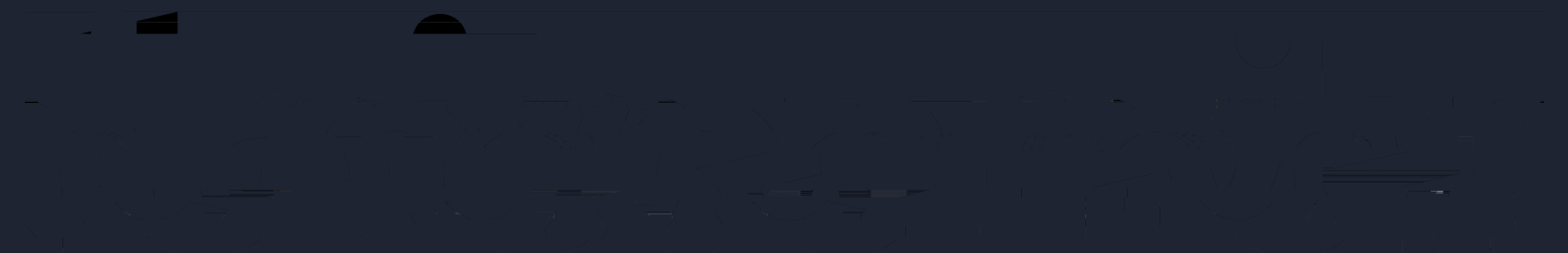


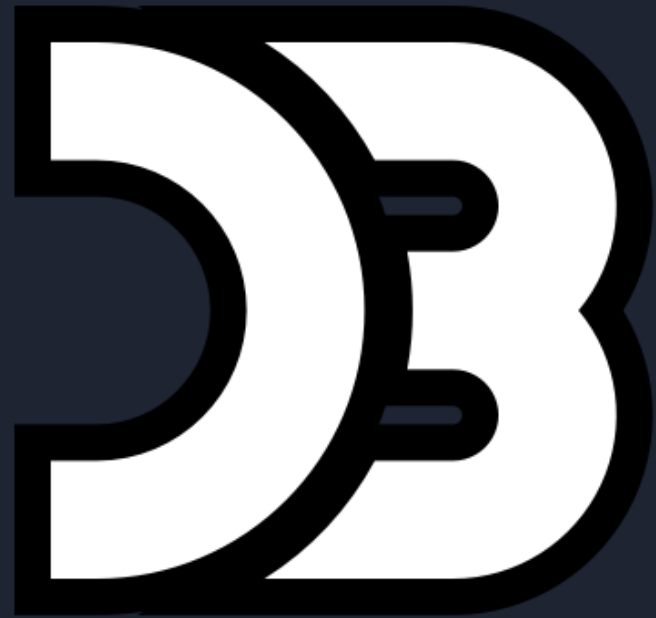
React, D3 and the Dataviz Ecosystem





Online

- [@golodhros](#)
- [Eventbrite Engineering Blog](#)



Next up

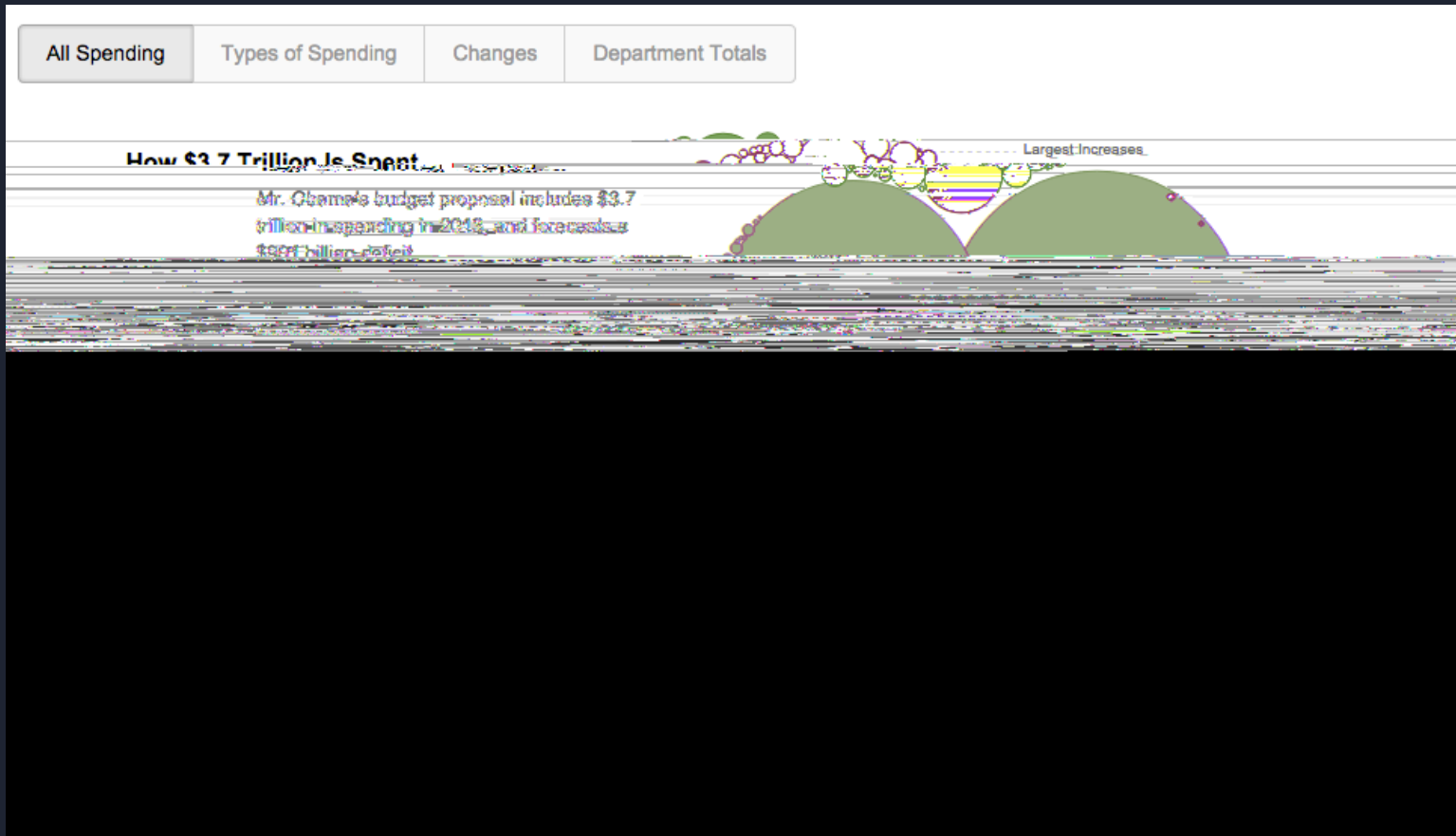
- D3
- React
- React + D3
- Choosing an approach

