



Diasporas, Data, D3.js, *Oh my!*





What I built.



How I built it.



Why I built it.

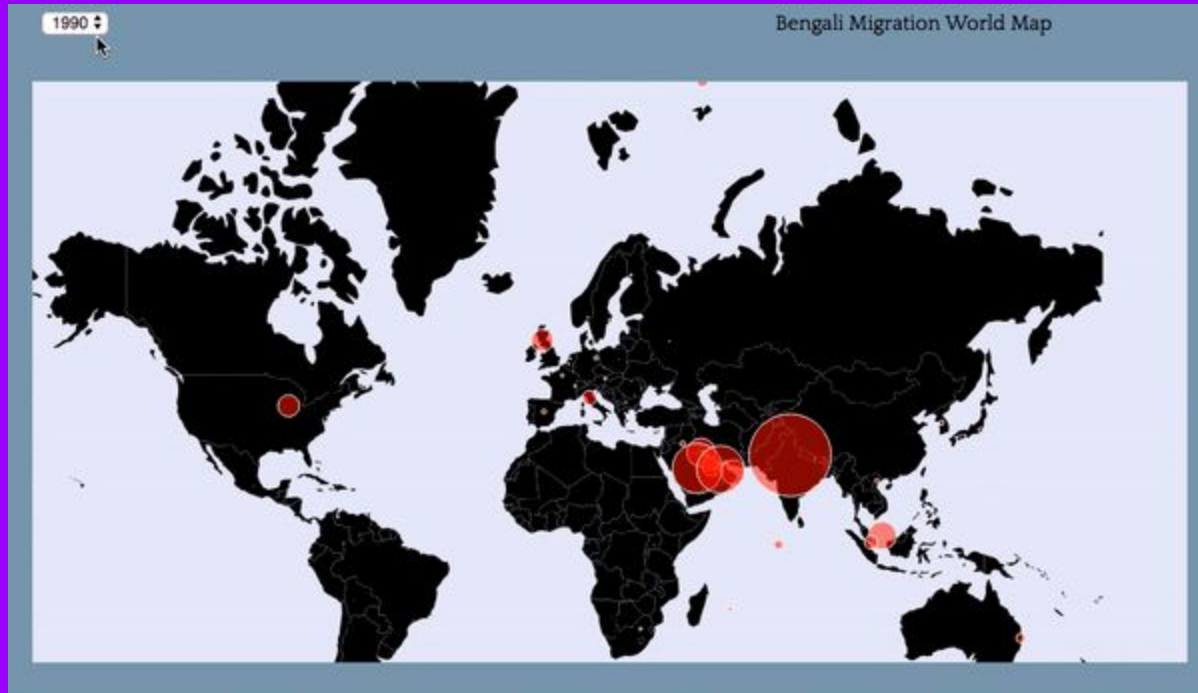


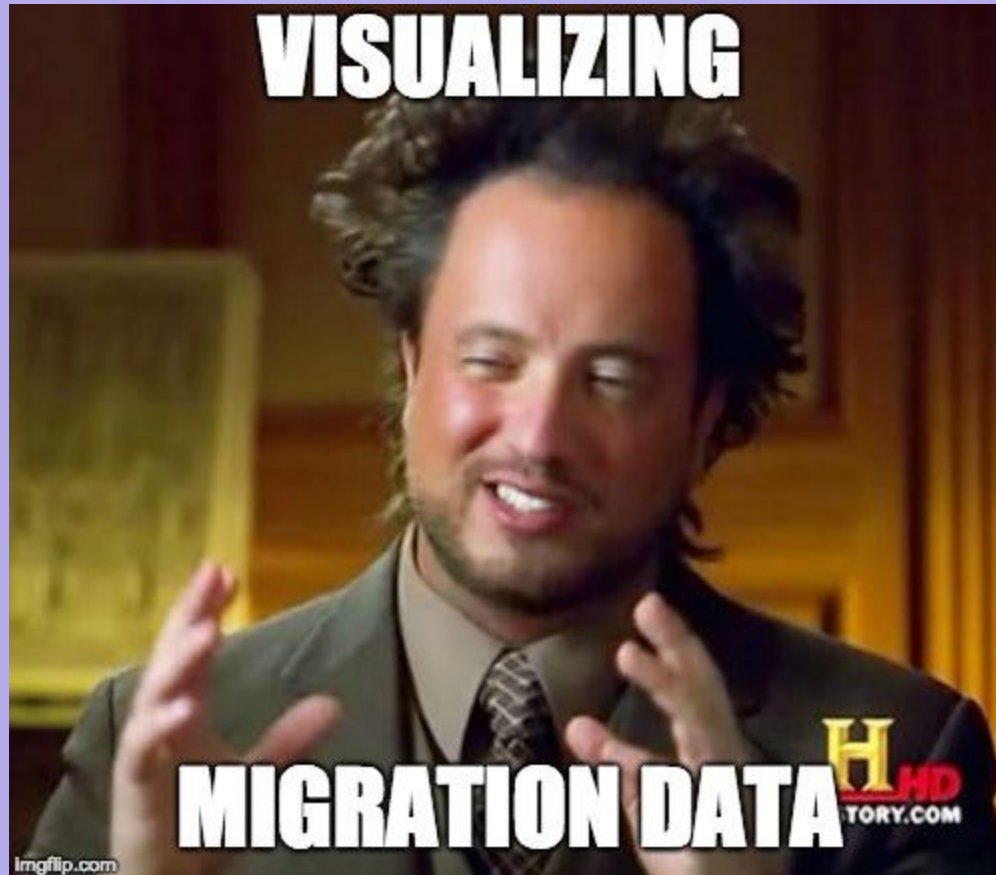
Was it all even worth it?

1

What I built

Visualizing the aftermath of a genocide.







200 - 300 Million

Bengalis in the world

32 Million

Have migrated out of Bangladesh since 2000





Running through the UN archives with my woes.




Year	Sort order	Major area, region, country or area of destination	Notes	Code	Type of data (a)	Bangladesh
1990	1990002	More developed regions	b	901		160,805
1990	1990003	Less developed regions	c	902		5,290,741
1990	1990004	Least developed countries	d	941		2,047
1990	1990005	Less developed regions, excluding least developed countries		934		5,288,694
1990	1990006	High-income countries	e	1503		1,038,610
1990	1990007	Middle-income countries	e	1517		4,412,312
1990	1990008	Upper-middle-income countries	e	1502		34,282
