

MAP CLUB

Session 09 — D3.js workshop (Part II)
December 2, 2016

Emily Fuhrman
@xxzvx emilyfuhrman.co

[https://github.com/
emilyfuhrman/map-club/tree/
master/2016_Fall/Session_09](https://github.com/emilyfuhrman/map-club/tree/master/2016_Fall/Session_09)

Mapping New York City art galleries

Data

GeoJSON encodes
geographic data structures.

Its syntax is derived from
JavaScript syntax.

```
{
  "type": "FeatureCollection",
  "crs": {
    "type": "name",
    "properties": {
      "name": "urn:ogc:def:crs:OGC:1.3:CRS84"
    }
  },

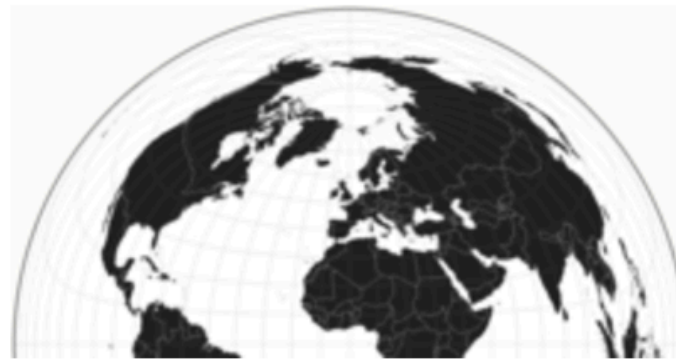
  "features": [
    { "type": "Feature", "id": 0, "properties": { "BoroCode": 5,
      "BoroName": "Staten Island", "Shape_Leng": 330385.03697, "Shape_Area":
      1623853249.910000085830688 }, "geometry": { "type": "MultiPolygon",
      "coordinates": [ [ [ [ -74.050508064032471, 40.566422034160816 ], [ -74.
      .050471371285454, 40.566417387938124 ], [ -74.050303212717623, 40.
      566417461703153 ], [ -74.050125873754567, 40.566405880335097 ], [ -74.
      049983525625748, 40.566395924928273 ], [ -74.049316403620878, 40.
      565887747780437 ], [ -74.049236298420453, 40.565362736368101 ], [ -74.
```

