



Data Driven Documents

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

The screenshot shows a GitHub repository page for 'rinckd / d3-workshop'. The top navigation bar includes links for 'Pull requests', 'Issues', and 'Gist'. Below the header, there's a search bar and a navigation bar with tabs for 'Code', 'Issues 0', 'Pull requests 0', 'Wiki', 'Pulse', 'Graphs', and 'Settings'. The main content area displays repository statistics: 80 commits, 1 branch, 0 releases, and 1 contributor. A 'New pull request' button is highlighted in green. The repository's history section shows updates for files like 'src', '.gitignore', 'Presentation_Slides.pdf', and 'README.md'. The 'README.md' file has a detailed view showing its content.

This repository Search Pull requests Issues Gist

rinckd / d3-workshop

Unwatch 1 Unstar 1 Fork

Code Issues 0 Pull requests 0 Wiki Pulse Graphs Settings

slides — Edit

80 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 Download ZIP

rinckd updates Latest commit 9d76da9 25 minutes ago

File	Commit Message	Time Ago
src	font-awesome	an hour ago
.gitignore	updated gitignore	23 days ago
Presentation_Slides.pdf	presentation update	15 days ago
README.md	updating readme	6 days ago

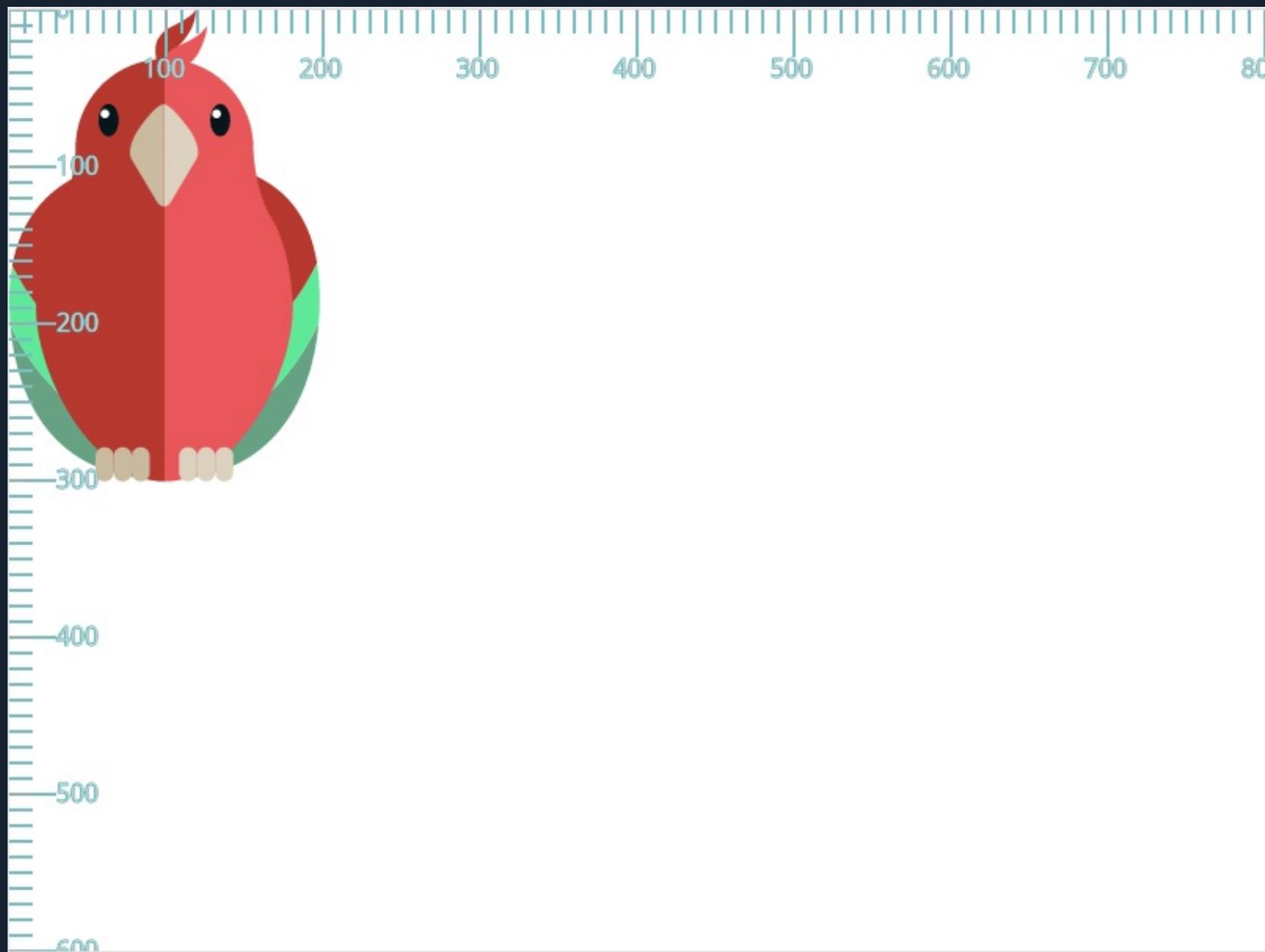
README.md

Updating the README.md file to include information about the D3 workshop.

SVG Paths and d3



Coordinate System



http://localhost:3000/test.html

A screenshot of a web browser window showing the source code of a file named "test.html". The browser's title bar reads "test.html - src - [~/_Code/_d3_workshop/src]". The address bar shows the URL "http://localhost:3000/test.html". The page content area displays the following HTML code:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <title>Title</title>
6  </head>
7  <body>
8
9  </body>
10 </html>
```

The code editor interface includes a "Structure" view on the left showing the document outline, and a toolbar at the top with various icons for file operations and navigation.

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



Search GitHub

Pull requests Issues Gist

Delivering Octicons with SVG

February 22, 2016



aaronshekey

Engineering

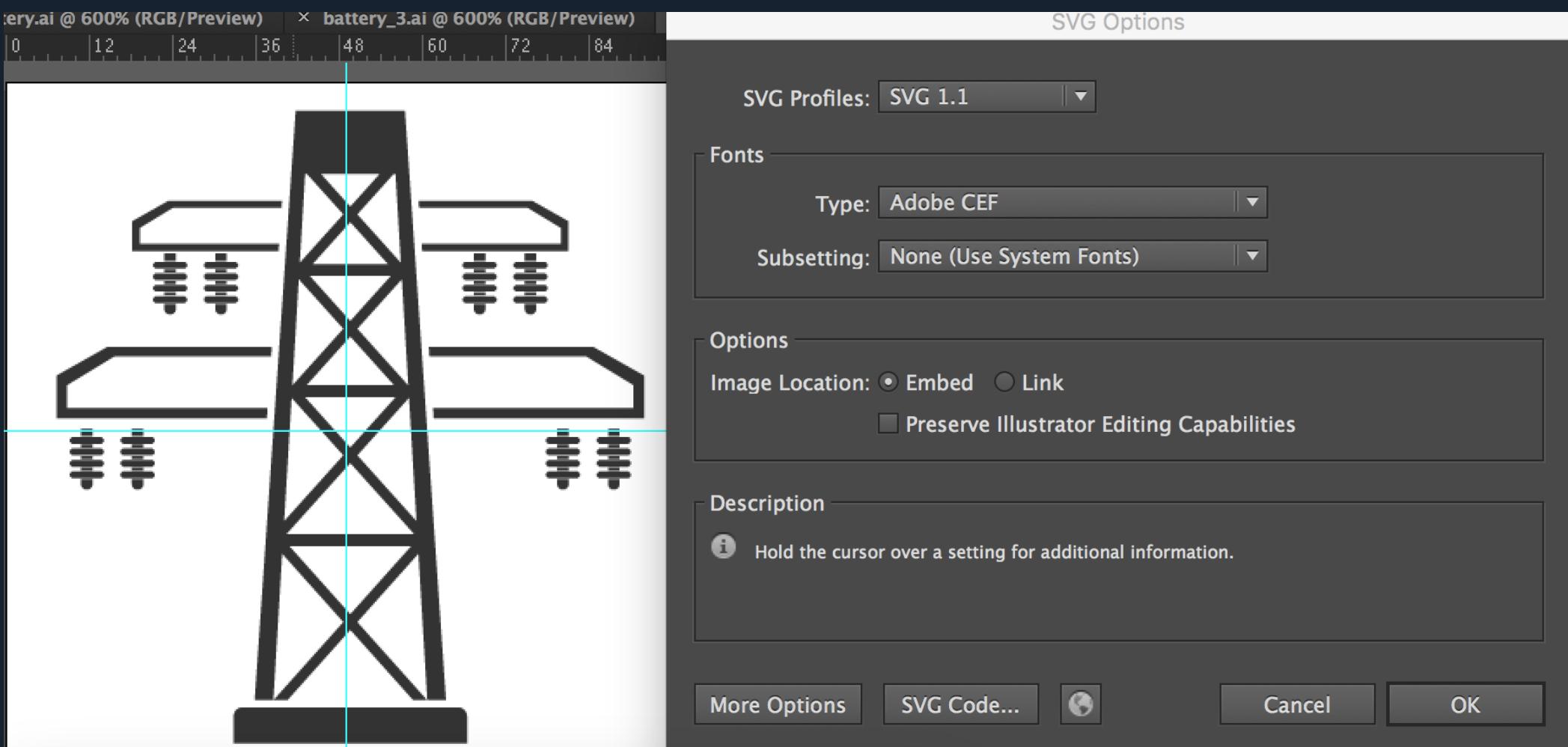
GitHub.com no longer delivers its icons via icon font. Instead, we've replaced all the [Octicons](#) throughout our codebase with SVG alternatives. While the changes are mostly under-the-hood, you'll immediately feel the benefits of the SVG icons.

```
<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 shapes. A vertical cyan selection bar is positioned to the left of the central tower.

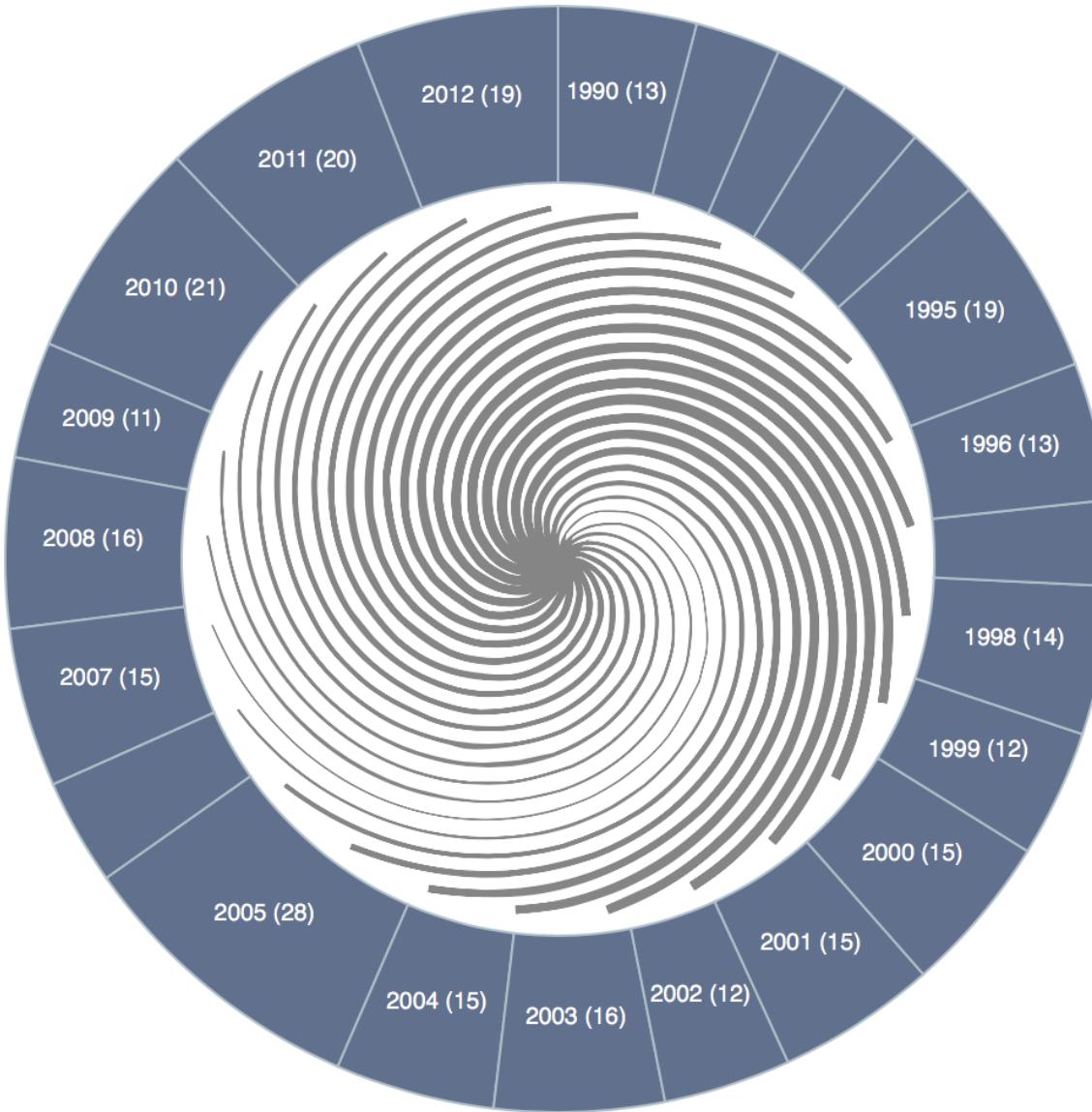
The "SVG Options" dialog box is open, set to "SVG 1.1" profile. It includes sections for "Fonts" (Type: Adobe CEF, Subsetting: None (Use System Fonts)), "Options" (Image Location: Embed selected, Link), and "Description" (Hold the cursor over a setting for additional information). At the bottom are buttons 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 battery's structure. Lines 37, 38, and 39 of the code are visible:

```
37 <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"/>
38 <rect x="30.259" y="30.154" fill="#333333" width="1.61" height="0.406"/>
```

