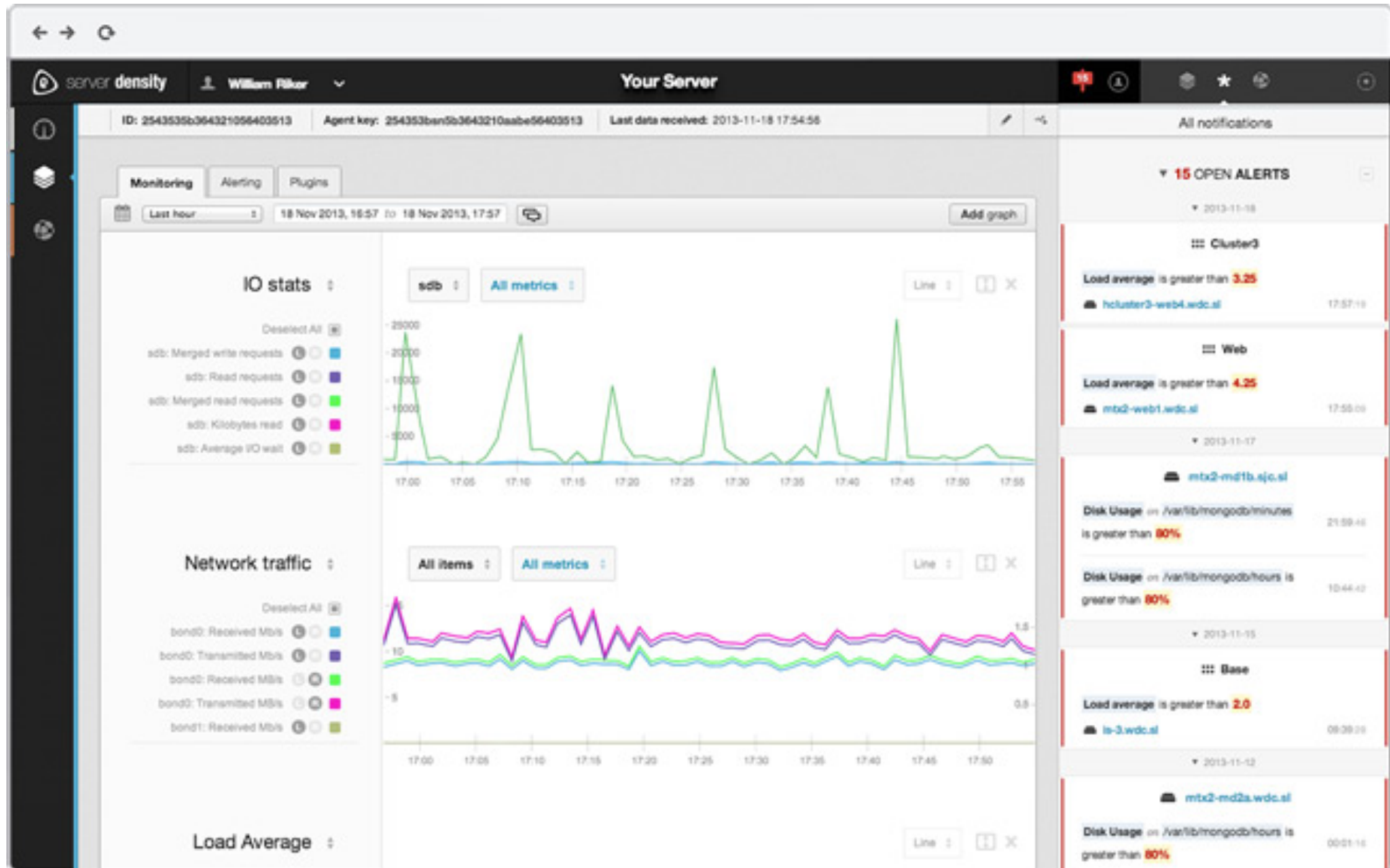


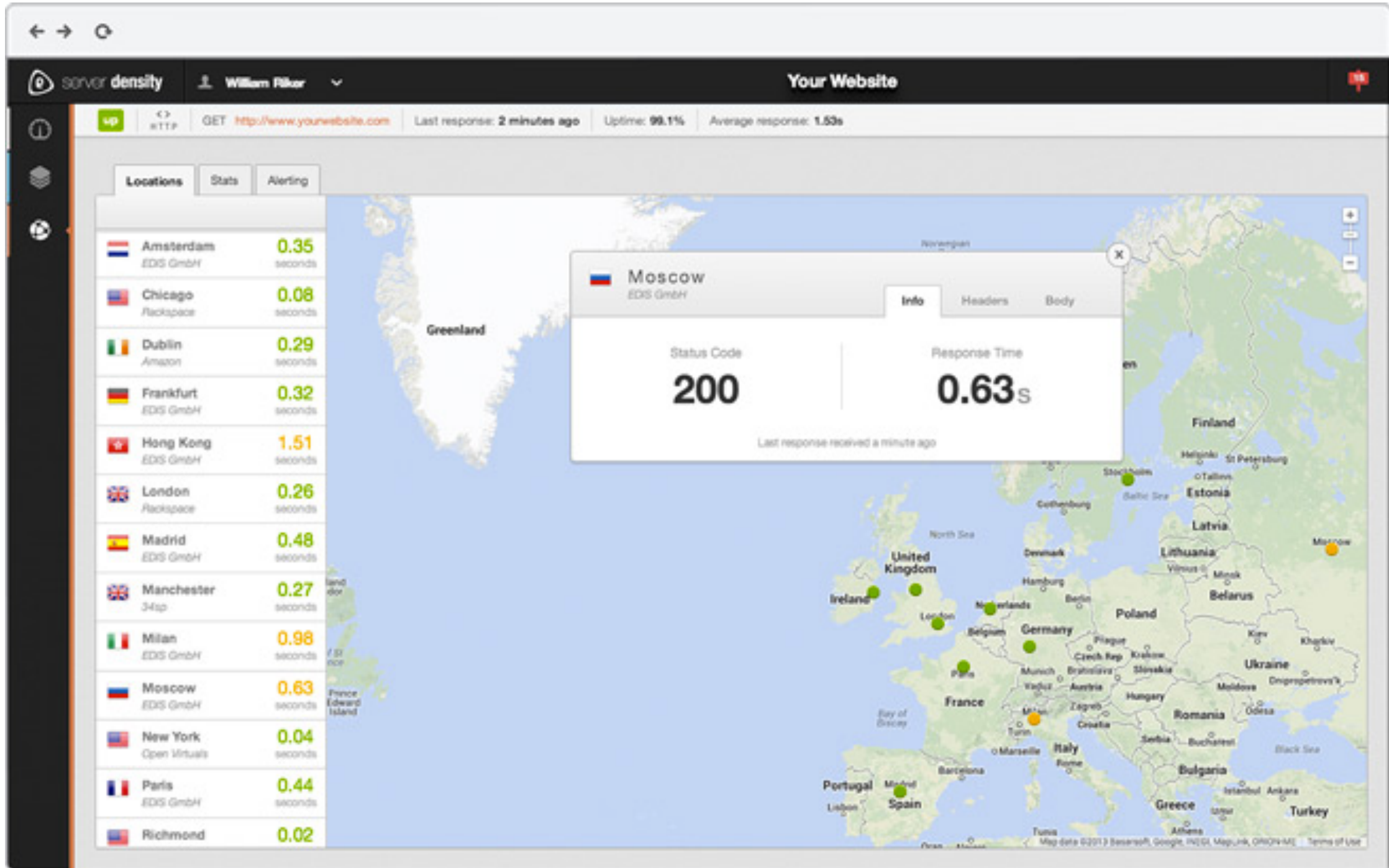


server **density**

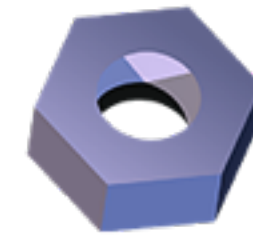
@serverdensity
@byrichardpowell



Monitor your infrastructure



Monitor your services



30 TB of data per month
2 billion documents a day

<http://bit.ly/30TB-pm>

A Story

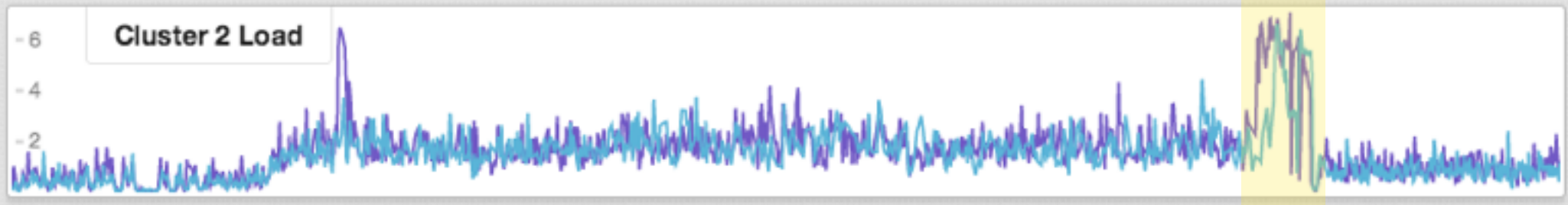
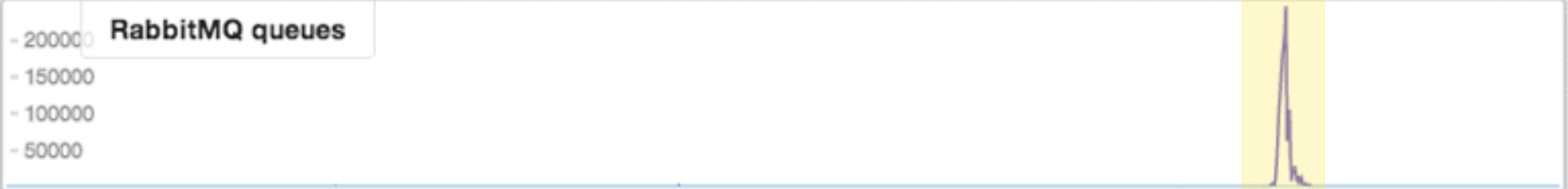
A close-up portrait of a man with dark hair and a slight smile, wearing a grey cardigan over a blue shirt. The background is a blurred outdoor setting with a brick wall and bare tree branches. The Amazon logo is visible in the bottom left corner.

