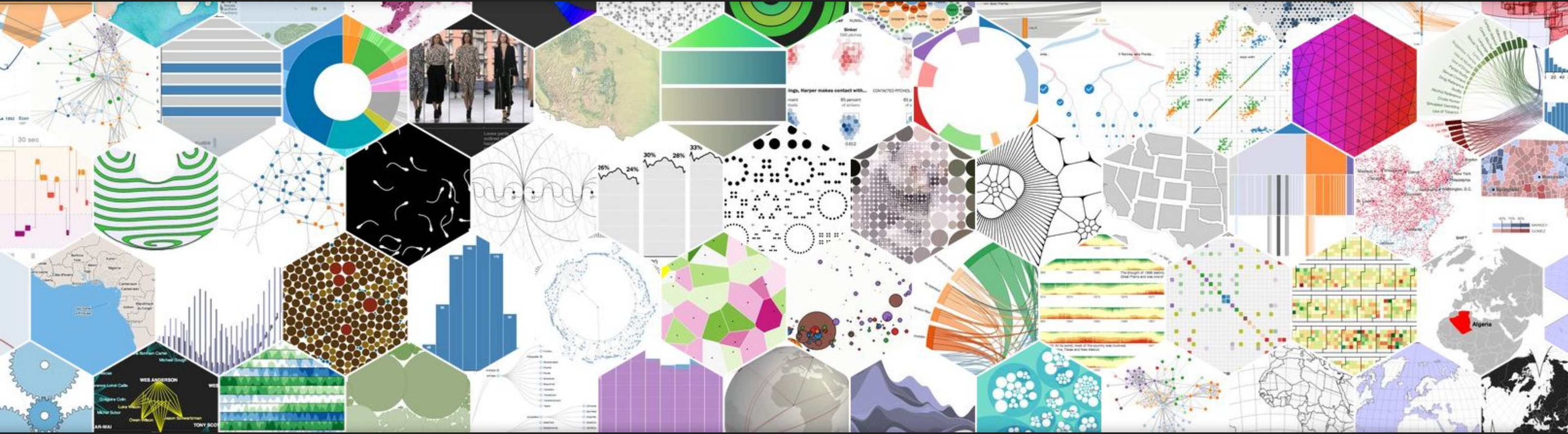
Mapping with D3.js

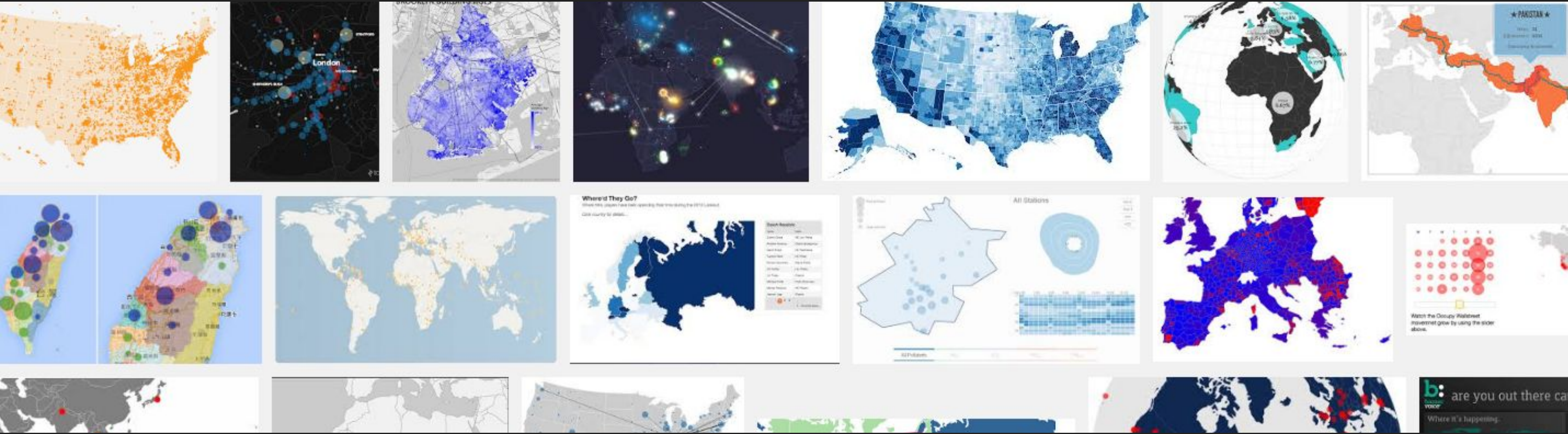
BerlinJS . June 16 2016 . @littlewebgiants

D3.js - Data Driven Documents



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

D3.js - Data Driven Documents



It has a lot of great tools for mapping data.

Let’s start with a basic webapp

I’ll be using the Yeoman (http://yeoman.io/) webapp generator to quickly scaffold this project.

Open Terminal

```
yo webapp
```

Moving on to the geodata

http://www.naturalearthdata.com/ has free vector and raster map data at 1:10m, 1:50m, and 1:110m scales.

We'll download the 1:110m Cultural Vectors set. This gives us a political map of the world's countries.

The download set includes files in the formats *.dbf, *.prj, *.shp and *.shx.

Inspect the shapefiles (*.shp) with QGIS

Free and open source geographic information system software that allows you to create, edit, visualise, analyse and publish geospatial information.
http://www.qgis.org/