1990	1990009	Lower-middle-income countries	e	1501		4,378,030
1990	1990010	Low-income countries	e	1500		624
1990	1990011	Sub-Saharan Africa	f	947		2,649
1990	1990012	AFRICA		903		3,079
1990	1990013	Eastern Africa		910		132



Index	Region, subregion, country or area	Notes	Code	More Developed Regions	Less Developed Regions	Least developed countries	High-income Countries
230	French Guiana		254	N	Y	N	N/A
231	Guyana		328	N	Y	N	N
232	Paraguay		600	N	Y	N	N
233	Peru		604	N	Y	N	N
234	Suriname		740	N	Y	N	N
235	Uruguay		858	N	Y	N	Y
236	Venezuela (Bolivarian Republic of)		862	N	Y	N	N
237	NORTHERN AMERICA		905				
238	Bermuda		66	Y	N	N	Y
239	Canada		124	Y	N	N	Y
240	Greenland		304	Y	N	N	Y
241	Saint Pierre and Miquelon		666	Y	N	N	N/A
242	United States of America		840	Y	N	N	Y
243	OCEANIA		909				
244	Australia/New Zealand		927				
245	Australia	24	36	Y	N	N	Y
246	New Zealand		554	Y	N	N	Y

D3.JS is a great introduction
to data visualization.