What is D3.js?

How does it work?

- Loads data
- Binds data to elements
- Transforms those elements
- Transitions between states

D3 Demo



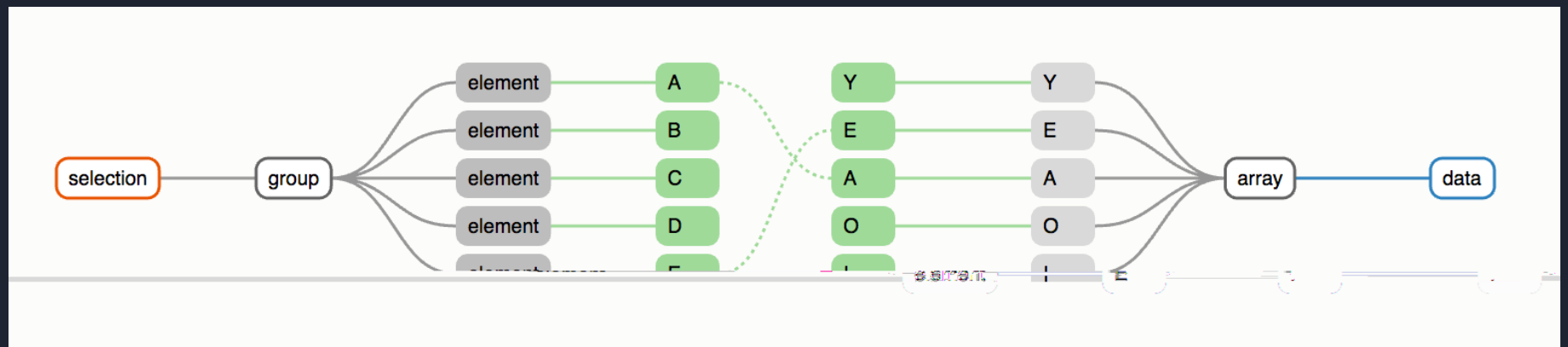
D3 v4 Update

- More modular
- Improved API
- Breaking changes
- Highly adopted

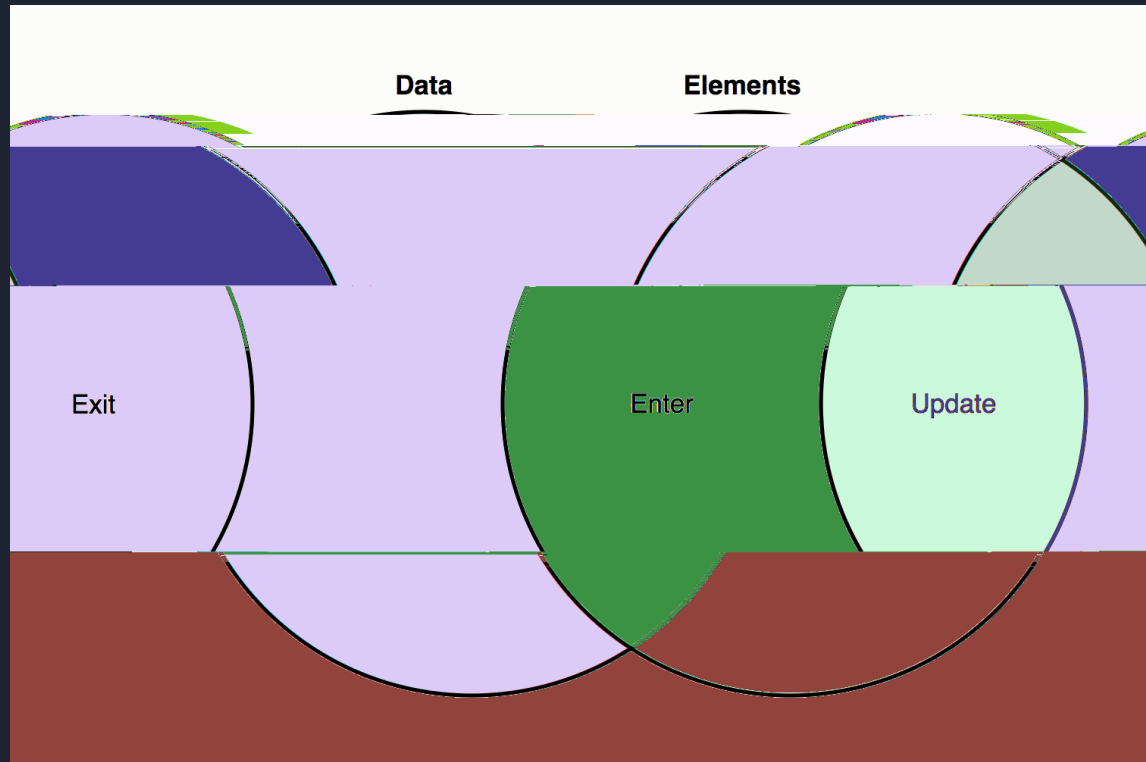
D3 DATA JOINS

**Transforms the DOM by
selecting elements and
joining to data**

Data Join



Update, Enter and Exit Pattern



Resources

D3 PATTERNS AND BEST PRACTICES

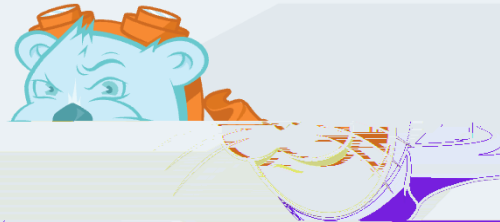
D3 Testing

D3 is hard

D3 Testing

D3 is hard

D3 LIBRARIES



Plottable

Flexible, interactive charts for the web.

DOWNLOAD (LATEST)

VIEW ON GITHUB

will on top of D3. Plottable gives you a set of flexible, pre-made components that you can use to create interactive charts.

<http://plottablejs.org/>

PLAYGROUND | DOCUMENTATION | ABOUT | COMMUNITY | FEATURES | DOWNLOAD | API | EXAMPLES | GITHUB

billboard.js

Re-usable, easy interface JavaScript chart library, based on D3 v4+.

★ Star 1,996

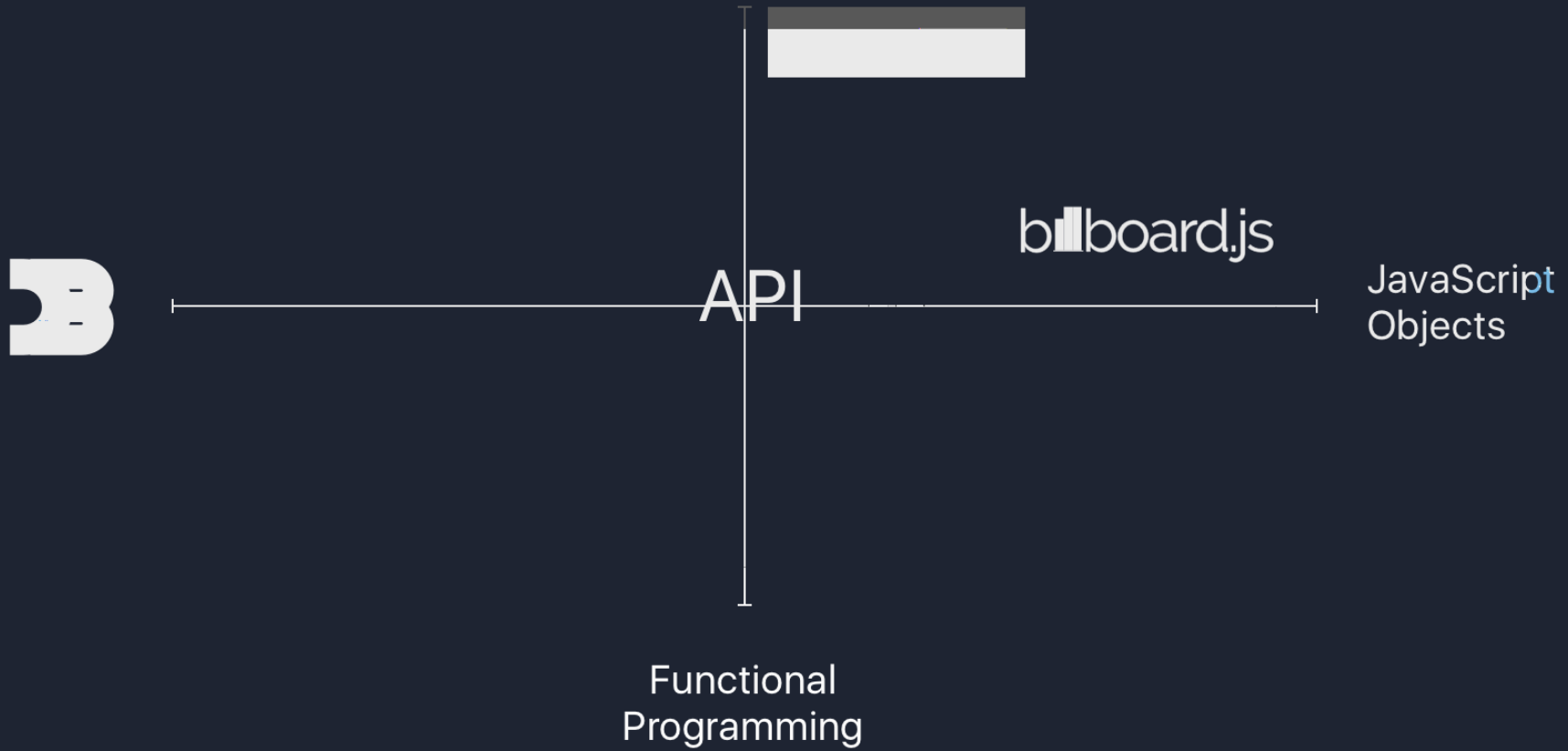
Category	data1	data2
0	130	20
1	100	200
2	140	100
3	200	400
4	150	150
5	80	280