// View the map

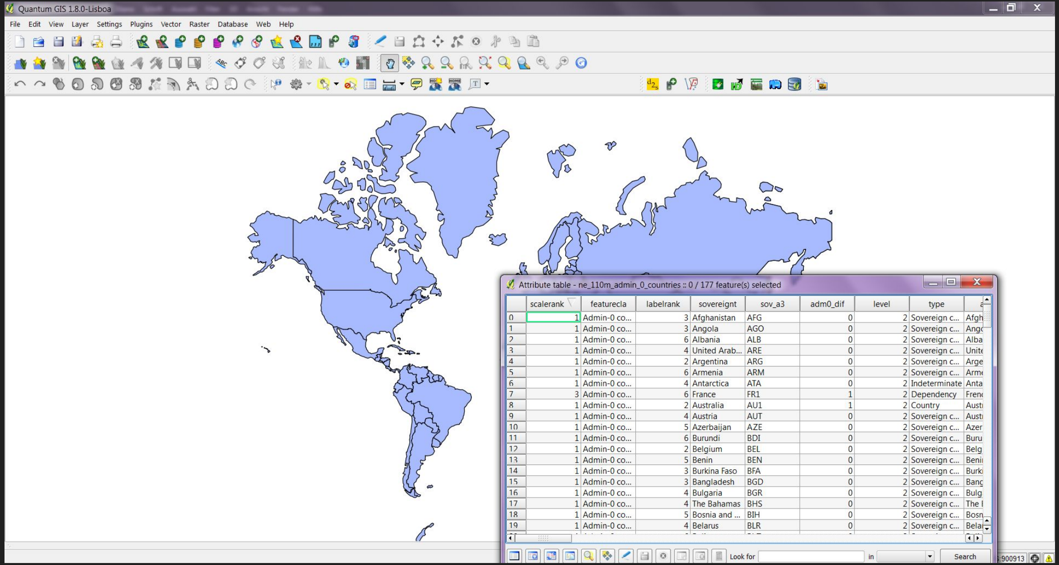
Open QGIS

Select Layer > Add Vector Layer

Open the shapefile

// Review the embedded data

Layer > Open Attribute Table



Attribute table - ne_110m_admin_0_countries :: 0 / 177 feature(s) selected

	scalerank	featurecla	labelrank	sovereignt	sov_a3	adm0_dif	level	type	a
0	1	Admin-0 co...	3	Afghanistan	AFG	0	2	Sovereign c...	Afgh
1	1	Admin-0 co...	3	Angola	AGO	0	2	Sovereign c...	Ango
2	1	Admin-0 co...	6	Albania	ALB	0	2	Sovereign c...	Alba
3	1	Admin-0 co...	4	United Arab...	ARE	0	2	Sovereign c...	Unite
4	1	Admin-0 co...	2	Argentina	ARG	0	2	Sovereign c...	Arge
5	1	Admin-0 co...	6	Armenia	ARM	0	2	Sovereign c...	Arme
6	1	Admin-0 co...	4	Antarctica	ATA	0	2	Indeterminate	Anta
7	3	Admin-0 co...	6	France	FR1	1	2	Dependency	Frenc
8	1	Admin-0 co...	2	Australia	AU1	1	2	Country	Austr
9	1	Admin-0 co...	4	Austria	AUT	0	2	Sovereign c...	Austr
10	1	Admin-0 co...	5	Azerbaijan	AZE	0	2	Sovereign c...	Azer
11	1	Admin-0 co...	6	Burundi	BDI	0	2	Sovereign c...	Buru
12	1	Admin-0 co...	2	Belgium	BEL	0	2	Sovereign c...	Belg
13	1	Admin-0 co...	5	Benin	BEN	0	2	Sovereign c...	Beni
14	1	Admin-0 co...	3	Burkina Faso	BFA	0	2	Sovereign c...	Burki
15	1	Admin-0 co...	3	Bangladesh	BGD	0	2	Sovereign c...	Bang
16	1	Admin-0 co...	4	Bulgaria	BGR	0	2	Sovereign c...	Bulg
17	1	Admin-0 co...	4	The Bahamas	BHS	0	2	Sovereign c...	The
18	1	Admin-0 co...	5	Bosnia and ...	BIH	0	2	Sovereign c...	Bosn
19	1	Admin-0 co...	4	Belarus	BLR	0	2	Sovereign c...	Belar

Option 1: Prepare geodata using QGIS

// Optional

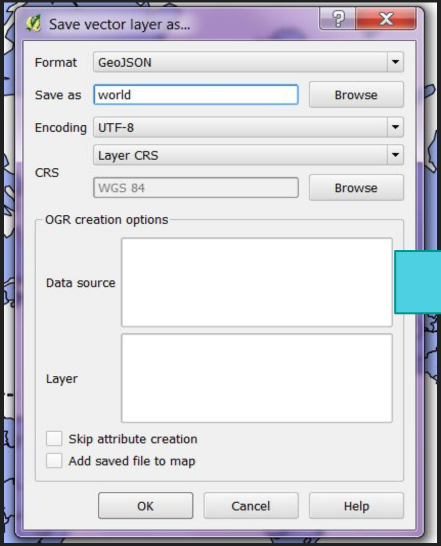
Edit the attribute table

// Export as GeoJSON

Layer > Save As

Select GeoJSON

Select file name and location



Format: GeoJSON

Save as: world

Encoding: UTF-8

CRS: Layer CRS

WGS 84

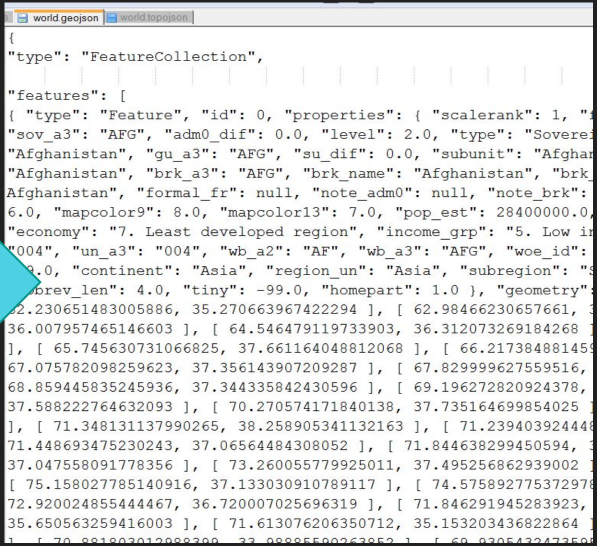
OGR creation options

Data source

Layer

☐ Skip attribute creation

☐ Add saved file to map



```
{
"type": "FeatureCollection",

"features": [
{ "type": "Feature", "id": 0, "properties": { "scalerank": 1, "
"sov_a3": "AFG", "adm0_dif": 0.0, "level": 2.0, "type": "Sovere
"Afghanistan", "gu_a3": "AFG", "su_dif": 0.0, "subunit": "Afgha
"Afghanistan", "brk_a3": "AFG", "brk_name": "Afghanistan", "brk_
Afghanistan", "formal_fr": null, "note_adm0": null, "note_brk":
6.0, "mapcolor9": 8.0, "mapcolor13": 7.0, "pop_est": 28400000.0,
"economy": "7. Least developed region", "income_grp": "5. Low i
"004", "un_a3": "004", "wb_a2": "AF", "wb_a3": "AFG", "woe_id":
.0, "continent": "Asia", "region_un": "Asia", "subregion": "
brev_len": 4.0, "tiny": -99.0, "homepart": 1.0 }, "geometry":
2.230651483005886, 35.270663967422294 ], [ 62.98466230657661,
36.00795746514660З ], [ 64.546479119733903, 36.31207326918426
], [ 65.745630731066825, 37.661164048812068 ], [ 66.21738488145
67.075782098259623, 37.356143907209287 ], [ 67.829999627559516,
68.859445835245936, 37.344335842430596 ], [ 69.196272820924378,
37.58822276463209З ], [ 70.270574171840138, 37.735164699854025
], [ 71.348131137990265, 38.258905341132163 ], [ 71.23940392444
71.448693475230243, 37.06564484308052 ], [ 71.844638299450594,
37.047558091778356 ], [ 73.260055779925011, 37.495256862939002
[ 75.158027785140916, 37.133030910789117 ], [ 74.57589277537297
72.920024855444467, 36.720007025696319 ], [ 71.846291945283923,
35.650563259416003 ], [ 71.613076206350712, 35.153203436822864
```

Option 2: Prepare geodata using TopoJSON

GeoJSON is a useful interchange format for geographical data and is widely used.

