CSCE 5320 Scientific Data Visualization

Marks and Channels

ICE-6: Tutorial

- **Encoding Data with Marks and Channels**

  Marks: Points, Lines, Areas (Rows)

  Channels: Position, Color, Shape, size (Column)

This is an example, please use your own dataset from ICE-5 or create another dataset which contains quantitative value (ex, sales, length, weight, distance etc).

CSV example data Link:

https://gist.github.com/haili0109/ea79c34fd1a2957ce4f1f38a7692eff4

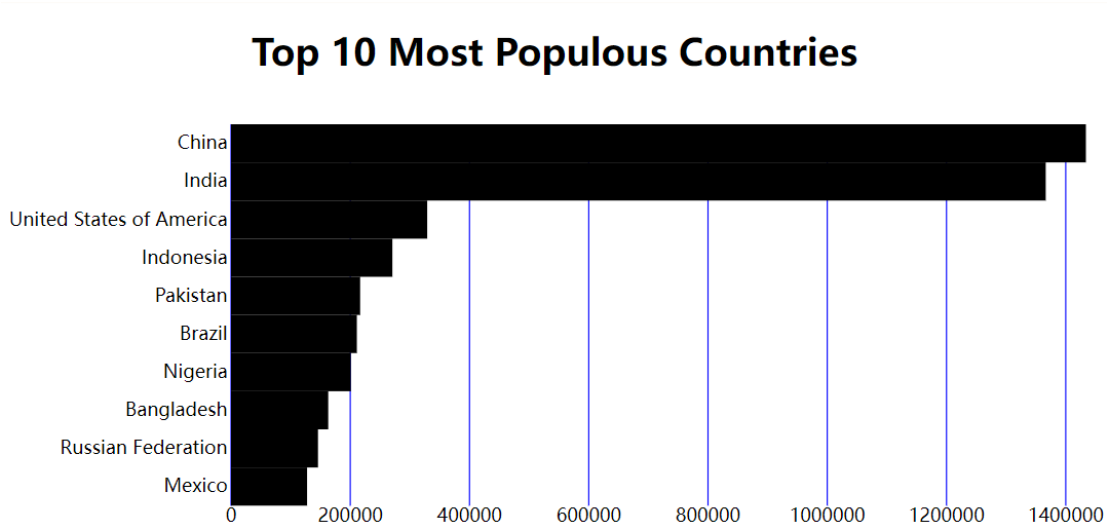
- **Rendering Marks and Channels with D3.js and SVG**

index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>ICE-6: Rendering Marks and Channels with React & D3</title>
    <style>
      body {
        margin: 0;
        overflow: hidden;
      }
      h1{
          font-size: 2em;
          text-align: center;
      }
    </style>
    <script src="https://unpkg.com/react@16.9.0/umd/react.production.min.js"></script>
    <script src="https://unpkg.com/react-dom@16.9.0/umd/react-dom.production.min.js"></script>
    <script src="https://unpkg.com/d3@5.11.0/dist/d3.min.js"></script>
  </head>
  <body>
    <h1>Top 10 Most Populous Countries</h1>
    <div id="root"></div>
    <script src="bundle.js"></script>
  </body>
</html>

```

index.js

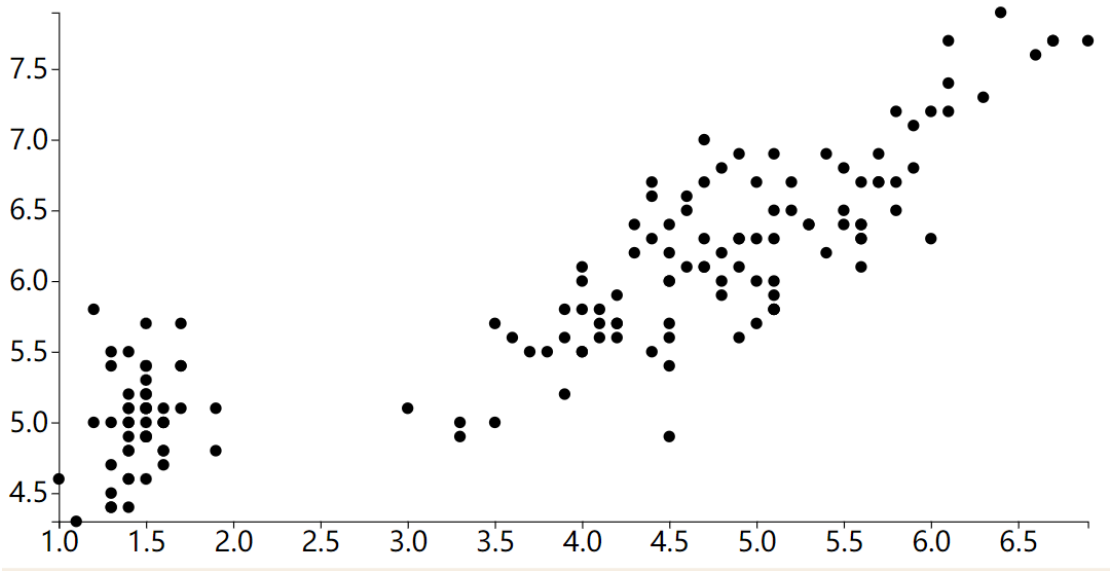
```
import React, { useState, useCallback, useEffect } from 'react';
import ReactDOM from 'react-dom';
import { select, axisLeft, axisBottom, csv, arc, pie, scaleBand, scaleLinear, max, format} from 'd3';

