

D3 Lab - Week 3 - Airlines Barchart

Instructions

Watch the [D3 - Part 3 - Drawing with Data](#) playlist on the [Media Gallery](#)

***Alert*:** This assignment assumes that you completed and passed in the D3 week 2 - Dealing 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.*

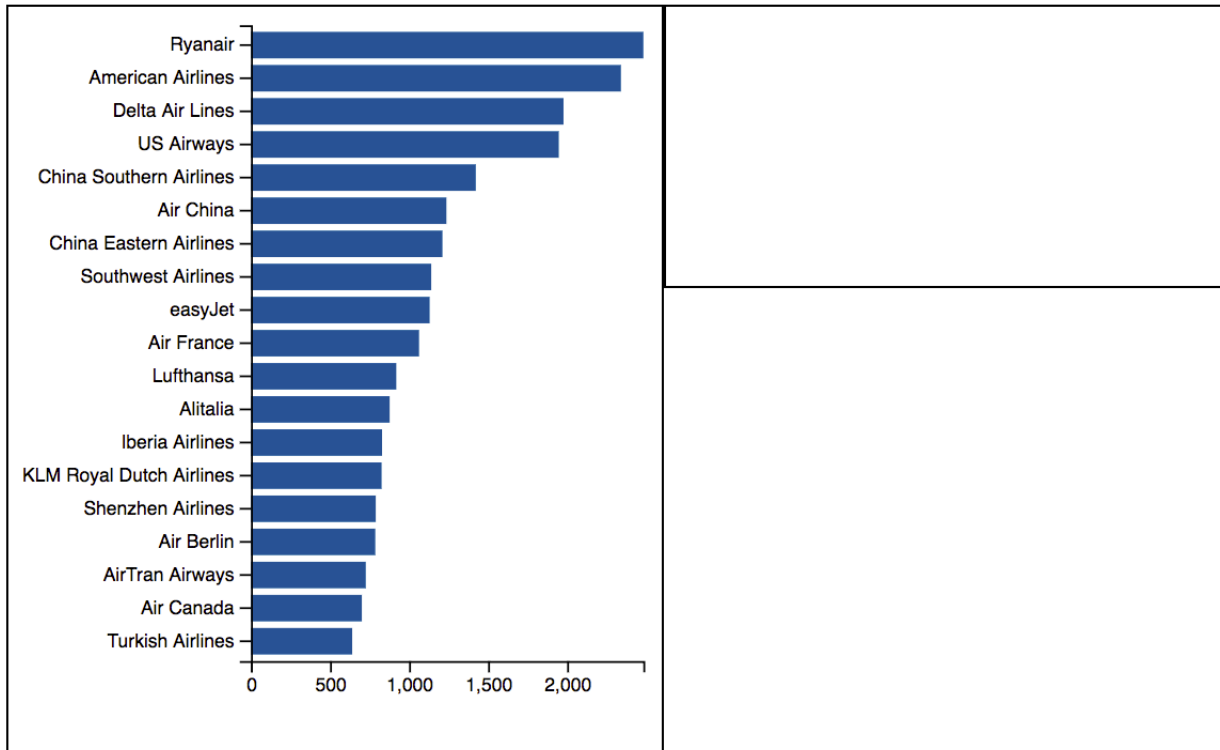
Goal

In this assignment we will create a horizontal bar to show the airlines and their number of routes. Our goal is to, by the end of the assignment, make our page look like the image bellow:

Airlines Routes

Airlines

Airports



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.

Main Function

Our main function will be called *drawAirlinesChart* and the goal of this function is to call a set of sub-functions that will create our chart. Let's start by defining this function. For now it is an empty, but we will fill it in a bit.

Let's add the following function to your index.html file.

```
function drawAirlinesChart(airlines) {  
  
}
```

ddd

Next let's add a call to this function on our *showData* function, 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)  
  // Draw airlines barchart  
  drawAirlinesChart(airlines)  
}
```

Nothing really exciting here, but this sets the stage for us to add the functions that will do the actual job.

Config

One important thing to consider is the size of our chart, here we will create a function that will set and return details about the size of our chart. The function will return an object

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

```
function getAirlinesChartConfig() {  
  let width = 350;  
  let height = 400;  
  let margin = {  
    top: 10,  
    bottom: 50,  
    left: 130,  
    right: 10  
  }  
  //The body is the area that will be occupied by the  
  bars.  
  let bodyHeight = height - margin.top - margin.bottom  
  let bodyWidth = //TODO: Compute the width of the body  
  by subtracting the left and right margins from the width.  
  
