



FORUM ONE™

Create Data Visualizations with D3

github.com/deviantpixel/d3_data_viz

Visualizing information is a form of knowledge compression. It's a way of squeezing an enormous amount of information and understanding into a small space.

— David McCandless

Before you create a visualization ask yourself...

What story do I want
to tell with my data?

Does my data tell that
story?

What visualization
best tells my story?

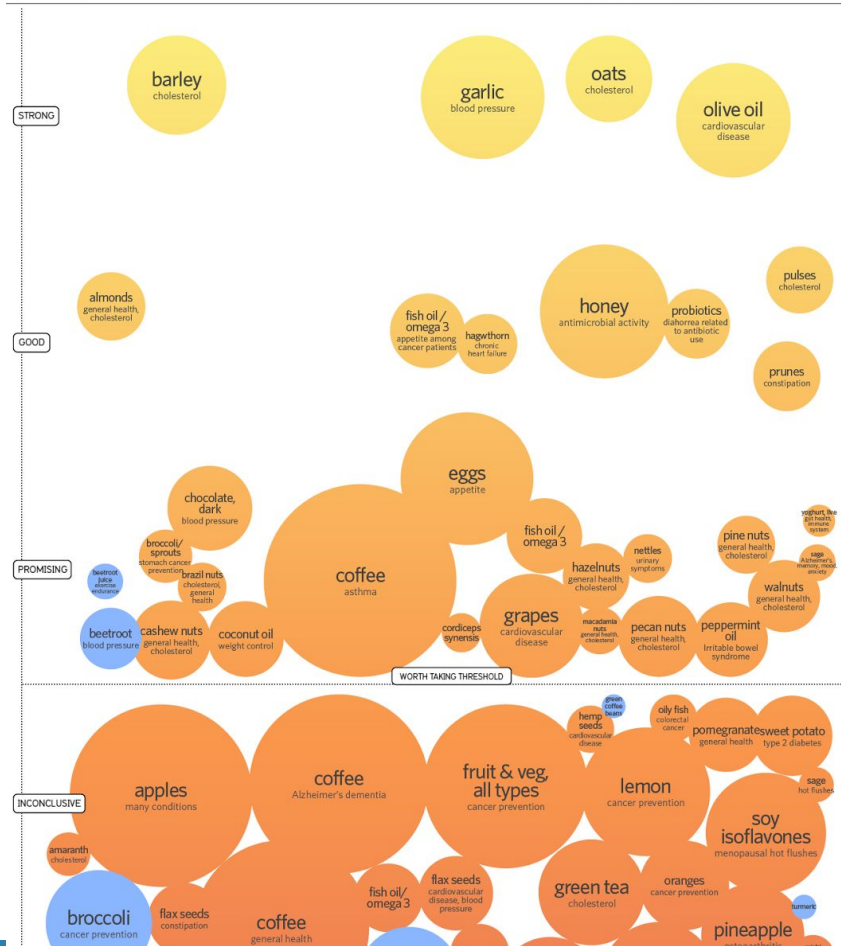
Snake Oil Superfoods?

Solid scientific evidence for extra health benefits of certain foods

EVIDENCE

Popularity
(Google hits)

One To Watch
(Few studies but promising results)

