Lecture Exercise 9 (5/25)

Submit your team number

Question *Submitted May 25th 2023 at 4:20:02 pm*

Please enter your team number.

12

1. US Map (Continuation from lecture slides)

Let's create the map of the United States. Use exercise9.html and us-states.json file.

Question 1 *Submitted May 25th 2023 at 4:26:52 pm*

Define `projection1` and `pathgeo1` . `projection1` is a function that converts [longitude, latitude] to [x, y] coordinate. Use `d3.geoAlbersUSA()` projection for `projection1`. `pathgeo1` is a path generator.

```
const projection1 = d3.geoAlbersUsa(); const pathgeo1 = d3.geoPath().projection(projection1);
```

Question 2 *Submitted May 25th 2023 at 4:28:38 pm*

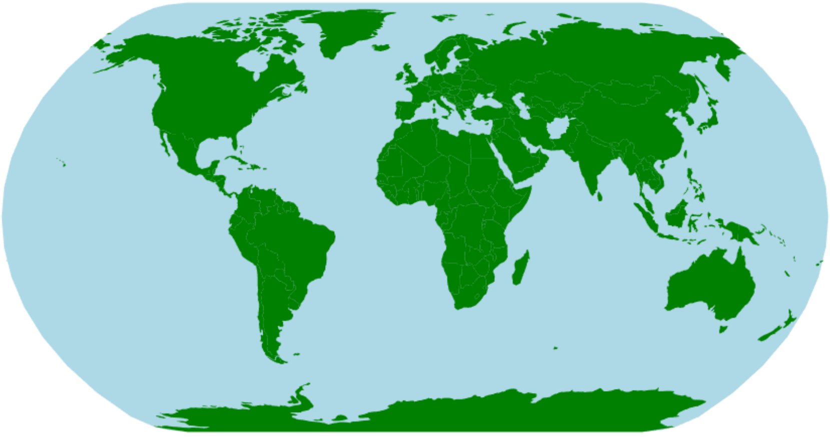
Now let's load us-states.json and render the map.

You will notice that the map is not visible entirely. This is because the map is not positioned and scaled properly. This can be rectified by adding projection functions - translate and scale.

```
const projection1 = d3.geoAlbersUsa()

.scale(700)

.translate([width/2, height /2]);

const pathgeo1 = d3.geoPath().projection(projection1);

// load us-states.json
```

```
const map = d3.json("./us-states.json");
map.then(map => {
svg1.selectAll("path")
.data(map.features)
.enter()
.append("path")
.attr("d", pathgeo1)
});
```

2. A static world map



Now let's draw a somewhat colorful world map!

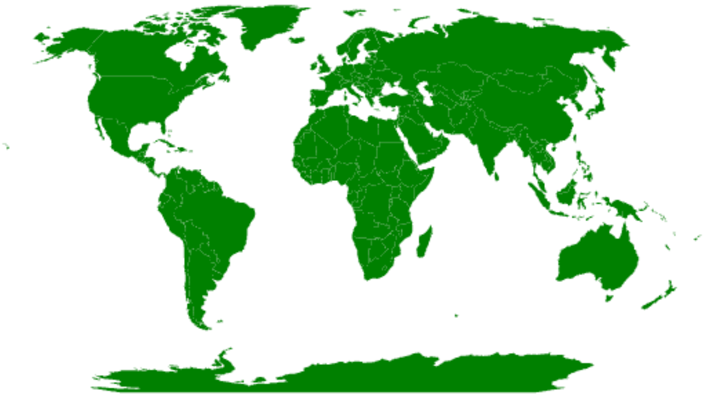
Question 1 *Submitted May 25th 2023 at 4:29:30 pm*

Define `projection2` `pathGeo2` using geoNatrualEarth type projection (`d3.geoNaturalEarth1()`). Load world.json to the svg canvas.

> You can actually use the file's **URL as an input** to `d3.json()` instead of downloading the file.

Change the color of the map by using the `fill` attribute as well.

You should see an output as following:



Copy/paste your code.

```
const projection2  = d3.geoNaturalEarth1()
            .scale(100)
            .translate([width/2, height/2])

        const pathgeo2 = d3.geoPath().projection(projection2);

        d3.json("world.json").then(map =>{
            svg2.selectAll('.worldpath')
                .data(map.features)
                .enter().append("path")
                .attr('class', 'worldpath')
                .attr('d', pathgeo2)
                .attr("fill", "green");
        });
```

Question 2 *Submitted May 25th 2023 at 4:32:48 pm*

Notice that your current map has **no** blue area - the ocean or the *outline* of the globe.