  //The container is the SVG where we will draw the  
  chart. In our HTML is the svg tag with the id AirlinesChart
```

```

    let container = //TODO: use d3.select to select the
    element with id AirlinesChart
    container
      .attr("width", width)
      //TODO: Set the height of the container

```

```

    return { width, height, margin, bodyHeight, bodyWidth,
    container }
  }
}

```

Add a call to this function to *drawAirlinesChart*, it should look like this now.

```

function drawAirlinesChart(airlines) {
  let config = getAirlinesChartConfig()
}

```

The size of the left square of the page should be bigger now, representing the size our chart will have.

Scales

The next step is to create the scales, that will help us to convert the data from the data space to the graphical space. We will use information from our dataset as information on what is the size of the chart to create those scales.

Copy the function bellow to your index.html file and complete the TODOs

```

function getAirlinesChartScales(airlines, config) {
  let { bodyWidth, bodyHeight } = config;
  let maximunCount = //TODO: Use d3.max to get the
  highest Count value we have on the airlines list.

  let xScale = d3.scaleLinear()
    //TODO: Set the range to go from 0 to the width of
  the body
    //TODO: Set the domain to go from 0 to the maximun
  value fount for the field 'Count'

  let yScale = d3.scaleBand()
    .range([0, bodyHeight])
    .domain(airlines.map(a => a.AirlineName)) //The
  domain is the list of ailines names
    .padding(0.2)
}

```

```
    return { xScale, yScale }  
}
```

Drawing the bars

We have finally have the information we need to draw the bars, we will use some information from our size configuration, as well as the scales generated by the *getAirlinesChartScales* function.

Copy the function bellow to your index.html file and complete the TODOs

```
function drawBarsAirlinesChart(airlines, scales, config) {  
  let {margin, container} = config; // this is equivalent  
to 'let margin = config.margin; let container =  
config.container'  
  let {xScale, yScale} = scales  
  let body = container.append("g")  
    .style("transform",  
    `translate(${margin.left}px,${margin.top}px)`  
  )  
  
  let bars = body.selectAll(".bar")  
    //TODO: Use the .data method to bind the airlines to  
the bars (elements with class bar)  
  
  //Adding a rect tag for each airline  
  bars.enter().append("rect")  
    .attr("height", yScale.bandwidth())  
    .attr("y", (d) => yScale(d.AirlineName))  
    //TODO: set the width of the bar to be proportional  
to the airline count using the xScale  
    .attr("fill", "#2a5599")  
}
```

Now lets add a call to this two new functions on our *drawAirlinesChart* function, it should now look like this:

```
function drawAirlinesChart(airlines) {  
  let config = getAirlinesChartConfig();  
  let scales = getAirlinesChartScales(airlines, config);  
  drawBarsAirlinesChart(airlines, scales, config)  
}
```

You should see one bar for each airline.

Axes

The last step is to add the axes, so we can know which bar belongs to which airline. Lets add a new function that will do this job, called *drawAxesAirlinesChart*. It will receive the same parameters of our last function.

```
function drawAxesAirlinesChart(airlines, scales, config){
  let {xScale, yScale} = scales
  let {container, margin, height} = config;
  let axisX = d3.axisBottom(xScale)
    .ticks(5)

  container.append("g")
    .style("transform",
      `translate(${margin.left}px,${height -
margin.bottom}px)`
    )
    .call(axisX)

  let axisY = //TODO: Create an axis on the left for the Y
scale
  //TODO: Append a g tag to the container, translate it
based on the margins and call the axisY axis to draw the
left axis.
}
```

Finally call the *drawAxesAirlinesChart* function from *drawAirlinesChart* to effectively draw the axes. It should look like this now:

```
function drawAirlinesChart(airlines) {
  let config = getAirlinesChartConfig();
  let scales = getAirlinesChartScales(airlines, config);
  drawBarsAirlinesChart(airlines, scales, config);
  drawAxesAirlinesChart(airlines, scales, config);
}
```

Submit

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.