Visualization: Bubble Chart

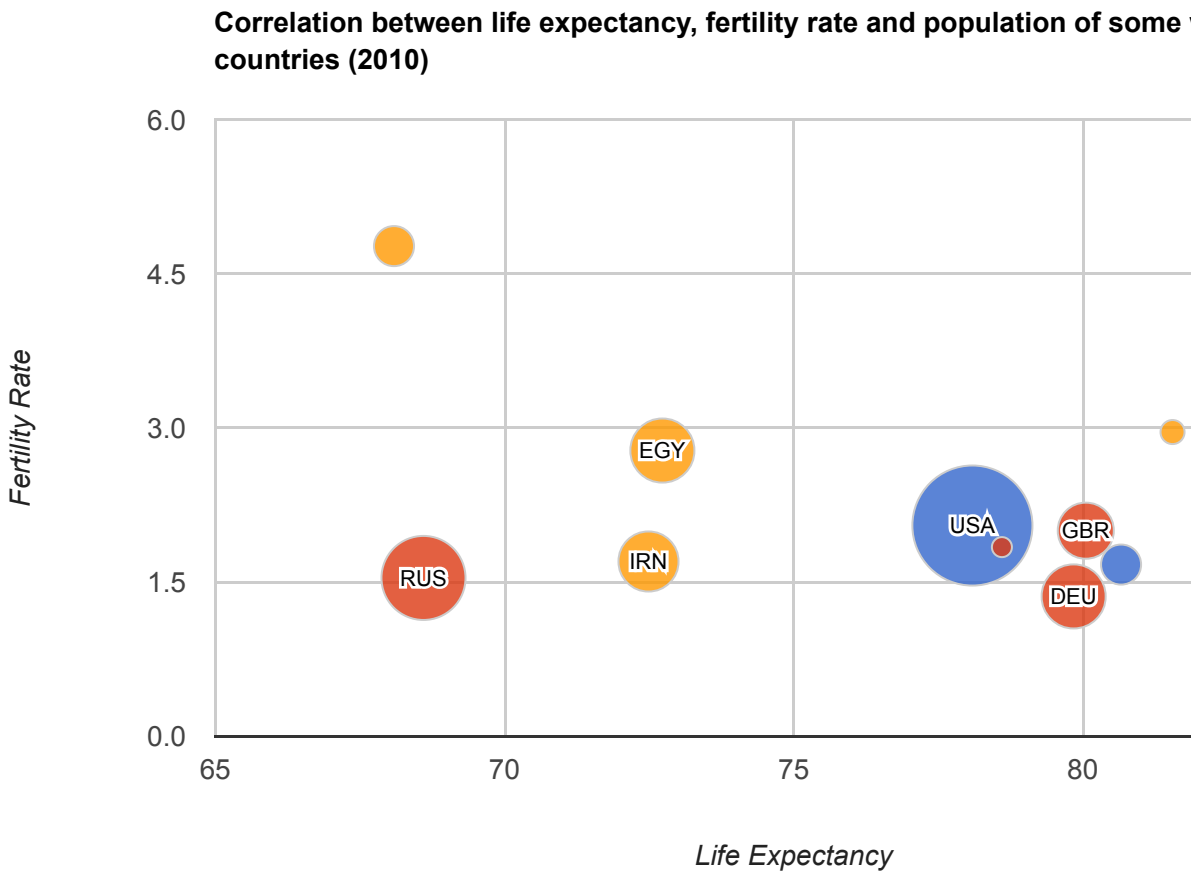
Overview

A bubble chart that is rendered within the browser using SVG (http://www.w3.org/Graphics/SVG/) or VML (http://en.wikipedia.org/wiki/Vector_Markup_Language). Displays tips when hovering over bubbles.

A bubble chart is used to visualize a data set with two to four dimensions. The first two dimensions are visualized as coordinates, the third as color and the fourth as size.

