown here:<br><br>`{title: 'Hello', titleTextStyle: {color: '#FF00`<br><br>**Type:** object<br>**Default:** null |
| vAxis.baseline | `vAxis` property that specifies the baseline for the vertical axis.<br>grid line or smaller than the lowest grid line, it will be rounded to<br><br>**Type:** number<br>**Default:** automatic |
| vAxis.baselineColor | Specifies the color of the baseline for the vertical axis. Can be a<br>`'red'` or `'#00cc00'`.<br><br>**Type:** number<br>**Default:** 'black' |
| vAxis.direction | The direction in which the values along the vertical axis grow. Sp<br>values.<br><br>**Type:** 1 or -1<br>**Default:** 1 |
| vAxis.format | A format string for numeric axis labels. This is a subset of the IC<br>(http://icu-project.org/apiref/icu4c/classDecimalFormat.html#<br>`{format:'#,###%'}` will display values "1,000%", "750%", and<br>can also supply any of the following:<br><br>• `{format: 'none'}`: displays numbers with no formatting<br>• `{format: 'decimal'}`: displays numbers with thousands<br>• `{format: 'scientific'}`: displays numbers in scientific<br>• `{format: 'currency'}`: displays numbers in the local cu<br>• `{format: 'percent'}`: displays numbers as percentages<br>• `{format: 'short'}`: displays abbreviated numbers (e.g.,<br>• `{format: 'long'}`: displays numbers as full words (e.g.,<br><br>The actual formatting applied to the label is derived from the loc<br>more details, see loading charts with a specific locale |

| | |
|---|---|
| | (https://developers.google.com/chart/interactive/docs/library_<br>.<br><br>**Type:** string<br>**Default:** auto |
| vAxis.gridlines | An object with members to configure the gridlines on the vertica<br>object, you can use object literal notation, as shown here:<br><br>`{color: '#333', count: 4}`<br><br>**Type:** object<br>**Default:** null |
| vAxis.gridlines.color | The color of the vertical gridlines inside the chart area. Specify a<br><br>**Type:** string<br>**Default:** '#CCC' |
| vAxis.gridlines.count | The number of vertical gridlines inside the chart area. Minimum<br>compute the number of gridlines.<br><br>**Type:** number<br>**Default:** 5 |
| vAxis.gridlines.units | Overrides the default format for various aspects of date/datetin<br>with chart computed gridlines. Allows formatting for years, mor<br>milliseconds.<br><br>General format is:<br><br>`gridlines: {`<br>`  units: {`<br>`  years: {format: [/*format strings here*/]},`<br>`  months: {format: [/*format strings here*/]}`<br>`  days: {format: [/*format strings here*/]}`<br>`  hours: {format: [/*format strings here*/]}`<br>`  minutes: {format: [/*format strings here*/]`<br>`  seconds: {format: [/*format strings here*/]`<br>`  milliseconds: {format: [/*format strings he`<br>`  }`<br>`}`<br><br>Additional information can be found in Dates and Times<br>(https://developers.google.com/chart/interactive/docs/datesa<br><br>**Type:** object<br>**Default:** null |

| | |
|---|---|
| vAxis.minorGridlines | An object with members to configure the minor gridlines on the vAxis.gridlines option.<br><br>**Type:** object<br>**Default:** null |
| vAxis.minorGridlines.color | The color of the vertical minor gridlines inside the chart area. Sp<br><br>**Type:** string<br>**Default:** A blend of the gridline and background colors |
| vAxis.minorGridlines.count | The number of vertical minor gridlines between two regular grid<br><br>**Type:** number<br>**Default:** 0 |
| vAxis.minorGridlines.units | Overrides the default format for various aspects of date/datetin with chart computed minorGridlines. Allows formatting for year seconds, and milliseconds.<br><br>General format is:<br><br>`gridlines: {`<br>`  units: {`<br>`  years: {format: [/*format strings here*/]},`<br>`  months: {format: [/*format strings here*/]}`<br>`  days: {format: [/*format strings here*/]}`<br>`  hours: {format: [/*format strings here*/]}`<br>`  minutes: {format: [/*format strings here*/]`<br>`  seconds: {format: [/*format strings here*/]`<br>`  milliseconds: {format: [/*format strings he`<br>`  }`<br>`}`<br><br>Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesa<br><br>**Type:** object<br>**Default:** null |
| vAxis.logScale | If true, makes the vertical axis a logarithmic scale. Note: All valu<br><br>**Type:** boolean<br>**Default:** false |
| vAxis.scaleType | `vAxis` property that makes the vertical axis a logarithmic scale<br><br>• null - No logarithmic scaling is performed. |

