 [Chart.js](http://docs.google.com/docs/3.9.1/)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

[Home](http://docs.google.com/docs/3.9.1/)

[API](http://docs.google.com/docs/3.9.1/api/)

[Samples](http://docs.google.com/docs/3.9.1/samples/)

Ecosystem Ecosystem

* [Awesome (opens new window)](https://github.com/chartjs/awesome)
* [Slack (opens new window)](https://chartjs-slack.herokuapp.com/)
* [Stack Overflow (opens new window)](https://stackoverflow.com/questions/tagged/chart.js)

[GitHub (opens new window)](https://github.com/chartjs/Chart.js)

* [Chart.js](http://docs.google.com/docs/3.9.1/)
* Getting Started
* General
* Configuration
* Chart Types
  + [Area Chart](http://docs.google.com/docs/3.9.1/charts/area.html)
  + [Bar Chart](http://docs.google.com/docs/3.9.1/charts/bar.html)
  + [Bubble Chart](http://docs.google.com/docs/3.9.1/charts/bubble.html)
  + [Doughnut and Pie Charts](http://docs.google.com/docs/3.9.1/charts/doughnut.html)
  + [Line Chart](http://docs.google.com/docs/3.9.1/charts/line.html)
  + [Mixed Chart Types](http://docs.google.com/docs/3.9.1/charts/mixed.html)
  + [Polar Area Chart](http://docs.google.com/docs/3.9.1/charts/polar.html)
  + [Radar Chart](http://docs.google.com/docs/3.9.1/charts/radar.html)
  + [Scatter Chart](http://docs.google.com/docs/3.9.1/charts/scatter.html)
* Axes
* Developers

[**#**](#gjdgxs) Bar Chart

A bar chart provides a way of showing data values represented as vertical bars. It is sometimes used to show trend data, and the comparison of multiple data sets side by side.

config setup

const config = { type: 'bar', data: data, options: { scales: { y: { beginAtZero: true } } }, };

const config = {  
 type: 'bar',  
 data: data,  
 options: {  
 scales: {  
 y: {  
 beginAtZero: true  
 }  
 }  
 },  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ label: 'My First Dataset', data: [65, 59, 80, 81, 56, 55, 40], backgroundColor: [ 'rgba(255, 99, 132, 0.2)', 'rgba(255, 159, 64, 0.2)', 'rgba(255, 205, 86, 0.2)', 'rgba(75, 192, 192, 0.2)', 'rgba(54, 162, 235, 0.2)', 'rgba(153, 102, 255, 0.2)', 'rgba(201, 203, 207, 0.2)' ], borderColor: [ 'rgb(255, 99, 132)', 'rgb(255, 159, 64)', 'rgb(255, 205, 86)', 'rgb(75, 192, 192)', 'rgb(54, 162, 235)', 'rgb(153, 102, 255)', 'rgb(201, 203, 207)' ], borderWidth: 1 }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 label: 'My First Dataset',  
 data: [65, 59, 80, 81, 56, 55, 40],  
 backgroundColor: [  
 'rgba(255, 99, 132, 0.2)',  
 'rgba(255, 159, 64, 0.2)',  
 'rgba(255, 205, 86, 0.2)',  
 'rgba(75, 192, 192, 0.2)',  
 'rgba(54, 162, 235, 0.2)',  
 'rgba(153, 102, 255, 0.2)',  
 'rgba(201, 203, 207, 0.2)'  
 ],  
 borderColor: [  
 'rgb(255, 99, 132)',  
 'rgb(255, 159, 64)',  
 'rgb(255, 205, 86)',  
 'rgb(75, 192, 192)',  
 'rgb(54, 162, 235)',  
 'rgb(153, 102, 255)',  
 'rgb(201, 203, 207)'  
 ],  
 borderWidth: 1  
 }]  
};

## [**#**](#30j0zll) Dataset Properties

Namespaces:

* data.datasets[index] - options for this dataset only
* options.datasets.bar - options for all bar datasets
* options.elements.bar - options for all [bar elements](http://docs.google.com/docs/3.9.1/configuration/elements.html#bar-configuration)
* options - options for the whole chart

The bar chart allows a number of properties to be specified for each dataset. These are used to set display properties for a specific dataset. For example, the color of the bars is generally set this way. Only the data option needs to be specified in the dataset namespace.

| Name | Type | [Scriptable](http://docs.google.com/docs/3.9.1/general/options.html#scriptable-options) | [Indexable](http://docs.google.com/docs/3.9.1/general/options.html#indexable-options) | Default |
| --- | --- | --- | --- | --- |
| [backgroundColor](#1fob9te) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | Yes | 'rgba(0, 0, 0, 0.1)' |
| [base](#3znysh7) | number | Yes | Yes |  |
| [barPercentage](#2et92p0) | number | - | - | 0.9 |
| [barThickness](#tyjcwt) | number|string | - | - |  |
| [borderColor](#1fob9te) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | Yes | 'rgba(0, 0, 0, 0.1)' |
| [borderSkipped](#3dy6vkm) | string|boolean | Yes | Yes | 'start' |
| [borderWidth](#1t3h5sf) | number|object | Yes | Yes | 0 |
| [borderRadius](#4d34og8) | number|object | Yes | Yes | 0 |
| [categoryPercentage](#2s8eyo1) | number | - | - | 0.8 |
| [clip](#3znysh7) | number|object|false | - | - |  |
| [data](#17dp8vu) | object|object[]| number[]|string[] | - | - | **required** |
| [grouped](#3znysh7) | boolean | - | - | true |
| [hoverBackgroundColor](#3rdcrjn) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | Yes |  |
| [hoverBorderColor](#3rdcrjn) | [Color](http://docs.google.com/docs/3.9.1/general/colors.html) | Yes | Yes |  |
| [hoverBorderWidth](#3rdcrjn) | number | Yes | Yes | 1 |
| [hoverBorderRadius](#3rdcrjn) | number | Yes | Yes | 0 |
| [indexAxis](#3znysh7) | string | - | - | 'x' |
| [inflateAmount](#26in1rg) | number|'auto' | Yes | Yes | 'auto' |
| [maxBarThickness](#lnxbz9) | number | - | - |  |
| [minBarLength](#1fob9te) | number | - | - |  |
| [label](#3znysh7) | string | - | - | '' |
| [order](#3znysh7) | number | - | - | 0 |
| [pointStyle](http://docs.google.com/docs/3.9.1/configuration/elements.html#point-styles) | [pointStyle](http://docs.google.com/docs/3.9.1/configuration/elements.html#types) | Yes | - | 'circle' |
| [skipNull](#3znysh7) | boolean | - | - |  |
| [stack](#3znysh7) | string | - | - | 'bar' |
| [xAxisID](#3znysh7) | string | - | - | first x axis |
| [yAxisID](#3znysh7) | string | - | - | first y axis |

All these values, if undefined, fallback to the scopes described in [option resolution](http://docs.google.com/general/options)

### [**#**](#35nkun2) Example dataset configuration

data: {  
 datasets: [{  
 barPercentage: 0.5,  
 barThickness: 6,  
 maxBarThickness: 8,  
 minBarLength: 2,  
 data: [10, 20, 30, 40, 50, 60, 70]  
 }]  
};

### [**#**](#3znysh7) General

| Name | Description |
| --- | --- |
| base | Base value for the bar in data units along the value axis. If not set, defaults to the value axis base value. |
| clip | How to clip relative to chartArea. Positive value allows overflow, negative value clips that many pixels inside chartArea. 0 = clip at chartArea. Clipping can also be configured per side: clip: {left: 5, top: false, right: -2, bottom: 0} |
| grouped | Should the bars be grouped on index axis. When true, all the datasets at same index value will be placed next to each other centering on that index value. When false, each bar is placed on its actual index-axis value. |
| indexAxis | The base axis of the dataset. 'x' for vertical bars and 'y' for horizontal bars. |
| label | The label for the dataset which appears in the legend and tooltips. |
| order | The drawing order of dataset. Also affects order for stacking, tooltip and legend. [more](http://docs.google.com/docs/3.9.1/charts/mixed.html#drawing-order) |
| skipNull | If true, null or undefined values will not be used for spacing calculations when determining bar size. |
| stack | The ID of the group to which this dataset belongs to (when stacked, each group will be a separate stack). [more](#1ksv4uv) |
| xAxisID | The ID of the x-axis to plot this dataset on. |
| yAxisID | The ID of the y-axis to plot this dataset on. |

### [**#**](#1fob9te) Styling

The style of each bar can be controlled with the following properties:

| Name | Description |
| --- | --- |
| backgroundColor | The bar background color. |
| borderColor | The bar border color. |
| [borderSkipped](#3dy6vkm) | The edge to skip when drawing bar. |
| [borderWidth](#1t3h5sf) | The bar border width (in pixels). |
| [borderRadius](#4d34og8) | The bar border radius (in pixels). |
| minBarLength | Set this to ensure that bars have a minimum length in pixels. |
| pointStyle | Style of the point for legend. [more...](http://docs.google.com/docs/3.9.1/configuration/elements.html#point-styles) |

All these values, if undefined, fallback to the associated [elements.bar.\*](http://docs.google.com/docs/3.9.1/configuration/elements.html#bar-configuration) options.

