



Data Driven Documents



slides — Edit

117 commits

1 branch

0 releases

1 contributor

Branch: master ▾

New pull request

New file

Upload files

Find file

HTTPS ▾

<https://github.com/rinckd/d3-workshop>

Dow

rinckd updated css

Latest commit f938f39 9

src	updated css	9
.gitignore	updated gitignore	a
Presentation_Slides.pdf	updated slides	
README.md	updated readme	
readme.gif	readme	

README.md

Getting Started

Prerequisites

This is a lightweight Node/Express app which provides a nice sketchpad for playing around with d3.js.

You'll need:

timeSeries0.js - src - [~/__Code/_d3_workshop/src]

Line Chart - blocks.org

src wwwroot js timeSeries0.js

Project

src (~/_Code/_d3_workshop/src)

- node_modules (library home)
- server
 - app.js
- views
 - partials
 - heat.ejs
 - scratchpad.ejs
 - svg.ejs
- wwwroot
 - app
 - css
 - inlet.css
 - site.css
 - data
 - js
 - test.html
 - .babelrc
 - .editorconfig
 - .jscsrc
 - .jshintrc
 - gulpfile.babel.js
 - package.json

External Libraries

1 (function() {
2 'use strict';
3
4 var margin = {
5 top: 40,
6 right: 0,
7 bottom: 40,
8 left: 40
9 };
10 var width = 600 - margin.left - margin.right;
11 var height = 400 - margin.top - margin.bottom;
12
13 var svg = d3.select('#time-series')
14 .append('svg')
15 .attr('width', width + margin.left + margin.right)
16 .attr('height', height + margin.top + margin.bottom)
17 .append('g')
18 .attr('transform', 'translate(' + margin.left + ',
+ margin.top + ')');
19
20 var yScale = d3.scale.linear()
21 .range([height, 0]);
22 var yAxis = d3.svg.axis()
23 .scale(yScale)
24 .orient('left');
25
26 var xScale = d3.scale.linear()
27 .range([0, width]);
28 var xAxis = d3.svg.axis()
29 .scale(xScale)
30 .orient('bottom');
31
32
33
34 d3.json('/data/timeseries/ac-load.json').function (e

Debug Unnamed

Debugger Console → Elements

AC Electric

The code in the editor is as follows:

```
(function() {
  'use strict';

  var margin = {
    top: 40,
    right: 0,
    bottom: 40,
    left: 40
  };
  var width = 600 - margin.left - margin.right;
  var height = 400 - margin.top - margin.bottom;

  var svg = d3.select('#time-series')
    .append('svg')
    .attr('width', width + margin.left + margin.right)
    .attr('height', height + margin.top + margin.bottom)
    .append('g')
    .attr('transform', 'translate(' + margin.left + ', ' +
      margin.top + ')');

  var yScale = d3.scale.linear()
    .range([height, 0]);
  var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient('left');

  var xScale = d3.scale.linear()
    .range([0, width]);
  var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient('bottom');

  d3.json('/data/timeseries/ac-load.json').function (e
```

Scratchpad

The image displays a development environment with a code editor and a web browser. On the left, a code editor window titled "scratchpad3.js" contains the following D3.js code:

```
1 (function () {
2     var width = 700;
3     var height = 800;
4     var svg = d3.select('#time-series')
5         .append('svg')
6         .attr('width', width)
7         .attr('height', height);
8     svg.append('text')
9         .attr('x', 10)
10        .attr('y', 20)
11        .text('hello world!');
12 })();
13
```

On the right, a browser window titled "D3" shows the output of the code. The address bar indicates the page is at "localhost:3000/scratchpad/3". The main content area of the browser displays the text "hello world!".

<http://localhost:3000/scratchpad/3>

The screenshot shows a code editor interface with a dark theme. The top bar displays the title "scratchpad11.js - src - [~/__Code/_d3_workshop/src]" and several tabs for other files like "timeSeries0.js", "scratchpad11.js", "heatmap.js", "gulpfile.babel.js", and "scratchpad3.js".

The left sidebar shows a project structure under "Project" with a "1-Project" icon. It lists files such as "webpack", "heatmap.js", "scratchpad0.js", "scratchpad2.js", "scratchpad3.js", "scratchpad4.js", "scratchpad5.js", "scratchpad6.js", "scratchpad7.js", "scratchpad11.js" (which is selected), "timeSeries0.js", "timeSeries1.js", "timeSeries2.js", "timeSeries3.js", "timeSeries4.js", "timeSeries5.js", "timeSeries6.js", "timeseries7.js", and "timeseries8.js".

The main editor area contains the following JavaScript code:

```
1 (function() {
2
3     var svg = d3.select('#time-series')
4         .append('svg')
5         .attr('width', 600)
6         .attr('height', 600);
7     svg.append('text')
8         .attr('x', 10)
9         .attr('y', 20)
10        .text('hello world!!!');
11 })();
```

The bottom left corner shows a terminal window with the command "Davids-MacBook-Pro:src rinckd\$".

scratchpad.ejs

```
1 <!DOCTYPE html>
2 <head>
3     <% include ./partials/head %>
4 </head>
5 <body>
6     <div class="mdl-layout mdl-js-layout mdl-layout--fixed-drawer mdl-layout--fixed-header">
7         <% include ./partials/nav %>
8         <main class="mdl-layout__content">
9             <div class="data-content">
10                <div class="mdl-shadow--2dp mdl-cell mdl-cell--12-col">
11                    <div id="time-series">
12                        <% if (scripts) { %>
13                            <script src="<%- scripts %>"></script>
14                        <% } %>
15                </div>
```

scratchpad3.js

scratchpad5.js

8

```
1     (function () {
2         'use strict';
3         const width = 700;
4         const height = 800;
5
6         var svg = d3.select('#time-series')
7             .append('svg')
8                 .attr('width', width)
9                 .attr('height', height);
10            svg.append('text')
11                .attr('x', 10)
12                .attr('y', 20)
13                .text('hello world');
14    })();
```

Chaining Methods

```
eSeries0.js x JS scratchpad3.js x JS scratchpad11.js x ▾ 3 JS scratchpad3.js x
1  ⌂(function() {                                     ✓
2    var svg = d3.select('#time-series')
3      .append('svg')
4      .attr('width', 600)
5      .attr('height', 600);
6    svg.append('text')
7      .attr('x', 10)
8      .attr('y', 20)
9      .text('hello world!!!');
10   })();
11
1  ⌂(function () {
2    var sel = d3.select('#time-series');
3    var svg = sel.append('svg');
4    svg.attr('width', 600);
5    svg.attr('height', 600);
6    var text = svg.append('text');
7    text.attr('x', 10);
8    text.attr('y', 20);
9    text.text('hello world!!!');
10   })();
11
```



Chains Gone Bad



A screenshot of a developer environment showing a code editor and a browser window. The code editor on the left contains a JavaScript file named 'scratchpad11.js' with the following code:

```
1 (function() {
2   var svg = d3.select('#time-series')
3     .attr('width', 600)
4     .append('svg')
5     .attr('height', 600);
6   svg.append('text')
7     .attr('x', 10)
8     .attr('y', 20)
9     .text('hello world!!');
10 })();
11
```

The browser window on the right shows a page titled 'Intro to' with the text 'hello world!!'. The browser's developer tools are open, specifically the Elements tab, which displays the DOM structure:

```
...</div>
<main class="mdl-layout__content">
  <div class="data-content">
    <div class="mdl-shadow--2dp mdl-cell mdl-cell--12-col">
      ...<div id="time-series" width="600">
        <script src="/js/scratchpad11.js"></script>
        <svg height="600">
          <text x="10" y="20">hello world!!</text>
        </svg>
      </div>
    </div>
  </main>
<div>
```

SVG Paths and d3



Why SVG?





Search GitHub

Pull requests Issues Gist

Delivering Octicons with SVG

February 22, 2016



aaronshekey

Engineering

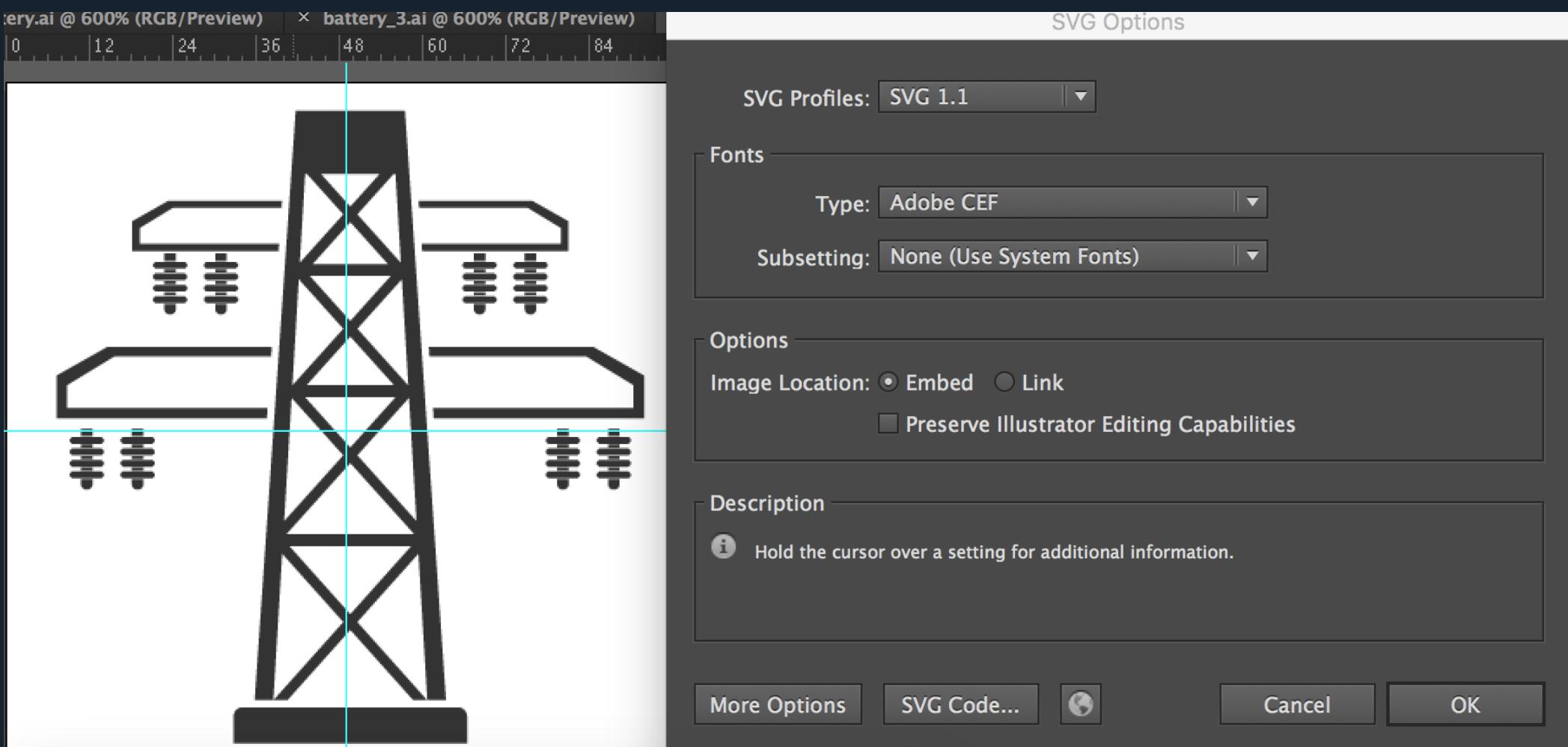
GitHub.com no longer delivers its icons via icon font. Instead, we've replaced all the [Octicons](#) throughout our codebase with SVG alternatives. While the changes are mostly under-the-hood, you'll immediately feel the benefits of the SVG icons.

```
<svg aria-hidden="true" class="octicon octicon-plus" width="12" height="16" role="img" version="1.1" viewBox="0 0 12 16">
  <path d="M12 9H7v5H5V9H0V7h5V2h2v5h5v2z"></path>
</svg>
```

Scalable



Illustrator Export



The screenshot shows the Adobe Illustrator interface with a document titled "battery_3.ai" at 600% (RGB/Preview). The artwork consists of a central vertical tower with diagonal cross-bracing, flanked by two horizontal rectangular components and four small cylindrical objects. A vertical cyan selection bar is positioned to the left of the central tower.

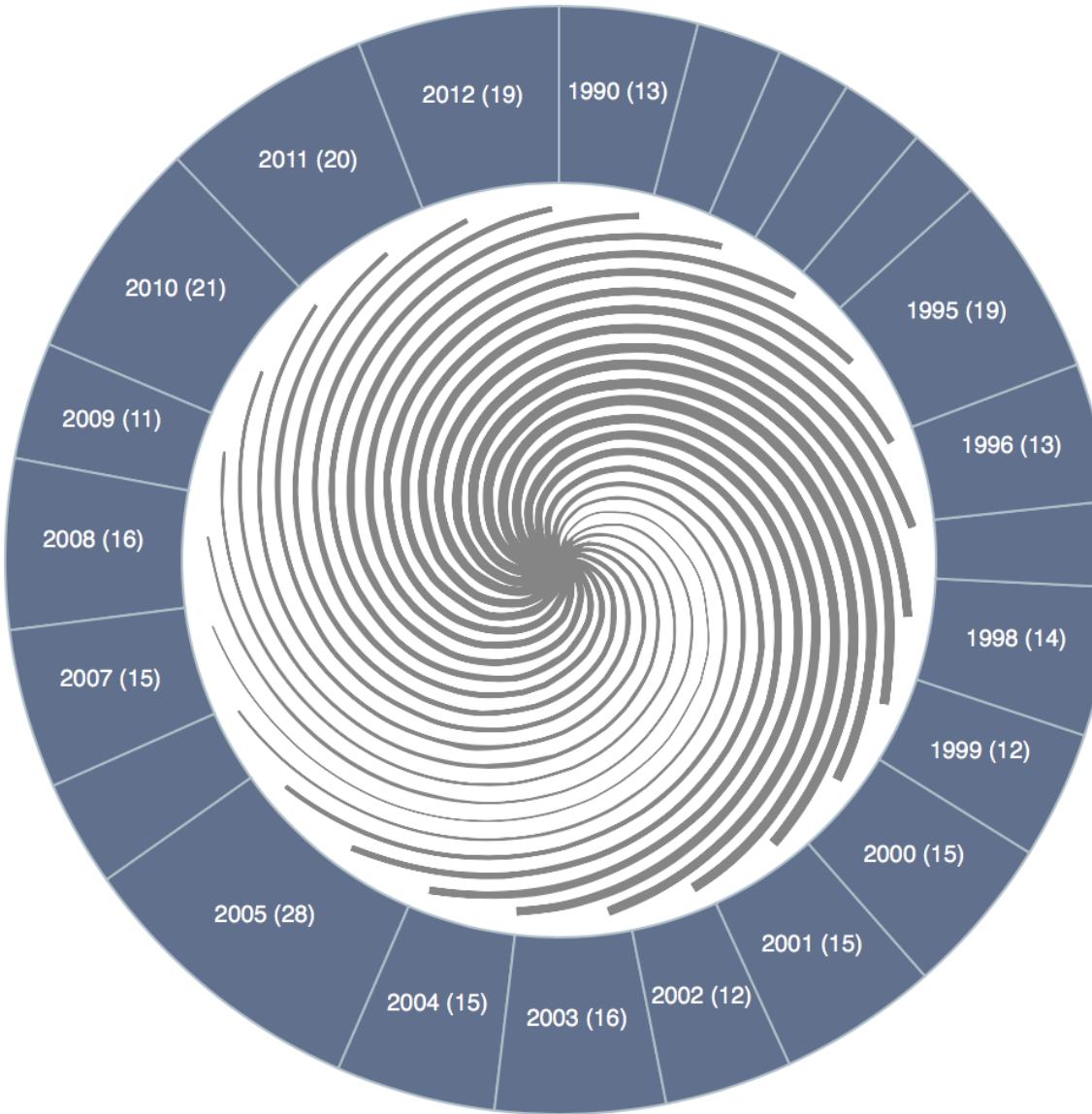
The "SVG Options" dialog box is open, set to "SVG 1.1" profile. It includes sections for "Fonts" (Type: Adobe CEF, Subsetting: None (Use System Fonts)), "Options" (Image Location: Embed selected, Link), and "Description" (Hold the cursor over a setting for additional information). At the bottom are buttons for "More Options", "SVG Code...", "Cancel", and "OK".

Below the main window, a text editor window titled "ai14574484501.txt" displays the generated SVG code. The code includes paths and rectangles defining the shapes in the design.

```
0.151,0.344-0.339v2z/.48z"/>
<path fill="#333333" d="M33.504,29.401c0-0.188-0.155-0.337-0.344-0.337h-4.195c-0.187,0-0.338,0.148-0.338,0.337v0.256c0,0.186,0.151,0.339,0.338,0.339h4.195c0.188,0,0.344-0.153,0.344-0.339v29.401z"/>
<rect x="30.259" y="30.154" fill="#333333" width="1.61" height="0.406"/>
```

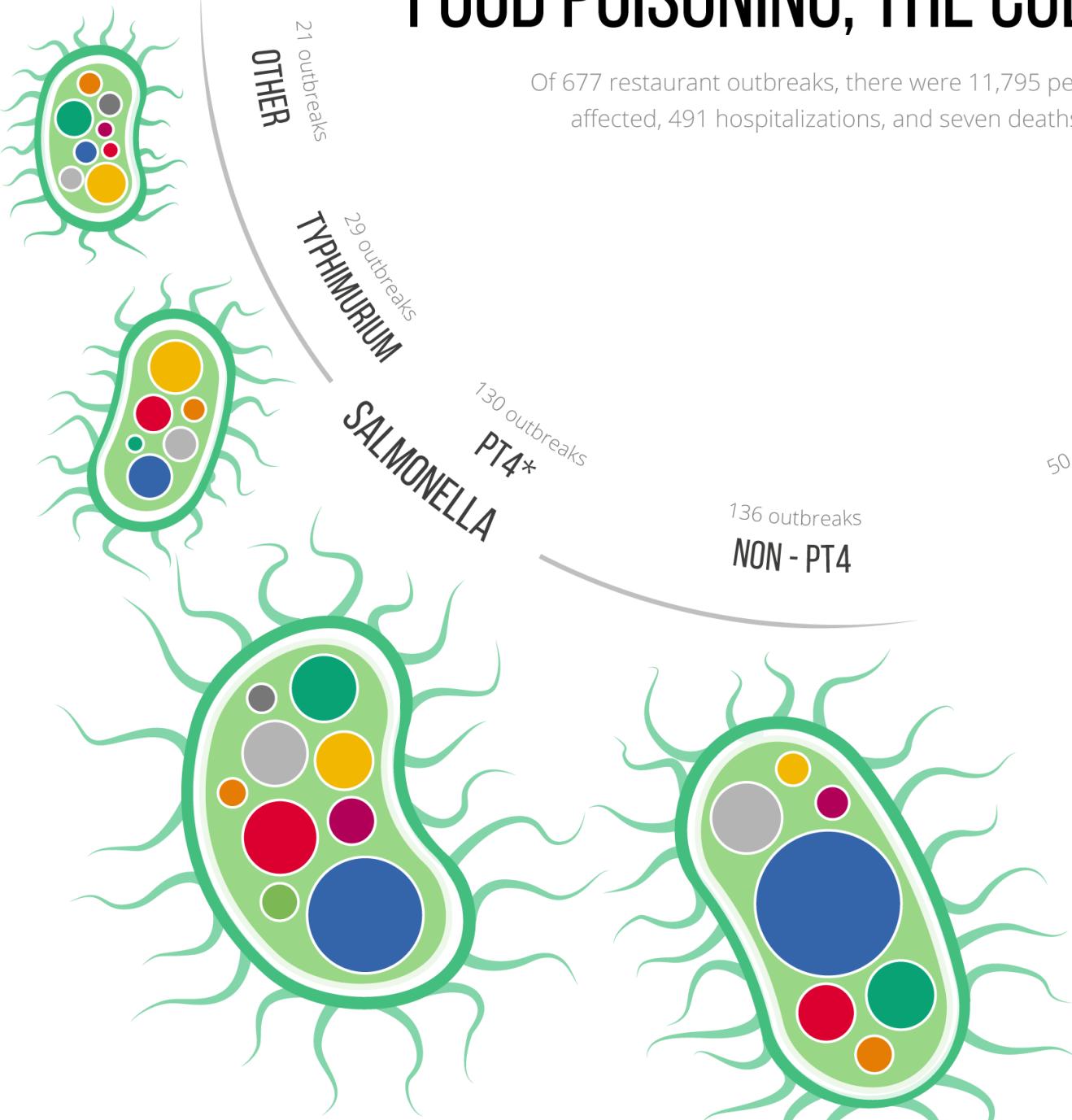
Illustrator SVG in D3

U.S. Hurricane Seasons : 1990 to 2012



FOOD POISONING, THE CULPRITS

Of 677 restaurant outbreaks, there were 11,795 people affected, 491 hospitalizations, and seven deaths



38 outbreaks
C. PERFRINGEN

50 outbreaks
BACILLUS

50 outbreaks
VIRUSES

- CUISINE**
- Indian
 - Sandwich
 - Italian
 - Continental
 - Seafood
 - Chinese
 - British
 - American

NUMBER OF OUTBREAKS

50

 Code

 Issues 11

 Pull requests 4

 Wiki

 Pulse

 Graphs

Extracts an SVG node and accompanying styles from an HTML document and allows you to download it all as an SVG file.