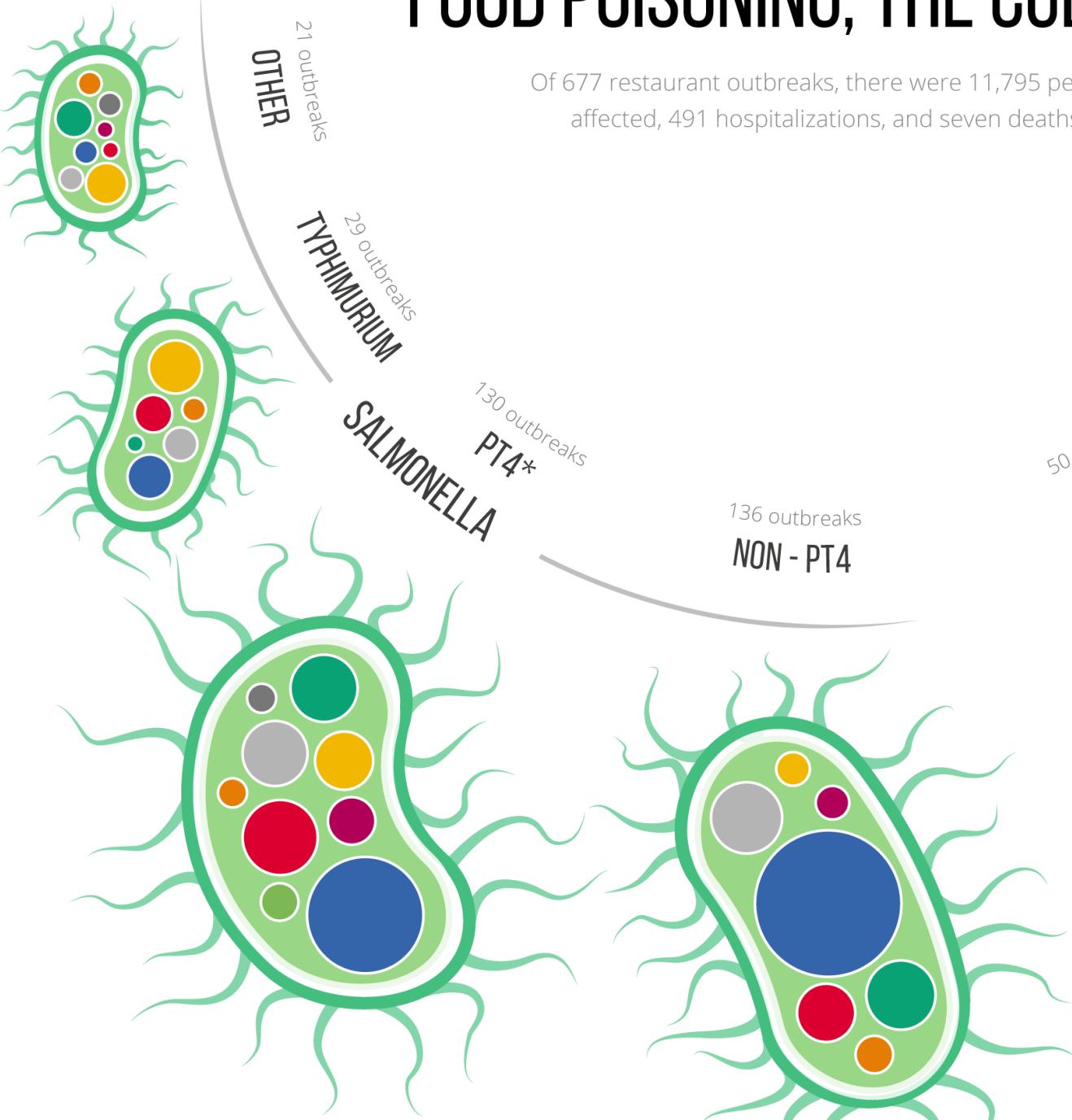
Illustrator SVG in D3

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

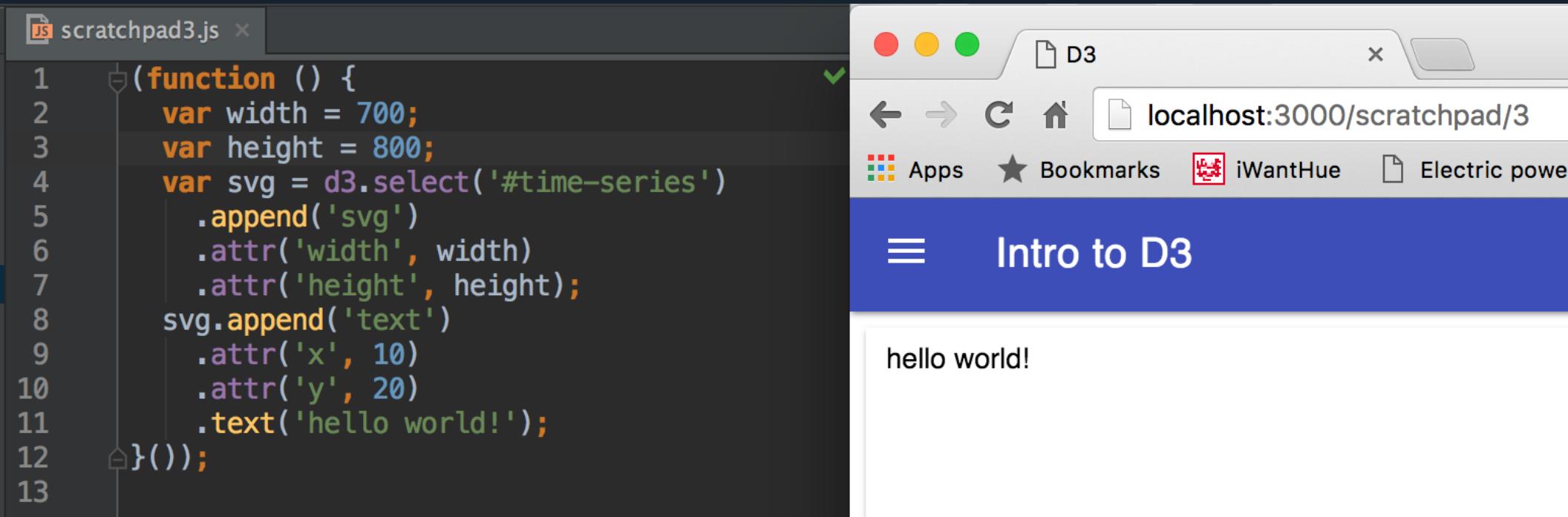
3 years ago

 index.html

typos

2 years ago

Scratchpad



```
scratchpad3.js
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
```

D3

localhost:3000/scratchpad/3

Apps Bookmarks iWantHue Electric power

Intro to D3

hello world!

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

The image shows a Mac OS X desktop environment with three open windows:

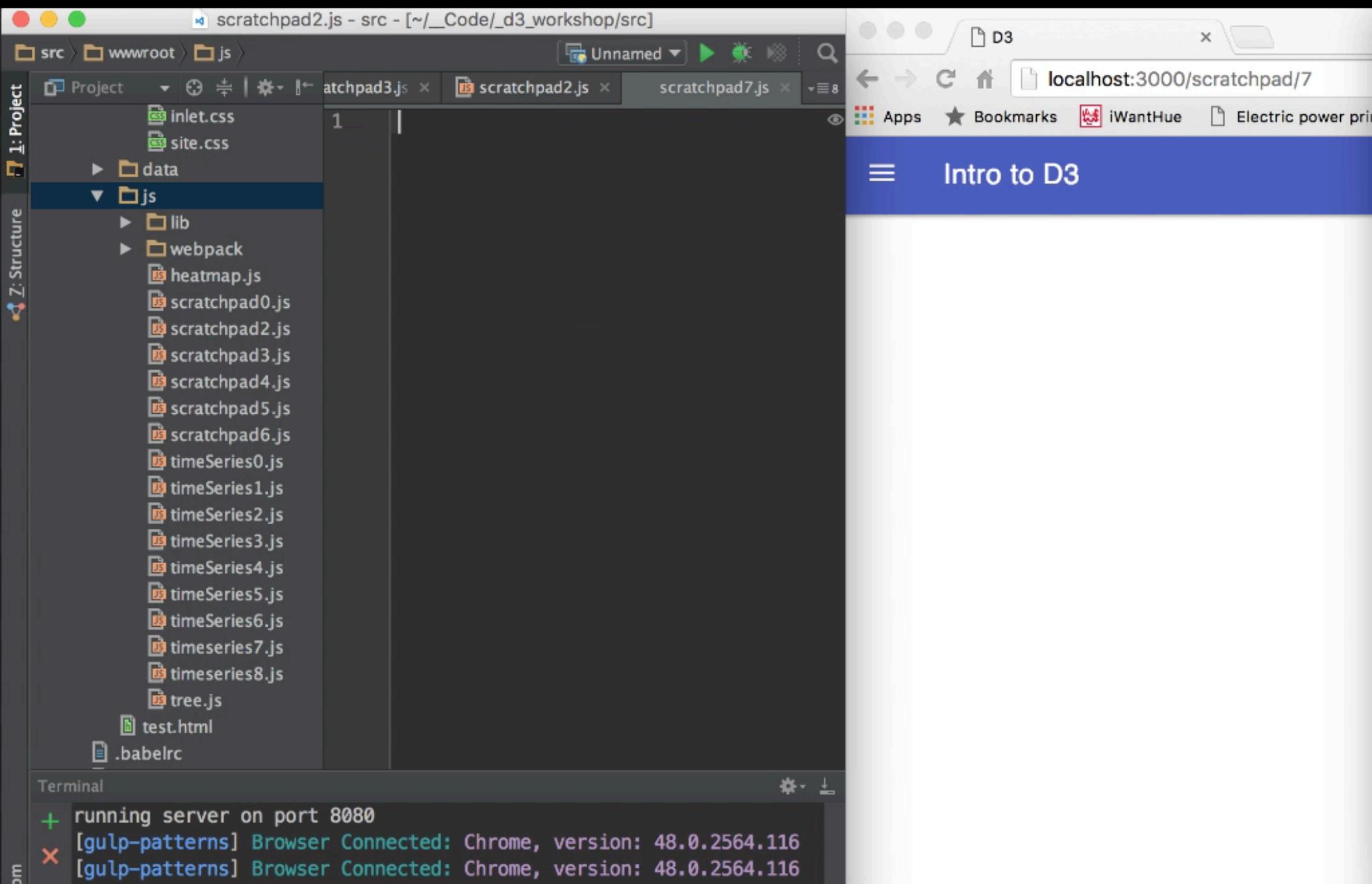
- Code Editor (Left):** A window titled "scratchpad.ejs" containing EJS (Embedded JavaScript) code. The code includes imports for head and nav partials, and defines a main content area with a time-series div. It also includes a script block that conditionally loads scripts based on the "scripts" variable.
- Browser (Top Right):** A window titled "scratchpad.ejs" showing the rendered HTML output. The page has a header, navigation, and a main content area with a "data-content" div containing a "time-series" div. The "time-series" div contains the rendered script block.
- Terminal (Bottom Left):** A window titled "scratchpad3.js" containing D3.js code. The code sets up a SVG container with width 700 and height 800, and appends a text element.
- Browser (Bottom Right):** A window titled "D3" showing the rendered D3 visualization at "localhost:3000/scratchpad/3". The page title is "Intro to D3" and the content area displays the text "hello world".

Chaining Methods

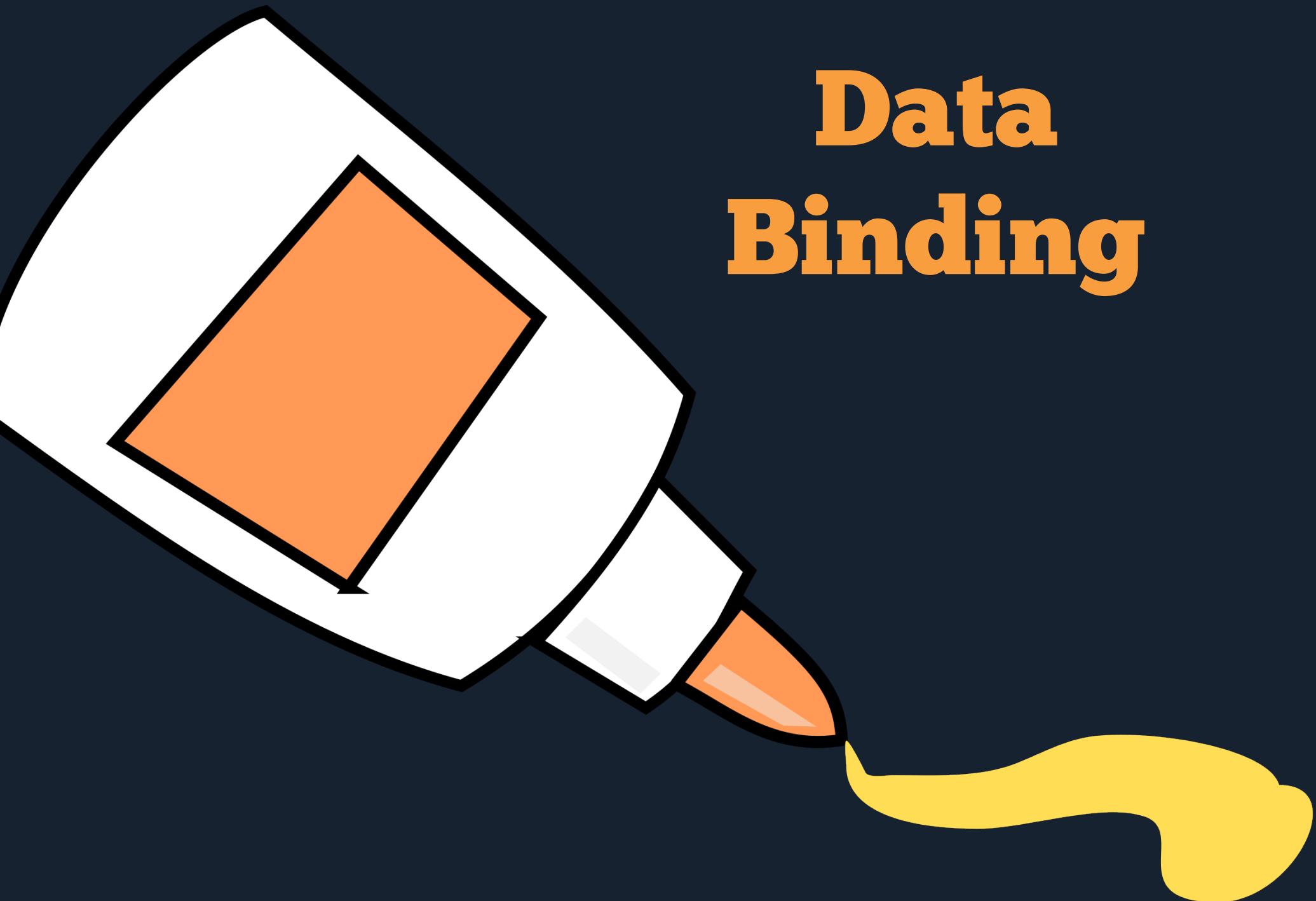


```
scratchpad3.js x scratchpad5.js x timeseries9.js x scratchpad3.js x ac_load_boxplot.json  
1 (function () {  
2   'use strict';  
3   const width = 700;  
4   const height = 800;  
5  
6   var svg = d3.select('#time-series')  
7     .append('svg')  
8     .attr('width', width)  
9     .attr('height', height);  
10  svg.append('text')  
11    .attr('x', 10)  
12    .attr('y', 20)  
13    .text('hello world');  
14})();  
15  
16 (function () {  
17   'use strict';  
18   const width = 700;  
19   const height = 800;  
20  
21   var divId = d3.select('#time-series');  
22   var svg = divId.append('svg');  
23   svg.attr('width', width);  
24   svg.attr('height', height);  
25   var text = svg.append('text');  
26   text.attr('x', 10);  
27   text.attr('y', 20);  
28   text.text('hello world');  
29})();  
30
```

http://localhost:3000/scratchpad/7



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.

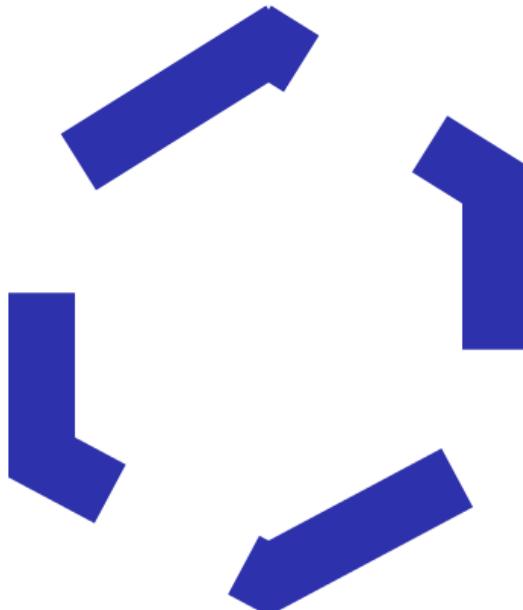
Code Editor (Left):

```
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>
```

http://localhost:3000/scratchpad/6

The screenshot shows a development environment with a code editor and a browser window.

