COS30045 Data Visualisation

Task 6.1 D3 Geo Paths and Projections

ILO	Create web-based interactive visualisations using real-world data sets.
Aim:	Display a map using D3
Resources:	*Textbook:* Murray Ch 14 Murray on ProQuest Murray on Safari (Make sure you use v2 of Murray as per links above)
To be marked as Complete your submission must:	Submit working code that meets the requirements specified in document below. Demonstrate appropriate use of HTML, CSS and D3. Properly formatted code Well commented code with references to code sourced from web, stack overflow etc. where appropriate. Demonstrate and explain code to tutor in class.
Submission	Submit to Doubtfire • code that demonstrates a map showing Victoria's LGAs Bring code to class to demonstrate to tutor

Note: The functions handling scale have changed between D3 v3 and D3 v4. This is something to be aware of if you are doing your on research into this topic. Make sure you use Murray Ed 2. Code examples from Ed 1 will not work.

Code in this Task based on Murray Ch 14

Overview

In this exercise we will be attempting to get D3 to draw us some nice maps so we can display data geographically.

Victorian LGA map

Step 1: Get a GeoJSON file

GeoJSON is a JSON file format that encodes geodata. Obtaining a GeoJSON file of the area you want to map can be tricky. If you can't find one for the area you are interested in you may need to generate it yourself from shapefiles (e.g., .shp). At the end of Ch 14 Murray describes some ways of converting shapefiles into GeoJSON files. For this task we will be using a pre-prepared GeoJSON file of Victoria's Local Government Areas (or LGA). This file was created by a former student Chris D. Many local government statistics are provided by LGA so you may also find this file useful. Below is an example of GeoJSON format for Glen Eira LGA. Only a small proportion of the path coordinates are shown here and some formatting has been applied.

```
{
    "type":"FeatureCollection",
    "features":[
        {
            "type": "Feature",
            "properties": { "LGA_name": "Glen Eira"},
            "geometry": {
            "type": "Polygon",
            "coordinates": [ [ [ 144.99705204000003, -37.88386502 ],
                [ 144.99674515000001, -37.88350142 ], [ 144.99800358000002, -37.87691677 ],
                [ 145.00003579000003, -37.86645509 ], [ 145.00515237, -37.86708727 ],
                [ 145.009166, -37.86759616 ], [ 145.00987834, -37.86374548 ],
                [ 145.01032084, -37.86136408 ], [ 145.01052136000001, -37.8602253 ],
                [ 145.01513662, -37.86084043000001 ], [ 145.01946651, -37.8627244 ],
                [ 145.02757912000001, -37.86627437 ], [ 145.02936622000001, -37.8670398 ],
                [ 145.03020455000001, -37.86753524 ], [ 145.03078782, -37.86790524 ],
```

Step 2: Set up the paths

As in Task 6.3 we need to express the data as a SVG path. We can do this with `d3.geo-Path()`. When using `geoPath()` we need to specify a projection. There are many different projections that could be chosen (see D3 Projections) but for this tasks we will use the Mercator projection which is a standard widely used since the 1500's.

```
//Width and height
var w = 500;
var h = 300;


var projection = d3.geoMercator()


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

We can then add our SVG canvas and read in our GeoJSON file and bind it to a path using `d3.json()`.
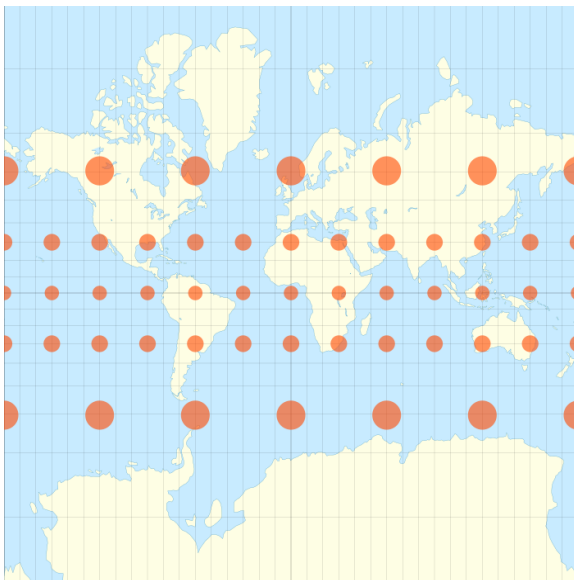
```
var svg = d3.select("body")
            .append("svg")
            .attr("width", w)
            .attr("height", h)
            .attr("fill", "grey");

d3.json("LGA_VIC.json").then(function(json) {

    svg.selectAll("path")
       .data(json.features)
       .enter()
       .append("path")
       .attr("d", path);
    });
```


Fig 1. Mercer projection showing distortions
https://en.wikipedia.org/wiki/Mercator_projection

Unfortunately, if you run this now it will appear as if nothing has happened. Although if you check the DOM you will see that paths have been generated. The problem is that the Mercator projection default view is of the whole world (see Fig 1), but Victoria, Australia is only one tiny bit of it. So you need to transform the view so we can see Victoria in our SVG. We can do this using `.centre()` `.translate()` and `.scale()`.

The code below will show you the map on our SVG. Work out a scale value that gives you a good view of our VIC map.

```
var projection = d3.geoMercator()
               .center([145, -36.5])
               .translate([w / 2, h / 2])
               .scale(2450);
```

In the end you should have something like this: