Unlocking the Potential of D3.js with Qlik Sense APIs

Using a popular JavaScript library to build things beyond the Qlik Sense out-of-box capabilities

Who am I? Kevin McGovern


BRANCH

Kevin McGovern

- Data Visualization Consultant at Slalom
- Qlik Branch Contributor
- Builds things with D3

Questions we will cover

- What is D3?
- How is D3 used?
- Where would you use D3 in Qlik Sense Mashups?
- What do you need to do differently for Extensions?

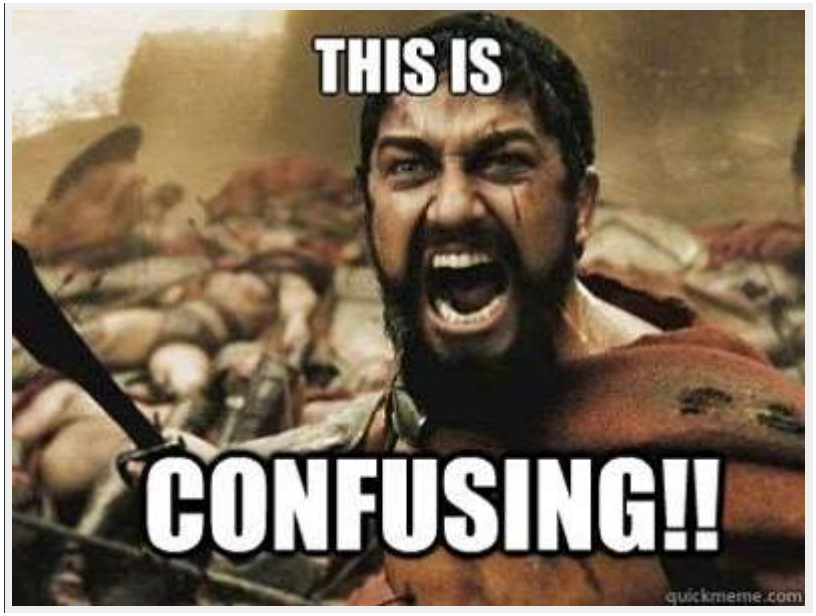
What is D3?

What D3 isn't

- Not a charting library
- Not a drawing library
- Not optimized for canvas and WebGL
- Not a framework

What D3 is

D3.js is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

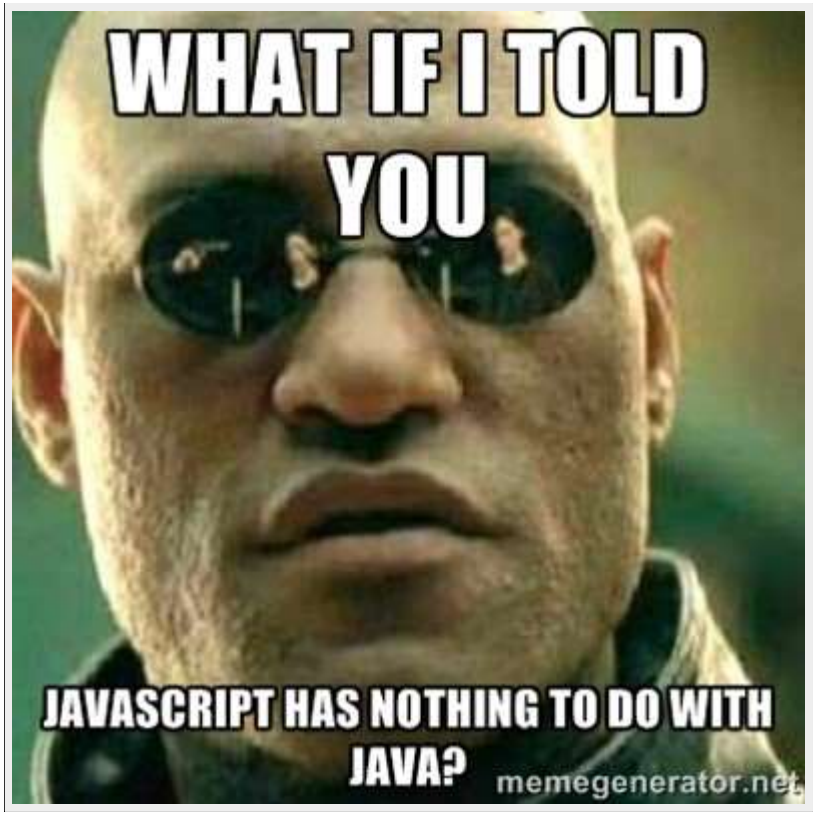

THIS IS
CONFUSING!!
quickmeme.com


PATIENCE YOU MUST HAVE
WORTH IT WILL BE
memecrunch.com

Backing Up

"A JavaScript Library"

What is JavaScript?



D3: "A JavaScript Library"

What is JavaScript?

- HTML sets document structure
- CSS sets styling
- JS informs your browser how to display content dynamically

D3: "A JavaScript Library"

What is a js library?

Extending what native js can do, makes developers lives easier

- jQuery
- Angular
- React
- Ember
- Three.js
- Node.js
- D3.js

D3: "manipulating documents based on data"

Browser renders HTML, CSS, JS to create Document Object Model

...or DOM

The DOM

More than just a fast and furious character API that lets you interact with rendered web pages

Where does D3 come in?

- Bind data to elements in the DOM
- Change existing elements based on bound data
- Add new elements when more data added
- Remove elements when data changes
- Basically...

HTML Magic

D3 in Action

Finding elements, binding data, creating new elements

Using Scott Murray's Awesome Tutorial

An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

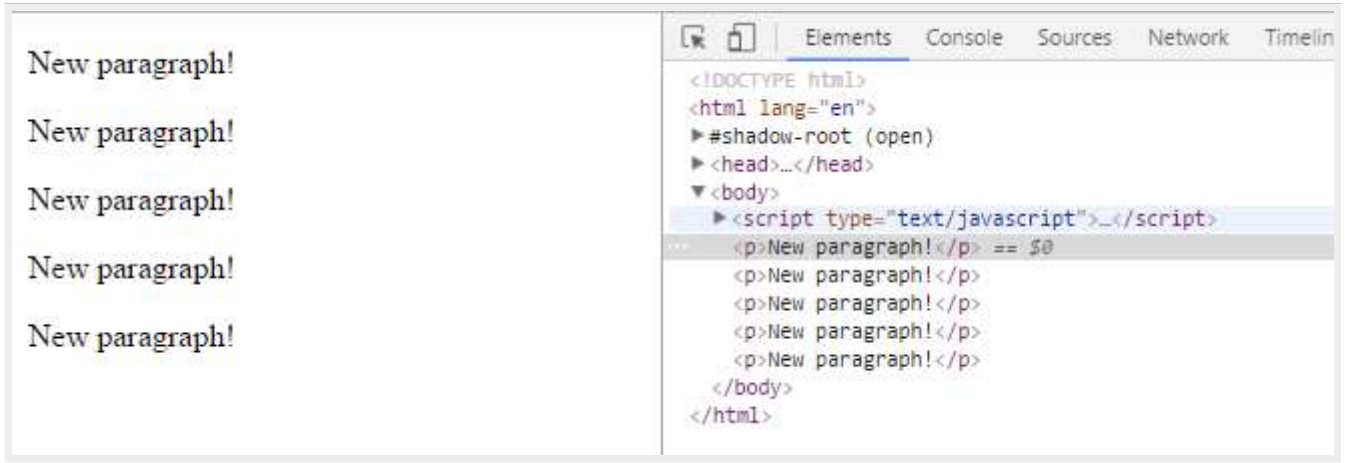
An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

An Example

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text("New paragraph!");
```

An Example - Results




```
console.log(d3.selectAll("p"))
```

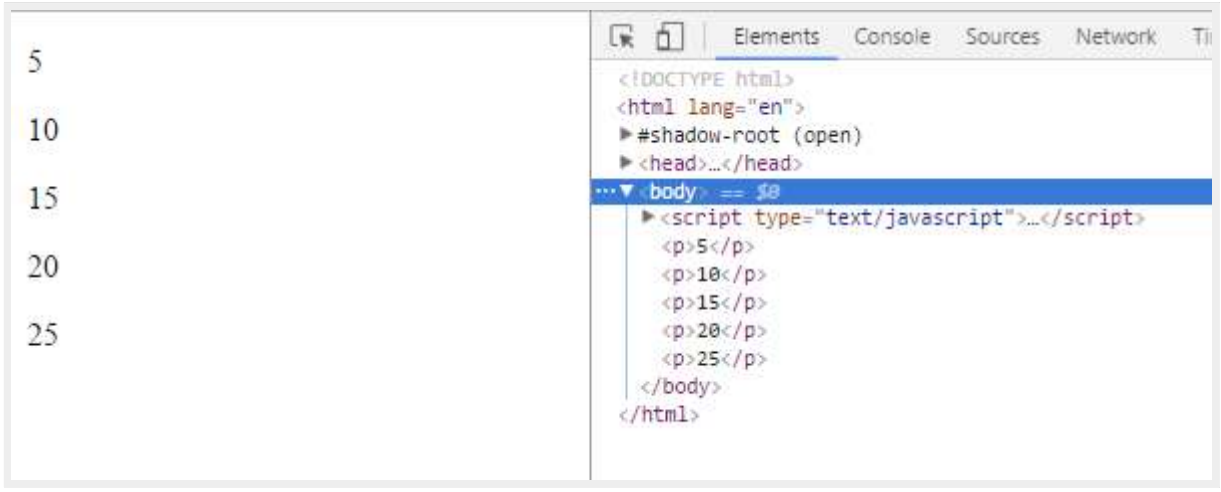
An Example - Exploring the Elements

```
> console.log(d3.selectAll("p"))

▼[Array(5)]
  ▼0: Array(5)
    ▼0: p
        accessKey: ""
        align: ""
        assignedSlot: null
      ▶attributes: NamedNodeMap
        translate: true
        webkitdropzone: ""
        __data__: 5
      ▶__proto__: HTMLParagraphElement
    ▶1: p
    ▶2: p
    ▶3: p
    ▶4: p
    ▶parentNode: document
      length: 5
    ▶__proto__: Array(0)
  length: 1
▶__proto__: Array(0)
```

Modified Example - Using Data to Add to Elements

```
var dataset = [ 5, 10, 15, 20, 25 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text(function(d) { return d; });
```

Modified Example - Results

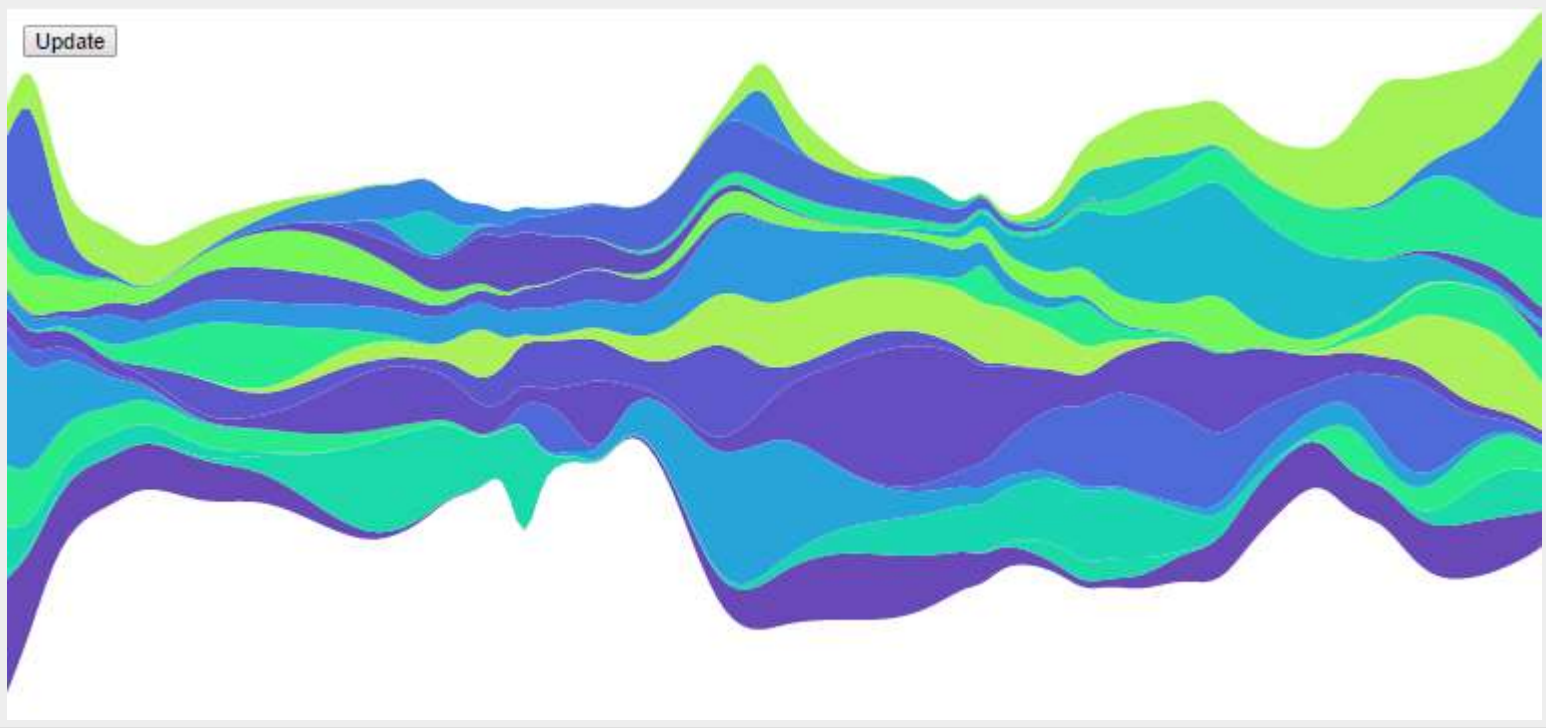


Modified Example - Changing the Data