`d3.geoPath()` actually allows you to add the outline as well - the outline is called *sphere*. Following block of code actually add an additional <path>, which defines the `d` attribute as an output from `pathgeo2({type: 'Sphere'})` . The resulting <path> will draw the sphere of the map and fill the outline with the skyblue color.

```
// drawing a sphere, the outline of the globe
svg2.append('path')
    .attr('class', 'sphere')
    .attr('d', pathgeo2({ type: 'Sphere' }))
    .attr("fill", "skyblue");
```

Insert the code block above to the correct place in your current exercise9.html, and copy/paste your code entire body of `d3.json("us-states.json").then(map => { ... UPLOAD EVERYTHING HERE ... });`

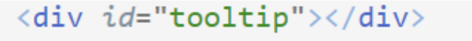
Hint: Think about which one is on top of the other (outline? or the map itself?)

~~~
```

        d3.json("world.json").then(map =>{
            svg2.selectAll('.worldpath')
                .data(map.features)
                .enter().append("path")
                .attr('class', 'worldpath')
                .attr('d', pathgeo2)
                .attr("fill", "green");
            svg2.append('path')
                .attr('class', 'sphere')
                .attr('d', pathgeo2({ type: 'Sphere' }))
~~~

```
                 .attr("fill", "skyblue");
```

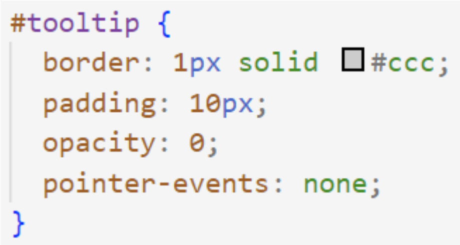
3. Add tooltips on the world map

Let's make the world map more interactive, by adding a tooltip which shows the country's name when hover on it. We have already created a `<div>` with the id "tooltip" on the html file.



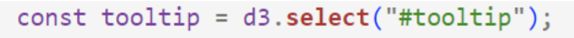
```
<div id="tooltip"></div>
```

Also, we pre-defined some style of the tooltip. We will use the `opacity` property to control the display of the tooltip to have it appeared or disappeared.



```
#tooltip {
  border: 1px solid □#ccc;
  padding: 10px;
  opacity: 0;
  pointer-events: none;
}
```

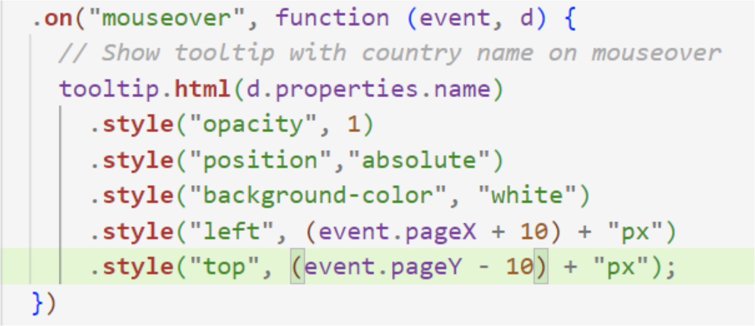
First of all, you need to define a tooltip selection as below. (we recommend to define it right below the definition of `pathgeo2` .



```
const tooltip = d3.select("#tooltip");
```

Tooltip is made up with two mouse events, each of which is responsible for "appear" and "disappear" transition.

1. Appear: add `.on()` with `mouseover` event.

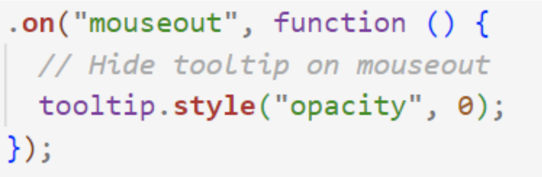