Example

Series Example

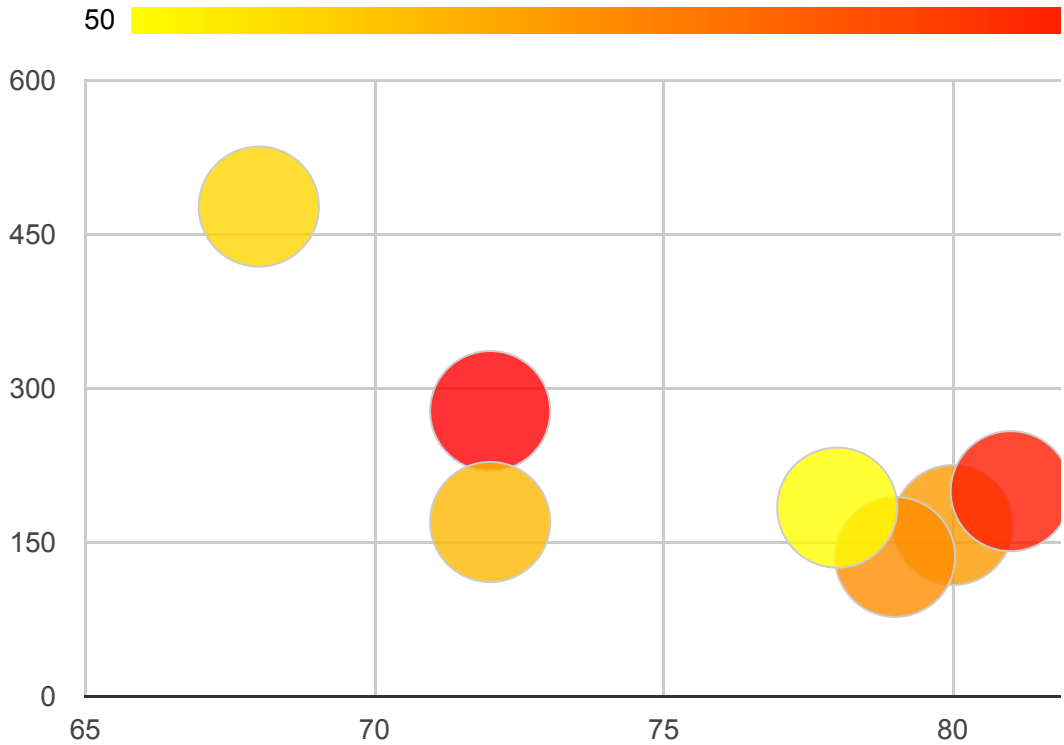


CODE IT YOURSELF ON JSFIDDLE

```
<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(drawSeriesChart);

    function drawSeriesChart() {

      var data = google.visualization.arrayToDataTable([
        ['ID', 'Life Expectancy', 'Fertility Rate', 'Region',     'Population
        ['CAN',    80.66,              1.67,      'North America',  33739900]
        ['DEU',    79.84,              1.36,      'Europe',         81902307]
        ['DNK',    78.6,               1.84,      'Europe',         5523095],
        ['EGY',    72.73,              2.78,      'Middle East',    79716203]
        ['GBR',    80.05,              2,         'Europe',         61801570]
        ['IRN',    72.49,              1.7,       'Middle East',    73137148]
        ['IRQ',    68.09,              4.77,      'Middle East',    31090763]
        ['ISR',    81.55,              2.96,      'Middle East',    7485600],
        ['RUS',    68.6,               1.54,      'Europe',         141850000
        ['USA',    78.09,              2.05,      'North America',  307007000
      ]);

      var options = {
        title: 'Correlation between life expectancy, fertility rate ' +
               'and population of some world countries (2010)',
        hAxis: {title: 'Life Expectancy'},
        vAxis: {title: 'Fertility Rate'},
        bubble: {textStyle: {fontSize: 11}}
      };

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

Color By Numbers

You can use the `colorAxis` option to color the bubbles in proportion to a value, as shown in the example below.

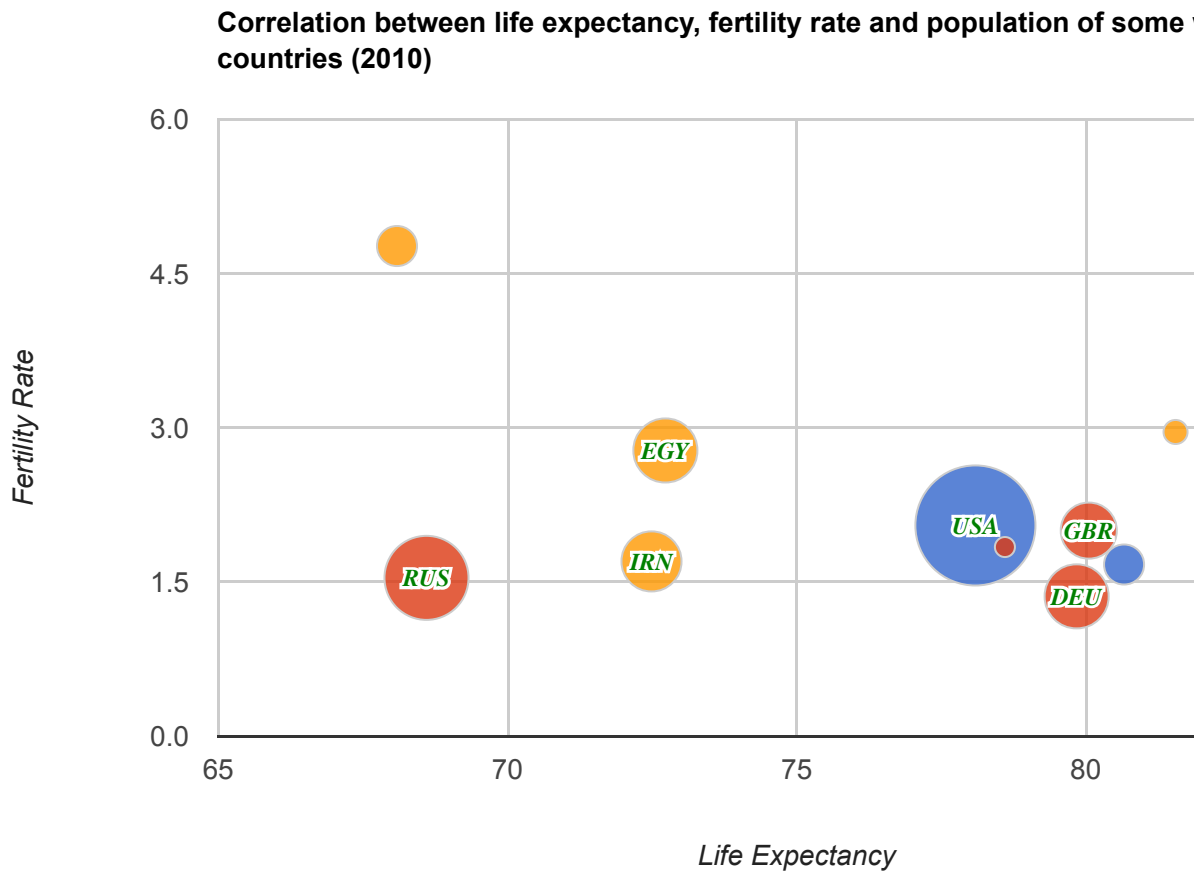



```
<html>
  <head>
    <script type="text/javascript" src="https://www.gstatic.com/charts/loader
    <script type="text/javascript">
      google.charts.load("current", {packages:["corechart"]});
      google.charts.setOnLoadCallback(drawChart);
      function drawChart() {
        var data = google.visualization.arrayToDataTable([
          ['ID', 'X', 'Y', 'Temperature'],
          ['',   80,  167,      120],
          ['',   79,  136,      130],
          ['',   78,  184,      50],
          ['',   72,  278,      230],
          ['',   81,  200,      210],
          ['',   72,  170,      100],
          ['',   68,  477,      80]
        ]);

        var options = {
```

```
          colorAxis: {colors: ['yellow', 'red']}
        };

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

Customizing Labels

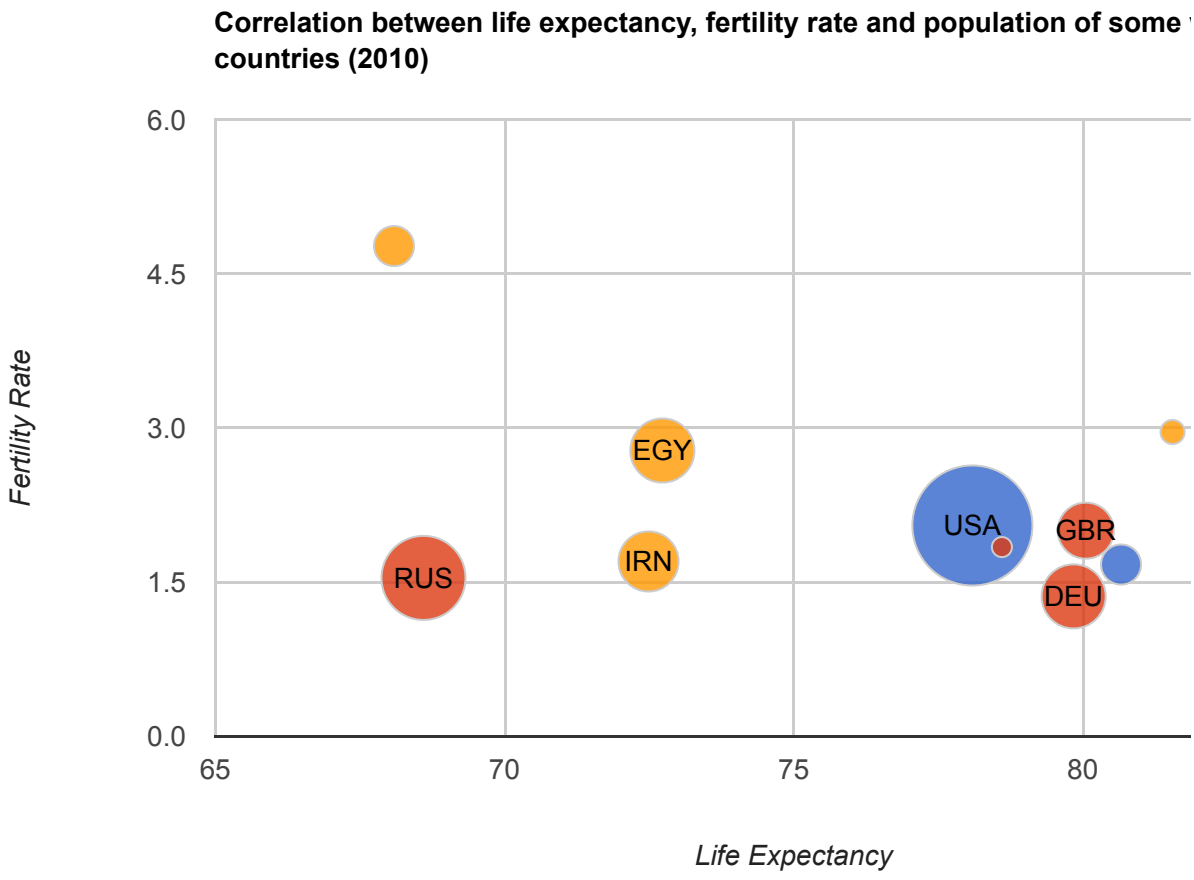
You can control the bubble typeface, font, and color with the `bubble.textStyle` option:



OPTIONS FULL WEB PAGE

```
var options = {
  title: 'Correlation between life expectancy, fertility rate ' +
         'and population of some world countries (2010)',
  hAxis: {title: 'Life Expectancy'},
  vAxis: {title: 'Fertility Rate'},
  bubble: {
    textStyle: {
      fontSize: 12,
      fontName: 'Times-Roman',
      color: 'green',
      bold: true,
      italic: true
    }
  }
};
```

Labels on the above chart are hard to read, and one of the reasons is the white space around them. That's called an *aura*, and if you'd prefer your charts without them, you can set `bubble.textStyle.auraColor` to `'none'`.



OPTIONS FULL WEB PAGE

```
var options = {
  title: 'Correlation between life expectancy, fertility rate ' +
         'and population of some world countries (2010)',
  hAxis: {title: 'Life Expectancy'},
  vAxis: {title: 'Fertility Rate'},
  bubble: {
    textStyle: {
      auraColor: 'none'
    }
  }
};
```

Loading

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

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

The visualization's class name is `google.visualization.BubbleChart`.

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

Data Format

Rows: Each row in the table represents a single bubble.

Columns:

	Column 0	Column 1	Column 2	Column 3 (optional)	Column 4 (optional)
Purpose:	ID (name) of the bubble	X coordinate	Y coordinate	Either a series ID or a value representing a color on a gradient scale, depending on the column type: • string A string that identifies bubbles in the same series. Use the same value to identify all bubbles that belong to the same series; bubbles in the same series will be	Size; values in this column are mapped to actual pixel values using the `sizeAxis` option.

				assigned the same color. Series can be configured using the `series` option. • number A value that is mapped to an actual color on a gradient scale using the `colorAxis` option.	
Data Type:	string	number	number	string or number	number