| | |
|---|---|
| | • 'log' - Logarithmic scaling. Negative and zero values are not p setting `vAxis: { logscale: true }`.<br>• 'mirrorLog' - Logarithmic scaling in which negative and zero v a negative number is the negative of the log of the absolute v linear scale.<br><br>**Type:** string<br>**Default:** null |
| vAxis.textPosition | Position of the vertical axis text, relative to the chart area. Suppo<br><br>**Type:** string<br>**Default:** 'out' |
| vAxis.textStyle | An object that specifies the vertical axis text style. The object ha<br><br>`{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }`<br><br>The `color` can be any HTML color string, for example: `'red'` o `fontSize`.<br><br>**Type:** object<br>**Default:** `{color: 'black', fontName: <global-font-n size>}` |
| vAxis.ticks | Replaces the automatically generated Y-axis ticks with the spec should be either a valid tick value (such as a number, date, datet an object, it should have a `v` property for the tick value, and an o string to be displayed as the label.<br><br>Examples:<br><br>• `vAxis: { ticks: [5,10,15,20] }`<br>• `vAxis: { ticks: [{v:32, f:'thirty two'}, {v:0`<br>• `vAxis: { ticks: [new Date(2014,3,15), new Dat`<br>• `vAxis: { ticks: [16, {v:32, f:'thirty two'},` `}`<br><br>**Type:** Array of elements<br>**Default:** auto |
| vAxis.title | `vAxis` property that specifies a title for the vertical axis.<br><br>**Type:** string |

| | |
|---|---|
| | **Default:** no title |
| vAxis.titleTextStyle | An object that specifies the vertical axis title text style. The obje<br><br>`{ color: <string>,`<br>`  fontName: <string>,`<br>`  fontSize: <number>,`<br>`  bold: <boolean>,`<br>`  italic: <boolean> }`<br><br>The **`color`** can be any HTML color string, for example: **`'red'`** c **`fontSize`**.<br><br>**Type:** object<br>**Default:** **`{color: 'black', fontName: <global-font-r size>}`** |
| vAxis.maxValue | Moves the max value of the vertical axis to the specified value; t Ignored if this is set to a value smaller than the maximum y-valu **`vAxis.viewWindow.max`** overrides this property.<br><br>**Type:** number<br>**Default:** automatic |
| vAxis.minValue | Moves the min value of the vertical axis to the specified value; t Ignored if this is set to a value greater than the minimum y-valu **`vAxis.viewWindow.min`** overrides this property.<br><br>**Type:** number<br>**Default:** null |
| vAxis.viewWindowMode | Specifies how to scale the vertical axis to render the values with values are supported:<br><br>• 'pretty' - Scale the vertical values so that the maximum and r inside the top and bottom of the chart area. This will cause **v** **`vaxis.viewWindow.max`** to be ignored.<br>• 'maximized' - Scale the vertical values so that the maximum and bottom of the chart area. This will cause **`vaxis.viewW`** **`vaxis.viewWindow.max`** to be ignored.<br>• 'explicit' - A deprecated option for specifying the top and bot (Deprecated because it's redundant with **`vaxis.viewWind`** **`vaxis.viewWindow.max`**. Data values outside these value **`vAxis.viewWindo`**w object describing the maximum and m<br><br>**Type:** string<br>**Default:** Equivalent to 'pretty', but **`vaxis.viewWindow.min`** an precedence if used. |