Projection

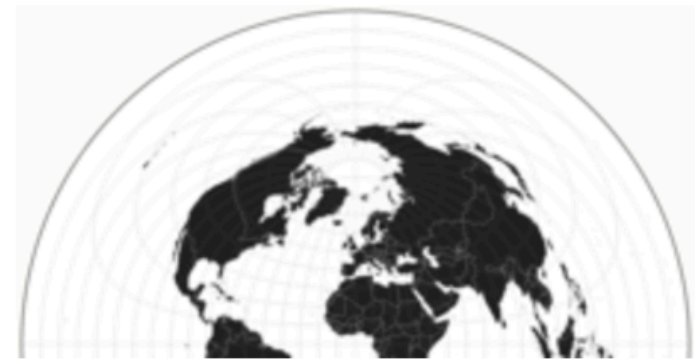
d3.geo.albersUsa



d3.geo.azimuthalEqualArea



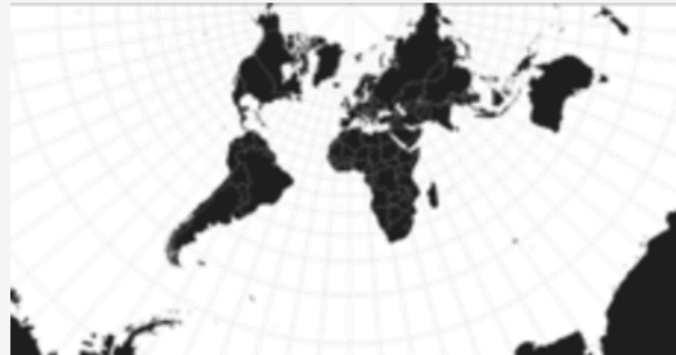
d3.geo.azimuthalEquidistant



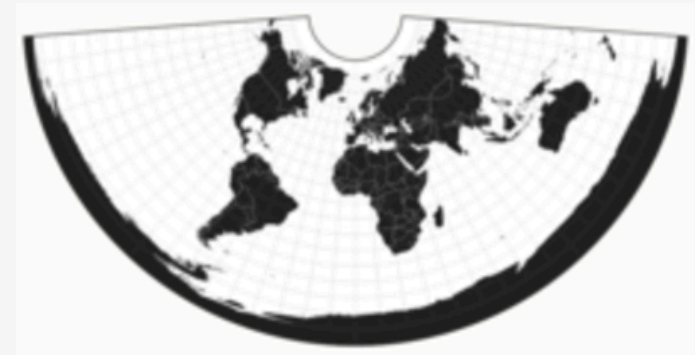
d3.geo.conicEqualArea



d3.geo.conicConformal



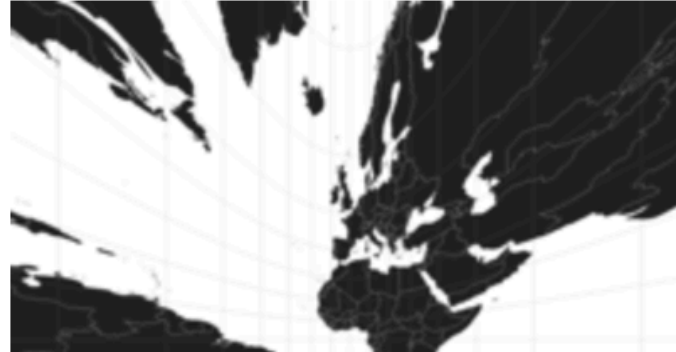
d3.geo.conicEquidistant



d3.geo.equirectangular



d3.geo.gnomonic



d3.geo.mercator



d3.geo.orthographic



d3.geo.stereographic



d3.geo.transverseMercator



The **first** thing D3.js needs to render geography.

A projection *projects* spherical coordinates to a Cartesian plane.

```
var projection = d3.geo.mercator()  
  .center([-74.25, 40.69])  
  .scale(70000)  
  .translate([w/4, h/2]);
```



Path

The **second** thing D3.js needs to render geography.

Takes the projected 2D geometry and formats it for the browser (SVG or Canvas).

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

Drawing a map




```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs' , true );  
map.attr( 'd' , path );  
map.exit().remove();
```

```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs' , true );  
map.attr( 'd' , path );  
map.exit().remove();
```

```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs', true );  
map.attr( 'd', path );  
map.exit().remove();
```

```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs' , true );  
map.attr( 'd' , path );  
map.exit().remove();
```

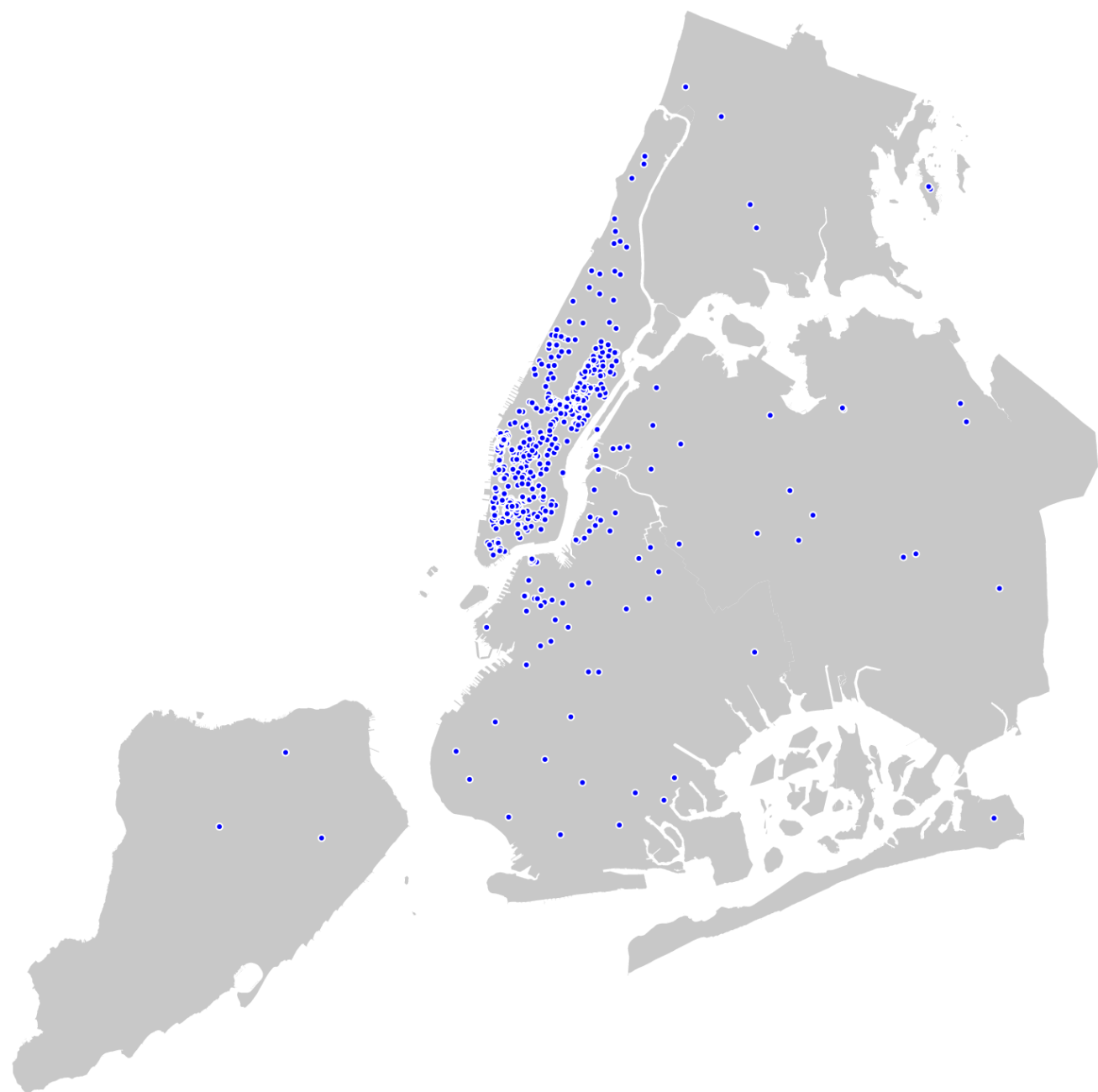
```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs' , true );  
map.attr( 'd' , path );  
map.exit().remove();
```

```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs' , true );  
map.attr( 'd' , path );  
map.exit().remove();
```

```
var map;  
map = map_g.selectAll( 'path.boroughs' )  
    .data(features);  
map.enter()  
    .append( 'path' )  
    .classed( 'boroughs' , true );  
map.attr( 'd' , path );  
map.exit().remove();
```