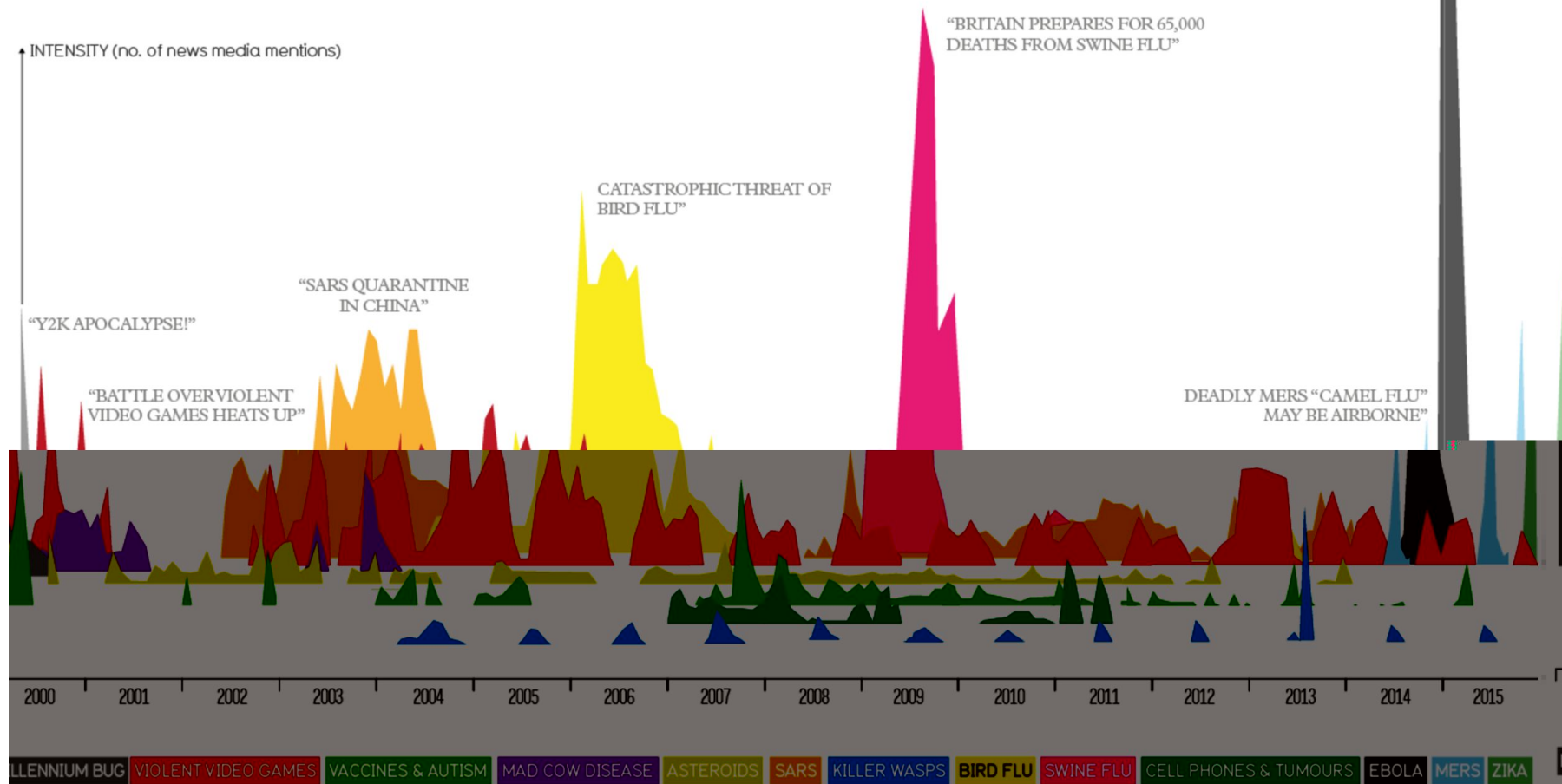


Mountains Out of Molehills

A timeline of media-inflamed fears

rollover to scale relative to ebola

"EBOLA OUTBREAK
OUT OF CONTROL"



What we'll cover today

Create Data Visualizations with D3

Part 1

D3 Anatomy: SVG, CSS, D3

Part 2

D3 Data Binding

Part 3

Drupal and D3

Part 4

D3 Examples



What we'll cover today

Create Data Visualizations with D3

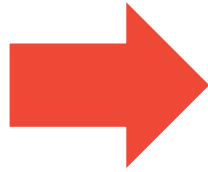
1

D3 Anatomy: SVG, CSS, D3

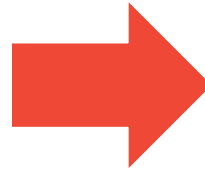
Architecture



Content



Structured
Data
(JSON, XML)



D3
(JS, CSS, HTML)

D3 Anatomy



Scalable Vector Graphics (SVG)

Scalable Vector Graphics (SVG) is an XML-based vector image format for two-dimensional graphics with support for interactivity and animation.

— Wikipedia

SVG Highlights

- DOM API
- Defines vector-based graphics for the Web
- Supports CSS styling
- Element grouping
- Hyperlinks
- Accessibility support (ARIA, etc)
- Path elements for arbitrary drawing

SVG (basic support) - REC

Global

97.18% + 0.49% = 97.67%

Method of displaying basic Vector Graphics features using the embed or object elements. Refers to the SVG 1.1 spec.

Current aligned Usage relative Date relative Show all

IE	Edge *	Firefox	Chrome	Safari	Opera	iOS Safari *	Opera Mini *	Android Browser *	Chrome for Android
		52	49			9.3		4.4	
	² 14	53	58		45	10.2		4.4.4	
² 11	² 15	54	59	10.1	46	10.3	all	56	59
	² 16	55	60	11	47	11			
		56	61	TP	48				
		57	62						

- Notes
- Known issues (4)
- Resources (7)
- Feedback

² IE9-11 desktop & mobile don't properly scale SVG files. Adding height, width, viewBox, and CSS rules seem to be the best workaround.



Common SVG Elements

svg

- Container element

circle, rect, line, ...