However it requires a lot of redundant information. In a world map data set, every country’s complete border is stored as geometry. This means that borders shared between countries are stored multiple times.

The TopoJSON format stores geometries as a set of arcs that do not overlap. In some cases this can reduce file sizes by a factor of 10.

TopoJSON files can be created using a command line tool from Mike Bostock, the creator of D3.js. See https://github.com/mbostock/topojson

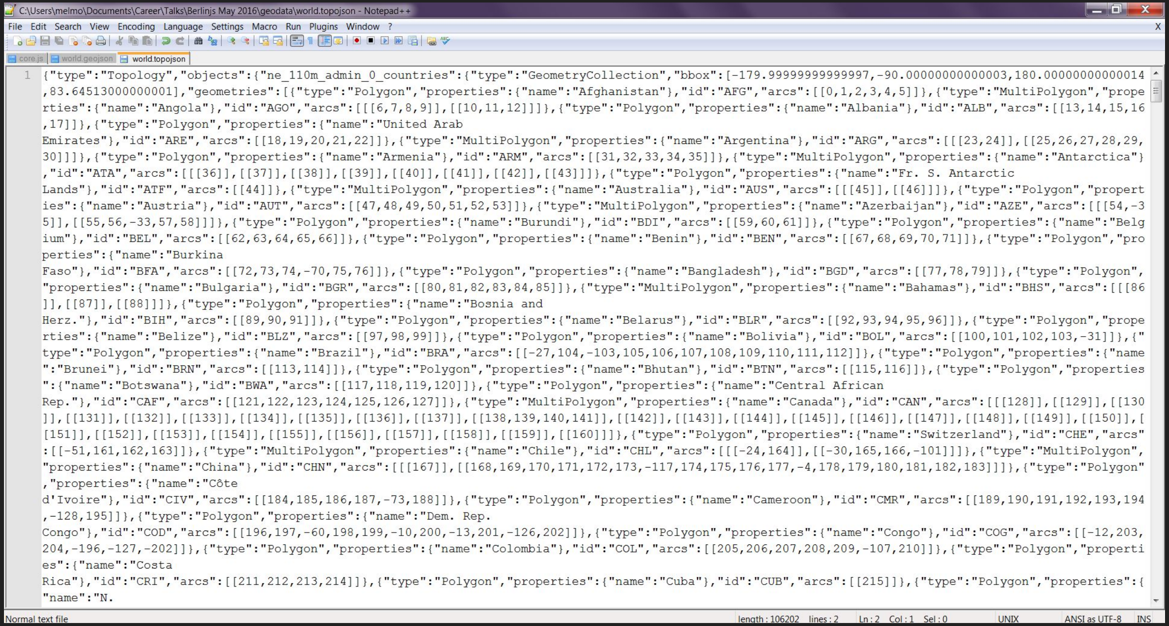
Option 2: Prepare geodata using TopoJSON

Open Terminal and navigate to the folder with your shapefiles.

```
topojson   -o world.topojson // output file
           --quantization 1e5
           --id-property iso_a3 // we’ll use 3-digit country codes as IDs
           --properties name=name  // properties to keep (see attribute table in QGIS)
           --io=countries
           --oo=land
           --no-key
           -- ne_110m_admin_0_countries.shp // input file
```

Option 2: Prepare geodata using TopoJSON

Our TopoJSON file is 85% smaller than the GeoJSON file (103KB vs 672KB)


C:\Users\melmo\Documents\Career\Talks\Berlinjs May 2016\geodata\world.topojson - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
X
core.js
world.geojson
world.topojson
Normal text file
length : 106202 lines : 2
Ln : 2 Col : 1 Sel : 0
UNIX
ANSI as UTF-8
INS

```
{"type":"Topology","objects":{"ne_110m_admin_0_countries":{"type":"GeometryCollection","bbox":[-179.99999999999997,-90.00000000000003,180.00000000000014
,83.64513000000001],"geometries":[{"type":"Polygon","properties":{"name":"Afghanistan"},"id":"AFG","arcs":[[0,1,2,3,4,5]]},{"type":"MultiPolygon","prope
rties":{"name":"Angola"},"id":"AGO","arcs":[[[6,7,8,9]],[[10,11,12]]]},{"type":"Polygon","properties":{"name":"Albania"},"id":"ALB","arcs":[[13,14,15,16
,17]]},{"type":"Polygon","properties":{"name":"United Arab
Emirates"},"id":"ARE","arcs":[[18,19,20,21,22]]},{"type":"MultiPolygon","properties":{"name":"Argentina"},"id":"ARG","arcs":[[[23,24]],[[25,26,27,28,29,
30]]]},{"type":"Polygon","properties":{"name":"Armenia"},"id":"ARM","arcs":[[31,32,33,34,35]]},{"type":"MultiPolygon","properties":{"name":"Antarctica"}
,"id":"ATA","arcs":[[[36]],[[37]],[[38]],[[39]],[[40]],[[41]],[[42]],[[43]]]},{"type":"Polygon","properties":{"name":"Fr. S. Antarctic
Lands"},"id":"ATF","arcs":[[44]]},{"type":"MultiPolygon","properties":{"name":"Australia"},"id":"AUS","arcs":[[[45]],[[46]]]},{"type":"Polygon","propert
ies":{"name":"Austria"},"id":"AUT","arcs":[[47,48,49,50,51,52,53]]},{"type":"MultiPolygon","properties":{"name":"Azerbaijan"},"id":"AZE","arcs":[[[54,-3
5]],[[55,56,-33,57,58]]]},{"type":"Polygon","properties":{"name":"Burundi"},"id":"BDI","arcs":[[59,60,61]]},{"type":"Polygon","properties":{"name":"Belg
ium"},"id":"BEL","arcs":[[62,63,64,65,66]]},{"type":"Polygon","properties":{"name":"Benin"},"id":"BEN","arcs":[[67,68,69,70,71]]},{"type":"Polygon","pro
perties":{"name":"Burkina
Faso"},"id":"BFA","arcs":[[72,73,74,-70,75,76]]},{"type":"Polygon","properties":{"name":"Bangladesh"},"id":"BGD","arcs":[[77,78,79]]},{"type":"Polygon",
"properties":{"name":"Bulgaria"},"id":"BGR","arcs":[[80,81,82,83,84,85]]},{"type":"MultiPolygon","properties":{"name":"Bahamas"},"id":"BHS","arcs":[[[86
]],[[87]],[[88]]]},{"type":"Polygon","properties":{"name":"Bosnia and
Herz."},"id":"BIH","arcs":[[89,90,91]]},{"type":"Polygon","properties":{"name":"Belarus"},"id":"BLR","arcs":[[92,93,94,95,96]]},{"type":"Polygon","prope
rties":{"name":"Belize"},"id":"BLZ","arcs":[[97,98,99]]},{"type":"Polygon","properties":{"name":"Bolivia"},"id":"BOL","arcs":[[100,101,102,103,-31]]},{"
type":"Polygon","properties":{"name":"Brazil"},"id":"BRA","arcs":[[-27,104,-103,105,106,107,108,109,110,111,112]]},{"type":"Polygon","properties":{"name
":"Brunei"},"id":"BRN","arcs":[[113,114]]},{"type":"Polygon","properties":{"name":"Bhutan"},"id":"BTN","arcs":[[115,116]]},{"type":"Polygon","properties
":{"name":"Botswana"},"id":"BWA","arcs":[[117,118,119,120]]},{"type":"Polygon","properties":{"name":"Central African
Rep."},"id":"CAF","arcs":[[121,122,123,124,125,126,127]]},{"type":"MultiPolygon","properties":{"name":"Canada"},"id":"CAN","arcs":[[[128]],[[129]],[[130
]],[[131]],[[132]],[[133]],[[134]],[[135]],[[136]],[[137]],[[138,139,140,141]],[[142]],[[143]],[[144]],[[145]],[[146]],[[147]],[[148]],[[149]],[[150]],[
[151]],[[152]],[[153]],[[154]],[[155]],[[156]],[[157]],[[158]],[[159]],[[160]]]},{"type":"Polygon","properties":{"name":"Switzerland"},"id":"CHE","arcs"
:[[-51,161,162,163]]},{"type":"MultiPolygon","properties":{"name":"Chile"},"id":"CHL","arcs":[[[-24,164]],[[-30,165,166,-101]]]},{"type":"MultiPolygon",
"properties":{"name":"China"},"id":"CHN","arcs":[[[167]],[[168,169,170,171,172,173,-117,174,175,176,177,-4,178,179,180,181,182,183]]]},{"type":"Polygon"
,"properties":{"name":"Côte
d'Ivoire"},"id":"CIV","arcs":[[184,185,186,187,-73,188]]},{"type":"Polygon","properties":{"name":"Cameroon"},"id":"CMR","arcs":[[189,190,191,192,193,194
,-128,195]]},{"type":"Polygon","properties":{"name":"Dem. Rep.
Congo"},"id":"COD","arcs":[[196,197,-60,198,199,-10,200,-13,201,-126,202]]},{"type":"Polygon","properties":{"name":"Congo"},"id":"COG","arcs":[[-12,203,
204,-196,-127,-202]]},{"type":"Polygon","properties":{"name":"Colombia"},"id":"COL","arcs":[[205,206,207,208,209,-107,210]]},{"type":"Polygon","properti
es":{"name":"Costa
Rica"},"id":"CRI","arcs":[[211,212,213,214]]},{"type":"Polygon","properties":{"name":"Cuba"},"id":"CUB","arcs":[[215]]},{"type":"Polygon","properties":{
"name":"N.
```