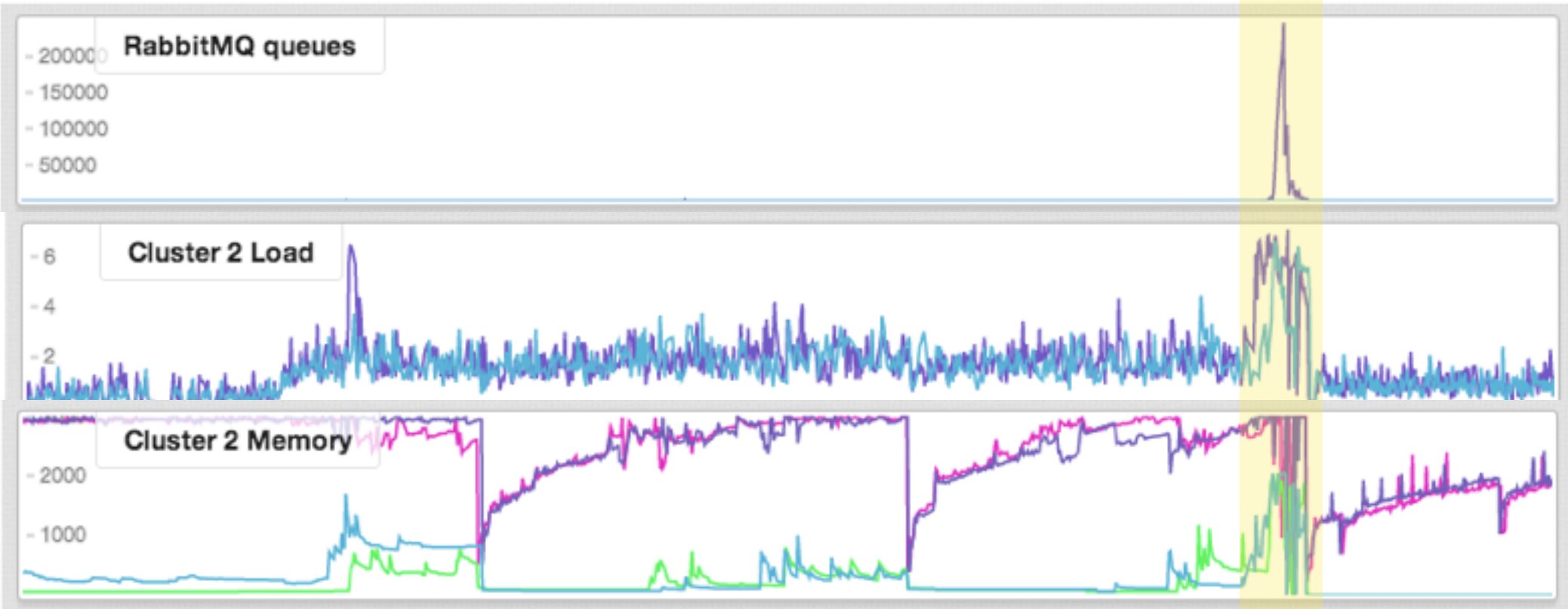
Pedro

Our Ops Engineer

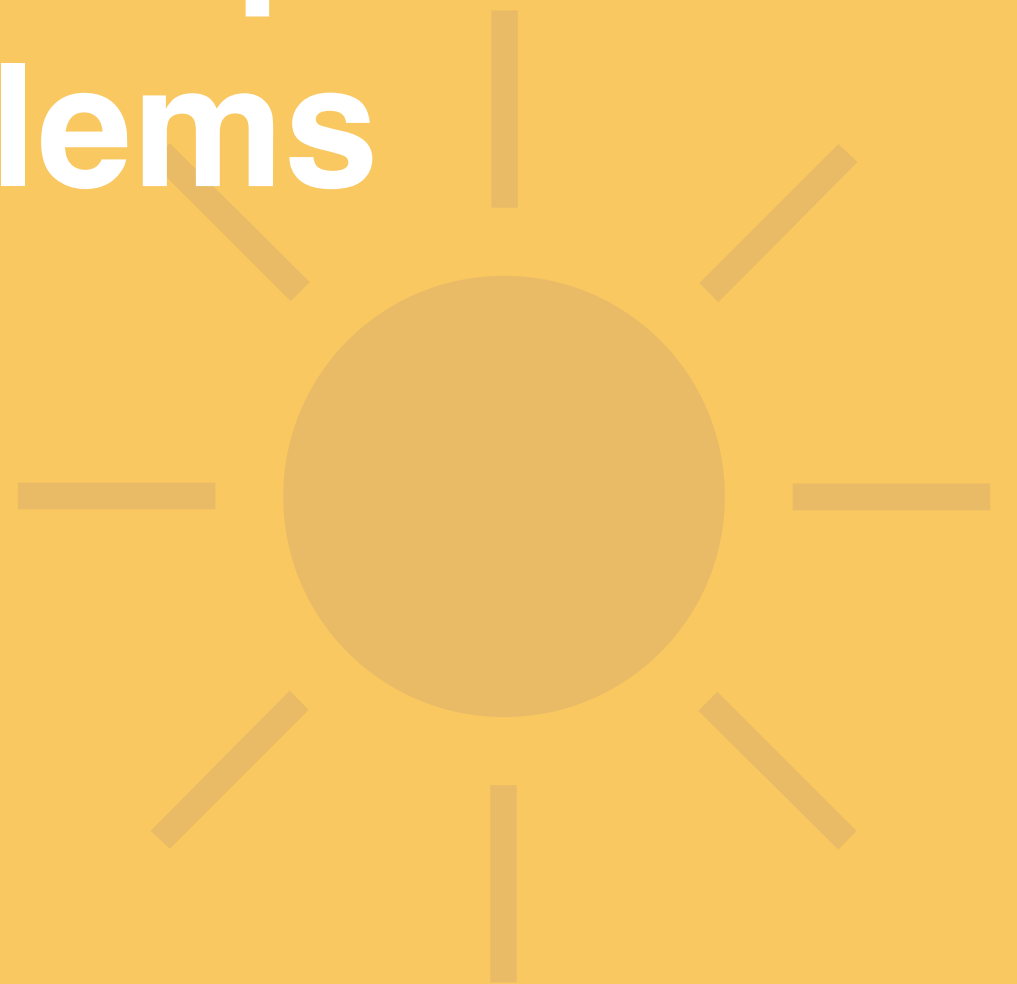








Time Series Graphs Solve Problems



- 1. D3**
- 2. Rickshaw**
- 3. Rickshaw ++**
- 4. The BIG Lesson!**



D3

Getting Started

```
data = [5, 10, 15, 20];

// Array Maths
d3.extent(data); // [5,20]
d3.mean(data); // 12.5

// Useful Stuff
d3.bisect(data, 8); // 1
d3.shuffle(data); // Random order
d3.merge([
  [5, 10],
  [15, 20]
]) // Deep merge

// Creating a Set
set = d3.set(data); // Creates a Set

set.has(25) // false
set.add(5) // 5 (Set remains unchanged)
```

```
person = { name: "Richard", age: 28 };

// Objects
d3.keys( person ); // ["name", "age"]
d3.values( person ); // ["Richard", 27]

// Maps
map = d3.map(person)

map.has("location") // false
map.set("location", "Berlin") // "Berlin"
map.size() // 3
```

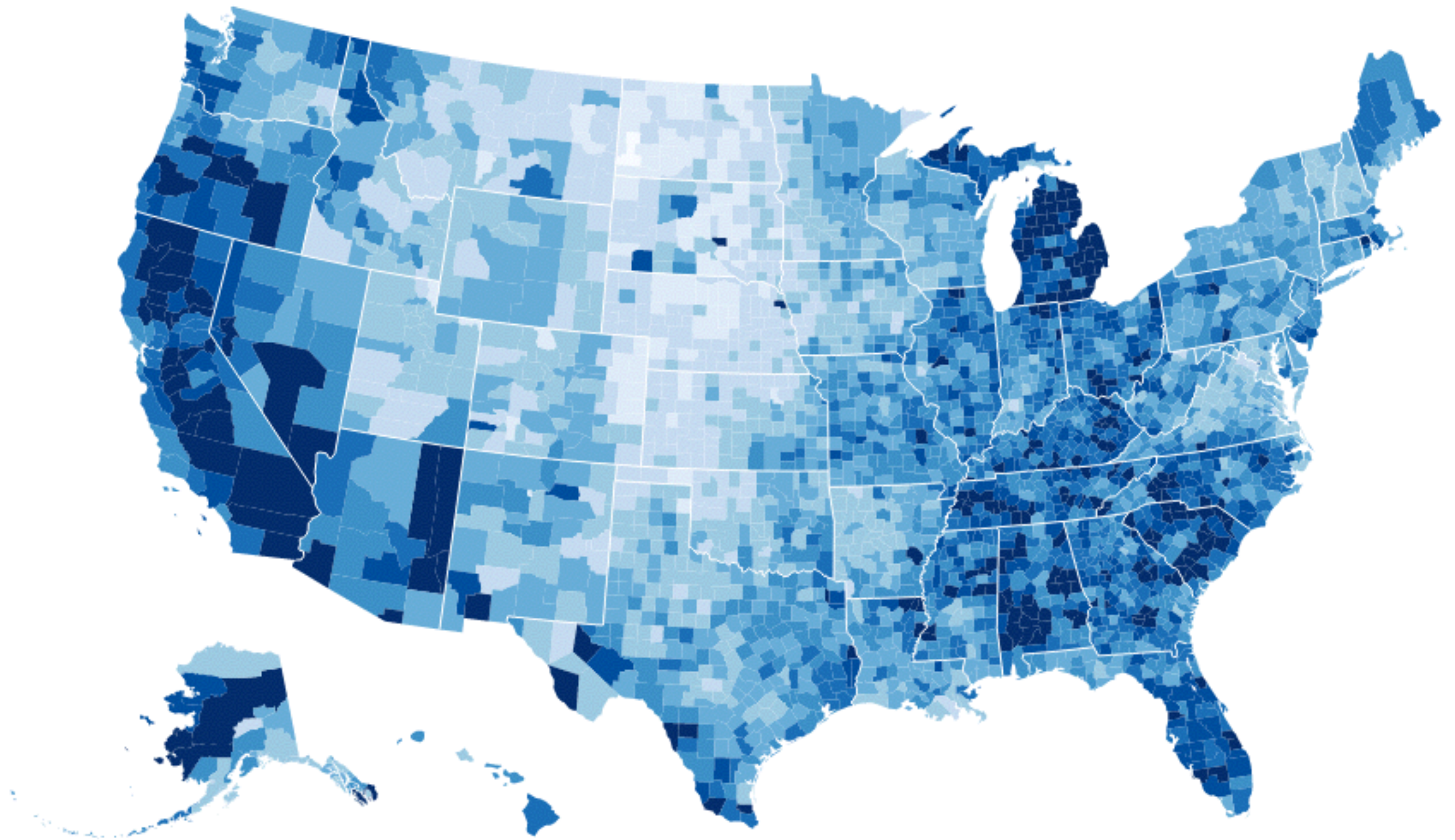
```
// Create a Linear Scale
scale = d3.scale.linear().domain([0,10]);

scale(5); // 0.5
scale(20); // 2

// Create a scale for colors
color = d3.scale.linear().domain([0,5,10]);
color.range(["green", "yellow", "red"]);

color(0); // "#008000" (green)
color(5); // "#ffff00" (yellow)
color(8); // "#ff6600" (orange)
color(10); // "#ff0000" (red)
```