#### [**#**](#3dy6vkm) borderSkipped

This setting is used to avoid drawing the bar stroke at the base of the fill, or disable the border radius. In general, this does not need to be changed except when creating chart types that derive from a bar chart.

Note

For negative bars in a vertical chart, top and bottom are flipped. Same goes for left and right in a horizontal chart.

Options are:

* 'start'
* 'end'
* 'middle' (only valid on stacked bars: the borders between bars are skipped)
* 'bottom'
* 'left'
* 'top'
* 'right'
* false (don't skip any borders)
* true (skip all borders)

#### [**#**](#1t3h5sf) borderWidth

If this value is a number, it is applied to all sides of the rectangle (left, top, right, bottom), except [borderSkipped](#3dy6vkm). If this value is an object, the left property defines the left border width. Similarly, the right, top, and bottom properties can also be specified. Omitted borders and [borderSkipped](#3dy6vkm) are skipped.

#### [**#**](#4d34og8) borderRadius

If this value is a number, it is applied to all corners of the rectangle (topLeft, topRight, bottomLeft, bottomRight), except corners touching the [borderSkipped](#3dy6vkm). If this value is an object, the topLeft property defines the top-left corners border radius. Similarly, the topRight, bottomLeft, and bottomRight properties can also be specified. Omitted corners and those touching the [borderSkipped](#3dy6vkm) are skipped. For example if the top border is skipped, the border radius for the corners topLeft and topRight will be skipped as well.

Stacked Charts

When the border radius is supplied as a number and the chart is stacked, the radius will only be applied to the bars that are at the edges of the stack or where the bar is floating. The object syntax can be used to override this behavior.

#### [**#**](#26in1rg) inflateAmount

This option can be used to inflate the rects that are used to draw the bars. This can be used to hide artifacts between bars when barPercentage(#barpercentage) \* categoryPercentage(#categorypercentage) is 1. The default value 'auto' should work in most cases.

### [**#**](#3rdcrjn) Interactions

The interaction with each bar can be controlled with the following properties:

| Name | Description |
| --- | --- |
| hoverBackgroundColor | The bar background color when hovered. |
| hoverBorderColor | The bar border color when hovered. |
| hoverBorderWidth | The bar border width when hovered (in pixels). |
| hoverBorderRadius | The bar border radius when hovered (in pixels). |

All these values, if undefined, fallback to the associated [elements.bar.\*](http://docs.google.com/docs/3.9.1/configuration/elements.html#bar-configuration) options.

### [**#**](#2et92p0) barPercentage

Percent (0-1) of the available width each bar should be within the category width. 1.0 will take the whole category width and put the bars right next to each other. [more...](#44sinio)

### [**#**](#2s8eyo1) categoryPercentage

Percent (0-1) of the available width each category should be within the sample width. [more...](#44sinio)

### [**#**](#tyjcwt) barThickness

If this value is a number, it is applied to the width of each bar, in pixels. When this is enforced, barPercentage and categoryPercentage are ignored.

If set to 'flex', the base sample widths are calculated automatically based on the previous and following samples so that they take the full available widths without overlap. Then, bars are sized using barPercentage and categoryPercentage. There is no gap when the percentage options are 1. This mode generates bars with different widths when data are not evenly spaced.

If not set (default), the base sample widths are calculated using the smallest interval that prevents bar overlapping, and bars are sized using barPercentage and categoryPercentage. This mode always generates bars equally sized.

### [**#**](#lnxbz9) maxBarThickness

Set this to ensure that bars are not sized thicker than this.

## [**#**](#2jxsxqh) Scale Configuration

The bar chart sets unique default values for the following configuration from the associated scale options:

| Name | Type | Default | Description |
| --- | --- | --- | --- |
| offset | boolean | true | If true, extra space is added to both edges and the axis is scaled to fit into the chart area. |
| grid.offset | boolean | true | If true, the bars for a particular data point fall between the grid lines. The grid line will move to the left by one half of the tick interval. If false, the grid line will go right down the middle of the bars. [more...](#z337ya) |

### [**#**](#3j2qqm3) Example scale configuration

options = {  
 scales: {  
 x: {  
 grid: {  
 offset: true  
 }  
 }  
 }  
};

### [**#**](#1y810tw) Offset Grid Lines

If true, the bars for a particular data point fall between the grid lines. The grid line will move to the left by one half of the tick interval, which is the space between the grid lines. If false, the grid line will go right down the middle of the bars. This is set to true for a category scale in a bar chart while false for other scales or chart types by default.