	A	B	C	D	E	F	G	H	I		A	B	C	D	E	F	A	B	C	D
1	 United Nations Population Division Department of Economic and Social Affairs										1	Year	location	latitude	longitude	num_of_people	1	Mauritius	1990	21
2											2	1990	Australia	-28.86944	153.04453	2315	2	Seychelles	1990	4
3											3	1990	Austria	48.210033	16.363449	1508	3	Uganda	1990	9
4											4	1990	Bahrain	26.156258	50.400505	20442	4	Egypt	1990	89
5											5	1990	Belgium	51.260197	4.402771	268	5	Libya	1990	252
6											6	1990	Bhutan	26.90069	89.199551	24	6	Namibia	1990	30
7											7	1990	Bolivia	-17.413977	-66.165321	5	7	SouthAfrica	1990	1763
8											8	1990	BruneiDarussalam	4.58361	115.06667	659	8	China	1990	631
9											9	1990	Bulgaria	42.698334	23.319941	22	9	HongKongSA	1990	194
10											10	1990	Canada	-124.01289	48.370846	4203	10	NorthKorea	1990	21
11											11	1990	China	22.210928	113.552971	1858	11	Japan	1990	1075
12											12	1990	HongKongSA	22.28552	114.15769	297	12	SouthKorea	1990	654
13											13	1990	CostaRica	9.934739	-84.087502	13	13	Bhutan	1990	16
14											14	1990	Cyprus	35.095192	33.20343	162	14	India	1990	2324893
15											15	1990	Czechia	50.598427	15.898574	8	15	Maldives	1990	892
16											16	1990	NorthKorea	37.90889	124.39806	74	16	Nepal	1990	43
17											17	1990	Denmark	55.676098	12.568337	241	17	SriLanka	1990	5
18											18	1990	Egypt	31.205753	29.924526	112	18	BruneiDarussalam	1990	593
19											19	1990	ElSalvador	13.90519	-89.500206	11	19	Malaysia	1990	22967
20											20	1990	Fiji	-17.713371	178.065033	1368	20	Myanmar	1990	1158
21											21	1990	Finland	60.165249	24.945831	115	21	Philippines	1990	39
22											22	1990	France	48.858093	2.294694	2198	22	Singapore	1990	5840
23											23	1990	Germany	52.520008	13.404954	4767	23	Thailand	1990	438
24											24	1990	Greece	37.98381	23.727539	7665	24	Timor-Leste	1990	60
25											25	1990	Hungary	47.497913	19.040236	60	25	VietNam	1990	215
26											26	1990	India	28.6448	77.216721	4375155	26	Bahrain	1990	18393
27											27	1990	Ireland	53.35014	-6.266155	157	27	Cyprus	1990	159
28											28	1990	Italy	43.769562	11.255814	5421	28	Jordan	1990	651
29											29	1990	Japan	35.652832	139.839478	2110	29	Kuwait	1990	109188
30											30	1990	Jordan	31.963158	35.930359	948	30	Lebanon	1990	297
31											31	1990	Kuwait	29.378586	47.990341	126783	31	Oman	1990	49323
32											32	1990	Latvia	56.946285	24.105078	3	32	Qatar	1990	33076
33											33	1990	Lebanon	33.88863	35.49548	433	33	SaudiArabia	1990	357396
34											34	1990	Libya	32.885353	13.180161	318	34	Turkey	1990	73
35											35	1990	Luxembourg	49.611622	6.131935	29	35	UnitedArabEmirates	1990	139177
36											36	1990	Malaysia	3.519863	101.538116	23930	36	Bulgaria	1990	18
37											37	1990	Maldives	0.5	73.399658	1623	37	Czechia	1990	7
38											38	1990	Malta	35.917073	14.400043	6	38	Hungary	1990	52