Code Editor:

- Project structure: `src`, `wwwroot`, `Js`, `scratchpad6.js`.
- File tabs: `scratchpad6.js`, `ac_load.json`, `scratchpad3.js`, `boulder_sun.json`.
- Code content (from line 18 to 50):

```
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   .attr('stroke-dasharray', '80 50')
48   .attr('stroke-dashoffset', 0);
49
50 }());
```

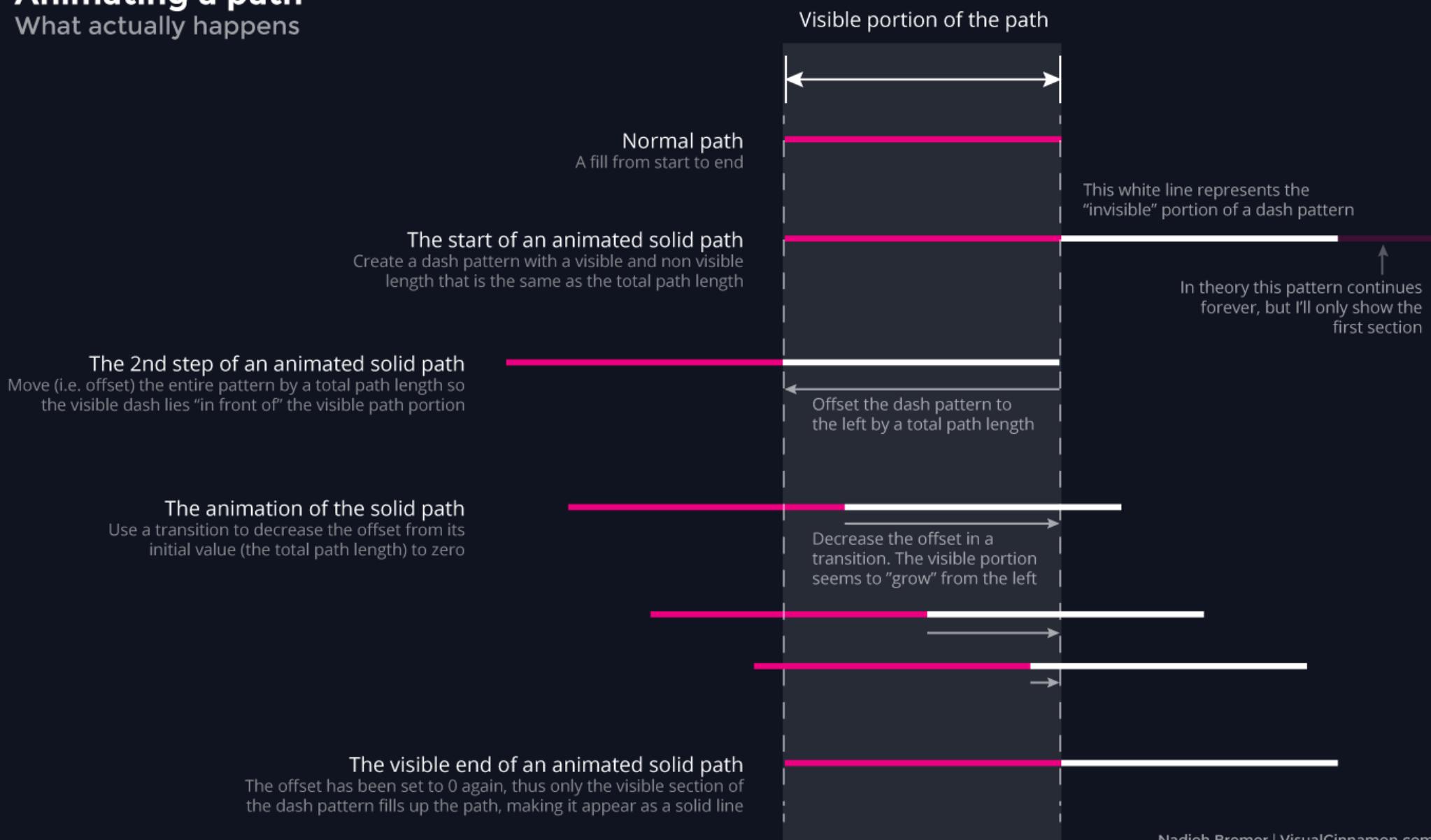
Terminal:

Browser Preview:

The browser title is "Intro to D3". The page displays a blue dashed line forming a circular path with four segments.

Animating a path

What actually happens



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:

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

Time Step Size: **60** minutes

Random Variability

Day-to-day (%): **6.308**

Timestep (%): **8.868**

Peak Month: **February**

Load Type: AC DC

Scaled Annual Average (kWh/d): **6,818.77**

Efficiency (Advanced)

Efficiency multiplier: **0.95**

Capital cost (\$): **20000**

Lifetime (yr): **20**

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) → DC → Li-ion
Grid → AC → DC

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 multiplier: 1 Capital cost (\$): 0 Lifetime (yr): 10

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

SUGGESTIONS:

Model does not match results Newer version of HOMER Pro available

HOMER PRO





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} ▾

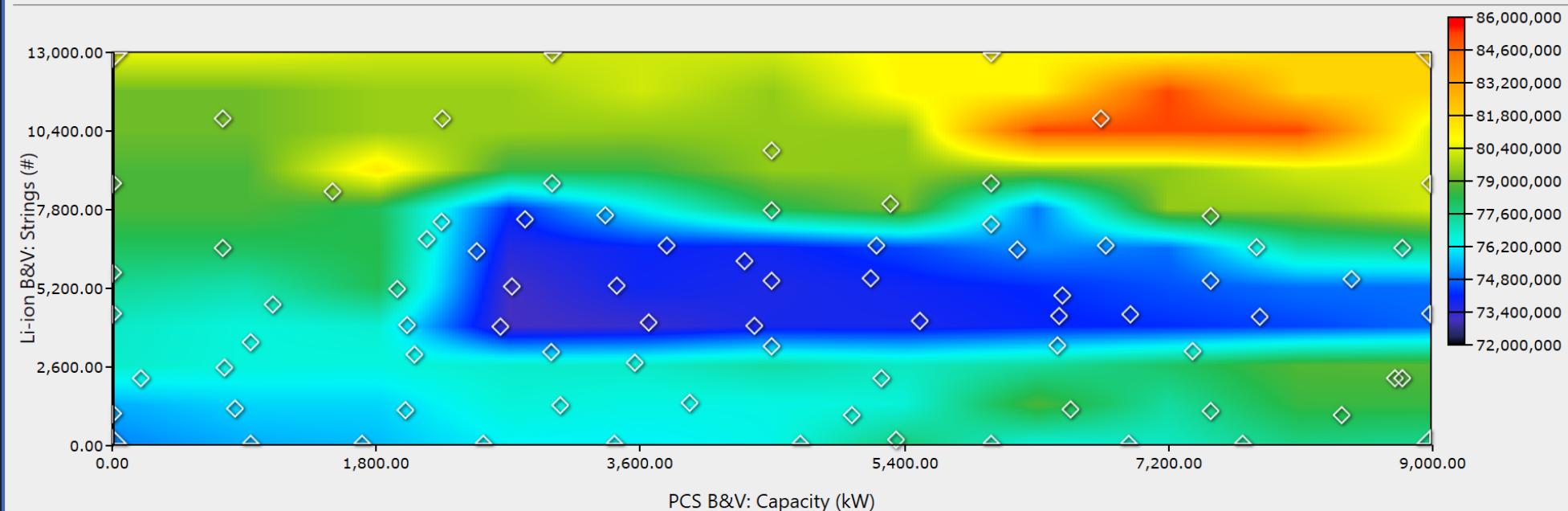
— Optimization Variables —

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 —

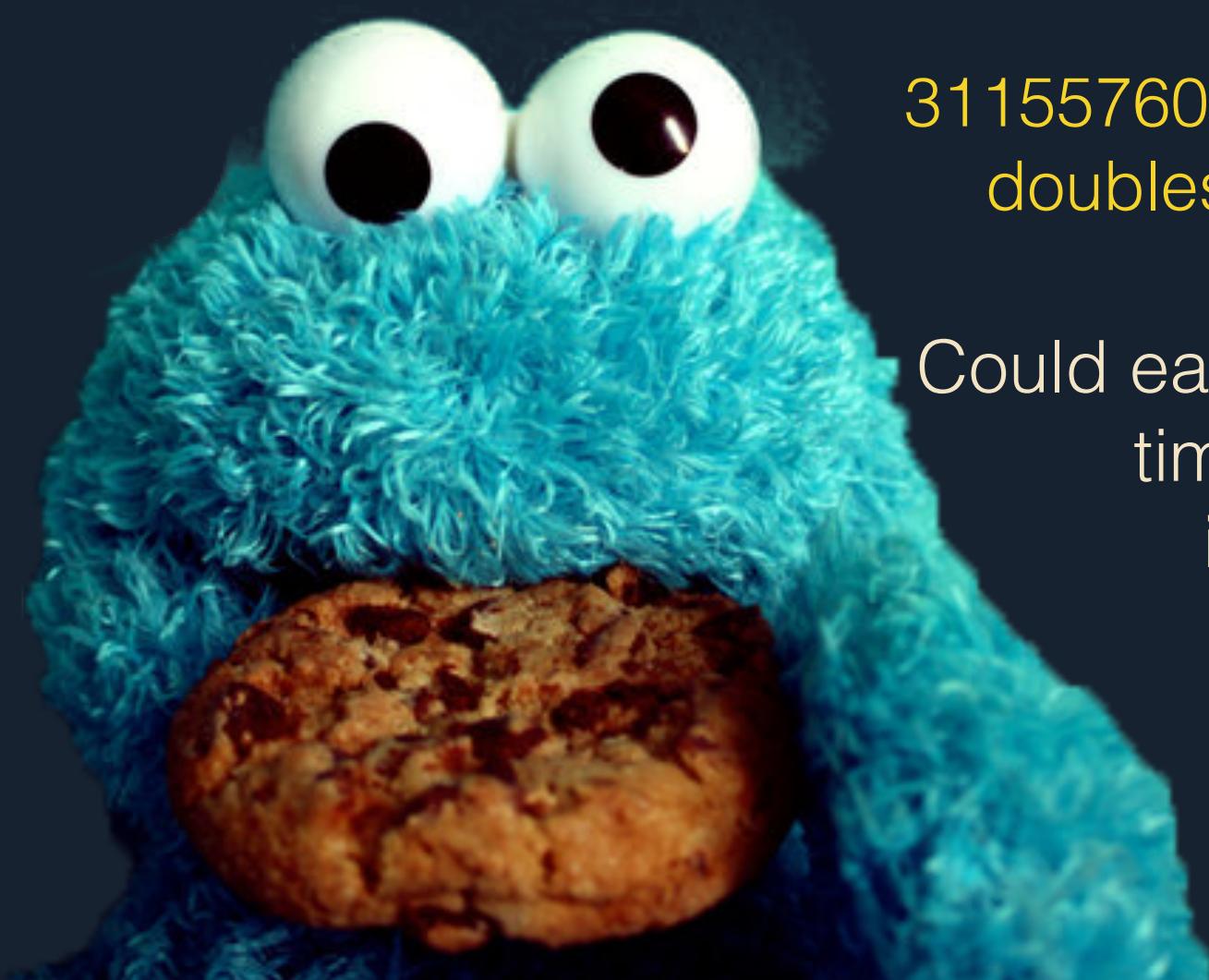
Surface Total Net Present Cost ▾ Superimposed <none> ▾





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

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     }  
]}  
X
```

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);
```



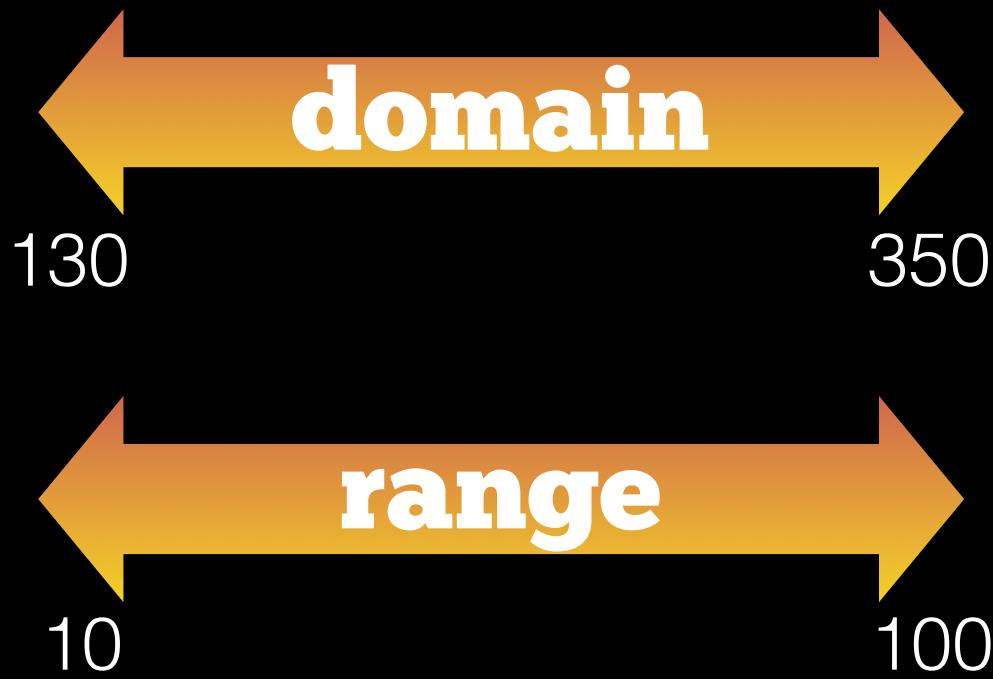
Components

```
var xScale = d3.scale.linear()  
  .range([margins.left, width - margins.  
    right])  
  .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=  
"...
```

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" and 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
```

ac_load.json - src - [~/Code/d3_workshop/src]

src wwwroot data ac_load.json Unnamed

ac_load.json

Project Structure

1: Project

1: Structure

data value

```
1 {  
2   "title": "AC Electrical Load. 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    }  
28  ]  
29}  
30
```

Terminal

[gulp-patterns] Access URLs:

Local: http://localhost:3000

D3

localhost:3000

Apps Bookmarks iWantHue Electric power principle

Intro to D3

Path SVG

Click on any of the numbers in the code to change them

```
<path d='M10,100 L131,66 L259,115 L200,50' fill='none' stroke='#25B0B0' stroke-width='7'></path>
```



```
1 {  
2   "title": "AC Electrical Load. Alaska",  
3   "units": "kW",  
4   "data": [  
5     {  
6       "date": "01/01/2015 00:00",  
7       "value": 193.3711  
8     },
```

1

```
var yScale = d3.scale.linear()  
.range([height - margins.bottom, margins.top])  
.domain([0, d3.max(json.data.map(function(data) { return data.value; }))]);
```

2

```
var yScale = d3.scale.linear()  
.range([height - margins.bottom, margins.top])  
.domain([0, d3.max(json.data, function(d) { return d.value; }))];
```

3

```
var yScale = d3.scale.linear()  
.range([height - margins.bottom, margins.top])  
.domain(d3.extent(json.data, function(d) { return d.value; }));
```

timeseries7.js - src - [~/_Code/_d3_workshop/src]

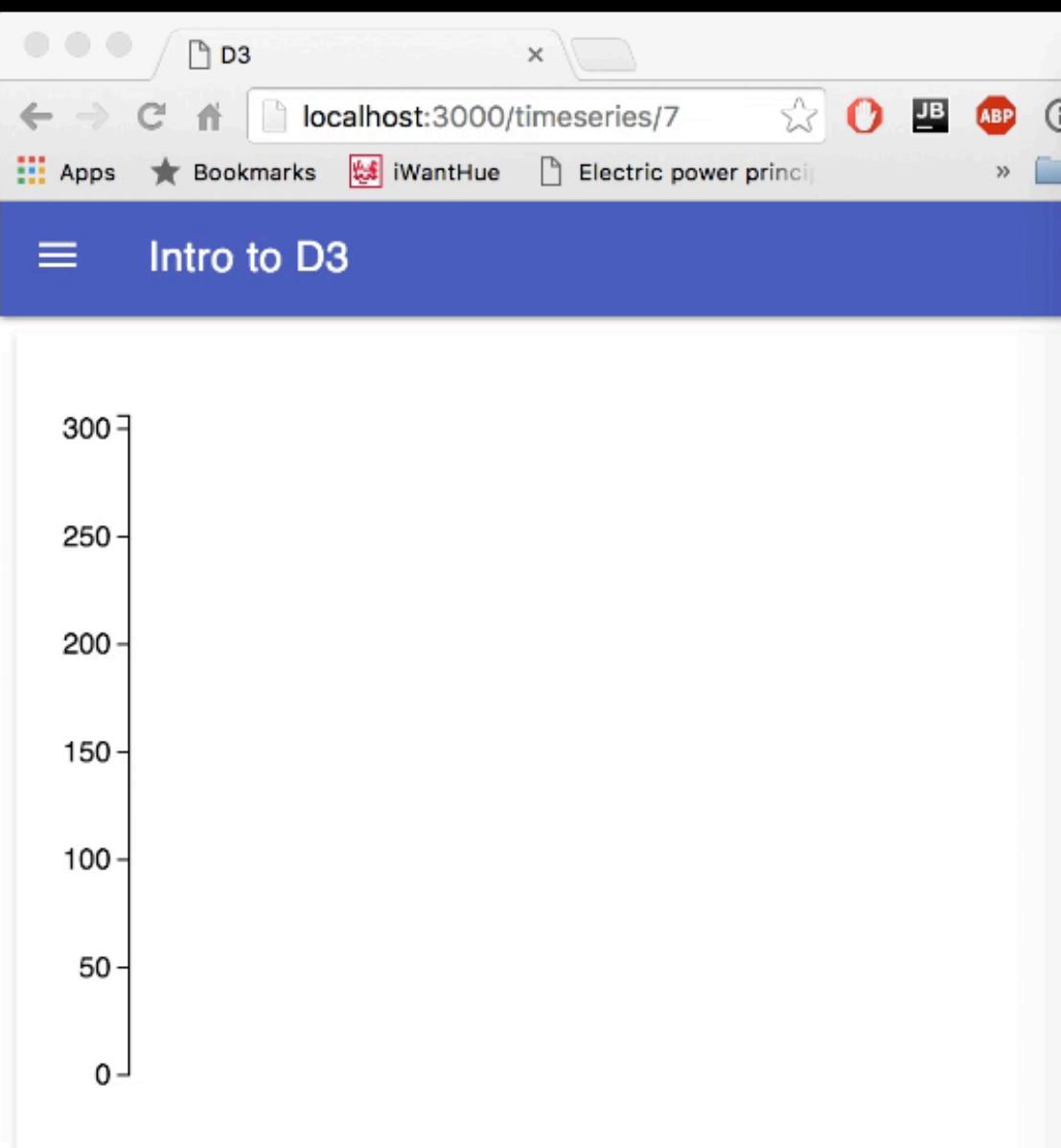
src > wwwroot > js > timeseries7.js > Unnamed

ac_load.json x timeseries7.js x

