Plotly Tip #2: SVG & z-index

Thomas Barrasso
Jan 23, 2018 · 4 min read

About Plot.ly

Plot.ly is a company that has created a set of open-source libraries for creating graphs in languages like Python, R, and client-side JavaScript. It's built on top of the popular d3 and stack.gl libraries, renders charts in SVG and WebGL, and targets compatibility for most of its features as far back as IE 9.

Plotly Tips

I've found that many more questions arise than can be easily answered from the plotly.js documentation and forum. So I'm putting together a series of small examples and code snippets that have helped my solved some of my data visualization problems.

Background on SVG Stack Order

Often I've wanted to apply the `z-index` CSS property to SVG elements, however, according to the SVG Render Model, `z-index` is a *computed* integer and setting it in CSS will have no effect.

> *CSS specifies a property named '`z-index`'. The CSS rules that define the effect of the 'z-index' property were written specifically for the CSS box model, and* ***those rules do not make sense as they stand for most SVG elements*** *(most SVG elements do not participate in or establish a CSS box model layout).*

In short, the **easiest way to affect the stack order in SVG is to change the order elements appear in the document**. For example:

```
<svg xmlns="http://www.w3.org/2000/svg">
    <text>Below</text>
    <text>In Between</text>
    <text>Above</text>
</svg>
```

