Creating a Plot component

We'll use our knowledge of React.js and Plotly.js to create a plot component that will show the weather forecast.

WE'LL COVER THE FOLLOWING

Plot component

To show our plot on the webpage, we will create a component which we'll call `Plot` (what a surprise!) so lets start by rendering just a div:

```
class Plot extends React.Component {
  render() {
    return (
      <div id="plot"></div>
    );
  }
}
```

> As you can see, I've added a `div` with an ID of `plot` above. This is the DOM element we'll reference in our `Plotly.newPlot` call!

Now, the problem we have here is that if we called `Plotly.newPlot` in our `render` method, it would be called over and over again, possibly multiple times per second! That's not optimal, we really want to call it once when we get the data and leave it be afterwards – how can we do that?

Thankfully, React gives us a lifecycle method called `componentDidMount`. It is called once when the component was first rendered, and never afterwards; perfect for our needs! Let's create a `componentDidMount` method and call `Plotly.newPlot` in there and pass it the ID of our `div`, `plot`, as the first

`Plotly.newPlot` in there and pass it the ID of our `div`, `plot`, as the first argument:

```
class Plot extends React.Component {
  componentDidMount() {
    Plotly.newPlot('plot');
  }

  render() {
    return (
      <div id="plot"></div>
    );
  }
}
```

Passing data to the Plot component

That alone won't do much though, we need to give it data too!

Let's modify our Plot component to accept following three props:

1. `xData` - an array containing numbers on x-axis.
2. `yData` - an array containing nmbers on y-axis.
3. `type` - which defines the type of the chart

Let's update our Plot component to receive data as props and call `Plotly.newPlot` to create the chart:

```
class Plot extends React.Component {
  componentDidMount() {
    Plotly.newPlot('plot', [{
      x: this.props.xData,
      y: this.props.yData,
      type: this.props.type
    }], {
      margin: {
        t: 0, r: 0, l: 30
      },
      xaxis: {
        gridcolor: 'transparent'
      }
    }, {
      displayModeBar: false
    });
  }
```

```
  render() {
    return (
      <div id="plot"></div>
    );
  }
}
```

Let's use the above `Plot` component to create a chart of squares of first 5 numbers. Here's how we are going to render it.

```
ReactDOM.render(
  <Plot
    xData={[1,2,3,4,5]}
    yData={[1, 4, 9, 16, 25]}
    type='scatter' />,
  document.getElementById('content')
);
```

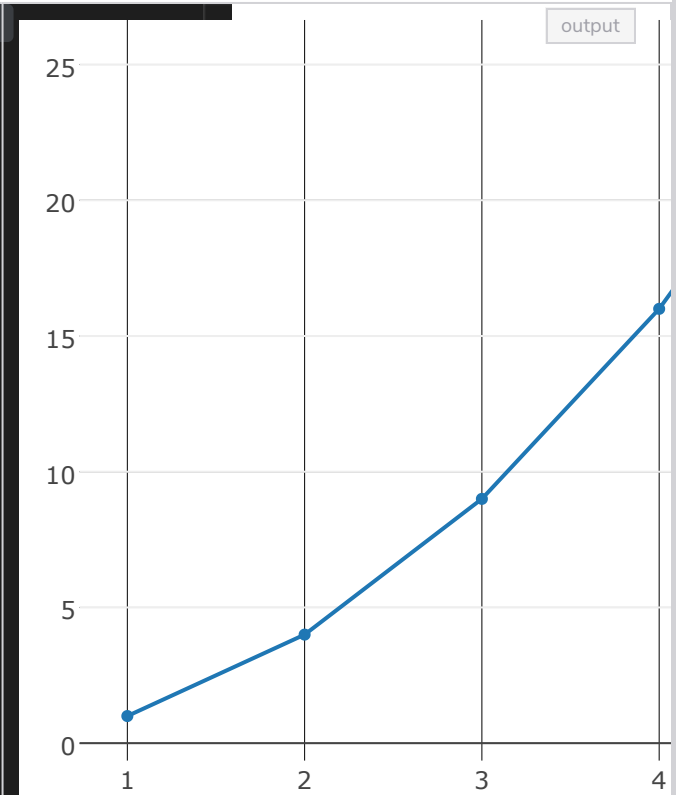
and finally, here's our working example:

HTML JavaScript Output

```
1  class Plot extends React.C
2    componentDidMount() {
3      Plotly.newPlot('plot', [{
4        x: this.props.xData,
5        y: this.props.yData,
6        type: this.props.type
7      }], {
8        margin: {
9          t: 0, r: 0, l: 30
10       },
11       xaxis: {
12         gridcolor: 'transparent'
13       }
14     }, {
15       displayModeBar: false
16     });
17   }
18
19   render() {
20     return (
21       <div id="plot"></div>
22     );
23   }
24 }
25
26 ReactDOM.render(
27   <Plot
```

javascript

output



```
28      xData={[1,2,3,4,5]}
29      yData={[1, 4, 9, 16, 25]}
30      type='scatter' />,
31    document.getElementById('content')
```

We now have a `Plot` component that can take weather data as input and draw the forecast graph. We're getting closer to our forecast graph.