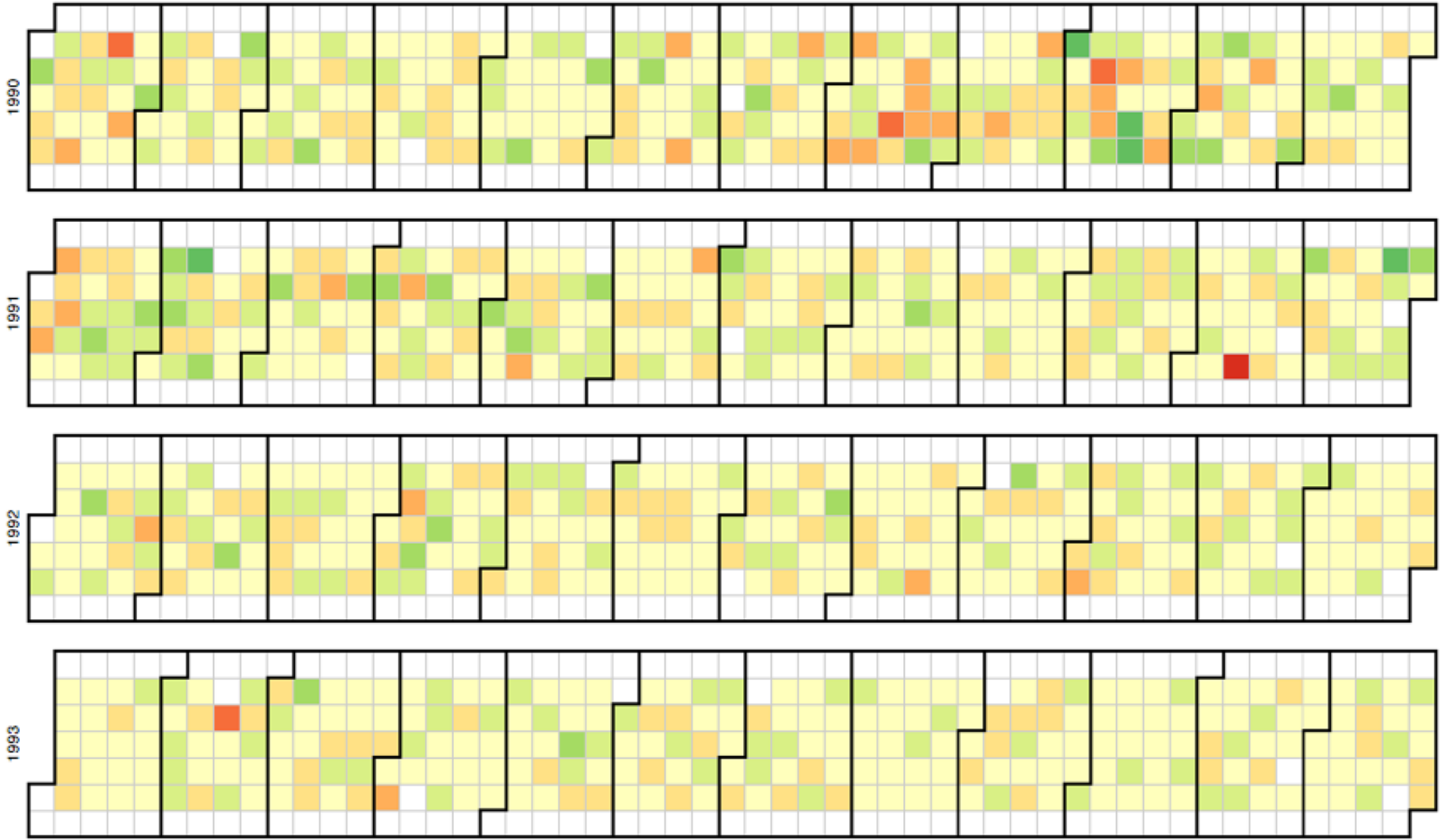
Data Visualisation is in D3's DNA



bl.ocks.org/mbostock/4060606

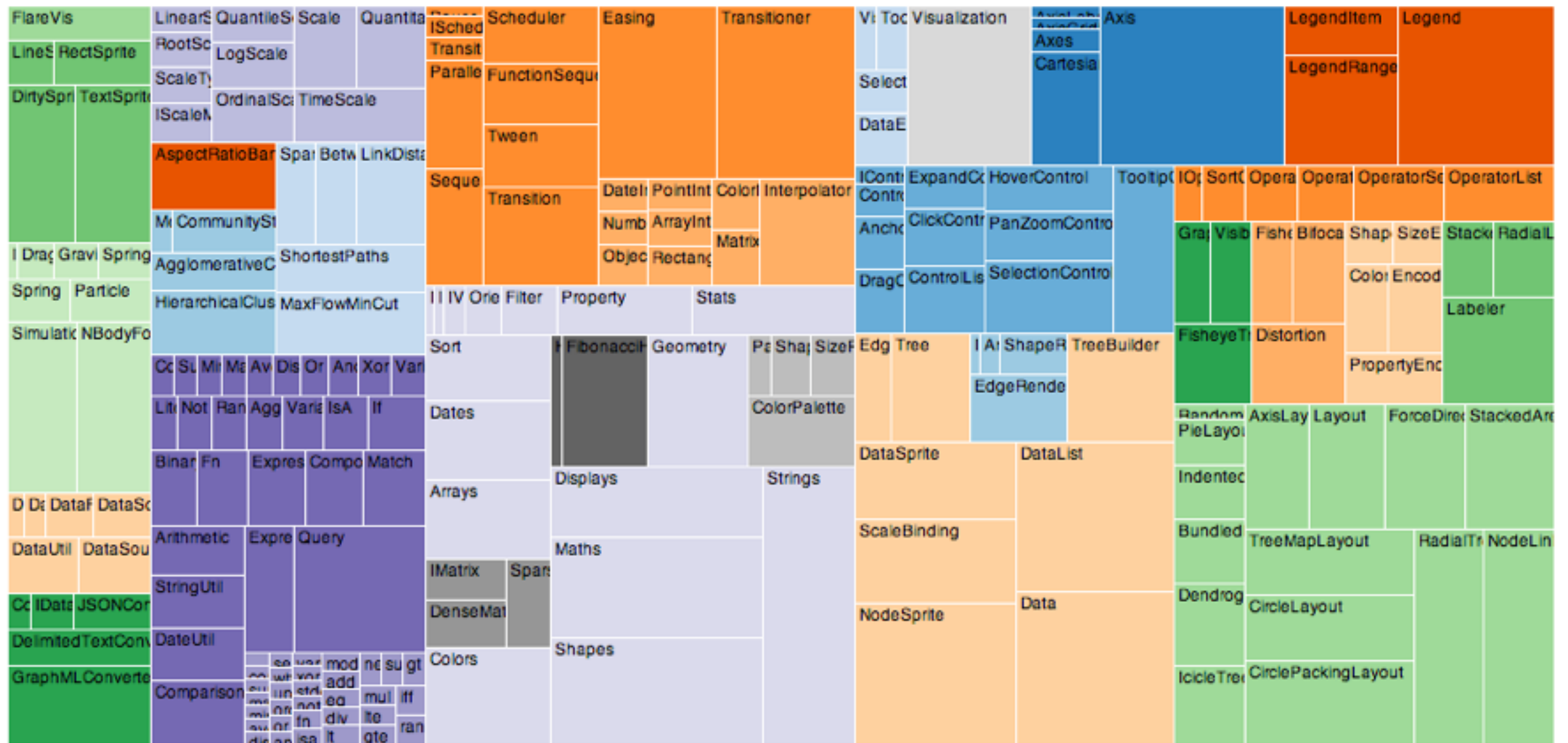


color.method.ac/

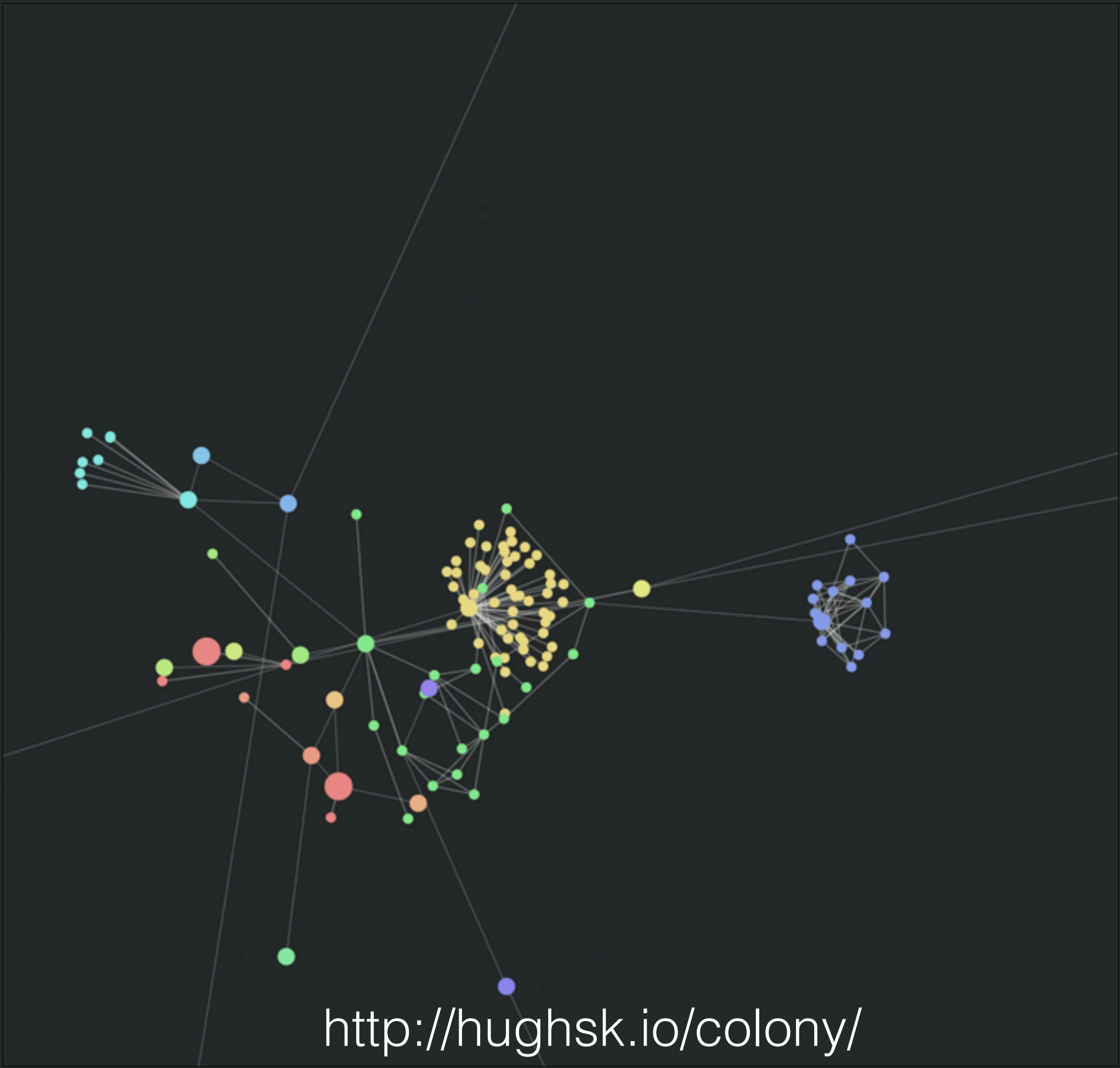


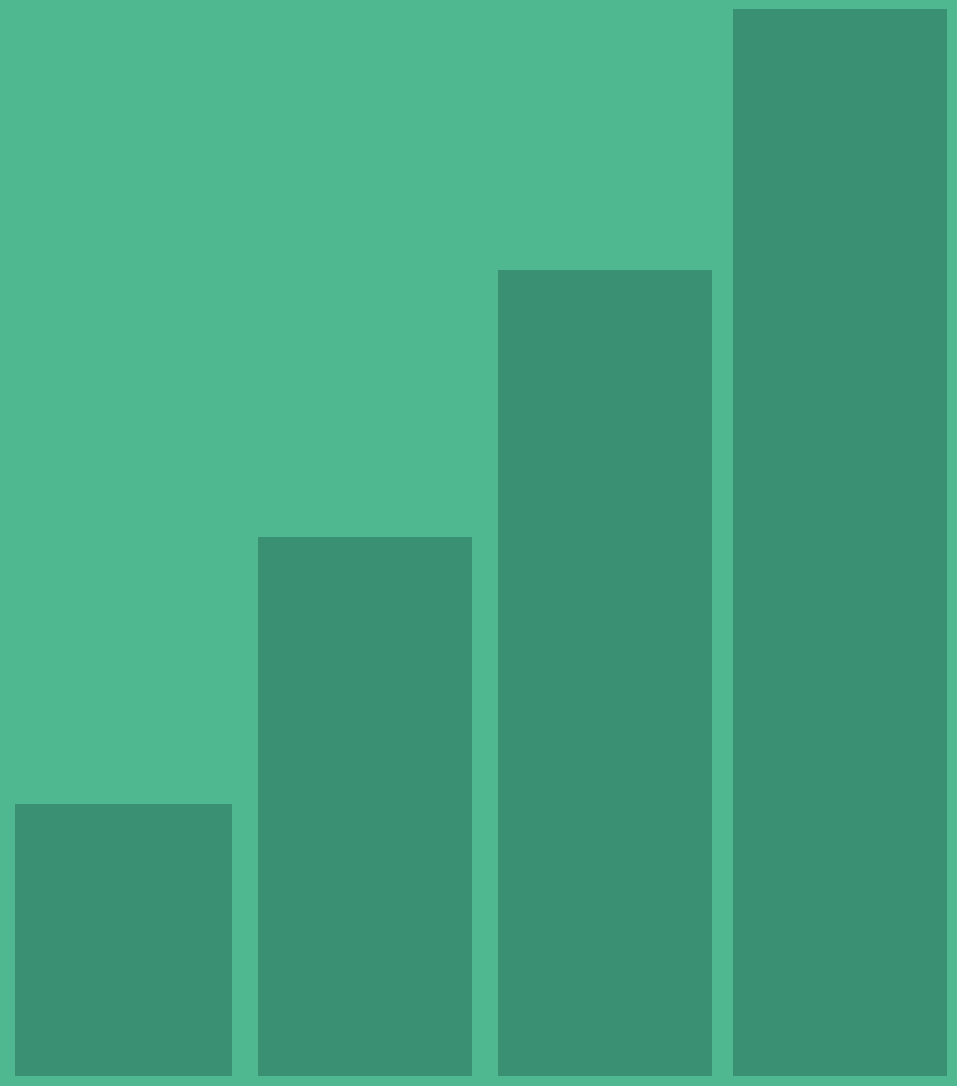
bl.ocks.org/mbostock/4063318

Size Count



<http://bl.ocks.org/mbostock/4063582>





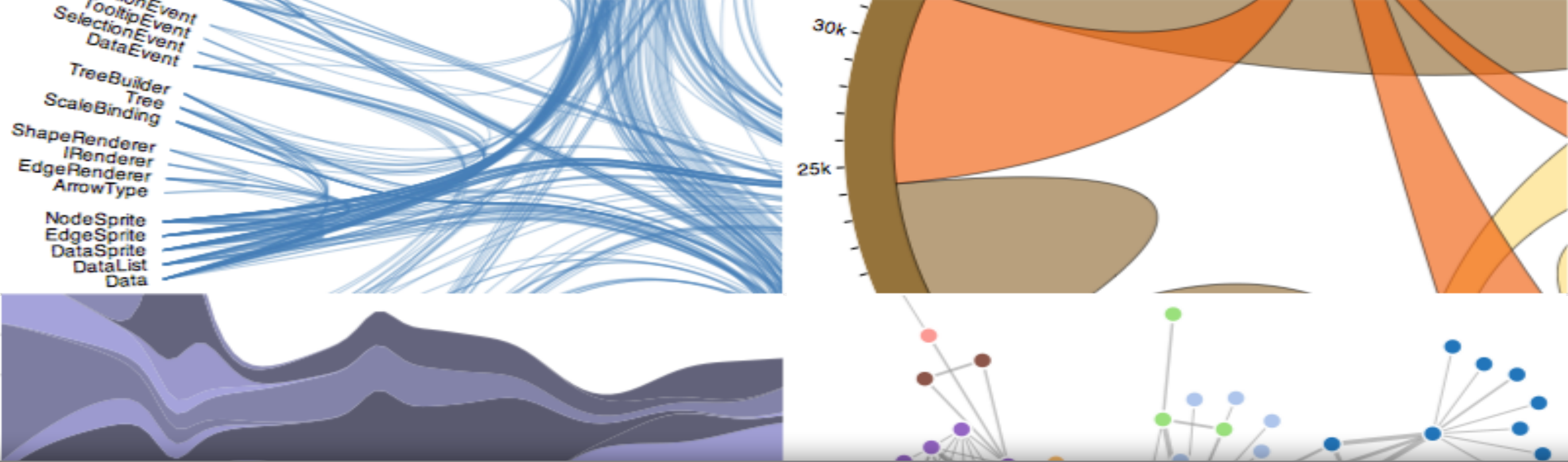
```
// Dimensions
height = 200;
barWidth = 40;
barGutter = 5;

// Our data
d = [5, 10, 15, 20]

// Scales For plotting the bars