Rendering data



```
var galleries;
galleries = map_g.selectAll('circle.pin')
    .data(data_galleries);
galleries.enter().append('circle')
    .classed('pin',true);
galleries
    .attr('cx',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[0];
    })
    .attr('cy',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[1];
    })
    .attr('r',2);
galleries.exit().remove();
```

```
var galleries;  
galleries = map_g.selectAll('circle.pin')  
    .data(data_galleries);  
galleries.enter().append('circle')  
    .classed('pin',true);  
galleries  
    .attr('cx',function(d){  
        var coords = projection([+d.lon,+d.lat]);  
        return coords[0];  
    })  
    .attr('cy',function(d){  
        var coords = projection([+d.lon,+d.lat]);  
        return coords[1];  
    })  
    .attr('r',2);  
galleries.exit().remove();
```

```
var galleries;  
galleries = map_g.selectAll('circle.pin')  
    .data(data_galleries);  
galleries.enter().append('circle')  
    .classed('pin',true);  
galleries  
    .attr('cx',function(d){  
        var coords = projection([+d.lon,+d.lat]);  
        return coords[0];  
    })  
    .attr('cy',function(d){  
        var coords = projection([+d.lon,+d.lat]);  
        return coords[1];  
    })  
    .attr('r',2);  
galleries.exit().remove();
```

```
var galleries;
galleries = map_g.selectAll('circle.pin')
    .data(data_galleries);
galleries.enter().append('circle')
    .classed('pin',true);
galleries
    .attr('cx',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[0];
    })
    .attr('cy',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[1];
    })
    .attr('r',2);
galleries.exit().remove();
```

```
var galleries;
galleries = map_g.selectAll('circle.pin')
    .data(data_galleries);
galleries.enter().append('circle')
    .classed('pin',true);
galleries
    .attr('cx',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[0];
    })
    .attr('cy',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[1];
    })
    .attr('r',2);
galleries.exit().remove();
```

```
var galleries;
galleries = map_g.selectAll('circle.pin')
    .data(data_galleries);
galleries.enter().append('circle')
    .classed('pin',true);
galleries
    .attr('cx',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[0];
    })
    .attr('cy',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[1];
    })
    .attr('r',2);
galleries.exit().remove();
```



```
var galleries;
galleries = map_g.selectAll('circle.pin')
    .data(data_galleries);
galleries.enter().append('circle')
    .classed('pin',true);
galleries
    .attr('cx',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[0];
    })
    .attr('cy',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[1];
    })
    .attr('r',2);
galleries.exit().remove();
```

```
var galleries;
galleries = map_g.selectAll('circle.pin')
    .data(data_galleries);
galleries.enter().append('circle')
    .classed('pin',true);
galleries
    .attr('cx',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[0];
    })
    .attr('cy',function(d){
        var coords = projection([+d.lon,+d.lat]);
        return coords[1];
    })
    .attr('r',2);
galleries.exit().remove();
```