```
var dataset = [ 5, 200, 15, 20, 25, 50 ];

d3.select("body")
        .selectAll("p")
        .data(dataset)
        .enter()
        .append("p")
        .text(function(d) { return d; });
```

Modified Example - Results

Cool chart but why do I care?

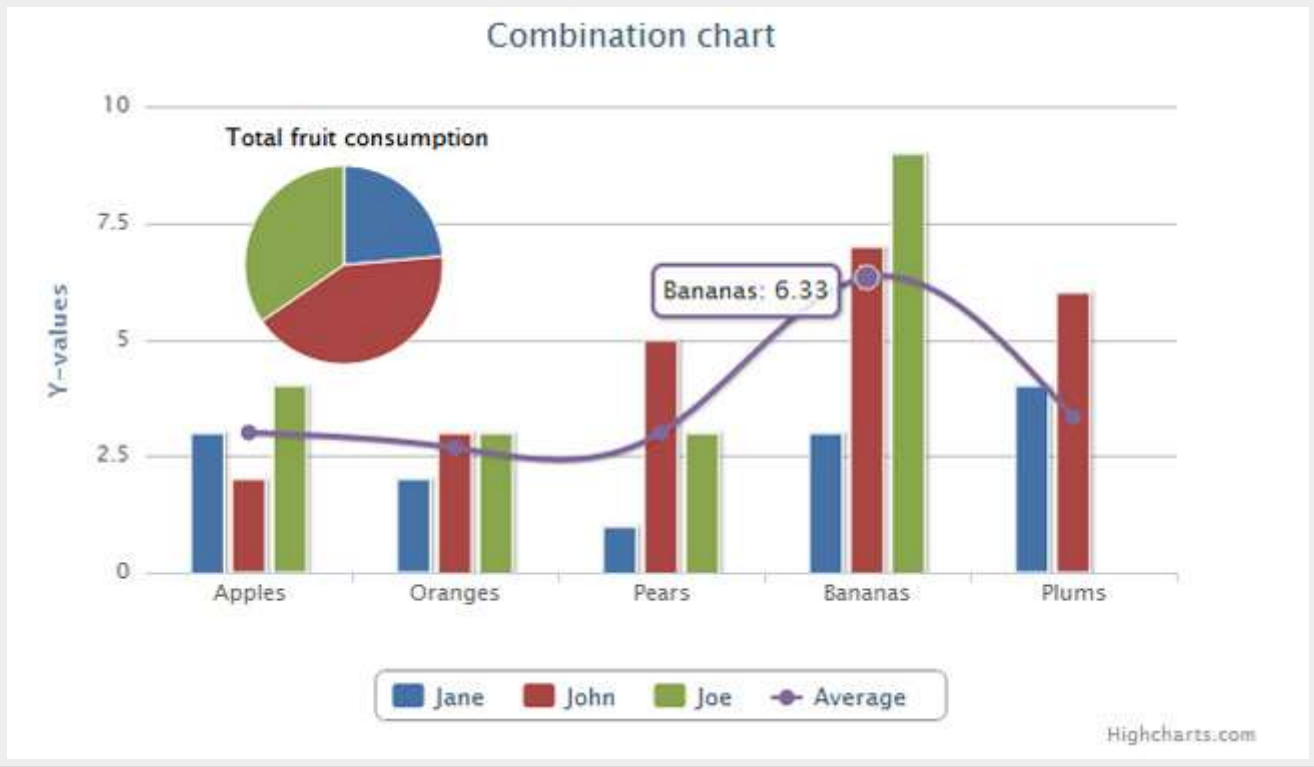


Dynamic data manipulation is really useful for data visualization

- Data changes over time
- User interactions
  - Clicks
  - Hover
  - Time-based delays

D3 looks hard, what about charting libraries?

- Few chart options
- Customization is difficult
- Limited page interaction

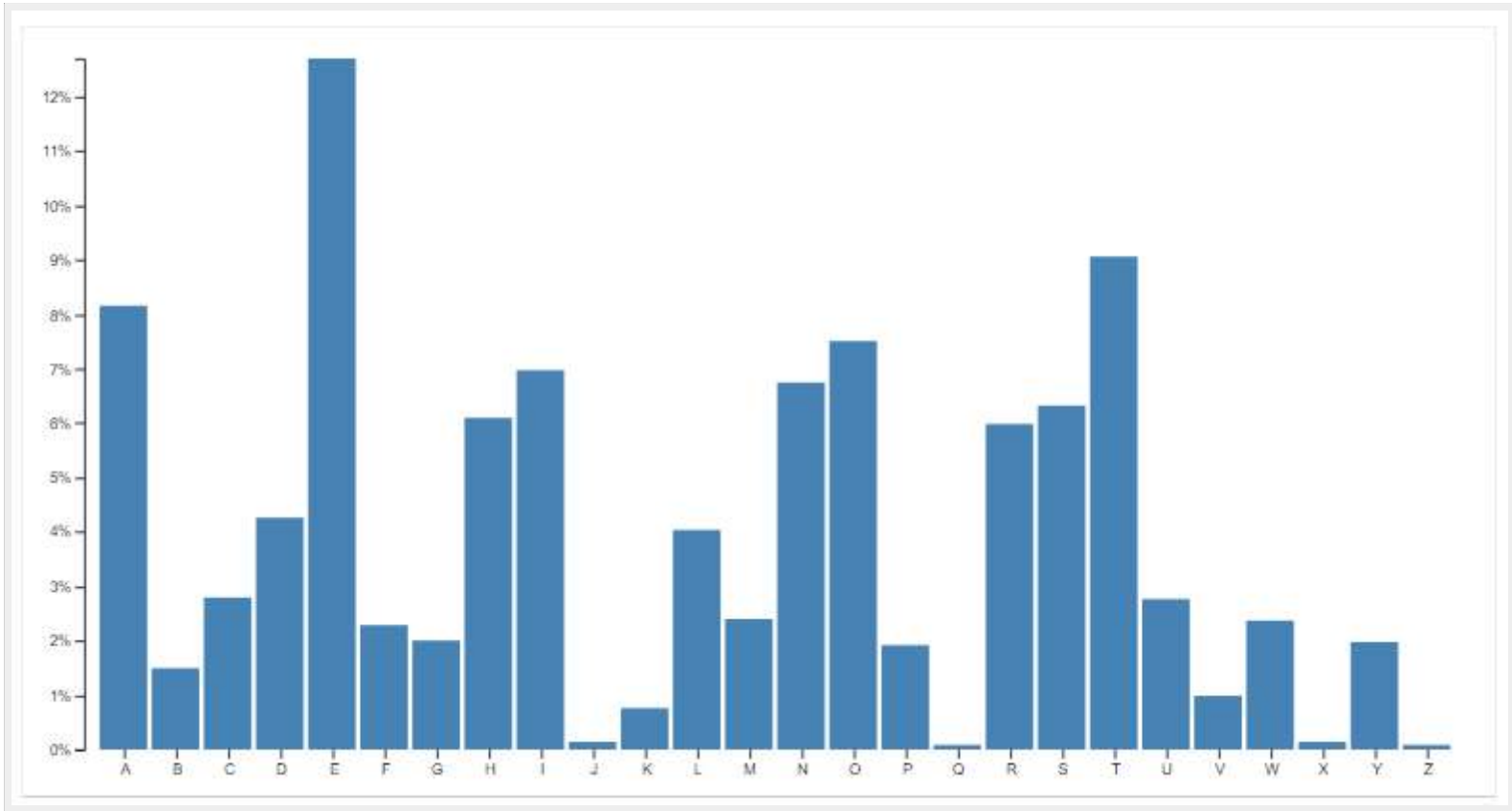

Combination chart
Total fruit consumption
Y-values
10
7.5
5
2.5
0
Bananas: 6.33
Apples
Oranges
Pears
Bananas
Plums
Jane
John
Joe
Average
Highcharts.com

D3 is more than a way to just change data on the page dynamically

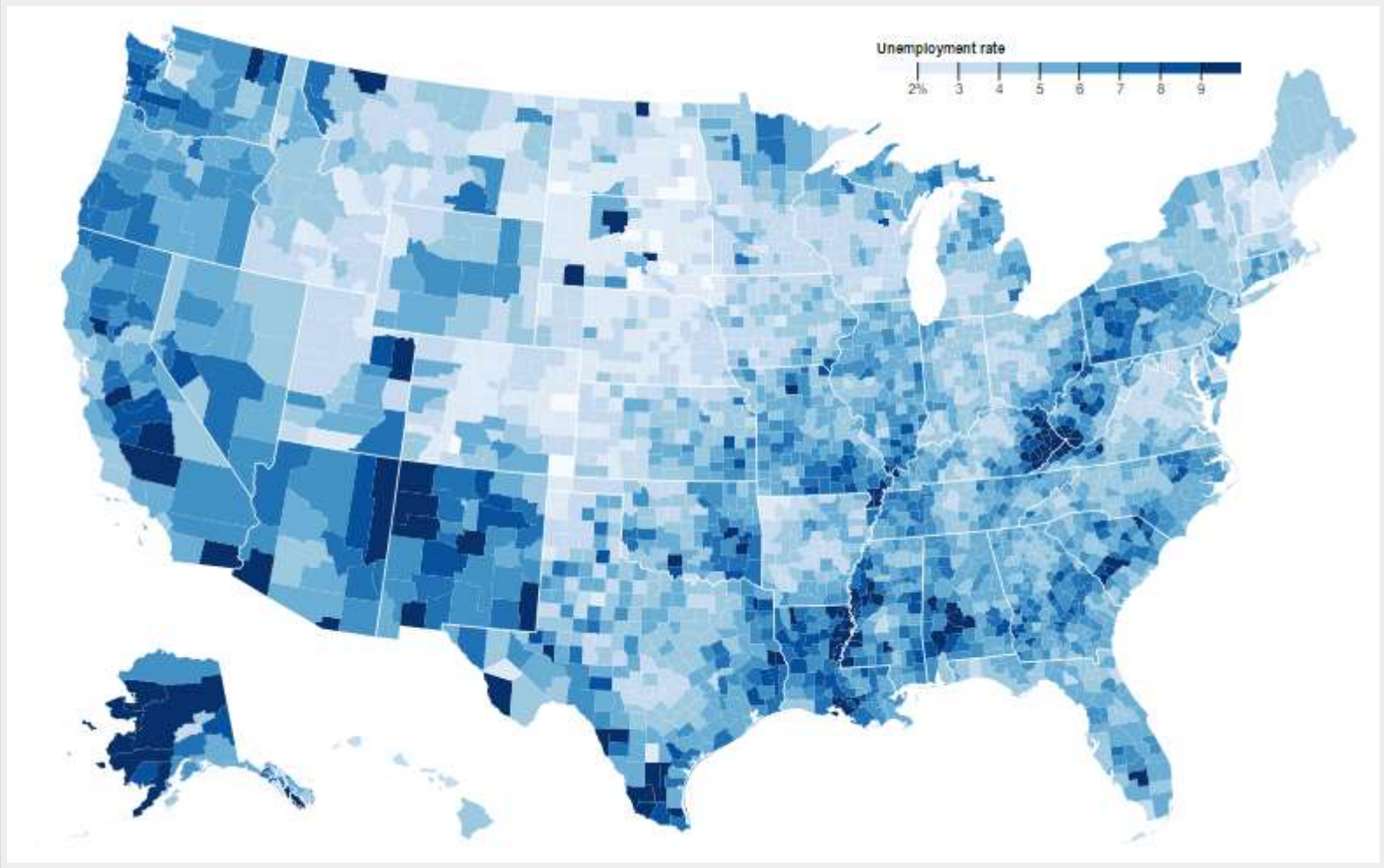
- Create interactive charts