```
.on("mouseover", function (event, d) {
  // Show tooltip with country name on mouseover
  tooltip.html(d.properties.name)
    .style("opacity", 1)
    .style("position","absolute")
    .style("background-color", "white")
    .style("left", (event.pageX + 10) + "px")
    .style("top", (event.pageY - 10) + "px");
})
```

  - Using .html() method to show the country name
  - Specify the CSS style of the tooltip `<div>`:
    - Set `opacity` to be 1.
    - Set `position` to be `absolute`, so it will be placed relative to `<body>`, check this link for more details.
    - Set the `background-color` to be `white`.

        - Set the `top` and `left` property by the current cursor position. The event.pageX and event.pageY will return the X (horizontal) coordinate and Y (vertical) coordinate in pixels of the cursor. We also add an offset value 10 to fine-tune the position of the tooltip.

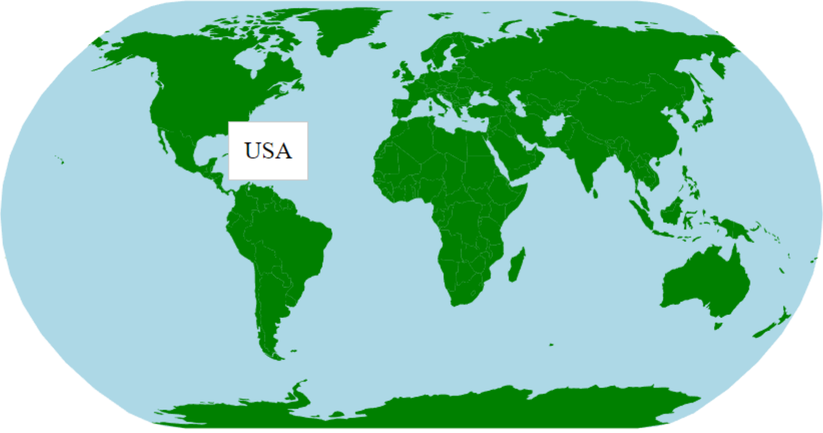
2. Disappear: add `.on()` with `mouseout` event.



```
.on("mouseout", function () {
  // Hide tooltip on mouseout
  tooltip.style("opacity", 0);
});
```

    - Hide it by changing the `opacity` to 0

Now when your mouse hovers on the map, it will show the country's name.



Question *Submitted May 25th 2023 at 4:42:03 pm*

I have finished the steps above!

● Yes

○ Not yet.

4. (Bonus) Add a <path> on top of the map

Let's add a route, represented as a line, on top of the map!

Define `route` which has pairs of [longitude, latitude] of three different cities.

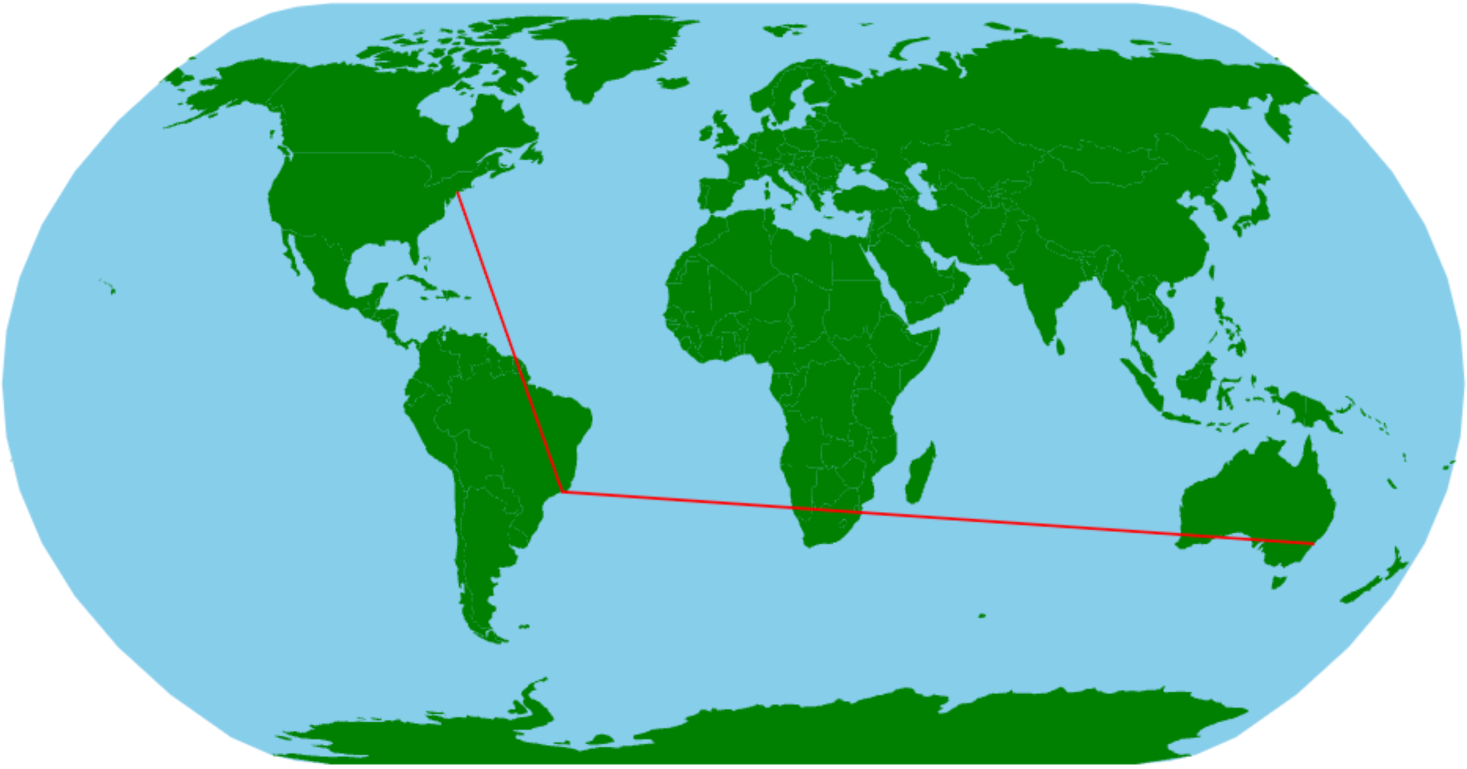
> Note that geojson uses System1984 coordinate system, not DMS. Something to keep in mind when searching the coordinates in the future.

```
const route = [[-73.935242, 40.730610],[-43.196388, -22.908333], [151.209900,-33.865143]]
                // NYC, Rio de Janeiro, Sydney
                // [longitude, latitude]

const routeProjected = [];
```

Now that we have `projection2` function, which converts a [long, lat] to [x, y] on the svg canvas, you can write code to fill `routeProjected`.

Then, create a path string (e.g. M20 10 L 40 50 Z) by using the line generator (refer to lecture 8 slide - Line Chart). This resulting string will simply be the value of the `d` attribute of a <path> element, which represents a route!

You will see something like this in the end :) Now you know how to overlay different elements on a map!



Question *Submitted May 25th 2023 at 4:42:18 pm*

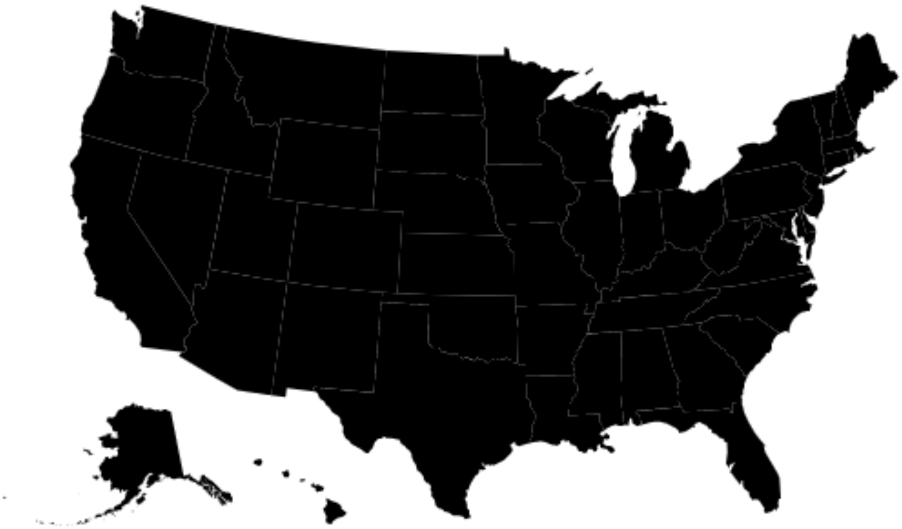
Copy/paste your code 1) that completes `routeProjected`, 2) that generates the path string for the route, and 3) that adds a <path> for the route.

No response

Upload Your Files

Question 1 *Submitted May 25th 2023 at 4:42:37 pm*

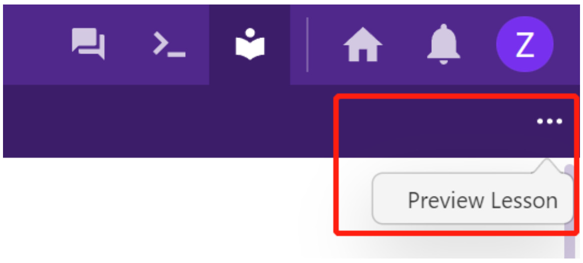
Upload the screenshot of your resulting webpage. You will need to click the "clip" button to upload a file into the Answer box.



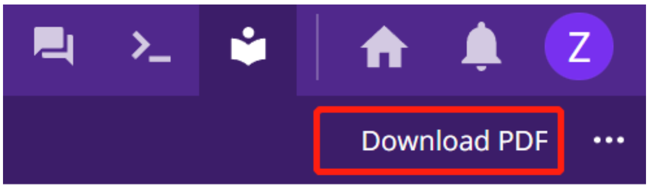
Question 2 *Submitted May 25th 2023 at 4:42:39 pm*

You need to download the PDF of lab 6 and upload it with other files to the Gradescope. Follow the instructions on how to download PDF file:

1. Click on the ellipsis button and the Preview Lesson.



2. After that, click on the Download PDF button.



● PDF downloaded!

○ Haven't done yet!

Question 3 *Submitted May 25th 2023 at 4:42:41 pm*

Upload the following files to Gradescope. You need to make **a group submission,adding all**

present members in your team, so that the present members get the participation credit.

Files to upload:

- lab6.html
- PDF you downloaded as Q2

● Our team uploaded the the files on gradescope!

○ Oops, our team did not upload the files on gradescope!

Feedback

Question *Submitted May 25th 2023 at 4:42:43 pm*

Was the activity today clear? If not, please share how the course can improve it. Your comments will help us design future lab content (and also future students).

No response

Example Solution

```
<!DOCTYPE html>
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <script src="https://d3js.org/d3.v7.min.js"></script>
    <script src="https://unpkg.com/topojson@3.0.2/dist/topojson.min.js"></script>
    <title>World Map</title>
    <style>
        #tooltip {
          border: 1px solid #ccc;
          padding: 10px;
          opacity: 0;
          pointer-events: none;
        }
    </style>
</head>

<body>
    <div id="tooltip"></div>
    <div id="us">
    </div>
    <div id="world">
    </div>
    <script>
      var width = 1000;
      var height = 800;

      //---------------------------USA ---------------------
      var svg1 = d3.select("#us")
          .append("svg").attr("width", width).
          attr("height", height);
      //TO DO: Create projection and pathgeo variables for US
      const projection1 = d3.geoAlbersUsa()
          .scale(700)
          .translate([width / 2, height / 2.5]); //chain translate and scale
      const pathgeo1 = d3.geoPath().projection(projection1)
      //TO DO: Load JSON file and create the map
      const map = d3.json("./us-states.json")
      map.then(map => {
          console.log(map)
          svg1.selectAll("path")
              .data(map.features)
              .enter()
              .append("path")
              .attr("d", pathgeo1)
          });

      const projection2 = d3.geoNaturalEarth1()
        .scale(100)
        .translate([width / 2, height / 2.5]); // Chain translate and scale
        const pathgeo2 = d3.geoPath().projection(projection2);

      var svg2 = d3.select("#world")
        .append("svg").attr("width", width)
        .attr("height", height);

      // Base sphere for the world map
      svg2.append('path')
        .attr('class', 'sphere')
        .attr('d', pathgeo2({ type: 'Sphere' }))
        .attr('fill', 'lightblue');

      // TO DO: Load JSON file and create the map
      const worldMap = d3.json("https://raw.githubusercontent.com/holtzy/D3-graph-gallery/master/DA
      const tooltip = d3.select("#tooltip");
      worldMap.then(d => {
        svg2.selectAll("path")
          .data(d.features)
          .enter()
          .append("path")
          .attr("d", pathgeo2)
          .attr("fill", "green")
          .on("mouseover", function (event, d) {
            // Show tooltip with country name on mouseover
            tooltip.html(d.properties.name)
              .style("opacity", 1)
              .style("position","absolute")
              .style("background-color", "white")
              .style("left", (event.pageX + 10) + "px")
              .style("top", (event.pageY - 10) + "px");
          })
          .on("mouseout", function () {
            // Hide tooltip on mouseout
            tooltip.style("opacity", 0);
          });
      });

    </script>
</body>

</html>
```