 [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)

* [Information](http://docs.google.com/docs/3.9.1/samples/information.html)
* Bar Charts
* Line Charts
* Other charts
  + [Bubble](http://docs.google.com/docs/3.9.1/samples/other-charts/bubble.html)
  + [Combo bar/line](http://docs.google.com/docs/3.9.1/samples/other-charts/combo-bar-line.html)
  + [Doughnut](http://docs.google.com/docs/3.9.1/samples/other-charts/doughnut.html)
  + [Multi Series Pie](http://docs.google.com/docs/3.9.1/samples/other-charts/multi-series-pie.html)
  + [Pie](http://docs.google.com/docs/3.9.1/samples/other-charts/pie.html)
  + [Polar area](http://docs.google.com/docs/3.9.1/samples/other-charts/polar-area.html)
  + [Polar area centered point labels](http://docs.google.com/docs/3.9.1/samples/other-charts/polar-area-center-labels.html)
  + [Radar](http://docs.google.com/docs/3.9.1/samples/other-charts/radar.html)
  + [Radar skip points](http://docs.google.com/docs/3.9.1/samples/other-charts/radar-skip-points.html)
  + [Scatter](http://docs.google.com/docs/3.9.1/samples/other-charts/scatter.html)
  + [Scatter - Multi axis](http://docs.google.com/docs/3.9.1/samples/other-charts/scatter-multi-axis.html)
  + [Stacked bar/line](http://docs.google.com/docs/3.9.1/samples/other-charts/stacked-bar-line.html)
* Area charts
* Scales
* Scale Options
* Legend
* Title
* Subtitle
* Tooltip
* Scriptable Options
* Animations
* Advanced
* Plugins
* [Utils](http://docs.google.com/docs/3.9.1/samples/utils.html)

[**#**](#gjdgxs) Doughnut

config setup actions

const config = { type: 'doughnut', data: data, options: { responsive: true, plugins: { legend: { position: 'top', }, title: { display: true, text: 'Chart.js Doughnut Chart' } } }, };

const config = {  
 type: 'doughnut',  
 data: data,  
 options: {  
 responsive: true,  
 plugins: {  
 legend: {  
 position: 'top',  
 },  
 title: {  
 display: true,  
 text: 'Chart.js Doughnut Chart'  
 }  
 }  
 },  
};

const DATA\_COUNT = 5; const NUMBER\_CFG = {count: DATA\_COUNT, min: 0, max: 100}; const data = { labels: ['Red', 'Orange', 'Yellow', 'Green', 'Blue'], datasets: [ { label: 'Dataset 1', data: Utils.numbers(NUMBER\_CFG), backgroundColor: Object.values(Utils.CHART\_COLORS), } ] };

const DATA\_COUNT = 5;  
const NUMBER\_CFG = {count: DATA\_COUNT, min: 0, max: 100};  
const data = {  
 labels: ['Red', 'Orange', 'Yellow', 'Green', 'Blue'],  
 datasets: [  
 {  
 label: 'Dataset 1',  
 data: Utils.numbers(NUMBER\_CFG),  
 backgroundColor: Object.values(Utils.CHART\_COLORS),  
 }  
 ]  
};

const actions = [ { name: 'Randomize', handler(chart) { chart.data.datasets.forEach(dataset => { dataset.data = Utils.numbers({count: chart.data.labels.length, min: 0, max: 100}); }); chart.update(); } }, { name: 'Add Dataset', handler(chart) { const data = chart.data; const newDataset = { label: 'Dataset ' + (data.datasets.length + 1), backgroundColor: [], data: [], }; for (let i = 0; i < data.labels.length; i++) { newDataset.data.push(Utils.numbers({count: 1, min: 0, max: 100})); const colorIndex = i % Object.keys(Utils.CHART\_COLORS).length; newDataset.backgroundColor.push(Object.values(Utils.CHART\_COLORS)[colorIndex]); } chart.data.datasets.push(newDataset); chart.update(); } }, { name: 'Add Data', handler(chart) { const data = chart.data; if (data.datasets.length > 0) { data.labels.push('data #' + (data.labels.length + 1)); for (let index = 0; index < data.datasets.length; ++index) { data.datasets[index].data.push(Utils.rand(0, 100)); } chart.update(); } } }, { name: 'Hide(0)', handler(chart) { chart.hide(0); } }, { name: 'Show(0)', handler(chart) { chart.show(0); } }, { name: 'Hide (0, 1)', handler(chart) { chart.hide(0, 1); } }, { name: 'Show (0, 1)', handler(chart) { chart.show(0, 1); } }, { name: 'Remove Dataset', handler(chart) { chart.data.datasets.pop(); chart.update(); } }, { name: 'Remove Data', handler(chart) { chart.data.labels.splice(-1, 1); // remove the label first chart.data.datasets.forEach(dataset => { dataset.data.pop(); }); chart.update(); } } ];

const actions = [  
 {  
 name: 'Randomize',  
 handler(chart) {  
 chart.data.datasets.forEach(dataset => {  
 dataset.data = Utils.numbers({count: chart.data.labels.length, min: 0, max: 100});  
 });  
 chart.update();  
 }  
 },  
 {  
 name: 'Add Dataset',  
 handler(chart) {  
 const data = chart.data;  
 const newDataset = {  
 label: 'Dataset ' + (data.datasets.length + 1),  
 backgroundColor: [],  
 data: [],  
 };  
 for (let i = 0; i < data.labels.length; i++) {  
 newDataset.data.push(Utils.numbers({count: 1, min: 0, max: 100}));  
 const colorIndex = i % Object.keys(Utils.CHART\_COLORS).length;  
 newDataset.backgroundColor.push(Object.values(Utils.CHART\_COLORS)[colorIndex]);  
 }  
 chart.data.datasets.push(newDataset);  
 chart.update();  
 }  
 },  
 {  
 name: 'Add Data',  
 handler(chart) {  
 const data = chart.data;  
 if (data.datasets.length > 0) {  
 data.labels.push('data #' + (data.labels.length + 1));  
 for (let index = 0; index < data.datasets.length; ++index) {  
 data.datasets[index].data.push(Utils.rand(0, 100));  
 }  
 chart.update();  
 }  
 }  
 },  
 {  
 name: 'Hide(0)',  
 handler(chart) {  
 chart.hide(0);  
 }  
 },  
 {  
 name: 'Show(0)',  
 handler(chart) {  
 chart.show(0);  
 }  
 },  
 {  
 name: 'Hide (0, 1)',  
 handler(chart) {  
 chart.hide(0, 1);  
 }  
 },  
 {  
 name: 'Show (0, 1)',  
 handler(chart) {  
 chart.show(0, 1);  
 }  
 },  
 {  
 name: 'Remove Dataset',  
 handler(chart) {  
 chart.data.datasets.pop();  
 chart.update();  
 }  
 },  
 {  
 name: 'Remove Data',  
 handler(chart) {  
 chart.data.labels.splice(-1, 1); // remove the label first  
 chart.data.datasets.forEach(dataset => {  
 dataset.data.pop();  
 });  
 chart.update();  
 }  
 }  
];

## [**#**](#30j0zll) Docs

* [Doughnut and Pie Charts](http://docs.google.com/docs/3.9.1/charts/doughnut.html)

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

←  [Combo bar/line](http://docs.google.com/docs/3.9.1/samples/other-charts/combo-bar-line.html)   [Multi Series Pie](http://docs.google.com/docs/3.9.1/samples/other-charts/multi-series-pie.html)  →