- Various shape elements

path

- Arbitrary drawing paths
- 'd' attribute for path data

g

- Used for grouping

a

- Links of course

text

- Textual content

Common SVG Elements

fill

- color of the inside of an element

stroke

- color of the border of an element

stroke-width

- width of the border

stroke-dasharray

- customizable dashes for lines

My first SVG

```
<html>
```

```
<body>
```

```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">
```

```
</svg>
```

```
</body>
```

```
</html>
```


My first SVG



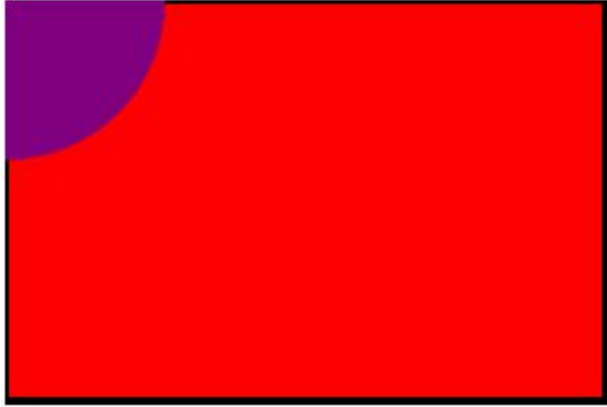
```
<html>  
<body>
```

```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">  
  <rect width="300" height="200" fill="red"  
    stroke="black" stroke-width="4">  
</svg>
```

```
</body>  
</html>
```

My first SVG



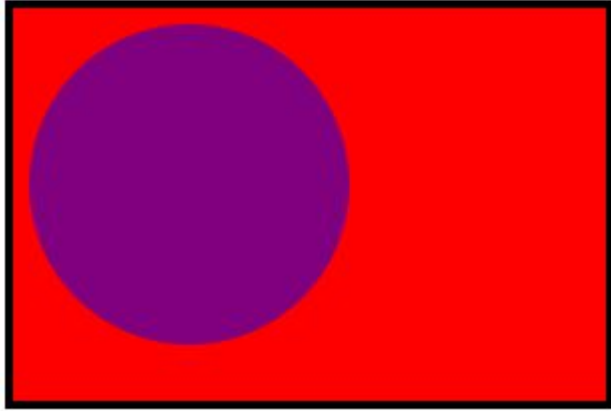
```
<html>  
<body>
```

```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">  
  <rect width="300" height="200" fill="red"  
stroke="black" stroke-width="4">  
    <circle r="80" fill="purple" />  
</svg>
```

```
</body>  
</html>
```

My first SVG



```
<html>  
<body>
```

```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">  
  <rect width="300" height="200" fill="red"  
stroke="black" stroke-width="4" x="10" y="10" />  
  <circle r="80" fill="purple" cx="100" cy="100" />  
</svg>
```

```
</body>  
</html>
```


My first SVG



```
<html>  
<body>
```

```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">  
  <rect width="300" height="200" fill="red"  
stroke="black" stroke-width="4" x="10" y="10" />  
  <circle r="80" fill="purple" cx="100" cy="100" />  
  <text x="70" y="80" fill="white">I love  
SVG!</text>  
</svg>
```

```
</body>  
</html>
```

SVG Path

M = moveto

L = lineto

H = horizontal lineto

V = vertical lineto

C = curveto

S = smooth curveto

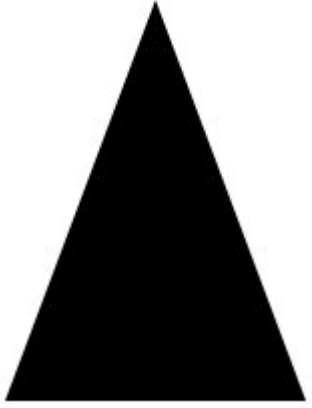
Q = quadratic Bézier curve

T = smooth quadratic Bézier curveto

A = elliptical Arc

Z = closepath

My first SVG



```
<html>  
<body>
```

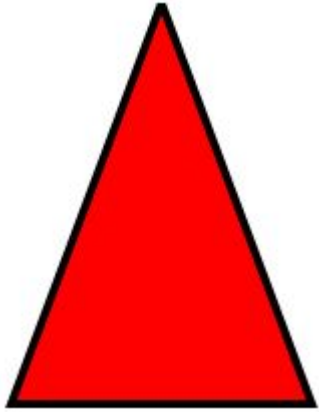
```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">  
  <path d="M150 0 L75 200 L225 200 Z" />  
</svg>
```

```
</body>  
</html>
```

CSS + SVG

My first SVG



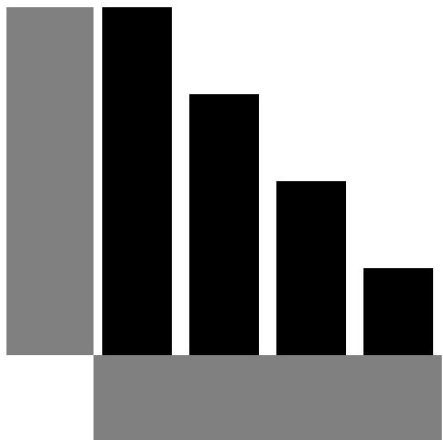
```
<html>  
<body>
```

```
<style>  
svg path {  
  fill:red;  
  stroke-width:4;  
  stroke:black;  
}  
</style>
```

```
<h1>My first SVG</h1>
```

```
<svg width="600" height="600">  
  <path d="M150 0 L75 200 L225 200 Z" />  
</svg>
```

```
</body>  
</html>
```



```
<html>  
<body>
```

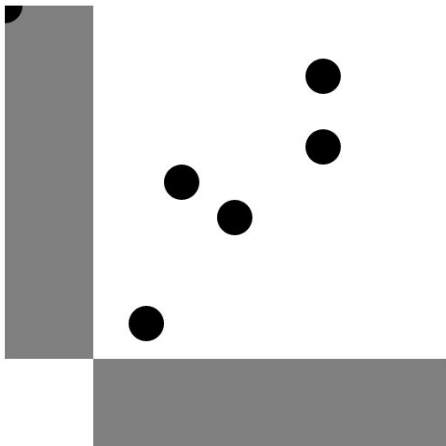
```
<svg width="250" height="250">
```

```
<rect x="0" y="0" width="50" height="200" fill="gray"/>  
<rect x="50" y="200" width="200" height="50" fill="gray"/>
```

```
<g transform="translate(50, 0)">  
  <rect x="5" y="0" width="40" height="200"/>  
  <rect x="55" y="50" width="40" height="150"/>  
  <rect x="105" y="100" width="40" height="100"/>  
  <rect x="155" y="150" width="40" height="50"/>  
</g>
```

```
</svg>
```

```
</body>  
</html>
```

```
<html>  
<body>
```

```
<svg width="250" height="250">
```

```
<rect x="0" y="0" width="50" height="200" fill="gray"/>  
<rect x="50" y="200" width="200" height="50" fill="gray"/>
```

```
<circle cx="0" cy="0" r="10"/>  
<circle cx="100" cy="100" r="10"/>  
<circle cx="130" cy="120" r="10"/>  
<circle cx="80" cy="180" r="10"/>  
<circle cx="180" cy="80" r="10"/>  
<circle cx="180" cy="40" r="10"/>
```

```
</svg>
```

```
</body>  
</html>
```

D3 + CSS + SVG

D3.js is a JavaScript library for manipulating documents based on data.

— D3js.org

D3js is the glue between data and SVG (or other DOM elements).

Why should I use it?

- Cross Browser Compatibility
- Easy to learn API
- Good documentation and examples
- Expansive library of data visualizations
- Out-of-the-box functions:
 - XHR data loading
 - Geo data conversion

D3 Selections

d3.select(selector: string)

- query one element

d3.selectAll(selector: string)

- query multiple elements

Example: SVG Element

```
var svg = d3.select('body')  
  .append('svg')  
  .attr('width', 960)  
  .attr('height', 500);
```


Adding Attributes

```
append("circle")  
    .attr('cx', 50)  
    .attr('cy', 50)  
    .attr('r', 40);
```



```
<div id="viz"/>
```

```
<script>
```

```
//Create a sized SVG surface within viz:
```

```
var vizsvg = d3.select("#viz")  
  .append("svg")  
  .attr("width", 600)  
  .attr("height", 600);
```

```
//Add to the svg surface a circle
```

```
var circle =  
  vizsvg.append("circle")  
  .attr("fill", "red")  
  .attr("r", 40)  
  .attr("cx", 50)  
  .attr("cy", 50);  
</script>
```

What we'll cover today

Create Data Visualizations with D3

2

D3 Data Binding



Data Joins

update = selection.data(data)

Bind array of data to selection.

update.enter()

Iterates over data points that don't have associated nodes.

update.exit()

Data nodes without matching data trigger this.



D3.js charts libraries

dimple, <http://dimplejs.org/> , 1/2/2015
Rickshaw, <http://code.shutterstock.com/rickshaw/> , 1/10/2015
dc.js, <http://dc-js.github.io/dc.js/> , 2/27/2015
Ember Charts, <http://addepar.github.io/#/ember-charts/overview> , 2/26/2015
MetricsGraphics.js, <http://metricsgraphicsjs.org/> , 2/27/2015
Epoch, <http://fastly.github.io/epoch/> , 2/22/2015
Plottable.js, <http://plottablejs.org/> , 2/21/2015
glimpse.js, <http://racker.github.io/glimpse.js-website/> , 10/7/2014
Vega, <https://github.com/trifacta/vega/> , 12/17/2014
xCharts, <http://tenxer.github.io/xcharts/> , 8/8/2013
uvCharts, <http://imaginea.github.io/uvCharts/> , 10/28/2013
Cubism.js, <http://square.github.io/cubism/> , 4/18/2014
c3, <http://c3js.org/> , 2/26/2015
TechanJS, <http://techanjs.org/> , 2/1/2015
Firespray, <https://github.com/boundary/firespray> , 2/25/2015
Micropolar, <http://micropolar.org/> , 11/4/2014
D4, <http://visible.io/> , 12/23/2014