- Visualize hierarchical data
- Leverage open source geo libraries
- Make super slick transitions
- Visualize relationship-base data
- Integrate with the newest JS standards and libraries

D3 in the wild

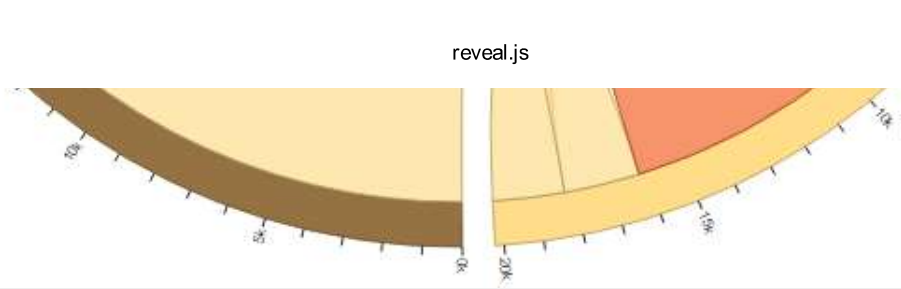
The most basic (source)



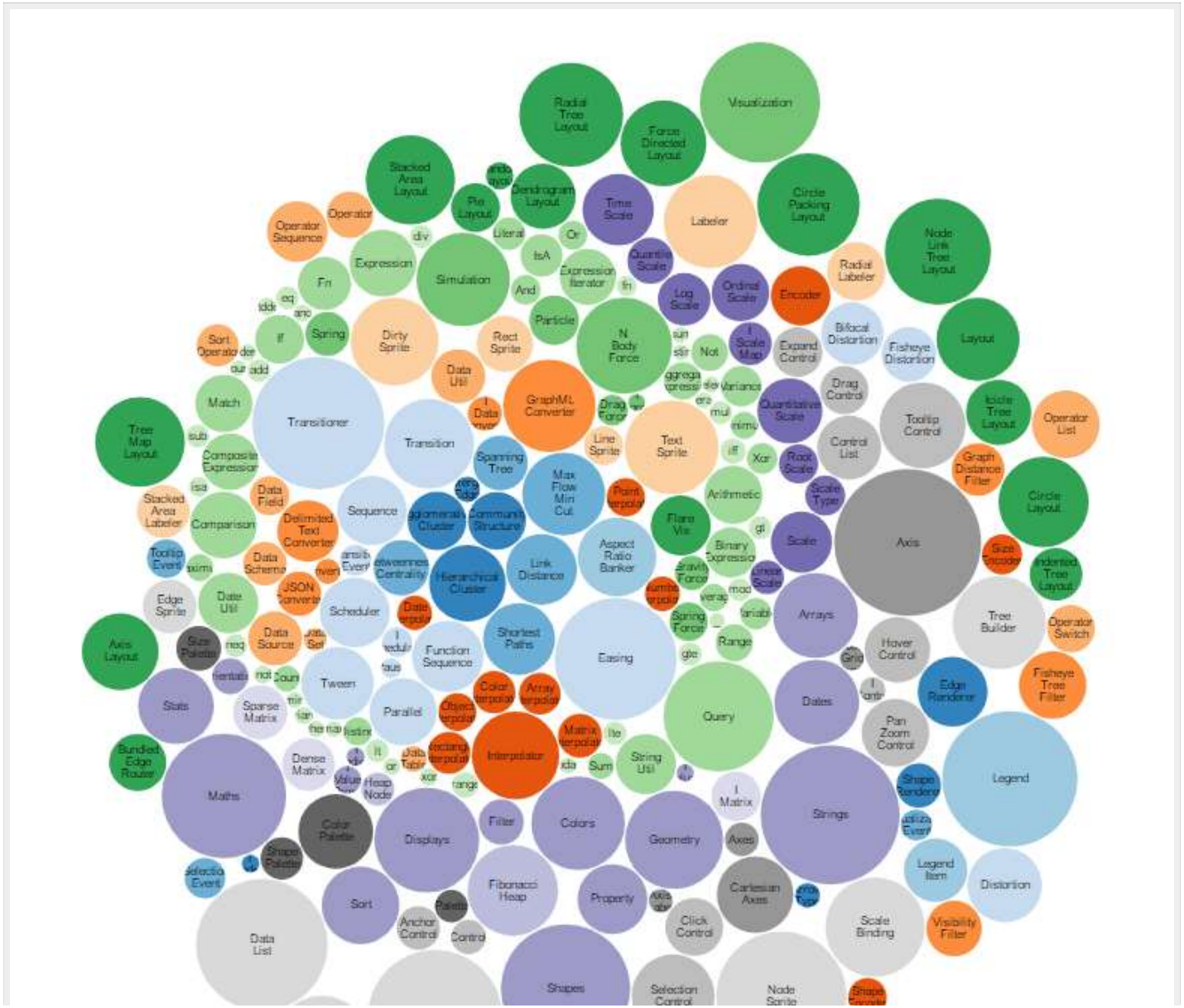
Maps (source)

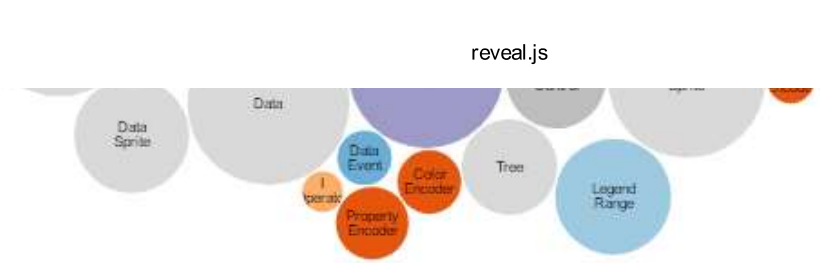


Chords (source)

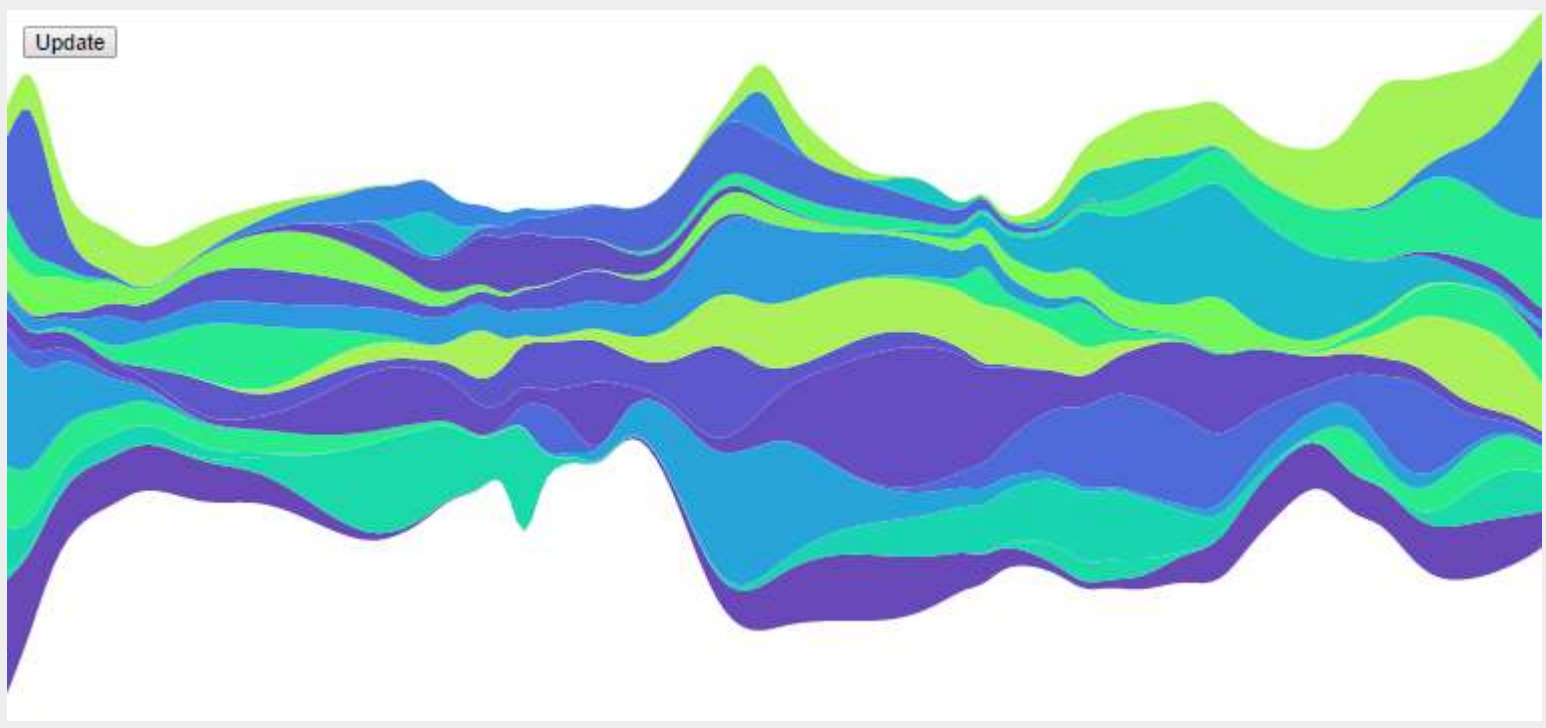

0k
5k
10k
15k
20k

Bubble (source)




Data
Data Sprite
Data Event
Color Encoder
Tree
Legend Range
Property Encoder

Stream (source)



Network (source)

Why are we talking about this at Qonnections?

Other than to show off appropriate gif/memes

Benefits of Using D3 with Qlik Sense

Specifically how does D3 integrate with Qlik Sense

A couple different reasons

Mashups

- Blend styles with the rest of your page
- Stand on the shoulders of other developers

Extensions

- Extend the capabilities of Qlik Sense
