 [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) Multi Series Pie

config setup

const config = { type: 'pie', data: data, options: { responsive: true, plugins: { legend: { labels: { generateLabels: function(chart) { // Get the default label list const original = Chart.overrides.pie.plugins.legend.labels.generateLabels; const labelsOriginal = original.call(this, chart); // Build an array of colors used in the datasets of the chart let datasetColors = chart.data.datasets.map(function(e) { return e.backgroundColor; }); datasetColors = datasetColors.flat(); // Modify the color and hide state of each label labelsOriginal.forEach(label => { // There are twice as many labels as there are datasets. This converts the label index into the corresponding dataset index label.datasetIndex = (label.index - label.index % 2) / 2; // The hidden state must match the dataset's hidden state label.hidden = !chart.isDatasetVisible(label.datasetIndex); // Change the color to match the dataset label.fillStyle = datasetColors[label.index]; }); return labelsOriginal; } }, onClick: function(mouseEvent, legendItem, legend) { // toggle the visibility of the dataset from what it currently is legend.chart.getDatasetMeta( legendItem.datasetIndex ).hidden = legend.chart.isDatasetVisible(legendItem.datasetIndex); legend.chart.update(); } }, tooltip: { callbacks: { label: function(context) { const labelIndex = (context.datasetIndex \* 2) + context.dataIndex; return context.chart.data.labels[labelIndex] + ': ' + context.formattedValue; } } } } }, };

const config = {  
 type: 'pie',  
 data: data,  
 options: {  
 responsive: true,  
 plugins: {  
 legend: {  
 labels: {  
 generateLabels: function(chart) {  
 // Get the default label list  
 const original = Chart.overrides.pie.plugins.legend.labels.generateLabels;  
 const labelsOriginal = original.call(this, chart);  
 // Build an array of colors used in the datasets of the chart  
 let datasetColors = chart.data.datasets.map(function(e) {  
 return e.backgroundColor;  
 });  
 datasetColors = datasetColors.flat();  
 // Modify the color and hide state of each label  
 labelsOriginal.forEach(label => {  
 // There are twice as many labels as there are datasets. This converts the label index into the corresponding dataset index  
 label.datasetIndex = (label.index - label.index % 2) / 2;  
 // The hidden state must match the dataset's hidden state  
 label.hidden = !chart.isDatasetVisible(label.datasetIndex);  
 // Change the color to match the dataset  
 label.fillStyle = datasetColors[label.index];  
 });  
 return labelsOriginal;  
 }  
 },  
 onClick: function(mouseEvent, legendItem, legend) {  
 // toggle the visibility of the dataset from what it currently is  
 legend.chart.getDatasetMeta(  
 legendItem.datasetIndex  
 ).hidden = legend.chart.isDatasetVisible(legendItem.datasetIndex);  
 legend.chart.update();  
 }  
 },  
 tooltip: {  
 callbacks: {  
 label: function(context) {  
 const labelIndex = (context.datasetIndex \* 2) + context.dataIndex;  
 return context.chart.data.labels[labelIndex] + ': ' + context.formattedValue;  
 }  
 }  
 }  
 }  
 },  
};

const DATA\_COUNT = 5; const NUMBER\_CFG = {count: DATA\_COUNT, min: 0, max: 100}; const labels = Utils.months({count: 7}); const data = { labels: ['Overall Yay', 'Overall Nay', 'Group A Yay', 'Group A Nay', 'Group B Yay', 'Group B Nay', 'Group C Yay', 'Group C Nay'], datasets: [ { backgroundColor: ['#AAA', '#777'], data: [21, 79] }, { backgroundColor: ['hsl(0, 100%, 60%)', 'hsl(0, 100%, 35%)'], data: [33, 67] }, { backgroundColor: ['hsl(100, 100%, 60%)', 'hsl(100, 100%, 35%)'], data: [20, 80] }, { backgroundColor: ['hsl(180, 100%, 60%)', 'hsl(180, 100%, 35%)'], data: [10, 90] } ] };

const DATA\_COUNT = 5;  
const NUMBER\_CFG = {count: DATA\_COUNT, min: 0, max: 100};  
const labels = Utils.months({count: 7});  
const data = {  
 labels: ['Overall Yay', 'Overall Nay', 'Group A Yay', 'Group A Nay', 'Group B Yay', 'Group B Nay', 'Group C Yay', 'Group C Nay'],  
 datasets: [  
 {  
 backgroundColor: ['#AAA', '#777'],  
 data: [21, 79]  
 },  
 {  
 backgroundColor: ['hsl(0, 100%, 60%)', 'hsl(0, 100%, 35%)'],  
 data: [33, 67]  
 },  
 {  
 backgroundColor: ['hsl(100, 100%, 60%)', 'hsl(100, 100%, 35%)'],  
 data: [20, 80]  
 },  
 {  
 backgroundColor: ['hsl(180, 100%, 60%)', 'hsl(180, 100%, 35%)'],  
 data: [10, 90]  
 }  
 ]  
};

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

* [Doughnut and Pie Charts](http://docs.google.com/docs/3.9.1/charts/doughnut.html)
* [Options](http://docs.google.com/docs/3.9.1/general/options.html)
  + [Scriptable Options](http://docs.google.com/docs/3.9.1/general/options.html#scriptable-options)

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

←  [Doughnut](http://docs.google.com/docs/3.9.1/samples/other-charts/doughnut.html)   [Pie](http://docs.google.com/docs/3.9.1/samples/other-charts/pie.html)  →