NVD3, <https://github.com/novus/nvd3> , 2/9/2015
Gneisschart, <https://github.com/Quartz/Chartbuilder/> , 9/17/2014
wq/chart.js, <https://github.com/wq/wq.app/blob/master/js/wq/chart.js> , 2/23/2015
Angular-charts, <http://chinmaymk.github.io/angular-charts/> , 2/12/2015
react-d3 <http://esbullington.github.io/react-d3-website/> , 2/25/2015
DexCharts, <https://github.com/PatMartin/DexCharts> , 12/4/2013
DViz, <https://github.com/akngs/dviz> , 9/17/2012
gg, <https://github.com/sirrice/gg/> , 11/16/2013
jsplotlib, <https://github.com/rameshvs/jsplotlib> , 10/18/2014
LPChart, <https://github.com/bergcloud/lp-chart> , 7/29/2013
Radian, <https://github.com/openbrainsrc/Radian> , 3/7/2014
Dance.js, <https://github.com/michael/dance> , 11/17/2012
Dynamic-Graphs, <https://github.com/mlarocca/Dynamic-Charts> , 1/6/2013
PykCharts.js <http://pykcharts.com/> , 3/2/2015

The background of the slide features a blurred photograph of a glass and several small toys on a white surface. The glass has a red 'F' logo and the word 'FORUM' printed on it. To the right of the glass is a red cylindrical container, also with 'FORUM ONE' printed vertically. Next to the container is a red articulated robot toy with a white circular sticker on its chest that says 'ORACLE'. In the foreground, there are several small, colorful toys, including a brown bear-like figure with a blue and red mask, a black figure with a white and orange mask, and a grey figure with a white and red mask.

What we'll cover today

Create Data Visualizations with D3

3

Drupal and D3



Common Integration Methods

D3 Module (D7)

Views/REST API with
Custom Module

D3 Module

D3 Module Features

- Simple API
- Out-of-the-box basic visualizations
- Custom library support
- Simplified data binding through Views integration and custom library

Views/REST API with Custom Module

Views/REST API with Custom Module

1. Create a views listing of your data via REST API display mode
2. Create a custom module
 - a. Add D3 Library JS file
 - b. Outputs needed markup via Block or other
 - c. Add your custom D3/JS



What we'll cover today

Create Data Visualizations with D3

5

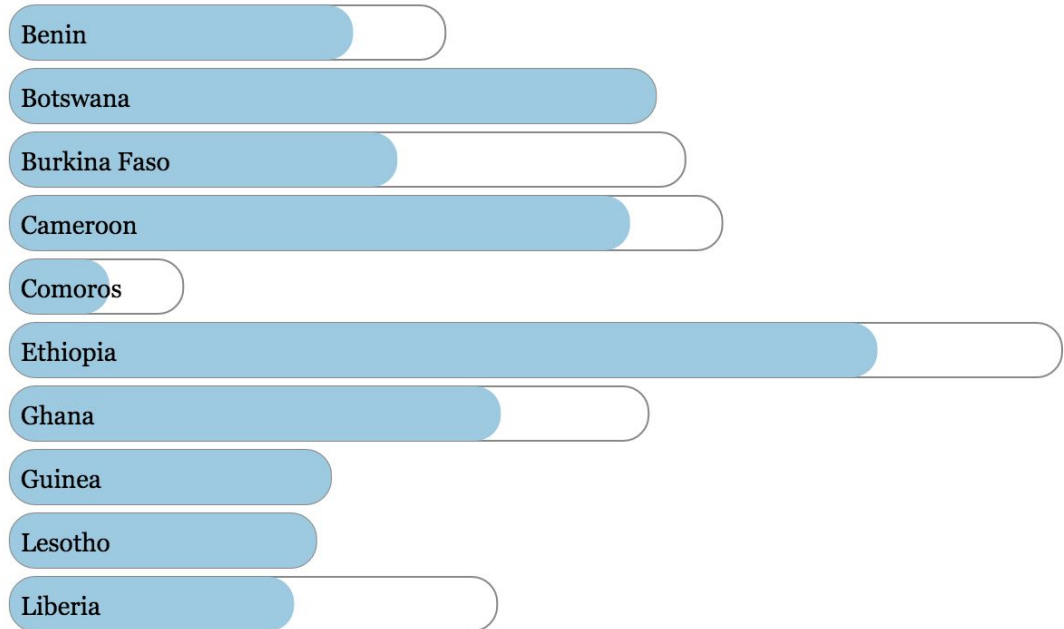
D3 Examples



What are we building?

- Shows Peace Corps Volunteers per country in Africa
- Country name is visible
- Visually see how many people are currently volunteering and how many openings are left

Volunteers per Country D3



REST API View (Drupal 8)

Enable the following core modules:

▼ WEB SERVICES

HAL

► Serializes entities using Hypertext Application Language

HTTP Basic Authentication

► Provides the HTTP Basic authentication provider

RESTful Web Services

► Exposes entities and other resources as RESTful web API

Serialization

► Provides a service for (de)serializing data to/from formats such as JSON and XML

REST API View (Drupal 8)

1. Navigate to the Views Admin page and create a new content view.
2. Create a new sub-view of type “REST export”.
 - a. Change “Show” to “Fields” to allow field selection
 - b. Configure the fields to include what you need for your visualization.
 - c. Configure the filters to only allow content you need for your visualization.