Nishat @ EmpireJS

@thathijabae



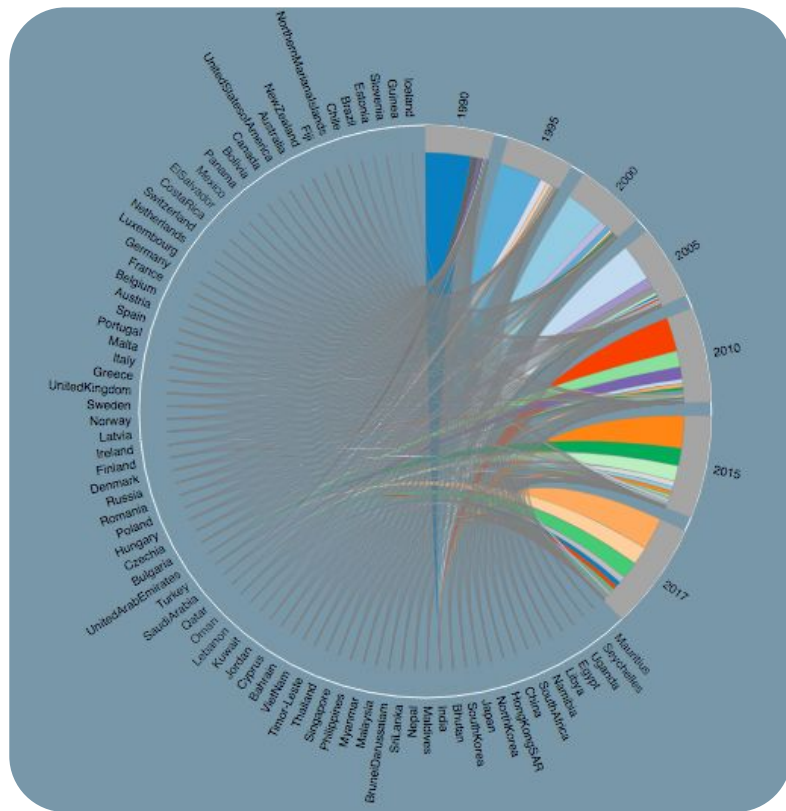
When people leave me in charge of their UI



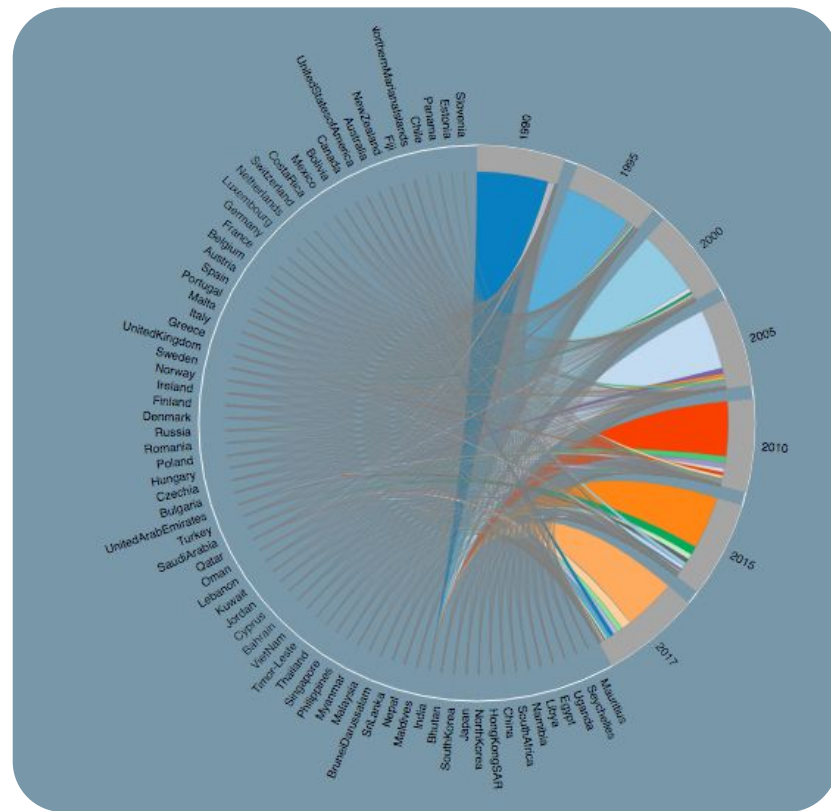
6:06 PM - 21 Sep 2017



Persons identified as male:



Persons identified as female:



2

How I built it

Let's look at some code.


```

var selectedCountry = null;

var margin = {
  top: 20,
  right: 20,
  bottom: 20,
  left: 20
};

var width = 1350,
    height = 800;

// Map sizing, numbers based on stylistic placement.
var projection = d3.geo.mercator()
  .scale(160)
  .translate([width / 1.9, height / 1.7]);

// Enables zoom behavior.
// The scaleExtent range prevents excessive zooming
var zoom = d3.behavior.zoom()
  .scaleExtent([1, 8])
  .on("zoom", zoom);

var path = d3.geo.path()
  .pointRadius(2)
  .projection(projection);

// Draws the map
var svg = d3.select(".map")
  .append('svg')
  .attr('width', width + margin.left + margin.right)
  .attr('height', height + margin.top + margin.bottom)
  .attr('transform', 'translate(' + margin.left + "," + margin.top + ')')
  .call(zoom);

// Makes sure that svg elements are bound together
var g = svg.append('g');
```

```

d3.json(worldMapJson, function(error, countries) {
  d3.json(populationsJson, function(error, populations) {
    let mapG = svg.append("g");
    let map = mapG.append("g");

    //Map Styles
    mapG.selectAll("path")
      .data(countries.features)
      .enter().append("path")
      .attr("d", path)
      .style("fill", "black")
      .style("stroke", "white")
      .style("stroke-width", "0.1");

    let trans = {};
    let zoom = d3.zoom()
      .scaleExtent([1, 8])
      .on("zoom", () => {
        trans["stroke-width"] = 1.5 / d3.event.transform.k + "px";
        trans["transform"] = d3.event.transform;
        mapG.style("stroke-width", trans["stroke-width"]);
        mapG.attr("transform", trans["transform"]);
        map.style("stroke-width", trans["stroke-width"]);
        map.attr("transform", trans["transform"]);
      });
    svg.call(zoom);
```

```
{ "type": "FeatureCollection", "features": [ { "type": "Feature", "properties": { "name": "Afghanistan", "geometry": { "type": "Polygon", "coordinates": [[ [61.210817, 35.650072], [62.230651, 35.270664], [62.984662, 35.404041], [63.193538, 35.857166], [63.982896, 36.007957], [64.546479, 36.312073], [64.746105, 37.111818], [65.588948, 37.305217], [65.745631, 37.661164], [66.217385, 37.39379], [66.518607, 37.362784], [67.075782, 37.356144], [67.83, 37.144994], [68.135562, 37.023115], [68.859446, 37.344336], [69.196273, 37.151144], [69.518785, 37.608997], [70.116578, 37.588223], [70.270574, 37.735165], [70.376304, 38.138396], [70.806821, 38.486282], [71.348131, 38.258905], [71.239404, 37.953265], [71.541918, 37.905774], [71.448693, 37.065645], [71.844638, 36.738171], [72.193041, 36.948288], [72.63689, 37.047558], [73.260056, 37.495257], [73.948696, 37.421566], [74.980002, 37.41999], [75.158028, 37.133031], [74.575893, 37.020841], [74.067552, 36.836176], [72.920025, 36.720007], [71.846292, 36.509942], [71.262348, 36.074388], [71.498768, 35.650563], [71.613076, 35.153203], [71.115019, 34.733126], [71.156773, 34.348911], [70.881803, 33.988856], [69.930543, 34.02012], [70.323594, 33.358533], [69.687147, 33.105499], [69.262522, 32.501944], [69.317764, 31.901412], [68.926677, 31.620189], [68.556932, 31.71331], [67.792689, 31.58293], [67.683394, 31.303154], [66.938891, 31.304911], [66.381458, 30.738899], [66.346473, 29.887943], [65.046862, 29.472181], [64.350419, 29.560031], [64.148002, 29.340819], [63.550261, 29.468331], [62.549857, 29.318572], [60.874248, 29.829239], [61.781222, 30.73585], [61.699314, 31.379506], [60.941945, 31.548075], [60.863655, 32.18292], [60.536078, 32.981269], [60.9637, 33.528832], [60.52843, 33.676446], [60.803193, 34.404102], [61.210817, 35.650072]] ] } }, { "id": "AFG" },
```

```

Year,location,latitude,longitude,num_of_people
1990,Australia,-28.86944,153.04453,2315
1995,Australia,-28.86944,153.04453,5562
2000,Australia,-28.86944,153.04453,8480
2005,Australia,-28.86944,153.04453,16130
2010,Australia,-28.86944,153.04453,29590
2015,Australia,-28.86944,153.04453,37092
2017,Australia,-28.86944,153.04453,38888
1990,Austria,48.210033,16.363449,1508
1995,Austria,48.210033,16.363449,1701
2000,Austria,48.210033,16.363449,1894
2005,Austria,48.210033,16.363449,2117
2010,Austria,48.210033,16.363449,2340
2015,Austria,48.210033,16.363449,2737
2017,Austria,48.210033,16.363449,3013
1990,Bahrain,26.156258,50.400505,20442
1995,Bahrain,26.156258,50.400505,24630
2000,Bahrain,26.156258,50.400505,28889
2005,Bahrain,26.156258,50.400505,49900
2010,Bahrain,26.156258,50.400505,81920
2015,Bahrain,26.156258,50.400505,78396
2017,Bahrain,26.156258,50.400505,80457
1990,Belgium,51.260197,4.402771,268
1995,Belgium,51.260197,4.402771,397
2000,Belgium,51.260197,4.402771,515
2005,Belgium,51.260197,4.402771,781