Let’s check in on the webapp

Install the Javascript libraries we’ll need for this tutorial.

```
bower install d3
bower install topojson
bower install d3-queue
```

Add the scripts to index.html

```
<script src="/bower_components/d3/d3.min.js"></script>
<script src="/bower_components/d3-queue/d3-queue.js"></script>
<script src="/bower_components/topojson/topojson.min.js"></script>
```

Build the HTML structure

Clear out the dummy content and add the following to index.html

```
<div class="row">
    <div class="col-md-8">
        <div id="map"></div>
    </div>
    <div class="col-md-4">
        <div id="info">
            <h3>Click on a country to explore</h3>
        </div>
    </div>
</div>
```

Let’s Make a Map!

Add the TopoJSON file into the webapp project under a new folder called data.

Let’s Make a Map!

Build the map visualisation in main.js.

```
// Define the map size
var width = 960,
    height = 500;

// Add an svg element that will be the parent of our data viz
var svg = d3.select("#map").append("svg")
    .attr("width", width)
    .attr("height", height);
```

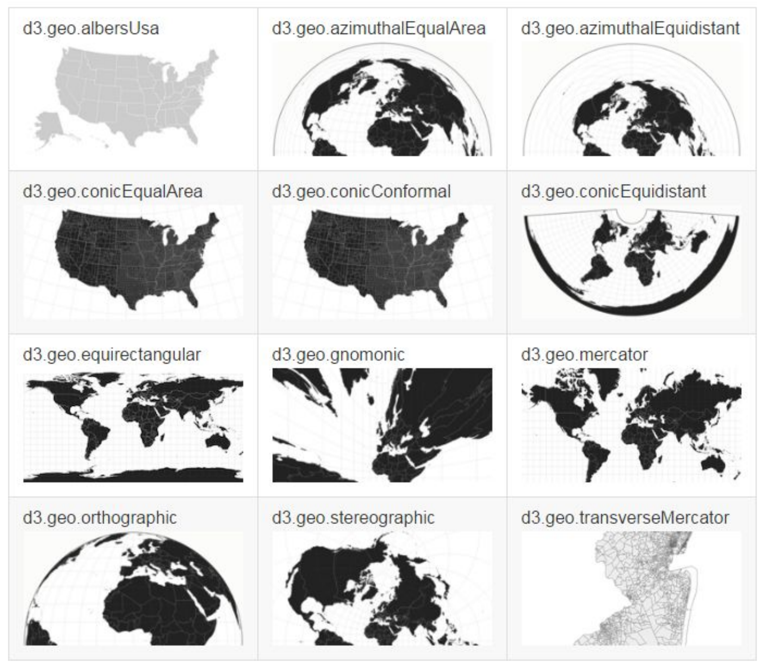
Let’s Make a Map!

Define how our map will interpret the geodata.

```
// Define the map projection, scale and position
var projection = d3.geo.mercator()
    .scale(120)
    .translate([width / 2, height / 2]);

// Create a path function that will plot the geodata according to the projection
var path = d3.geo.path()
    .projection(projection);
```

Let’s Make a Map!

Learn more about projections and how you can use in them in D3.js at https://github.com/d3/d3/wiki/Geo-Projections