max = d3.max(d);
x = d3.scale.linear()
  .domain([0, d.length])
  .range([0, (barWidth + barGutter) * d.length])
y = d3.scale.linear()
  .domain([0, max])
  .range([0, height]);
```

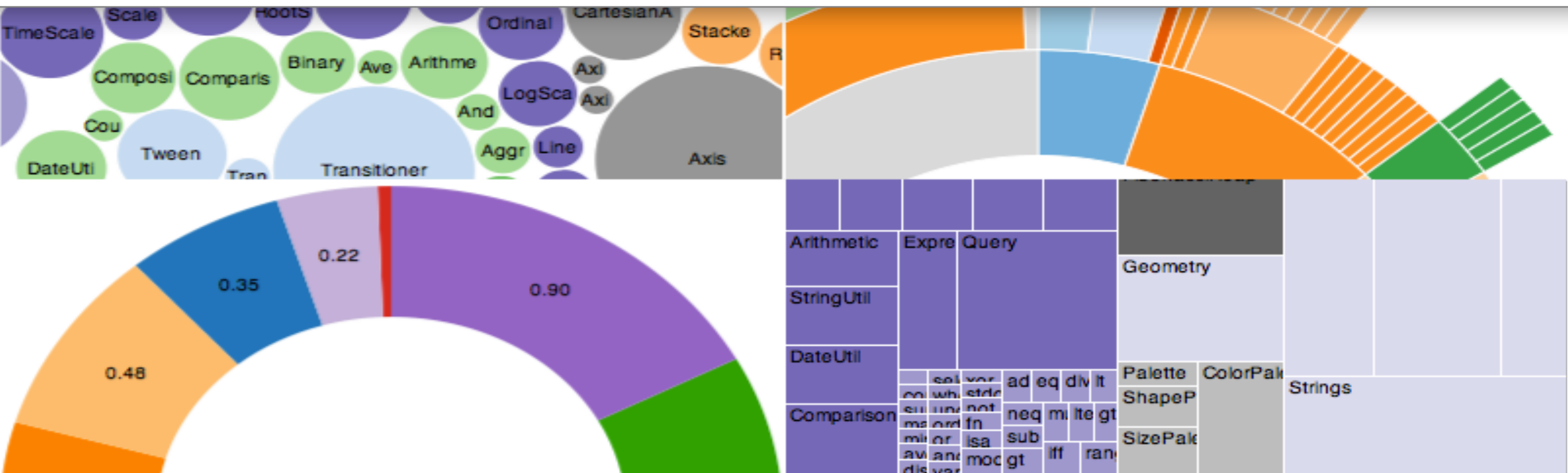
```
// Create the bar chart
d3.select("svg").selectAll("rect")
  .data(d)
  .enter()
  .append("rect")
  .attr("width", barWidth)
  .attr("height", function(d) {
    return y(d);
  })
  .attr("x", function(d, i) {
    return x(i);
  })
  .attr("y", function(d) {
    return height - y(d);
  })
```