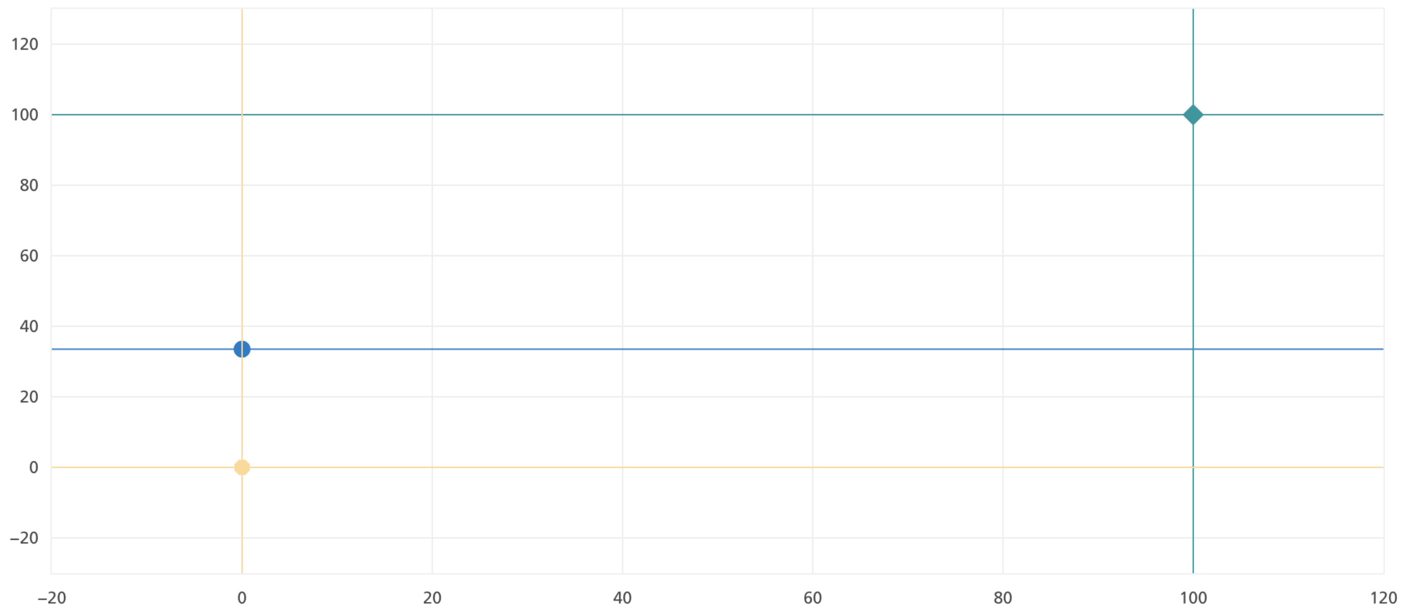
This example is oversimplified, but gets the point across that an element's stack order is determined by its position in the document. How does this affect Plotly? The following is the simplified output structure of a scatter plot rendered in Plotly 1.33.0:

```
<svg class="main-svg" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
    <defs>
        <g class="clips"></g>
        <g class="gradients"></g>
    </defs>
    <g class="bglayer">
        <rect class="bg"></rect>
    </g>
    <g class="draglayer">
        <g class="xy"></g>
    </g>
    <g class="layer-below">
        <g class="imagelayer"></g>
        <g class="shapelayer"></g>
    </g>
    <g class="cartesianlayer">
        <g class="subplot xy">
            <g class="layer-subplot">
                <g class="shapelayer"></g>
                <g class="imagelayer"></g>
            </g>
            <g class="gridlayer"></g>
            <g class="zerolinelayer"></g>
            <path class="xlines-below"></path>
            <path class="ylines-below"></path>
            <g class="overlines-below"></g>
            <g class="xaxislayer-below"></g>
            <g class="yaxislayer-below"></g>
            <g class="overaxes-below"></g>
            <g class="plot">
                <g class="imagelayer"></g>
                <g class="maplayer"></g>
                <g class="barlayer"></g>
                <g class="carpetlayer"></g>
                <g class="violinlayer"></g>
                <g class="boxlayer"></g>
                <g class="scatterlayer"></g>
            </g>
            <g class="overplot"></g>
            <path class="xlines-above crisp"></path>
            <path class="ylines-above crisp"></path>
            <g class="overlines-above"></g>
            <g class="xaxislayer-above"></g>
            <g class="yaxislayer-above"></g>
            <g class="overaxes-above"></g>
        </g>
    </g>
    <g class="polarlayer"></g>
    <g class="ternarylayer"></g>
    <g class="geolayer"></g>
    <g class="pielayer"></g>
    <g class="glimages"></g>
</svg>

<svg class="main-svg" xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">
    <defs>
        <g class="clips"></g>
    </defs>
    <g class="layer-above">
        <g class="imagelayer"></g>
        <g class="shapelayer"></g>
    </g>
    <g class="infolayer">
        <g class="g-gtitle"></g>
        <g class="rangeslider-container">
            <rect class="rangeslider-bg"></rect>
            <g class="rangeslider-rangeplot xy">
                <g class="layer-subplot">
                    <g class="shapelayer"></g>
                    <g class="imagelayer"></g>
                </g>
                <g class="gridlayer">
                    <g class="x"></g>
                    <g class="y"></g>
                </g>
                <g class="zerolinelayer"></g>
                <path class="xlines-below"></path>
                <path class="ylines-below"></path>
                <g class="overlines-below"></g>
                <g class="xaxislayer-below"></g>
                <g class="yaxislayer-below"></g>
                <g class="overaxes-below"></g>
                <g class="plot">
                    <g class="imagelayer"></g>
                    <g class="maplayer"></g>
                    <g class="barlayer"></g>
                    <g class="carpetlayer"></g>
                    <g class="violinlayer"></g>
                    <g class="boxlayer"></g>
                    <g class="scatterlayer"></g>
                </g>
                <g class="overplot"></g>
                <path class="xlines-above crisp"></path>
                <path class="ylines-above crisp"></path>
                <g class="overlines-above"></g>
                <g class="xaxislayer-above"></g>
                <g class="yaxislayer-above"></g>
                <g class="overaxes-above"></g>
            </g>
            <rect class="rangeslider-mask-min"></rect>
            <rect class="rangeslider-mask-max"></rect>
            <rect class="rangeslider-slidebox"></rect>
            <g class="rangeslider-grabber-min">
                <rect class="rangeslider-handle-min"></rect>
                <rect class="rangeslider-grabarea-min"></rect>
            </g>
```