```

```

function render(year,cause) {
  map.remove();
  map = svg.append("g");
  map.style("stroke-width", trans["stroke-width"]);
  map.attr("transform", trans["transform"]);
  let data = nestD.find(n => +n.key === year);
  if(!data) return;
  data = data.values;

  if(cause) {
    data = data.filter(d=>d["location"] === cause);
  }
  for(let i = 0; i < data.length; i++) {
    map.append("circle")
      .attr("cx", projection([data[i].longitude, data[i].latitude])[0])
      .attr("cy", projection([data[i].longitude, data[i].latitude])[1])
      .attr("r", (Math.sqrt(data[i]["num_of_people"])/15)
      .style("fill", "red");
  }

  let causeArr = causeArr2.find(d => +d.time === year);
}
render(2017);

```

```
Year, location, latitude, longitude, num_of_people
1990,Australia,-28.86944,153.04453,2315
1995,Australia,-28.86944,153.04453,5562
2000,Australia,-28.86944,153.04453,8480
2005,Australia,-28.86944,153.04453,16130
2010,Australia,-28.86944,153.04453,29590
2015,Australia,-28.86944,153.04453,37092
2017,Australia,-28.86944,153.04453,38888
1990,Austria,48.210033,16.363449,1508
1995,Austria,48.210033,16.363449,1701
2000,Austria,48.210033,16.363449,1894
2005,Austria,48.210033,16.363449,2117
2010,Austria,48.210033,16.363449,2340
2015,Austria,48.210033,16.363449,2737
2017,Austria,48.210033,16.363449,3013
1990,Bahrain,26.156258,50.400505,20442
1995,Bahrain,26.156258,50.400505,24630
2000,Bahrain,26.156258,50.400505,28889
2005,Bahrain,26.156258,50.400505,49900
2010,Bahrain,26.156258,50.400505,81920
2015,Bahrain,26.156258,50.400505,78396
2017,Bahrain,26.156258,50.400505,80457
1990,Belgium,51.260197,4.402771,268
1995,Belgium,51.260197,4.402771,397
2000,Belgium,51.260197,4.402771,515
2005,Belgium,51.260197,4.402771,781
```

```
function render(year, cause) {
  map.remove();
  map = svg.append("g");
  map.style("stroke-width", trans["stroke-width"]);
  map.attr("transform", trans["transform"]);
  let data = nestD.find(n => +n.key === year);
  if(!data) return;
  data = data.values;

  if(cause) {
    data = data.filter(d=>d["location"] === cause);
  }
  for(let i = 0; i < data.length; i++) {
    map.append("circle")
      .attr("cx", projection([data[i].longitude, data[i].latitude])[0])
      .attr("cy", projection([data[i].longitude, data[i].latitude])[1])
      .attr("r", (Math.sqrt(data[i]["num_of_people"])/15)
      .style("fill", "red");
  }

  let causeArr = causeArr2.find(d => +d.time === year);
}
render(2017);
```



```
Year,location,latitude,longitude,num_of_people
1990,Australia,-28.86944,153.04453,2315
1995,Australia,-28.86944,153.04453,5562
2000,Australia,-28.86944,153.04453,8480
2005,Australia,-28.86944,153.04453,16130
2010,Australia,-28.86944,153.04453,29590
2015,Australia,-28.86944,153.04453,37092
2017,Australia,-28.86944,153.04453,38888
1990,Austria,48.210033,16.363449,1508
1995,Austria,48.210033,16.363449,1701
2000,Austria,48.210033,16.363449,1894
2005,Austria,48.210033,16.363449,2117
2010,Austria,48.210033,16.363449,2340
2015,Austria,48.210033,16.363449,2737
2017,Austria,48.210033,16.363449,3013
1990,Bahrain,26.156258,50.400505,20442
1995,Bahrain,26.156258,50.400505,24630
2000,Bahrain,26.156258,50.400505,28889
2005,Bahrain,26.156258,50.400505,49900
2010,Bahrain,26.156258,50.400505,81920
2015,Bahrain,26.156258,50.400505,78396