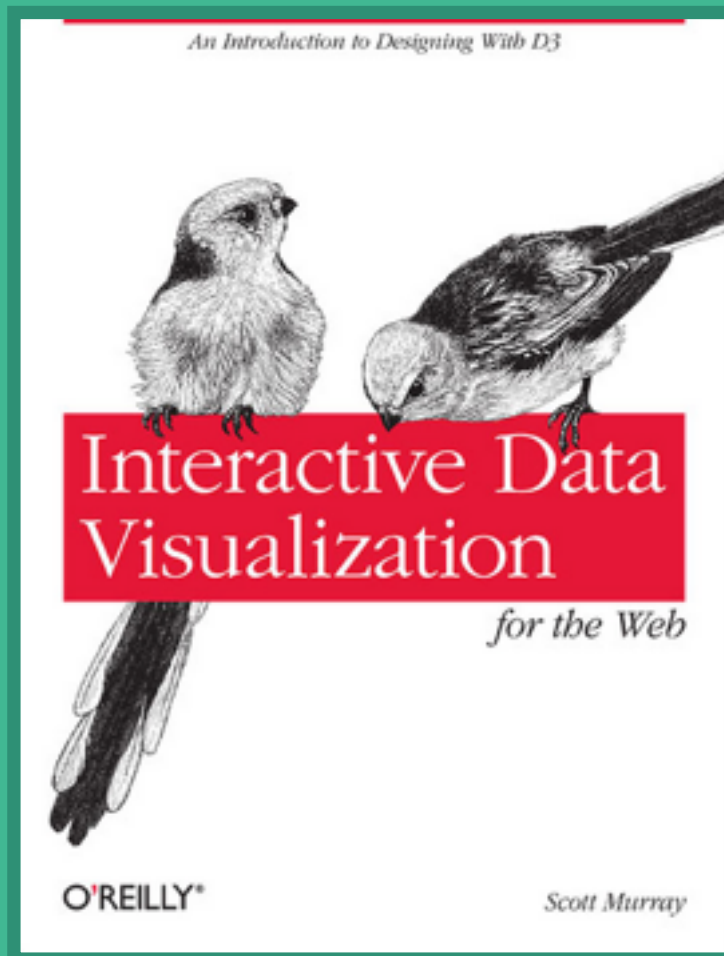
[jsbin.com/golafiye/1/edit?](https://jsbin.com/golafiye/1/edit?html,js,output)
[html,js,output](#)



Layouts

<https://github.com/mbostock/d3/wiki/Layouts>





alignedleft.com/tutorials/d3/

github.com/mbostock/d3/wiki/Gallery

github.com/mbostock/d3/wiki/Tutorials

@mbostock



Rickshaw

branch: master

rickshaw / Commits

Apr 02, 2014



Merge pull request #415 from dwt/support-empty-first-series-in-stack

dchester authored 6 hours ago

55f649c241

Browse code

Mar 26, 2014



Find correct domain even if the first series in stack is empty.

dwt authored 7 days ago

d021fc41be

Browse code



Allow empty data arrays in series.

dwt authored 7 days ago

faa8cbe209

Browse code

Mar 07, 2014



Merge pull request #391 from luke3141/fix_dot

dchester authored a month ago

877db9983f

Browse code



Add start and end scripts to allow use of UMD by Rickshaw

fjorgemota authored a month ago

dchester committed a month ago

61f1709fe9

Browse code

Feb 27, 2014



Merge pull request #400 from dwt/jquery-noconflict-fix

dchester authored a month ago

0f1b1ffb3e

Browse code

Feb 25, 2014



Define \$ variable in correct scope

rbu authored a month ago

4f12dc4d25

Browse code



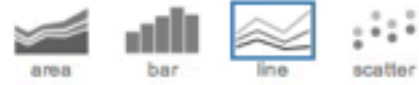
Make jQuery known as the global symbol instead of \$

rbu authored a month ago

5268d8a83e

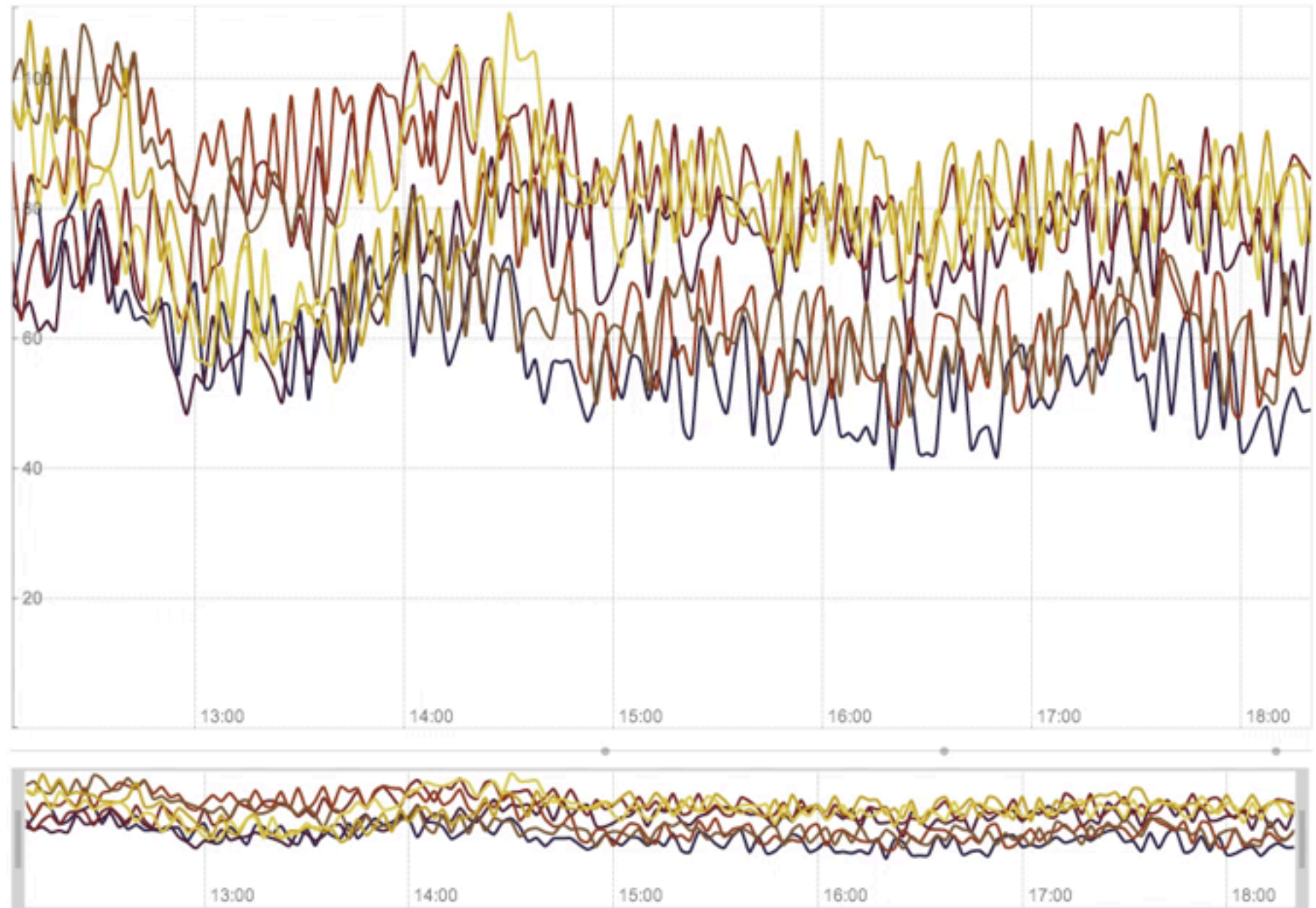
Random Data in the Future

- ✓ New York
- ✓ London
- ✓ Tokyo
- ✓ Paris
- ✓ Amsterdam
- ✓ Shanghai
- ✓ Moscow



- stack
- stream
- pct
- value
- cardinal
- linear
- step

Smoothing



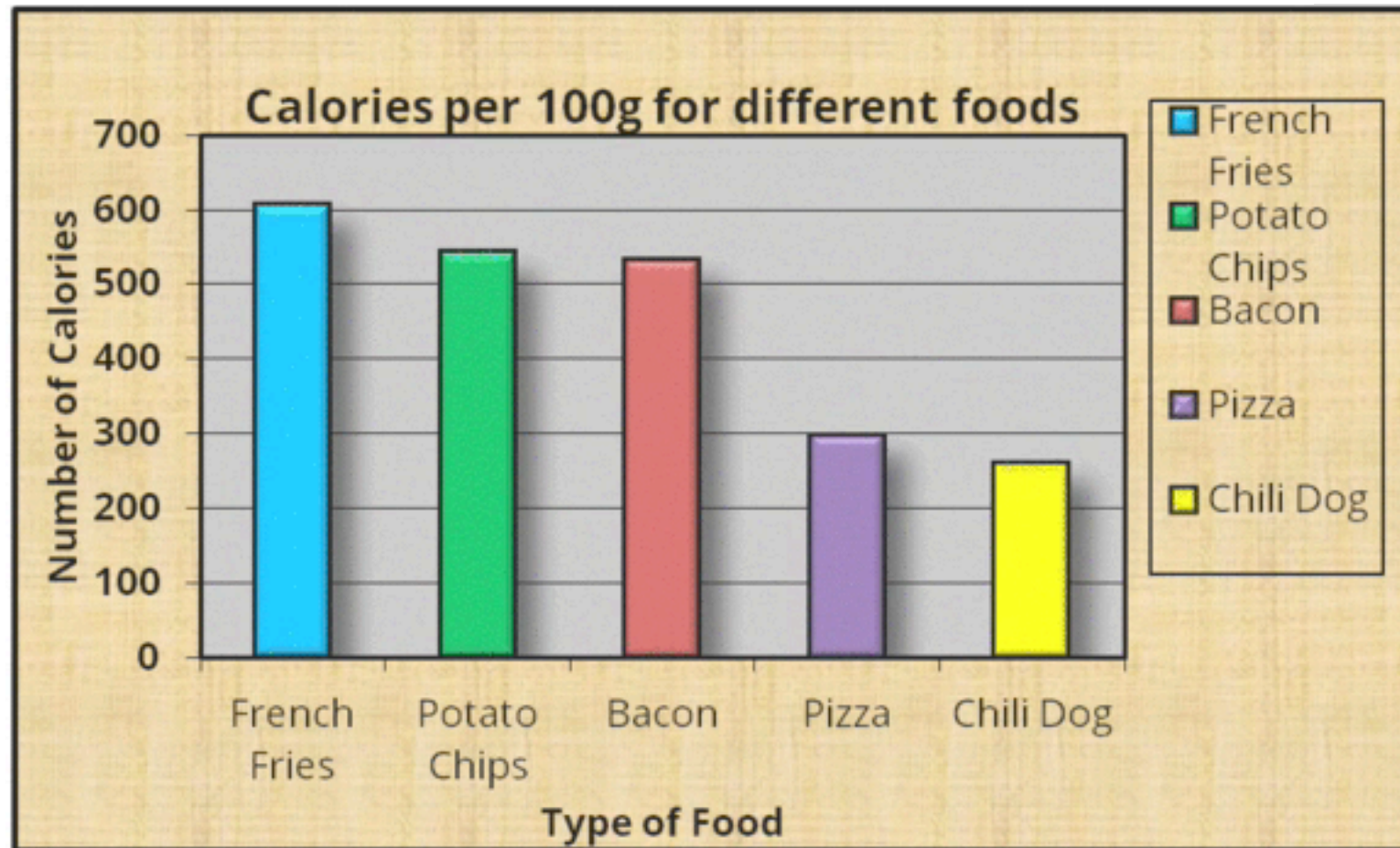
code.shutterstock.com/rickshaw/examples

```
var graph = new Rickshaw.Graph( {  
  element: document.querySelector("#chart"),  
  width: 300,  
  height: 200,  
  series: [{  
    color: 'steelblue',  
    data: [  
      { x: 0, y: 10 },  
      { x: 1, y: 20 },  
      { x: 2, y: 15 },  
      { x: 3, y: 30 },  
      { x: 4, y: 5 } ]  
    } ]  
});
```