```
18     var yScale = d3.scale.Linear()
19         .range([height - margins.bottom, margins.t
20         .domain([0, d3.max(json.data.map(function
(data) { return data.value; }))]);
21
22     var yAxis = d3.svg.axis()
23         .scale(yScale)
24         .orient('left');
25
26     svg.append('g')
27         .call(yAxis)
28         .attr('transform', 'translate(' + (margins
.left) + ',0)')
29         .attr('class', 'axis');
30
31
32
33
34
35     })
36
37 })();
38
```

Terminal

```
+ running server on port 8080
x [gulp-patterns] Browser Connected: Chrome, version:
.2564.116
```



timeseries7.js - src - [~/_Code/_d3_workshop/src]

src > wwwroot > js > timeseries7.js > Unnamed

ac_load.json x timeseries7.js x

1: Project

1: Structure

10 .domain([0, d3.max(json.data.map(function
11 (data) { return data.value; })))];
12
13 var yAxis = d3.svg.axis()
14 .scale(yScale)
15 .orient('left');
16
17 svg.append('g')
18 .call(yAxis)
19 .attr('transform', 'translate(' + (margins
20 .left) + ',0)')
21 .attr('class', 'axis');
22
23 var xScale = d3.scale.linear()
24 .range([margins.left, width - margins.right])
25 .domain([0,23]);
26
27 var xAxis = d3.svg.axis()
28 .scale(xScale);
29
30 svg.append('g')
31 .call(xAxis)
32 .attr('class', 'axis')
33 .attr('transform', 'translate(0, ' + (height -
34 margins.bottom) + ')');
35
36
37
38
39
40
41
42
43
44

D3

localhost:3000/timeseries/7

Bookmarks iWantHue Electric power principle

Intro to D3

0 2 4 6 8 10 12 14 16 18

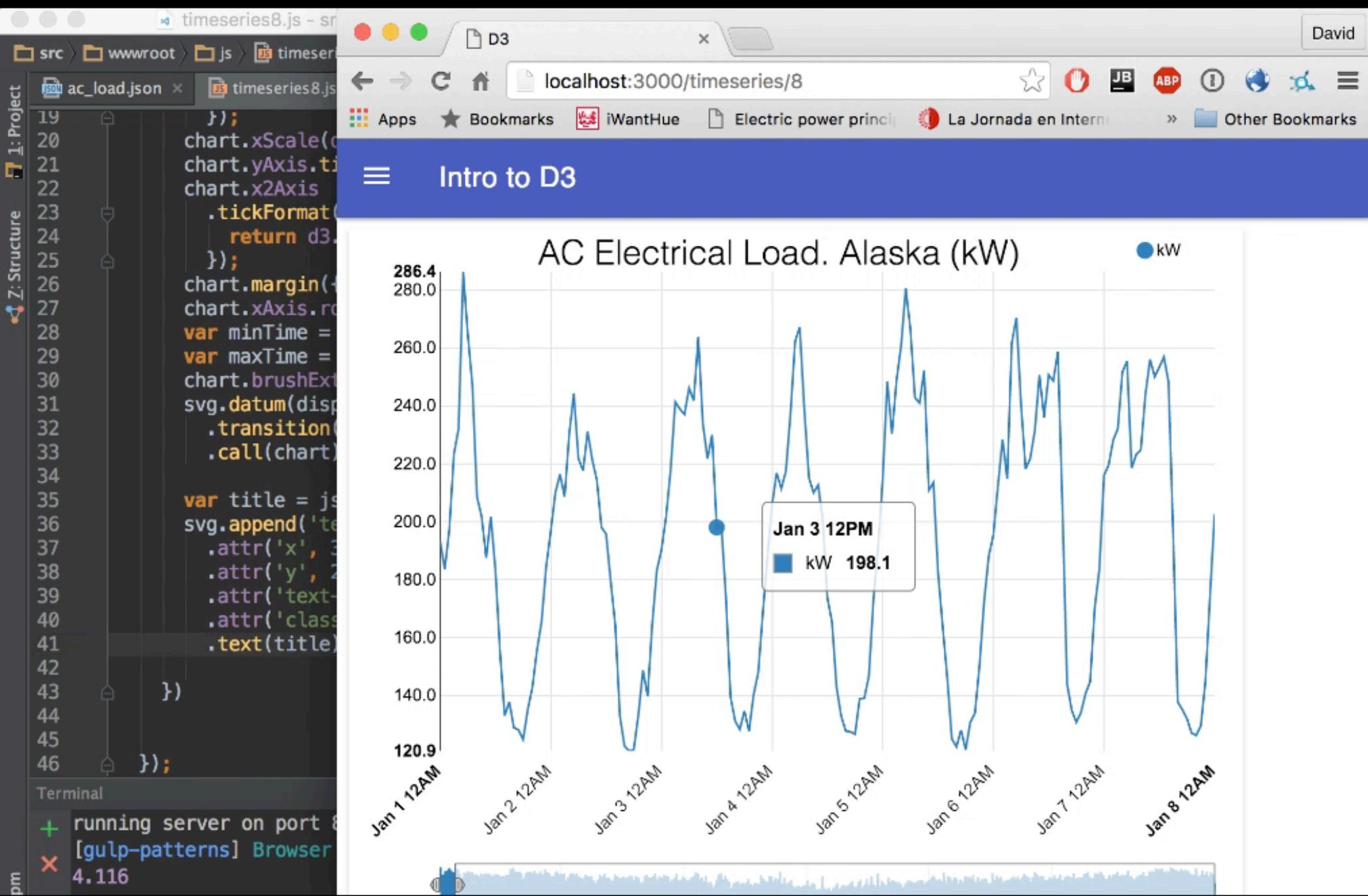
Terminal

+ running server on port 8080
x [gulp-patterns] Browser Connected: Chrome, version: 48.0
.2564.116

Encapsulating Complexity

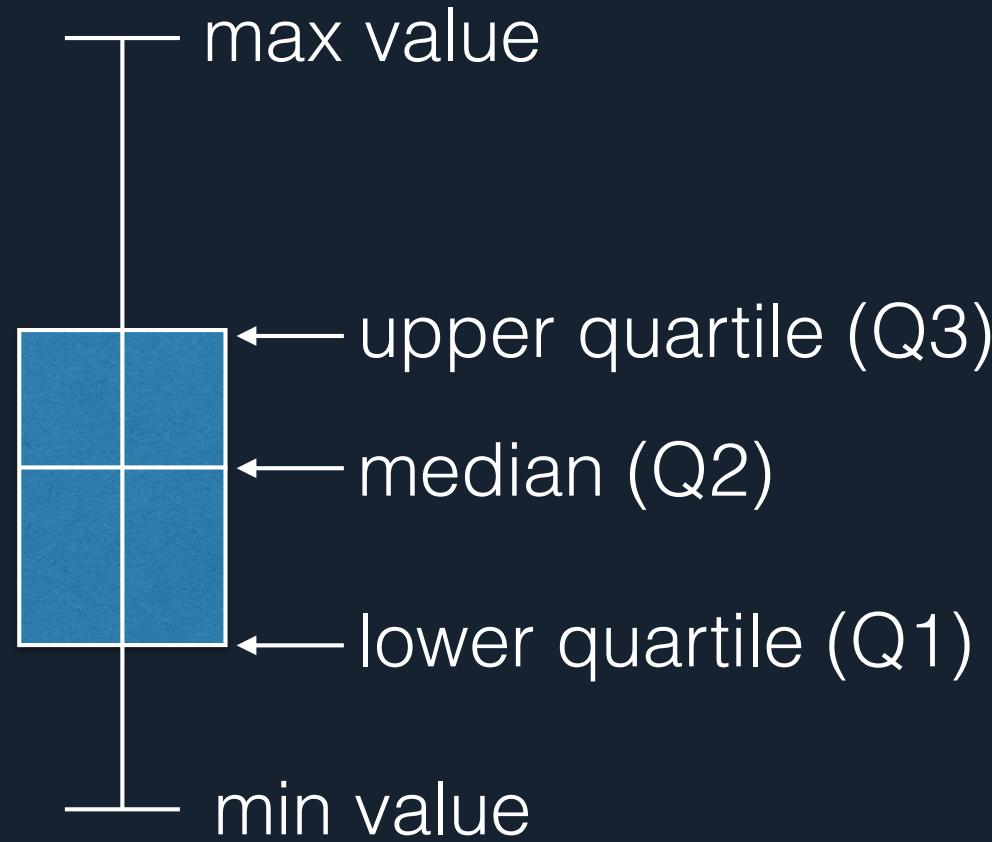
NVD3 Re-usable charts for [d3.js](#)







nv.models.boxPlotChart()



ac_load_boxplot.json - src - [~/_Code/_d3_workshop/src]

src wwwroot data boxplots ac_load_boxplot.json Unnamed timeseries9.js ac_load_boxplot.json timeSeries3.js

1: Project title

```
1  "title": "AC Electrical Load. Yakutat, Alaska",
2  "units": "kW",
3  "data": [
4      {
5          "label": "Jan",
6          "values": {
7              "Q1": 140.1,
8              "Q2": 201.7,
9              "Q3": 226.0,
10             "whisker_low": 113.9,
11             "whisker_high": 306.5
12         }
13     },
14     {
15         "label": "Feb",
16         "values": {
17             "Q1": 144.4,
18             "Q2": 208.0,
19             "Q3": 233.7,
20             "whisker_low": 118.5,
21             "whisker_high": 277.8
22         }
23     },
24     {
25         "label": "Mar",
26         ...
27     }
28 ]
```

1: Structure

Terminal