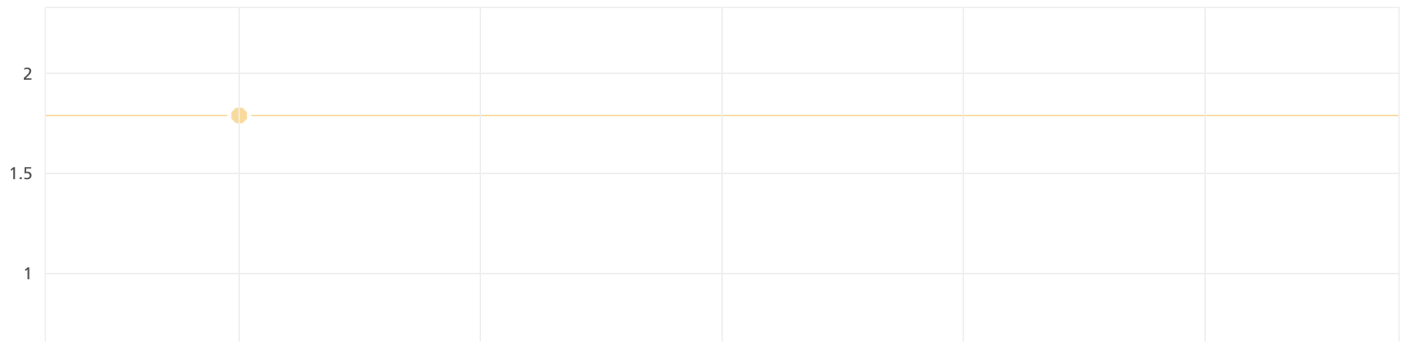
```
                <g class="rangeslider-grabber-max">
                    <rect class="rangeslider-handle-max"></rect>
                    <rect class="rangeslider-grabarea-max"></rect>
                </g>
            </g>
            <g class="g-xtitle"></g>
            <g class="g-ytitle"></g>
        </g>
        <g class="menulayer"></g>
        <g class="zoomlayer"></g>
        <g class="hoverlayer"></g>
    </svg>
```

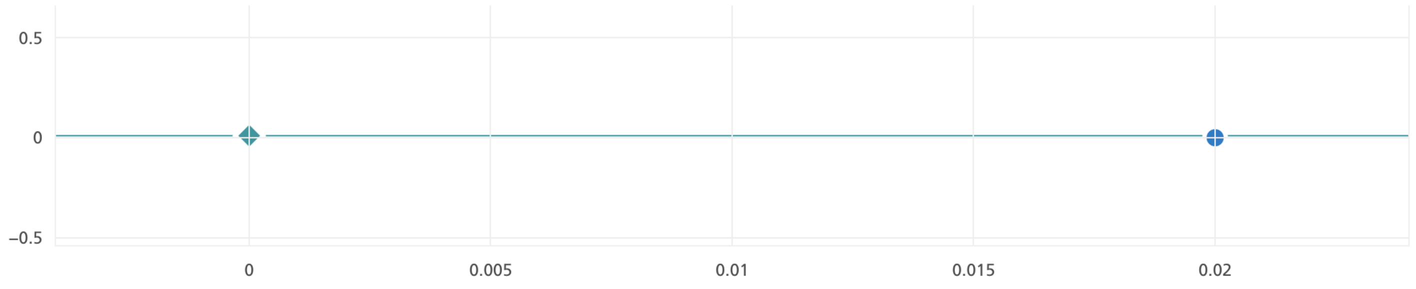
Issue #2



Reticle scatter plot with lines above points

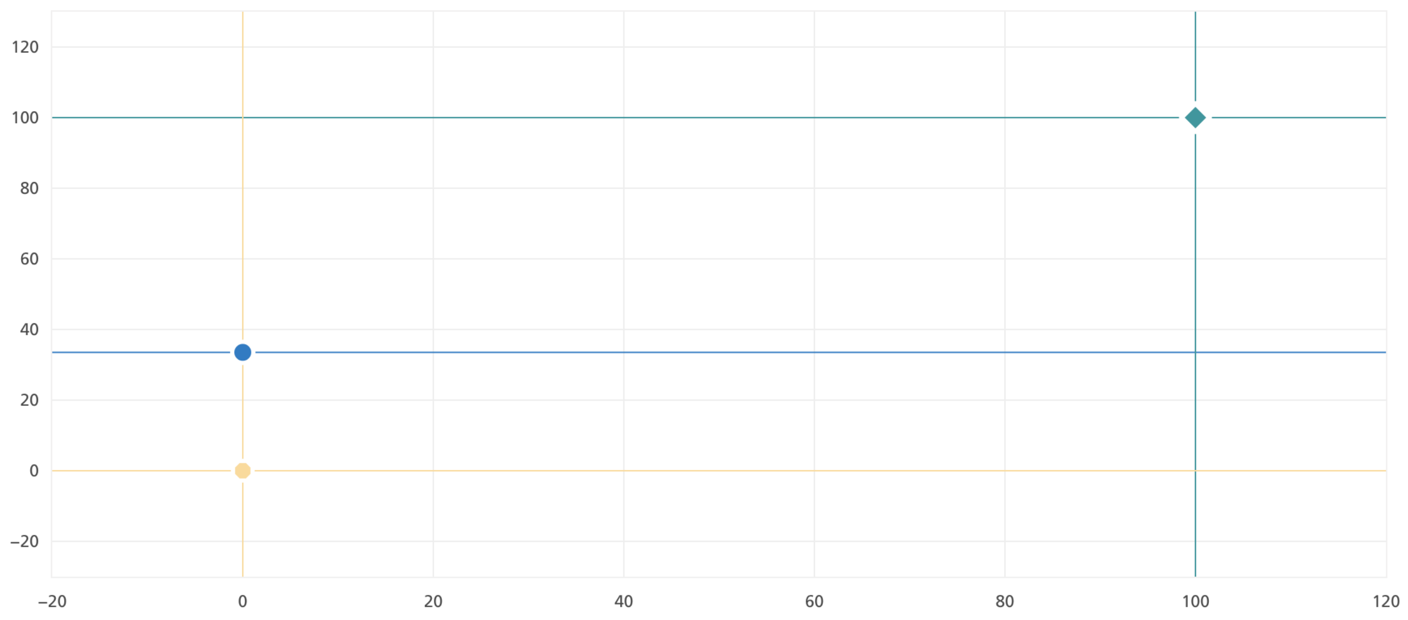
In the above example, I'm using Plotly Shapes to render lines along the x and y of each data point. As shown above, the styling looks bad because collinear points have lines above the data points. Yet we run into another problem if we apply the `layer : 'below'` property:





Grid lines above Plotly shapes

The `layer` API was suggested back in 2015 and is often too simple to be of use (it accepts values of `above` or `below`). One workaround I've found is simply to reorder elements after rendering a chart. It relies on class names that aren't guaranteed to stay the same, but is the most comprehensive way to position elements exactly where needed.



Manually rearranged Plotly elements

To achieve the above example, I set the line Shapes `layer : 'below'` and added a callback for the `plotly_afterplot` event. Using jQuery:

```
let gridlayer = $(chart).find('.gridlayer');
let layerbelow = $(chart).find('.layer-below');

if (gridlayer.length && layerbelow.length) {
    gridlayer.insertBefore(layerbelow);
}
```

This simple example moves `gridlayer` below Shapes, which is itself below the scatter plot data points (with 1px white stroke outlines). Now we've got the desired style with just a few lines of code! This is a much more robust solution than relying solely on `layer`, and this simple DOM manipulation incurs no noticeable performance penalty. If you're working with annotations, they are placed in `infolayer` above the plot. Since they lack a similar layer property, the above method is the only way to change their stack order (at the time of writing for v1.33.0).

SVG Data Visualization D3js Plotly JavaScript