***“Above all else
show the data”***

Edward Rolf Tufte, 1983



Created by Darkhorse Analytics

www.darkhorseanalytics.com

bit.ly/183R6sT

```
graph = new Rickshaw.Graph({ /* ... */ })  
  
xAxis = new Rickshaw.Graph.Axis.Time({  
  graph: graph  
});  
  
yAxis = new Rickshaw.Graph.Axis.Y({  
  graph: graph  
})
```

- New York
- London
- Tokyo
- Paris
- Amsterdam
- Shanghai
- Moscow



- ✓ ■ New York
- ✓ ■ London
- ✓ ■ Tokyo
- ✓ ■ Paris
- ✓ ■ Amsterdam
- ✓ ■ Shanghai
- ✓ ■ Moscow


```
graph = new Rickshaw.Graph({ /* ... */ });  
  
legend = new Rickshaw.Graph.Legend({  
  graph: graph,  
  element: document.querySelector( '#legend' )  
});  
  
new Rickshaw.Graph.Behaviour.Series.Toggle({  
  graph: graph,  
  legend: legend  
});
```

```
graph = new Rickshaw.Graph({ /* ... */ });  
  
legend = new Rickshaw.Graph.Legend({  
  graph: graph,  
  element: document.querySelector( '#legend' )  
});  
  
new Rickshaw.Graph.Behaviour.Series.Toggle({  
  graph: graph,  
  legend: legend  
});
```

Rickshaw

++

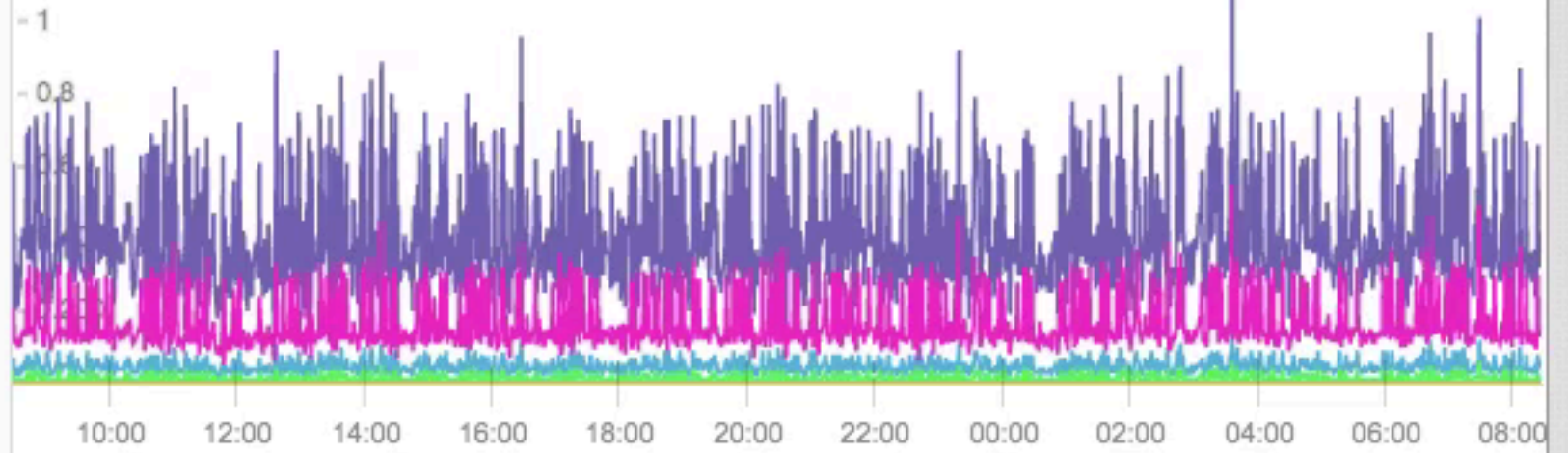
Network traffic

- Deselect All
- eth0: Received MB/s
 - eth0: Received Mb/s
 - eth0: Transmitted M...
 - eth0: Transmitted M...
 - lo: Received MB/s

All items

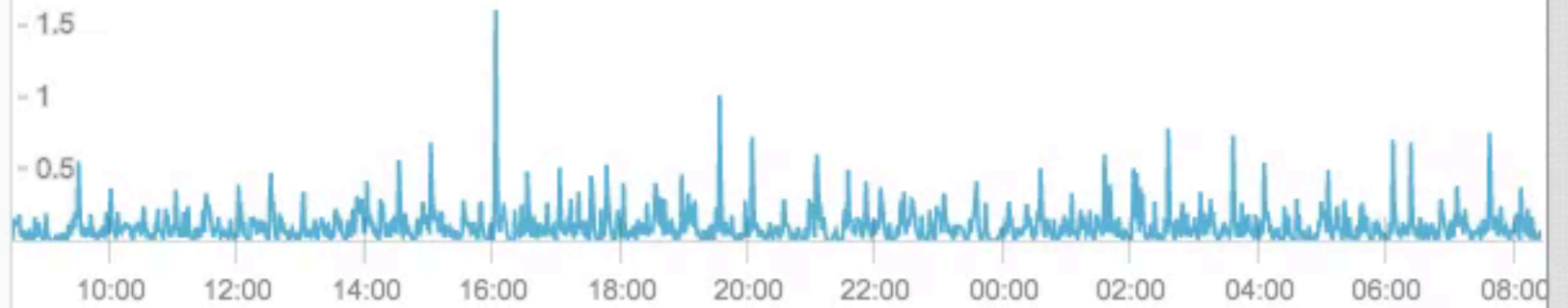
All metrics

Line



Load average

Line

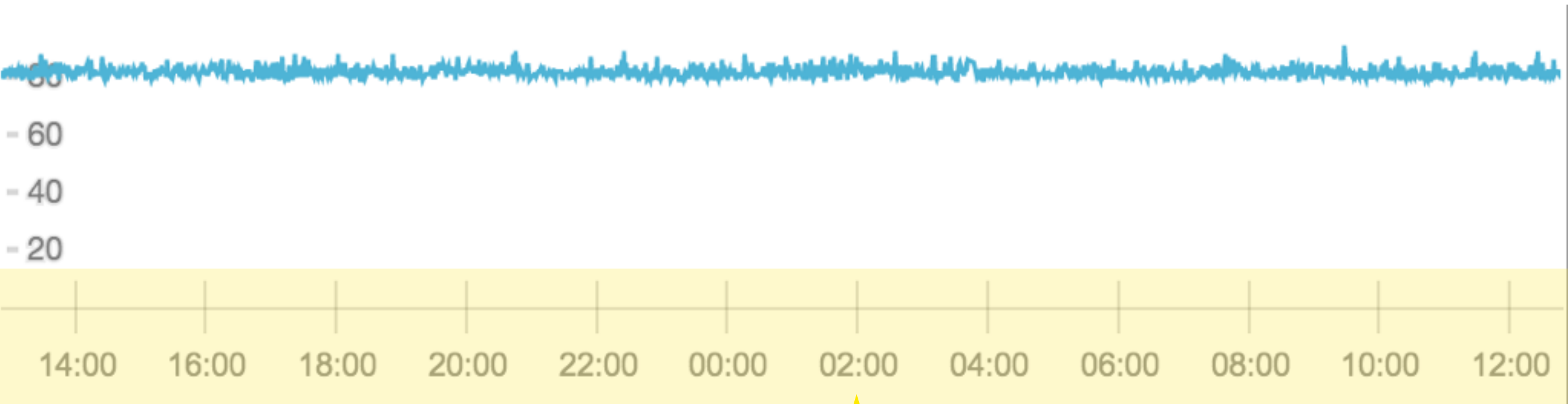


```
tooltips = new Rickshaw.Graph HoverDetail({
  graph: graph,
  onRender: function(point) {
    mediator.trigger('series: hover', point)
  },
  onHide: function() {
    mediator.trigger('series: hover: end')
  }
})
```



```
Rickshaw.namespace( 'Rickshaw.Graph HoverDetailUpgrade' );

Rickshaw.Graph HoverDetailUpgrade = Rickshaw.Class.create(
  Rickshaw.Graph HoverDetail, {
    render: function( args ) {
      // Overwrite the render function
      // The rest of the functions will
      // inherit from Rickshaw.Graph HoverDetail
    }
  }
);
```

A tick every 2 hours

	Tick Every	No of Ticks	Date Format
30 minutes	2 minutes	15	HH:mm
1 hour	5 minutes	12	HH:mm
24 hours	2 Hours	12	HH:mm
> 24 hours	Dynamic	15	D MMM YYYY

```
var timeFixture = new Rickshaw.Fixtures.Time()

timeFixture.units = timeFixture.units.push({
  name: '5 minute',
  seconds: 60 * 5,
  formatter: (d) -> timeFixture.formatTime(d)
})
```

```
timeFixture.formatTime = function(d) {  
  var time = moment(d).tz(ActiveUser.get('timezone'))  
  return time.format('HH:mm');  
};
```

```
xAxis = new Rickshaw.Graph.Axis.Time({  
  grid: false,  
  graph: graph,  
  timeUnit: timeFixture.unit("5 minute")  
});
```

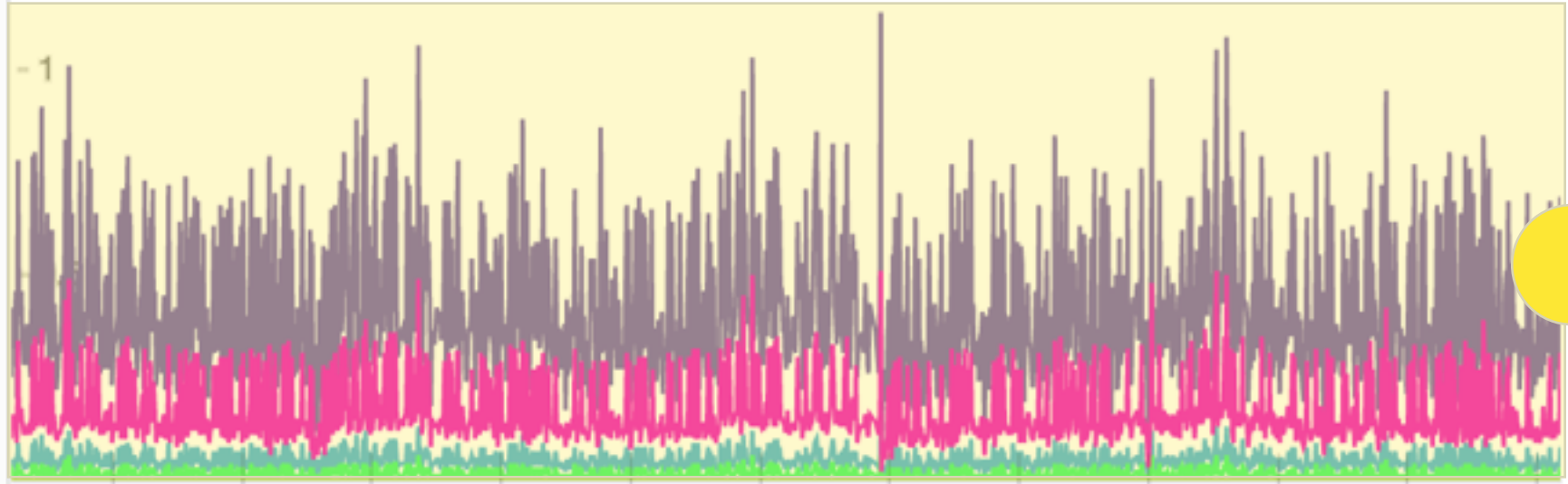
Network traffic

- Deselect All
- eth0: Received ...
 - eth0: Received ...
 - eth0: Transmite...
 - eth0: Transmite...
 - Io: Received MB/s