2017,Bahrain,26.156258,50.400505,80457
1990,Belgium,51.260197,4.402771,268
1995,Belgium,51.260197,4.402771,397
2000,Belgium,51.260197,4.402771,515
2005,Belgium,51.260197,4.402771,781
```

```
function render(year,cause) {
  map.remove();
  map = svg.append("g");
  map.style("stroke-width", trans["stroke-width"]);
  map.attr("transform", trans["transform"]);
  let data = nest0.find(n => +n.key === year);
  if(!data) return;
  data = data.values;

  if(cause) {
    data = data.filter(d=>d["location"] === cause);
  }
  for(let i = 0; i < data.length; i++) {
    map.append("circle")
      .attr("cx", projection([data[i].longitude, data[i].latitude])[0])
      .attr("cy", projection([data[i].longitude, data[i].latitude])[1])
      .attr("r", (Math.sqrt(data[i]["num_of_people"])/15))
      .style("fill", "red");
  }

  let causeArr = causeArr2.find(d => +d.time === year);
}
render(2017);
```

```
Year,location,latitude,longitude,num_of_people
1990,Australia,-28.86944,153.04453,2315
1995,Australia,-28.86944,153.04453,5562
2000,Australia,-28.86944,153.04453,8480
2005,Australia,-28.86944,153.04453,16130
2010,Australia,-28.86944,153.04453,29590
2015,Australia,-28.86944,153.04453,37092
2017,Australia,-28.86944,153.04453,38888
1990,Austria,48.210033,16.363449,1508
1995,Austria,48.210033,16.363449,1701
2000,Austria,48.210033,16.363449,1894
2005,Austria,48.210033,16.363449,2117
2010,Austria,48.210033,16.363449,2340
2015,Austria,48.210033,16.363449,2737
2017,Austria,48.210033,16.363449,3013
1990,Bahrain,26.156258,50.400505,20442
1995,Bahrain,26.156258,50.400505,24630
2000,Bahrain,26.156258,50.400505,28889
2005,Bahrain,26.156258,50.400505,49900
2010,Bahrain,26.156258,50.400505,81920
2015,Bahrain,26.156258,50.400505,78396
2017,Bahrain,26.156258,50.400505,80457
1990,Belgium,51.260197,4.402771,268
1995,Belgium,51.260197,4.402771,397
2000,Belgium,51.260197,4.402771,515
2005,Belgium,51.260197,4.402771,781
```

```
function render(year,cause) {
  map.remove();
  map = svg.append("g");
  map.style("stroke-width", trans["stroke-width"]);
  map.attr("transform", trans["transform"]);
  let data = nestD.find(n => +n.key === year);
  if(!data) return;
  data = data.values;

  if(cause) {
    data = data.filter(d => d["location"] === cause);
  }
  for(let i = 0; i < data.length; i++) {
    map.append("circle")
      .attr("cx", projection([data[i].longitude, data[i].latitude])[0])
      .attr("cy", projection([data[i].longitude, data[i].latitude])[1])
      .attr("r", (Math.sqrt(data[i]["num_of_people"])/15)
      .style("fill", "red");
  }

  let causeArr = causeArr2.find(d => +d.time === year);
}
render(2017);
```

3

Why did I build this?

Existential Crisis Time!



N. R. Narayana Murthy:

Engineering or technology is all about using the power of science to make life better for people, to reduce cost, to improve comfort, to improve productivity, etc.

“

4

Does it even matter?

diaspora

noun | di·as·po·ra | \ dī-'as-p(ə-)rə , dē- \

the movement, migration, or scattering of a people away from an established or ancestral homeland

or

any group migration or flight from a country or region.

Data is always collected with an agenda.

I wanted to make my father's memories of war concrete with some hard numbers.

**With great power comes
great responsibility.**



Thank you!

EmpireJs || @thathijabae