Configuration Options

Name	
animation.duration	The duration of the animation, in milliseconds. For details, see the anin (https://developers.google.com/chart/interactive/docs/animation). **Type:** number **Default:** 0
animation.easing	The easing function applied to the animation. The following options ar • 'linear' - Constant speed. • 'in' - Ease in - Start slow and speed up. • 'out' - Ease out - Start fast and slow down. • 'inAndOut' - Ease in and out - Start slow, speed up, then slow down. **Type:** string **Default:** 'linear'
animation.startup	Determines if the chart will animate on the initial draw. If `true`, the cha animate to its final state. **Type:** boolean **Default** false
axisTitlesPosition	Where to place the axis titles, compared to the chart area. Supported v • in - Draw the axis titles inside the chart area. • out - Draw the axis titles outside the chart area. • none - Omit the axis titles. **Type:** string **Default:** 'out'

backgroundColor	The background color for the main area of the chart. Can be either a si example: `'red'` or `'#00cc00'`, or an object with the following proper **Type:** string or object **Default:** 'white'
backgroundColor.stroke	The color of the chart border, as an HTML color string. **Type:** string **Default:** '#666'
backgroundColor.strokeWidth	The border width, in pixels. **Type:** number **Default:** 0
backgroundColor.fill	The chart fill color, as an HTML color string. **Type:** string **Default:** 'white'
bubble	An object with members to configure the visual properties of the bubbl **Type:** object **Default:** null
bubble.opacity	The opacity of the bubbles, where 0 is fully transparent and 1 is fully op **Type:** number between 0.0 and 1.0 **Default:** 0.8
bubble.stroke	The color of the bubbles' stroke. **Type:** string **Default:** '#ccc'

bubble.textStyle	An object that specifies the bubble text style. The object has this forma
	
	`{color: <string>, fontName: <string>, fontSize: <num`
	
	The `color` can be any HTML color string, for example: `'red'` or `'#00`
	`fontSize`.
	
	Type: object
	Default: `{color: 'black', fontName: <global-font-name>,`
	`size>}`

chartArea	An object with members to configure the placement and size of the ch drawn, excluding axis and legends). Two formats are supported: a num A simple number is a value in pixels; a number followed by % is a perce `{left:20,top:0,width:'50%',height:'75%'}`

	Type: object **Default:** null
chartArea.backgroundColor	Chart area background color. When a string is used, it can be either a h English color name. When an object is used, the following properties c • **`stroke`**: the color, provided as a hex string or English color name. • **`strokeWidth`**: if provided, draws a border around the chart area of color of **`stroke`**). **Type:** string or object **Default:** 'white'
chartArea.left	How far to draw the chart from the left border. **Type:** number or string **Default:** auto
chartArea.top	How far to draw the chart from the top border. **Type:** number or string **Default:** auto
chartArea.width	Chart area width. **Type:** number or string **Default:** auto
chartArea.height	Chart area height. **Type:** number or string **Default:** auto
colors	The colors to use for the chart elements. An array of strings, where eac string, for example: **`colors:['red','#004411']`**. **Type:** Array of strings **Default:** default colors
colorAxis	An object that specifies a mapping between color column values and c specify properties of this object, you can use object literal notation, as `{minValue: 0,  colors: ['#FF0000', '#00FF00']}` **Type:** object **Default:** null
colorAxis.minValue	If present, specifies a minimum value for chart color data. Color data v be rendered as the first color in the **`colorAxis.colors`** range. **Type:** number **Default:** Minimum value of color column in chart data

colorAxis.maxValue	If present, specifies a maximum value for chart color data. Color data v will be rendered as the last color in the **`colorAxis.colors`** range. **Type:** number **Default:** Maximum value of color column in chart data
colorAxis.values	If present, controls how values are associated with colors. Each value i corresponding color in the **`colorAxis.colors`** array. These values a Coloring is done according to a gradient of the values specified here. N option is equivalent to specifying [minValue, maxValue]. **Type:** array of numbers **Default:** null
colorAxis.colors	Colors to assign to values in the visualization. An array of strings, wher string, for example: **`colorAxis: {colors:['red','#004411']}`** values; the gradient will include all your values, plus calculated interme as the smallest value, and the last color as the highest. **Type:** array of color strings **Default:** null
colorAxis.legend	An object that specifies the style of the gradient color legend. **Type:** object **Default:** null
colorAxis.legend.position	Position of the legend. Can be one of the following: • 'top' - Above the chart. • 'bottom' - Below the chart. • 'in' - Inside the chart, at the top. • 'none' - No legend is displayed. **Type:** object **Default:** 'top'
colorAxis.legend.textStyle	An object that specifies the legend text style. The object has this forma `{color: <string>, fontName: <string>, fontSize: <num` The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00`** **`fontSize`**. **Type:** object **Default:** **`{color: 'black', fontName: <global-font-name>, size>}`**
colorAxis.legend.numberFormat	A format string for numeric labels. This is a subset of the ICU pattern s

	(http://icu-project.org/apiref/icu4c/classDecimalFormat.html#_details `{numberFormat:'.##'}` will display values "10.66", "10.6", and "10.0 **Type:** string **Default:** auto
enableInteractivity	Whether the chart throws user-based events or reacts to user interacti throw 'select' or other interaction-based events (but *will* throw ready or hovertext or otherwise change depending on user input. **Type:** boolean **Default:** true
explorer	The `explorer` option allows users to pan and zoom Google charts. `ex` default explorer behavior, enabling users to pan horizontally and vertica and out by scrolling. This feature is **experimental** and may change in future releases.  **Note:** The explorer only works with continuous axes (such as numbers **Type:** object **Default:** null
explorer.actions	The Google Charts explorer supports three actions: • `dragToPan`: Drag to pan around the chart horizontally and verticall horizontal axis, use `explorer: { axis: 'horizontal' }`. Sir • `dragToZoom`: The explorer's default behavior is to zoom in and out `explorer: { actions: ['dragToZoom', 'rightClickToR` across a rectangular area zooms into that area. We recommend usi whenever `dragToZoom` is used. See `explorer.maxZoomIn`, `expl` `explorer.zoomDelta` for zoom customizations. • `rightClickToReset`: Right clicking on the chart returns it to the d **Type:** Array of strings **Default:** ['dragToPan', 'rightClickToReset']
explorer.axis	By default, users can pan both horizontally and vertically when the `exp` want to users to only pan horizontally, use `explorer: { axis: 'ho` `explorer: { axis: 'vertical' }` enables vertical-only panning **Type:** string **Default:** both horizontal and vertical panning
explorer.keepInBounds	By default, users can pan all around, regardless of where the data is. To beyond the original chart, use `explorer: { keepInBounds: true` **Type:** boolean **Default:** false