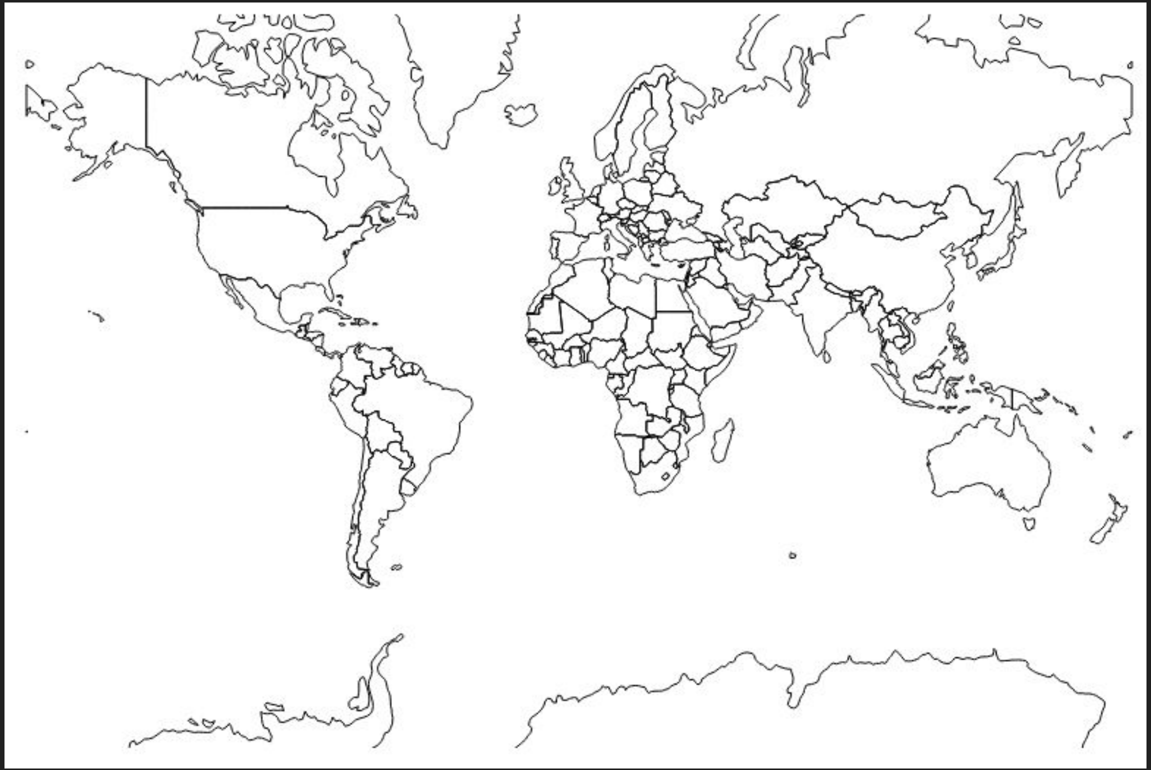
Let’s Make a Map!

```
// Load the geodata
d3.json("data/world.topojson", function(error, world) {
    if (error) return console.error(error);
    console.log(world);
});
```

Let’s Make a Map!

```
d3.json("data/world.topojson", function(error, world) {
    ...
    var countries = topojson.feature(world,
world.objects.ne_110m_admin_0_countries); // get the geometries

    svg.append('path') // add svg path elements
        .datum(countries) // bind geometry data
        .attr('d', path) // use path function to plot points
        .attr('stroke','black')
        .attr('fill','white');
});
```

Let’s Make a Map!



Time to find some data

We’ll be using data on participation in the workforce and wages by gender.

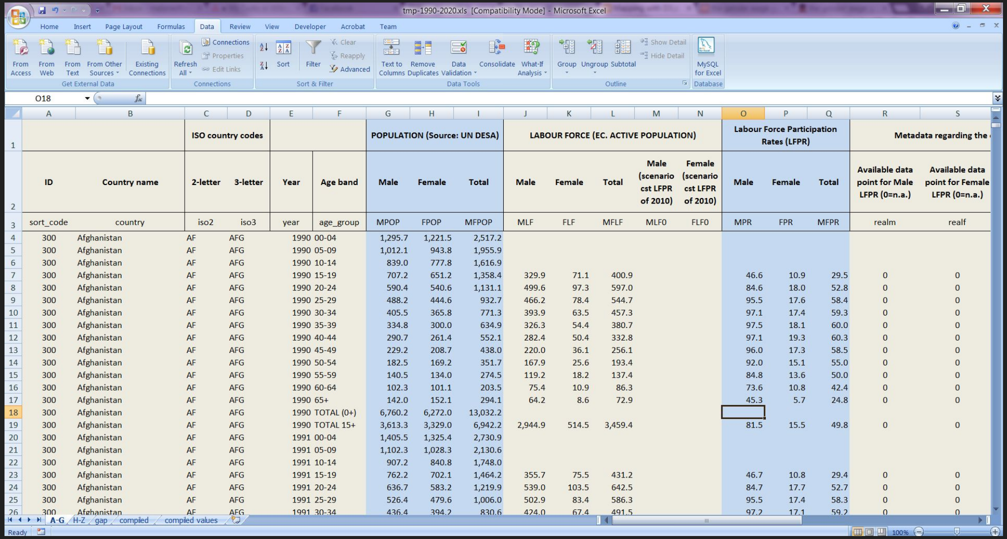
Data on workforce participation is available from the ILO at http://laborsta.ilo.org/applv8/data/EAPEP/eapep_E.html

Data on the gender wage gap is available from UNECE at http://w3.unece.org/PXWeb2015/pxweb/en/STAT/STAT__30-GE__03-WorkAndeconomy

Cleaning up data

The downloaded data sheets contain lots of excess data and fancy formatting that we don't need. Before we can use the data, we need to:

- Remove unnecessary data
- Combine the two data sets
- Remove formatting
- Save the data in *.csv format



