D3 Lab - Week 4 - Adding Base Map

Instructions

Watch the D3 - Part 4 - Maps playlist on the Media Gallery

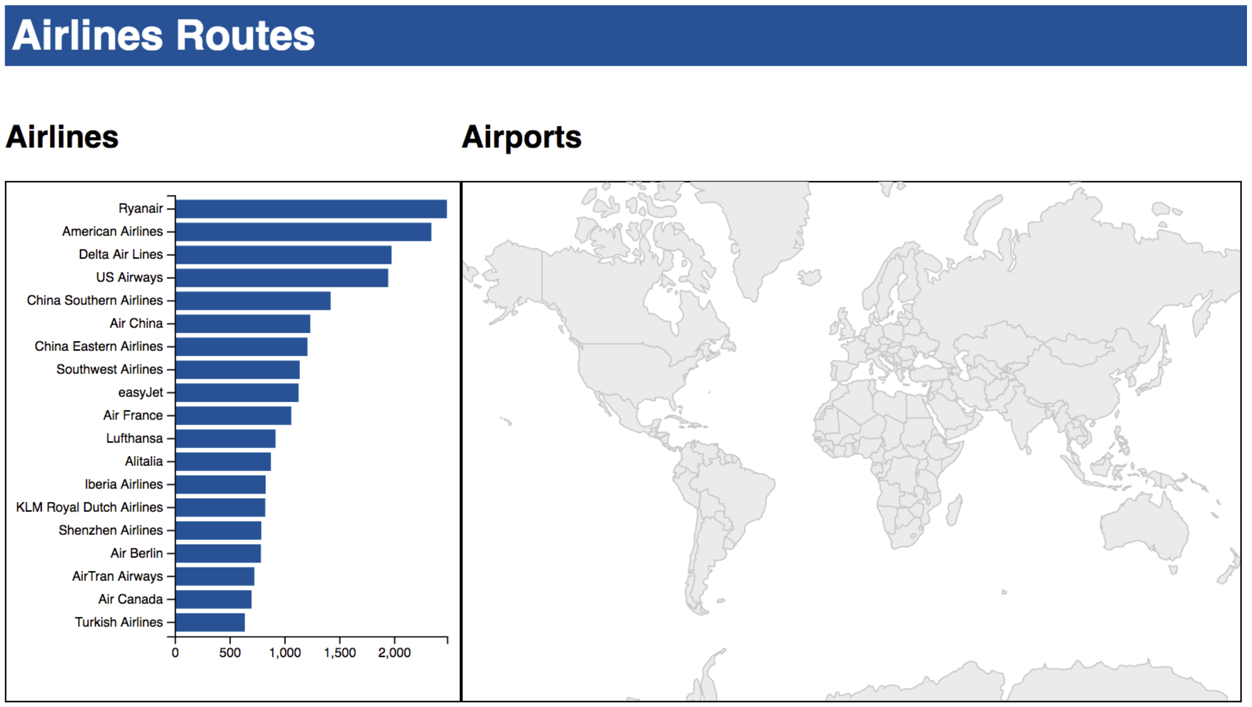
***Alert*:** This assignment assumes that you completed and passed in the D3 week 3 - Drawing with data assignment, as this assignment will build on the final solution of that assignment. If you didn't finish it, please go back to and complete the assignment before starting this one.

****Important*:*** *In the provided code, CHANGE ONLY THE TODOs SECTION, It is important that you KEEP THE OTHER PARTS OF THE CODE UNCHANGED, and do not change names of functions, CSS classes or element IDs.*

Goal

In this assignment, our goal is to add a base map to our application. For now, this map will show just the borders of the countries, in the next assignment we will plot dots representing the airport on top of this map.

By the end of the assignment, our page should look like this:



Once Again, we will build it using the index.html file you submitted in the last assignment. Open that file on your preferred editor and let's get started.

Loading Map Information

To draw our map, we need to load the GeoJSON file with the specifications of which countries we have and their borders. Download the file below, and save it to the same folder of your index.html file.

countries.geo.json (you may have to right-click and select "Save as")

The next step is to change our loadData function to also load the countries.geo.json file. It should now look like this.

```
function loadData() {
    return Promise.all([
        d3.csv("routes.csv"),
        d3.json("countries.geo.json"),
    ]).then(datasets => {
        store.routes = datasets[0];
        store.geoJSON = datasets[1]
        return store;
    })
}
```

Note that we are using Promise.all to load multiple files, and the datasets variable will contain those files in the same order the were passed to the .all function. Even with this change, your previous code should still work correctly.

Map Config

Similar to what we did for the bar chart, we will create a function to set and return information about the size of the map called getMapConfig().

Add the function bellow to your code, and complete the TODOs

```
function getMapConfig(){
  let width = 600;
  let height = 400;
  let container = //TODO: select the svg with id Map
     //TODO: set the width and height of the conatiner to 
be equal the width and height variables.
  return {width, height, container}
}
```

Projection

The next step is to choose which projection to use, we will create a function that return the projection. We will need to translate the projection to make sure that the center of the world is also in the center of our svg, for this reason, this function will receive as parameter the width and height of the map. We will also save the projection on our store, so we can easily access it later.

Add the function bellow to your code, and complete the TODOs

```
function getMapProjection(config) {
  let {width, height} = config;
  let projection = //TODO: Create a projection of type 
Mercator.
  projection.scale(97)
            .translate([width / 2, height / 2 + 20])

  store.mapProjection = projection;
  return projection;
}
```

Drawing The Map

Our last function is the one responsible for actually drawing the map, this function will receive as parameter, the container, the features of the GeoJSON file, a.k.a. the countries we want to draw, and the projection to use.

Add the function bellow to your code, and complete the TODOs

```
function drawBaseMap(container, countries, projection){
  let path = //TODO: create a geoPath generator and set its 
projection to be the projection passed as parameter.

  container.selectAll("path").data(countries)
      .enter().append("path")
      .attr("d",//TODO: use the path generator to draw each 
country )
      .attr("stroke", "#ccc")
      .attr("fill", "#eee")
}
```

Calling The Functions

Finally, lets add a function that will be responsible for calling the previous functions in the right sequence. It will be called *drawMap.* And it should look like the code below.

```
function drawMap(geoJeon) {
    let config = getMapConfig();
    let projection = getMapProjection(config)
    drawBaseMap(config.container, geoJeon.features, 
projection)
}
```

Also, we need to add a call to drawMap in our showData function, that should now look like this:

```
function showData() {
    //Get the routes from our store variable
    let routes = store.routes
    // Compute the number of routes per airline.
    let airlines = groupByAirline(store.routes);
    console.log(airlines)
    drawAirlinesChart(airlines)
    drawMap(store.geoJSON) //Using the data saved on 
loadData
}
```

Submit

If everything went fine, you should see a page that looks like the image on the top of this assignment. Also check the console (on Chrome, right-click the page, select inspect and then, the console tab) for any error

Save your file as index.html and submit it. You don't need to submit the other files in your folder, just the index.html file. All your code should be in this file.