explorer.maxZoomIn	The maximum that the explorer can zoom in. By default, users will be a they'll see only 25% of the original view. Setting **`explorer: { maxZo`** zoom in only far enough to see half of the original view. **Type:** number **Default:** 0.25
explorer.maxZoomOut	The maximum that the explorer can zoom out. By default, users will be that the chart will take up only 1/4 of the available space. Setting **`expl`** would let users zoom out far enough that the chart would take up only **Type:** number **Default:** 4
explorer.zoomDelta	When users zoom in or out, **`explorer.zoomDelta`** determines how r the number, the smoother and slower the zoom. **Type:** number **Default:** 1.5
fontSize	The default font size, in pixels, of all text in the chart. You can override chart elements. **Type:** number **Default:** automatic
fontName	The default font face for all text in the chart. You can override this usin elements. **Type:** string **Default:** 'Arial'
forceIFrame	Draws the chart inside an inline frame. (Note that on IE8, this option is in i-frames.) **Type:** boolean **Default:** false

hAxis	An object with members to configure various horizontal axis elements. object, you can use object literal notation, as shown here:
	
	`{`
	`  title: 'Hello',`
	`  titleTextStyle: {`
	`  color: '#FF0000'`
	`  }`
	`}`
	
	Type: object

	Default: null
hAxis.baseline	The baseline for the horizontal axis. **Type:** number **Default:** automatic
hAxis.baselineColor	The color of the baseline for the horizontal axis. Can be any HTML colo `'#00cc00'`. **Type:** number **Default:** 'black'
hAxis.direction	The direction in which the values along the horizontal axis grow. Specif values. **Type:** 1 or -1 **Default:** 1
hAxis.format	A format string for numeric axis labels. This is a subset of the ICU patt (http://icu-project.org/apiref/icu4c/classDecimalFormat.html#_details `{format:'#,###%'}` will display values "1,000%", "750%", and "50%" can also supply any of the following: • `{format: 'none'}`: displays numbers with no formatting (e.g., 8 • `{format: 'decimal'}`: displays numbers with thousands separa • `{format: 'scientific'}`: displays numbers in scientific notati • `{format: 'currency'}`: displays numbers in the local currency • `{format: 'percent'}`: displays numbers as percentages (e.g., 8 • `{format: 'short'}`: displays abbreviated numbers (e.g., 8M) • `{format: 'long'}`: displays numbers as full words (e.g., 8 millio The actual formatting applied to the label is derived from the locale the more details, see loading charts with a specific locale (https://developers.google.com/chart/interactive/docs/library_loading . **Type:** string **Default:** auto
hAxis.gridlines	An object with members to configure the gridlines on the horizontal ax object, you can use object literal notation, as shown here: `{color: '#333', count: 4}` **Type:** object **Default:** null

hAxis.gridlines.color	The color of the horizontal gridlines inside the chart area. Specify a val **Type:** string **Default:** '#CCC'
hAxis.gridlines.count	The number of horizontal gridlines inside the chart area. Minimum valu automatically compute the number of gridlines. **Type:** number **Default:** 5

hAxis.gridlines.units	Overrides the default format for various aspects of date/datetime/time with chart computed gridlines. Allows formatting for years, months, da milliseconds.
	
	General format is:
	
	`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 here*/]`
	`  }`
	`}`
	
	Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesandtime
	
	Type: object
	Default: null

hAxis.minorGridlines	An object with members to configure the minor gridlines on the horizo hAxis.gridlines option. **Type:** object **Default:** null
hAxis.minorGridlines.color	The color of the horizontal minor gridlines inside the chart area. Specif **Type:** string **Default:** A blend of the gridline and background colors
hAxis.minorGridlines.count	The number of horizontal minor gridlines between two regular gridlines **Type:** number

	Default: 0

hAxis.minorGridlines.units	Overrides the default format for various aspects of date/datetime/time with chart computed minorGridlines. Allows formatting for years, mont seconds, and milliseconds.
	
	General format is:
	
	`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 here*/]`
	`  }`
	`}`
	
	Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesandtime
	
	Type: object
	Default: null

hAxis.logScale	**`hAxis`** property that makes the horizontal axis a logarithmic scale (req Set to true for yes. **Type:** boolean **Default:** false
hAxis.scaleType	**`hAxis`** property that makes the horizontal axis a logarithmic scale. Car • null - No logarithmic scaling is performed. • 'log' - Logarithmic scaling. Negative and zero values are not plotted. setting **`hAxis: { logscale: true }`**. • 'mirrorLog' - Logarithmic scaling in which negative and zero values a negative number is the negative of the log of the absolute value. V linear scale. **Type:** string **Default: null**
hAxis.textPosition	Position of the horizontal axis text, relative to the chart area. Supporte **Type:** string **Default:** 'out'

hAxis.textStyle	An object that specifies the horizontal axis text style. The object has th
	
	`{ color: <string>,`
	`fontName: <string>,`
	`fontSize: <number>,`
	`bold: <boolean>,`
	`italic: <boolean> }`
	
	The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00`**
	`fontSize`.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-name>,`**
	`size>}`

hAxis.ticks	Replaces the automatically generated X-axis ticks with the specified ar should be either a valid tick value (such as a number, date, datetime, or an object, it should have a **`v`** property for the tick value, and an optional string to be displayed as the label. Examples: • **`hAxis: { ticks: [5,10,15,20] }`** • **`hAxis: { ticks: [{v:32, f:'thirty two'}, {v:64, f`** • **`hAxis: { ticks: [new Date(2014,3,15), new Date(20`** • **`hAxis: { ticks: [16, {v:32, f:'thirty two'}, {v:64`** **`}`** **Type:** Array of elements **Default:** auto
hAxis.title	**`hAxis`** property that specifies the title of the horizontal axis. **Type:** string **Default:** null

hAxis.titleTextStyle	An object that specifies the horizontal axis title text style. The object ha
	
	`{ color: <string>,`
	`fontName: <string>,`
	`fontSize: <number>,`
	`bold: <boolean>,`
	`italic: <boolean> }`
	
	The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00`**
	`fontSize`.