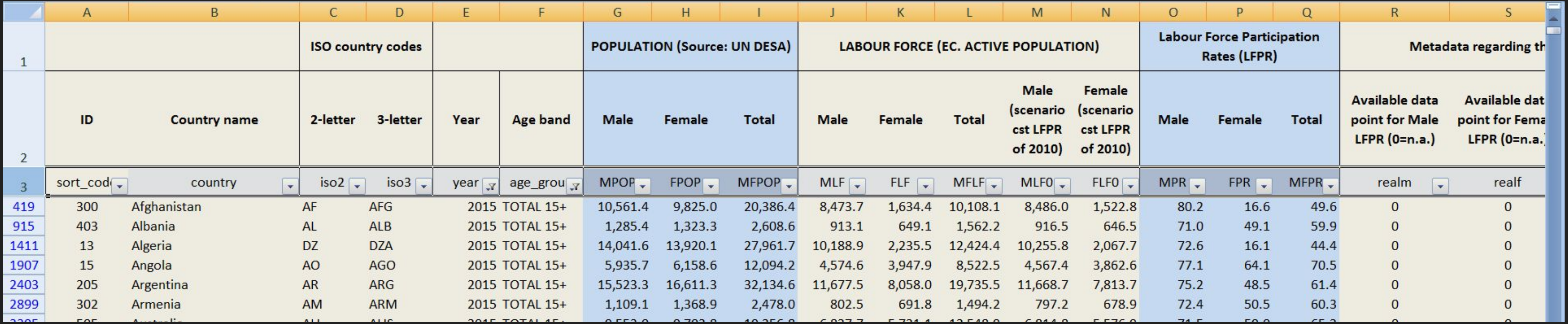
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1			ISO country codes				POPULATION (Source: UN DESA)			LABOUR FORCE (EC. ACTIVE POPULATION)					Labour Force Participation Rates (LFPR)			Metadata regarding the	
2	ID	Country name	2-letter	3-letter	Year	Age band	Male	Female	Total	Male	Female	Total	Male (scenario cst LFPR of 2010)	Female (scenario cst LFPR of 2010)	Male	Female	Total	Available data point for Male LFPR (0=n.a.)	Available data point for Female LFPR (0=n.a.)
3	sort_code	country	iso2	iso3	year	age_group	MPOP	FPOP	MFPOP	MLF	FLF	MFLF	MLF0	FLF0	MPR	FPR	MFPR	realm	realf
4	300	Afghanistan	AF	AFG	1990	00-04	1,295.7	1,221.5	2,517.2										
5	300	Afghanistan	AF	AFG	1990	05-09	1,012.1	943.8	1,955.9										
6	300	Afghanistan	AF	AFG	1990	10-14	839.0	777.8	1,616.9										
7	300	Afghanistan	AF	AFG	1990	15-19	707.2	651.2	1,358.4	329.9	71.1	400.9			46.6	10.9	29.5	0	0
8	300	Afghanistan	AF	AFG	1990	20-24	590.4	540.6	1,131.1	499.6	97.3	597.0			84.6	18.0	52.8	0	0
9	300	Afghanistan	AF	AFG	1990	25-29	488.2	444.6	932.7	466.2	78.4	544.7			95.5	17.6	58.4	0	0
10	300	Afghanistan	AF	AFG	1990	30-34	405.5	365.8	771.3	393.9	63.5	457.3			97.1	17.4	59.3	0	0
11	300	Afghanistan	AF	AFG	1990	35-39	334.8	300.0	634.9	326.3	54.4	380.7			97.5	18.1	60.0	0	0
12	300	Afghanistan	AF	AFG	1990	40-44	290.7	261.4	552.1	282.4	50.4	332.8			97.1	19.3	60.3	0	0
13	300	Afghanistan	AF	AFG	1990	45-49	229.2	208.7	438.0	220.0	36.1	256.1			96.0	17.3	58.5	0	0
14	300	Afghanistan	AF	AFG	1990	50-54	182.5	169.2	351.7	167.9	25.6	193.4			92.0	15.1	55.0	0	0
15	300	Afghanistan	AF	AFG	1990	55-59	140.5	134.0	274.5	119.2	18.2	137.4			84.8	13.6	50.0	0	0
16	300	Afghanistan	AF	AFG	1990	60-64	102.3	101.1	203.5	75.4	10.9	86.3			73.6	10.8	42.4	0	0
17	300	Afghanistan	AF	AFG	1990	65+	142.0	152.1	294.1	64.2	8.6	72.9			45.3	5.7	24.8	0	0
18	300	Afghanistan	AF	AFG	1990	TOTAL (0+)	6,760.2	6,272.0	13,032.2										
19	300	Afghanistan	AF	AFG	1990	TOTAL 15+	3,613.3	3,329.0	6,942.2	2,944.9	514.5	3,459.4			81.5	15.5	49.8	0	0
20	300	Afghanistan	AF	AFG	1991	00-04	1,405.5	1,325.4	2,730.9										
21	300	Afghanistan	AF	AFG	1991	05-09	1,102.3	1,028.3	2,130.6										
22	300	Afghanistan	AF	AFG	1991	10-14	907.2	840.8	1,748.0										
23	300	Afghanistan	AF	AFG	1991	15-19	762.2	702.1	1,464.2	355.7	75.5	431.2			46.7	10.8	29.4	0	0
24	300	Afghanistan	AF	AFG	1991	20-24	636.7	583.2	1,219.9	539.0	103.5	642.5			84.7	17.7	52.7	0	0
25	300	Afghanistan	AF	AFG	1991	25-29	526.4	479.6	1,006.0	502.9	83.4	586.3			95.5	17.4	58.3	0	0
26	300	Afghanistan	AF	AFG	1991	30-34	436.4	394.2	830.6	424.0	67.4	491.5			97.2	17.1	59.2	0	0

A-G / H-Z / gap / compiled / compiled values

Step 1: Remove unneeded data

Start with the data on workplace participation. This contains data for a number of years and age ranges. We're only going to use the data from 2015, and the combined figure for all age groups over 15.

Open the data in Excel and add filters to the third row. Filter the age and year columns so that we have only one figure for each country.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
1			ISO country codes				POPULATION (Source: UN DESA)			LABOUR FORCE (EC. ACTIVE POPULATION)					Labour Force Participation Rates (LFPR)			Metadata regarding th	
2	ID	Country name	2-letter	3-letter	Year	Age band	Male	Female	Total	Male	Female	Total	Male (scenario cst LFPR of 2010)	Female (scenario cst LFPR of 2010)	Male	Female	Total	Available data point for Male LFPR (0=n.a.)	Available dat point for Fema LFPR (0=n.a.
3	sort_cod	country	iso2	iso3	year	age_grou	MPOP	FPOP	MFPOP	MLF	FLF	MFLF	MLF0	FLF0	MPR	FPR	MFPR	realm	realf
419	300	Afghanistan	AF	AFG	2015	TOTAL 15+	10,561.4	9,825.0	20,386.4	8,473.7	1,634.4	10,108.1	8,486.0	1,522.8	80.2	16.6	49.6	0	0
915	403	Albania	AL	ALB	2015	TOTAL 15+	1,285.4	1,323.3	2,608.6	913.1	649.1	1,562.2	916.5	646.5	71.0	49.1	59.9	0	0
1411	13	Algeria	DZ	DZA	2015	TOTAL 15+	14,041.6	13,920.1	27,961.7	10,188.9	2,235.5	12,424.4	10,255.8	2,067.7	72.6	16.1	44.4	0	0
1907	15	Angola	AO	AGO	2015	TOTAL 15+	5,935.7	6,158.6	12,094.2	4,574.6	3,947.9	8,522.5	4,567.4	3,862.6	77.1	64.1	70.5	0	0
2403	205	Argentina	AR	ARG	2015	TOTAL 15+	15,523.3	16,611.3	32,134.6	11,677.5	8,058.0	19,735.5	11,668.7	7,813.7	75.2	48.5	61.4	0	0
2899	302	Armenia	AM	ARM	2015	TOTAL 15+	1,109.1	1,368.9	2,478.0	802.5	691.8	1,494.2	797.2	678.9	72.4	50.5	60.3	0	0

Step 1: Remove unneeded data

Select only the visible cells in the sheet and copy them with these shortcuts.

```
ctrl a // Select all
alt ;  // Select visible
ctrl c // Copy selected
ctrl v // Paste selected
```

Repeat for the second sheet of data, then delete unneeded columns. We’ll keep the country name, ISO code, and the male and female participation rates as percentages.

Step 2: Integrate the two data sets

Open the second data set in Excel. Copy the contents into a new sheet on the first Excel file.

To combine the two data sets, we will use a combination of Excel’s INDEX and MATCH functions to look up each country name in the gender gap data and then copy across the value into our compiled data set.