| | |
|---|---|
| vAxis.viewWindow | Specifies the cropping range of the vertical axis.<br><br>**Type:** object<br>**Default:** null |
| vAxis.viewWindow.max | The maximum vertical data value to render.<br><br>Ignored when **`vAxis.viewWindowMode`** is 'pretty' or 'maximize<br><br>**Type:** number<br>**Default:** auto |
| vAxis.viewWindow.min | The minimum horizontal data value to render.<br><br>Ignored when **`vAxis.viewWindowMode`** is 'pretty' or 'maximize<br><br>**Type:** number<br>**Default:** auto |
| width | Width of the chart, in pixels.<br><br>**Type:** number<br>**Default:** width of the containing element |

## Methods

| Method | |
|---|---|
| `draw(data, options)` | Draws the chart. The chart accepts further method calls only after the (#Events)event is fired. `Extended description` (https://developers.google.com/chart/interactive/docs/reference#vis<br><br>**Return Type:** none |
| `getAction(actionID)` | Returns the tooltip action object with the requested **`actionID`**.<br><br>**Return Type:** object |
| `getBoundingBox(id)` | Returns an object containing the left, top, width, and height of chart ele The format for **`id`** isn't yet documented (they're the return values of eve (https://developers.google.com/chart/interactive/docs/events)), but h some examples:<br><br>`var cli = chart.getChartLayoutInterface();`<br><br>**Height of the chart area** |

| | |
|---|---|
| | `cli.getBoundingBox('chartarea').height`<br>**Width of the third bar in the first series of a bar or column cl**<br>`cli.getBoundingBox('bar#0#2').width`<br>**Bounding box of the fifth wedge of a pie chart**<br>`cli.getBoundingBox('slice#4')`<br>**Bounding box of the chart data of a vertical (e.g., column) c**<br>`cli.getBoundingBox('vAxis#0#gridline')`<br>**Bounding box of the chart data of a horizontal (e.g., bar) cha**<br>`cli.getBoundingBox('hAxis#0#gridline')`<br><br>Values are relative to the container of the chart. Call this *after* the chart<br>**Return Type:** object |
| `getChartAreaBoundingBox()` | Returns an object containing the left, top, width, and height of the chart (i.e., excluding labels and legend):<br>`var cli = chart.getChartLayoutInterface();`<br><br>`cli.getChartAreaBoundingBox().left`<br>`cli.getChartAreaBoundingBox().top`<br>`cli.getChartAreaBoundingBox().height`<br>`cli.getChartAreaBoundingBox().width`<br><br>Values are relative to the container of the chart. Call this *after* the chart<br>**Return Type:** object |
| `getChartLayoutInterface()` | Returns an object containing information about the onscreen placeme chart and its elements.<br>The following methods can be called on the returned object:<br>• `getBoundingBox`<br>• `getChartAreaBoundingBox`<br>• `getHAxisValue` |