	Type: object **Default:** `{color: 'black', fontName: <global-font-name>, size>}`
hAxis.maxValue	Moves the max value of the horizontal axis to the specified value; this v Ignored if this is set to a value smaller than the maximum x-value of th **`hAxis.viewWindow.max`** overrides this property. **Type:** number **Default:** automatic
hAxis.minValue	Moves the min value of the horizontal axis to the specified value; this v Ignored if this is set to a value greater than the minimum x-value of the **`hAxis.viewWindow.min`** overrides this property. **Type:** number **Default:** automatic
hAxis.viewWindowMode	Specifies how to scale the horizontal axis to render the values within th string values are supported: • 'pretty' - Scale the horizontal values so that the maximum and minir bit inside the left and right of the chart area. This will cause **`haxis`** **`haxis.viewWindow.max`** to be ignored. • 'maximized' - Scale the horizontal values so that the maximum and left and right of the chart area. This will cause **`haxis.viewWindo`** **`haxis.viewWindow.max`** to be ignored. • 'explicit' - A deprecated option for specifying the left and right scale (Deprecated because it's redundant with **`haxis.viewWindow.mi`** **`haxis.viewWindow.max`**.) Data values outside these values will b **`hAxis.viewWindow`** object describing the maximum and minimur **Type:** string **Default:** Equivalent to 'pretty', but **`haxis.viewWindow.min`** and **`haxi`** precedence if used.
hAxis.viewWindow	Specifies the cropping range of the horizontal axis. **Type:** object **Default:** null
hAxis.viewWindow.max	The maximum horizontal data value to render. Ignored when **`hAxis.viewWindowMode`** is 'pretty' or 'maximized'. **Type:** number **Default:** auto
hAxis.viewWindow.min	The minimum horizontal data value to render. Ignored when **`hAxis.viewWindowMode`** is 'pretty' or 'maximized'.

	Type: number **Default:** auto
height	Height of the chart, in pixels. **Type:** number **Default:** height of the containing element
legend	An object with members to configure various aspects of the legend. To you can use object literal notation, as shown here: `{position: 'top', textStyle: {color: 'blue', fontSiz` **Type:** object **Default:** null
legend.alignment	Alignment of the legend. Can be one of the following: • 'start' - Aligned to the start of the area allocated for the legend. • 'center' - Centered in the area allocated for the legend. • 'end' - Aligned to the end of the area allocated for the legend. Start, center, and end are relative to the style -- vertical or horizontal -- c 'right' legend, 'start' and 'end' are at the top and bottom, respectively; fo would be at the left and right of the area, respectively. The default value depends on the legend's position. For 'bottom' legend legends default to 'start'. **Type:** string **Default:** automatic
legend.maxLines	Maximum number of lines in the legend. Set this to a number greater th legend. Note: The exact logic used to determine the actual number of l This option currently works only when legend.position is 'top'. **Type:** number **Default:** 1
legend.position	Position of the legend. Can be one of the following: • 'bottom' - Below the chart. • 'left' - To the left of the chart, provided the left axis has no series ass the legend on the left, use the option `targetAxisIndex: 1`. • 'in' - Inside the chart, by the top left corner. • 'none' - No legend is displayed. • 'right' - To the right of the chart. Incompatible with the `vAxes` option

	• 'top' - Above the chart. **Type:** string **Default:** 'right'

legend.textStyle	An object that specifies the legend text style. The object has this forma
	
	`{ color: <string>,`
	`  fontName: <string>,`
	`  fontSize: <number>,`
	`  bold: <boolean>,`
	`  italic: <boolean> }`
	
	The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00`**
	`fontSize`.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-name>,`**
	`size>}`

selectionMode	When **`selectionMode`** is **`'multiple'`**, users may select multiple dat **Type:** string **Default:** 'single'
series	An object of objects, where the keys are series names (the values in th describing the format of the corresponding series in the chart. If a seri global value will be used. Each object supports the following properties • **`color`** - The color to use for this series. Specify a valid HTML color • **`visibleInLegend`** - A boolean value, where true means that the s and false means that it should not. Default is true. Example: `series: {'Europe': {color: 'green'}}` **Type:** Object with nested objects **Default:** {}
sizeAxis	An object with members to configure how values are associated with b of this object, you can use object literal notation, as shown here: `{minValue: 0,  maxSize: 20}` **Type:** object **Default:** null
sizeAxis.maxSize	Maximum radius of the largest possible bubble, in pixels.

	Type: number **Default:** 30
sizeAxis.maxValue	The size value (as appears in the chart data) to be mapped to **sizeAx** be cropped to this value. **Type:** number **Default:** Maximum value of size column in chart data
sizeAxis.minSize	Mininum radius of the smallest possible bubble, in pixels. **Type:** number **Default:** 5
sizeAxis.minValue	The size value (as appears in the chart data) to be mapped to **sizeAx** be cropped to this value. **Type:** number **Default:** Minimum value of size column in chart data
sortBubblesBySize	If true, sorts the bubbles by size so the smaller bubbles appear above t bubbles are sorted according to their order in the DataTable. **Type:** boolean **Default:** true

theme	A theme is a set of predefined option values that work together to achi
	visual effect. Currently only one theme is available:
	
	• 'maximized' - Maximizes the area of the chart, and draws the legend
	chart area. Sets the following options:
	
	`chartArea: {width: '100%', height: '100%'},`
	`legend: {position: 'in'},`
	`titlePosition: 'in', axisTitlesPosition: 'in',`
	`hAxis: {textPosition: 'in'}, vAxis: {textPosition`
	
	Type: string
	Default: null

title	Text to display above the chart. **Type:** string **Default:** no title
titlePosition	Where to place the chart title, compared to the chart area. Supported v • in - Draw the title inside the chart area. • out - Draw the title outside the chart area.

