 [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
* Area charts
  + [Line Chart Boundaries](http://docs.google.com/docs/3.9.1/samples/area/line-boundaries.html)
  + [Line Chart Datasets](http://docs.google.com/docs/3.9.1/samples/area/line-datasets.html)
  + [Line Chart drawTime](http://docs.google.com/docs/3.9.1/samples/area/line-drawtime.html)
  + [Line Chart Stacked](http://docs.google.com/docs/3.9.1/samples/area/line-stacked.html)
  + [Radar Chart Stacked](http://docs.google.com/docs/3.9.1/samples/area/radar.html)
* 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) Line Chart Datasets

data config setup actions

const data = { labels: generateLabels(), datasets: [ { label: 'D0', data: generateData(), borderColor: Utils.CHART\_COLORS.red, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red), hidden: true }, { label: 'D1', data: generateData(), borderColor: Utils.CHART\_COLORS.orange, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.orange), fill: '-1' }, { label: 'D2', data: generateData(), borderColor: Utils.CHART\_COLORS.yellow, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.yellow), hidden: true, fill: 1 }, { label: 'D3', data: generateData(), borderColor: Utils.CHART\_COLORS.green, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.green), fill: '-1' }, { label: 'D4', data: generateData(), borderColor: Utils.CHART\_COLORS.blue, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue), fill: '-1' }, { label: 'D5', data: generateData(), borderColor: Utils.CHART\_COLORS.grey, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.grey), fill: '+2' }, { label: 'D6', data: generateData(), borderColor: Utils.CHART\_COLORS.purple, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.purple), fill: false }, { label: 'D7', data: generateData(), borderColor: Utils.CHART\_COLORS.red, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red), fill: 8 }, { label: 'D8', data: generateData(), borderColor: Utils.CHART\_COLORS.orange, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.orange), fill: 'end', hidden: true }, { label: 'D9', data: generateData(), borderColor: Utils.CHART\_COLORS.yellow, backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.yellow), fill: {above: 'blue', below: 'red', target: {value: 350}} } ] };

const data = {  
 labels: generateLabels(),  
 datasets: [  
 {  
 label: 'D0',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.red,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red),  
 hidden: true  
 },  
 {  
 label: 'D1',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.orange,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.orange),  
 fill: '-1'  
 },  
 {  
 label: 'D2',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.yellow,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.yellow),  
 hidden: true,  
 fill: 1  
 },  
 {  
 label: 'D3',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.green,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.green),  
 fill: '-1'  
 },  
 {  
 label: 'D4',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.blue,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.blue),  
 fill: '-1'  
 },  
 {  
 label: 'D5',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.grey,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.grey),  
 fill: '+2'  
 },  
 {  
 label: 'D6',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.purple,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.purple),  
 fill: false  
 },  
 {  
 label: 'D7',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.red,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.red),  
 fill: 8  
 },  
 {  
 label: 'D8',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.orange,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.orange),  
 fill: 'end',  
 hidden: true  
 },  
 {  
 label: 'D9',  
 data: generateData(),  
 borderColor: Utils.CHART\_COLORS.yellow,  
 backgroundColor: Utils.transparentize(Utils.CHART\_COLORS.yellow),  
 fill: {above: 'blue', below: 'red', target: {value: 350}}  
 }  
 ]  
};

const config = { type: 'line', data: data, options: { scales: { y: { stacked: true } }, plugins: { filler: { propagate: false }, 'samples-filler-analyser': { target: 'chart-analyser' } }, interaction: { intersect: false, }, }, };

const config = {  
 type: 'line',  
 data: data,  
 options: {  
 scales: {  
 y: {  
 stacked: true  
 }  
 },  
 plugins: {  
 filler: {  
 propagate: false  
 },  
 'samples-filler-analyser': {  
 target: 'chart-analyser'  
 }  
 },  
 interaction: {  
 intersect: false,  
 },  
 },  
};

const inputs = { min: 20, max: 80, count: 8, decimals: 2, continuity: 1 }; const generateLabels = () => { return Utils.months({count: inputs.count}); }; const generateData = () => (Utils.numbers(inputs)); Utils.srand(42);

const inputs = {  
 min: 20,  
 max: 80,  
 count: 8,  
 decimals: 2,  
 continuity: 1  
};  
const generateLabels = () => {  
 return Utils.months({count: inputs.count});  
};  
const generateData = () => (Utils.numbers(inputs));  
Utils.srand(42);

let smooth = false; let propagate = false; const actions = [ { name: 'Randomize', handler(chart) { chart.data.datasets.forEach(dataset => { dataset.data = generateData(); }); chart.update(); } }, { name: 'Propagate', handler(chart) { propagate = !propagate; chart.options.plugins.filler.propagate = propagate; chart.update(); } }, { name: 'Smooth', handler(chart) { smooth = !smooth; chart.options.elements.line.tension = smooth ? 0.4 : 0; chart.update(); } } ];

let smooth = false;  
let propagate = false;  
const actions = [  
 {  
 name: 'Randomize',  
 handler(chart) {  
 chart.data.datasets.forEach(dataset => {  
 dataset.data = generateData();  
 });  
 chart.update();  
 }  
 },  
 {  
 name: 'Propagate',  
 handler(chart) {  
 propagate = !propagate;  
 chart.options.plugins.filler.propagate = propagate;  
 chart.update();  
 }  
 },  
 {  
 name: 'Smooth',  
 handler(chart) {  
 smooth = !smooth;  
 chart.options.elements.line.tension = smooth ? 0.4 : 0;  
 chart.update();  
 }  
 }  
];

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

* [Area](http://docs.google.com/docs/3.9.1/charts/area.html)
  + [Filling modes](http://docs.google.com/docs/3.9.1/charts/area.html#filling-modes)
* [Line](http://docs.google.com/docs/3.9.1/charts/line.html)
* [Data structures (labels)](http://docs.google.com/docs/3.9.1/general/data-structures.html)
* [Axes scales](http://docs.google.com/docs/3.9.1/axes/)
  + [Common options to all axes (stacked)](http://docs.google.com/docs/3.9.1/axes/#common-options-to-all-axes)

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

←  [Line Chart Boundaries](http://docs.google.com/docs/3.9.1/samples/area/line-boundaries.html)   [Line Chart drawTime](http://docs.google.com/docs/3.9.1/samples/area/line-drawtime.html)  →