```
[08:04:26] [nodemon] starting `node server/app.js`
x running server on port 8080
x [gulp-patterns] [proxy error] socket hang up
```

D3 David

Apps Bookmarks iWantHue Other Bookmarks

Intro to D3

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)

Qualitative Color Palettes

Number of data classes: 8

Nature of your data: sequential diverging qualitative

Pick a color scheme:

Only show:

colorblind safe
 print friendly
 photocopy safe

Context:

roads
 cities
 borders

Background:

solid color
 terrain

color transparency

how to use | updates | downloads | credits

COLORBREWER 2.0
color advice for cartography

8-class Pastel1

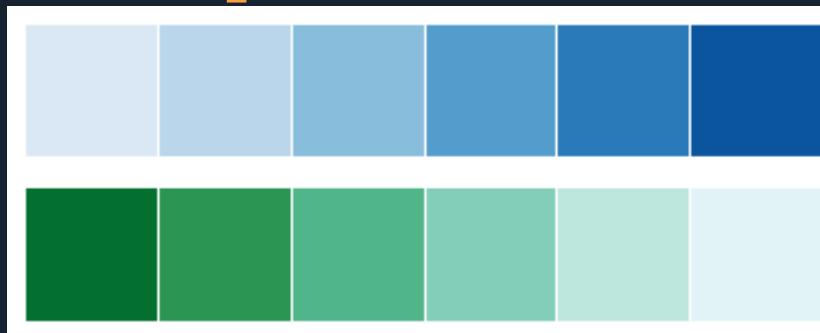
EXPORT

HEX

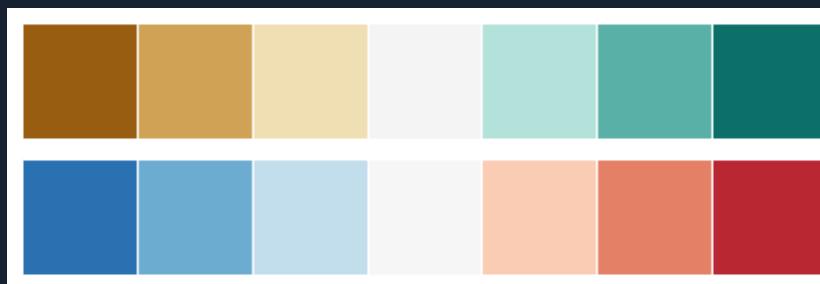
#fbbaae
#b3cde3
#ccebc5
#decbe4
#fed9a6
#ffffcc
#e5d8bd
#fddaec

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

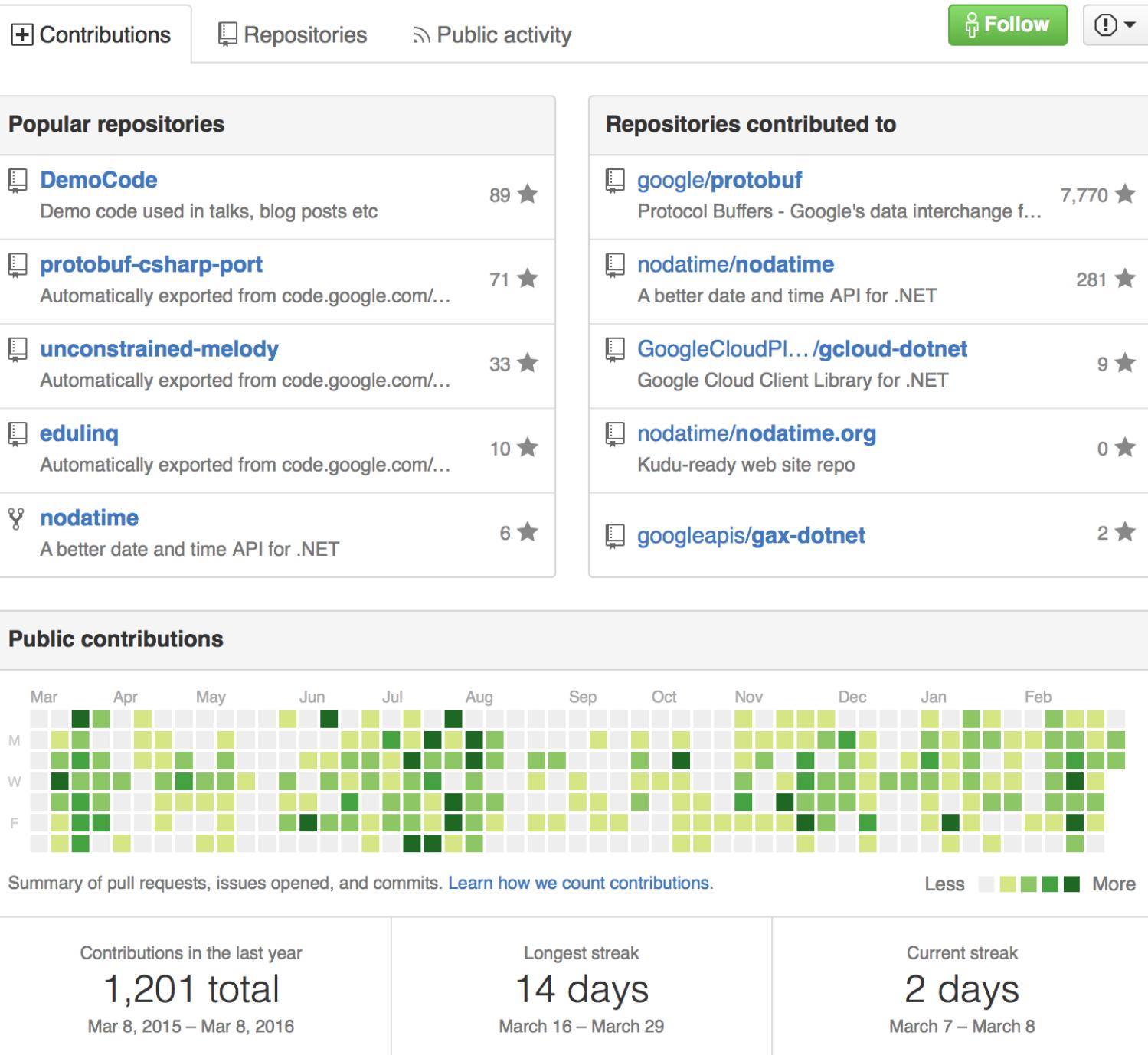
0

Followers

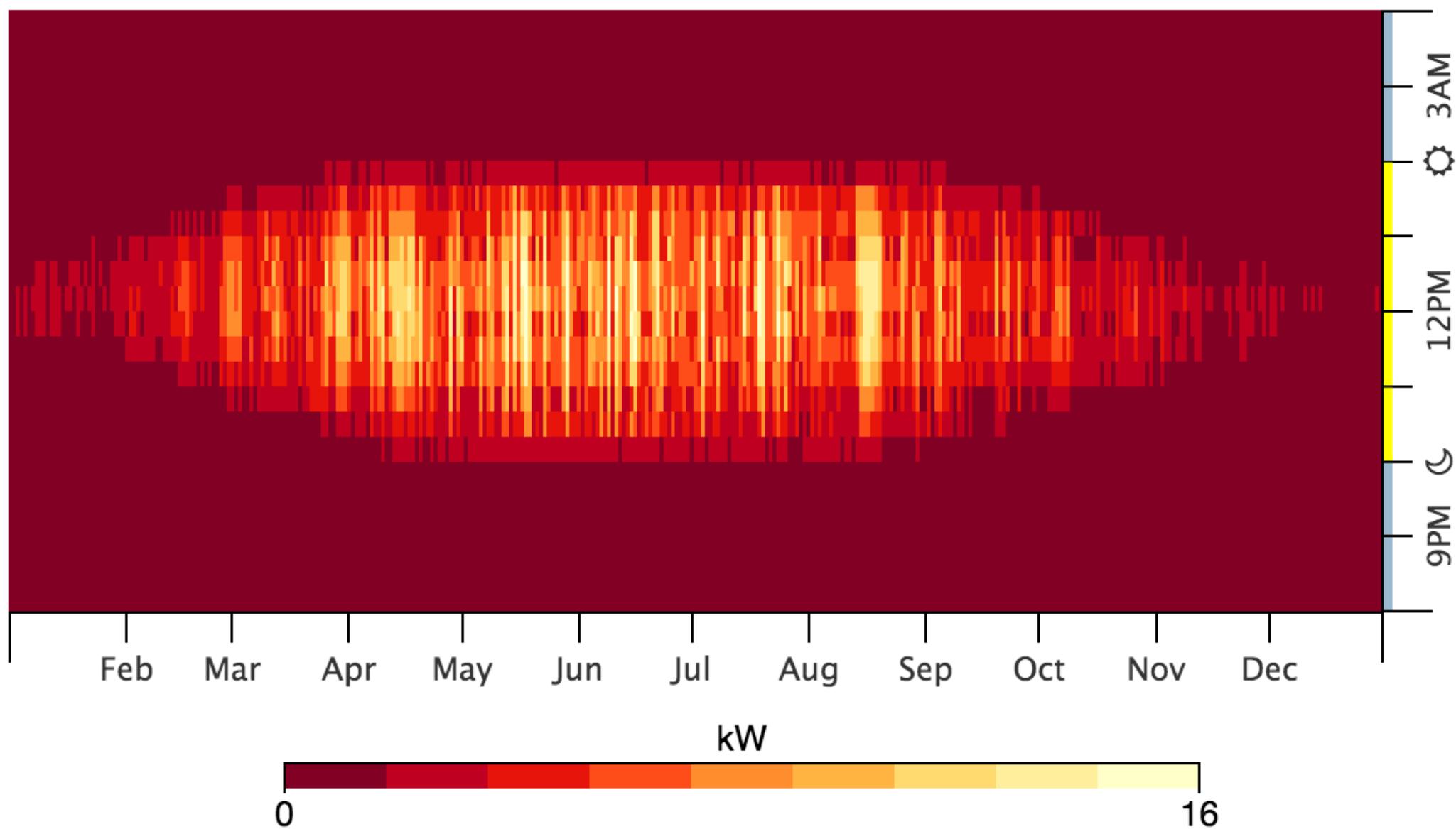
Starred

Following

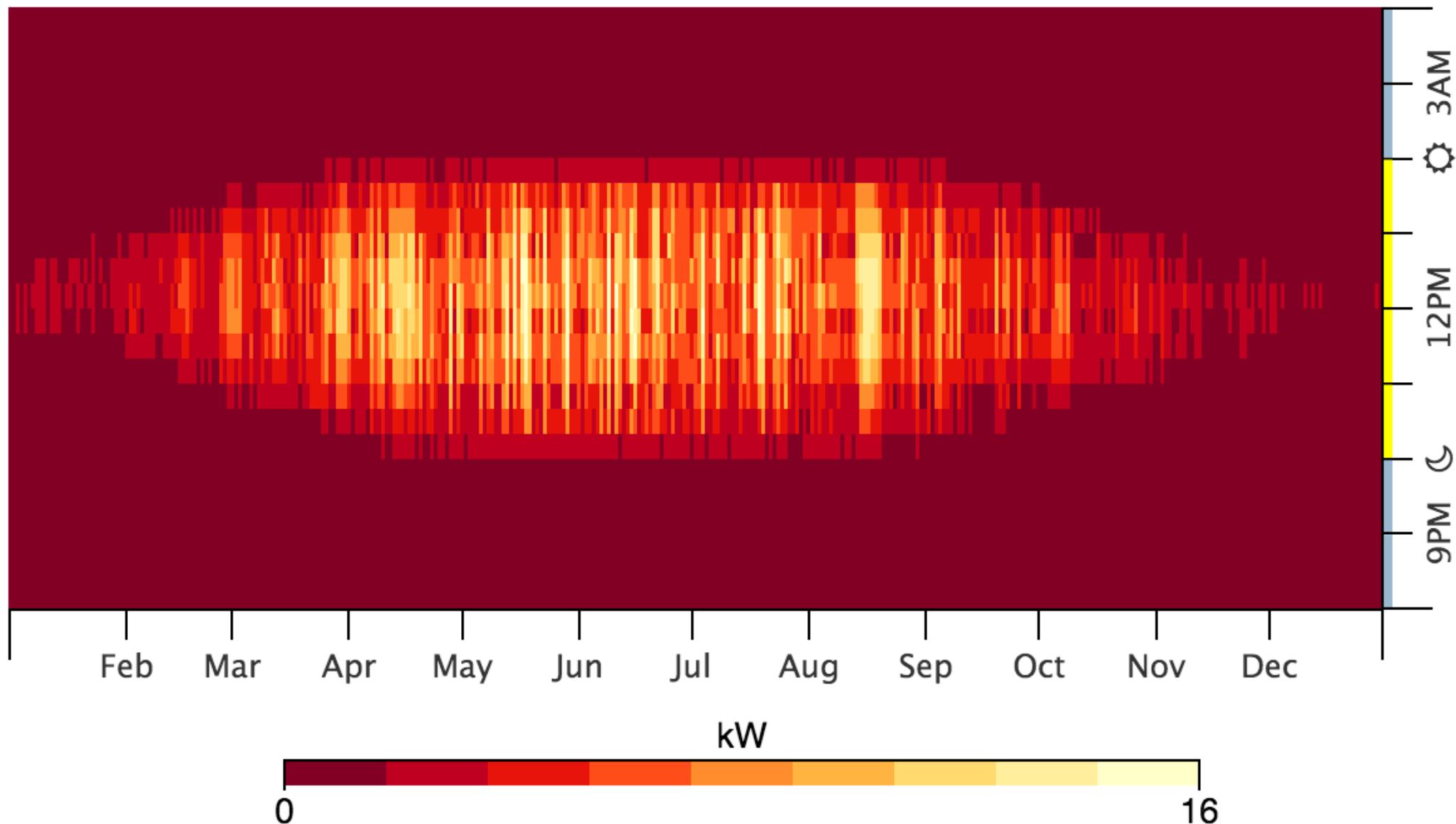
Organizations



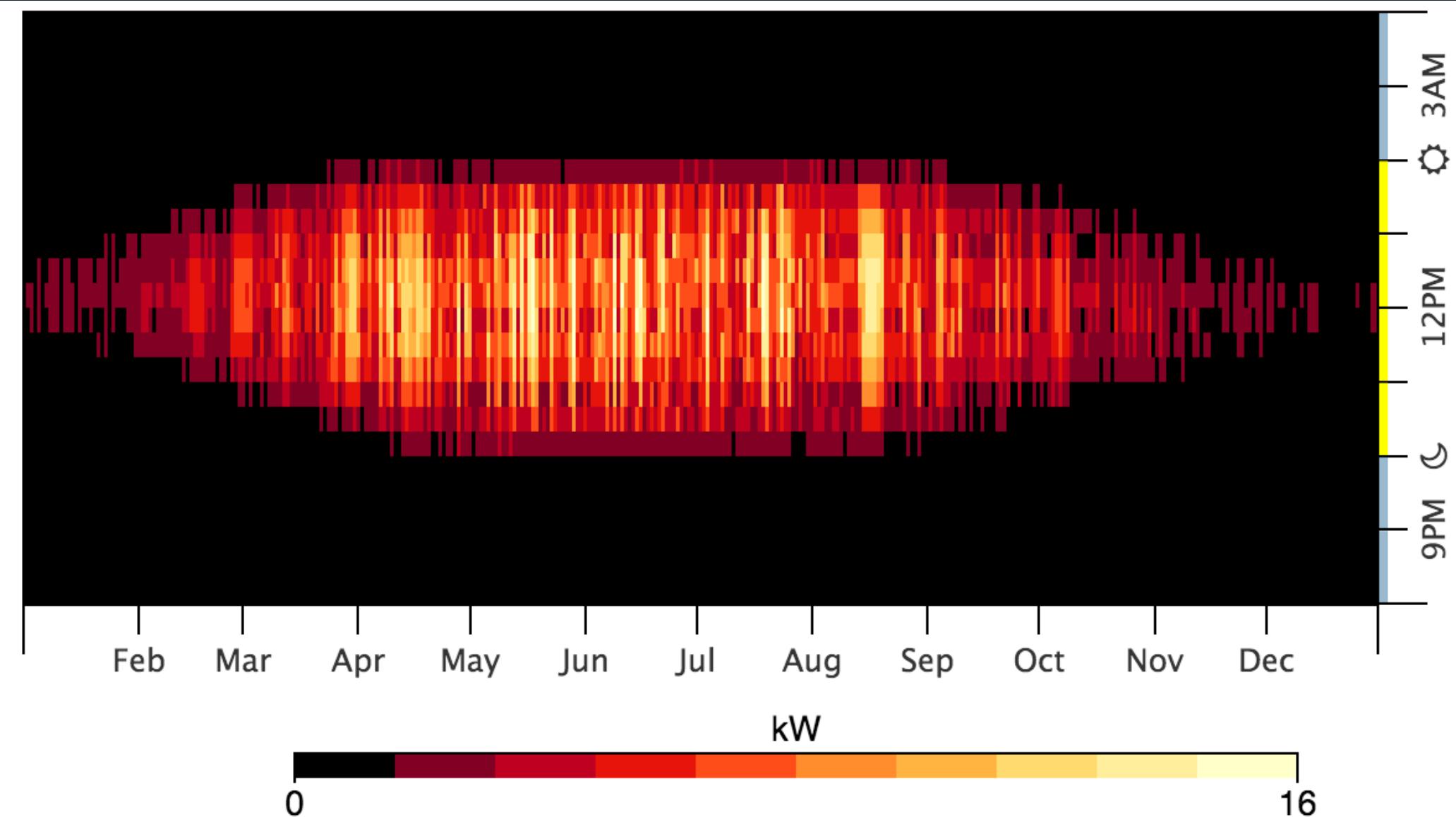
8760 Heat Map



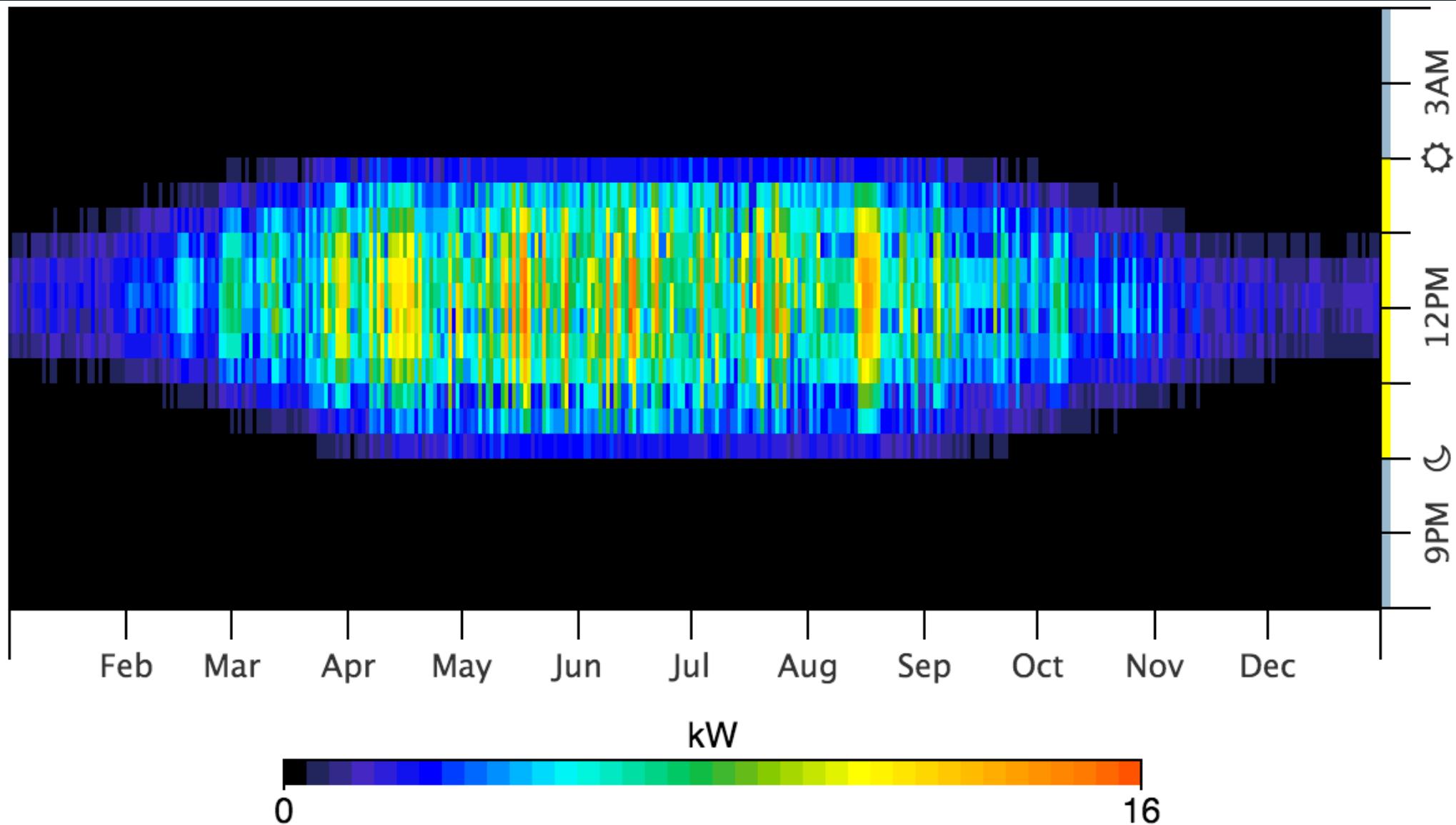
Sunny Alaska?



better zero-crossing



go rainbow



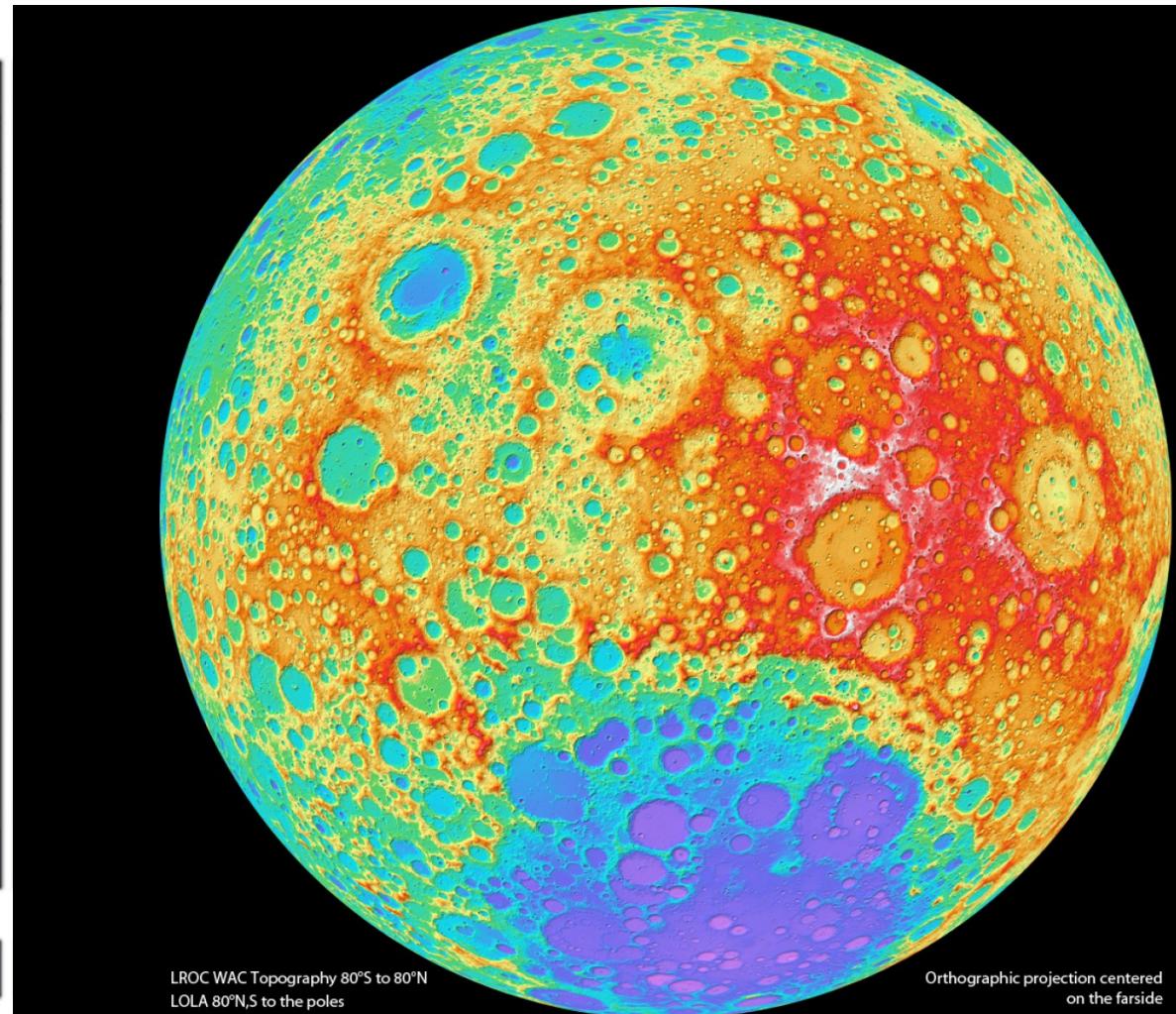
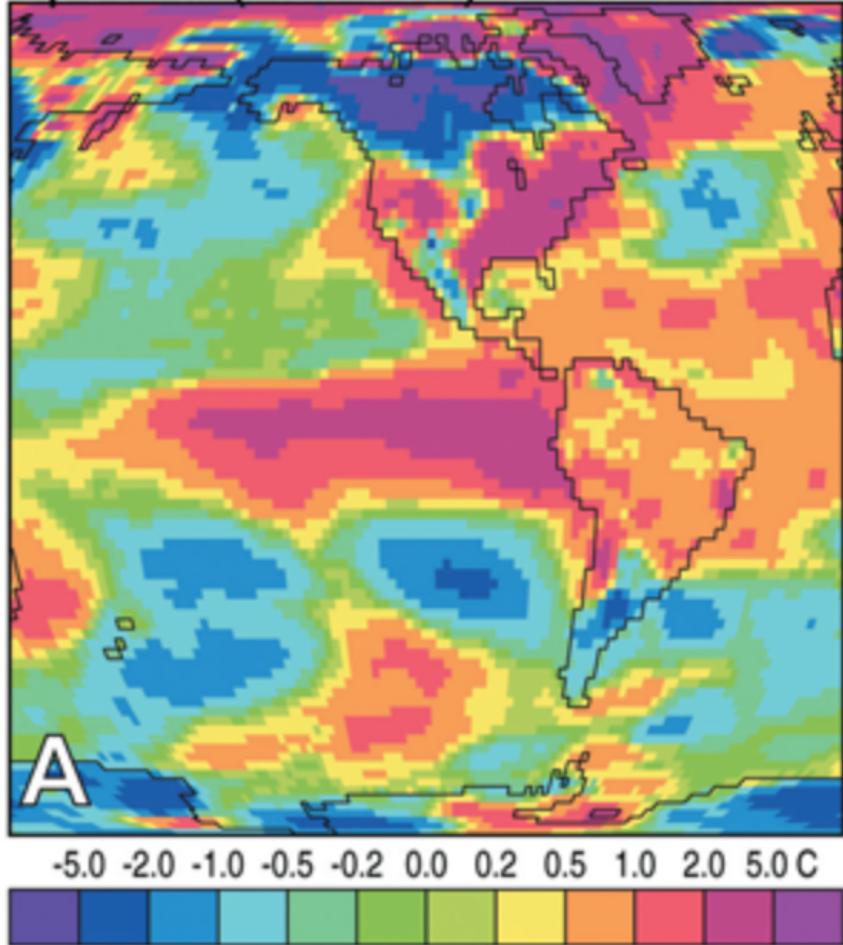
Dear NASA: No More Rainbow Color Scales, Please

