* [Chart.js](http://docs.google.com/)
* [Getting Started](http://docs.google.com/getting-started/)
  + [Installation](http://docs.google.com/getting-started/installation.html)
  + [Integration](http://docs.google.com/getting-started/integration.html)
  + [Usage](http://docs.google.com/getting-started/usage.html)
* [General](http://docs.google.com/general/)
  + [Responsive](http://docs.google.com/general/responsive.html)
  + [Pixel Ratio](http://docs.google.com/general/device-pixel-ratio.html)
  + [Interactions](http://docs.google.com/general/interactions/)
    - [Events](http://docs.google.com/general/interactions/events.html)
    - [Modes](http://docs.google.com/general/interactions/modes.html)
  + [Options](http://docs.google.com/general/options.html)
  + [Colors](http://docs.google.com/general/colors.html)
  + [Fonts](http://docs.google.com/general/fonts.html)
* [Configuration](http://docs.google.com/configuration/)
  + [Animations](http://docs.google.com/configuration/animations.html)
  + [Layout](http://docs.google.com/configuration/layout.html)
  + [Legend](http://docs.google.com/configuration/legend.html)
  + [Title](http://docs.google.com/configuration/title.html)
  + [Tooltip](http://docs.google.com/configuration/tooltip.html)
  + [Elements](http://docs.google.com/configuration/elements.html)
* [Charts](http://docs.google.com/charts/)
  + [Line](http://docs.google.com/charts/line.html)
  + [Bar](http://docs.google.com/charts/bar.html)
  + [Radar](http://docs.google.com/charts/radar.html)
  + [Doughnut & Pie](http://docs.google.com/charts/doughnut.html)
  + [Polar Area](http://docs.google.com/charts/polar.html)
  + [Bubble](http://docs.google.com/charts/bubble.html)
  + [Scatter](http://docs.google.com/charts/scatter.html)
  + [Area](http://docs.google.com/charts/area.html)
  + [Mixed](http://docs.google.com/charts/mixed.html)
* [Axes](http://docs.google.com/)
  + [Cartesian](http://docs.google.com/)
    - [Category](http://docs.google.com/category.html)
    - [Linear](http://docs.google.com/linear.html)
    - [Logarithmic](http://docs.google.com/logarithmic.html)
    - [Time](http://docs.google.com/time.html)
  + [Radial](http://docs.google.com/radial/)
    - [Linear](http://docs.google.com/radial/linear.html)
  + [Labelling](http://docs.google.com/labelling.html)
  + [Styling](http://docs.google.com/styling.html)
* [Developers](http://docs.google.com/developers/)
  + [Chart.js API](http://docs.google.com/developers/api.html)
  + [Updating Charts](http://docs.google.com/developers/updates.html)
  + [Plugins](http://docs.google.com/developers/plugins.html)
  + [New Charts](http://docs.google.com/developers/charts.html)
  + [New Axes](http://docs.google.com/developers/axes.html)
  + [Contributing](http://docs.google.com/developers/contributing.html)
* [Additional Notes](http://docs.google.com/notes/)
  + [Comparison Table](http://docs.google.com/notes/comparison.html)
  + [Popular Extensions](http://docs.google.com/notes/extensions.html)
  + [License](http://docs.google.com/notes/license.html)
* [Published with GitBook](https://www.gitbook.com)

[**Linear**](http://docs.google.com/)

Linear Cartesian Axis

The linear scale is use to chart numerical data. It can be placed on either the x or y axis. The scatter chart type automatically configures a line chart to use one of these scales for the x axis. As the name suggests, linear interpolation is used to determine where a value lies on the axis.

## Tick Configuration Options

The following options are provided by the linear scale. They are all located in the ticks sub options. These options extend the [common tick configuration](http://docs.google.com/#tick-configuration).

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Default | Description |
| beginAtZero | Boolean |  | if true, scale will include 0 if it is not already included. |
| min | Number |  | User defined minimum number for the scale, overrides minimum value from data. [more...](#gjdgxs) |
| max | Number |  | User defined maximum number for the scale, overrides maximum value from data. [more...](#gjdgxs) |
| maxTicksLimit | Number | 11 | Maximum number of ticks and gridlines to show. |
| stepSize | Number |  | User defined fixed step size for the scale. [more...](#30j0zll) |
| suggestedMax | Number |  | Adjustment used when calculating the maximum data value. [more...](#gjdgxs) |
| suggestedMin | Number |  | Adjustment used when calculating the minimum data value. [more...](#gjdgxs) |

## Axis Range Settings

Given the number of axis range settings, it is important to understand how they all interact with each other.

The suggestedMax and suggestedMin settings only change the data values that are used to scale the axis. These are useful for extending the range of the axis while maintaining the auto fit behaviour.

let minDataValue = Math.min(mostNegativeValue, options.ticks.suggestedMin);  
let maxDataValue = Math.max(mostPositiveValue, options.ticks.suggestedMax);

In this example, the largest positive value is 50, but the data maximum is expanded out to 100. However, because the lowest data value is below the suggestedMin setting, it is ignored.

let chart = new Chart(ctx, {  
 type: 'line',  
 data: {  
 datasets: [{  
 label: 'First dataset',  
 data: [0, 20, 40, 50]  
 }],  
 labels: ['January', 'February', 'March', 'April']  
 },  
 options: {  
 scales: {  
 yAxes: [{  
 ticks: {  
 suggestedMin: 50,  
 suggestedMax: 100  
 }  
 }]  
 }  
 }  
});

In contrast to the suggested\* settings, the min and max settings set explicit ends to the axes. When these are set, some data points may not be visible.

## Step Size

If set, the scale ticks will be enumerated by multiple of stepSize, having one tick per increment. If not set, the ticks are labeled automatically using the nice numbers algorithm.

This example sets up a chart with a y axis that creates ticks at 0, 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5, 5.

let options = {  
 scales: {  
 yAxes: [{  
 ticks: {  
 max: 5,  
 min: 0,  
 stepSize: 0.5  
 }  
 }]  
 }  
};

results matching ""

No results matching ""