- Get an appreciation for bar charts

Ok Qlik Sense needs D3 but why does D3 benefit from Qlik Sense?

- Meta data
  - Min and Max
  - Glyph counts
  - Aggregations in D3 are a pain
  - Limits and methods on number of rows
  - Sorting and formatting

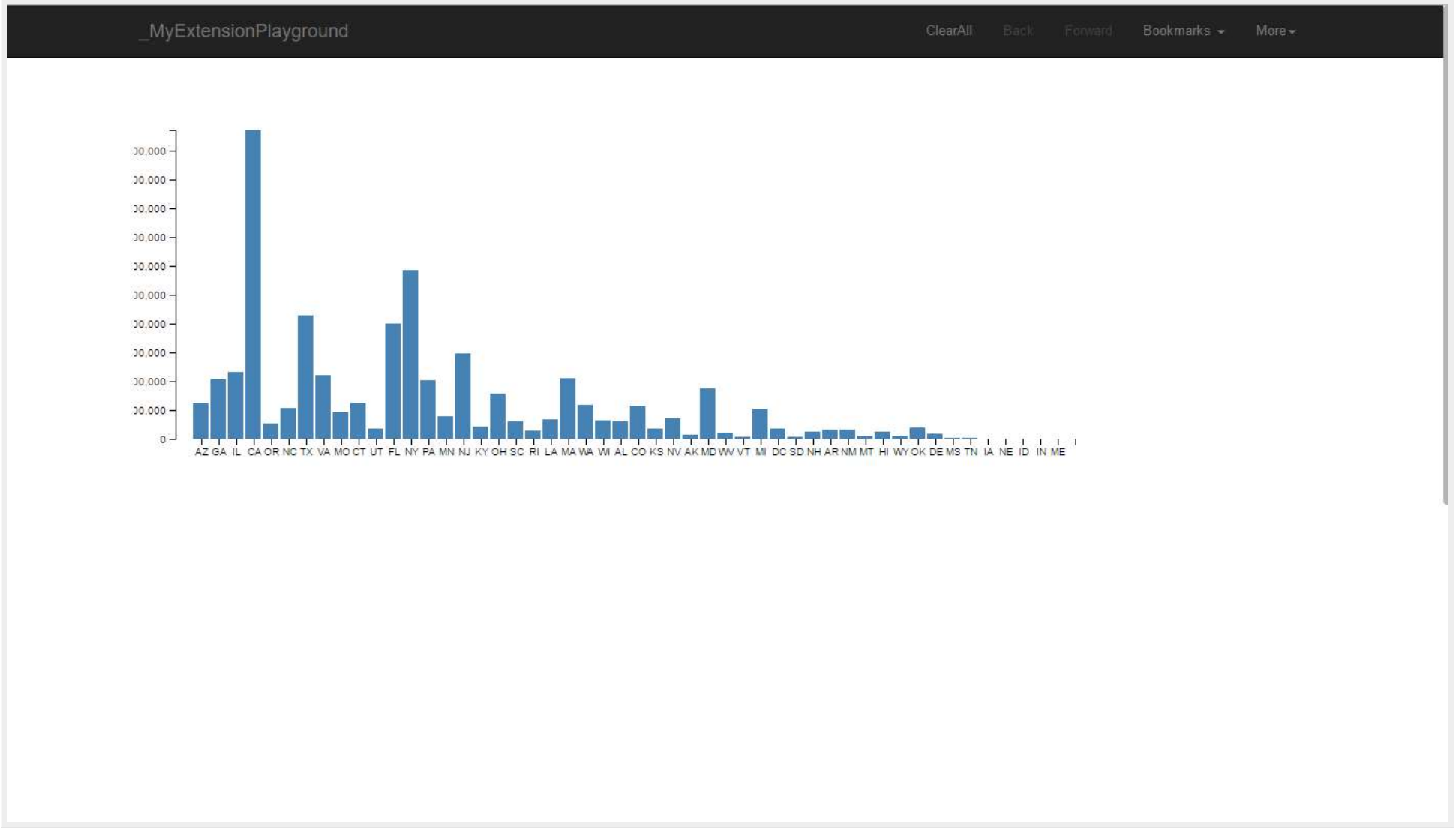
Branch blog post

Alright, let's integrate in Qlik Sense

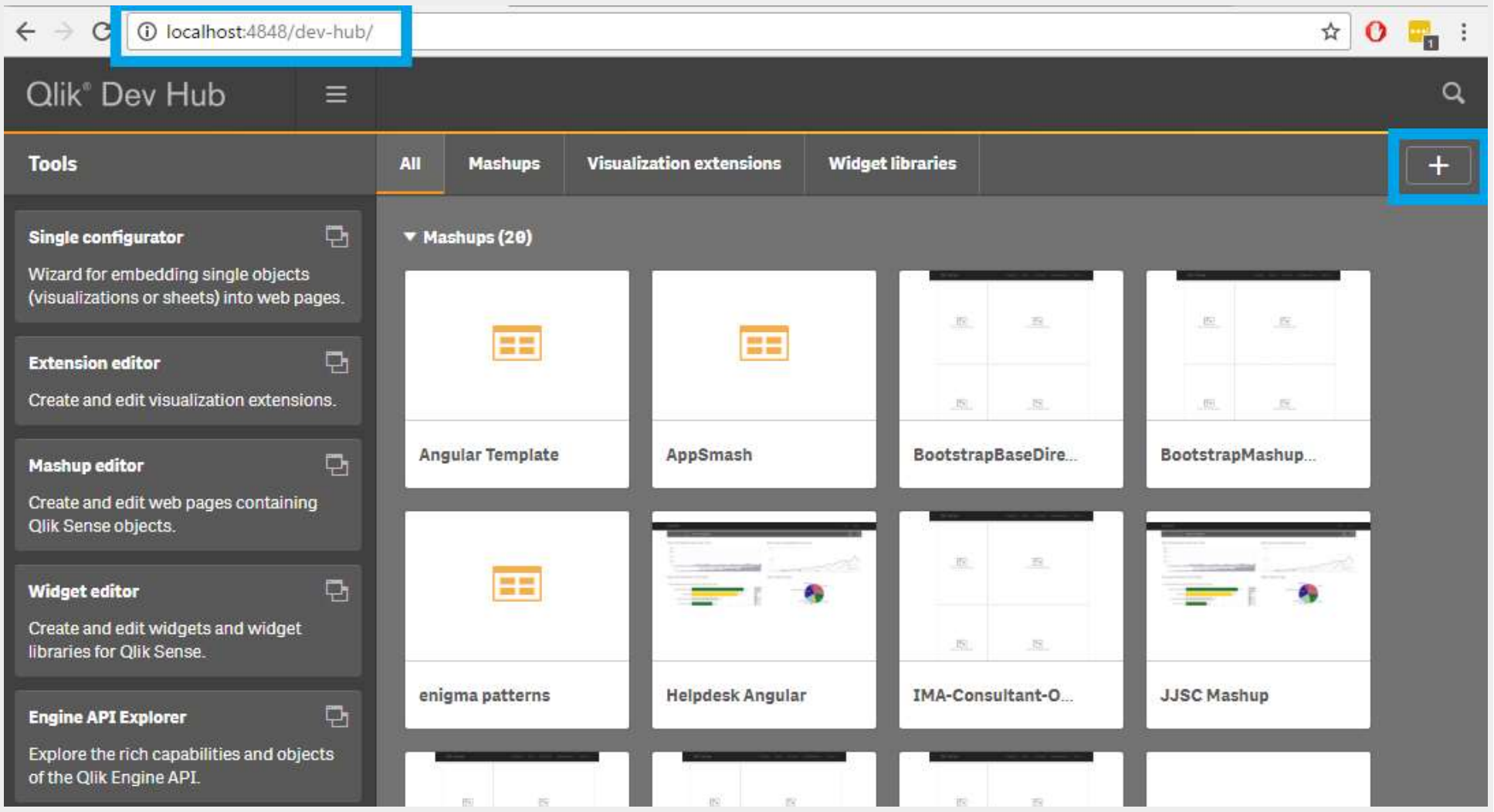
Using D3 in Mashups

or how to use the capabilities APIs with D3.js

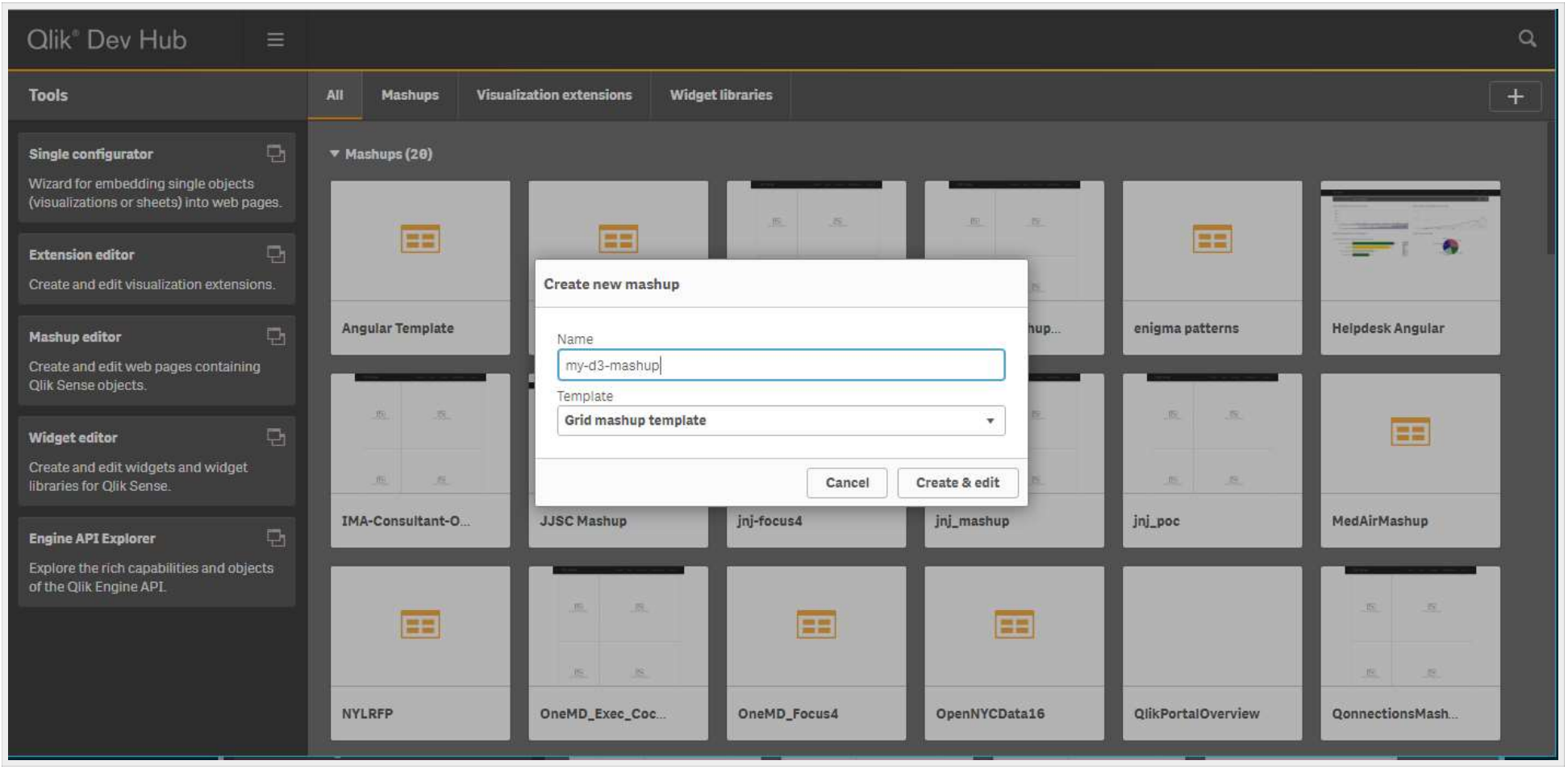
What are we building?



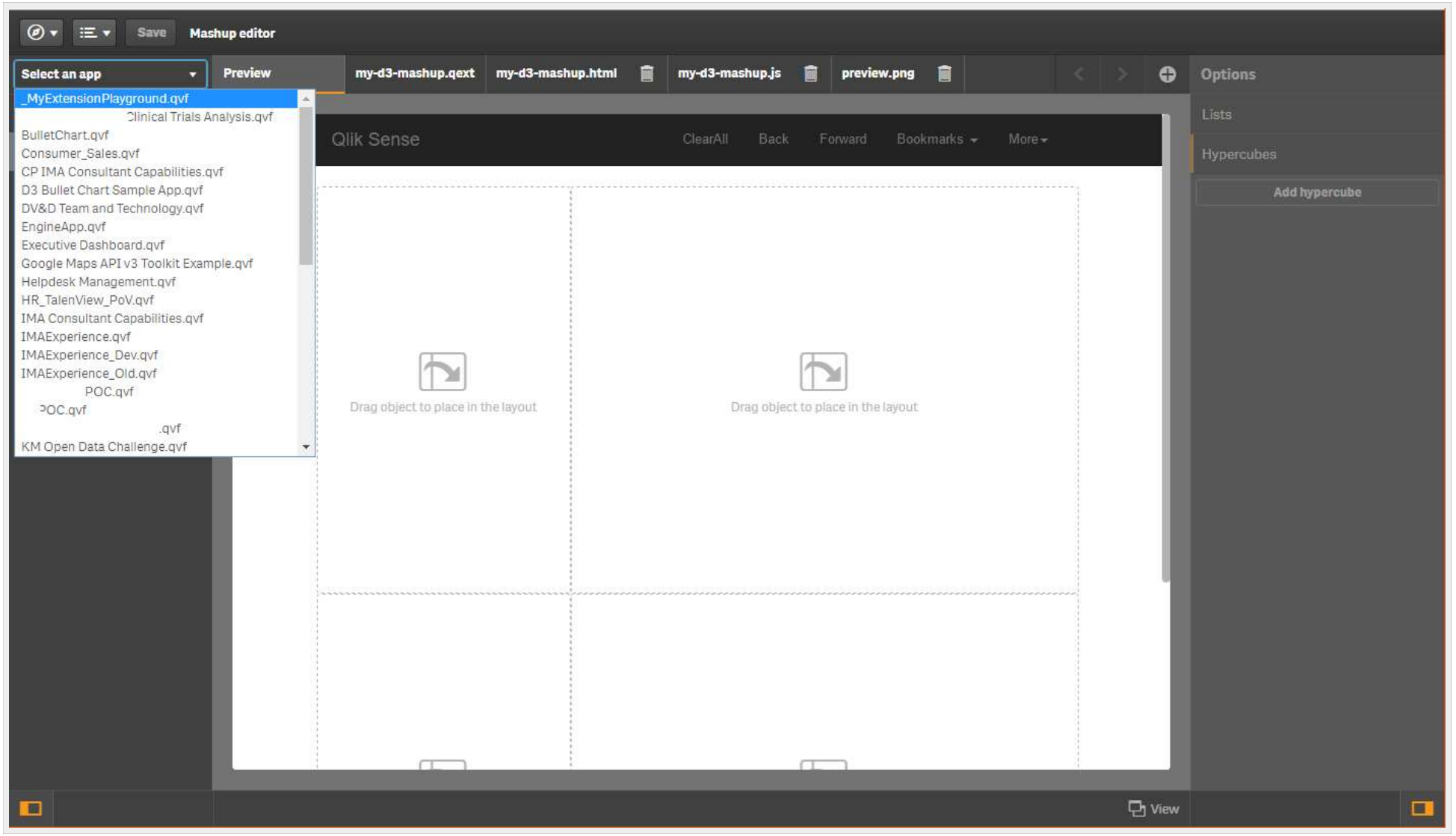
Create a new mashup



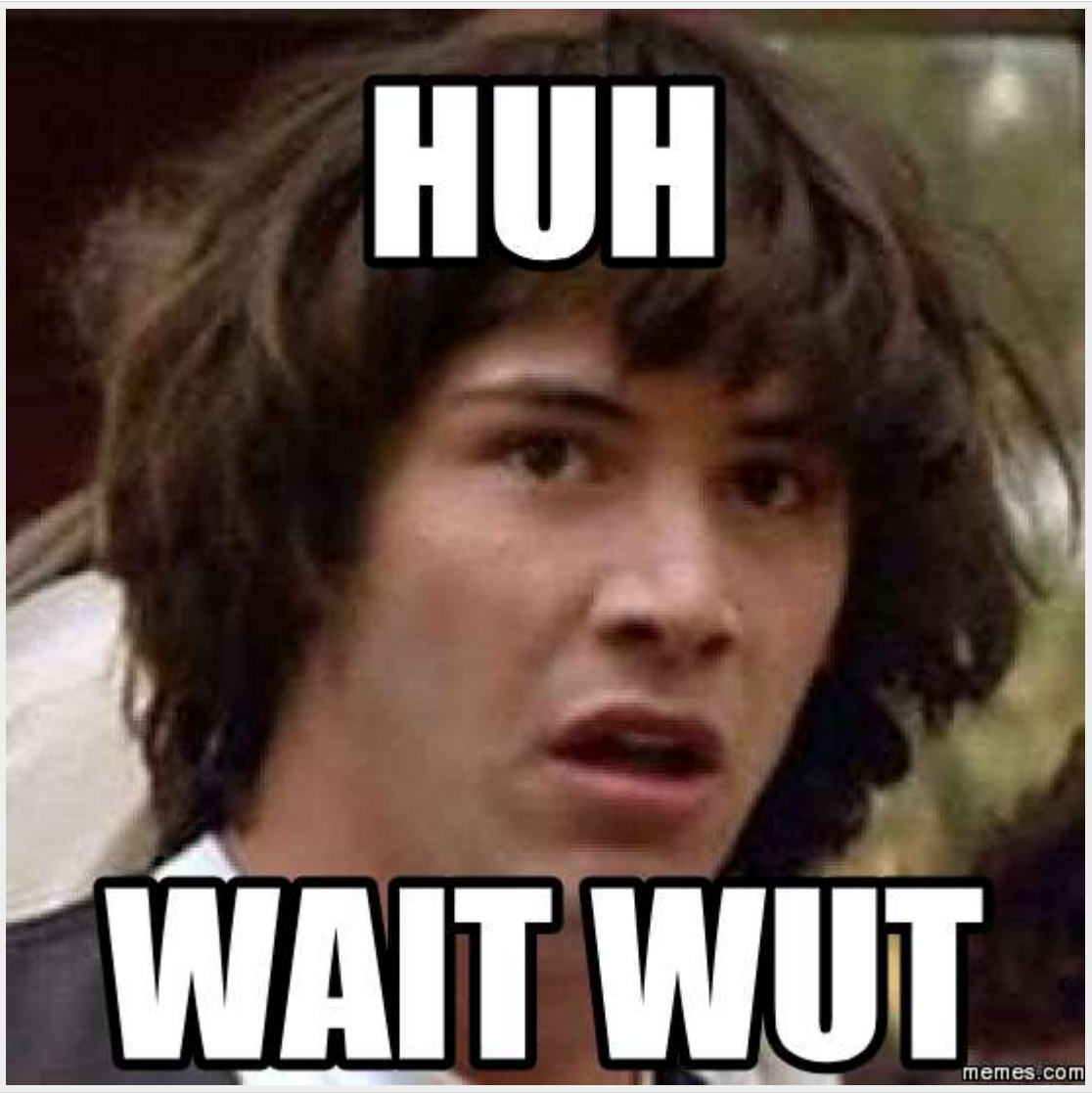
Create a new mashup



Connect to a Qlik Sense app



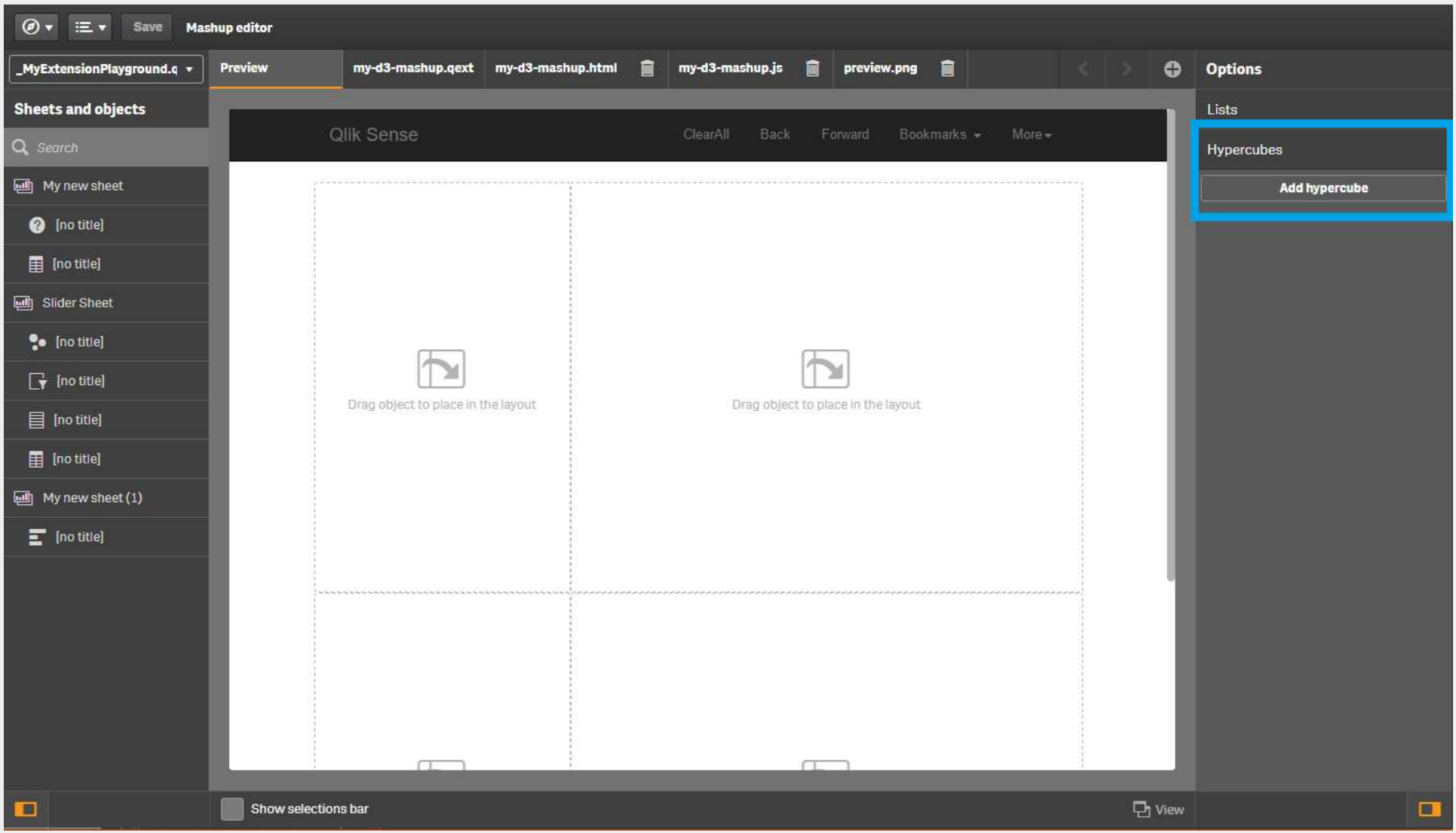
Add a hypercube



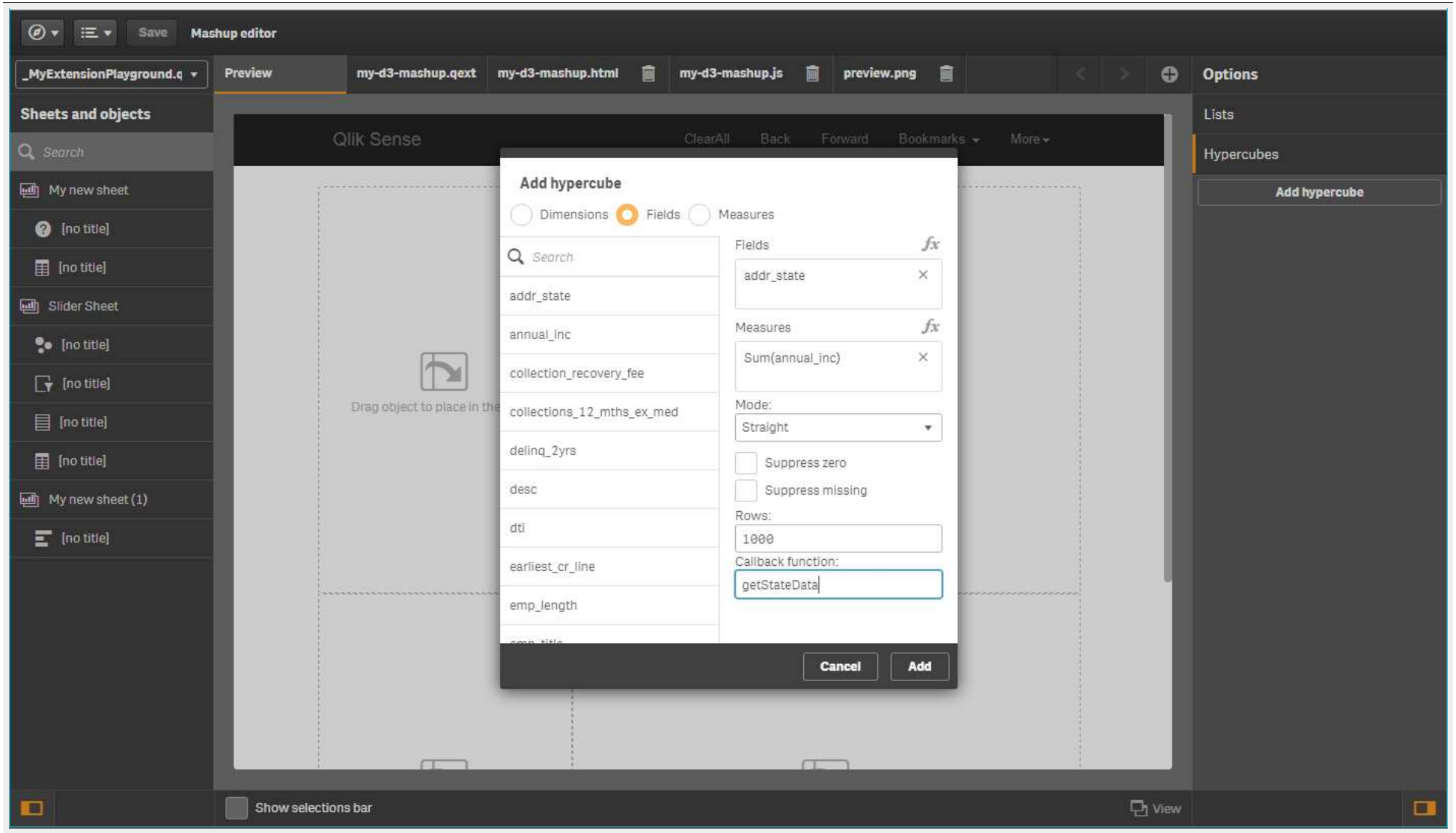
What in tarnation is a hypercube

- Define dimensions and measures
- Tell Qlik the max number of rows expected