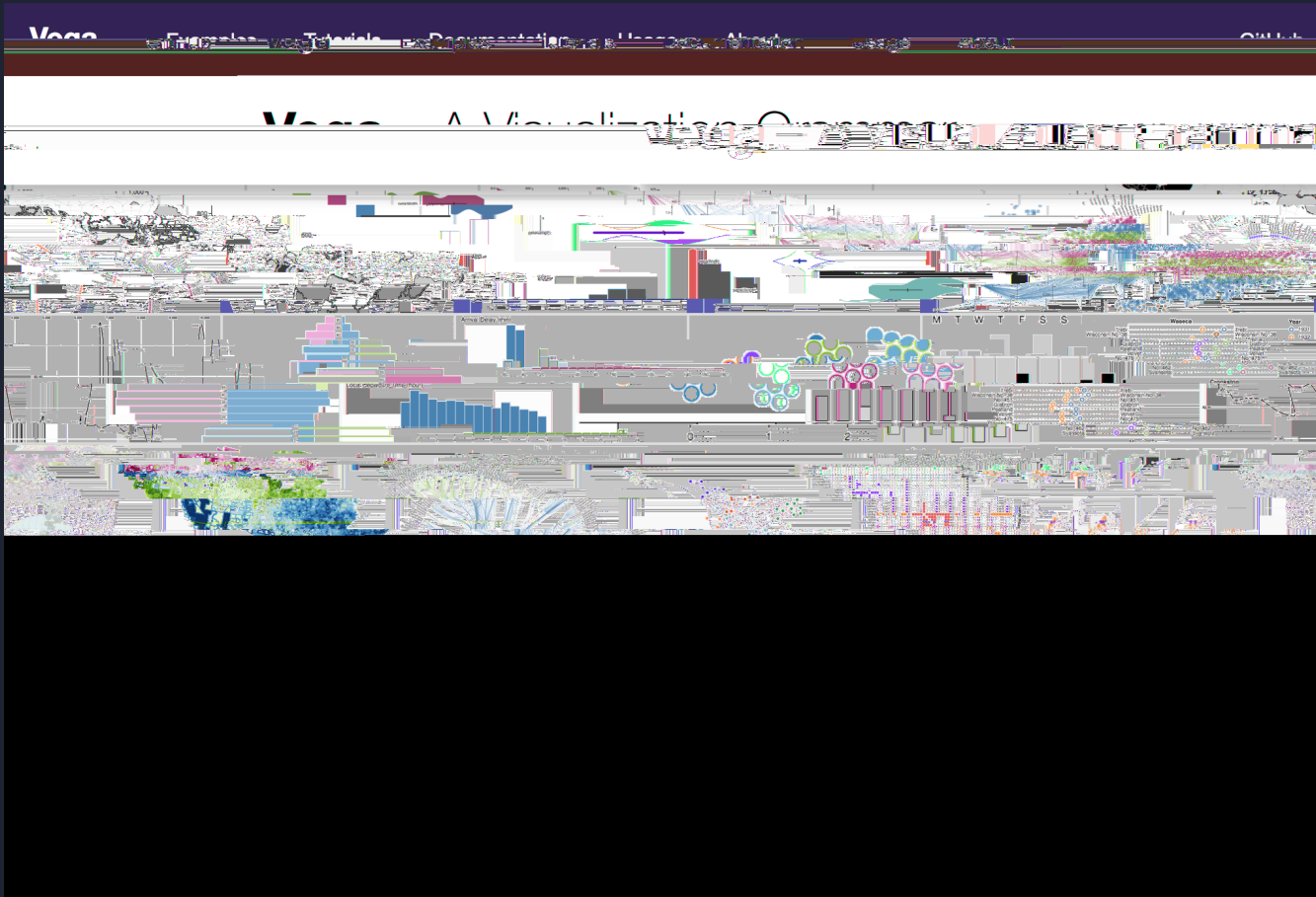
ed

Get Started

<https://naver.github.io/billboard.js/>

Object Oriented Programming





<https://vega.github.io/vega/>

Object Oriented Programming

Plc

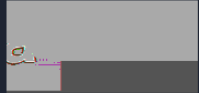
board is

JavaScript
Objects

API

API

Functional
Programming



D3FC

DOCUMENTATION

API

EXAMPLES

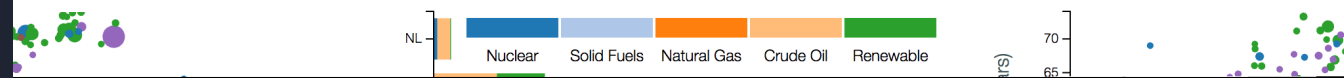
GITHUB

*Components for building interactive charts
with D3.js*

Health of Nations

2013 Energy Production

The Wealth & Health of Nations



<https://d3fc.io/>

Object Oriented Programming

Plottable

board.is

JavaScript

JavaScript

Objects

js

Functional

Functional
Programming



API

API



Britecharts API Demos Global Github

Search

Britecharts

Key Features

Usage

API

Installation

Readman

See Also

Acknowledgments

License

Britecharts Library based on **D3.js v4** that allows easy and reusable components that can be composed together creating amazing visualizations.

build passing npm package 1.4.6 bower package 1.5.1

Britecharts components have been written in ES2015 with a Test-Driven methodology so they are fully tested, and we are committed to keep them that way.

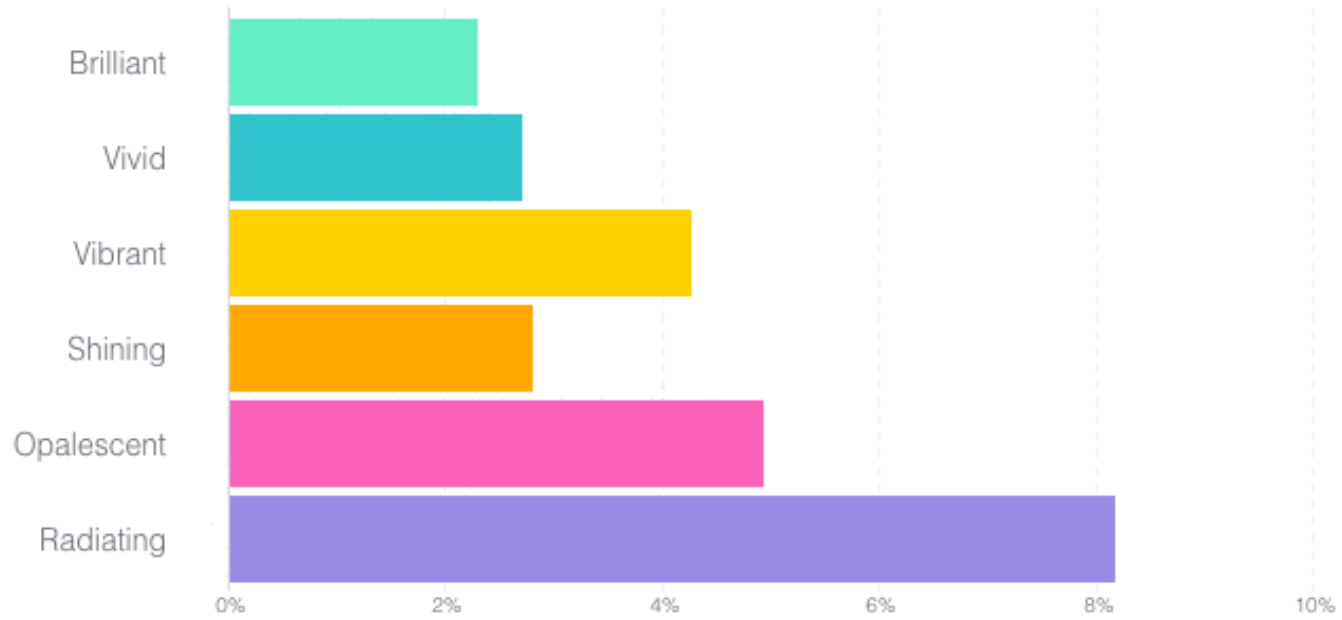
Key Features

The main characteristics of this library are:

- Reusability

Eventbrite Copyright

<http://eventbrite.github.io/britecharts>



Object Oriented Programming





What is ReactJS?

- User Interface library
- Component-based
- Fast and efficient

Reconciliation and Rendering

- The "Renderer" and the "Reconciler" are separated
- Render in Web, Native Apps and Virtual Reality

React Dynamic Child Components

- Unique 'key' to each child component rendered
- More efficient rendering
- Similar to D3's data joins

Reconciliation - React Docs



Commonalities

- Help us with the DOM
- Love pure functions

Challenges

- D3 creates and transforms the DOM
- React as well, and keeps track of it
- Not meant to work together

REACT + D3 APPROACHES

D3 within React

- React renders root svg element
- D3 creates chart in *componentDidUpdate*
- Block chart updates with *shouldComponentUpdate*

D3 within React



D3 within React

PROS/CONS

- Work

React Faux DOM

*It's a way to use existing D3 tooling
but render it efficiently through
React with the React ethos*

D3 within React the right way

React Faux DOM



React Faux DOM

PROS/CONS

- Use all D3 APIs
- Good integration with already built D3
- Server Side Rendering
- **X/** Need to use React Animations
- **X** Less performant
- **X** Limited to small/medium size dataviz