<http://nytimes.github.com/svg-crowbar/>

 149 commits

 1 branch

 0 releases

3 contributors

Branch: **gh-pa...** ▾

New pull request

New file

Upload files

Find file

HTTPS ▾

<https://github.com/NYTimes/>



Download ZIP



shancarter Removed check for font-family

Latest commit 9fdd16a on Jan 22, 2015

 assets

better samples

3 years ago

 LICENSE

Create LICENSE

3 years ago

 README.md

Update README.md

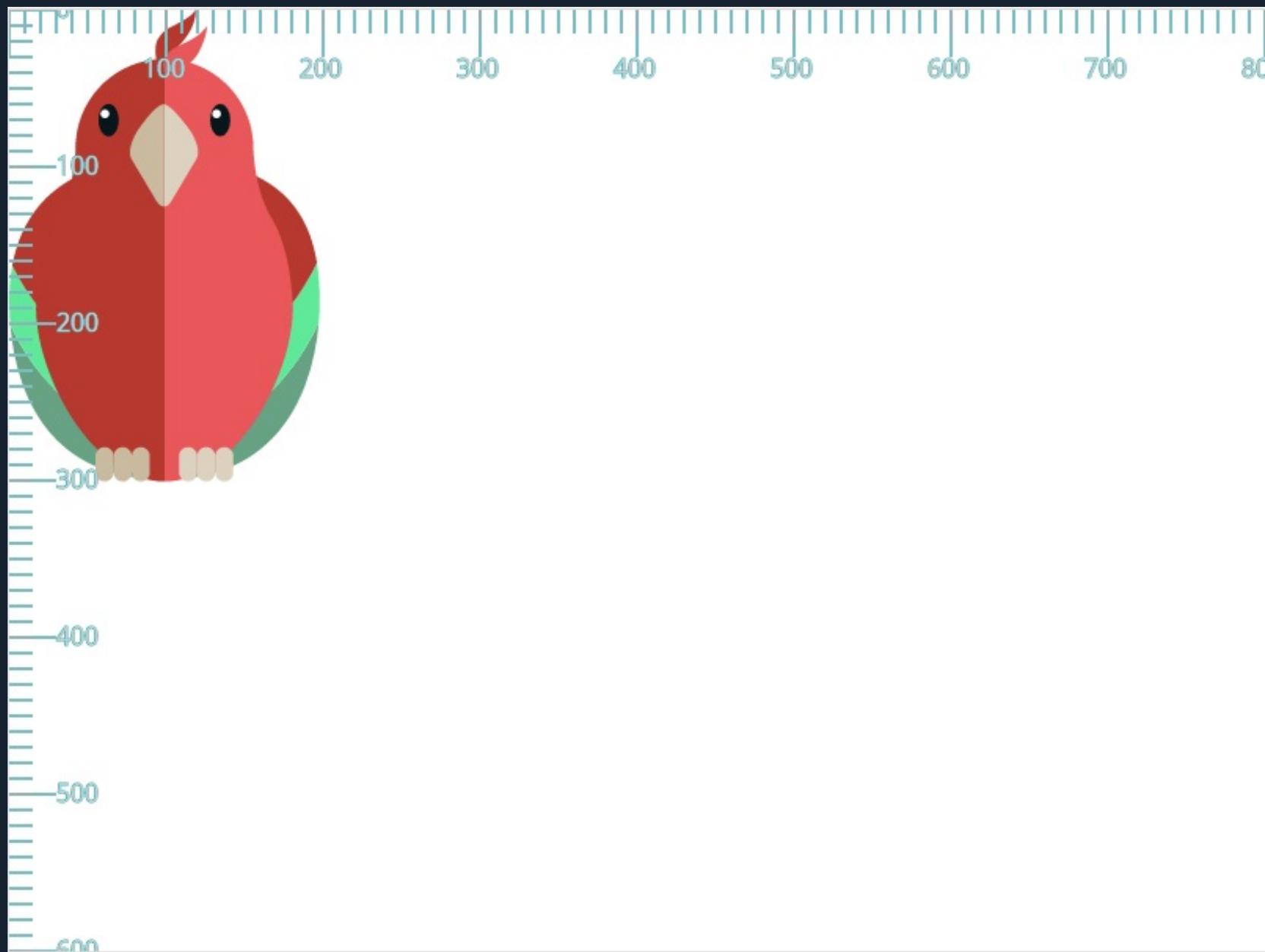
3 years ago

 index.html

typos

2 years ago

Coordinate System



A screenshot of a development environment showing a code editor and a browser window side-by-side.

The code editor (left) displays a file named `scratchpad7.js` with the following content:

```
1 (function() {
2     var svg = d3.select('#time-series')
3         .append('svg')
4         .attr('width', 700)
5         .attr('height', 700);
6
7
8
9 })();
10
```

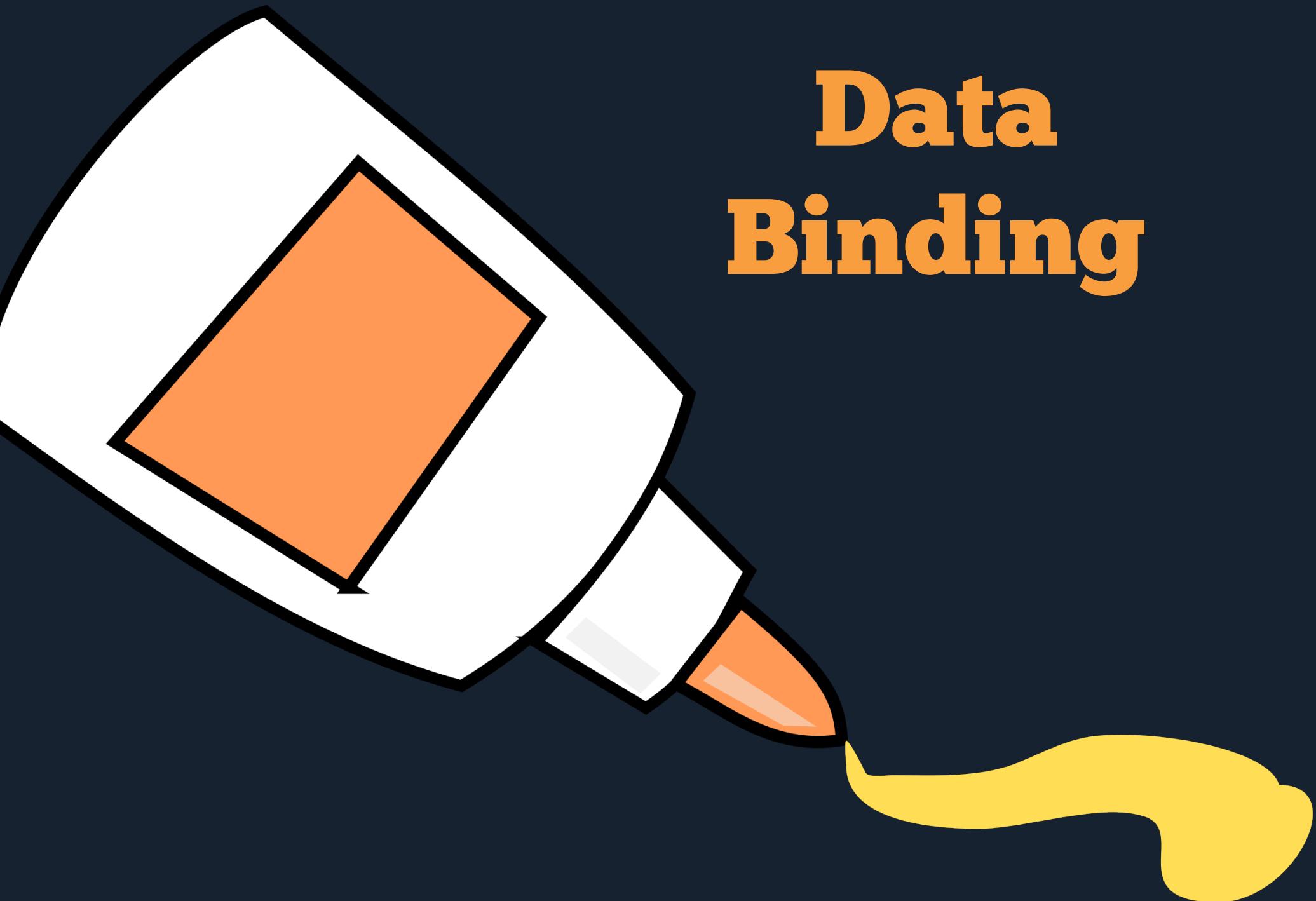
The browser window (right) shows a blank white page with a blue header bar containing the text "Intro to D3". The address bar indicates the page is at `localhost:3002/scratchpad/7`.

SVG Path

```
<path d = "M100 100 L300 100 L200 300z"/>
```

- M = move to
- L = line to
- H = horizontal line to
- V = vertical line to
- C = curve to
- S = smooth curvet
- Q = quadratic Bezier curve
- T = smooth quadratic Bz curve to
- A = elliptical arc
- Z = close path

Data Binding





Generators

```
var someArray = [1,2,3, 4, 200, 240];
```



```
<path d=
  "M50,358.9558292950576L53.9855072463768,35
  8.78180084423377C57.97101449275362,358.607
  77239341,65.94202898550725,358.25971549176
  25,73.91304347826086,357.91165859011505C81
  .8840579710145,357.56360168846754,89.85507
  246376812,357.21554478682003,97.8260869565
  2173,356.8674878851726C105.79710144927536,
  356.51943098352507,113.76811594202897,356.
  1713740818776,121.7391304347826,321.887769
  26959935C129.71014492753622,287.6041644573
  2113,137.68115942028984,219.38501173441213
  ,145.65217391304347,178.31429734000776C153
```

<http://localhost:3000/scratchpad/6>

The image shows a dual-pane interface. On the left is a code editor in a dark-themed IDE, and on the right is a web browser window.

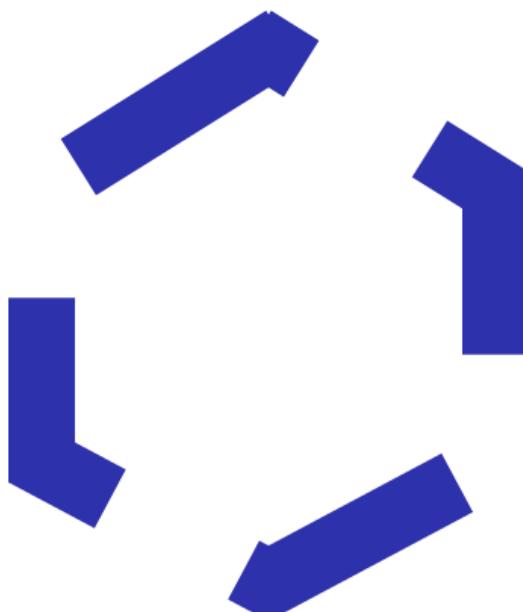
Code Editor (Left):

```
1 (function() {
2     var pointData = [
3         {
4             'x':205,
5             'y':110
6         },
7         {
8             'x':125,
9             'y':160
10    },
11    {
12        'x':125,
13        'y':250
14    },
15    {
16        'x':200,
17        'y':290
18    },
19    {
20        'x':275,
21        'y':250
22    },
23    {
24        'x':275,
25        'y':160
26    },
27    {
28        'x':195,
29        'y':110
30    }
31];
32 var svg = d3.select('#time-series')
33     .append('svg')
```

Browser Window (Right):

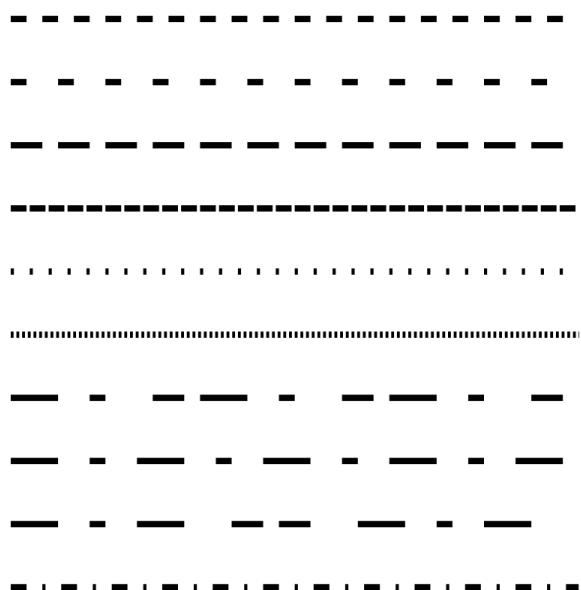
The browser title bar says "D3". The address bar shows "localhost". The page content area has a blue header "Intro to D3".

```
(function() {  
  var svg = d3.select('#time-series').append('svg')  
    .attr('width', 1000)  
    .attr('height', 1000);  
  var pointData = [{x:205,y:110}, {x:125, y:160},  
    {x:125, y:250}, {x:200, y:290},  
    {x:275, y:250}, {x:275, y:160},{x:195,y:110}];  
  var line = d3.svg.line()  
    .interpolate('linear')  
    .x(function(d) { return d.x; })  
    .y(function(d) { return d.y; });
```



```
▼<div class="mdl-shadow--2dp mdl-cell mdl-cell--12-col">  
  ▼<div id="time-series">  
    <script src="/js/scratchpad6.js"></script>  
    ▼<svg width="1000" height="1000">  
      <path d=  
        "M205,110L125,160L125,250L200,290L275,250L275,  
        160L195,110" stroke-width="22" stroke=  
        "rgb(47,54,169" fill="none" stroke-dasharray=  
        "80 50" stroke-dashoffset="0"></path>  
    </svg>  
  </div>  
</div>  
</div>  
</main>  
<div class="mdl-layout__obfuscator"></div>  
</div>
```

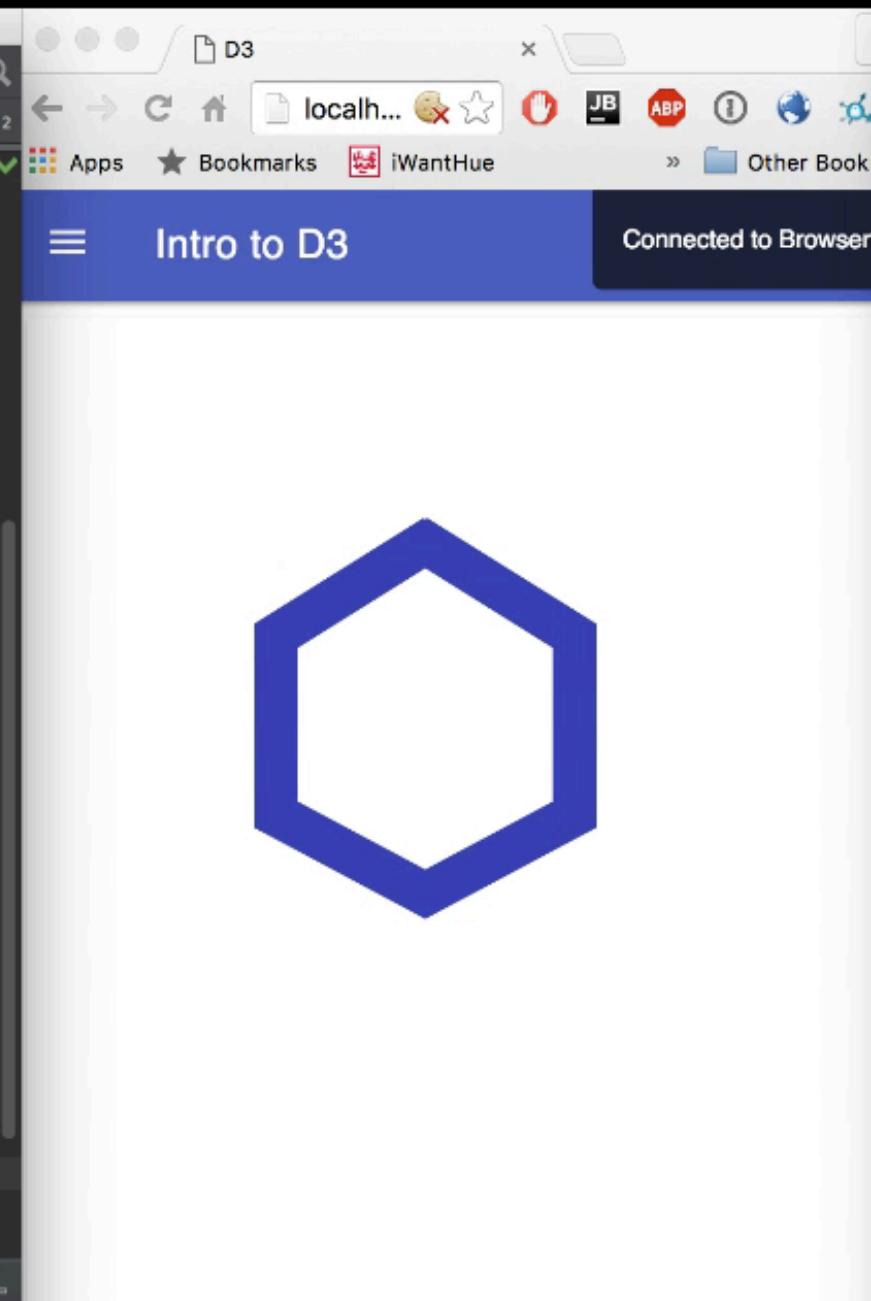
stroke-dasharray



```
<line stroke-dasharray="5, 5"
<line stroke-dasharray="5, 10"
<line stroke-dasharray="10, 5"
<line stroke-dasharray="5, 1"
<line stroke-dasharray="1, 5"
<line stroke-dasharray="0.9"
<line stroke-dasharray="15, 10, 5"
<line stroke-dasharray="15, 10, 5, 10"
<line stroke-dasharray="15, 10, 5, 10, 15"
<line stroke-dasharray="5, 5, 1, 5"
```