- Create a callback function to handle the data returned
- User selects data, callback function is called again

So about that hypercube....

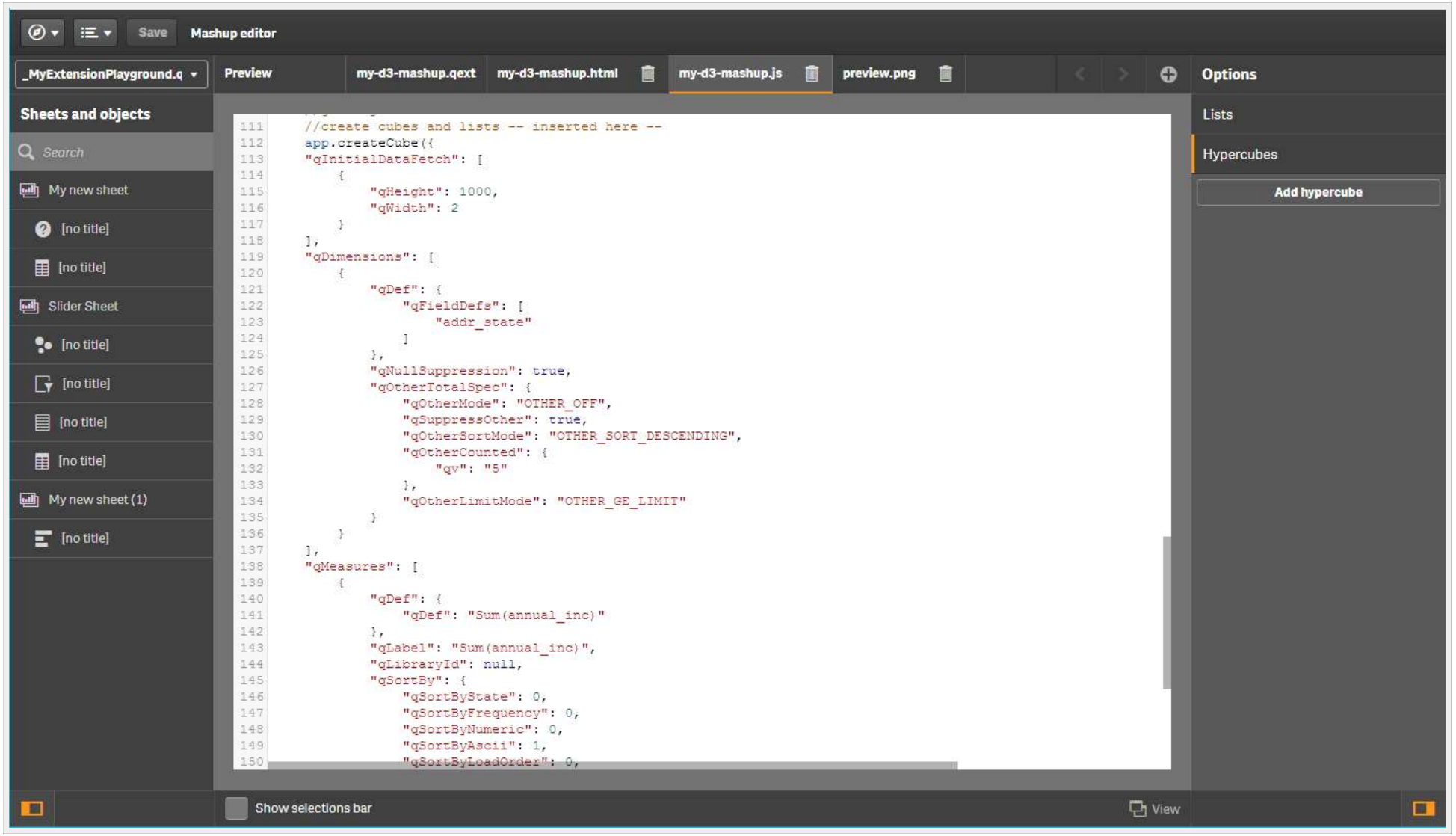


Configuring the hypercube

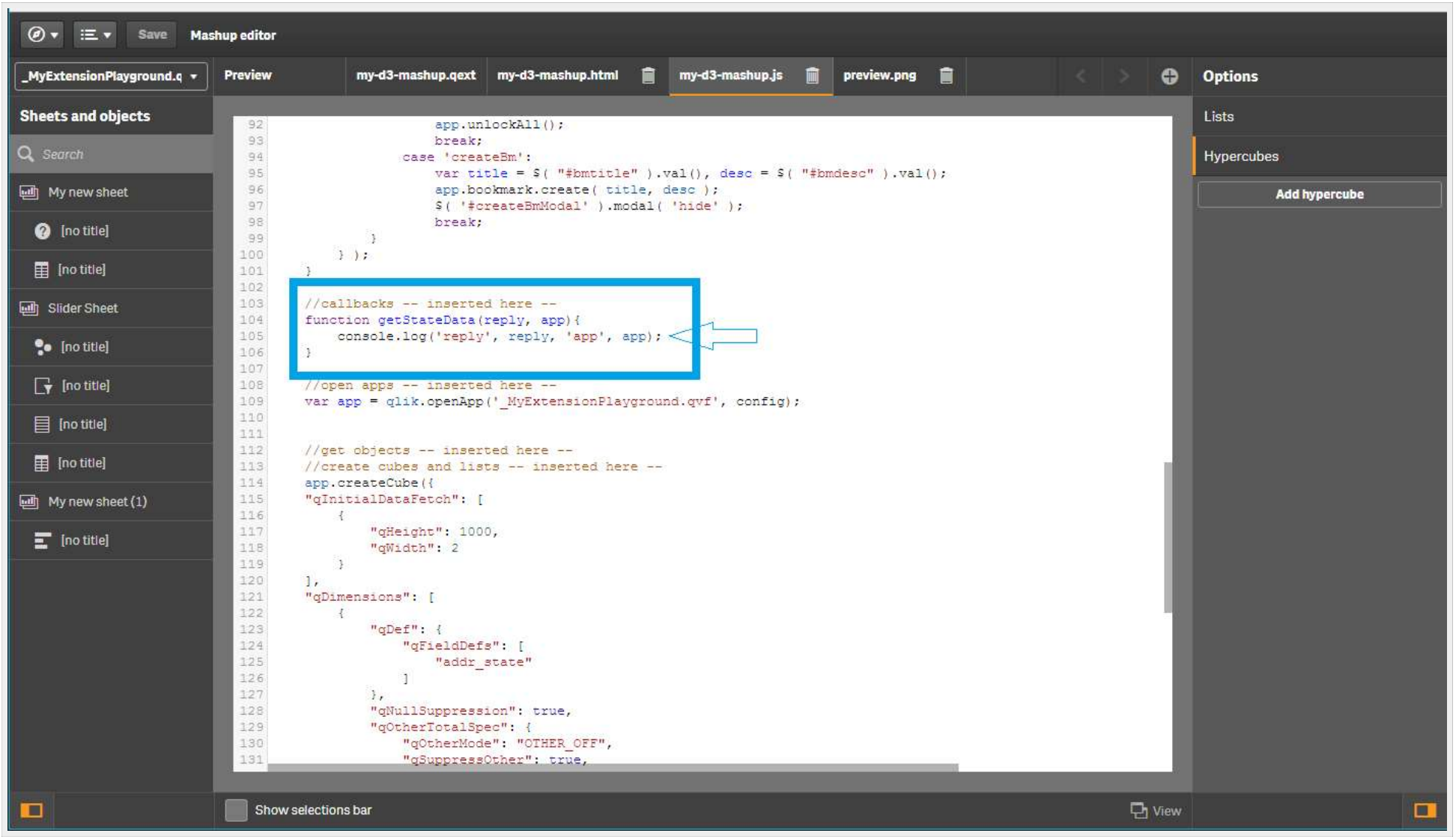


What did you do!?!

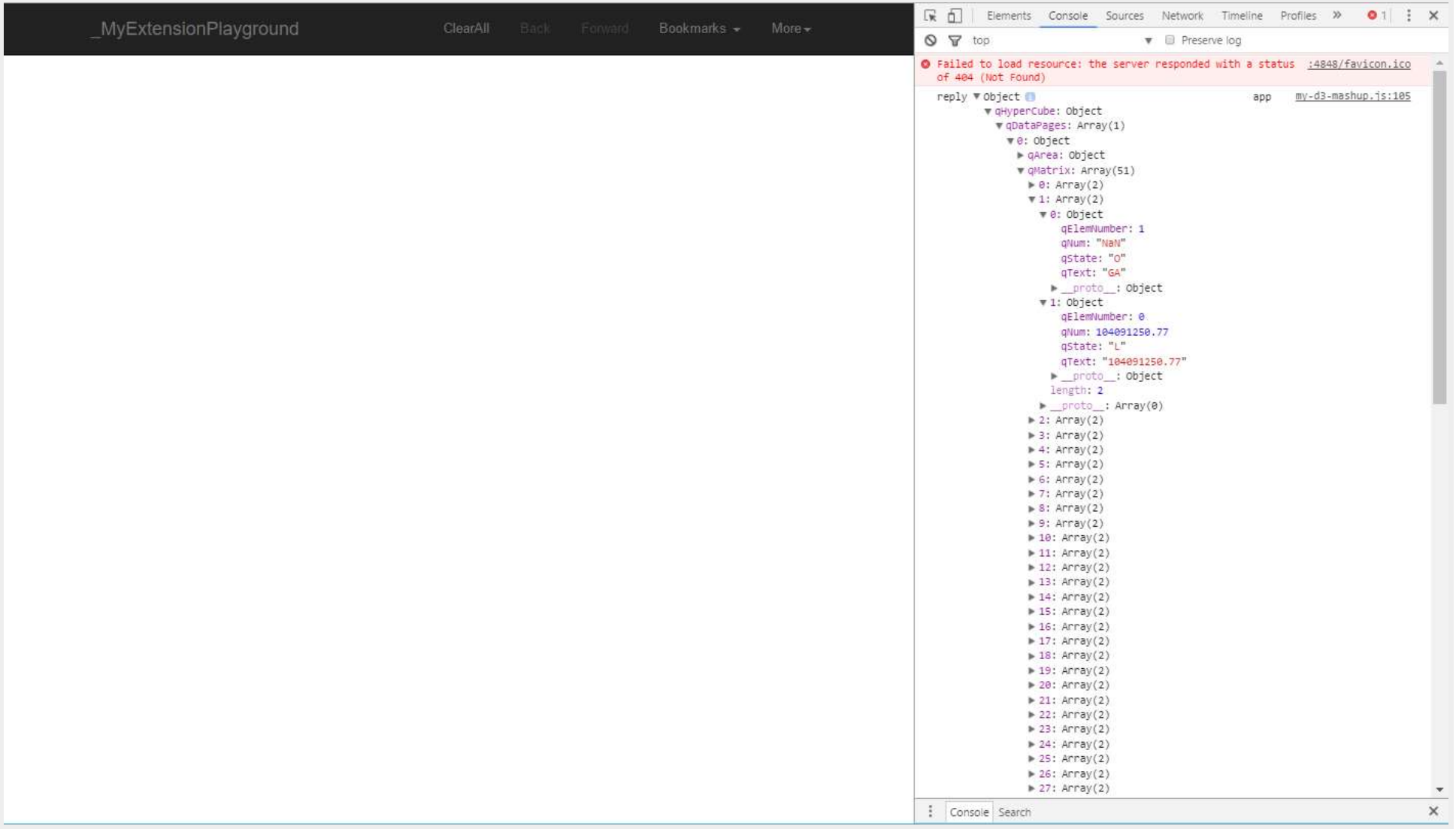
What did all that do?



Let's see the return values

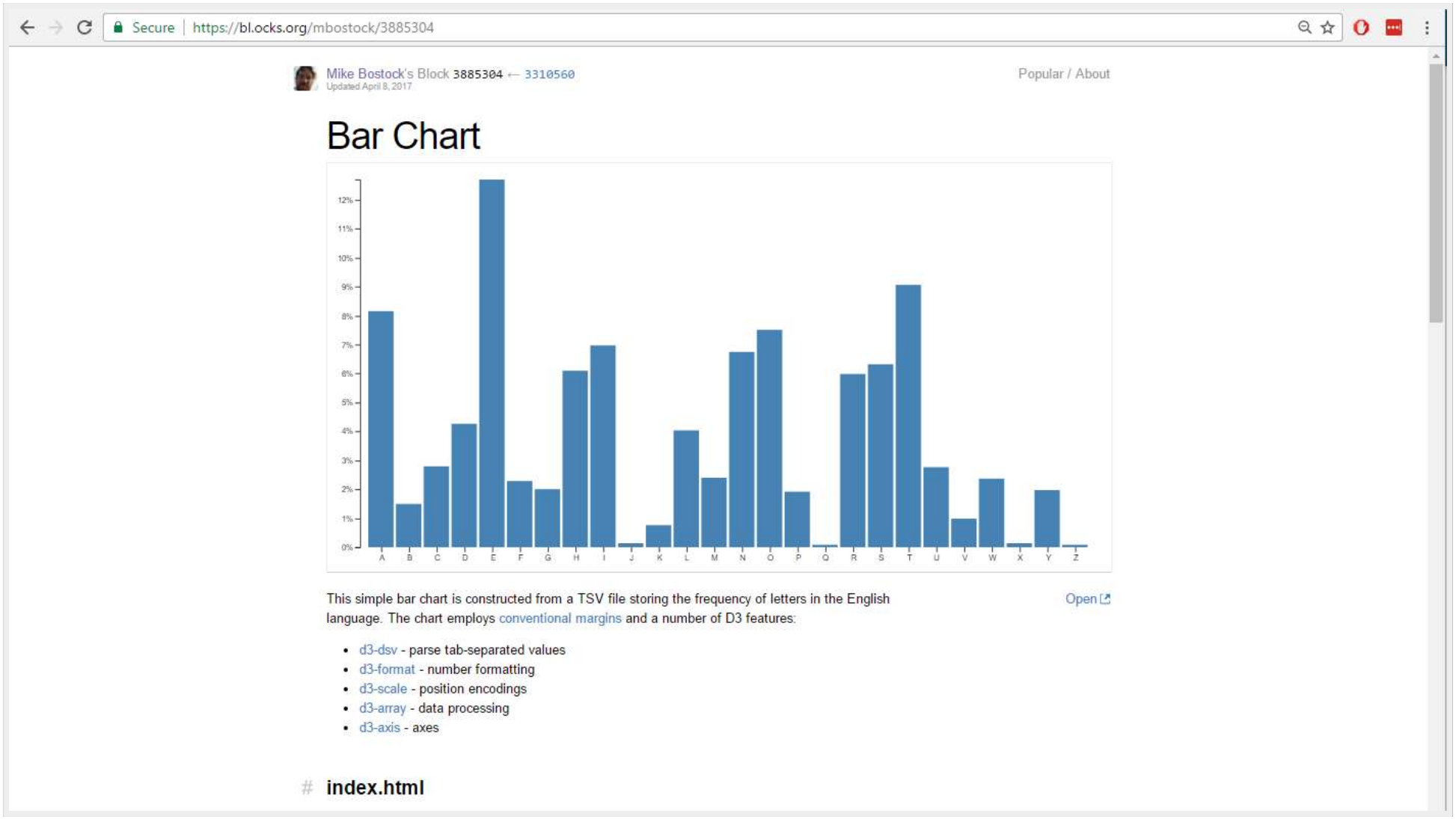


The developer panel

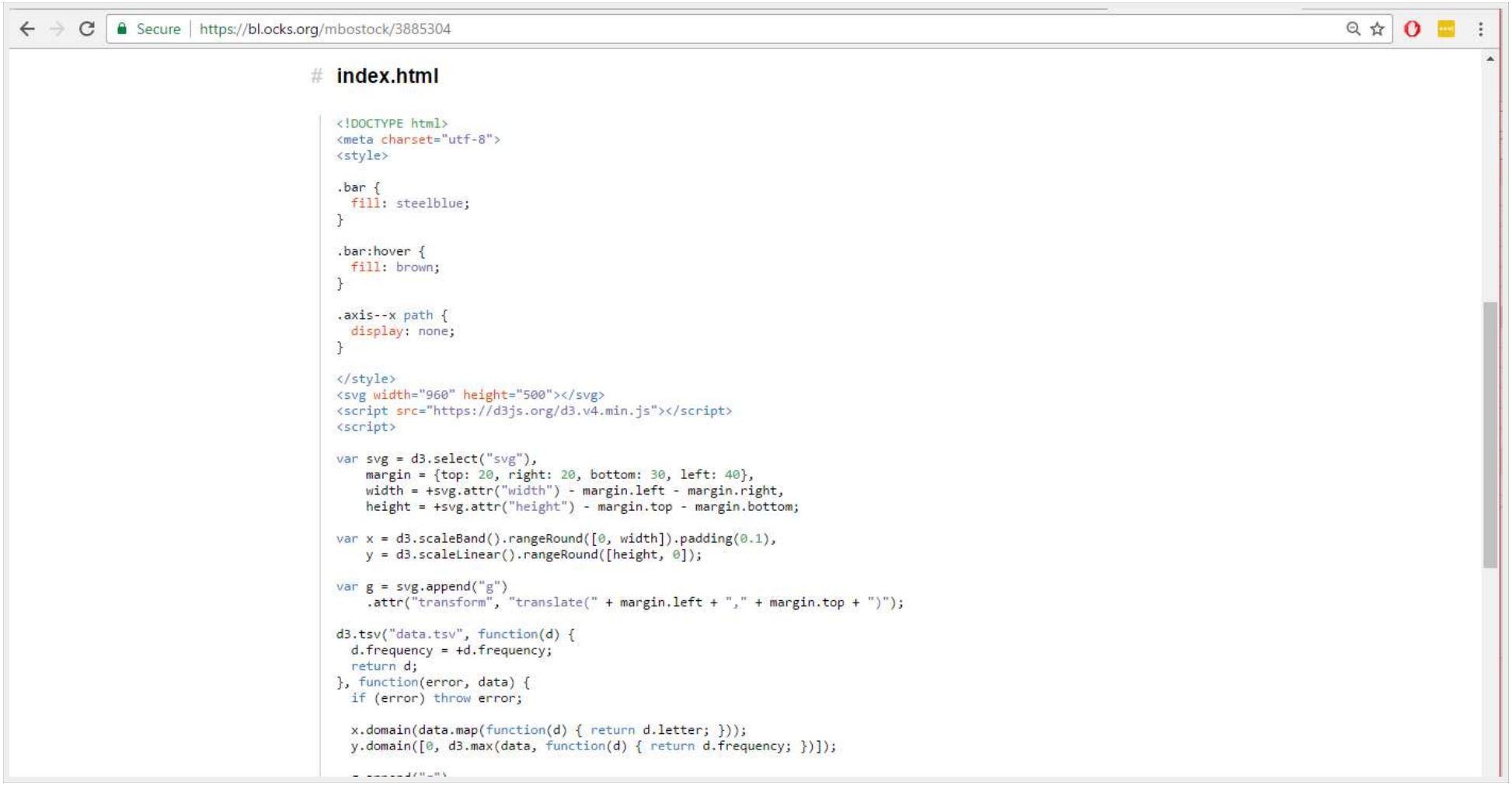



OH YEAH!!!!!!!
quickmeme.com

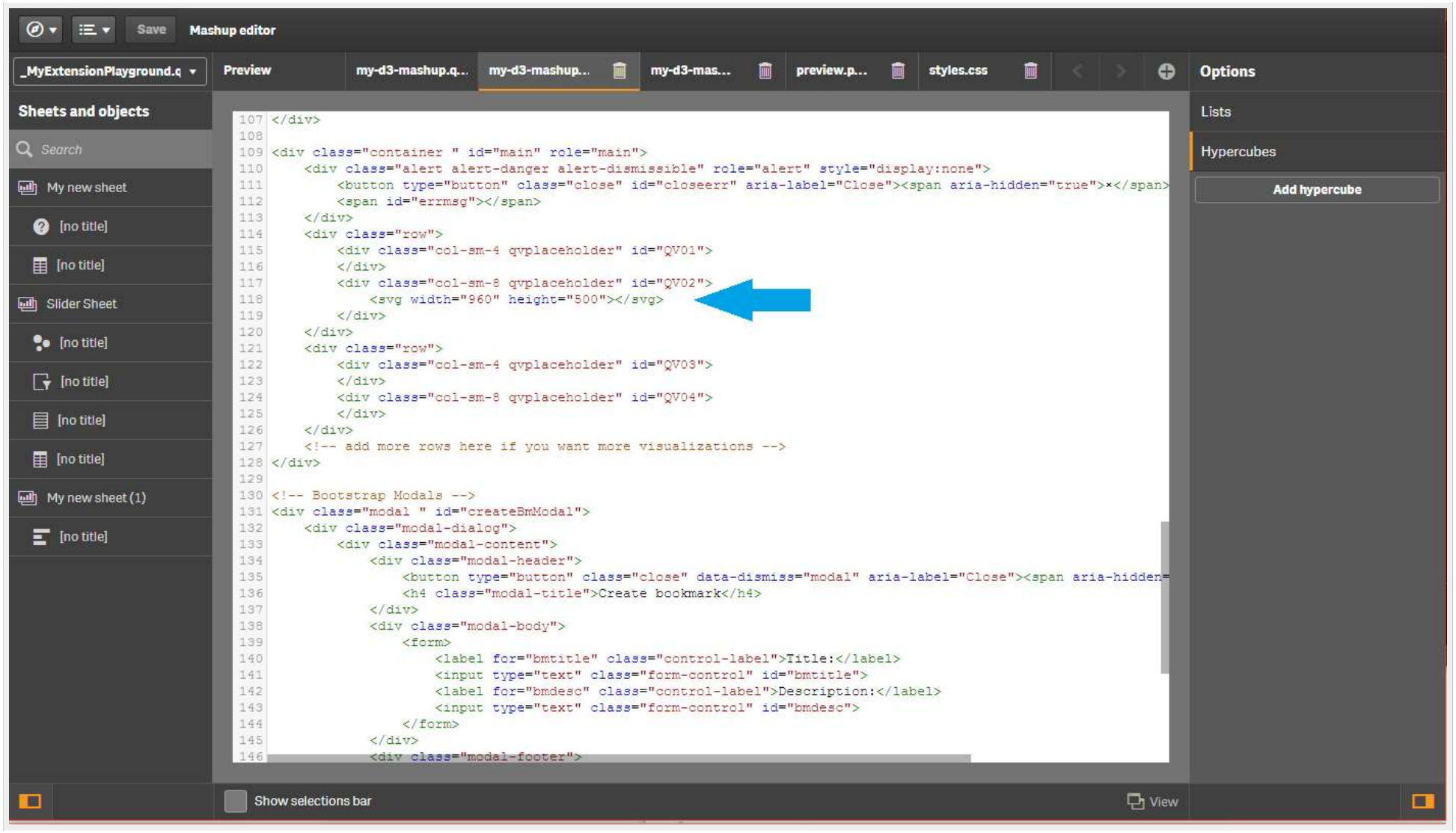
Go grab an example (source)



Get HTML/CSS/JS from example



Alter HTML


Save
Mashup editor
_MyExtensionPlayground.q
Preview
my-d3-mashup.q...
my-d3-mashup...
my-d3-mas...
preview.p...
styles.css
Options
Sheets and objects
Search
My new sheet
[no title]
[no title]
Slider Sheet
[no title]
[no title]
[no title]
[no title]
My new sheet (1)
[no title]
Lists
Hypercubes