Lifecycle Methods Mapping

- Lightweight React Component Wrapper
- D3-only file with create, update and unmount methods

Integrating D3.js visualizations in a React app

Lifecycle Methods Mapping



Lifecycle Methods Mapping - D3Line



Lifecycle Methods Mapping

PROS/CONS

- Easy to integrate D3.js code
- Flexible, could encapsulate any chart
- ~~x/~~ Adds another file

D3 for the Math, React for the DOM

- D3 is used for math and formats
- React rules the DOM

React-D3 Layouts

D3 sub-modules: Non-DOM Related

- Arrays (d3-array)
- Chords (d3-chord)
- Collections (d3-collection)
- Colors (d3-color)
- Dispatches (d3-dispatch)
- Easings (d3-ease)
- Forces (d3-force)
- Number Formats (d3-format)
- Hierarchies (d3-hierarchy)

D3 sub-modules: DOM Related



D3 for the Math, React for the DOM



D3 for the Math, React for the DOM

PROS/CONS

- Consistent with the React way
- **X** A lot of work upfront
- **X** D3 Reimplementation of certain parts
- **X** Limited to SVG rendering

React Faux DOM

D3 within the React

React Lifecycle

Methods

D3 for the Math,
React for the DOM

Methods

Reference

REACT + D3 LIBRARIES

Ecosystem

- Non-maintained libraries
- Not easy to keep up with D3 and React
- D3 v4 cleaned up the field

React is components for data visualization modular charting *and*

Q5

4.0 5.0 1.0 2.0 3.0 4.0 5.0

```
$ victory-cli data.json script.js --print
```

25

20

1.5

1.0

```
npm install victory
```

Linechart Code

```
class LineChartVictory extends Component {
  render() {
    return (
      <VictoryChart
        height={300}
        width={500}
        containerComponent={
          <VictoryVoronoiContainer dimension="x"
            labels={(d) => `uv: ${d.uv}`}
            labelComponent={<VictoryTooltip cornerRadius={0} flvoutStyle={{fill: 'white'}}/>}
          />
        }
        theme={VictoryTheme.material}
      >
        <VictoryLine
          labelComponent={<VictoryTooltip/>}
          data={data}
          x="name"
          y="uv"
          theme={VictoryTheme.grayscale}
        />
        <VictoryLine
          labelComponent={<VictoryTooltip/>}
          data={data}
          x="name"
          y="pv"
          theme={VictoryTheme.grayscale}
        />
        <VictoryAxis
          scale="time"
          standalone={false}
        />
      </VictoryChart>
    );
  }
}
```

VictoryJS

- Easy to get started
- Zoom and Voronoi
- React Native option

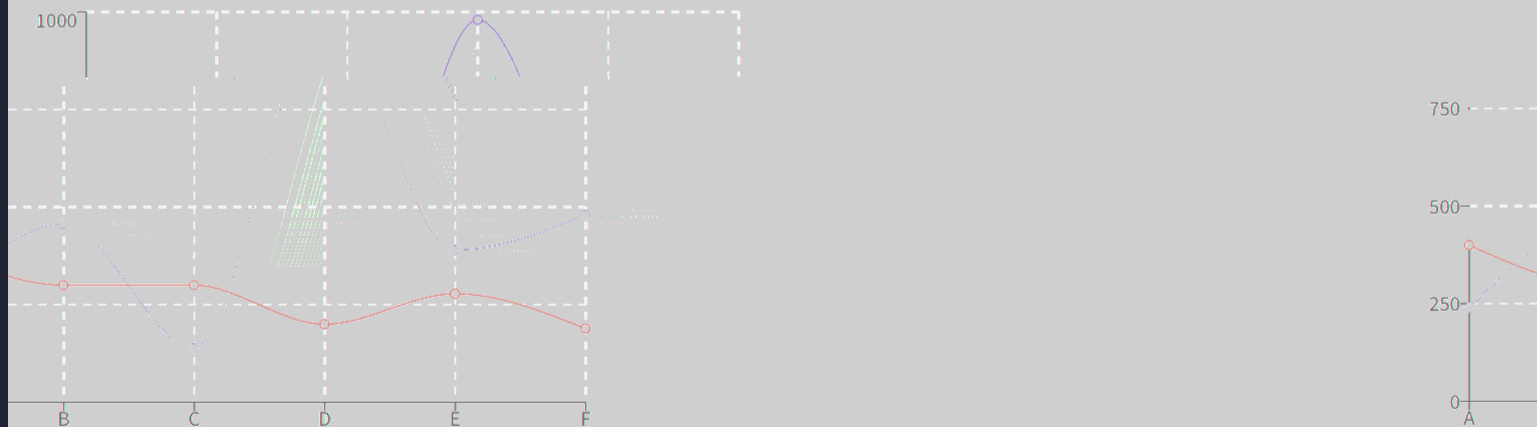
Recharts

composable charting library built on React components

A d

⚡ Install v0.22.1

★ Star 5,493



```
width={500} height={300} data={data}>  
  dataKey="name" />
```

```
  <YAxis />  
  <CartesianGrid stroke="#eee" strokeDasharray="5 5" />  
  <Line type="monotone" dataKey="uv" stroke="#8884d8" />  
  <Line type="monotone" dataKey="nv" stroke="#82ca9d" />
```

```
</LineChart>
```

```
<LineChart  
  <XAxis d
```