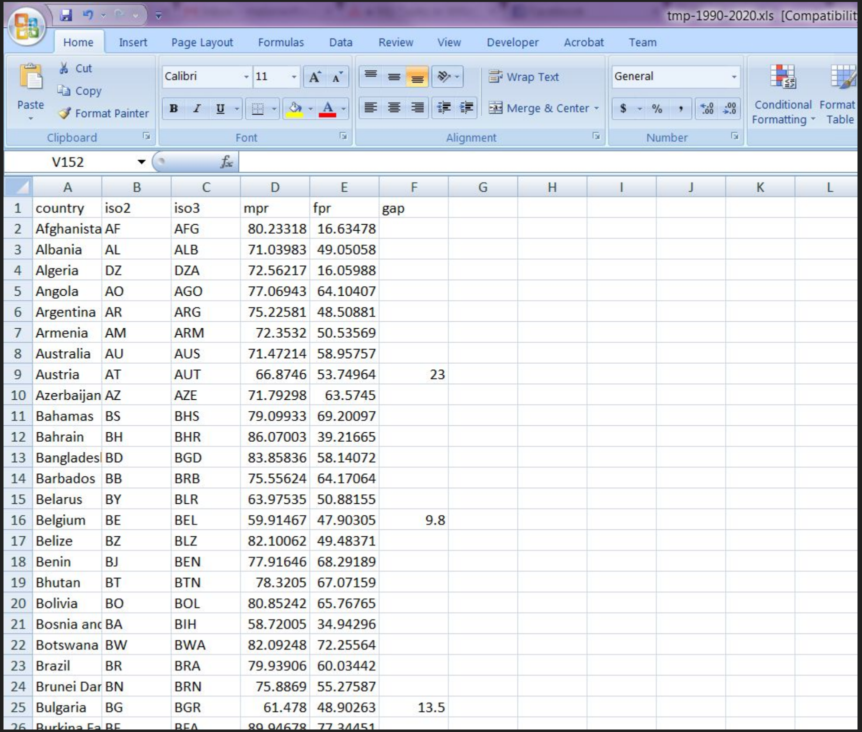
```
=IFERROR(  // Excel throws an error if no matching values are found
    INDEX( gap!C$52:C$99, // Return the value for column C & found row number
        MATCH(A2,gap!B$52:B$99,0)  // Find exact match in & return row number
    ),
"..")
```

Step 3: Clean up formatting

Select all content from our compiled sheet and do Paste Special > Paste Values into a new sheet.

Clean up any last details (such as the cells with "..") and clear formatting with Clear > Clear Formats.

Save the results to a csv file and put it into the data folder in your webapp project.



	A	B	C	D	E	F
1	country	iso2	iso3	mpr	fpr	gap
2	Afghanista	AF	AFG	80.23318	16.63478	
3	Albania	AL	ALB	71.03983	49.05058	
4	Algeria	DZ	DZA	72.56217	16.05988	
5	Angola	AO	AGO	77.06943	64.10407	
6	Argentina	AR	ARG	75.22581	48.50881	
7	Armenia	AM	ARM	72.3532	50.53569	
8	Australia	AU	AUS	71.47214	58.95757	
9	Austria	AT	AUT	66.8746	53.74964	23
10	Azerbaijan	AZ	AZE	71.79298	63.5745	
11	Bahamas	BS	BHS	79.09933	69.20097	
12	Bahrain	BH	BHR	86.07003	39.21665	
13	Banglades	BD	BGD	83.85836	58.14072	
14	Barbados	BB	BRB	75.55624	64.17064	
15	Belarus	BY	BLR	63.97535	50.88155	
16	Belgium	BE	BEL	59.91467	47.90305	9.8
17	Belize	BZ	BLZ	82.10062	49.48371	
18	Benin	BJ	BEN	77.91646	68.29189	
19	Bhutan	BT	BTN	78.3205	67.07159	
20	Bolivia	BO	BOL	80.85242	65.76765	
21	Bosnia anc	BA	BIH	58.72005	34.94296	
22	Botswana	BW	BWA	82.09248	72.25564	
23	Brazil	BR	BRA	79.93906	60.03442	
24	Brunei Dar	BN	BRN	75.8869	55.27587	
25	Bulgaria	BG	BGR	61.478	48.90263	13.5
26	Burkina Fa	BF	BFA	89.94678	77.34451	

Putting it all together

In the final steps, we combine the workplace data with the geodata and use it to make a choropleth (colour coded) map.

Putting it all together - namespacing

```
(function( map ) { // Wrap everything up in a function for namespacing

    var go = function(error, world, data) {}; // Private draw function
    map.init = function() {};  // Publicly accessible init function

}( window.map = window.map || {} ));

map.init(); // Let’s do it!
```

Putting it all together - set up variables

```
(function( map ) {

    // Data storage and processing
    var world = {},
    data = {},
    queue = d3_queue.queue,  // Control data loading
    countryByIso = d3.map();  // Will let us access country data by ISO code
    ...

}( window.map = window.map || {} ));
```

Putting it all together - set up variables

```
(function( map ) {
    ...
    // Map size
    var width = 600,
        height = 500;
    ...
}( window.map = window.map || {} ));
```

Putting it all together - set up variables

```
(function( map ) {
    ...
    // Map settings
    var svg = d3.select("#map").append("svg")
        .attr("width", width)
        .attr("height", height),
    projection = d3.geo.mercator()
        .scale(390)
        .translate([width * .5, height * 1.45]),
    path = d3.geo.path()
        .projection(projection);
    ...
}( window.map = window.map || {} ));
```

Putting it all together - set up variables

```
(function( map ) {
    ...
    // Map colour scale
    var minGap = 3,  // smallest wage gap is 3.2
    maxGap = 30,  // smallest wage gap is 29
    minGapColor = "#bcbddc",    // light blue
    maxGapColor = "#990000",  // red
    gapColor = d3.scale.linear().domain([minGap,
maxGap]).range([minGapColor, maxGapColor]);  // map range of values to RGB
    ...
}( window.map = window.map || {} ));
```

Putting it all together - loading the data

```
...
map.init = function() { // Fill out the init function
    queue() // Wait until both data files are loaded
        .defer(d3.json, "data/world.topojson")
        .defer(d3.csv, "data/gender_gap.csv", function(d) {
            // Row walker function maps data to country ISO code
            countryByIso.set(d.iso3, d); return d;
        })
        .await(go); // Call go() when both files are loaded
};
...
```