Add hypercube

<button type="button" class="close" id="closeerr" aria-label="Close">×

<svg width="960" height="500"></svg>

<!-- add more rows here if you want more visualizations -->

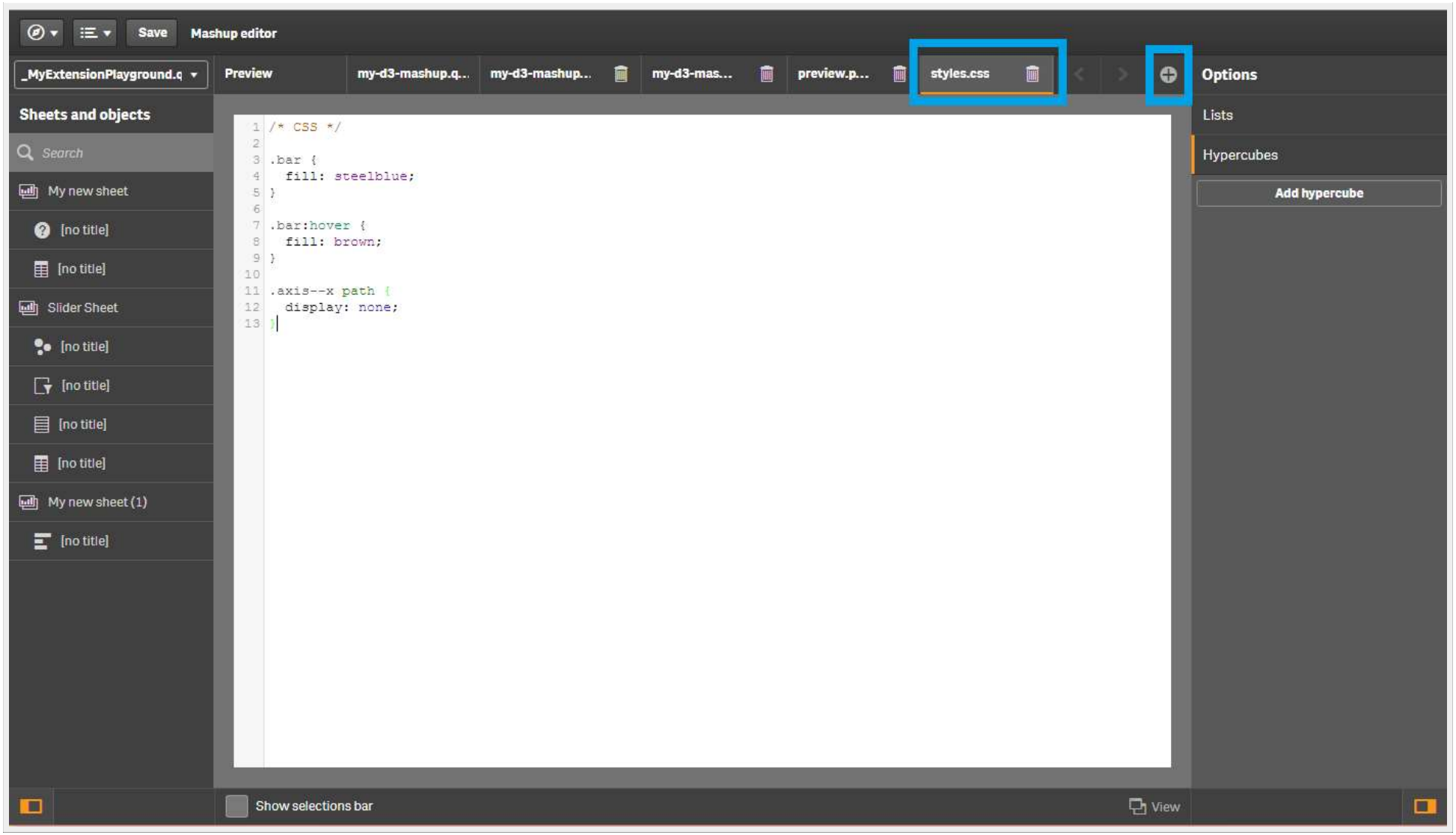
<!-- Bootstrap Modals -->

<button type="button" class="close" data-dismiss="modal" aria-label="Close"><span aria-hidden=
<h4 class="modal-title">Create bookmark</h4>

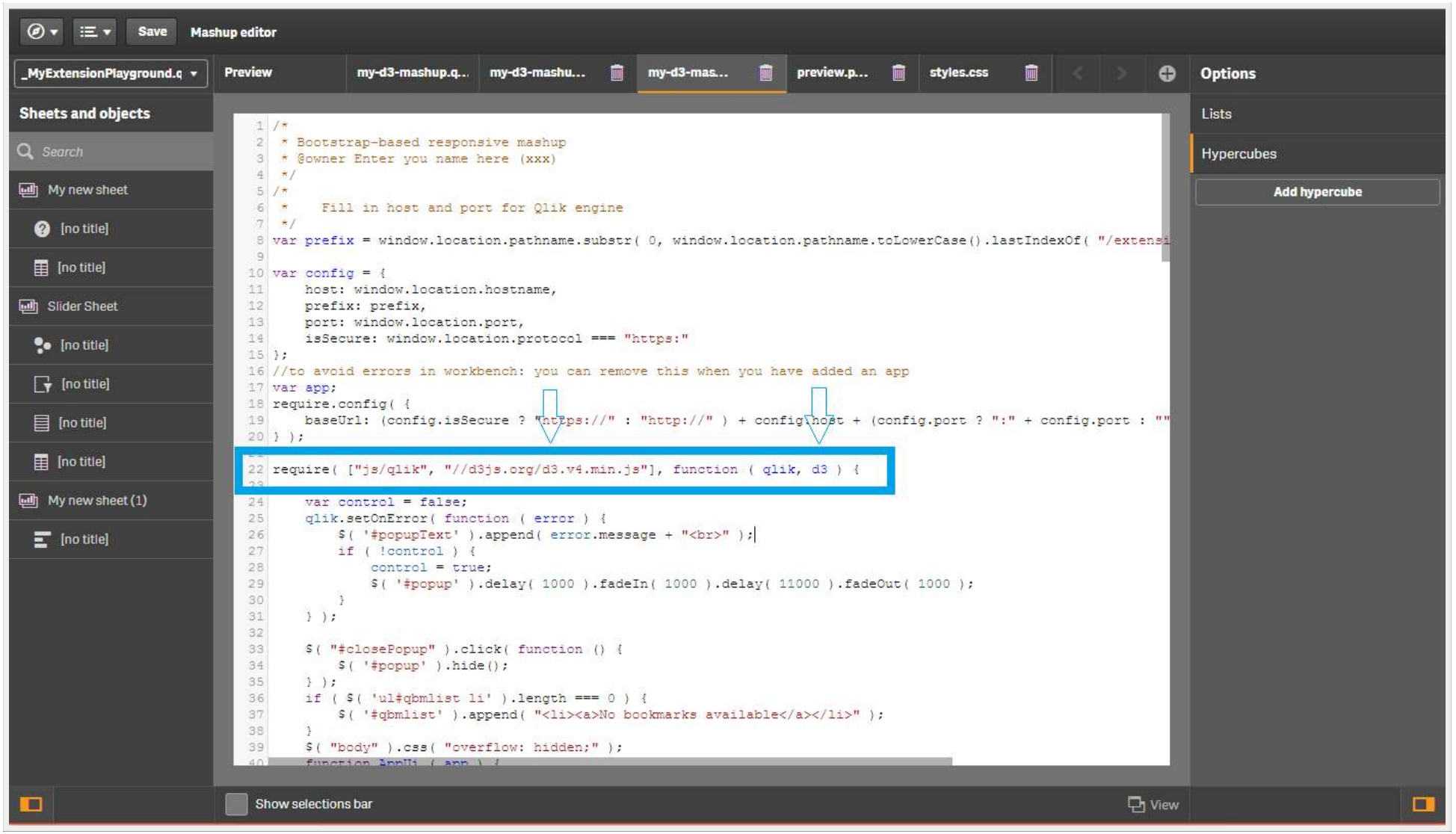
<form>
<label for="bmtitle" class="control-label">Title:</label>
<input type="text" class="form-control" id="bmtitle">
<label for="bmdesc" class="control-label">Description:</label>
<input type="text" class="form-control" id="bmdesc">
</form>

Show selections bar
View

Insert css into script



Load D3 library in

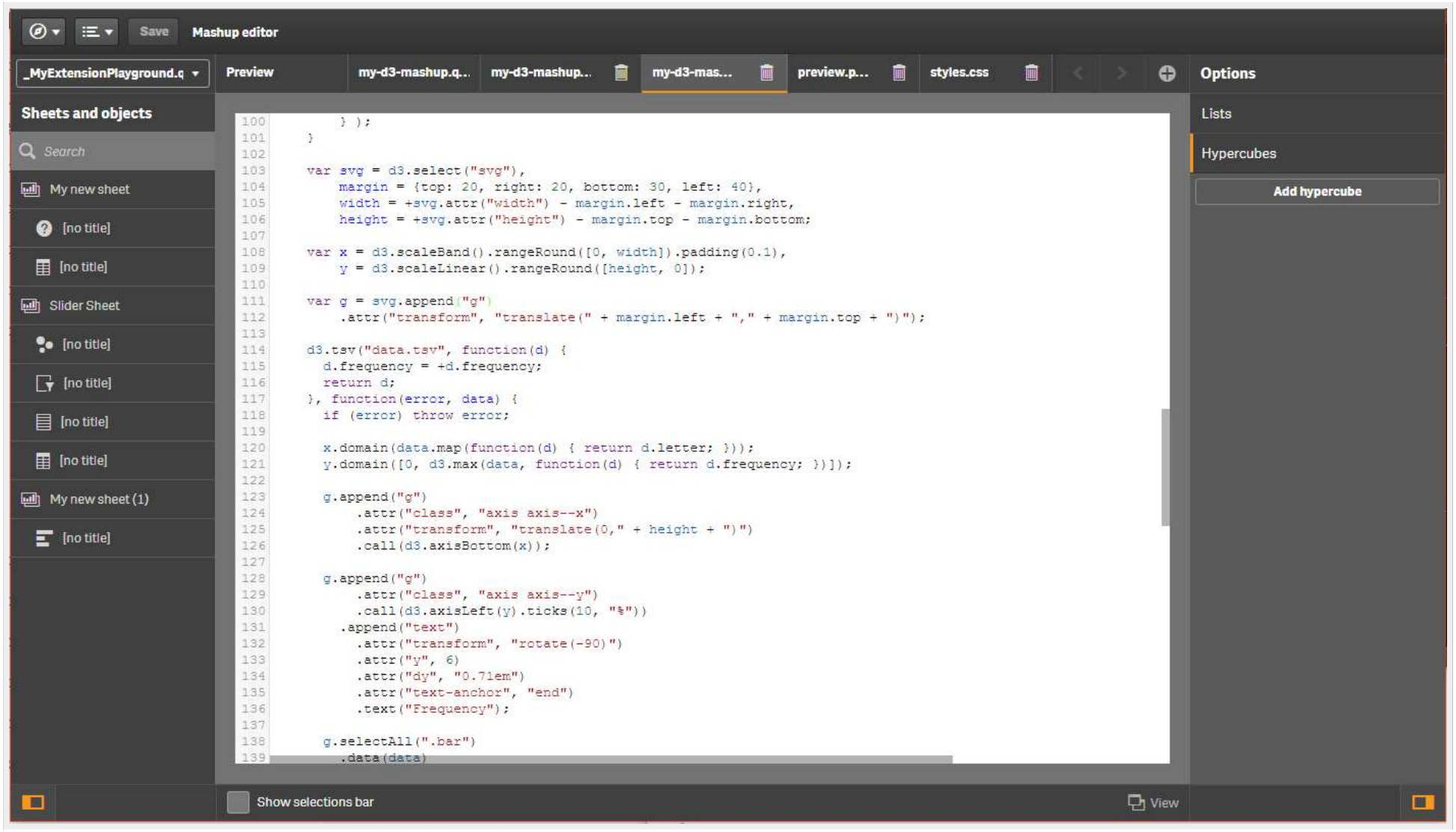


Get D3 from example

Secure | https://bl.ocks.org/mbostock/3885304

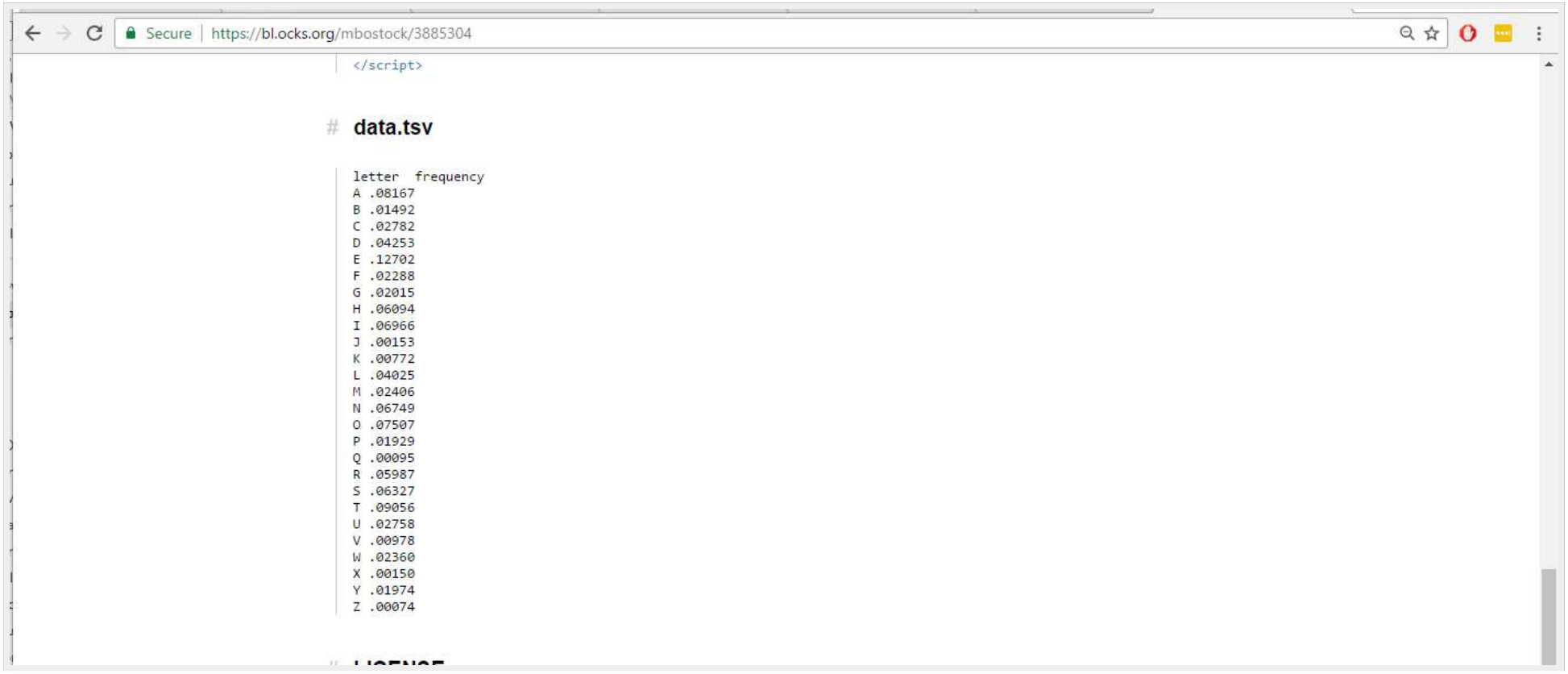
```
<script>