	• none - Omit the title. **Type:** string **Default:** 'out'

titleTextStyle	An object that specifies the title text style. The object has this format:
	
	`{ color: <string>,`
	`fontName: <string>,`
	`fontSize: <number>,`
	`bold: <boolean>,`
	`italic: <boolean> }`
	
	The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00`**
	`fontSize`.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-name>,`**
	`size>}`

tooltip	An object with members to configure various tooltip elements. To spec can use object literal notation, as shown here: `{textStyle: {color: '#FF0000'}, showColorCode: true}` **Type:** object **Default:** null
tooltip.isHtml	If set to true, use HTML-rendered (rather than SVG-rendered) tooltips. S (https://developers.google.com/chart/interactive/docs/customizing_t ★ **Note:** customization of the HTML tooltip content via the tooltip columr (https://developers.google.com/chart/interactive/docs/roles#tooltipr Bubble Chart (https://developers.google.com/chart/interactive/docs/g **Type:** boolean **Default:** false

tooltip.textStyle	An object that specifies the tooltip text style. The object has this forma
	
	`{ color: <string>,`
	`fontName: <string>,`
	`fontSize: <number>,`
	`bold: <boolean>,`
	`italic: <boolean> }`

	The `color` can be any HTML color string, for example: `'red'` or `'#00 fontSize`. **Type:** object **Default:** `{color: 'black', fontName: <global-font-name>, size>}`
tooltip.trigger	The user interaction that causes the tooltip to be displayed: • 'focus' - The tooltip will be displayed when the user hovers over the • 'none' - The tooltip will not be displayed. • 'selection' - The tooltip will be displayed when the user selects the e **Type:** string **Default:** 'focus'
vAxis	An object with members to configure various vertical axis elements. To you can use object literal notation, as shown here: `{title: 'Hello', titleTextStyle: {color: '#FF0000'}}` **Type:** object **Default:** null
vAxis.baseline	`vAxis` property that specifies the baseline for the vertical axis. If the b grid line or smaller than the lowest grid line, it will be rounded to the clo **Type:** number **Default:** automatic
vAxis.baselineColor	Specifies the color of the baseline for the vertical axis. Can be any HTM `'red'` or `'#00cc00'`. **Type:** number **Default:** 'black'
vAxis.direction	The direction in which the values along the vertical axis grow. Specify - values. **Type:** 1 or -1 **Default:** 1
vAxis.format	A format string for numeric axis labels. This is a subset of the ICU patt (http://icu-project.org/apiref/icu4c/classDecimalFormat.html#_details `{format:'#,###%'}` will display values "1,000%", "750%", and "50%" can also supply any of the following: • `{format: 'none'}`: displays numbers with no formatting (e.g., 8 • `{format: 'decimal'}`: displays numbers with thousands separ

	• `{format: 'scientific'}`: displays numbers in scientific notati • `{format: 'currency'}`: displays numbers in the local currency • `{format: 'percent'}`: displays numbers as percentages (e.g., 8 • `{format: 'short'}`: displays abbreviated numbers (e.g., 8M) • `{format: 'long'}`: displays numbers as full words (e.g., 8 millio The actual formatting applied to the label is derived from the locale the more details, see loading charts with a specific locale (https://developers.google.com/chart/interactive/docs/library_loading . **Type:** string **Default:** auto
vAxis.gridlines	An object with members to configure the gridlines on the vertical axis. object, you can use object literal notation, as shown here: `{color: '#333', count: 4}` **Type:** object **Default:** null
vAxis.gridlines.color	The color of the vertical gridlines inside the chart area. Specify a valid **Type:** string **Default:** '#CCC'
vAxis.gridlines.count	The number of vertical gridlines inside the chart area. Minimum value i compute the number of gridlines. **Type:** number **Default:** 5

vAxis.gridlines.units	Overrides the default format for various aspects of date/datetime/time
	with chart computed gridlines. Allows formatting for years, months, da
	milliseconds.
	
	General format is:
	
	`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 here*/]`

	} } Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesandtime **Type:** object **Default:** null
vAxis.minorGridlines	An object with members to configure the minor gridlines on the vertica vAxis.gridlines option. **Type:** object **Default:** null
vAxis.minorGridlines.color	The color of the vertical minor gridlines inside the chart area. Specify a **Type:** string **Default:** A blend of the gridline and background colors
vAxis.minorGridlines.count	The number of vertical minor gridlines between two regular gridlines. **Type:** number **Default:** 0

vAxis.minorGridlines.units	Overrides the default format for various aspects of date/datetime/time with chart computed minorGridlines. Allows formatting for years, mont seconds, and milliseconds.
	
	General format is:
	
	`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 here*/]`
	`  }`
	`}`
	
	Additional information can be found in Dates and Times (https://developers.google.com/chart/interactive/docs/datesandtime
	
	Type: object
	Default: null

vAxis.logScale	If true, makes the vertical axis a logarithmic scale. Note: All values mus **Type:** boolean **Default:** false
vAxis.scaleType	**`vAxis`** property that makes the vertical axis a logarithmic scale. Can b • null - No logarithmic scaling is performed. • 'log' - Logarithmic scaling. Negative and zero values are not plotted. setting **`vAxis: { logscale: true }`**. • 'mirrorLog' - Logarithmic scaling in which negative and zero values a negative number is the negative of the log of the absolute value. V linear scale. **Type:** string **Default:** null
vAxis.textPosition	Position of the vertical axis text, relative to the chart area. Supported v **Type:** string **Default:** 'out'

vAxis.textStyle	An object that specifies the vertical axis text style. The object has this
	
	`{ color: <string>,`
	`fontName: <string>,`
	`fontSize: <number>,`
	`bold: <boolean>,`
	`italic: <boolean> }`
	
	The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00 fontSize`**.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-name>, size>}`**

vAxis.ticks	Replaces the automatically generated Y-axis ticks with the specified ar should be either a valid tick value (such as a number, date, datetime, or an object, it should have a **`v`** property for the tick value, and an optional string to be displayed as the label. Examples: • **`vAxis: { ticks: [5,10,15,20] }`** • **`vAxis: { ticks: [{v:32, f:'thirty two'}, {v:64, f`** • **`vAxis: { ticks: [new Date(2014,3,15), new Date(20`**

	• `vAxis: { ticks: [16, {v:32, f:'thirty two'}, {v:64 }` **Type:** Array of elements **Default:** auto
vAxis.title	**`vAxis`** property that specifies a title for the vertical axis. **Type:** string **Default:** no title

vAxis.titleTextStyle	An object that specifies the vertical axis title text style. The object has
	
	`{ color: <string>,`
	`fontName: <string>,`
	`fontSize: <number>,`
	`bold: <boolean>,`
	`italic: <boolean> }`
	
	The **`color`** can be any HTML color string, for example: **`'red'`** or **`'#00 fontSize`**.
	