const csvUrl =
  'https://gist.githubusercontent.com/haili0109/ea79c34fd1a2957ce4f1f38a7692eff4/raw/Top10Population2019.csv'
const width = 960;
const height = 500;
const margin = { top: 20, right: 20, bottom: 50, left: 200 };



const App = () => {
  const [data, setData] = useState(null);

  useEffect(() => {
    const row = d => {
      d.Population = +d['2019'];
      return d;
    };
    csv(csvUrl, row).then(data => {
      setData(data.slice(0, 10));
    });
  }, []);

  if (!data) {
    return <pre>Loading...</pre>;
  }

  const innerHeight = height - margin.top - margin.bottom - 100;
  const innerWidth = width - margin.left - margin.right;

  const yScale = scaleBand()
    .domain(data.map(d => d.Country))
    .range([0, innerHeight]);

  const xScale = scaleLinear()
    .domain([0, max(data, d => d.Population)])
    .range([0, innerWidth]);

  return (
    <svg width={width} height={height}>
      <g transform={`translate(${margin.left},${margin.top})`}>
        {xScale.ticks().map(tickValue => (
          <g key={tickValue} transform={`translate(${xScale(tickValue)},0)`}>
            <line y2={innerHeight} stroke="blue" />
            <text
              style={{ textAnchor: 'middle' }}
              dy=".70em"
              y={innerHeight + 3}
            >
              {tickValue}
            </text>
          </g>
        ))}
        {yScale.domain().map(tickValue => (
          <text
            key={tickValue}
            style={{ textAnchor: 'end' }}
            x={-3}
            dy=".32em"
            y={yScale(tickValue) + yScale.bandwidth() / 2}
          >
            {tickValue}
          </text>
        ))}
        {data.map(d => (
          <rect
            key={d.Country}
            x={0}
            y={yScale(d.Country)}
            width={xScale(d.Population)}
            height={yScale.bandwidth()}
          />
        ))}
      </g>
    </svg>
  );
};
const rootElement = document.getElementById('root');
ReactDOM.render(<App />, rootElement);
```



Try to stylish the bar chart and submit the VizHub link.

- **Creating a Scatter Plot with D3.js**

Use Iris Dataset as an example for this lab:
https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv



index.html

```html
<!DOCTYPE html>
<html>
  <head>
    <title>Iris Dataset Scatter Plot</title>
    <style>
      body {
        margin: 0;
        overflow: hidden;
      }
      .tick text {
        font-size: 24px;
      }
    </style>
    <script src="https://unpkg.com/d3@6.7.0/dist/d3.min.js"></script>
  </head>
  <body>
    <script>
const {csv, select, scaleLinear, extent, axisLeft, axisBottom, } = d3;
const csvUrl =
'https://gist.githubusercontent.com/curran/a08a1080b88344b0c8a7/raw/0e7a9b0a5d22642a06d3d5b9bcbad9890c8ee534/iris.csv'
      const parseRow = (d) => {
        d.sepal_length = +d.sepal_length;
        d.sepal_width = +d.sepal_width;
        d.petal_length = +d.petal_length;
        d.petal_width = +d.petal_width;
        return d;
      };  //LOAD THE DATA
      const xValue = (d) => d.petal_length;

      const yValue = (d) => d.sepal_length;

      const margin = {
        top: 20,
        right: 20,
        bottom: 40,
        left: 50,
      };
      const radius = 5;

      const width = window.innerWidth;
      const height = window.innerHeight;
      const svg = select('body')
        .append('svg')
        .attr('width', width)
        .attr('height', height);
      const main = async () => {
        const data = await csv(csvUrl, parseRow);
        const x = scaleLinear()
          .domain(extent(data, xValue))
          .range([margin.left, width - margin.right]);
        const y = scaleLinear()
          .domain(extent(data, yValue))
          .range([height - margin.bottom, margin.top]);
        const marks = data.map((d) => ({
          x: x(xValue(d)),
          y: y(yValue(d)),
        }));
        svg
          .selectAll('circle')
          .data(marks)
          .join('circle')
          .attr('cx', (d) => d.x)
          .attr('cy', (d) => d.y)
          .attr('r', radius);
        svg
          .append('g')
          .attr('transform', `translate(${margin.left},0)`)
          .call(axisLeft(y));
        svg
          .append('g')
          .attr(
            'transform',
            `translate(0,${height - margin.bottom})`
          )
          .call(axisBottom(x));
      };
      main();
    </script>
  </body>
</html>
```

Please add labels for x and y axis in this scatter plot.

Submit the link for this code.