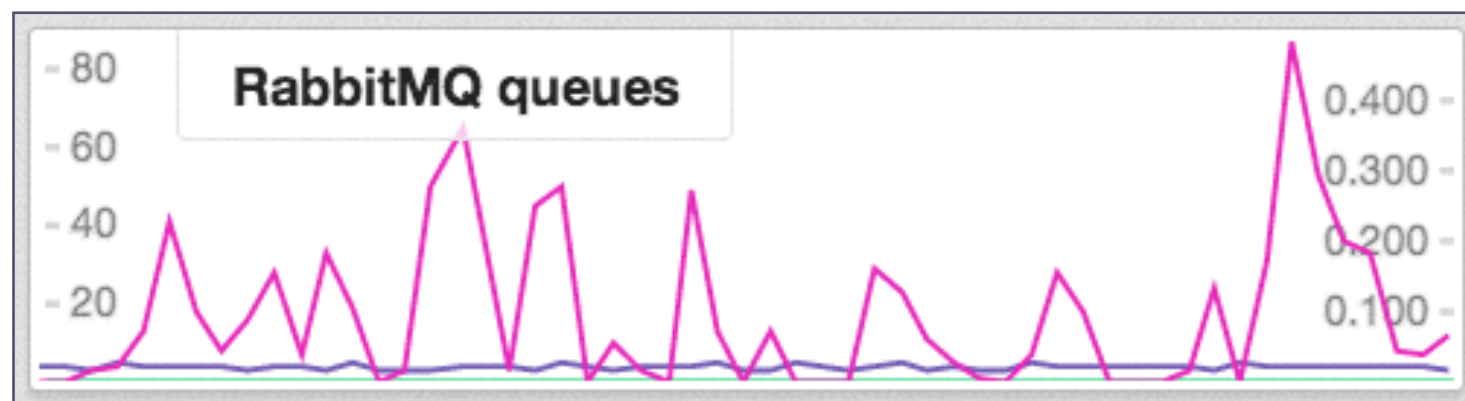
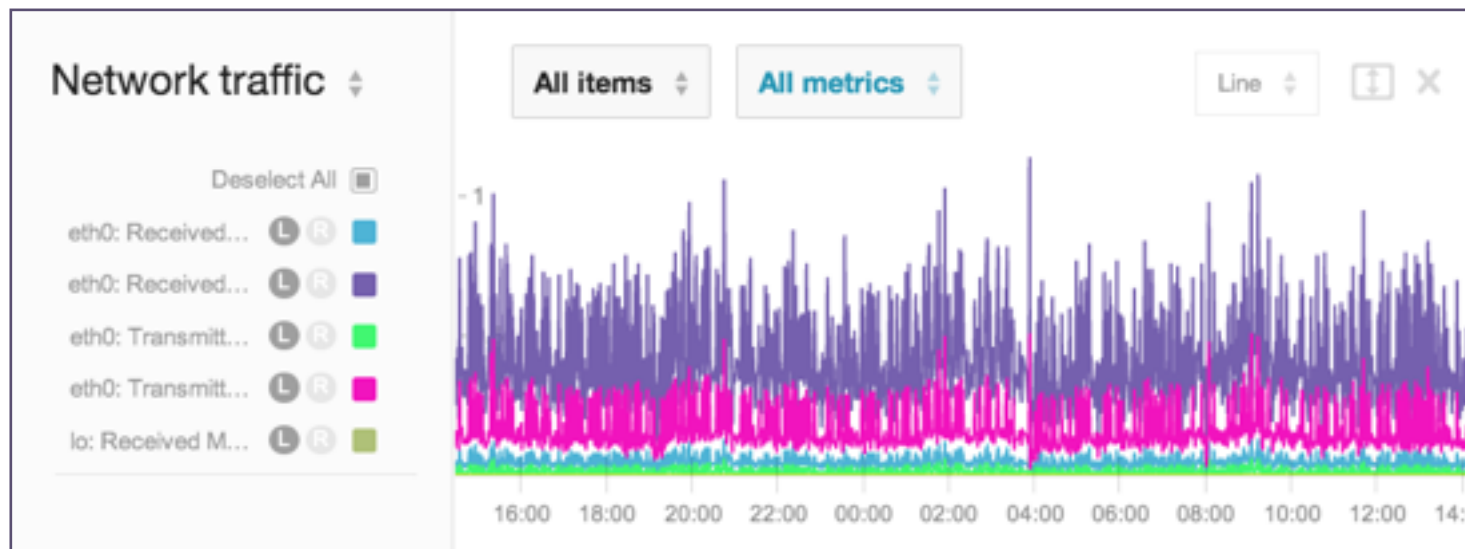
All items

All metrics

Line



16:00 18:00 20:00 22:00 00:00 02:00 04:00 06:00 08:00 10:00 12:00 14:00

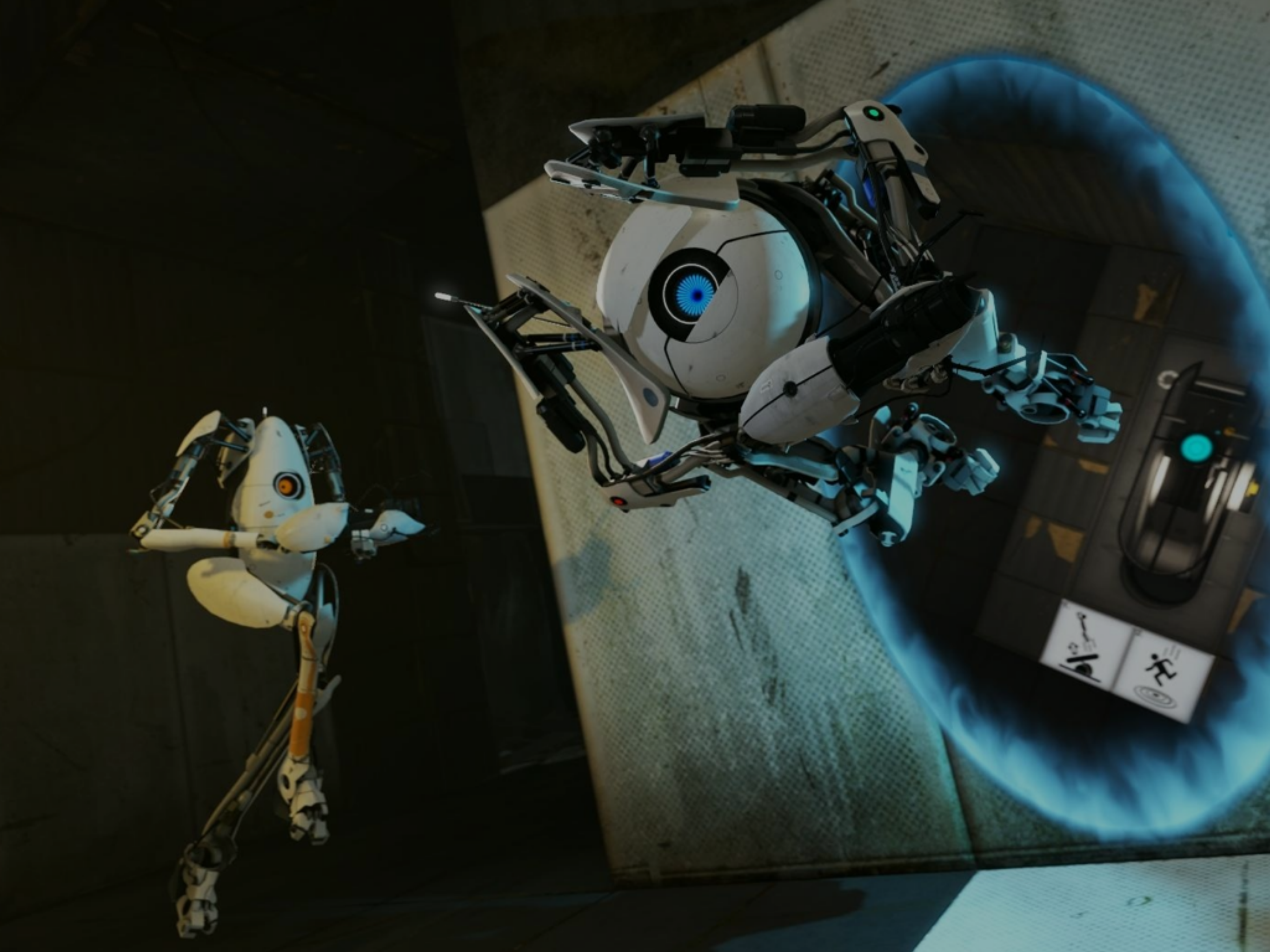




Knockout.