var svg = d3.select("svg"),
    margin = {top: 20, right: 20, bottom: 30, left: 40},
    width = +svg.attr("width") - margin.left - margin.right,
    height = +svg.attr("height") - margin.top - margin.bottom;

var x = d3.scaleBand().rangeRound([0, width]).padding(0.1),
    y = d3.scaleLinear().rangeRound([height, 0]);

var g = svg.append("g")
    .attr("transform", "translate(" + margin.left + "," + margin.top + ")");

d3.tsv("data.tsv", function(d) {
  d.frequency = +d.frequency;
  return d;
}, function(error, data) {
  if (error) throw error;

  x.domain(data.map(function(d) { return d.letter; }));
  y.domain([0, d3.max(data, function(d) { return d.frequency; })]);

  g.append("g")
      .attr("class", "axis axis--x")
      .attr("transform", "translate(0," + height + ")")
      .call(d3.axisBottom(x));

  g.append("g")
      .attr("class", "axis axis--y")
      .call(d3.axisLeft(y).ticks(10, "%"))
    .append("text")
      .attr("transform", "rotate(-90)")
      .attr("y", 6)
      .attr("dy", "0.71em")
      .attr("text-anchor", "end")
      .text("Frequency");

  g.selectAll(".bar")
    .data(data)
    .enter().append("rect")
      .attr("class", "bar")
      .attr("x", function(d) { return x(d.letter); })
      .attr("y", function(d) { return y(d.frequency); })
      .attr("width", x.bandwidth())
      .attr("height", function(d) { return height - y(d.frequency); });
});