| | |
|---|---|
| | • `getVAxisValue`<br>• `getXLocation`<br>• `getYLocation`<br><br>Call this *after* the chart is drawn.<br><br>**Return Type:** object |
| `getHAxisValue(position, optional_axis_index)` | Returns the logical horizontal value at `position`, which is an offset fr container's left edge. Can be negative.<br><br>Example: `chart.getChartLayoutInterface().getHAxisValue`<br><br>Call this *after* the chart is drawn.<br><br>**Return Type:** number |
| `getImageURI()` | Returns the chart serialized as an image URI.<br><br>Call this *after* the chart is drawn.<br><br>See Printing PNG Charts (https://developers.google.com/chart/interactive/docs/printing).<br><br>**Return Type:** string |
| `getSelection()` | Returns an array of the selected chart entities. Selectable entities are b entries and categories. For this chart, only one entity can be selected a moment. `Extended description` (https://developers.google.com/chart/interactive/docs/reference#vis .<br><br>**Return Type:** Array of selection elements |
| `getVAxisValue(position, optional_axis_index)` | Returns the logical vertical value at `position`, which is an offset from container's top edge. Can be negative.<br><br>Example: `chart.getChartLayoutInterface().getVAxisValue`<br><br>Call this *after* the chart is drawn.<br><br>**Return Type:** number |
| `getXLocation(position, optional_axis_index)` | Returns the screen x-coordinate of `position` relative to the chart's co<br><br>Example: `chart.getChartLayoutInterface().getXLocation(`<br><br>Call this *after* the chart is drawn.<br><br>**Return Type:** number |
| `getYLocation(position,` | Returns the screen y-coordinate of `position` relative to the chart's co |

| | |
|---|---|
| `optional_axis_index)` | Example: `chart.getChartLayoutInterface().getYLocation(`<br><br>Call this *after* the chart is drawn.<br><br>**Return Type:** number |
| `removeAction(actionID)` | Removes the tooltip action with the requested `actionID` from the cha<br><br>**Return Type: none** |
| `setAction(action)` | Sets a tooltip action to be executed when the user clicks on the action<br><br>The `setAction` method takes an object as its action parameter. This should specify 3 properties: `id`— the ID of the action being set, `text` – should appear in the tooltip for the action, and `action` — the function be run when a user clicks on the action text.<br><br>Any and all tooltip actions should be set prior to calling the chart's `dra` method. Extended description (https://developers.google.com/chart/interactive/docs/reference#vis<br><br>**Return Type: none** |
| `setSelection()` | Selects the specified chart entities. Cancels any previous selection. Se entities are bars, legend entries and categories. For this chart, only one be selected at a time. `Extended description` (https://developers.google.com/chart/interactive/docs/reference#vis .<br><br>**Return Type:** none |
| `clearChart()` | Clears the chart, and releases all of its allocated resources.<br><br>**Return Type:** none |

## Events

For more information on how to use these events, see Basic Interactivity (https://developers.google.com/chart/interactive/docs/basic_interactivity), Handling Events (https://developers.google.com/chart/interactive/docs/events), and Firing Events (https://developers.google.com/chart/interactive/docs/dev/events).

| Name | |
|---|---|
| `animationfinish` | Fired when transition animation is complete.<br><br>**Properties:** none |

| | |
|---|---|
| `click` | Fired when the user clicks inside the chart. Can be used to identify when the title, data elements, legend entries, axes, gridlines, or labels are clicked.<br><br>**Properties:** targetID |
| `error` | Fired when an error occurs when attempting to render the chart.<br><br>**Properties:** id, message |
| `onmouseover` | Fired when the user mouses over a visual entity. Passes back the row and column indices of the corresponding data table element. A bar correlates to a cell in the data table, a legend entry to a column (row index is null), and a category to a row (column index is null).<br><br>**Properties:** row, column |
| `onmouseout` | Fired when the user mouses away from a visual entity. Passes back the row and column indices of the corresponding data table element. A bar correlates to a cell in the data table, a legend entry to a column (row index is null), and a category to a row (column index is null).<br><br>**Properties:** row, column |
| `ready` | The chart is ready for external method calls. If you want to interact with the chart, and call methods after you draw it, you should set up a listener for this event *before* you call the `draw` method, and call them only after the event was fired.<br><br>**Properties:** none |
| `select` | Fired when the user clicks a visual entity. To learn what has been selected, call `getSelection()` (#Methods).<br><br>**Properties:** none |

## Data Policy

All code and data are processed and rendered in the browser. No data is sent to any server.

*上次更新日期：二月 23, 2017*