REST API View (Drupal 8)

Display name: **Countries REST export** View Countries REST export

TITLE
Title: [None](#)

FORMAT

[Format: Serializer](#) | [Settings](#)

[Show: Fields](#) | [Settings](#)

FIELDS

Add

▼

[Content: Title](#)

[Content: Flag](#)

[Content: Active volunteers total](#)

[Content: Volunteer positions available](#)

[Content: Region](#)

FILTER CRITERIA

Add

▼

[Content: Publishing status \(= Yes\)](#)

[Content: Content type \(= Country\)](#)

SORT CRITERIA

Add

▼

[Content: Title \(asc\)](#)

PATH SETTINGS

Path: [/countries-api](#)

~~Access: [Permission](#)~~ | [View published content](#)


HEADER
The selected display type does not use header plugins

FOOTER
The selected display type does not use footer plugins

NO RESULTS BEHAVIOR
The selected display type does not use empty plugins

PAGER
Items to display:
[Display a specified number of items](#) | [10 items](#)

► ADVANCED



REST API View (Drupal 8)

Your JSON data will be returned by GET request to your specified URL path.

Note:

We are using a defined query without custom GET attributes. To access all REST actions you will need to enable additional core modules and configure REST permissions and configuration.

```
[
  - {
    title: "Benin",
    field_flag: "1",
    field_active_volunteers_total: "93",
    field_volunteer_positions_availa: "25",
    field_region: "Africa"
  },
  - {
    title: "Botswana",
    field_flag: "2",
    field_active_volunteers_total: "175",
    field_volunteer_positions_availa: false,
    field_region: "Africa"
  },
  - {
    title: "Burkina Faso",
    field_flag: "3",
    field_active_volunteers_total: "105",
    field_volunteer_positions_availa: "78",
    field_region: "Africa"
  },
]
```

Custom Module (Drupal 8)

d3vis (directory at modules/custom)

- **d3vis.info.yml** (contains module info)
- **d3vis.module** (contains theme hook for block)
- **d3vis.libraries.yml** (includes D3.js, custom JS and JS dependencies)
- **js** (directory)
 - **d3.min.js** (D3.js version 3.x)
 - **d3vis_src.js** (Our custom D3)
- **src/Plugin/Block** (directories)
 - **D3VisBlock.php** (extends block class to create our custom block)
- **templates** (directories)
 - **d3vis.html.twig** (custom twig template file for our custom block)

d3vis.info.yml

- Sets standard module information

name: D3 Data Visualization

type: module

description: Provides D3 rendering of content.

core: 8.x

package: d3

D3VisBlock.php

- Extend the Block class Plugin to create our custom block
- Sets the block theme to our custom block theme
- Attaches our custom D3/JS file to the block so it is included when the block is rendered by our custom library
'ModuleName/LibraryName'

```
<?php
namespace Drupal\d3vis\Plugin\Block;

use Drupal\Core\Block\BlockBase;

/**
 * Provides a 'D3VisBlock' block.
 */
class D3VisBlock extends BlockBase {

  /**
   * {@inheritdoc}
   */
  public function build() {
    $build = [];
    $build['d3_vis_block']['#markup'] = 'D3 vis here.';
    $build['d3_vis_block']['#attached']['library'][] = 'd3vis/d3vis_src';
    $build['d3_vis_block']['#theme'] = 'd3vis';
    return $build;
  }
}
```

d3vis.libraries.yml

- Define 'd3' custom library
 - Includes the D3 release file Version 3.x
 - Sets another dependency on Drupal core jQuery
- Define 'd3vis_src' custom library
 - Includes our custom D3/JS
 - Sets dependencies on our d3 library as well as needed Drupal core libraries

```
d3:  
  version: 3.x  
  js:  
    js/d3.min.js: {}  
  dependencies:  
    - core/jquery  
d3vis_src:  
  version: 1.x  
  js:  
    js/d3vis_src.js: {}  
  dependencies:  
    - core/jquery  
    - core/jquery.once  
    - core/drupal  
    - core/drupalSettings  
    - d3vis/d3
```

d3vis.module

- Defines our custom 'd3vis' theme which will provide the block theme for our custom block

...

```
function d3vis_theme($existing, $type, $theme, $path) {  
  return array(  
    'd3vis' => array(  
      'variables' => array(),  
    ),  
  );  
}
```

...

d3vis.html.twig

- Essentially the same as the core block template file except:
 - Container added for the visualization to be placed in
 - We directly inject styles

Note:

We should include CSS using a custom library.

```
<style>
    #d3vis-container .total-bar {
        fill: #fff;
        stroke: #888;
        stroke-width: 1;
    }
    #d3vis-container .progress-bar {
        fill: rgb(158, 202, 225);
    }
</style>

<div{{ attributes }}>
    {{ title_prefix }}
    {% if label %}
        <h2{{ title_attributes }}>{{ label }}</h2>
    {% endif %}
    {{ title_suffix }}
    {% block content %}
        {{ content }}
        <div id="d3vis-container" />
    {% endblock %}
</div>
```

d3vis_src.js

- All custom JS is wrapped in a Drupal behavior and uses 'once' to ensure the JS is only included one time and only when needed

```
(function ($) {  
  'use strict';  
  Drupal.behaviors.d3vis_src = {  
    attach: function(context, settings) {  
      // This just ensures this JS is called once to avoid Drupal ajax recalls  
      $(context).find('#block-d3visblock').once('d3vis_src').each(function  
    ) {  
  
      ...  
  
    }  
  };  
}(jQuery));
```

D3vis_src.js (cont.)