	Type: object
	Default: **`{color: 'black', fontName: <global-font-name>, size>}`**

vAxis.maxValue	Moves the max value of the vertical axis to the specified value; this will Ignored if this is set to a value smaller than the maximum y-value of th **`vAxis.viewWindow.max`** overrides this property. **Type:** number **Default:** automatic
vAxis.minValue	Moves the min value of the vertical axis to the specified value; this will Ignored if this is set to a value greater than the minimum y-value of the **`vAxis.viewWindow.min`** overrides this property. **Type:** number **Default:** null
vAxis.viewWindowMode	Specifies how to scale the vertical axis to render the values within the values are supported: • 'pretty' - Scale the vertical values so that the maximum and minimu inside the top and bottom of the chart area. This will cause **`vaxis. vaxis.viewWindow.max`** to be ignored. • 'maximized' - Scale the vertical values so that the maximum and mi and bottom of the chart area. This will cause **`vaxis.viewWindow vaxis.viewWindow.max`** to be ignored.

	• 'explicit' - A deprecated option for specifying the top and bottom sc (Deprecated because it's redundant with **`vaxis.viewWindow.mi`** **`vaxis.viewWindow.max`**. Data values outside these values will b **`vAxis.viewWindow`** object describing the maximum and minimur **Type:** string **Default:** Equivalent to 'pretty', but **`vaxis.viewWindow.min`** and **`vaxi`** precedence if used.
vAxis.viewWindow	Specifies the cropping range of the vertical axis. **Type:** object **Default:** null
vAxis.viewWindow.max	The maximum vertical data value to render. Ignored when **`vAxis.viewWindowMode`** is 'pretty' or 'maximized'. **Type:** number **Default:** auto
vAxis.viewWindow.min	The minimum horizontal data value to render. Ignored when **`vAxis.viewWindowMode`** is 'pretty' or 'maximized'. **Type:** number **Default:** auto
width	Width of the chart, in pixels. **Type:** number **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 **Return Type:** none
`getAction(actionID)`	Returns the tooltip action object with the requested **`actionID`**. **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:
	
	`var cli = chart.getChartLayoutInterface();`
	
	Height of the chart area
	
	`cli.getBoundingBox('chartarea').height`
	
	Width of the third bar in the first series of a bar or column cl
	
	`cli.getBoundingBox('bar#0#2').width`
	
	Bounding box of the fifth wedge of a pie chart
	
	`cli.getBoundingBox('slice#4')`
	
	Bounding box of the chart data of a vertical (e.g., column) c
	
	`cli.getBoundingBox('vAxis#0#gridline')`
	
	Bounding box of the chart data of a horizontal (e.g., bar) cha
	
	`cli.getBoundingBox('hAxis#0#gridline')`
	
	Values are relative to the container of the chart. Call this *after* the chart
	
	Return Type: object

`getChartAreaBoundingBox()`	Returns an object containing the left, top, width, and height of the chart (i.e., excluding labels and legend):
	
	`var cli = chart.getChartLayoutInterface();`
	
	`cli.getChartAreaBoundingBox().left`
	
	`cli.getChartAreaBoundingBox().top`
	
	`cli.getChartAreaBoundingBox().height`
	
	`cli.getChartAreaBoundingBox().width`
	
	Values are relative to the container of the chart. Call this *after* the chart
	
	Return Type: object

`getChartLayoutInterface()`	Returns an object containing information about the onscreen placemer chart and its elements. The following methods can be called on the returned object: • `getBoundingBox` • `getChartAreaBoundingBox` • `getHAxisValue` • `getVAxisValue` • `getXLocation` • `getYLocation` Call this *after* the chart is drawn. **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. Example: `chart.getChartLayoutInterface().getHAxisValue` Call this *after* the chart is drawn. **Return Type:** number
`getImageURI()`	Returns the chart serialized as an image URI. Call this *after* the chart is drawn. See Printing PNG Charts (https://developers.google.com/chart/interactive/docs/printing). **Return Type:** string
`getSelection()`	Returns an array of the selected chart entities. Selectable entities are b this chart, only one entity can be selected at any given moment. `Exter description` (https://developers.google.com/chart/interactive/docs/reference#vis . **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. Example: `chart.getChartLayoutInterface().getVAxisValue` Call this *after* the chart is drawn. **Return Type:** number

`getXLocation(position, optional_axis_index)`	Returns the screen x-coordinate of **`position`** relative to the chart's co Example: **`chart.getChartLayoutInterface().getXLocation(`** Call this *after* the chart is drawn. **Return Type:** number
`getYLocation(position, optional_axis_index)`	Returns the screen y-coordinate of **`position`** relative to the chart's co Example: **`chart.getChartLayoutInterface().getYLocation(`** Call this *after* the chart is drawn. **Return Type:** number
`removeAction(actionID)`	Removes the tooltip action with the requested **`actionID`** from the cha **Return Type: none**
`setAction(action)`	Sets a tooltip action to be executed when the user clicks on the action 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. 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 **Return Type: none**
`setSelection()`	Selects the specified chart entities. Cancels any previous selection. Se entities are bubbles. For this chart, only one entity can be selected at a Extended description (https://developers.google.com/chart/interactive/docs/reference#vis . **Return Type:** none
`clearChart()`	Clears the chart, and releases all of its allocated resources. **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. **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. **Properties:** targetID
`error`	Fired when an error occurs when attempting to render the chart. **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 bubble correlates to a row in the data table (column index is null). **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 bubble correlates to a row in the data table (column index is null). **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. **Properties:** none
`select`	Fired when the user clicks a visual entity. To learn what has been selected, call `getSelection()` (#Methods). **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