## [**#**](#4i7ojhp) Default Options

It is common to want to apply a configuration setting to all created bar charts. The global bar chart settings are stored in Chart.overrides.bar. Changing the global options only affects charts created after the change. Existing charts are not changed.

## [**#**](#44sinio) barPercentage vs categoryPercentage

The following shows the relationship between the bar percentage option and the category percentage option.

// categoryPercentage: 1.0  
// barPercentage: 1.0  
Bar: | 1.0 | 1.0 |  
Category: | 1.0 |  
Sample: |===========|  
// categoryPercentage: 1.0  
// barPercentage: 0.5  
Bar: |.5| |.5|  
Category: | 1.0 |  
Sample: |==============|  
// categoryPercentage: 0.5  
// barPercentage: 1.0  
Bar: |1.0||1.0|  
Category: | .5 |  
Sample: |==================|

## [**#**](#17dp8vu) Data Structure

All of the supported [data structures](http://docs.google.com/docs/3.9.1/general/data-structures.html) can be used with bar charts.

## [**#**](#1ksv4uv) Stacked Bar Chart

Bar charts can be configured into stacked bar charts by changing the settings on the X and Y axes to enable stacking. Stacked bar charts can be used to show how one data series is made up of a number of smaller pieces.

const stackedBar = new Chart(ctx, {  
 type: 'bar',  
 data: data,  
 options: {  
 scales: {  
 x: {  
 stacked: true  
 },  
 y: {  
 stacked: true  
 }  
 }  
 }  
});

## [**#**](#2xcytpi) Horizontal Bar Chart

A horizontal bar chart is a variation on a vertical bar chart. It is sometimes used to show trend data, and the comparison of multiple data sets side by side. To achieve this you will have to set the indexAxis property in the options object to 'y'. The default for this property is 'x' and thus will show vertical bars.

config setup

const config = { type: 'bar', data, options: { indexAxis: 'y', } };

const config = {  
 type: 'bar',  
 data,  
 options: {  
 indexAxis: 'y',  
 }  
};

const labels = Utils.months({count: 7}); const data = { labels: labels, datasets: [{ axis: 'y', label: 'My First Dataset', data: [65, 59, 80, 81, 56, 55, 40], fill: false, backgroundColor: [ 'rgba(255, 99, 132, 0.2)', 'rgba(255, 159, 64, 0.2)', 'rgba(255, 205, 86, 0.2)', 'rgba(75, 192, 192, 0.2)', 'rgba(54, 162, 235, 0.2)', 'rgba(153, 102, 255, 0.2)', 'rgba(201, 203, 207, 0.2)' ], borderColor: [ 'rgb(255, 99, 132)', 'rgb(255, 159, 64)', 'rgb(255, 205, 86)', 'rgb(75, 192, 192)', 'rgb(54, 162, 235)', 'rgb(153, 102, 255)', 'rgb(201, 203, 207)' ], borderWidth: 1 }] };

const labels = Utils.months({count: 7});  
const data = {  
 labels: labels,  
 datasets: [{  
 axis: 'y',  
 label: 'My First Dataset',  
 data: [65, 59, 80, 81, 56, 55, 40],  
 fill: false,  
 backgroundColor: [  
 'rgba(255, 99, 132, 0.2)',  
 'rgba(255, 159, 64, 0.2)',  
 'rgba(255, 205, 86, 0.2)',  
 'rgba(75, 192, 192, 0.2)',  
 'rgba(54, 162, 235, 0.2)',  
 'rgba(153, 102, 255, 0.2)',  
 'rgba(201, 203, 207, 0.2)'  
 ],  
 borderColor: [  
 'rgb(255, 99, 132)',  
 'rgb(255, 159, 64)',  
 'rgb(255, 205, 86)',  
 'rgb(75, 192, 192)',  
 'rgb(54, 162, 235)',  
 'rgb(153, 102, 255)',  
 'rgb(201, 203, 207)'  
 ],  
 borderWidth: 1  
 }]  
};

### [**#**](#1ci93xb) Horizontal Bar Chart config Options

The configuration options for the horizontal bar chart are the same as for the [bar chart](#2jxsxqh). However, any options specified on the x-axis in a bar chart, are applied to the y-axis in a horizontal bar chart.

## [**#**](#3whwml4) Internal data format

{x, y, \_custom} where \_custom is an optional object defining stacked bar properties: {start, end, barStart, barEnd, min, max}. start and end are the input values. Those two are repeated in barStart (closer to origin), barEnd (further from origin), min and max.

Last Updated: 8/3/2022, 12:46:38 PM

←  [Area Chart](http://docs.google.com/docs/3.9.1/charts/area.html)   [Bubble Chart](http://docs.google.com/docs/3.9.1/charts/bubble.html)  →