- Create the svg element for our visualization
- GET the data from the REST API path

Note:

All remaining code will be inside the context of this data load function.

...

// Set the variables used to control the svg size

var width = 600;

var height = 600;

// Nicer to set how high the bars should be

var barHeight = 30;

//Create a sized SVG surface within d3vis-container:

var vizsvg = d3.select("#d3vis-container")

.append("svg")

.attr("width", width)

.attr("height", height);

// Use D3 to fetch the json data from the Drupal REST path

d3.json("/countries-api", function(error, data) {

if (error) throw error; // always great to check for errors

...

D3vis_src.js (cont.)

- D3.scale.linear is used to “scale” data
 - Domain is the upper and lower bounds of the data
 - Range is the upper and lower bounds you are constraining it to

...

/ We use this to scale all data to fit the max width of the viewing area. It translates our data range which is the domain into the viewable area width which is the range */*

```
var widthScale = d3.scale.linear()  
  .domain([0, d3.max(data, function(d) {  
    /* We determine the upper range of our data by using the max  
    function which determines the maximum value in a dataset */  
    return checkInt(d.field_active_volunteers_total) +  
    checkInt(d.field_volunteer_positions_availa); }]))  
  .range([0, width-20]);
```

...

D3vis_src.js (cont.)

- Add a rect for each data node that is as wide as the scaled result of total active volunteers and open positions available

Ghana



```
...
// Create dom elements for the total bar
vizsvg.selectAll(".total-bar")
  // Iterate over each data node
  .data(data).enter()
  // create a rect element
  .append("rect")
  // Assign attributes to the rect element
  .attr("class", "total-bar")
  .attr("rx", 14) //Just adds rounded corners
  .attr("ry", 14) //Just adds rounded corners
  .attr("x", 5) // Set the horizontal position of the bar
  // Set the vertical position of the bar
  .attr("y", function(d, i) {
    // i is the current data node index
    return (i * (barHeight + 5)) + 25;
  })
  /* The width of the bar uses a function to combine the two
  field values to calculate the total width */
  .attr("width", function(d, i) {
    return
    (widthScale(checkInt(d.field_active_volunteers_total) +
    checkInt(d.field_volunteer_positions_availa)));
  })
  .attr("height", barHeight);
```

...

D3vis_src.js (cont.)

- Add a rect for each data node that is as wide as the scaled result of total active volunteers



Note:

We set the width to 0px as a transition will animate the width to the right size.

...

```
// Create dom elements for the progress bar
vizsvg.selectAll(".progress-bar")
  // Iterate over each data node
  .data(data).enter()
  // create a rect element
  .append("rect")
    .attr("class", "progress-bar")
    .attr("rx", 14) //Just adds rounded corners
    .attr("ry", 14) //Just adds rounded corners
    .attr("x", 5) // Set the horizontal position of the bar
    // Set the vertical position of the bar
    .attr("y", function(d, i) {
      // i is the current data node index
      return (i * (barHeight + 5)) + 25;
    })
    // Width is set to 0 as we will transition it in later
    .attr("width", 0)
    .attr("height", barHeight);
```

...

D3vis_src.js (cont.)

- Add a text for each data node that contains the name of the country



...

// Create dom elements for the Country names

vizsvg.selectAll(".title")

// Iterate over each data node

.data(data).enter()

// create a text element

.append("text")

.attr("class", "title")

// Position the title to be on the corresponding bar

.attr("x", 11)

.attr("y", function(d, i) {

// i is the current data node index

return (i * (barHeight + 5)) + 46;

})

// Set the content of the text element to be the country

name

.text(function(d) {

return (d.title);

});

...

D3vis_src.js (cont.)

- Add a transition that animates the progress bars to transition from 0 width to the width of the scaled total active volunteers

Ghana



...