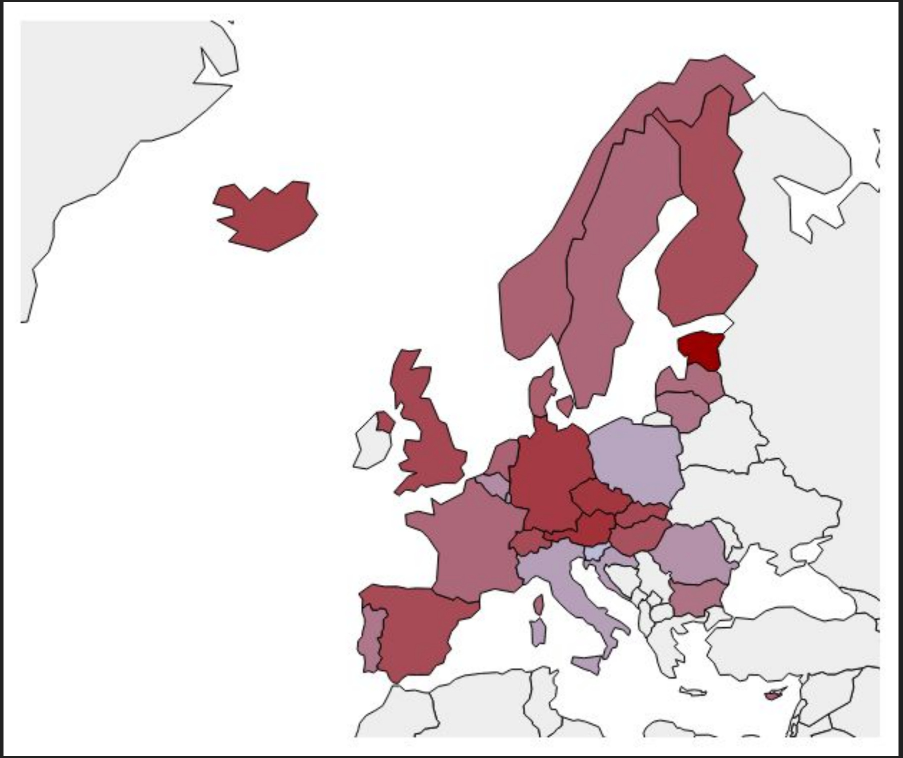
Putting it all together - draw the map

```
…
var go = function(error, world, data) {
       console.log(error);
       console.log(world);
       console.log(data);

       var countries = topojson.feature(world,
world.objects.ne_110m_admin_0_countries);

};
…
```

Putting it all together - draw the map

```
svg.selectAll("path") // Select path
    .data(countries.features) // Bind geodata
        .enter().append("path") // Add svg path for each country
            .attr("fill", function(d) { // Fill colour determined by wage gap data
                var country = countryByIso.get(d.id); // Use map to get country data
                if (typeof country !== 'undefined' && country.gap) {
                    return gapColor(country.gap); // Pass gap data to colour scale
                }
                return "#eee"; // If no data then set fill to white
            })
            // Add “country” class to use to bind click functions
            .attr("class",function(d) { return d.id + " country"})
            .attr("stroke",'#000')
            .attr("d", path); // Send geometry data to path function to plot points
```

Putting it all together



Sneaky bonus: adding click events

```
// Create a scale to show participation rates as a bar chart
var xMax = 320,
xScale = d3.scale.linear()
    .domain([0, 100])
    .range([0, xMax]);

...
```

Sneaky bonus: adding click events

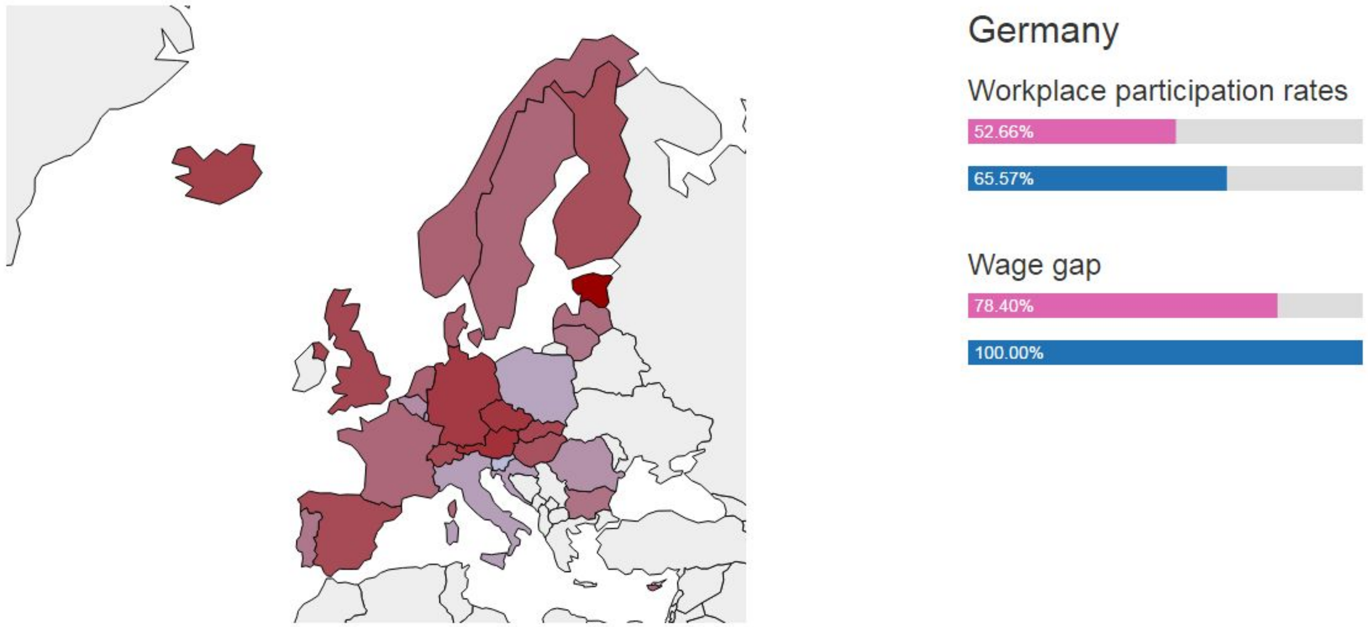
```
d3.selectAll("path.country")
    .on("click",function(d) {
        var country = countryByIso.get(d.id);  // Get country data from map function
        if (typeof country !== 'undefined') {  // Check we have data to use
                d3.select('#info').html('');  // Empty old data from the info page
                var selection = d3.select('#info')
                    .append("svg")  // add a chart
                        .attr("width", xMax)
                        .attr("height", 80)
                            .append('g')
                                .selectAll("rect")  // add svg rect for male & female
                                .data([country.fpr,country.mpr])
                                .enter();  // store enter selection to add multiple elem
...
```

Sneaky bonus: adding click events

```
...
                selection.insert("rect") // add a coloured bar for each gender
                    .attr("y", function(d, i) { return i * 38; })
                    .attr("x", 0)
                    .attr("height", 20)
                    .style('fill', function(d, i) {
                        if (i < 1) return "rgb(223, 101, 176)";
                            return "rgb(33, 113, 181)";
                        })
                    .attr("width", xScale); // width determined by data point
...
```

Sneaky bonus: adding click events

```
...
                     selection.insert("text")  // text with percentage figures
                         .attr("y", function(d, i) { return i * 38 + 15; })
                         .attr("x", 5)
                         .attr("height", 20)
                         .attr("width", xMax)
                         .attr("fill","white")
                         .style("color","white")
                         .text(function(d, i) {
                             // show to two decimal points
                             return parseFloat(d).toFixed(2) + '%';
                         });
        }
    });
```



Thanks for listening :)

Find the demo online at http://melmo.github.io/d3-map-demo/