<http://localhost:3000/scratchpad/6>

```
18     },
19     {
20         'x':275,
21         'y':250
22     },
23     {
24         'x':275,
25         'y':160
26     },
27     {
28         'x':195,
29         'y':110
30     }
31 ];
32 var svg = d3.select('#time-series')
33     .append('svg')
34     .attr('width', 1000)
35     .attr('height', 1000);
36
37 var line = d3.svg.line()
38     .interpolate('linear')
39     .x(function(d) { return d.x;})
40     .y(function(d) { return d.y;});
41
42 var path = svg.append('path')
43     .attr('d', line(pointData))
44     .attr('stroke-width', 22)
45     .attr('stroke', 'rgb(47,54,169')
46     .attr('fill', 'none');
47
48
49
50 }());
```



HOMER Pro Microgrid Analysis Tool [Chevak4 - Efficiency Measures.homer] nightly #113 3.2.5584

FILE **Design** **Results** **Library** **LOAD** **COMPONENTS** **RESOURCES** **SYSTEM** **HELP**

Calculate

SCHEMATIC

ELECTRIC LOAD

January Profile

Hour	Load (kW)
0	300.18
1	261.66
2	231.2
3	221.31
4	211.63
5	209.36
6	213.71
7	217.61
8	268.71
9	329.25
10	353.9
11	370.99
12	380.36
13	375.53
14	375.53

Daily Profile

Seasonal Profile

Yearly Profile

SUGGESTIONS:

Design

Remove

Metric **Baseline** **Scaled**

Average (kWh/d)	6,818.7	6,818.7
Average (kW)	284.12	284.12
Peak (kW)	519.4	519.4
Load Factor	.55	.55

Efficiency (Advanced)

Efficiency multiplier:

Capital cost (\$):

Lifetime (yr):

Load Type: AC DC

Scaled Annual Average (kWh/d):

Plot... **Export...**

HOMER PRO

HOMER Pro Microgrid Analysis Tool [hhh.homer] nightly #200 1.0.5898

FILE LOAD COMPONENTS RESOURCES PROJECT SYSTEM HELP

Home Design Results Library View Multi-Year Inputs Search Space Sensitivity Inputs Input Report Estimate Clear Results Calculate

SCHEMATIC

Gen1200 AC Estimated Load 72000.00 kWh/d 5592.54 kW peak converter Grid PV DC Li-ion

ELECTRIC LOAD

Name: Estimated Load Remove

January Profile Daily Profile Seasonal Profile

Hour	Load (kW)
0	2,100,000
1	2,100,000
2	2,100,000
3	2,100,000
4	2,100,000
5	2,100,000
6	2,100,000
7	3,300,000
8	4,500,000
9	4,500,000

Yearly Profile

Time Step Size: 60 minutes Random Variability Day-to-day (%): 5 Timestep (%): 5 Metric Baseline Scaled Average (kWh/d) 72,000 72,000 Average (kW) 3,000 3,000 Peak (kW) 5,592.5 5,592.5 Load Factor .54 .54 Efficiency (Advanced) Efficiency multiplier: 1 Capital cost (\$): 0 Lifetime (yr): 10

Show All Months... Plot... Export...

AC DC

Peak Month: January Scaled Annual Average (kWh/d): 72,000.00

SUGGESTIONS:

Model does not match results Newer version of HOMER Pro available

HOMER PRO

DESIGN



RESULTS

 Tabular Graphical

Optimization Surface Plot ▾

— Sensitivity Variables —

{Li-ion B&V: Capital Cost Multiplier (*), Li-ion B&V: O&M Cost (*), ...} {0.90, 0.90, 0.90} ▾

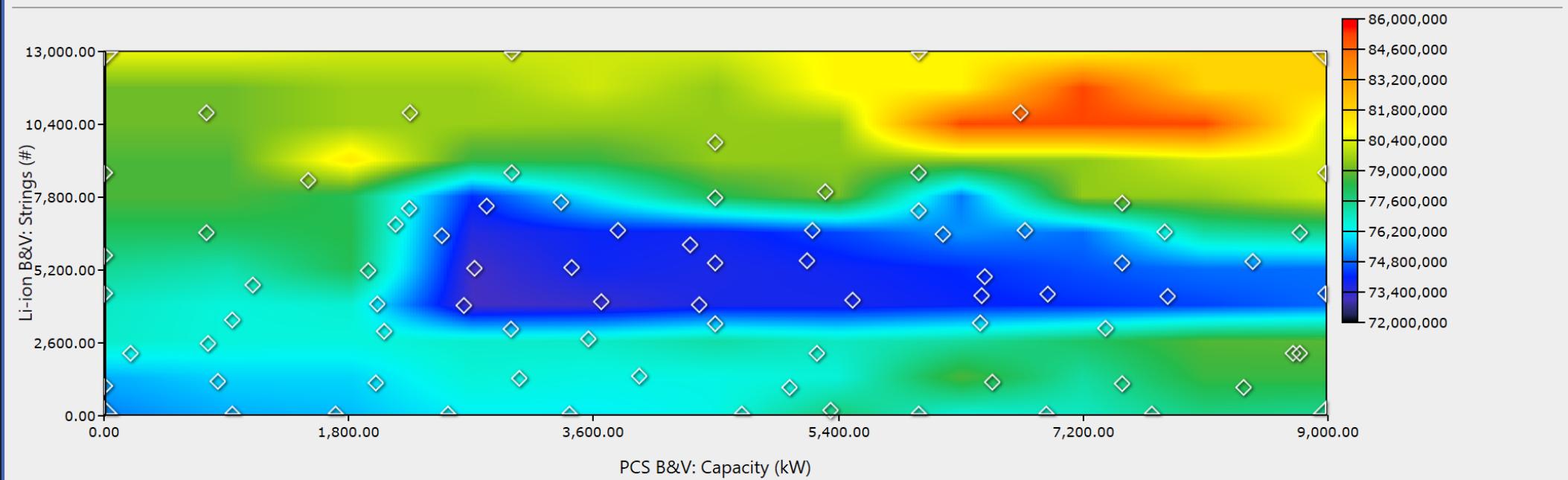
— Optimization Variables —

Gen1200: Capacity (kW) 3600 ▾ Dispatch Strategy CC ▾ Li-ion B&V: Strings (#) y-axis ▾ PV: Capacity (kW) 16000 ▾

PCS B&V: Capacity (kW) x-axis ▾

— Variables to Plot —

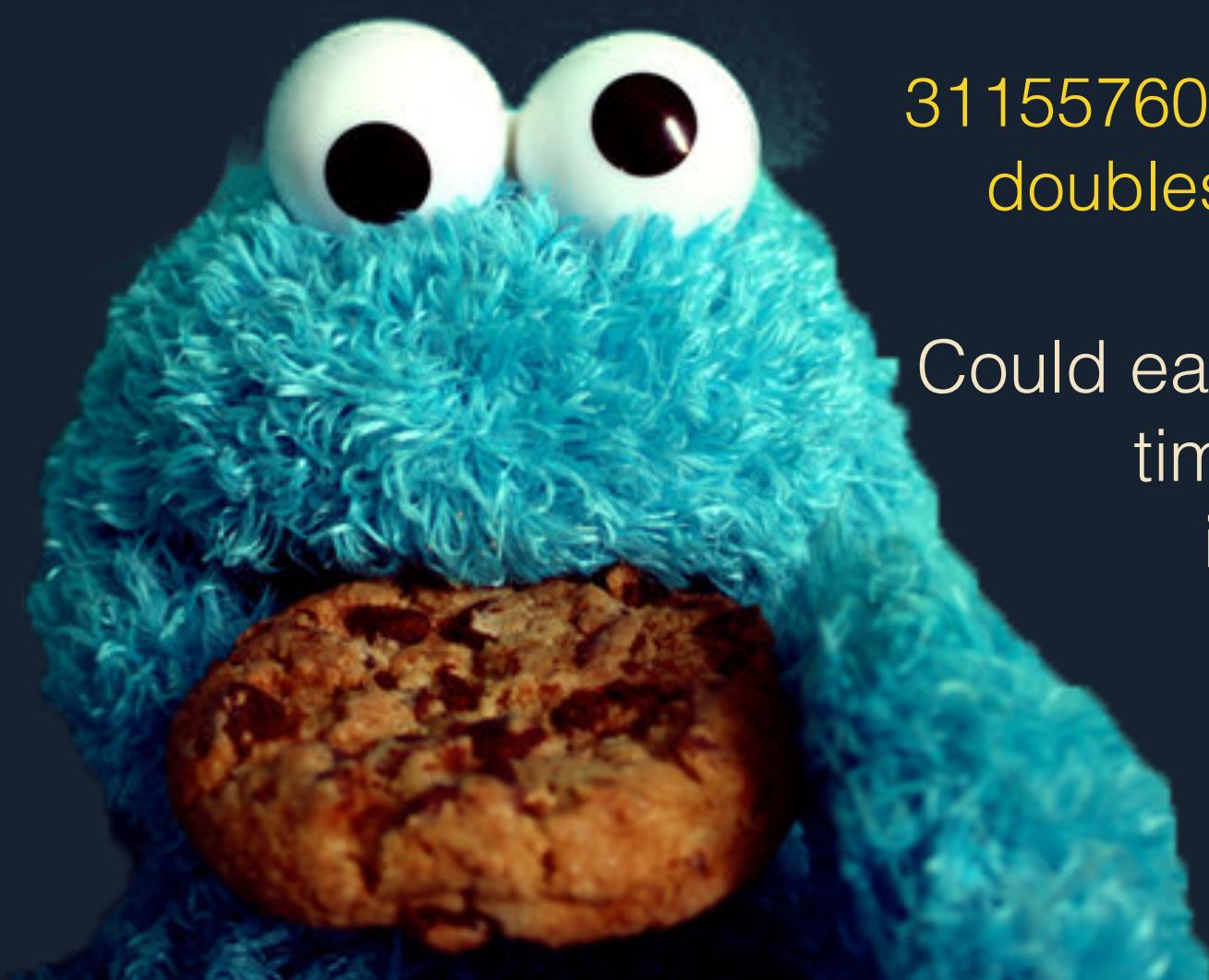
Surface Total Net Present Cost ▾ Superimposed <none> ▾





Give me data

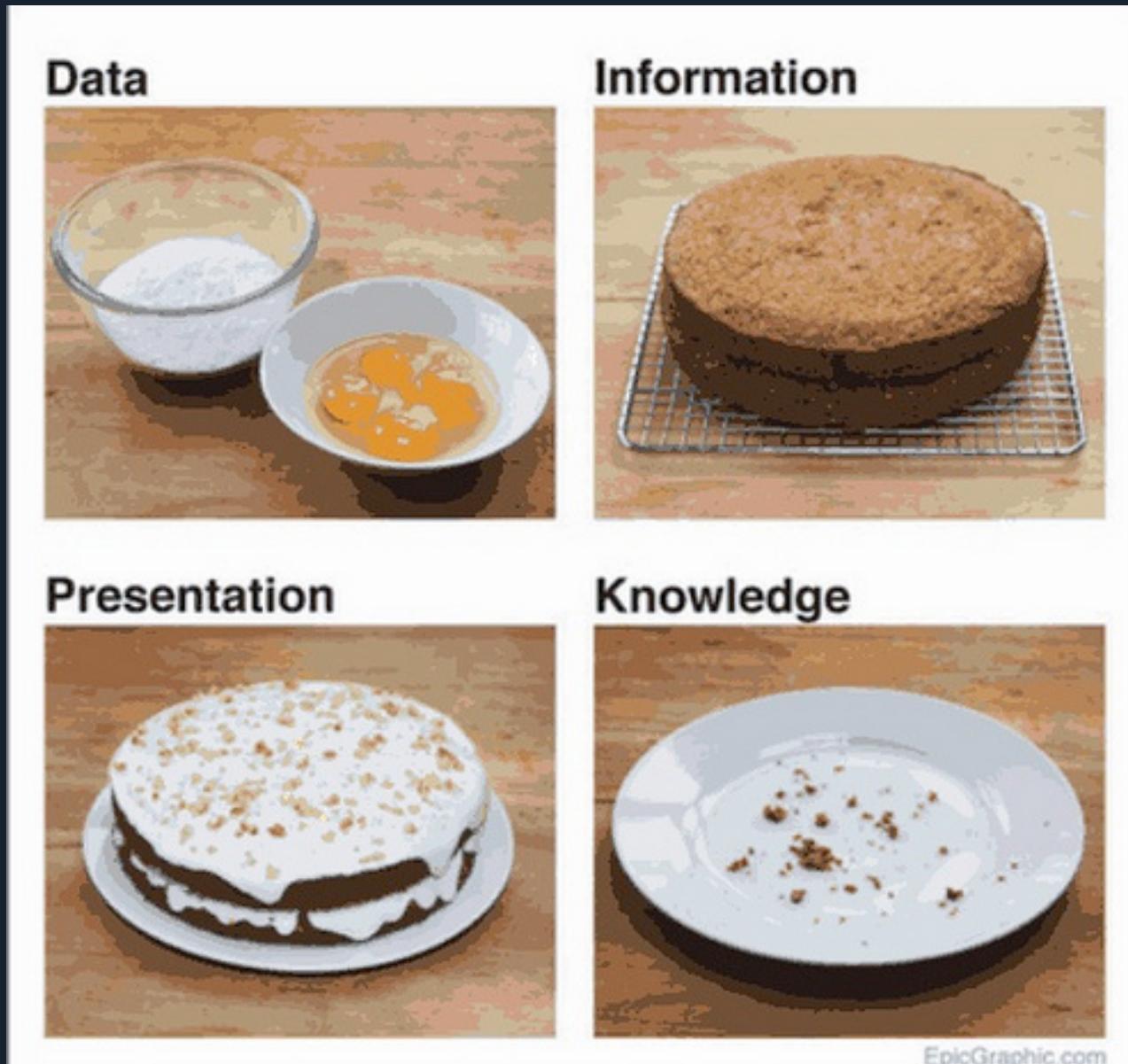
1 year = 365 days * 24 hours/day * 3600 seconds/hour =
311557600 seconds/year

A close-up photograph of the Cookie Monster from Sesame Street. He has his signature large, white, bulging eyes and a blue, textured, fur-like body. He is holding a single, round, brown cookie with chocolate chips in his right hand.

311557600 javascript Numbers =
doubles = 252.4 megabytes

Could easily have 50 important
time series arrays
in a project =
12 gigabytes

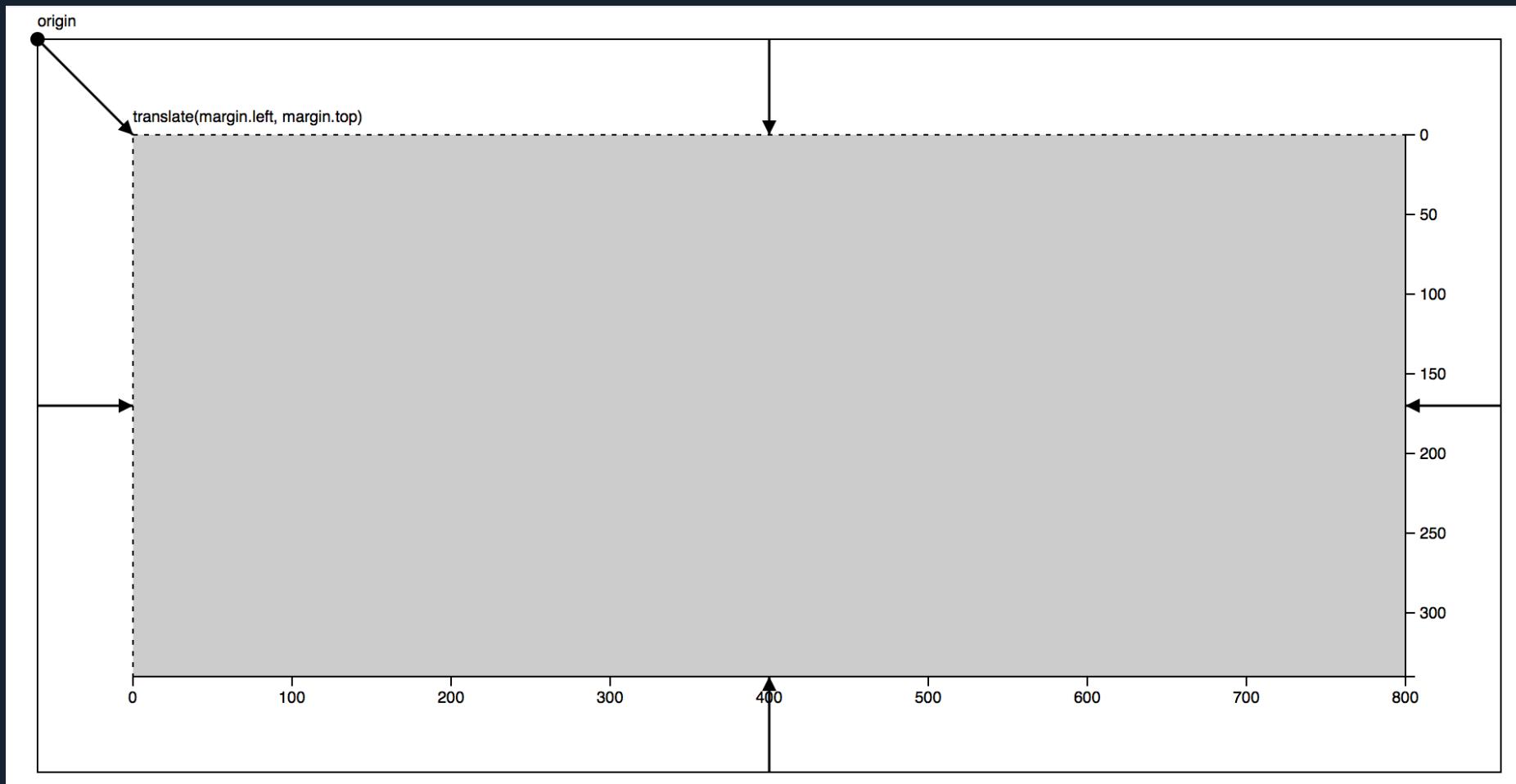
Data is not information



<http://epicgraphic.com/data-cake/>

EpicGraphic.com

Margin Convention



```
var margin = {  
    top: 40,  
    right: 0,  
    bottom: 40,  
    left: 50  
};  
var width = 600 - margin.left - margin.right;  
var height = 400 - margin.top - margin.bottom;
```