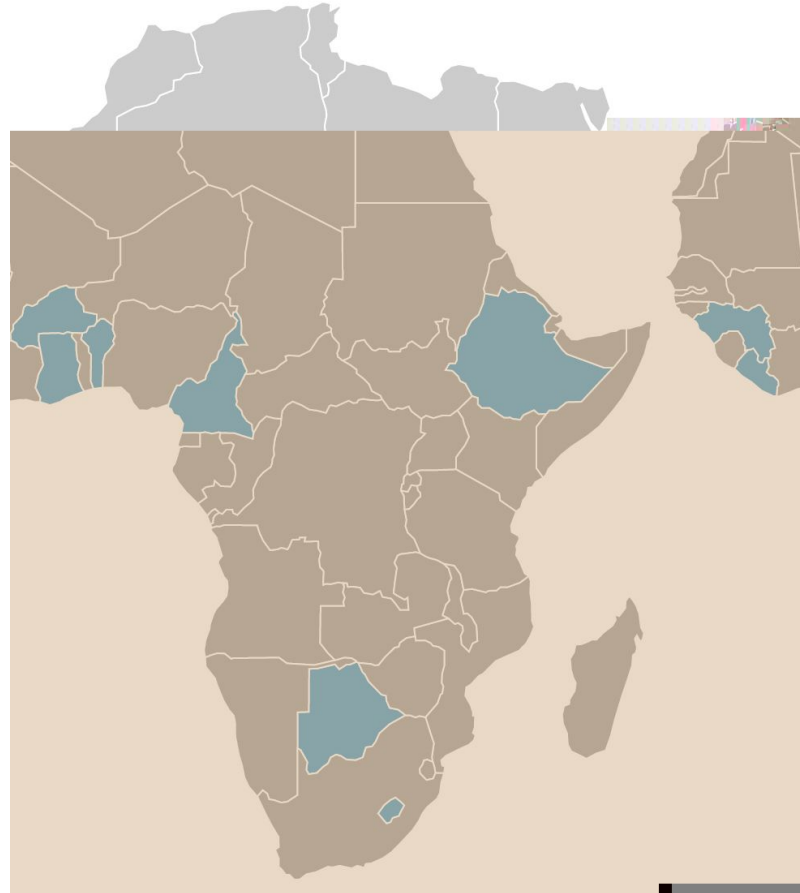
```
/* Create a transition effect on the progress bar. This makes it grow from 0
width to the total width it should be in an animated way */
vizsvg.selectAll(".progress-bar")
  .transition()
    .duration(750) // The transition will take less than a second
/* We set the ending state for the transition which is that it have
a width attribute of the total active volunteers scaled of course.
The D3 animation function takes care of everything in between
*/
    .attr("width", function(d, i) {
      return
        (widthScale(checkInt(d.field_active_volunteers_total)));
    });
```

...

Bring in the Maps

What are we building?

- Map of Africa highlighting countries that have volunteer programs



D3vis_src.js (cont.)

- Create the svg and group element for our map
- Set a map projection so all geo data is translated correctly

Note:

This assumes a container element with an id of “d3map-container” exists.

...

```
/* Start map section */
```

```
//Create a sized SVG surface within d3vis-container:
```

```
var mapsvg = d3.select("#d3map-container")  
  .append("svg")  
  .attr("width", width)  
  .attr("height", height);
```

```
// Append a group element to place all country shapes in
```

```
var g = mapsvg.append("g");
```

```
// Set a map projection as well as other map configuration
```

```
var projection = d3.geo.mercator()  
  .scale( 400 )  
  .center( [20, 0] )  
  .translate( [width/2,height/2] );
```

```
// Create a geographic path generator and set the projection to be used on  
all GeoJson paths
```

```
var geoPath = d3.geo.path()  
  .projection( projection );
```

...

D3 Geo Paths

d3.geo.path()

Given a geometry or feature object, it generates the path data string suitable for the "d" attribute of an SVG path element

D3vis_src.js (cont.)

- GET Africa geo.json data from the Africa shape file and store in mapdata
- GET the data from the REST API path and store in countryData

Note:

The following code will be inside this nested data load.

...

```
// Load the GeoJson shape data of countries in Africa
d3.json("/modules/custom/d3vis/js/africa.geo.json", function(error,
mapdata) {
  if (error) throw error; // always great to check for errors
  // Load the countries data again to use this data as well
  d3.json("/countries-api", function(error, countryData) {
    if (error) throw error; // always great to check for errors
```

...

D3vis_src.js (cont.)

- Create a Path element for each country in the Africa geojson data set
- Color that path blue if it is present in the countryData data set or gray if not
- D3 generates the actual “d” path value using the projection variable we created

...

```
// Select all path to begin creating path elements
g.selectAll("path")
  // We will be creating a path for each data item in mapdata
  .data( mapdata.features ).enter()
  .append("path")
  /* Fill the color with the default gray unless it has a country name
  that appears in our countryData dataset. If it does color it blue. */
  .attr("fill", function(d, i) {
    var fillColor = "#ccc"; // Set default gray color
    jQuery.each(countryData, function() {
      if (this.title == d.properties.name_long) {
        // Found this country in our countrydata list. make it blue
        fillColor = "rgb(158, 202, 225)";
      }
    });
    return fillColor; // Return the resulting background color
  })
  .attr("d", geoPath ) // Add the GeoJson path data to the d attribute
  // using the geographic path generator
  .attr("stroke-width", 1)
  .attr("stroke", "#fff");

});
});
```

...

Learn More...

- [Introduction to D3 Screencast - Curran Kelleher](#)
- [Visualization Monsters - Christophe Viau](#)
- [D3 Data Visualization Code Repository](#)
- [Mapping with D3 - Maptime Boston](#)
- [A Data-Driven Tour of the Universe](#)
- [The beauty of data visualization - David McCandless Ted talk](#)
- [Let my dataset change your mindset - Hans Rosling Ted talk](#)

Questions?

Keenan Holloway, Technical Architect
kholloway@forumone.com
www.forumone.com