THE LESSON



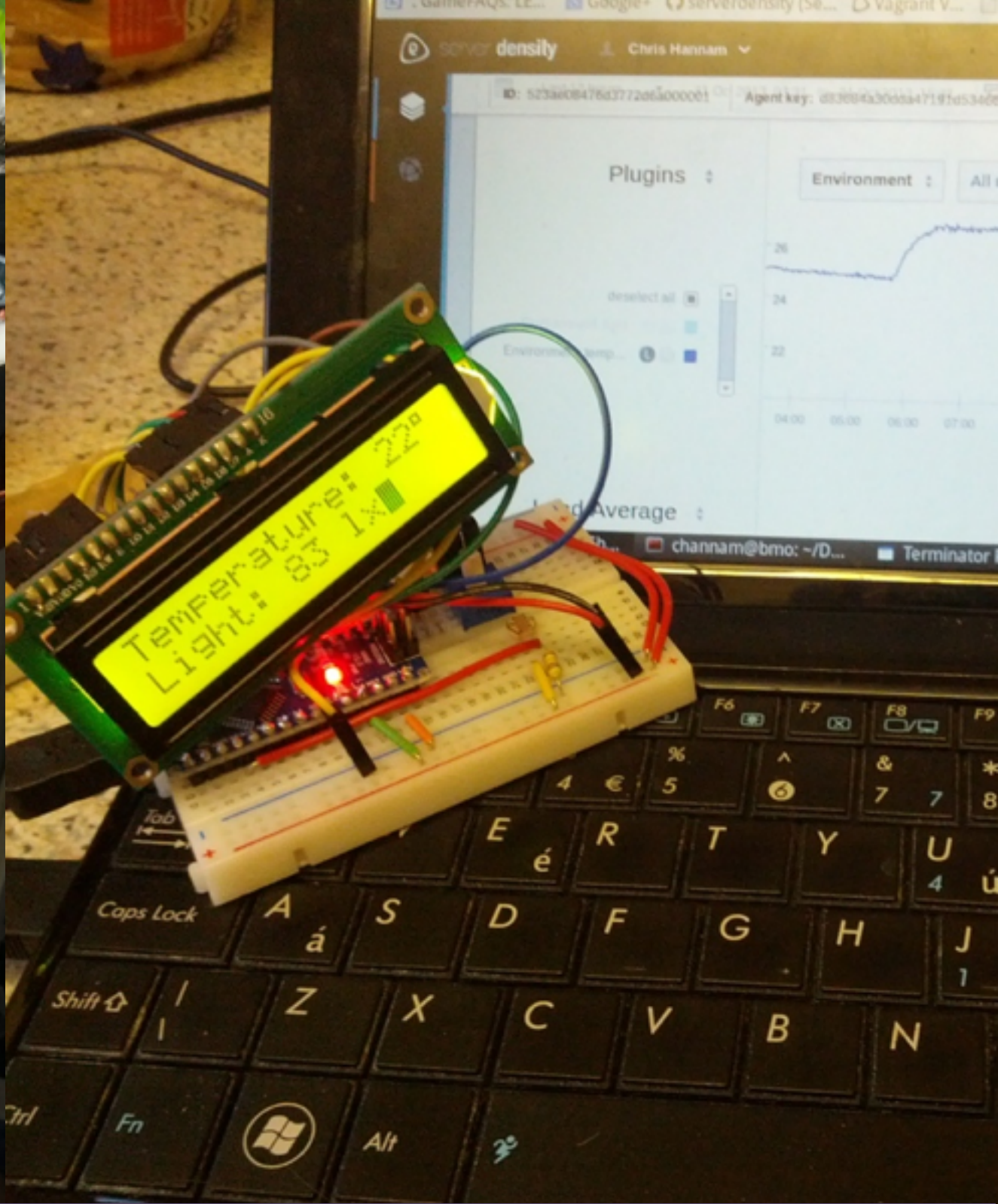


***“A state of joy, creativity
and total involvement
in which problems
seem to disappear”***

Mihaly Csikszentmihalyi

- 1. Intuitive interfaces**
- 2. Forgiving workflows**
- 3. Attractive designs**
- 4. Personality**
- 5. And avoiding...**

✘ ▶ TypeError: Cannot call method 'toString' of null



Flexible Data. Sigh.


```
series = {  
  data = [  
    {x: 1384851113, y: null},  
    {x: 1384851169, y: null}  
  ]  
}  
  
_.each( series.data, function(point) {  
  if ($.type(point.y) !== "number") {  
    series.error = "Y values aren't numbers"  
    series.hasError = true  
  }  
})
```

```
return {  
  valid: validSeries  
  invalid: invalidSeries  
}
```

```
<li {{#hasError}}class="has-error"{{#hasError}}>  
  <div {{#hasError}}popover="{{error}}"{{#hasError}}>  
    {{seriesName}}  
  </div>  
</li>
```

IO stats

Deselect All

Contains y values that aren't numbers

~~Level One: Error 1~~

~~Level One: Error 2~~

Level One: Valid 2

Level One: Valid 3



```
if serie.data.length < 2
  serie.error = "Does not have enough data"
  return

for point, i in serie.data

  if $.type(point.x) isnt "number"
    serie.error = "Has x values that aren't dates"
    break

  if $.type(point.y) isnt "number"
    serie.error = "Has y values that aren't numbers"
    break

  if point.x < lastDate
    serie.error = "Has badly ordered dates"
    break
```



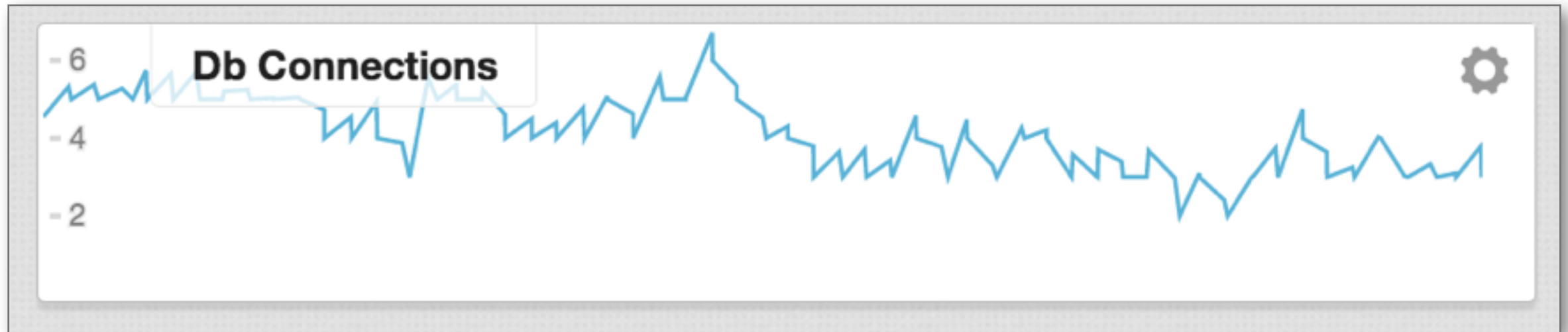
```
testedPoints = {}
lastDate = null
indexToRemove = []

for point, i in serie.data

  # Test x values for duplicates
  if testedPoints[point.x]?
    indexToRemove.push(i)

  testedPoints[point.x] = point.y
  lastDate = point.x

if indexToRemove.length
  for point, i in indexToRemove
    serie.data.splice(point-i, 1)
```



Jaggedy Data

```
for point, i in serie.data

# The first loop has no lastDate
if lastDate?

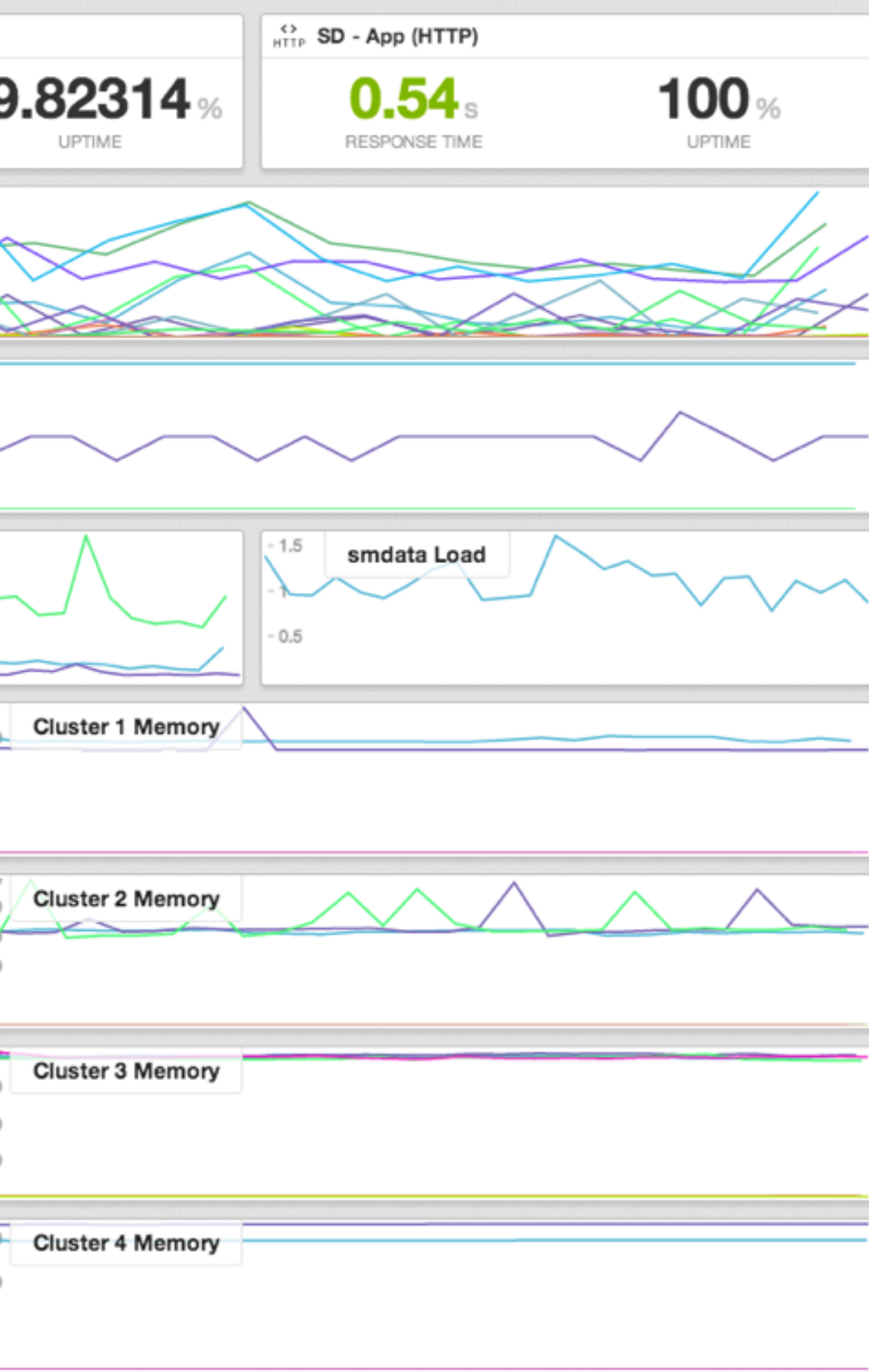
    thisTime = moment.unix(point.x).format('D:M:H:YY:m')
    lastTime = moment.unix(lastDate).format('D:M:H:YY:m')

# Maximum of one point per minute
if thisTime is lastTime
    indexToRemove.push(i)
    testedPoints[point.x] = point.y
    lastDate = point.x
    continue

if indexToRemove.length
    for point, i in indexToRemove
        serie.data.splice(point-i, 1)
```

- 1. Missing X & Y Data**
- 2. Badly ordered dates**
- 3. Duplicate X Values**
- 4. X Values too close**
- 5. Missing Series Names**

Unpredictable Data?



Monitor servers & websites
Configure Alerts
Manage your infrastructure

@serverdensity
@byrichardpowell