AC Electrical Load. Yakutat, Alaska (kW)



g 209487px x 363.938px

```
<div class="data-content">
  <div class="mdl-shadow--2dp mdl-cell mdl-cell--12-col">
    <div id="time-series">
      <script src="/js/timeSeries0.js"></script>
    <svg width="600" height="400">
      <g transform="translate(50,40)">
        <path d=
          "M0,113.7374933333333L3.98550724637
          6811,115.4618488888888C7.9710144927
          53622,117.18620444444443,15.94202898
          5507244,120.63491555555554,23.913043
          478260867,119.90115555555553C31.8840
          5797101449,119.16739555555553,39.855
          07246376811,114.2511644444441,47.82
          6086956521735,107.1702933333333C55.7
          9710144927536,100.089422222222,63.7
          6811594202898,90.843911111111,71.73
          913043478261,84.6802488888888C79.71
          014492753622,78.51658666666667,87.68
          115942028984,75.43477333333334,95.65
          217391304347,64.24480000000001C103.6
          231884057971,53.0548266666666,111.5
          942028985507,33.7566933333331,119.5
          6521739130434,27.95991111111083C127
          .53623188405795,22.16312888888885,13
          5.50724637681157,29.86769777777742,
          113.17826086056522,36.83850666666666613
```

... body div div main div div #time-series svg g
Styles Event Listeners DOM Breakpoints Properties

```
var svg = d3.select('#time-series')
  .append('svg')
  .attr('width', width + margin.left + margin.right)
  .attr('height', height + margin.top + margin.bottom)
  .append('g')
  .attr('transform', 'translate(' + margin.left + ','
  + margin.top + ')');
```

```
3
4     var margin = {
5         top: 40,
6         right: 0,
7         bottom: 40,
8         left: 40
9     };
10    var width = 600 - margin.left - margin.right;
11    var height = 400 - margin.top - margin.bottom;
12
13    var svg = d3.select('#time-series')
14        .append('svg')
15        .attr('width', width + margin.left + margin.right)
16        .attr('height', height + margin.top + margin.bottom)
17        .append('g')
18        .attr('transform', 'translate(' + margin.left + ','
19            + margin.top + ')');
20
21
22    })();
23
```

Terminal

```
+ [17:26:53] [nodemon] restarting due to changes...
x restarting...
x [17:26:53] [nodemon] starting `node server/app.js`
running server on port 8080
[gulp-patterns] File changed: wwwroot/js/scratchpad11.js
[17:28:04] [nodemon] restarting due to changes...
x restarting...
x [17:28:04] [nodemon] restarting due to changes...
x restarting...
x [17:28:04] [nodemon] starting `node server/app.js`
running server on port 8080
[20:21:55] [nodemon] restarting due to changes...
x restarting...
x [20:21:55] [nodemon] restarting due to changes...
```





Components

```
var xScale = d3.scale.linear()  
    .range([0, width]);  
xScale.domain([0,23]);
```



```
><g class="tick" transform=  
"translate(0,360)" style="opacity: 1;">  
...</g>  
▼<g class="tick" transform=  
"translate(0,307.79146475287575)" style=  
"opacity: 1;">  
  <line x2="-6" y2="0"></line>  
  <text dy=".32em" x="-9" y="0" style=  
  "text-anchor: end;">50</text>  
  </g>  
▼<g class="tick" transform=  
"translate(0,255.5829295057516)" style=  
"...
```

Data is Asynchronous

Code that depends on data must be invoked via callback.

```
// Method One
d3.json(url, function (error, json) {
  if (error) {
    return console.error(error);
  }
  // do something with jsonFile
}

// Method Two
var myData;
var callback = function (error, jsonFile) {
  if (error) {
    return console.error(error);
  }
  // do something with jsonFile
  myData = jsonFile;
}
d3.json(url, callback);
```

The screenshot shows a code editor interface with the following details:

- Title Bar:** timeSeries0.js - src - [~/__Code/_d3_workshop/src]
- Project Explorer:** Shows 'src' and 'wwwroot' with 'js' and 'timeSeries0.js' selected.
- Code Editor:** Displays the content of 'timeSeries0.js'. The code defines two scales (yScale and xScale) and two axes (yAxis and xAxis) using d3.js methods like scale.linear() and axis().
- Toolbars:** Includes tabs for 'timeSeries0.js', 'scratchpad11.js', 'scratchpad4.js', 'scratchpad3.js', and 'timeSeries6.js'.
- Console:** Shows the output of the application's execution:
 - Initial logs from 'Object.Module._extensions..js' and 'Module.load'.
 - A crash message: '[20:25:23] [nodemon] app crashed – waiting for file changes before starting...'.
 - File change detection: '[gulp-patterns] File changed: wwwroot/js/timeSeries0.js'.
 - Restarting process: '[20:32:06] [nodemon] restarting due to changes...', '[20:32:06] [nodemon] starting `node server/app.js`', and '[20:32:06] [nodemon] restarting due to changes...'.
- Bottom Bar:** Includes tabs for 'Terminal', 'Logs', 'Info', 'Debug', and 'Handled'.

src (~/_Code/_d3_workshop)

- node_modules (library host)
- server
- views
- wwwroot
 - app
 - css
 - data
 - boxplots
 - colors
 - timeseries
 - ac_load.json
 - alaska_sun.json
 - battery.json
 - boulder_sun.json
 - wind.json
 - hourly_load_profile
 - js
 - test.html
- .babelrc
- .editorconfig
- .jscsrc
- .jshintrc
- gulpfile.babel.js
- package.json
- External Libraries

title

```
1 {  
2   "title": "AC Electrical Load. Yakutat, Alaska",  
3   "units": "kW",  
4   "data": [  
5     {  
6       "date": "01/01/2015 00:00",  
7       "value": 193.3711  
8     },  
9     {  
10       "date": "01/01/2015 01:00",  
11       "value": 183.6716  
12     },  
13     {  
14       "date": "01/01/2015 02:00",  
15       "value": 197.4985  
16     },  
17     {  
18       "date": "01/01/2015 03:00",  
19       "value": 223.5015  
20     },  
21     {  
22       "date": "01/01/2015 04:00",  
23       "value": 232.1691  
24     },  
25     {  
26       "date": "01/01/2015 05:00",  
27       "value": 286.4451  
28     }  
]
```

A screenshot of a developer's workspace, likely from a web-based IDE or code editor. The interface includes a top navigation bar, a left sidebar for project structure, and several tabs for code files.

Project Structure: Shows a tree view with 'src' as the root, containing 'wwwroot' and 'js'. 'js' contains 'timeSeries0.js'.

Code Editor: The main area displays the content of 'timeSeries0.js'.

```
20     var yScale = d3.scale.linear()
21         .range([height, 0]);
22     var yAxis = d3.svg.axis()
23         .scale(yScale)
24         .orient('left');
25
26     var xScale = d3.scale.linear()
27         .range([0, width]);
28     var xAxis = d3.svg.axis()
29         .scale(xScale)
30         .orient('bottom');
31
32     d3.json('/data/timeseries/ac_load.json', function (error, json)
33     if (error) throw error;
34     x|
35
36
37     })
38   })
39
40 })();
41
```

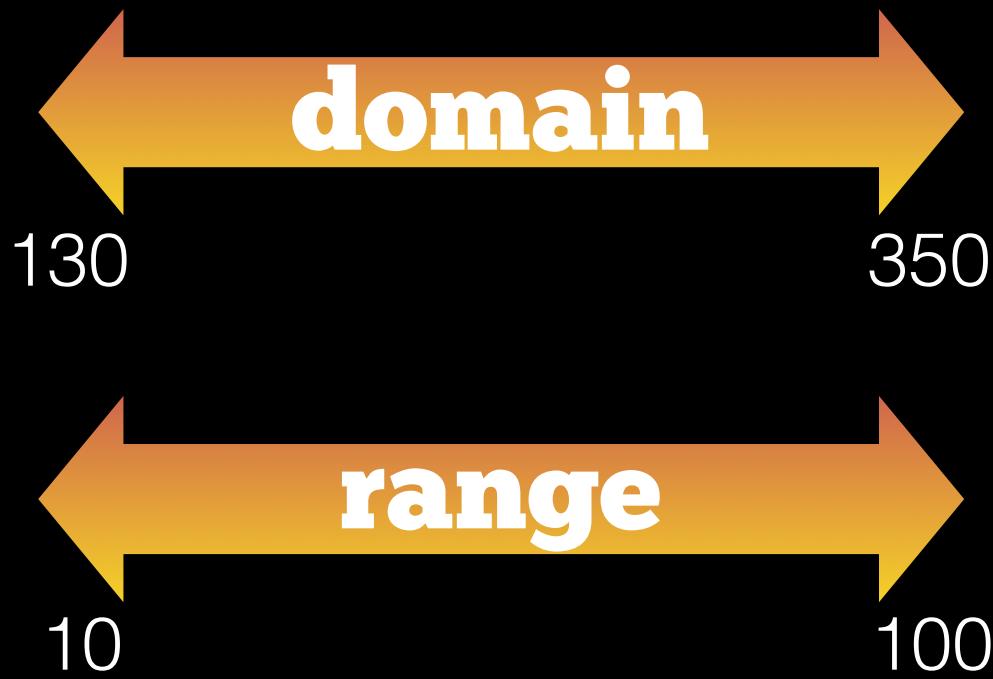
Terminal: Shows the output of a command, indicating a crash during app startup.

```
+ server/app.js:93:5)
    at Module._compile (module.js:413:34)
    at Object.Module._extensions..js (module.js:422:10)
    at Module.load (module.js:357:32)
[20:58:21] [nodemon] app crashed - waiting for file changes before
starting...
```

The screenshot shows a developer environment with several windows open:

- Code Editor:** The main window displays a JavaScript file named `timeSeries0.js`. The code uses D3.js to create a line chart. A tooltip is visible over the word `svg`, listing several definitions from the current file and its dependencies.
- Terminal:** The bottom-left window shows the output of a Node.js application running with `nodemon`. It logs messages about restarting due to changes and starting the server.
- Browser DevTools:** An inspection window is open, showing the DOM structure of the page. The `body` element contains a script tag for `browser-sync-client` and a container for the MDL layout.
- Console:** The bottom-right window shows the browser's developer console. It includes a network tab, a log tab with a message about navigating to the local host, and an errors tab with a syntax error from line 56 of `timeSeries0.js`.

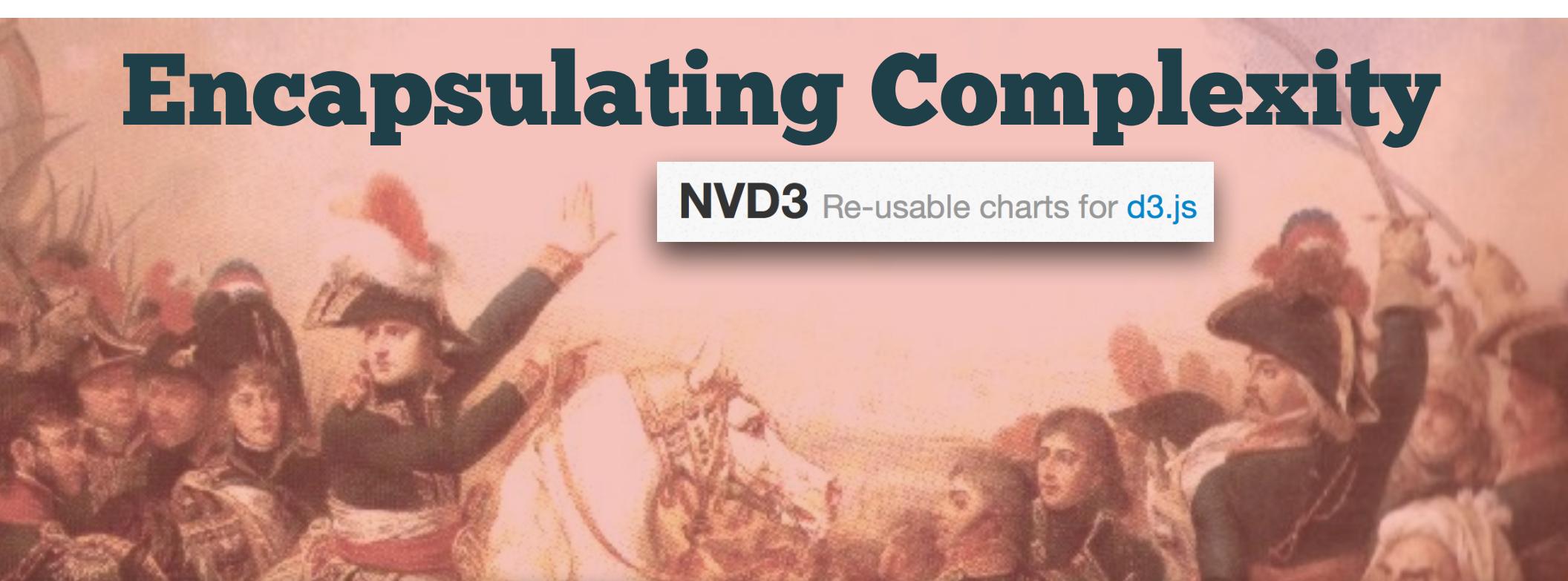
Quantitative Scales: map from an input domain to an output range.



A screenshot of a browser's developer tools console, specifically the "Console" tab. The interface includes a header with "Console", a dropdown for "Preserve log", and tabs for "All", "Errors", "Warnings", "Info", "Logs", "Debug", and "Handled". Below the tabs is a text input field for "Filter" with checkboxes for "Regex" and "Hide network messages". The main area shows a command being typed and its execution results:

```
var scale = d3.scale.linear()  
.domain([130,350]) // data min/max  
.range([10,100]); // output pixels for SVG  
< undefined
```

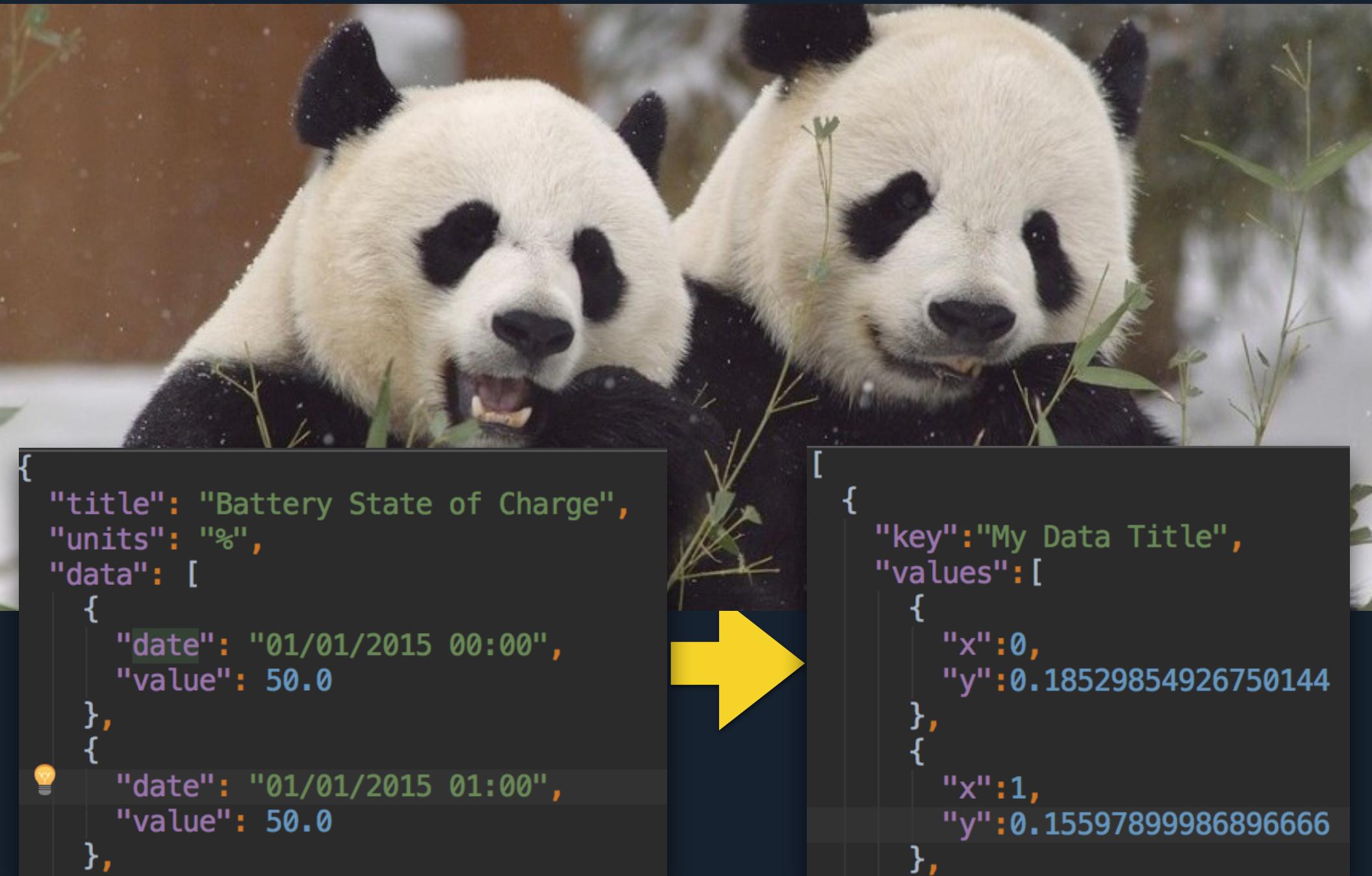
Encapsulating Complexity



NVD3 Re-usable charts for [d3.js](#)

NVD3's codebase is heavily inspired by the work of Mike Bostock. In particular, his article "[Towards Reusable Charts](#)" serves as our guide.

