D3 Tutorial

Geographical Map
Three Key Concepts

• GeoJSON
  • A JSON-based format for specifying geographic data
  • D3 creates map based on GeoJSON data

• Map projections
  • Functions that convert from \textit{latitude/longitude} co-ordinates to \textit{x} and \textit{y} co-ordinates

• Geographic path generators - d3.geoPath()
  • Functions that convert GeoJSON shapes into SVG paths
  • Similar to shape generators e.g. d3.line(), d3.area(), etc.
GeoJSON

- A JSON-based format for specifying geographic data
  - [http://geojson.org/](http://geojson.org/)

- A GeoJSON data for Ohio

```json
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "STUSPS10": "OH",
        "NAME10": "Ohio"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [-84.820157, 39.105489],
          [-84.820157, 39.105545],
          [-84.820157, 39.10561],
          ...
        ]
      }
    }
  ]
}
```

- `properties` usually contains the name, id or other attributes for the region
GeoJSON

• A JSON-based format for specifying geographic data
  • [http://geojson.org/](http://geojson.org/)
• A GeoJSON data for Ohio

```json
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "properties": {
        "STUSPS10": "OH",
        "NAME10": "Ohio"
      },
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [-84.820157, 39.105489],
          [-84.820157, 39.105545],
          [-84.820157, 39.10561],
          ...
        ]
      }
    }
  ]
}
```

- **geometry** specify the shape of the region
- The shape **type** can be a Point, a Polygon, a MultiPolygon, etc.
- **coordinates** stores the boundary of this region. For Polygon, it is an array of [latitude, longitude].
Map Projections

• Functions that convert from \textit{latitude/longitude} co-ordinates to \textit{x} and \textit{y} co-ordinates

• Introduction of various map projections
  - \url{http://www.progonos.com/furuti/MapProj/Normal/TOC/cartTOC.html}

• Choosing a projection
  - Every projection will distort shape, area, distance and/or direction
  - Choosing which property you don’t want to be distorted and accepting that there’ll be distortion in the other properties
Map Projections

• D3 supports various map projections
  • https://github.com/d3/d3-geo-projection

• For example
  • we have a position [latitude, longitude] on the map
  • We can project the position to the screen by D3 projection functions

    ```javascript
    var projection = d3.geoEquirectangular();
    var posMap = [latitude, longitude];
    var posScreen = projection(posMap);
    ```

• A tool help you understand different projections
  • https://bl.ocks.org/d3indepth/f7ece0ab9a3df06a8cecd2c0e33e54ef
Example: Map of Ohio

First, load the GeoJSON for Ohio to `ohio`

Then, create a equirectangular projection (plate carrée projection)

`.fitExtent(extent, GeoJSON)`

- The specified region will be scaled to fill the `extent` on the screen
Example: Map of Ohio

```javascript
var geoGenerator = d3.geoPath()
  .projection(projection);

var ohioPath = d3.select('svg')
  .append('path')
  .attr('d', geoGenerator(ohio));
```

- Create a geographic path generators
  - Specify the projection by `.projection()`
- Draw a map of Ohio by svg path
Example: Map of USA Mainland

- We can draw the map of USA mainland through a similar process