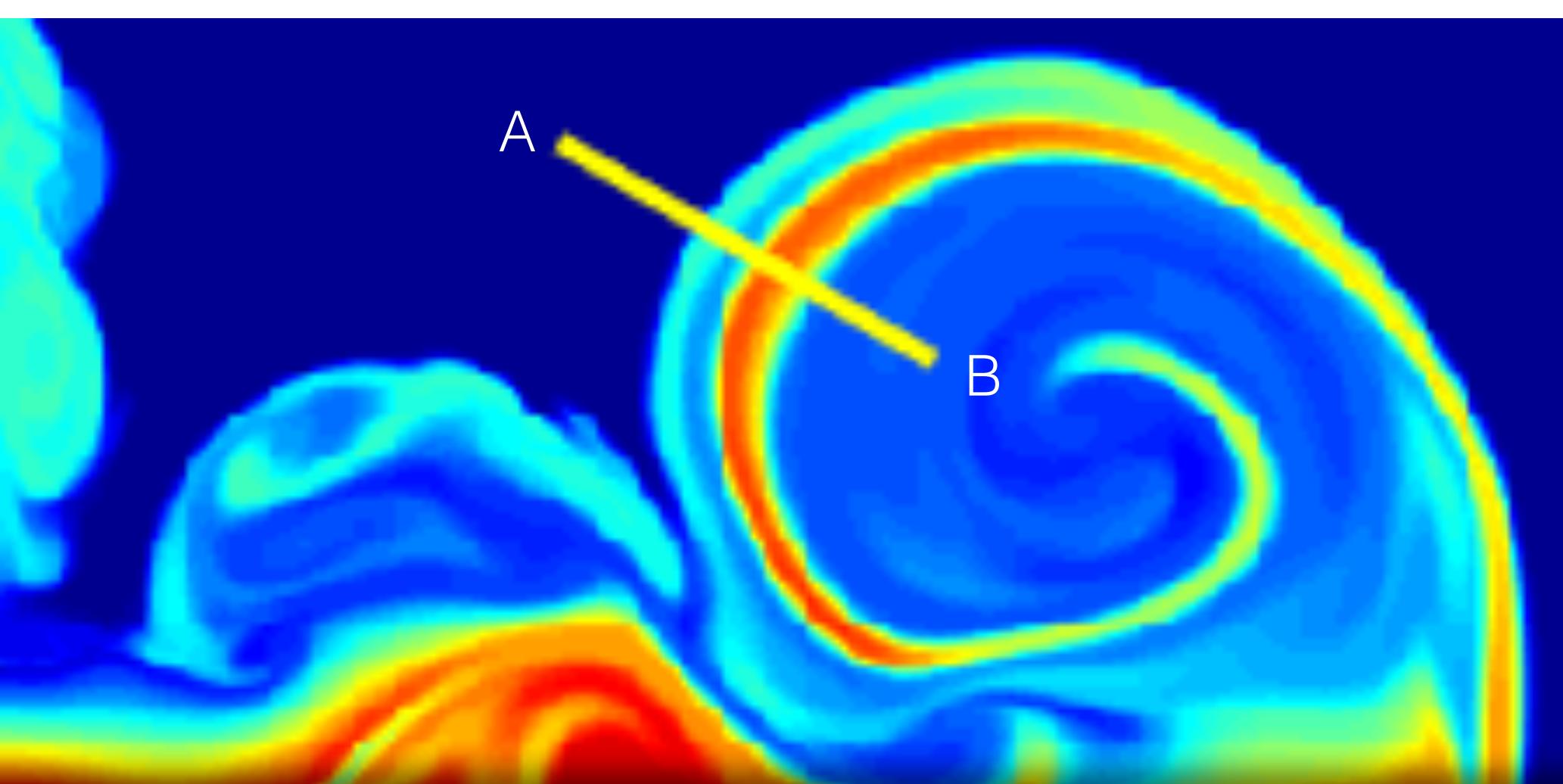


Drew Skau

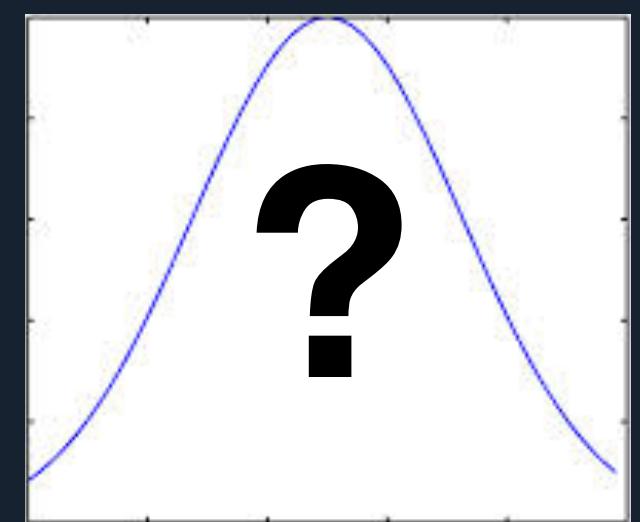
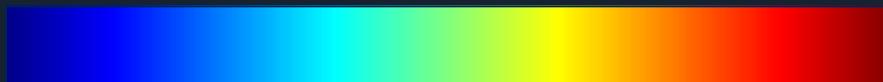
published on April 4, 2012 in **Design**

Spectral (Rainbow) Color Scale

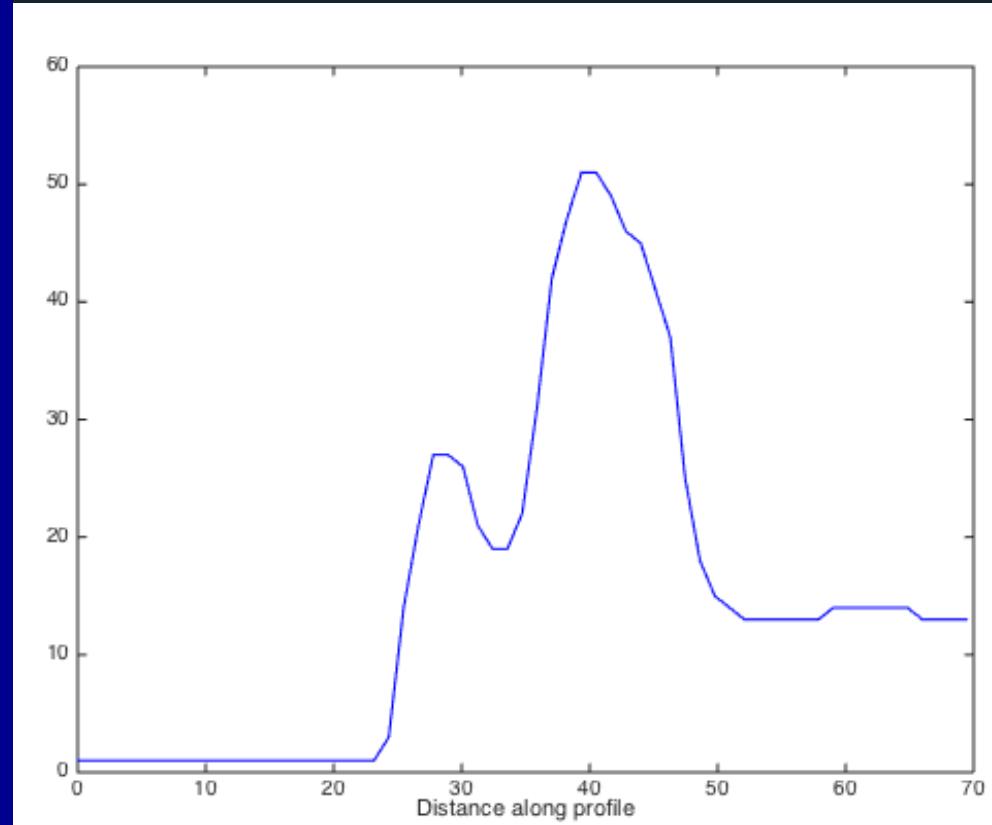
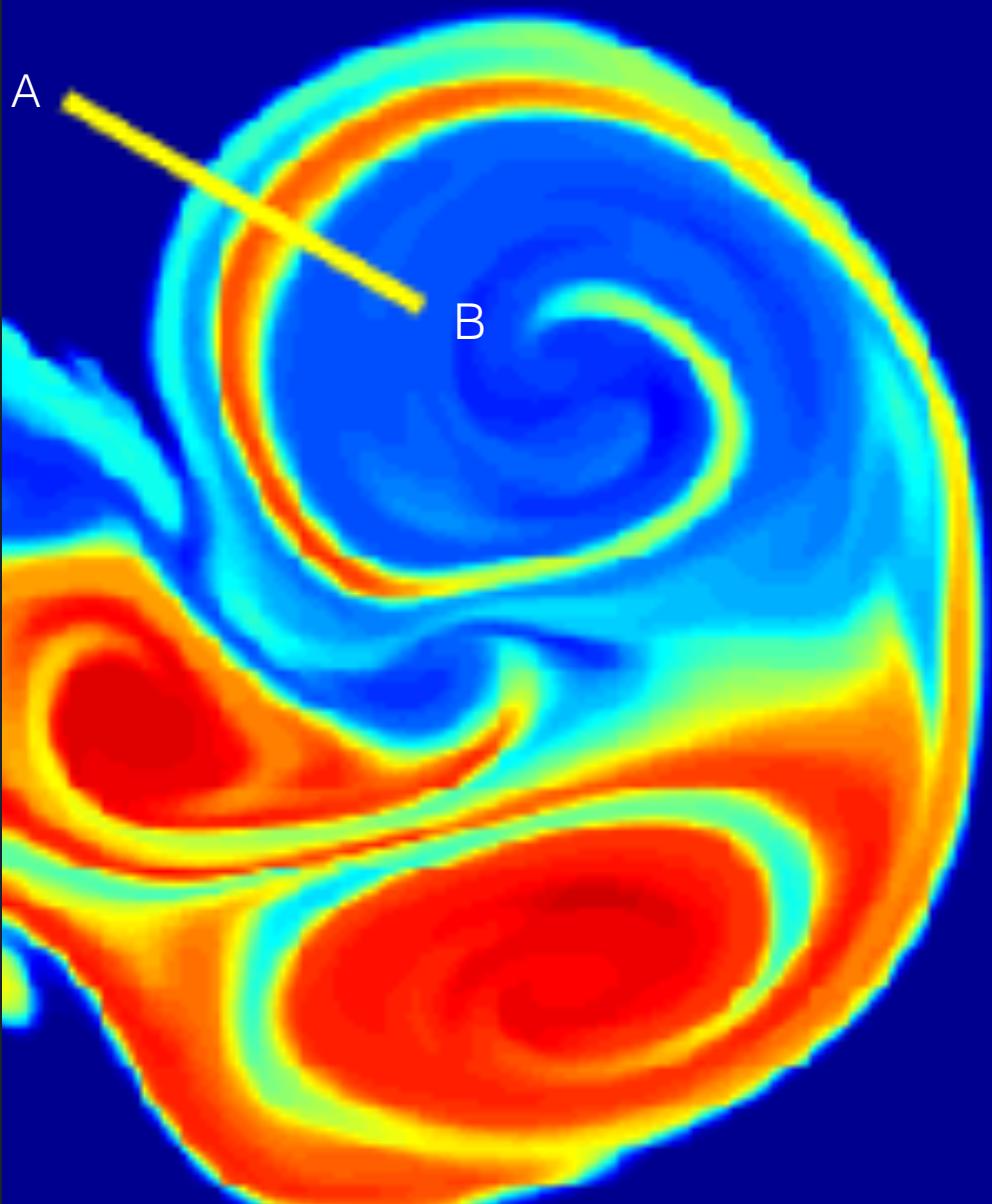




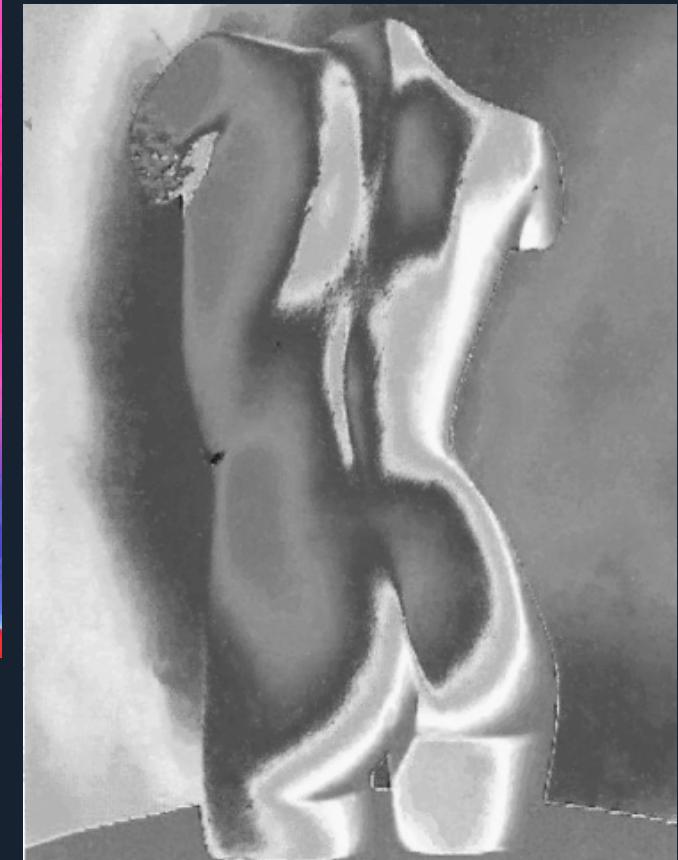
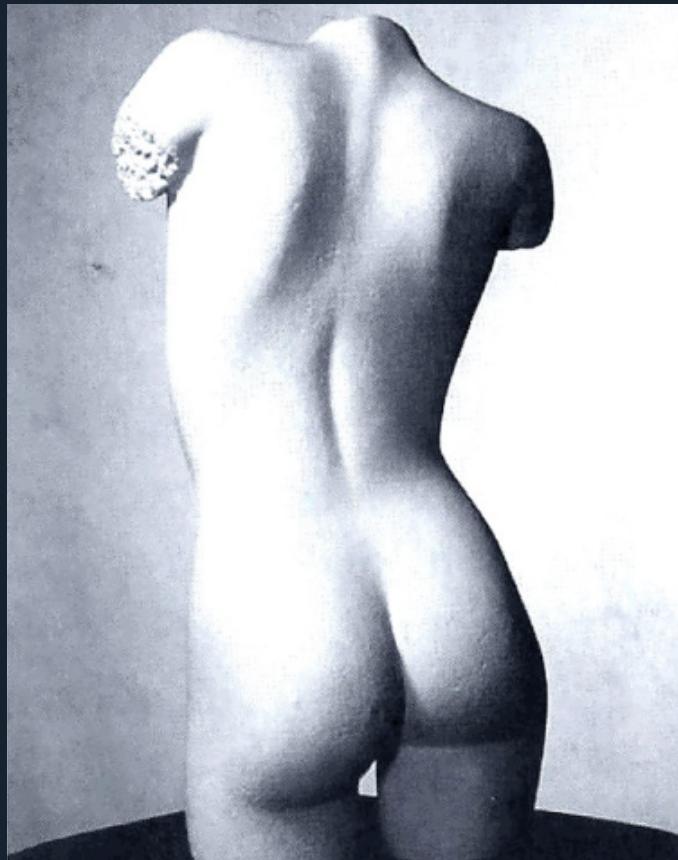
**Guess the 2-d Profile of this shape.
What do the stripes mean?**



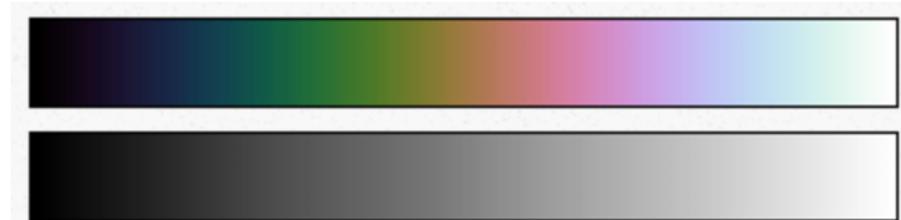
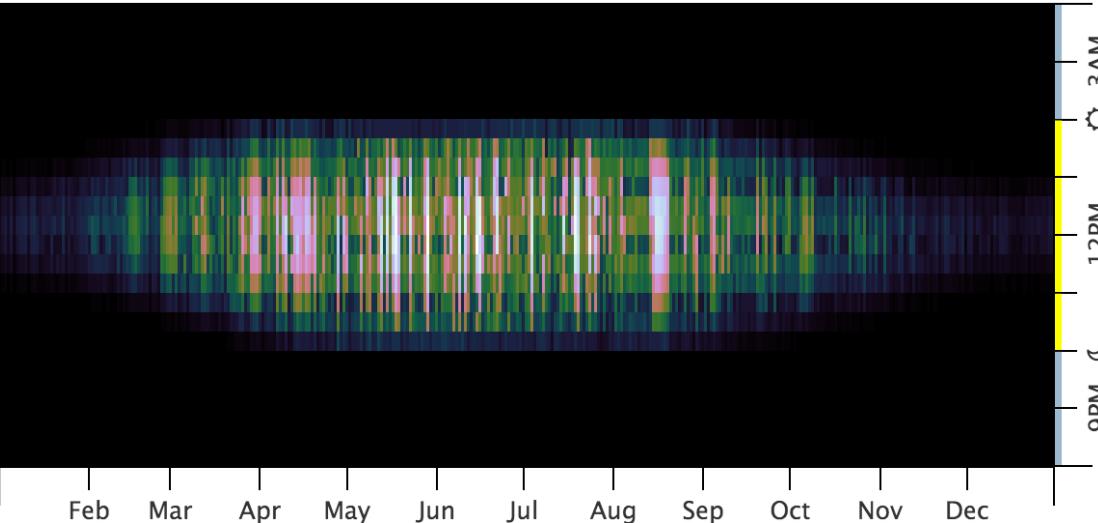
Some of the color stripes are meaningless. But not all of them!



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