Data Munging



A large yellow arrow points from the left code block to the right code block, indicating a transformation or comparison between the two pieces of JSON data.

```
{  
  "title": "Battery State of Charge",  
  "units": "%",  
  "data": [  
    {  
      "date": "01/01/2015 00:00",  
      "value": 50.0  
    },  
    {  
      "date": "01/01/2015 01:00",  
      "value": 50.0  
    },  
    {  
      "date": "01/01/2015 02:00",  
      "value": 50.0  
    },  
    {  
      "date": "01/01/2015 03:00",  
      "value": 50.0  
    },  
    {  
      "date": "01/01/2015 04:00",  
      "value": 50.0  
    }  
  ]}  
  
[  
  {  
    "key": "My Data Title",  
    "values": [  
      {  
        "x": 0,  
        "y": 0.18529854926750144  
      },  
      {  
        "x": 1,  
        "y": 0.15597899986896666  
      }  
    ]}  
]
```

```
// {
//   "title": "AC Electrical Load. Yakutat, Alaska",
//   "units": "kW",
//   "data": [
//     {
//       "date": "01/01/2015 00:00",
//       "value": 193.3711
//     }
//   ]
// }

d3.json('/data/ac_load.json', function(error, json) {
  var displayData = [{key: json.units, values: []}];
  json.data.forEach(function(el, i) {
    var format = d3.time.format('%m/%d/%Y %H:%M');
    var d3Date = format.parse(el.date);
    displayData[0].values.push({x: d3Date, y: el.value});
  });
})
```

- `%m` - month as a decimal number [01,12].
- `%d` - zero-padded day of the month as a decimal number [01,31].
- `%Y` - year with century as a decimal number.
- `%H` - hour (24-hour clock) as a decimal number [00,23].
- `%M` - minute as a decimal number [00,59].

timeseries8.js - src - [~/__Code/_d3_workshop/src]

src wwwroot js timeseries8.js

timeseries8.js ac_load.json

Unamed

```
4     var displayData = [{key: json.units, values: []}];  
5     json.data.forEach(function(el, i) {  
6         var format = d3.time.format('%m/%d/%Y %H:%M');  
7         var d3Date = format.parse(el.date);  
8         displayData[0].values.push({x: d3Date, y: el.value});  
9     });  
10    var svg = d3.select('#time-series').append('svg')  
11        .attr('height', 500);  
12  
13    nv.addGraph(function() {  
14        var chart = nv.models.lineWithFocusChart();  
15        chart.xAxis  
16            .tickFormat(function(d) {  
17                return d3.time.format('%b %-d %-I%p')(new Date(d));  
18            });  
19        chart.xScale(d3.time.scale());  
20        chart.yAxis.tickFormat(d3.format('.1f'));  
21        chart.x2Axis  
22            .tickFormat(function(d) {  
23                return d3.time.format('%b %-d')(new Date(d));  
24            });  
25        chart.margin({bottom:80});  
26        chart.xAxis.rotateLabels(-45);  
27        var minTime = displayData[0].values[0].x;  
28        var maxTime = d3.time.hour.offset(minTime, 24 * 7);  
29        chart.brushExtent([minTime, maxTime]);  
30        svg.datum(displayData)  
31            .transition().duration(500)  
32            .call(chart);  
33  
34        var title = json.title + ' (' + json.units + ')';  
35        svg.append('text')  
36            .attr('x', 300)  
37            .attr('y', 25)
```

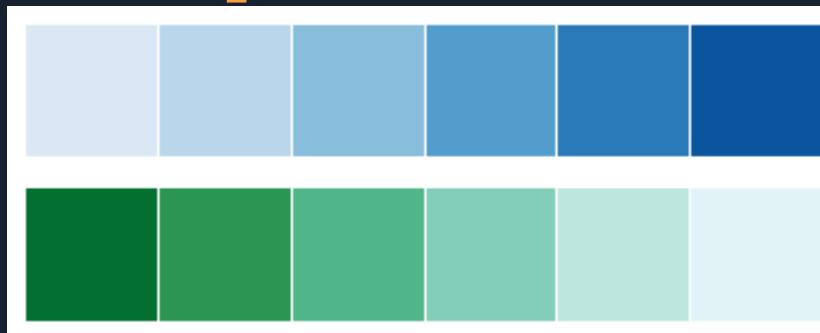
Color Palettes



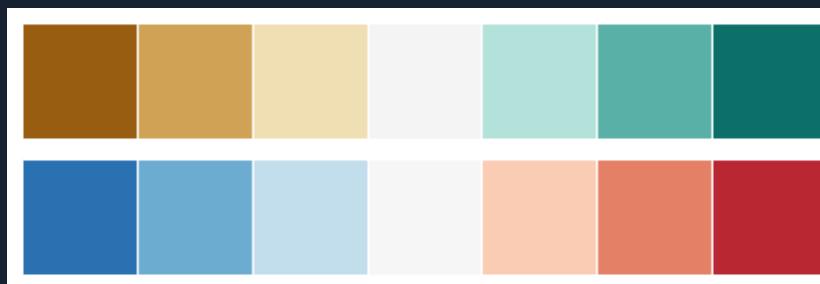
[https://commons.wikimedia.org/wiki/File:Vincent_Willem_van_Gogh_-_Cafe_Terrace_at_Night_\(Yorck\).jpg](https://commons.wikimedia.org/wiki/File:Vincent_Willem_van_Gogh_-_Cafe_Terrace_at_Night_(Yorck).jpg)

Quantitative Color Palettes

Sequential Palettes



Diverging Palettes



Picking a Sequential Palette

Step 1: Try Colorbrewer



Intro to Heat Maps



Jon Skeet

jskeet

Google

London, UK

skeet@pobox.com

<http://jonskeet.uk>

Joined on Jul 14, 2008

1k

1

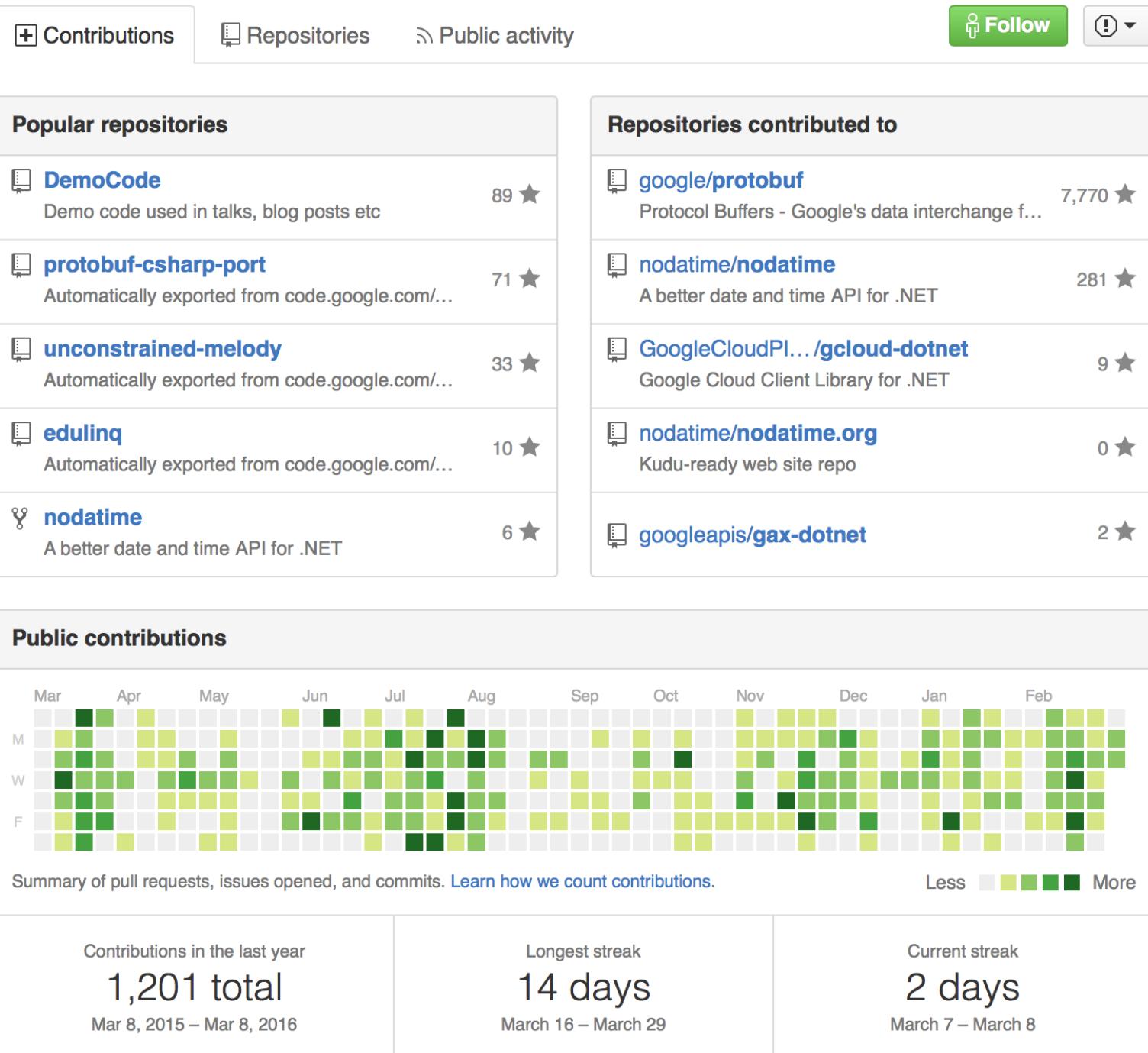
0

Followers

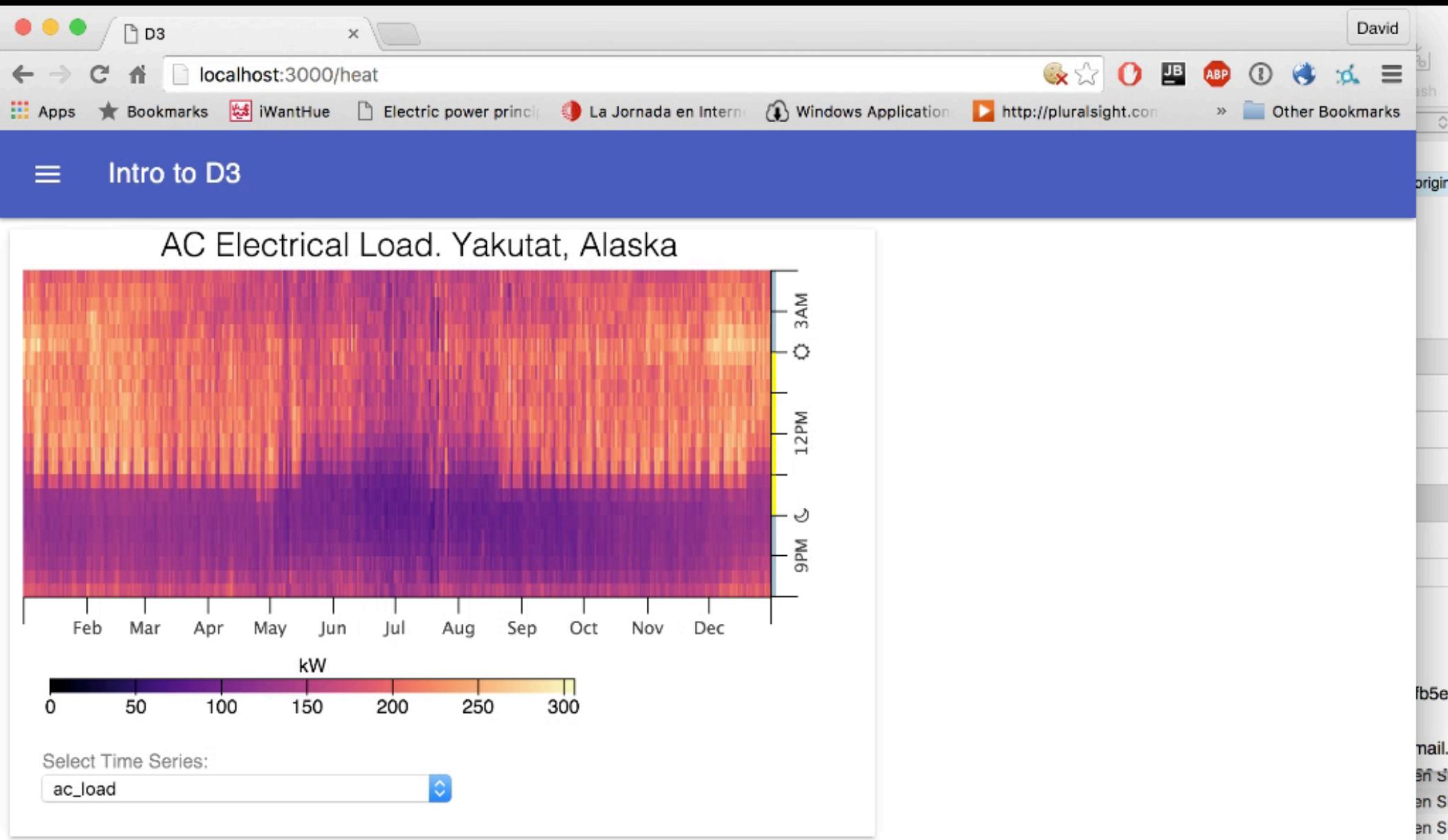
Starred

Following

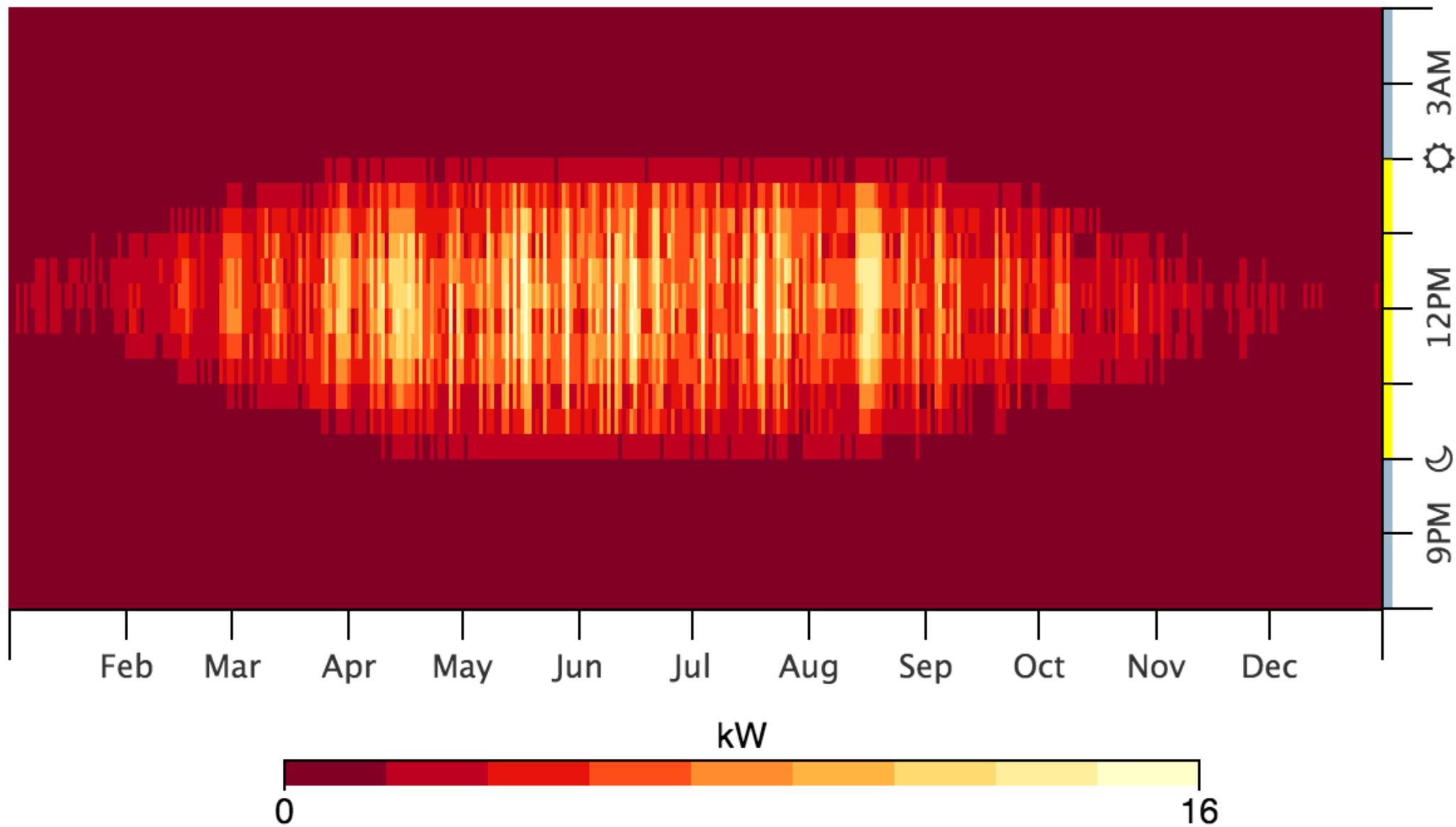
Organizations



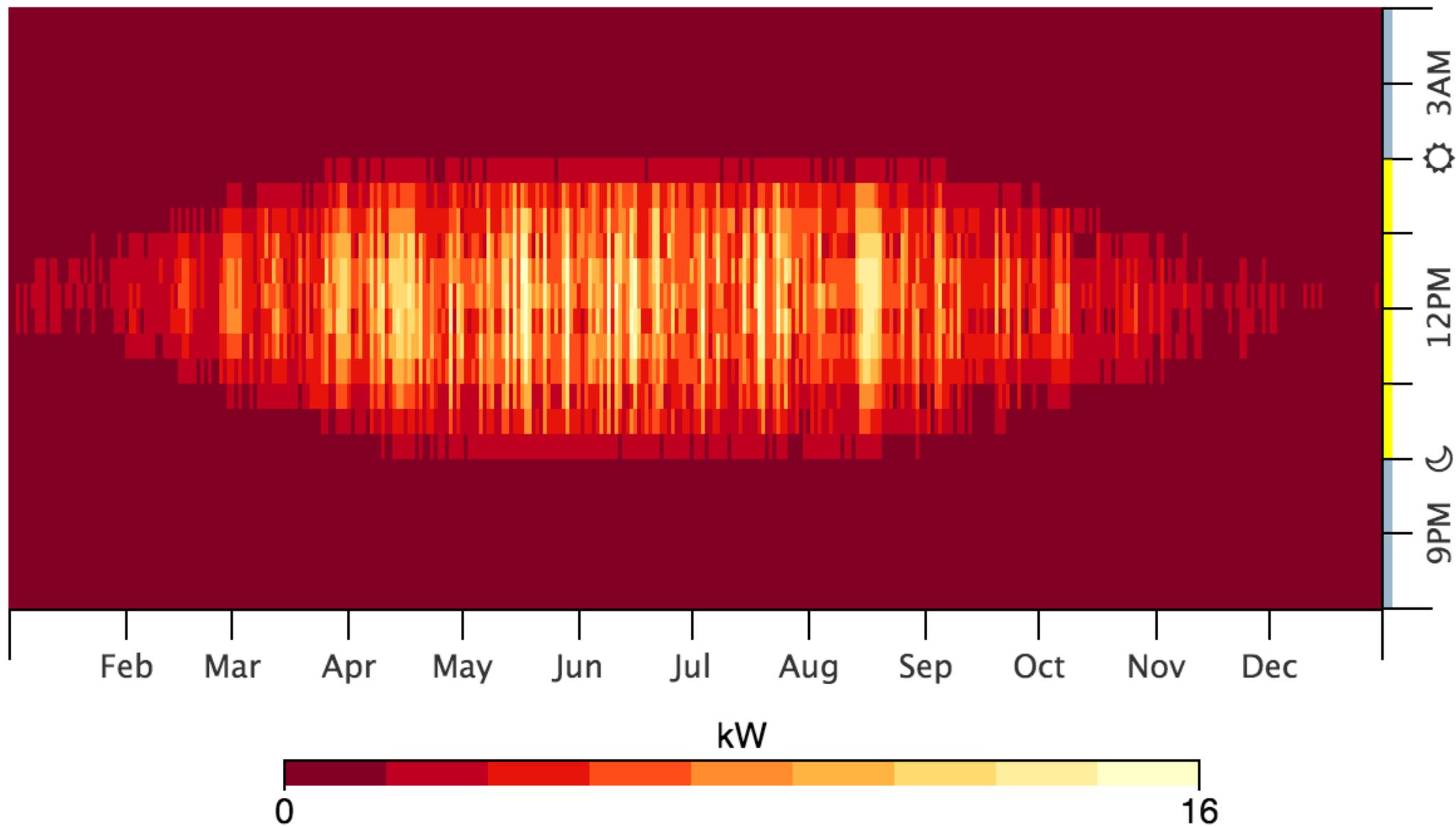
http://localhost:3000/heat



Developing a color palette for heat maps

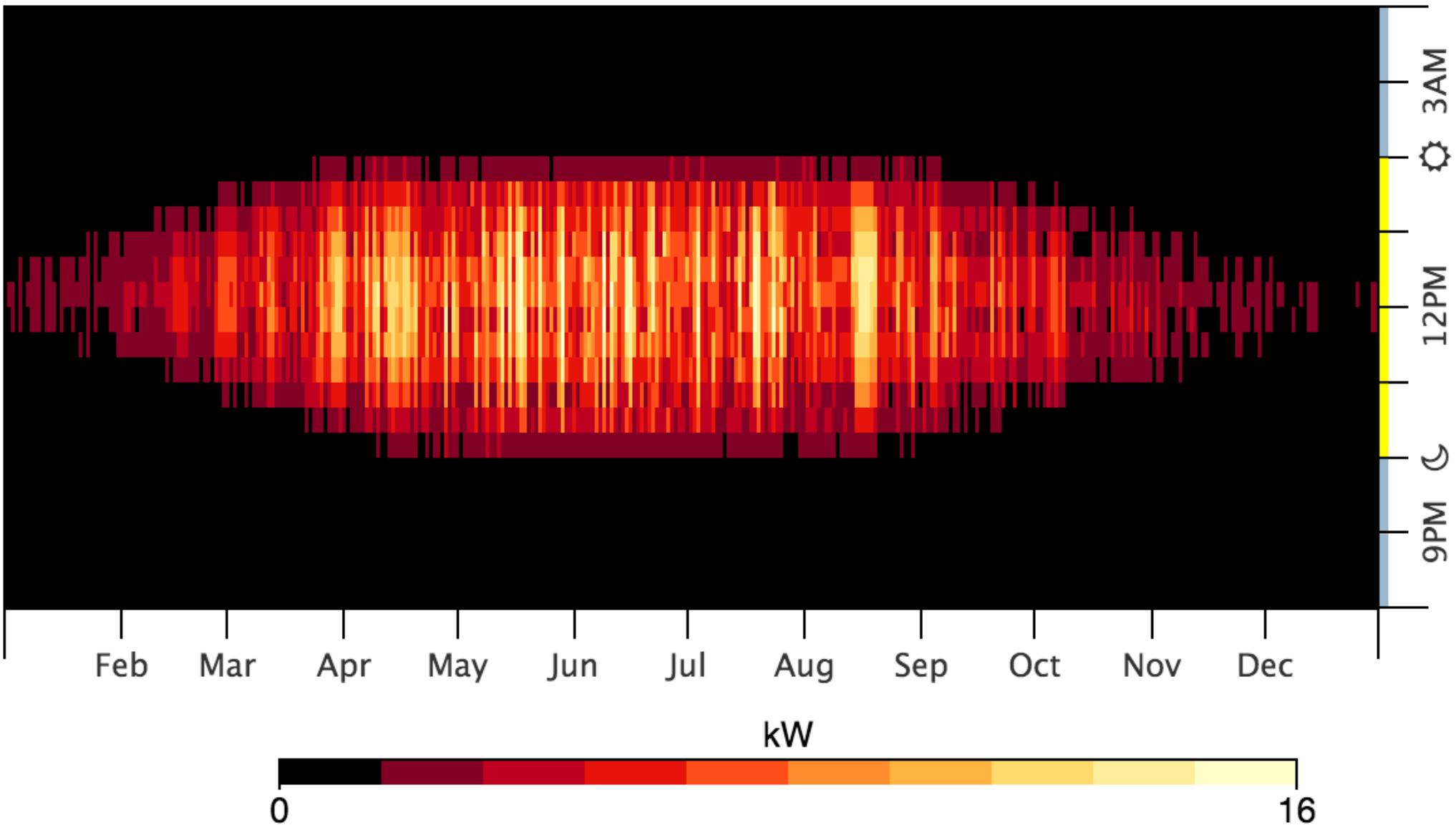


Sunny Alaska?

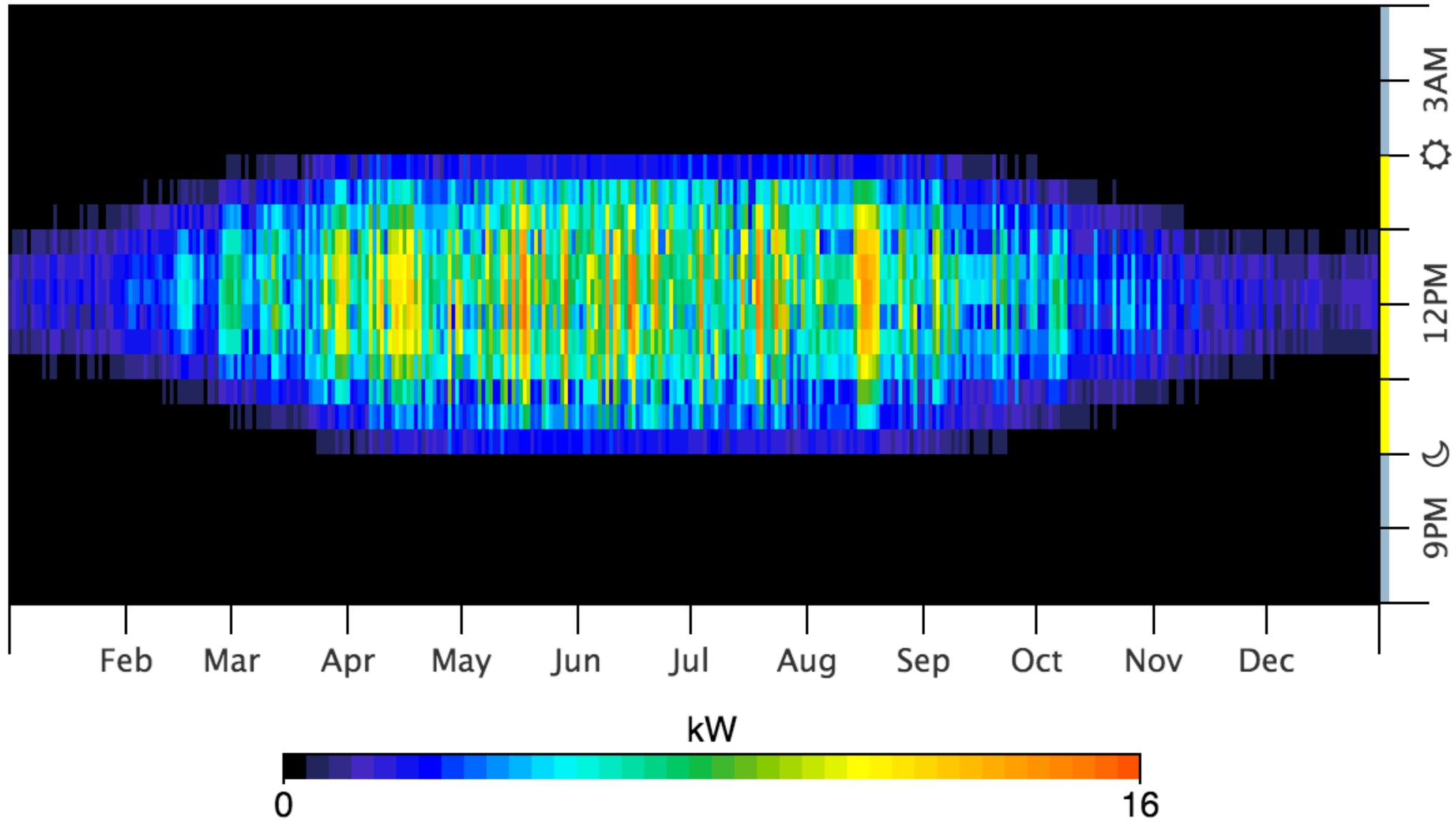


This heat map shows solar output from a PV array in Alaska

better zero-crossing



go rainbow



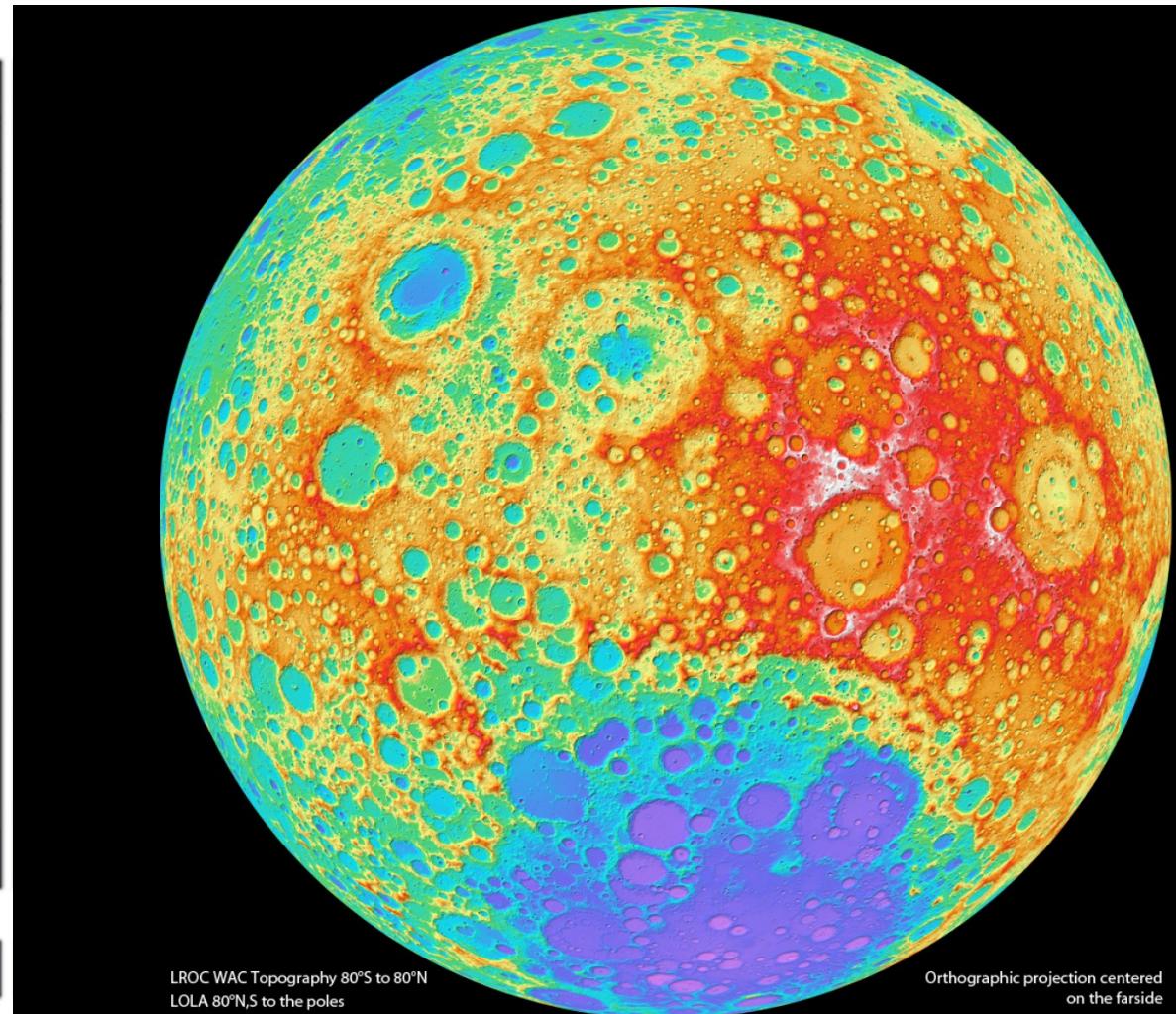
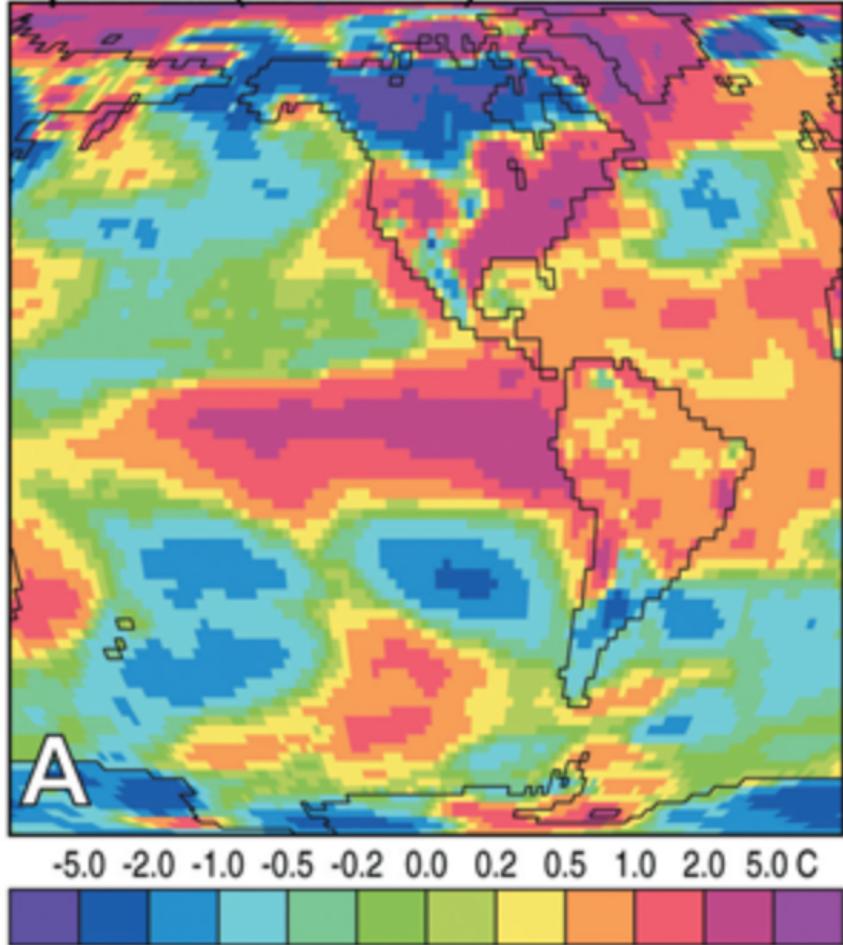
Dear NASA: No More Rainbow Color Scales, Please

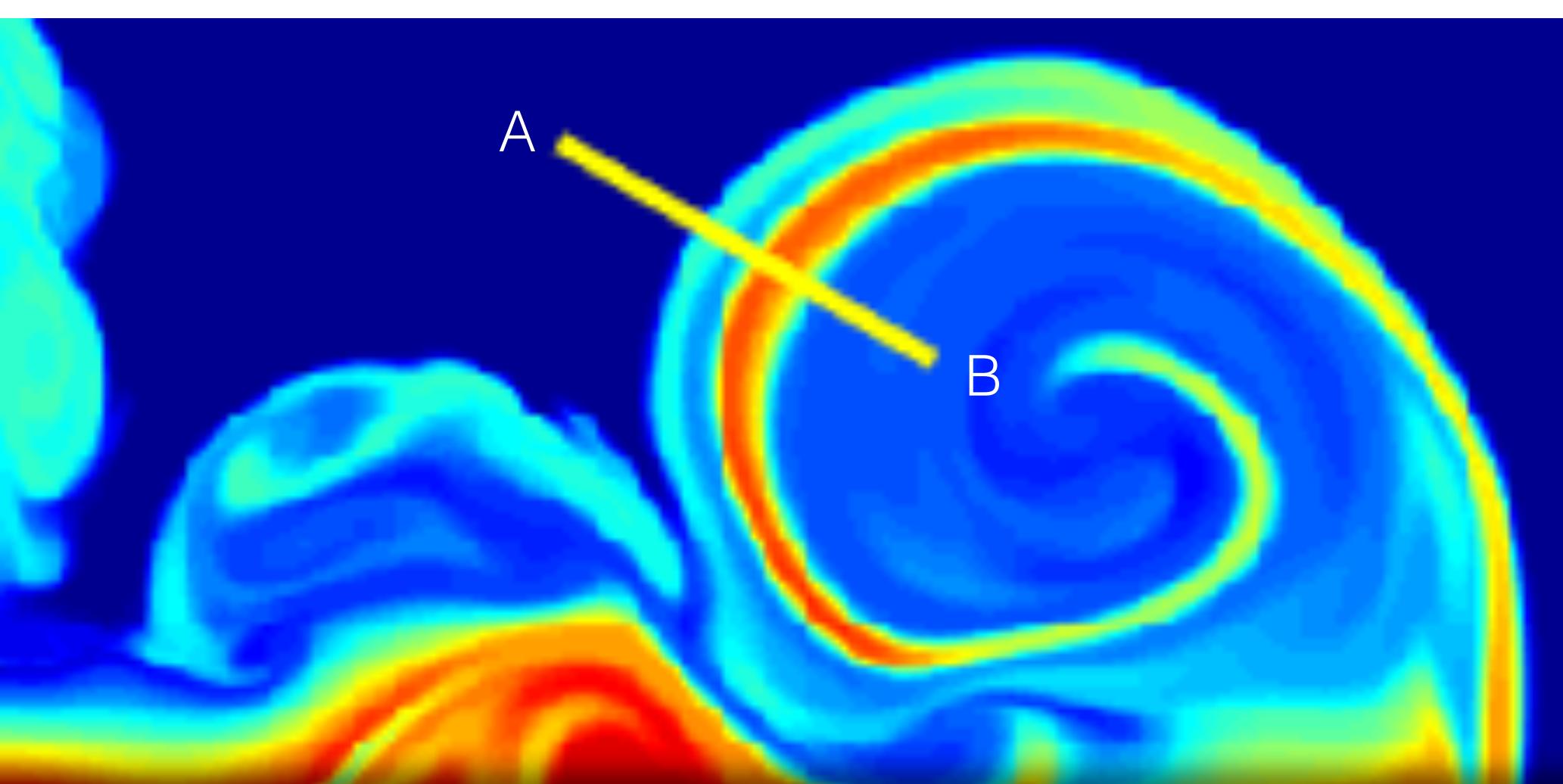


Drew Skau

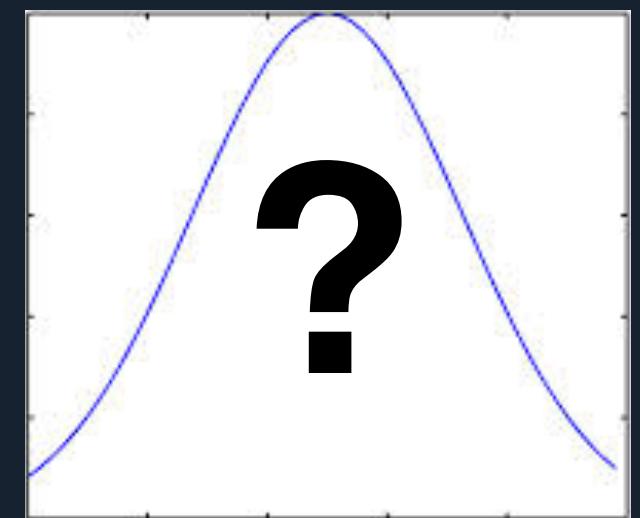
published on April 4, 2012 in **Design**