Linechart Code

```
class SimpleLineChart extends Component {
  render() {
    return (
      <LineChart
        width={500}
        height={300}
        data={data}
        margin={{top: 5, right: 30, left: 20, bottom: 5}}
        dataKey="uv"
        type="monotone"
        stroke="#8884d8"
        activeDot={{r: 8}}
      />
      <Line
        dataKey="uv"
        stroke="#82ca9d"
      />
    )
  }
}
```

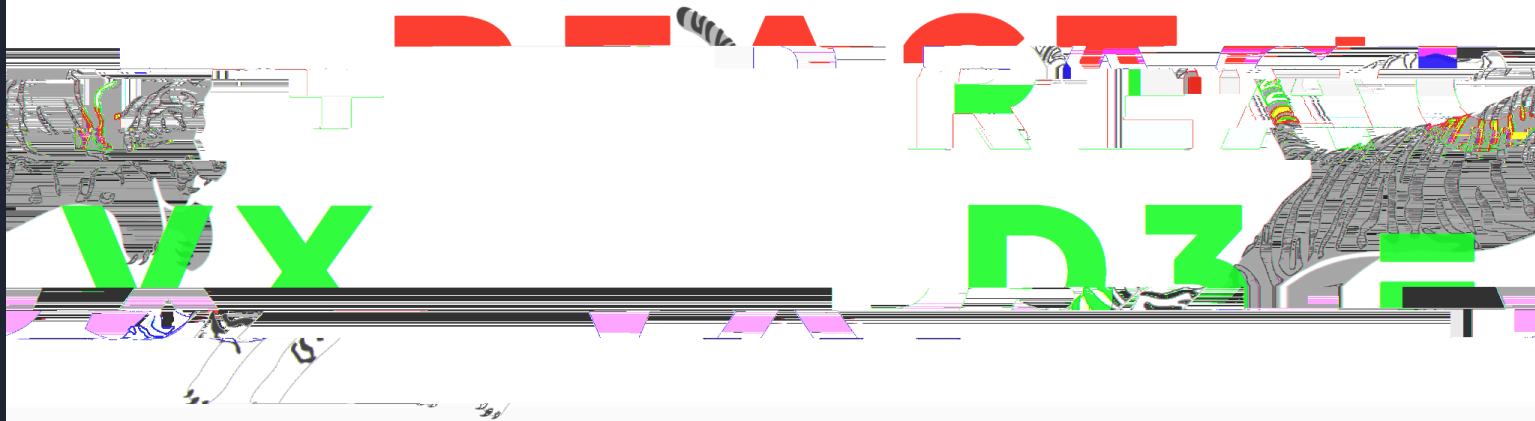
Recharts

- Really well tested
- Charts plus legend, tooltip and brush
- Great docs



Linechart Code

```
class LineChartNivo extends Component {
  render() {
    return (
      <Line
        data={[]}
        width={600}
        height={300}
        minY="auto"
        stacked={true}
        axisBottom={{
          axisLeft: {}
        }}
        dotSize={10}
        dotColor="inherit:darker(0.3)"
        dotBorderWidth={2}
        dotBorderColor="inherit:darker(0.3)"
        enableDotLabel={true}
        dotLabel="y"
        dotLabelYOffset={-12}
        animate={true}
        motionStiffness={90}
        motionDamping={15}
        legends={[]}
      />
    );
  }
}
```

[VIEW ON GITHUB](#)

`vx` is collection of reusable low-level visualization components. `vx` combines the power of `d3` to generate your visualization with the benefits of `react` for updating the DOM.

Linechart Code

```
class LineChartVX extends Component {
  render () {
    let {width, height, margin} = this.props;
    // bounds
    const vMax = width * margin.left * margin
    .top - margin.bottom;
    const xMax = width * margin.left * margin
    .top - margin.bottom;
    const yMax = height - margin
    // scales
    const xScale = scaleTime({
    });
    const yScale = scaleLinear({
    });
    return (
      <svg
        width={width}
        height={height}
      >
        <rect
          x={0}
          y={0}
          width={width}
          height={height}
          fill="white"
          rx={14}
        />
        <Group top={margin.t
          <linePath
            data={data}
            xScale={xScale}
            yScale={yScale}
            x={x}
            y={y}
            stroke="#32deaa"
            strokeWidth={2}
          />
        </Group>
      </svg>
    );
  }
}
```

VX

- Similar to a "D3.js for React"
- Flexible about the animation library

Components

- Bar
- Donut
- Legend
- Line

Tooltip
About

Shiny	84,000
300,000	30
276,000	30

Other	11
Los Angeles	1
Oakland	0

60%

- Blazing
- Dazzling

Linechart Code

```
class LineChartBritechartsReact extends Component {
  render() {
    const margin = {
      top: 60,
      right: 50,
      bottom: 60,
      left: 70,
    };

    return (
      <Tooltip
        data={lineSet()}
        title="Tooltip Title"
        render={(props) => (
          <Line>
            width={600}
            height={300}
            isAnimated={true}
            margin={margin}
            curve=basis
            {...props}
          />
        )}
      />
    );
  }
}
```

