

bar-chart-comp-d3

bar-chart-comp-d3 is a Vue.js (≥ 2.5) web component that draws svg bar charts. **bar-chart-comp-d3** depends on [vue.js](#), various modules from [d3.js](#), along with [button-comp](#) and [input-comp](#) from the [deandevl](#) repositories. **bar-chart-comp-d3** can be installed via [npm install](#) with the included `package.json` file. **bar-chart-comp-d3** offers several features including:

- clickable y axis for redefining the axis scaling
- legends and axis titles are draggable to new locations
- handles grouped data
- bars can be stacked/unstacked
- axis tic label rotation for better readability
- tooltip showing a bar's value on hover
- control over decimal places for both axis and tooltip values
- CSS variables are provided for easily controlling chart colors, backgrounds and font sizes

bar-chart-comp-d3 can be installed via with the included `package.json` file for a local installation via the [npm install](#) command. Two demo folders are provided that used [Parcel](#) together with its associated `package.json` files to bundle together **bar-chart-comp-d3** along with its dependencies for a couple of simple applications. Further details are provided below for running the demos.

Props

A prop in Vue.js is a custom attribute for passing information from a parent component hosting **bar-chart-comp-d3** instance(s) to an **bar-chart-comp-d3** as a child component. **bar-chart-comp-d3** has the following props for a parent to bind and send information to:

- `chart_data` -- an array of javascript objects with pairs of variable names for keys and a string/numeric value as the keys' values
- `axis` -- a javascript object defining the x and y axis' -- see below
- `title_1` -- a string for the chart's main title
- `title_2` -- a string for the chart's sub-main title
- `stack_bars` -- a boolean to stack the bars if `true` (default is false)
- `margin_left` -- defines the number of pixels for the chart's left margin (default is 80)
- `margin_bottom` -- defines the number of pixels for the chart's bottom margin (default is 50)
- `css_variables` -- a javascript object that defines the css variables (see below)

The `axis` property is central to the component's setup -- here is an example from the first demo:

```
axis: {
  x: {
    title: 'States',
    key: 'state',
    rotation: 0
  },
  y: {
    title: 'Population',
    tic_format: '~s',
    groups: [
      {key: 'Under 5 Years', format: '.3s', color: '#98abc5'},
      {key: '5 to 13 Years', format: '.3s', color: '#8a89a6'},
```

```

      {key: '14 to 17 Years', format: '.3s', color: '#7b6888'},
      {key: '18 to 24 Years', format: '.3s', color: '#6b486b'},
      {key: '25 to 44 Years', format: '.3s', color: '#a05d56'},
      {key: '45 to 64 Years', format: '.3s', color: '#d0743c'},
      {key: '65 Years and Over', format: '.3s', color: '#ff8c00'}
    ],
  }
}

```

The `x.key/y.groups.key` properties refers to the object keys in the data file. Note that for the `y` property we can have just one group or an array of groups with objects defining the data file key, output format, and fill color for the bar. For the formats property values see the [d3.formats](#) module.

Styling

The **css_variables** prop is a javascript object that contains any combination of css variable names as keys and associated values. The following list are the css variable names along with their default values for a quick styling of **bar-chart-comp-d3**:

```

{
  bar_chart_compD3_font_family: Verdana, serif,
  bar_chart_compD3_color: black,
  bar_chart_compD3_background_color: white,

  bar_chart_compD3_title_font_size: 1rem,

  bar_chart_compD3_axis_font_size: .8rem,
  bar_chart_compD3_axis_color: black,

  bar_chart_compD3_rect_stroke: black,
  bar_chart_compD3_rect_fill_opacity: 1.0,
  bar_chart_compD3_tooltip_fill: black
}

```

Demonstration

Two demonstrations of **bar-chart-comp-d3** are provided in the folders named `demo_1/dist`, and `demo_2/dist`. They can be viewed by hosting their respective `index.html` files. `demo_1` is a redo of [Grouped Stacked Bar Chart](#) from Mike Bostock's array of D3 examples. The demo folders contain a `package.json` file that can be used to setup dependencies for these demos and as a template for other applications using **bar-chart-comp-d3**. The following is the setup for this chart contained in the `App.vue` file where `axis` is a reference to the `axis` in our above example:

```

<bar-chart-comp-d3>
  title_1="Population of States by Age Group"
  title_2="(Mike Bostock's Block 3886208 Stacked Bar Chart)"
  :axis="axis"
  :stackBars="stackBars"
  :chart_data="chart_data"
  :css_variables="css_variables">
  <svg class="svg_chart_1" width="1200" height="800"></svg>
</bar-chart-comp-d3>

```

Note how the `bar-chart-comp-d3` tag wraps around the `<svg>` element. It is important that a class be assigned to the `svg` for **bar-chart-comp-d3** to locate the element. Among `bar-chart-comp-d3`'s attributes, it makes a reference to `chart_data` for the data attribute. `chart_data` is defined using an imported function called `read_csv` (from the [d3fetchmodule](#) module). Below is some of the code for reading the data.csv file:

```

read_data: function(){
  const convert = [
    {field: 'state',type: 'band'},
    {field: 'Under 5 Years',type: 'linear'},
    {field: '5 to 13 Years',type: 'linear'},
    {field: '14 to 17 Years',type: 'linear'},
    {field: '18 to 24 Years',type: 'linear'},
    {field: '25 to 44 Years',type: 'linear'},
    {field: '45 to 64 Years',type: 'linear'},
    {field: '65 Years and Over',type: 'linear'}
  ];
  const get_data = async () => {
    try {
      this.chart_data = await read_csv('data.csv',convert);
      //debug
      console.log(JSON.stringify(this.raw_data));
    }catch(e){
      console.log(e);
    }
  };
  get_data();
}

```

To view the demos, as a suggestion install [http-server](#) locally/globally via [npm](#) then enter the command `http-server` in the **bar-chart-comp-d3** `dist` directory. From a browser enter the url: `localhost:8080/`.