Spectral (Rainbow) Color Scale

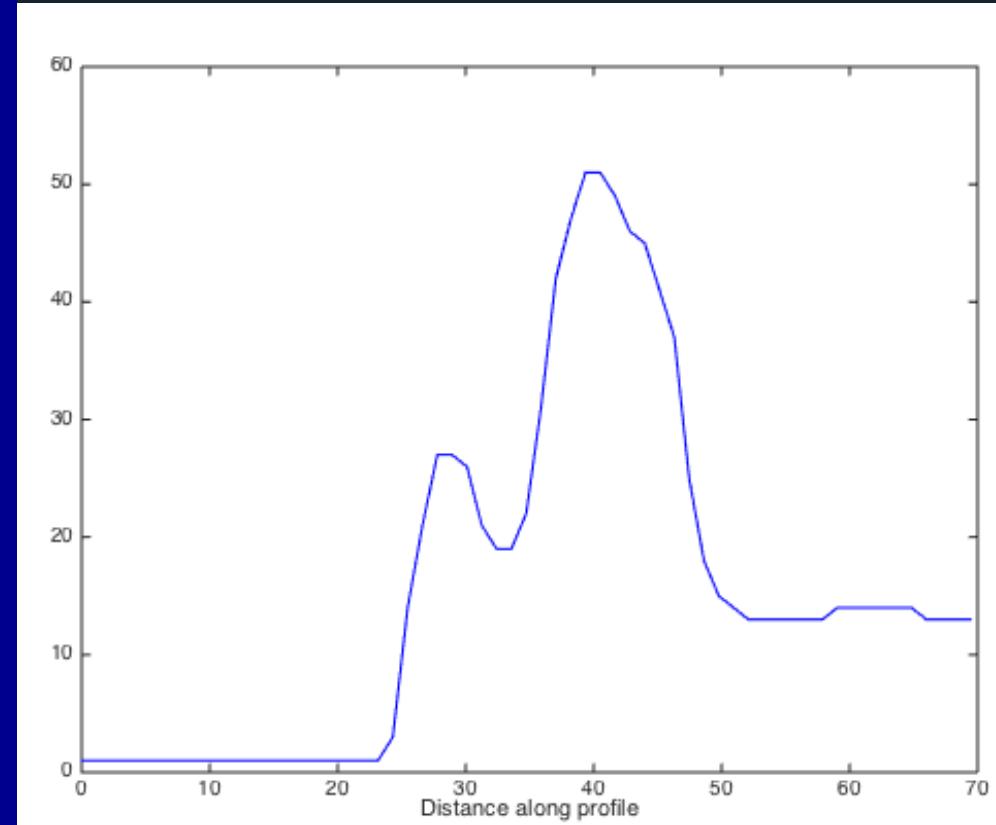
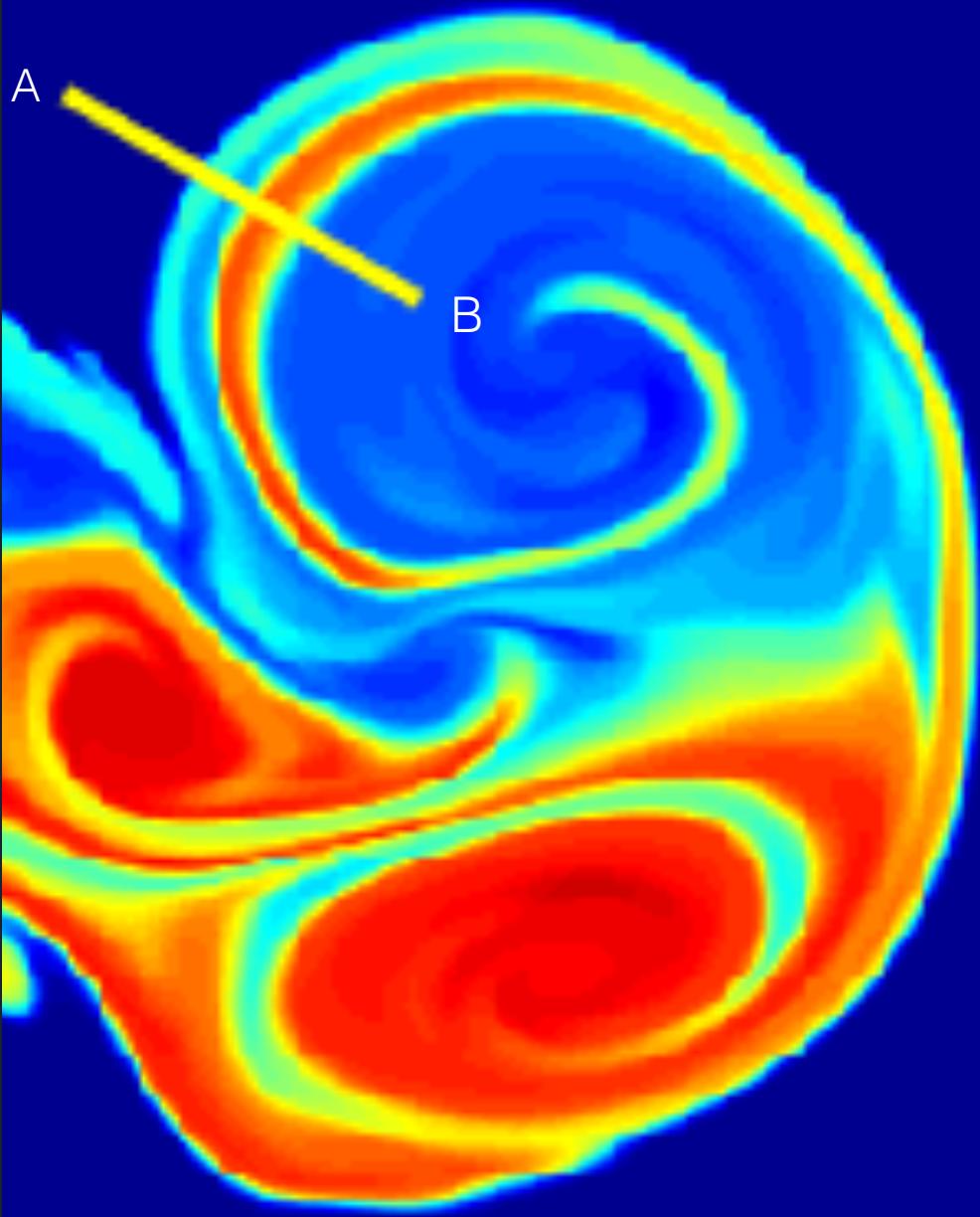




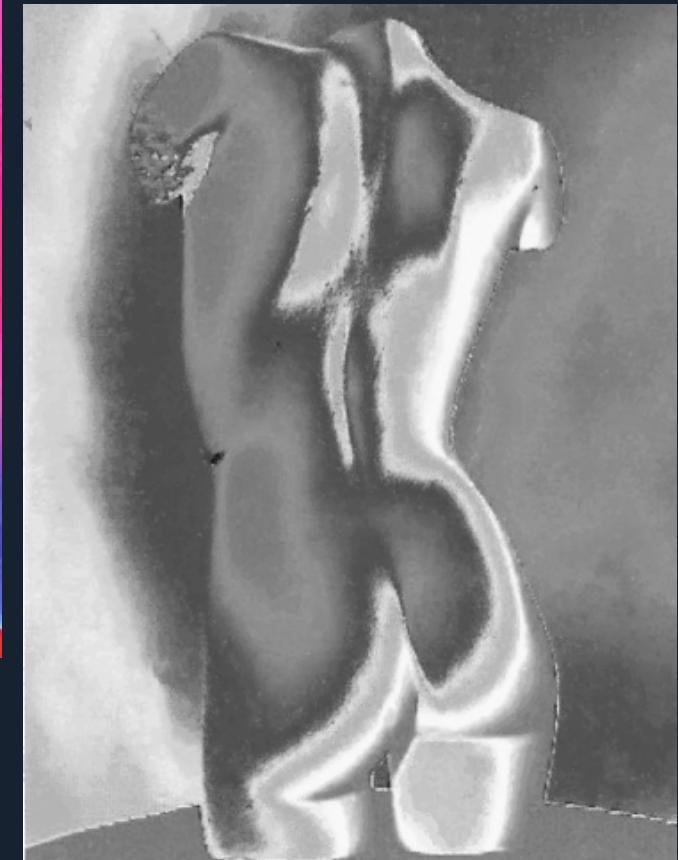
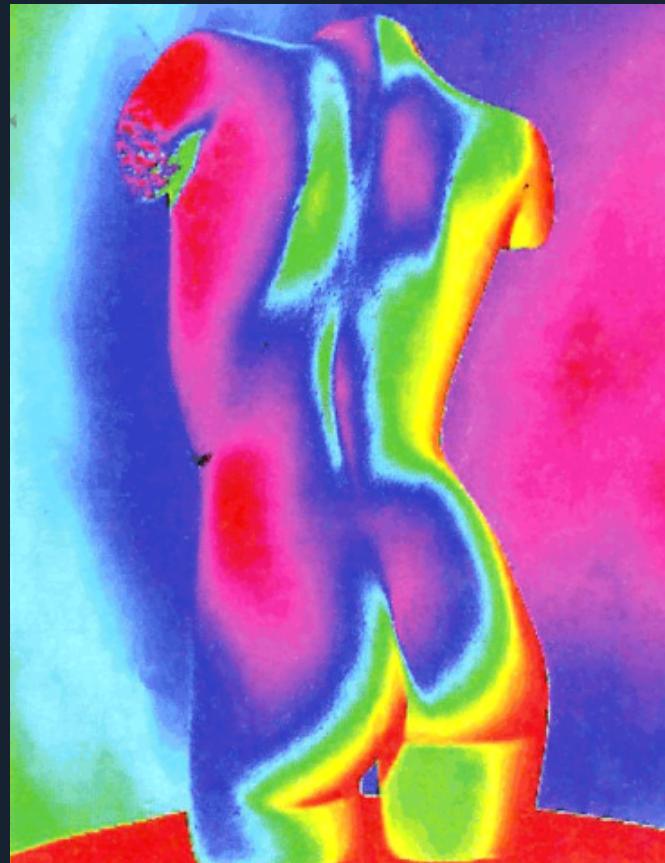
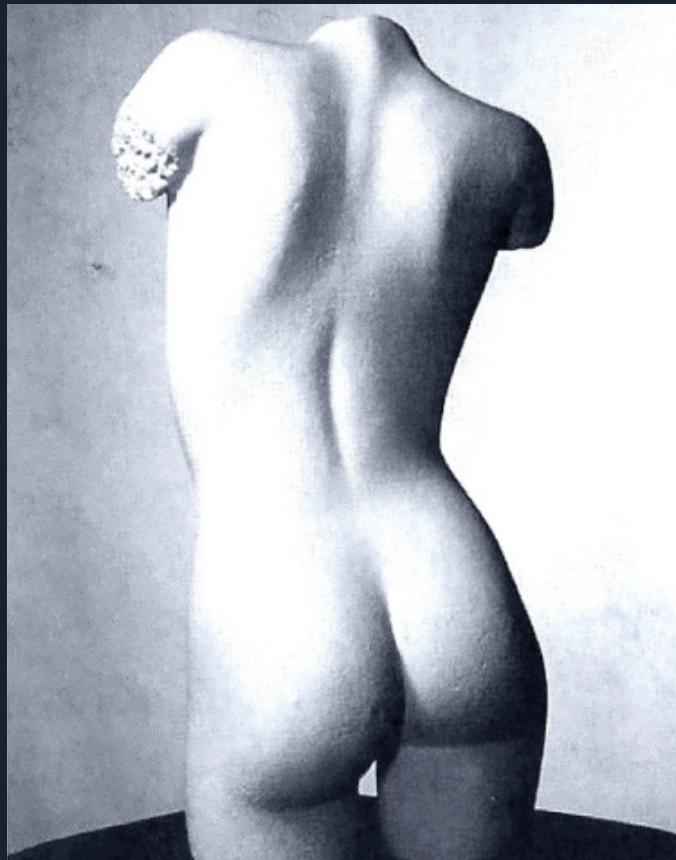
Pretend this is a 3d shape and you are
sliding your finger over A to B.
What grooves do you feel?
What do the stripes mean?



Some of the color stripes are meaningless. But not all of them!



**Those edges create
artifacts that aren't really there**



src > wwwroot > data > colors > ametrine.json

Project + | * | ⌂ | ⌂ | ⌂

1: Project

2: Structure

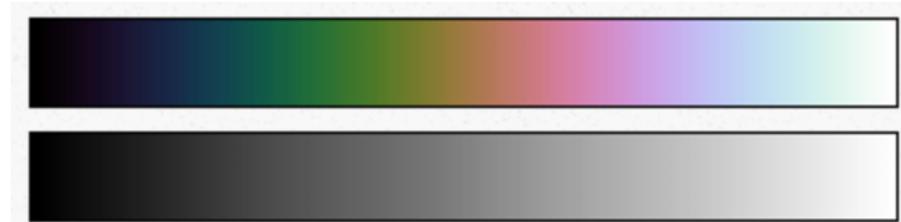
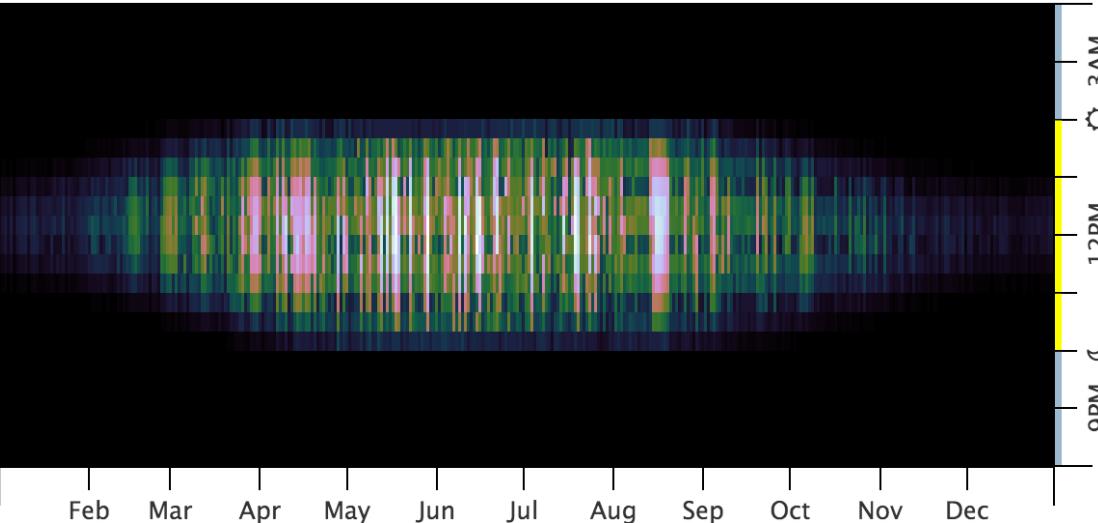
3: npm

4: timeseries8.js x ametrine.json x ac_load.json x

colors

```
1 {  
2   "legend": "Ametrine",  
3   "url": "https://github.com/ashao/matlab/blob/master/external/ametrine.m",  
4   "theoreticalPaperUrl": "https://www.osapublishing.org/oe/abstract.cfm?uri=oe-2  
5   "colors":  
6     [  
7       "rgb(30, 60, 151)",  
8       "rgb(32, 61, 151)",  
9       "rgb(34, 61, 151)",  
10      "rgb(35, 61, 151)",  
11      "rgb(37, 62, 151)",  
12      "rgb(39, 62, 151)",  
13      "rgb(41, 62, 151)",  
14      "rgb(43, 63, 151)",  
15      "rgb(44, 63, 151)",  
16      "rgb(46, 63, 151)",  
17      "rgb(48, 64, 151)",  
18      "rgb(50, 64, 151)",  
19      "rgb(51, 64, 151)",  
20      "rgb(53, 65, 151)",  
21      "rgb(55, 65, 151)",  
22      "rgb(57, 66, 151)",  
23      "rgb(58, 66, 152)",  
24      "rgb(60, 66, 152)",  
25      "rgb(62, 67, 152)",  
26      "rgb(64, 67, 152)",  
27      "rgb(66, 67, 152)",  
28      "rgb(67, 68, 152)",  
29      "rgb(69, 68, 152)",  
30      "rgb(71, 68, 152)",  
31      "rgb(73, 69, 152)",  
32      "rgb(74, 69, 152)",  
33      "rgb(76, 69, 152)"  
34 ]
```

Cube Helix



Feb Mar Apr May Jun Jul Aug Sep Oct Nov Dec

💻 d3 / d3-plugins

Watch ▾ 191

Star 1,484

Fork 564

Code

Issues 31

Pull requests 33

Pulse

Graphs

Branch: master ▾

d3-plugins / cubehelix /

New file

Upload files

Find file

History

mbostock Optimize.

Latest commit fecbc79 on Jun 24, 2014

..

📄 README.md

Restore example.

2 years ago

📄 cubehelix.js

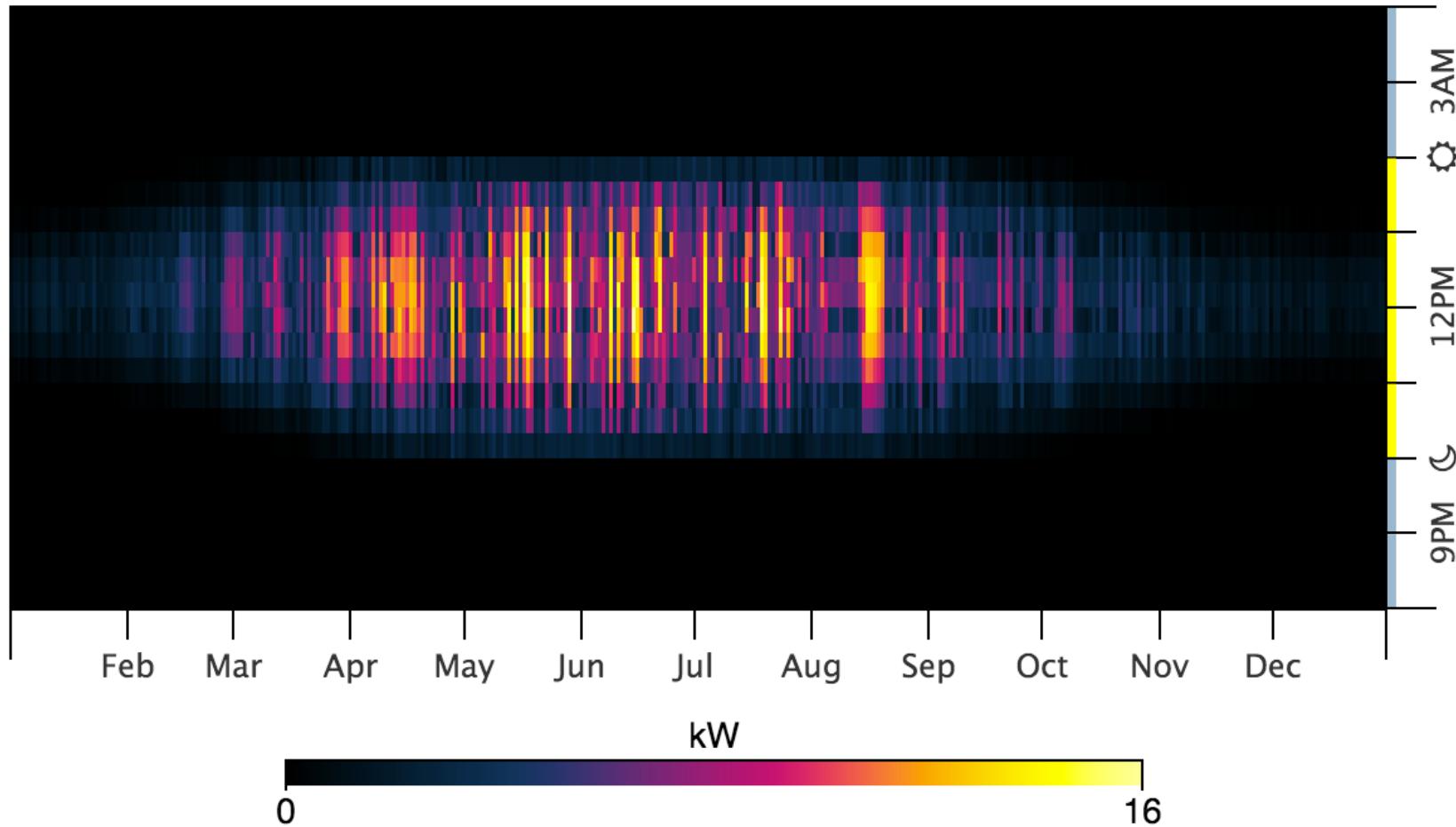
Optimize.

2 years ago

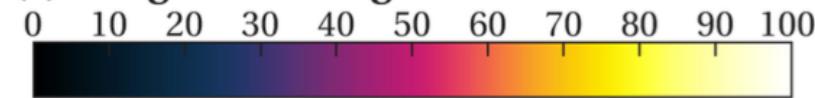
📄 README.md

An interpolator that implements Dave Green's [cubehelix color scheme](#).

morganstemning



(c) Morgenstemning



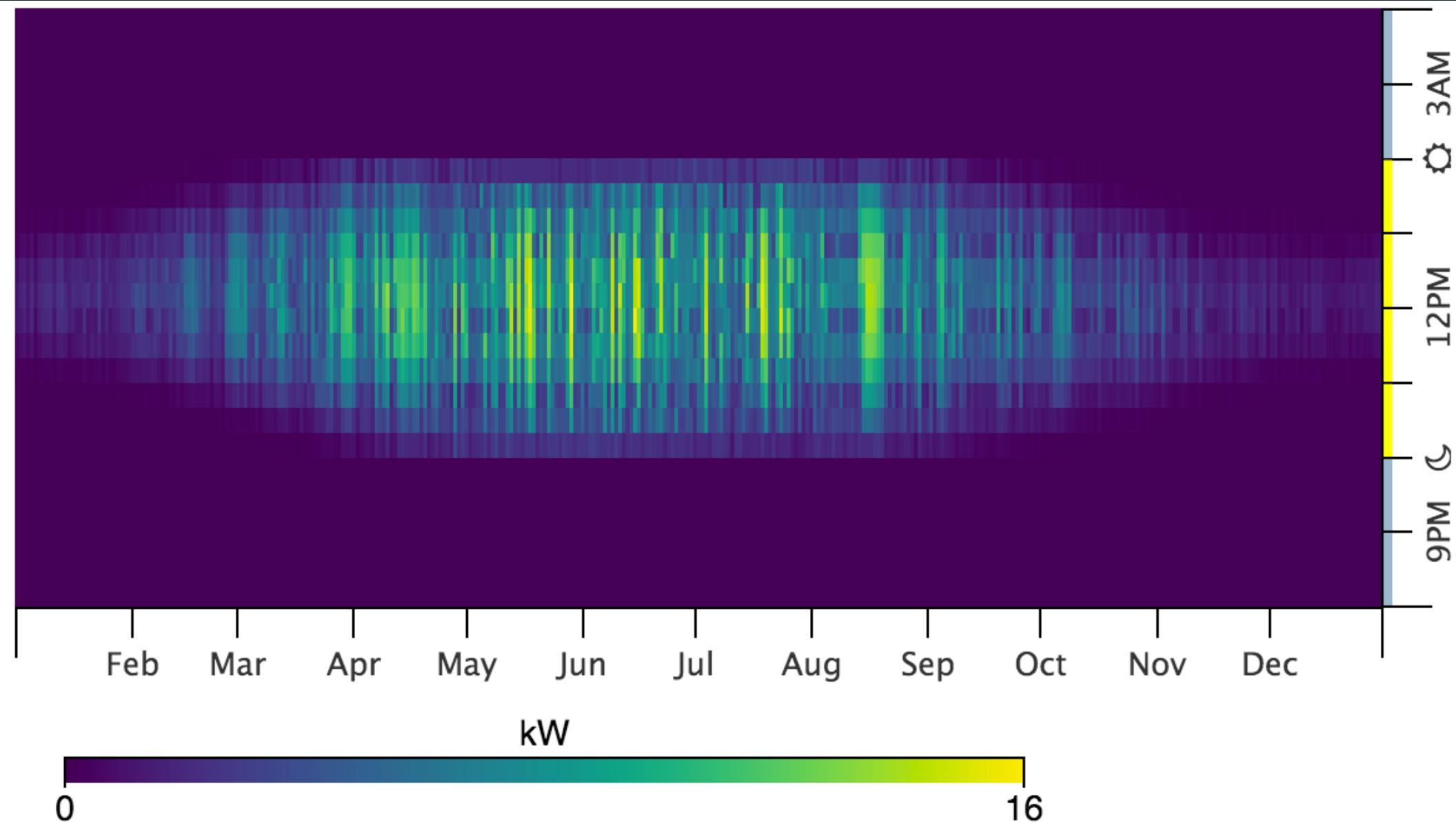
BW conversion

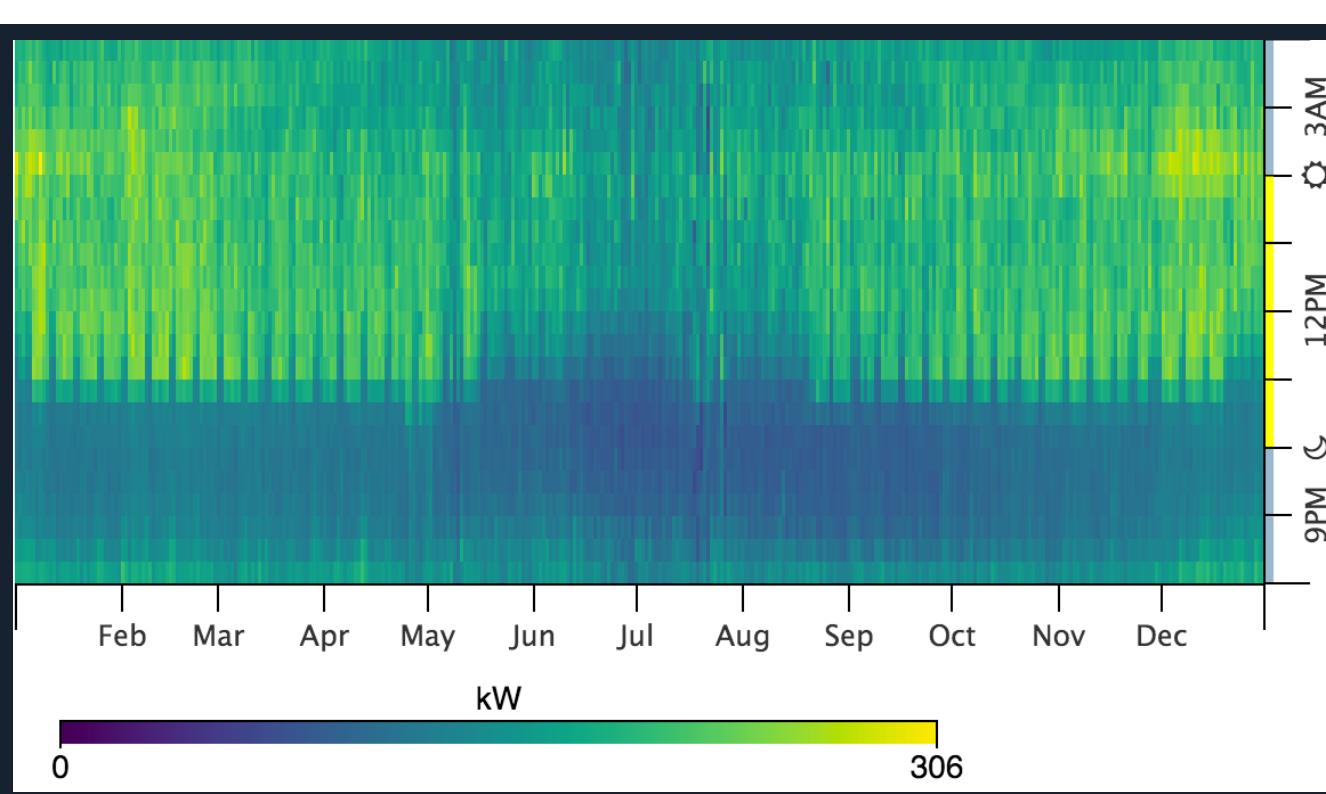
CMYK conversion

Deuteranope perception

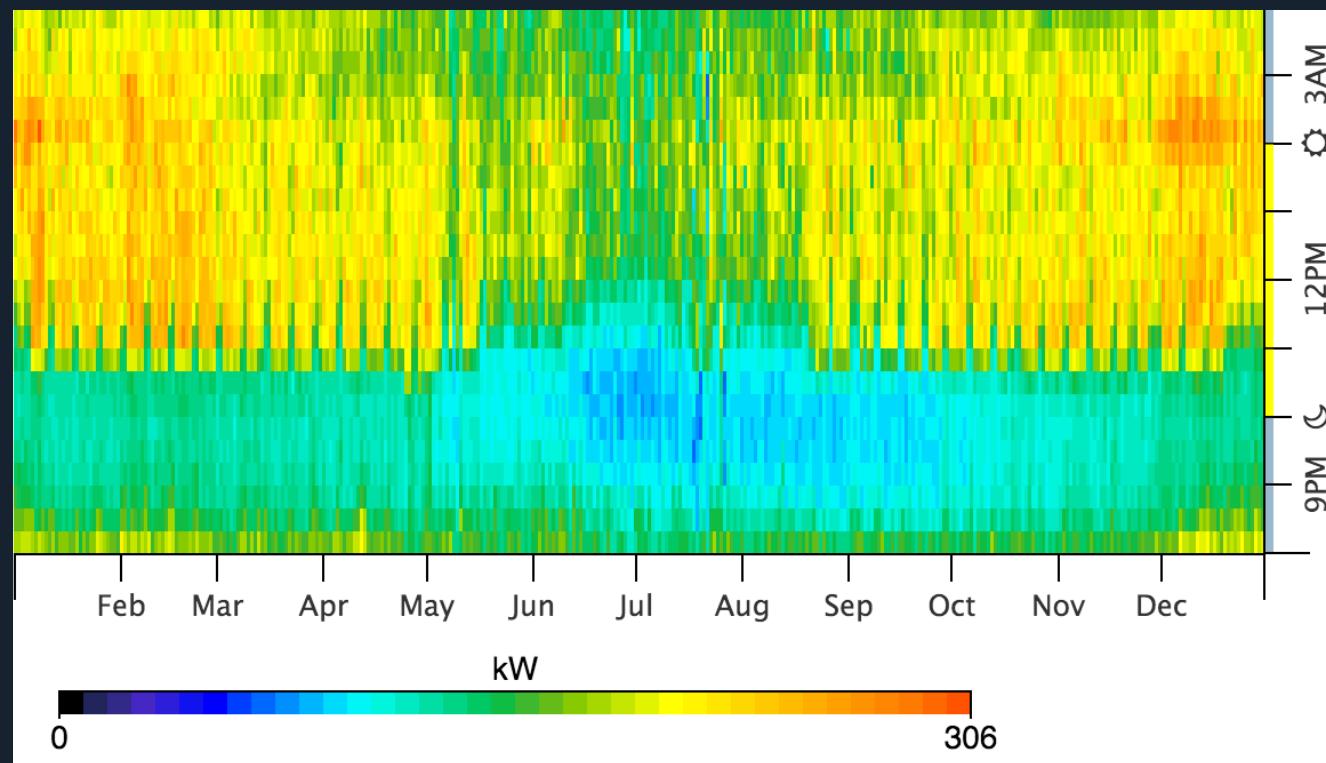
Protanope perception

viridis



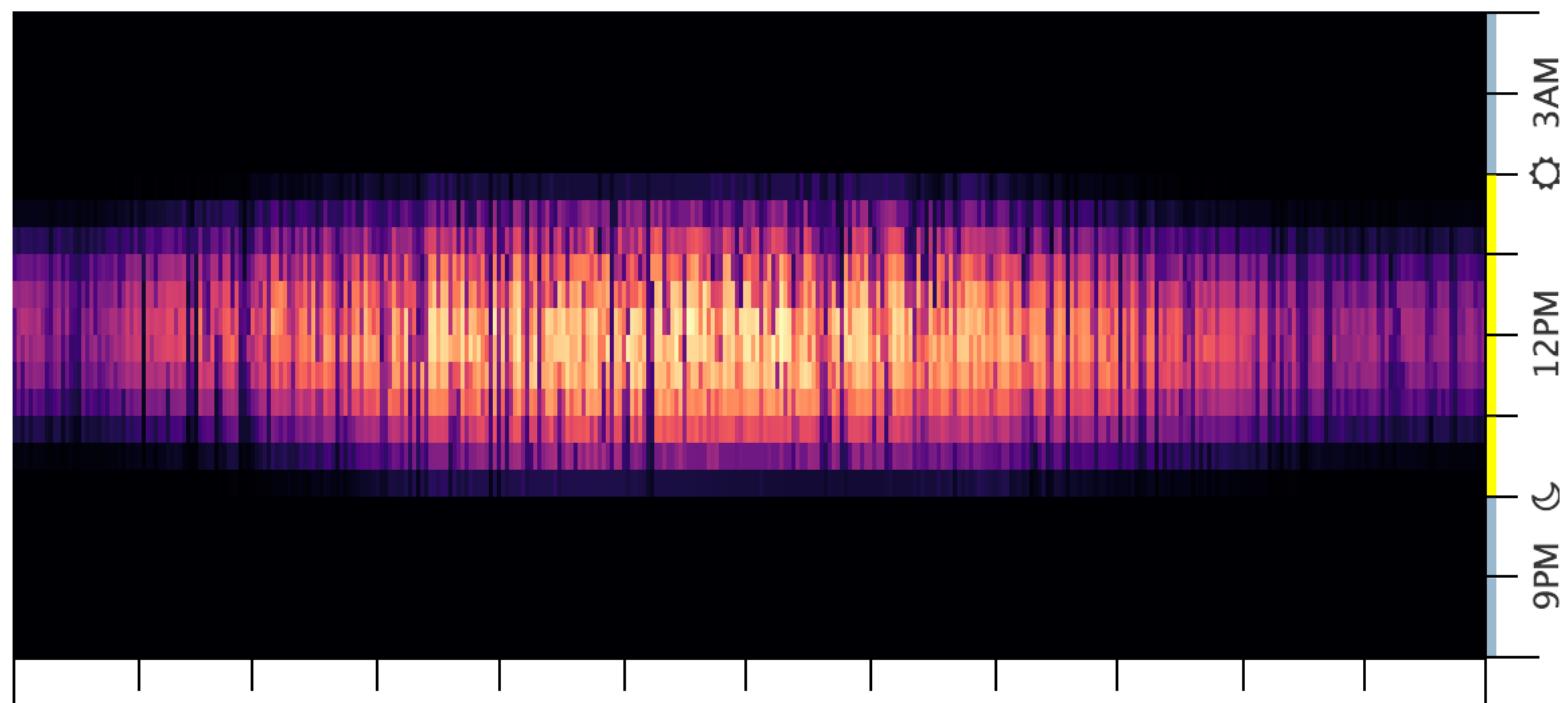


Viridis

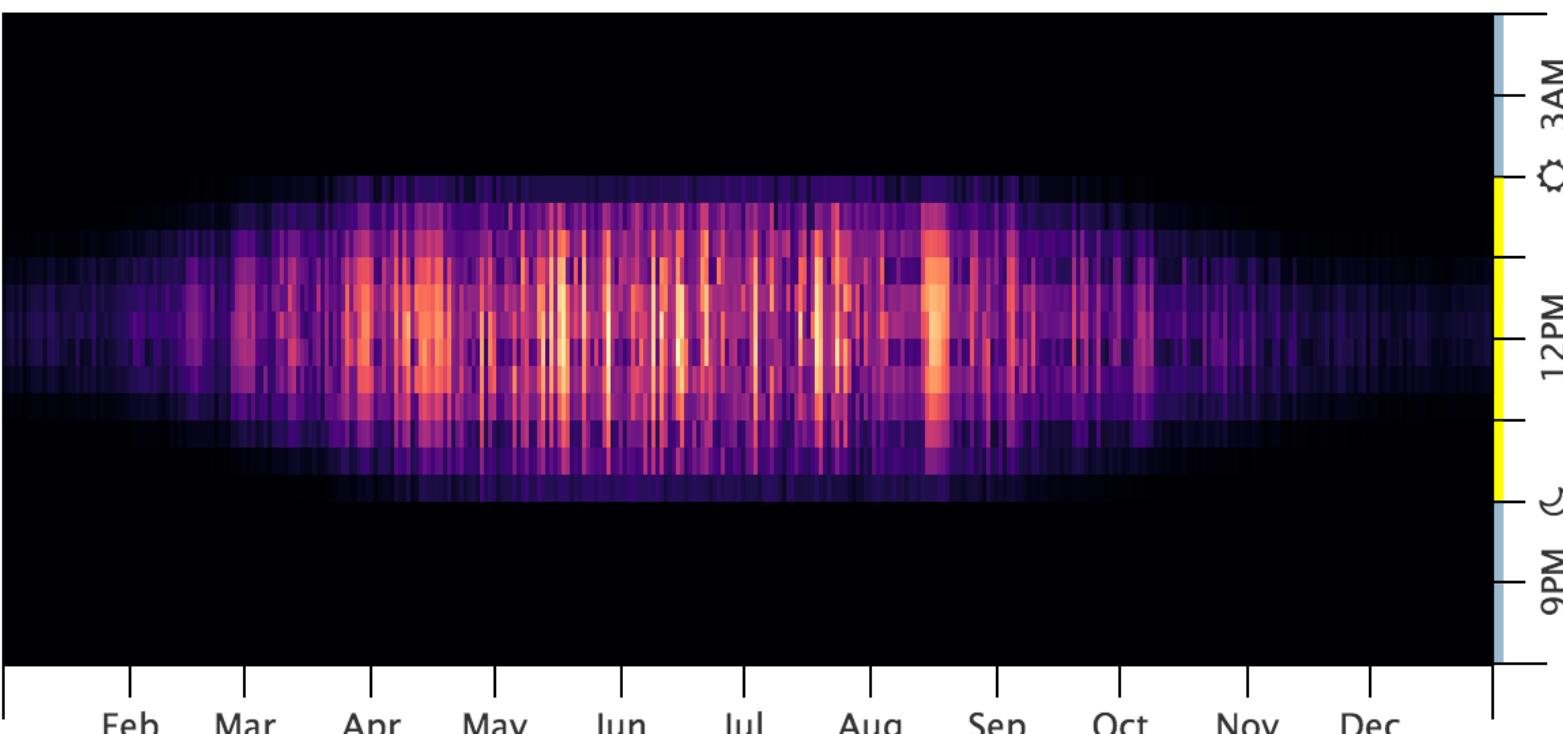


Rainbow (Jet)

PV Solar Output. Boulder, Colorado



PV Solar Output. Yakutat, Alaska





The End