</script>
```

Copy D3 code into script

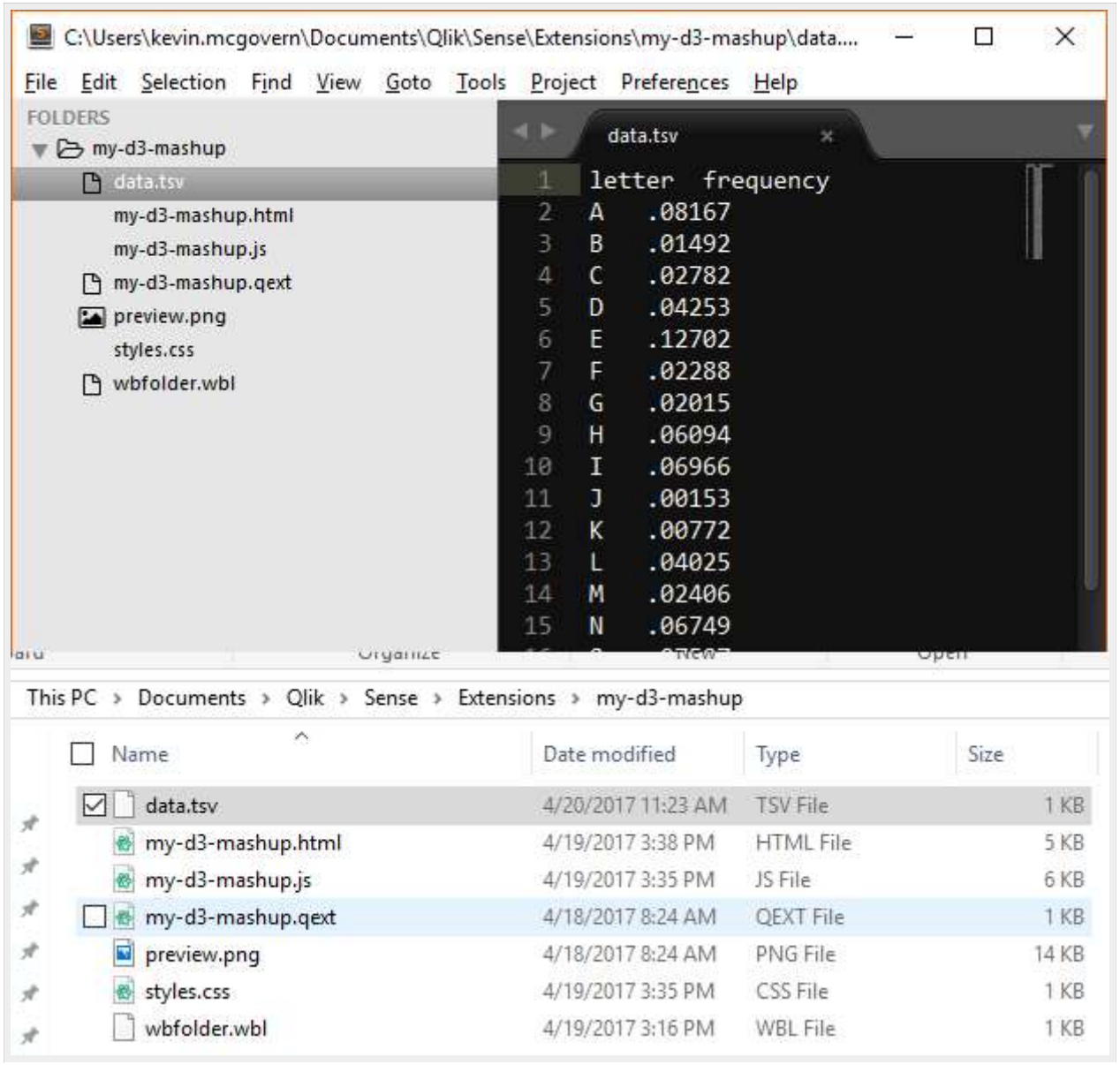


Before we connect to hypercube...

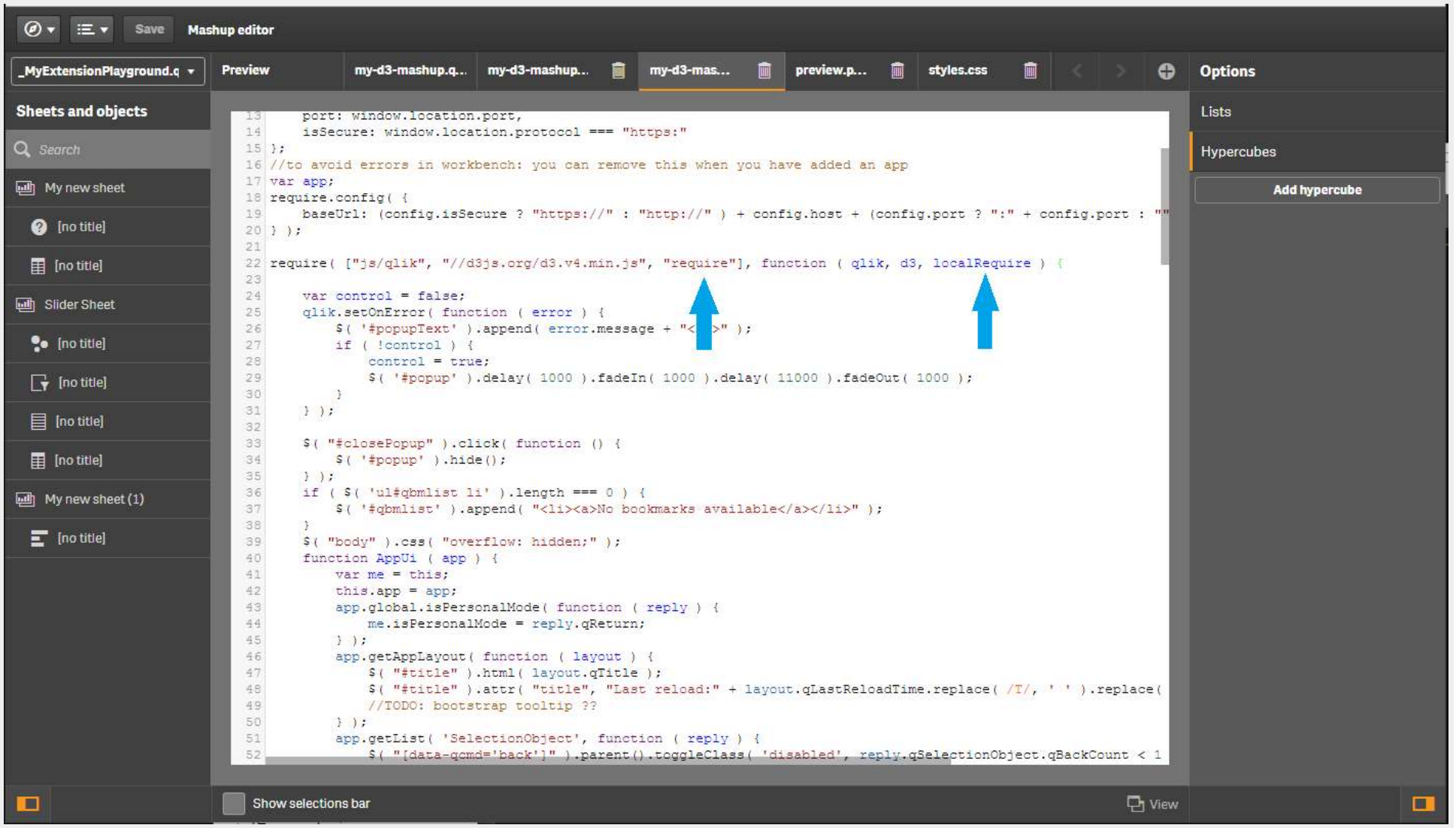
let's get the sample data



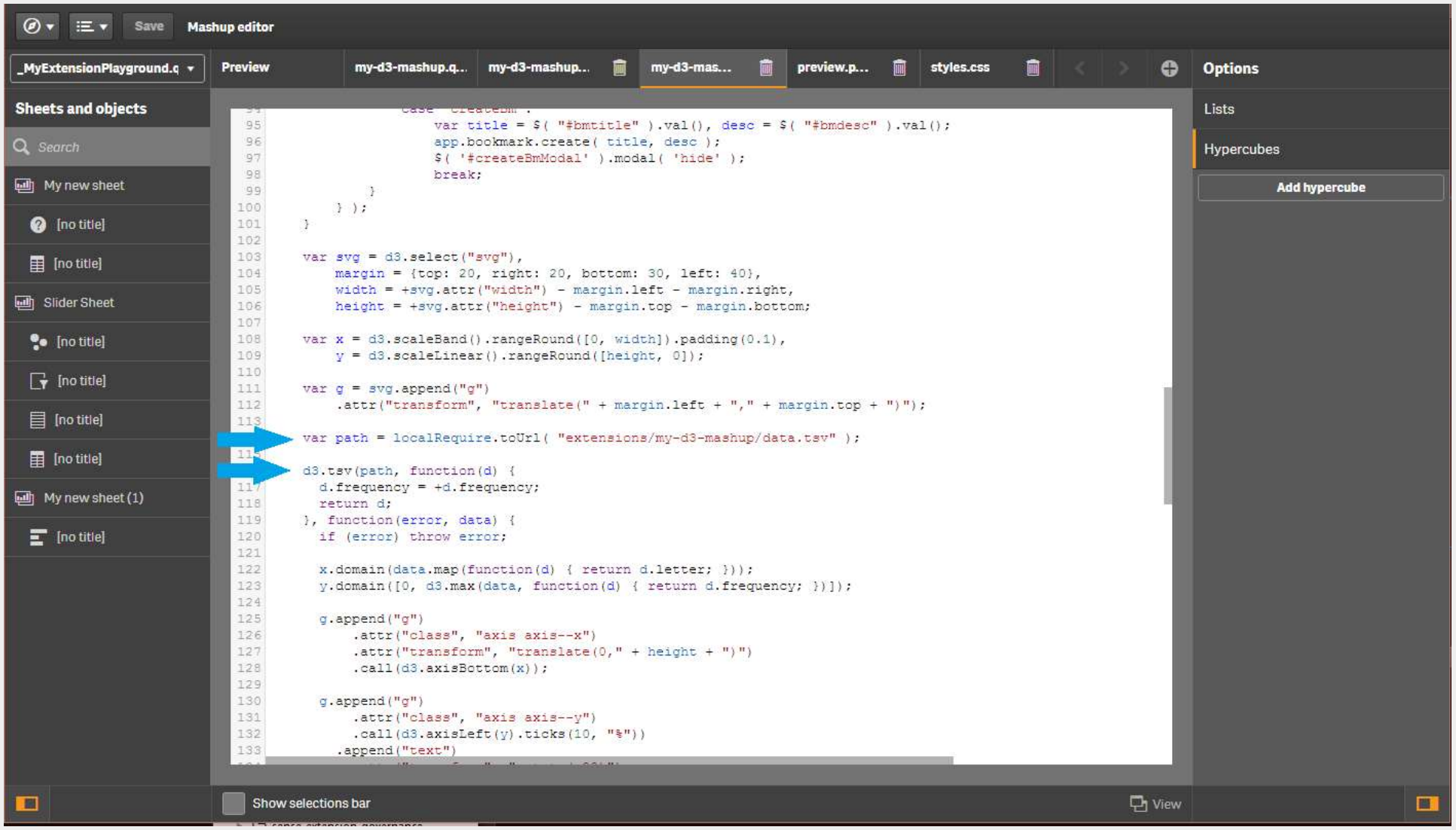
Drop the data file in the directory


C:\Users\kevin.mcgovern\Documents\Qlik\Sense\Extensions\my-d3-mashup\data....
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
my-d3-mashup
data.tsv
my-d3-mashup.html
my-d3-mashup.js
my-d3-mashup.qext
preview.png
styles.css
wbfolder.wbl
data.tsv
letter frequency
A .08167
B .01492
C .02782
D .04253
E .12702
F .02288
G .02015
H .06094
I .06966
J .00153
K .00772
L .04025
M .02406
N .06749
This PC › Documents › Qlik › Sense › Extensions › my-d3-mashup
Name
Date modified
Type
Size
data.tsv 4/20/2017 11:23 AM TSV File 1 KB
my-d3-mashup.html 4/19/2017 3:38 PM HTML File 5 KB
my-d3-mashup.js 4/19/2017 3:35 PM JS File 6 KB
my-d3-mashup.qext 4/18/2017 8:24 AM QEXT File 1 KB
preview.png 4/18/2017 8:24 AM PNG File 14 KB
styles.css 4/19/2017 3:35 PM CSS File 1 KB
wbfolder.wbl 4/19/2017 3:16 PM WBL File 1 KB

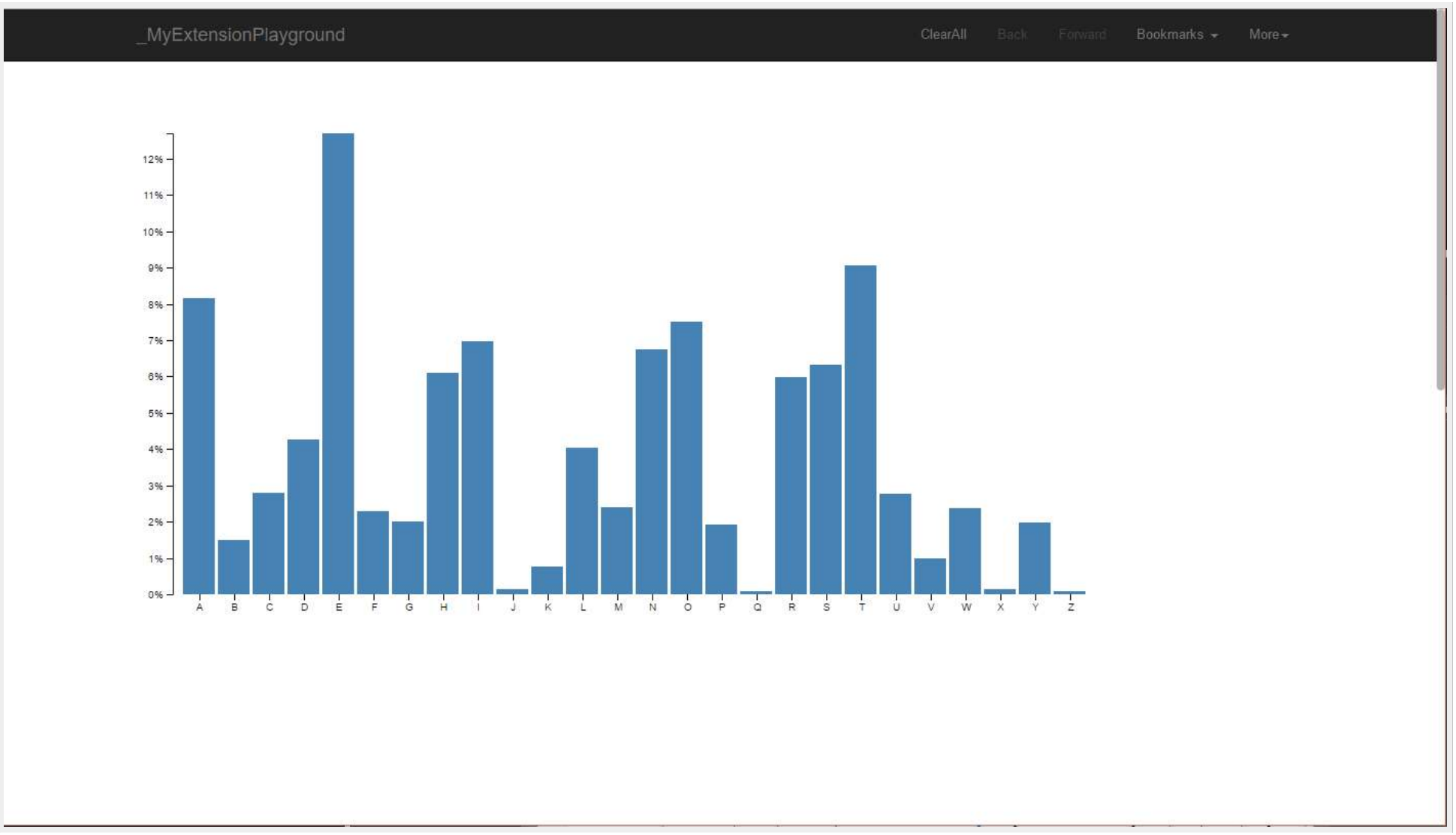
Load an instance of require.js



Reference the tsv file in js



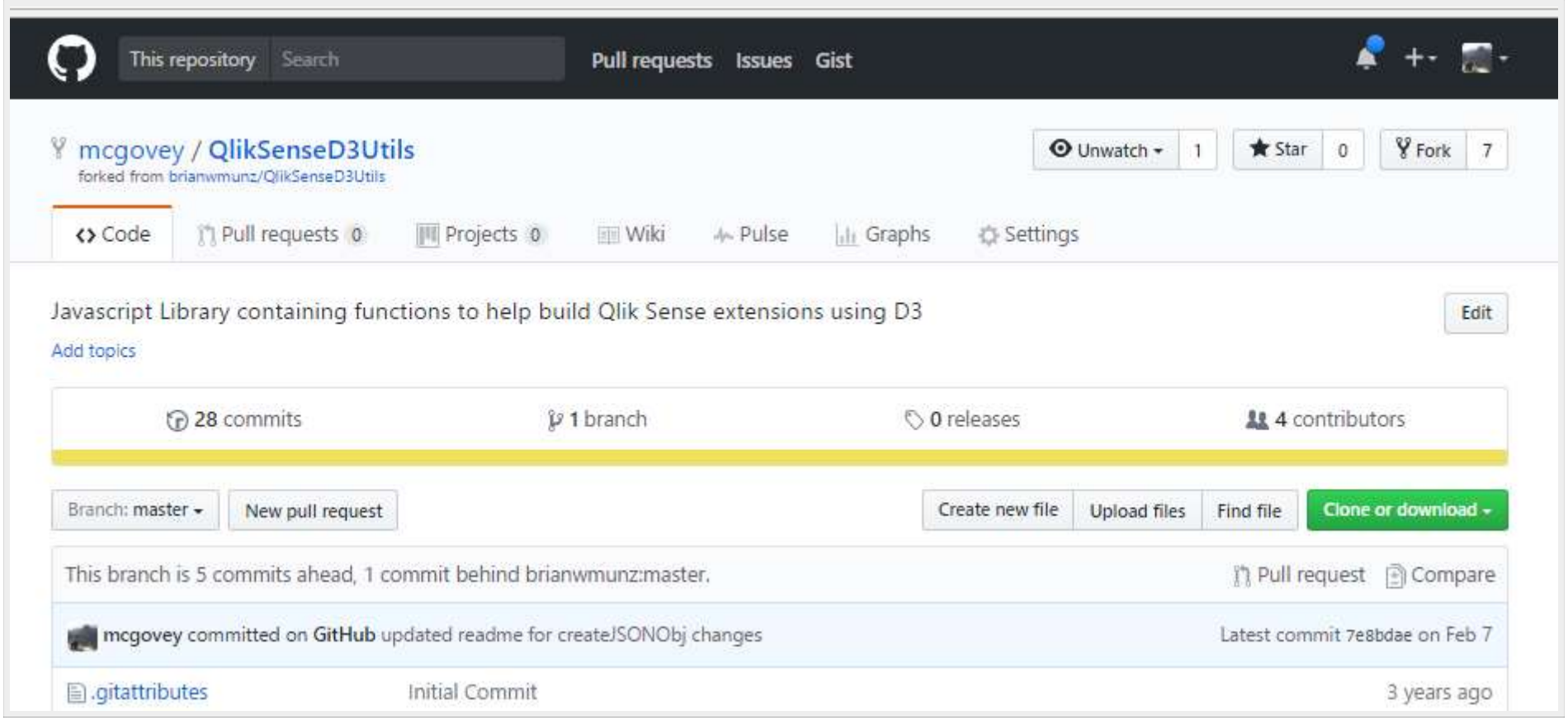
Eureka!


_MyExtensionPlayground
ClearAll
Back
Forward
Bookmarks
More
12%
11%
10%
9%
8%
7%
6%
5%
4%
3%
2%
1%
0%
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Link to source so far

Get data from your hypercube

But first prep data (**source**)



Minor edits

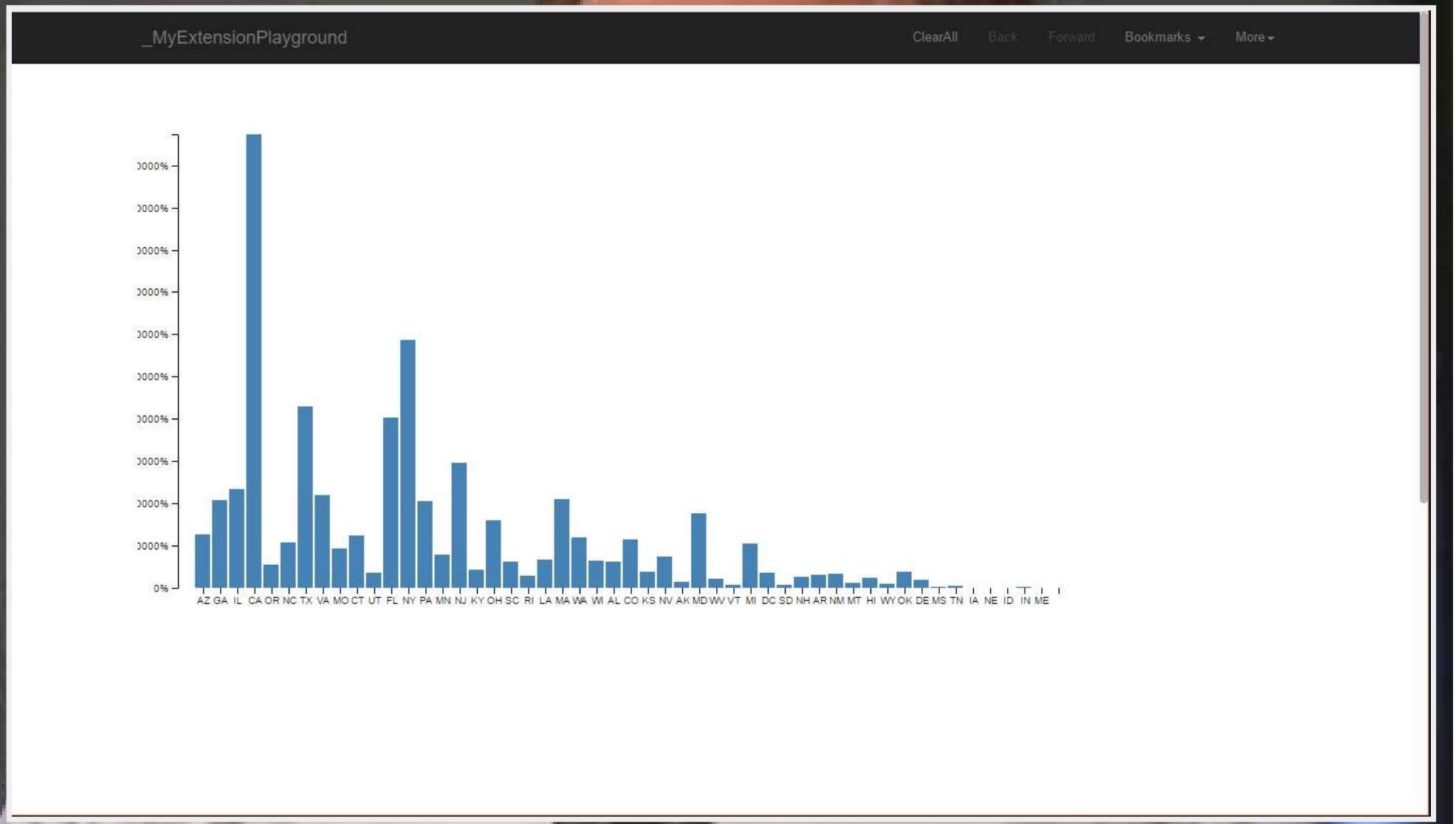
- Put the D3 code inside the callback function
- Clear the SVG before writing to it
- Comment out the call to the tsv
- Point the chart data to the hypercube variable

Source with these changes

Modified Example - Changing the Data

```
function getStateData(reply, app){
  //remove all d3 elements from svg
  $('#QV01 svg').empty();
  //store cleansed data
  var data = senseD3.createJSONObj(reply);
...
  g.selectAll(".bar")
  .data(data)
  .enter().append("rect")
    .attr("class", "bar")
    .attr("x", function(d) { return x(d.dim_0); })
    .attr("y", function(d) { return y(d.meas_0); })
    .attr("width", x.bandwidth())
    .attr("height", function(d) { return height - y(d.meas_0); })

}
```

Refresh and....


_MyExtensionPlayground
ClearAll
Back
Forward
Bookmarks
More
0%
AZ GA IL CA OR NC TX VA MO CT UT FL NY PA MN NJ KY OH SC RI LA MA WA WI AL CO KS NV AK MD WV VT MI DC SD NH AR NM MT HI WY OK DE MS TN IA NE ID IN ME


THAT'S PERFECT

We're done right? Right!?

Link to source

Well kinda

Some other things we would want to handle

- Window size changes
- Smoothing data changes
- Fix hard coded D3 selections
- Add selection listeners to the chart
- Leverage hypercube data

Selections (using the capability apis)

```
.on("click", function(d) {
        app.field('addr_state').select([d.dim_0_id], true, true);
});
```

Documentation

Capabilities API

What else can we do?

- Make/clear selections
- Forward/Backward through Selections
- Navigate to an app page
- Alternative States

Much More

Mashup source code

source

Using D3 in Extensions

Mostly the same as mashups, a few differences

- No hypercube creation, controlled by properties
- More error handling and data validation
- Handle data changes in a different way

What's different?

How does our data look different

- 'Element' is roughly equal to 'App' from Mashup (different contexts)
- 'Layout' is roughly equal to 'Reply' from HC
- Different data types could come in
- Varying numbers of dimensions and measures

What happens when our data changes in an extension?

- Redraw the full chart (cop out)
- Use Angular to detect data changes (difficult to pull off)
  - Too heavy for this but...source

What about using APIs with D3 extensions?

- Extension APIs used to set extension properties
  - Color picker
  - SelectValues
- Capability APIs from earlier still available

So what's possible?

An example!

Where can we go from here?

- D3 Project Page
- Scott Murray Tutorial***
- Bl.ocks
- D3 Documentation
- Sample Slideshare Presentation

Where can we go from here?

These slides at:

https://mcgovey.github.io/qonnections-2017-d3

Supporting repo:

https://github.com/mcgovey/qonnections-2017-d3