Britecharts React

- Helps to render D3 in React components
- No Server-side Rendering



Demo: <https://golodhros.github.io/talk-react-d3/>
Code: <https://github.com/Golodhros/talk-react-d3>

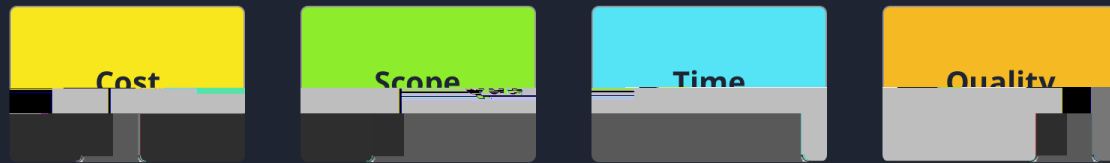
React D3 Libraries Info - Draft

Github URL	Active	Prod Ready	Tests	D3 V4	Github Stars	Name
https://github.com/recharts/recharts	Yes	Yes	Great - 93%	Yes	5403	recharts
https://github.com/uber/react-vis	Yes	Yes	OK	Yes	1824	react-vis
https://github.com/behoff/arc	Yes	No	No	Yes	99	arc
https://github.com/artvov/trivak/d3-explorer	-	-	-	No	7	d3-explorer
https://github.com/ReactD3/react-d3	No	Yes	OK	Yes	~200	React D3
https://github.com/yang-wei/rd3	No	-	OK	No	1566	react-d3
https://github.com/codesuki/react-d3-components	Barely	-	No	No	1191	react-d3-components
https://github.com/react-d3-library/react-d3-library	Barely	-	No	No	731	React D3 Library
https://github.com/bgrsqared/d3-react-squared	Barely	-	No	Yes	174	d3-react-squared
					12	

<http://tinyurl.com/d3-react-lib-comparative>

CHOOSING AN APPROACH

Choosing criteria





Cost

Scope

Time

Quality

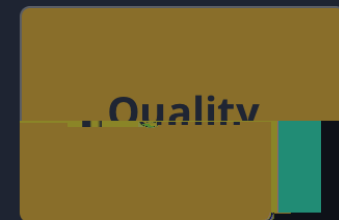
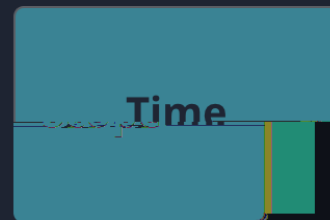
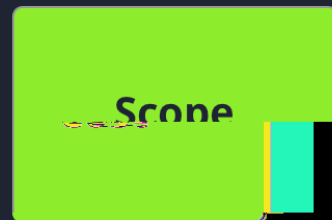
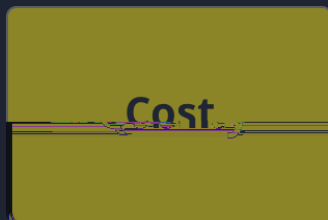
Cost

Scope

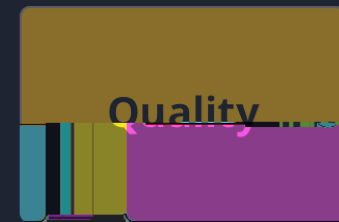
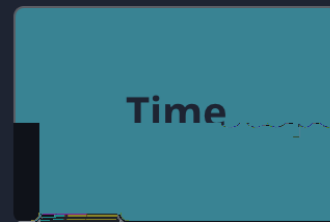
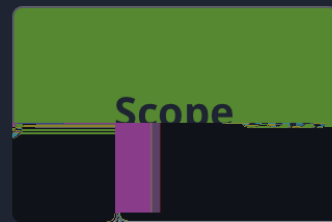
Time

Quality

- Long term investment?
- Needs to be done ASAP?



- Basic charts?
- One-off work?
- Highly customized?



- Limited budget?
- Need training?

React

Can't

React

D3 and React can work together

RESOURCES

D3 Resources

- [Mike Bostock](#)
- [D3 Reference](#)
- [d3 slack channel](#)

Data-Joins Resources

React-D3 Resources

React D3 Libraries Info - Draft

File Edit View Insert Format Data Tools Add-ons Help Last edit was made 9 minutes ago by Marcos Jolasias Valle

ID	V4	GitHub Stars	Name	GitHub URL	Active	Prod Ready	Tests	
	Great	Yes	4195	3 victory	https://github.com/Formidablelabs/victory	Yes	Yes	
	Yes	OK	Yes	1824	4 react-vis	https://github.com/uber/react-vis	Yes	
	No	No	Yes	99	5 vx	https://github.com/hshoff/vx	Yes	
	No	Yes	OK	Yes	200	7 React D3	https://github.com/react-d3	
	No	Yes	OK	1588	6 react-d3	https://github.com/yan-wu/react-d3		
	nte	Barely	No	No	1191	8 react-d3-components	https://github.com/codesuki/react-d3-components	
	react-d3-libraries/react-d3-libraries	Barely	No	No	724	10 React D3 Libraries	https://github.com/react-d3-libraries/react-d3-libraries	

<http://tinyurl.com/d3-react-lib-comparative>