D3 Lab - Week 5 - Showing Airports

Instructions

This assignment is also based on the D3 - Part 4 - Maps playlist, but please also watch the D3 - Part 5 - Lines and Arcs on the Media Gallery

***Alert*:** This assignment assumes that you completed and passed in the D3 week 4 - 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.

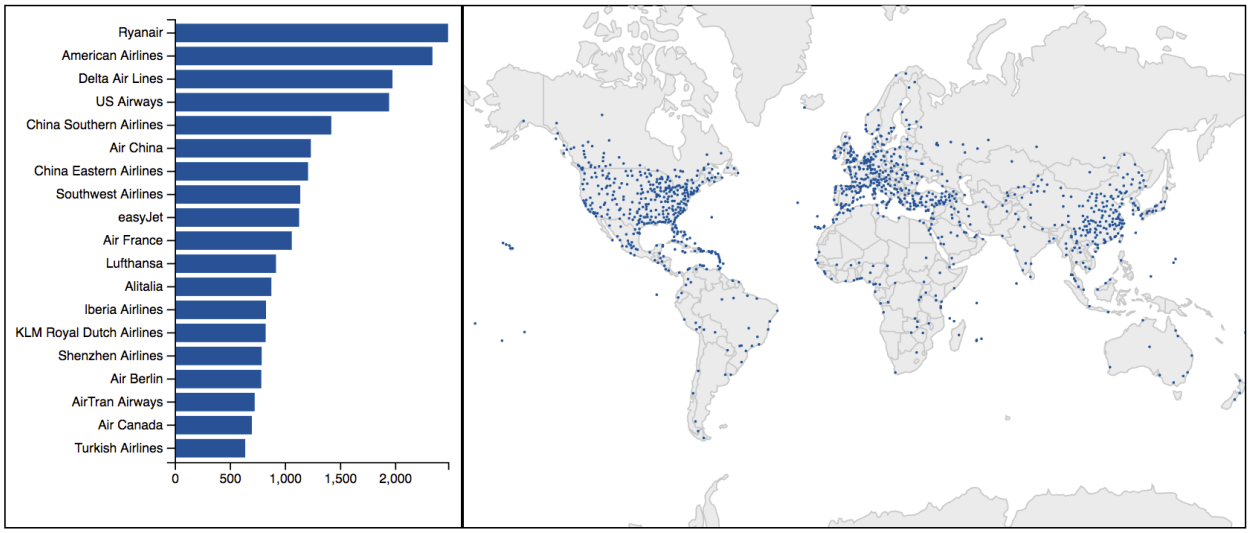
****Important*:** In the provided code, CHANGE ONLY THE TODOs SECTION, It is important that you KEEP THE OTHER PARTS OF THE CODE UNCHANGED, and do not change names of functions, CSS classes or element IDs.*

Goal

In this assignment, our goal is to add to our map a dot for each airport we have in the data. That will help us to see where in the world are the airports that those airlines operate. At the end of the assignment, our page should look like the image below.

Airlines Routes

Airlines **Airports**



Grouping By Airports

Similarly to what we did fro the airlines bar chart, we will have to group the routes by airport this time. One major difference here is that each route have two airports, so we have to make sure to consider both airports when doing the aggregation. Bellow is the *groupByAirport* function that will group the routes by airport. Make sure you understand it before continuing.

Copy this function to your file

```
function groupByAirport(data) {
    //We use reduce to transform a list into a object where each key points to an aiport. This way makes it easy to check if is the first time we are seeing the airport.
    let result = data.reduce((result, d) => {
        //The || sign in the line below means that in case the first option is anything that Javascript consider false (this insclude undefined, null and 0), the second option will be used. Here if result[d.DestAirportID] is false, it means that this is the first time we are seeing the airport, so we will create a new one (second part after ||)
```

```
        let currentDest = result[d.DestAirportID] || {
            "AriportID": d.DestAirportID,
            "Airport": d.DestAirport,
            "Latitude": +d.DestLatitude,
            "Longitude": +d.DestLongitude,
            "City": d.DestCity,
            "Country": d.DestCountry,
            "Count": 0
        }
        currentDest.Count += 1
        result[d.DestAirportID] = currentDest

        //After doing for the destination airport, we also update the airport the airplane is departing from.
        let currentSource = result[d.SourceAirportID] || {
            "AriportID": d.SourceAirportID,
            "Airport": d.SourceAirport,
            "Latitude": +d.SourceLatitude,
            "Longitude": +d.SourceLongitude,
            "City": d.SourceCity,
            "Country": d.SourceCountry,
            "Count": 0
        }
        currentSource.Count += 1
        result[d.SourceAirportID] = currentSource

        return result
    }, {})

    //We map the keys to the actual ariorts, this is an way to transform the object we got in the previous step into a list.
    result = Object.keys(result).map(key => result[key])
    return result
}
```

Draw Airports

Because we have already created the projection and set the configurations of the map container in the previous part of our project, we will need only one function to draw the airports in the map, we will call this function *drawAirports*. It will receive as parameter the airports, and it will use some methods we created before to obtain the projection and the container it should use.

Copy the function bellow to your file and complete the TODOs

```
function drawAirports(airports) {
  let config = getMapConfig(); //get the config
  let projection = getMapProjection(config) //get the
projection
  let container = config.container; //get the container

  let circles = container.selectAll("circle");
  //TODO: bind the airports to the circles using the .data
method.
  //TODO: for each new airport (hint: .enter)
  //      - Set the radius to 1
  //      - set the x and y position of the circle using
the projection to convert longitude and latitude to x and y
porision.
  //      - Set the fill color of the circle to  "#2a5599"
}
```

Adding The Calls To showData Function

The last step is to call both function from showData so we can first get the data aggregated by airport, and second draw the airports into the map. it should look like this now

```
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)

    let airports = groupByAirport(store.routes);
    drawAirports(airports)
}
```

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.