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

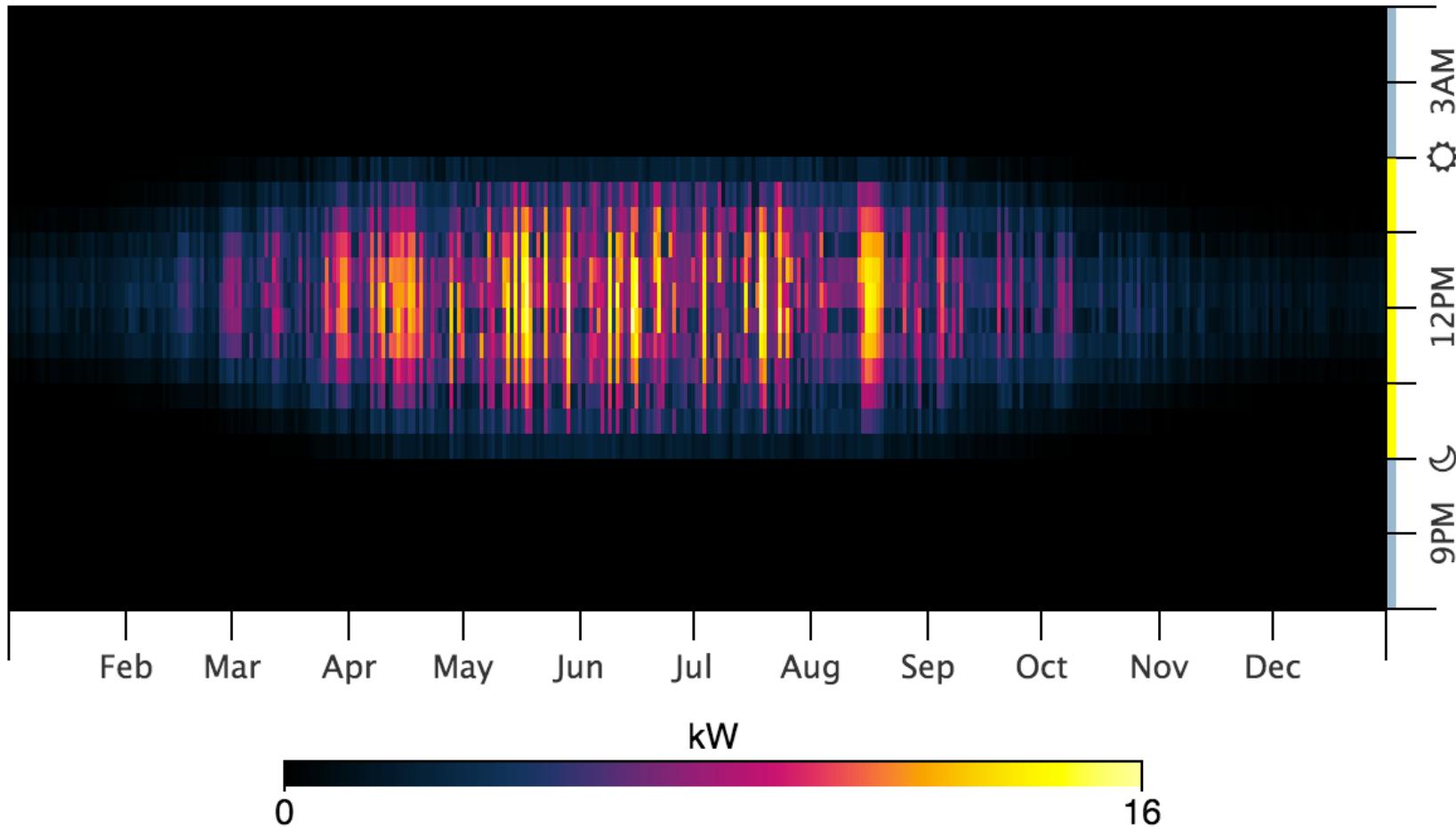
Optimize.

2 years ago

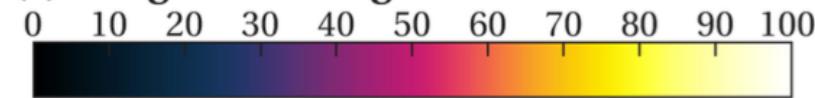
 README.md

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

morganstemning



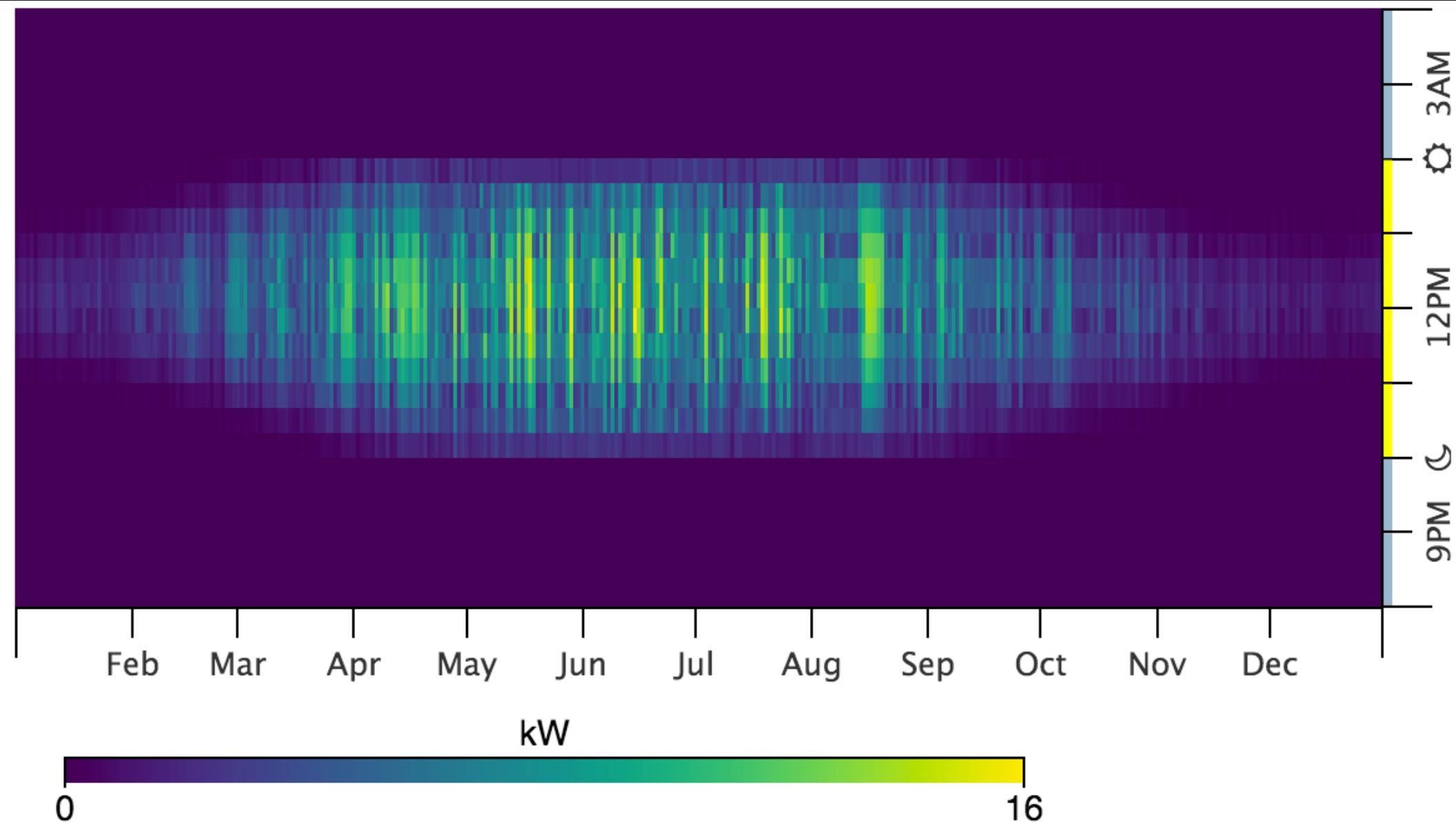
(c) Morgenstemning

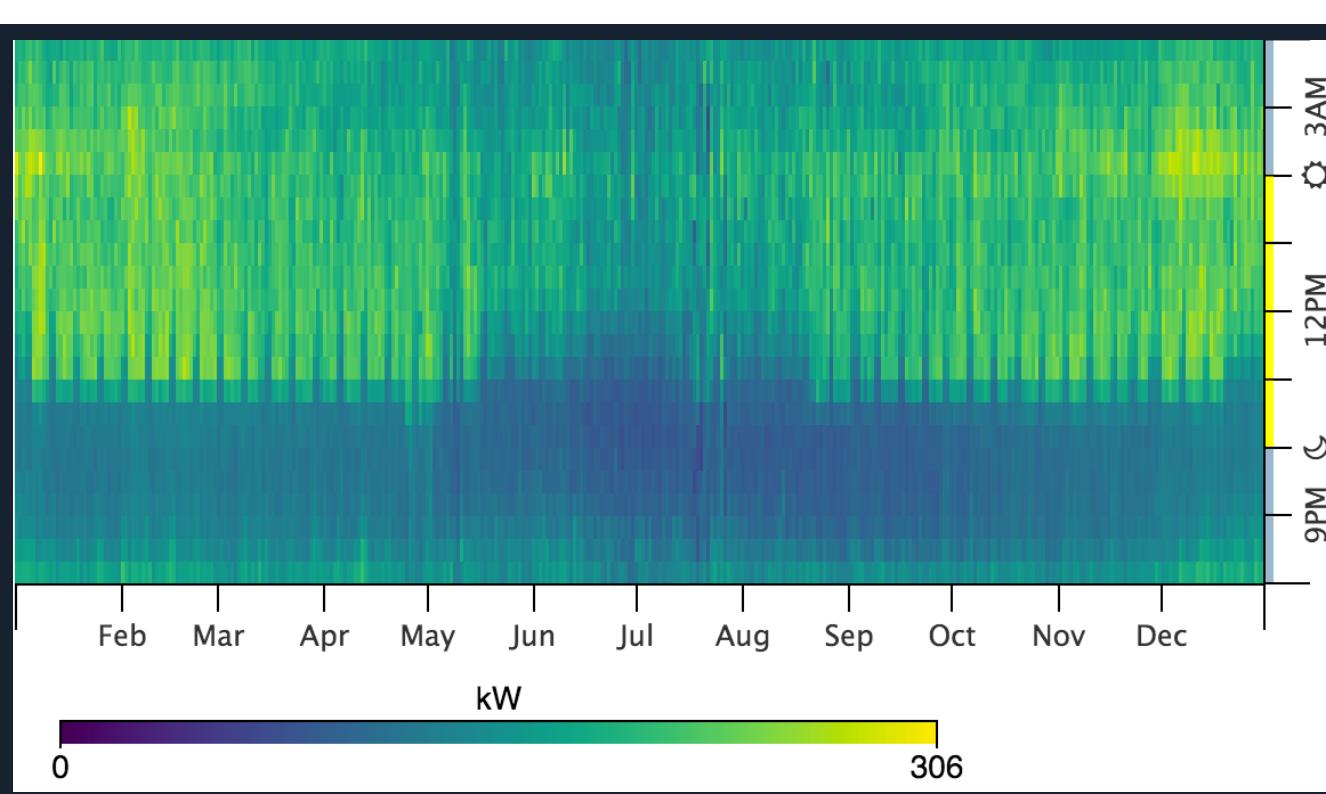


Deuteranope perception

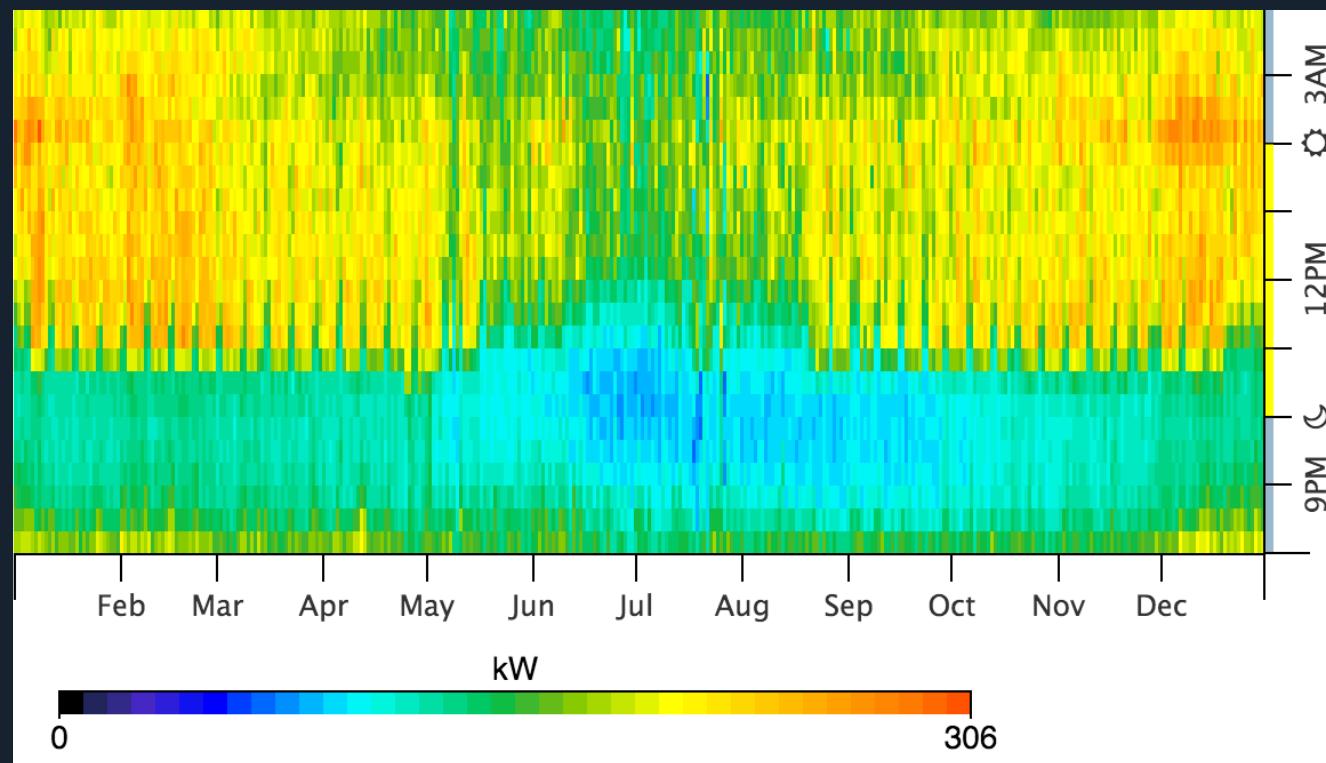
Protanope perception

viridis



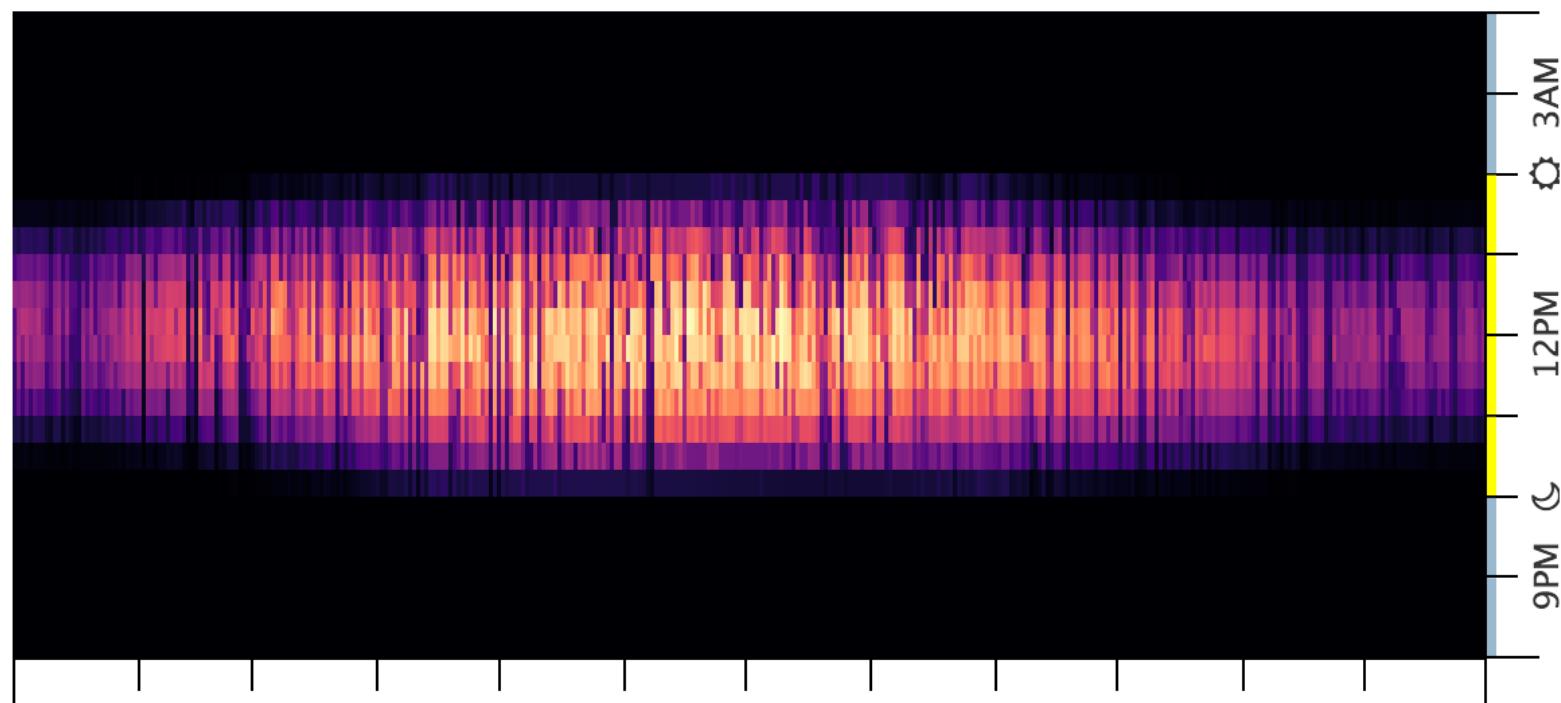


Viridis

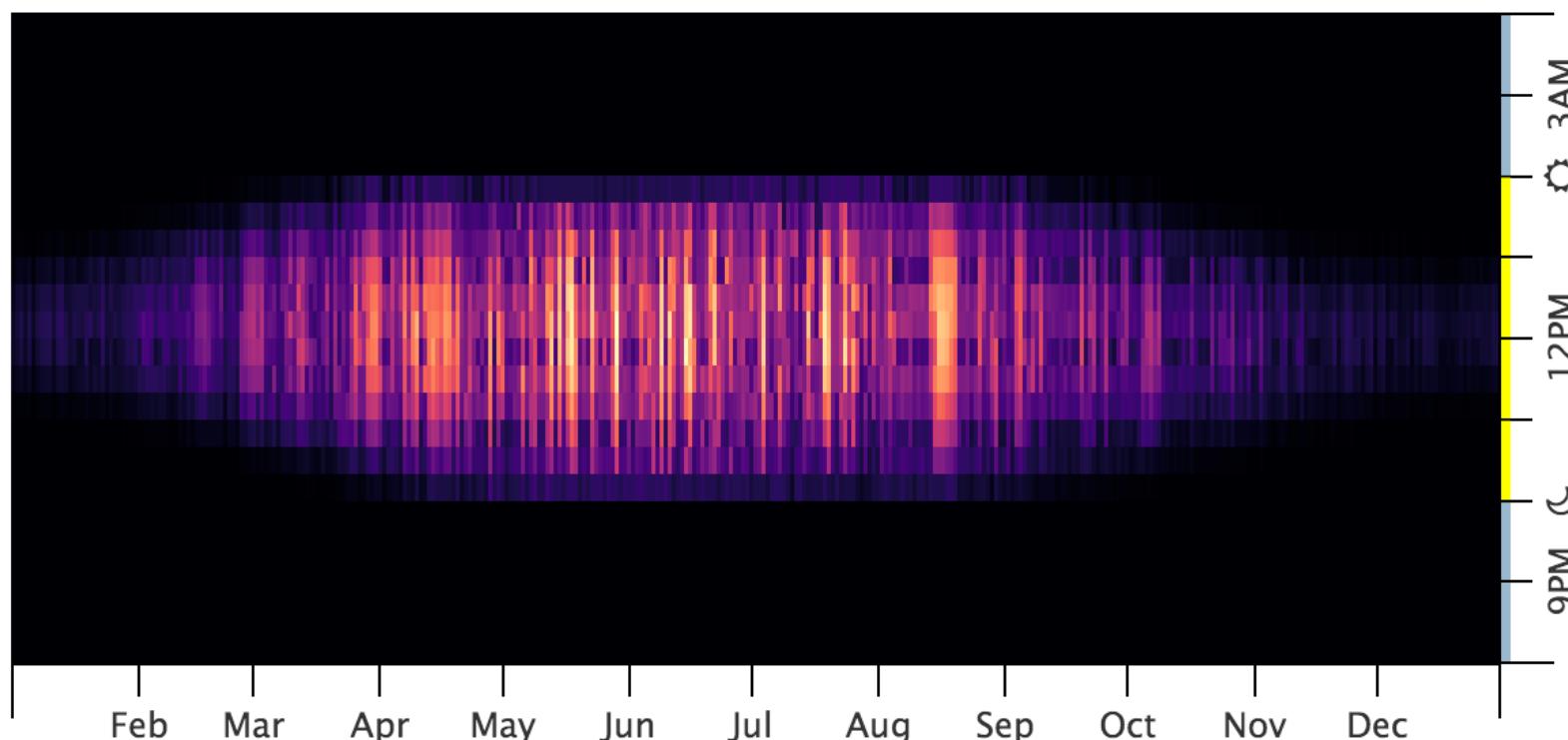


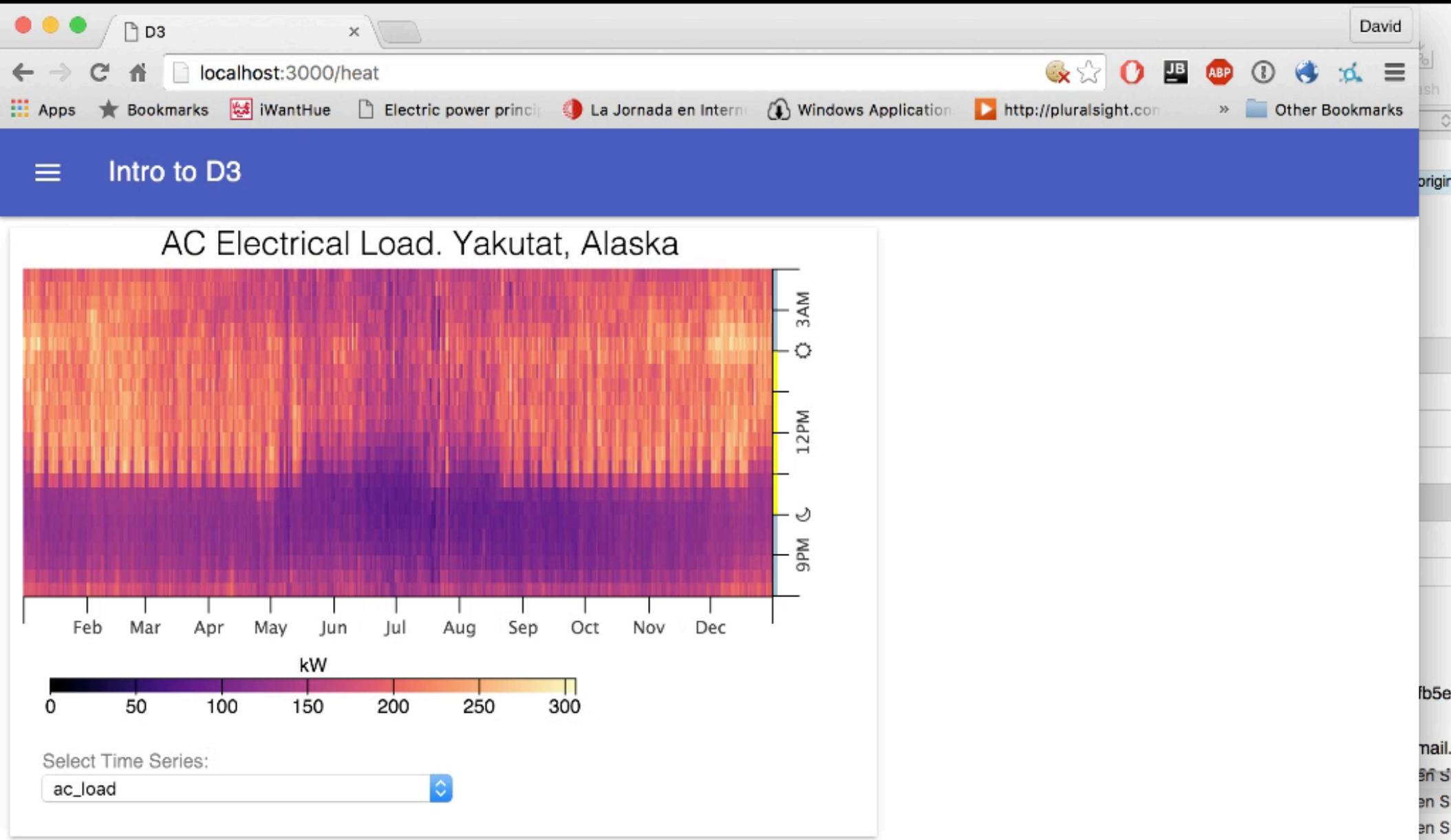
Rainbow (Jet)

PV Solar Output. Boulder, Colorado



PV Solar Output. Yakutat, Alaska









The End