Visualization of FIFA 19 Player Attributes

Objective :
To Visualize the player attributes of FIFA - 19.

Data Set :
We are using a dataset that contains the in-game attributes of football players from the computer video game FIFA-19.
Data source - https://www.kaggle.com/karangadiya/fifa19/data

Data Attributes :

Name	**Type**	**What it means**
Category	Categorical	FIFA approved player ranking category
Market Value	Numerical	Transfer market value of player
Weekly Wages	Numerical	Wages earned on a weekly basis
Finishing	Numerical	Clinicality in converting a final move into goal
Crossing	Numerical	Crossing or Long aerial pass
Acceleration	Numerical	Sprint acceleration of player
Shooting	Numerical	Shooting skill from distance
Passing	Numerical	Passing long and short
Dribbling	Numerical	Dribbling skill
International Fame	Categorical	Reputation of player across globe
Work Rate (Att \ Def)	Categorical	Distance covered per game
Nationality	Categorical	Country player belongs to
Age	Numerical	Age
Skill	Categorical	Freestyle skills of player
Position	Categorical	Preferred position of player

Data Cleansing and Transformation :

- Raw file was read using python to remove redundant fields
- Missing data was eliminated by taking mean of neighbouring data points

Project :

Full Video Demonstration is available here.
The project makes use of d3(data driven documents) library to dynamically create a bar chart(categorical) or histogram(numerical) visualization of player attributes.

How to run Project :

1. Unzip the source code file named (vizlab-1.zip)
2. Install npm (version 6 or more) and node (v13)
3. Run npm install to download required packages
4. Run npm run build
5. Run npm run start to start server
6. Open localhost:8080 to access the visualization

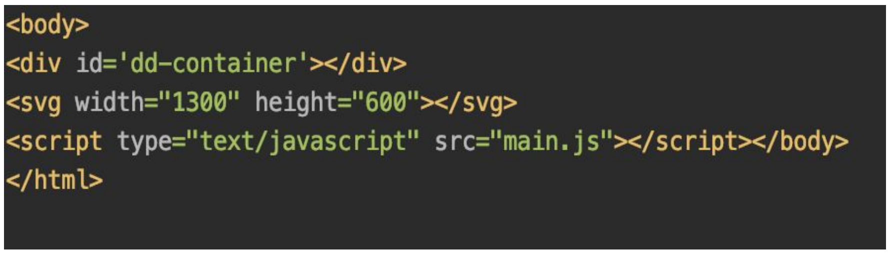
Project files :

1. data.json
2. index.html which contains html code
3. index.js which contains javascript code
4. styles.css which contains style specifications and css

Code Implementation

Canvas Specifications

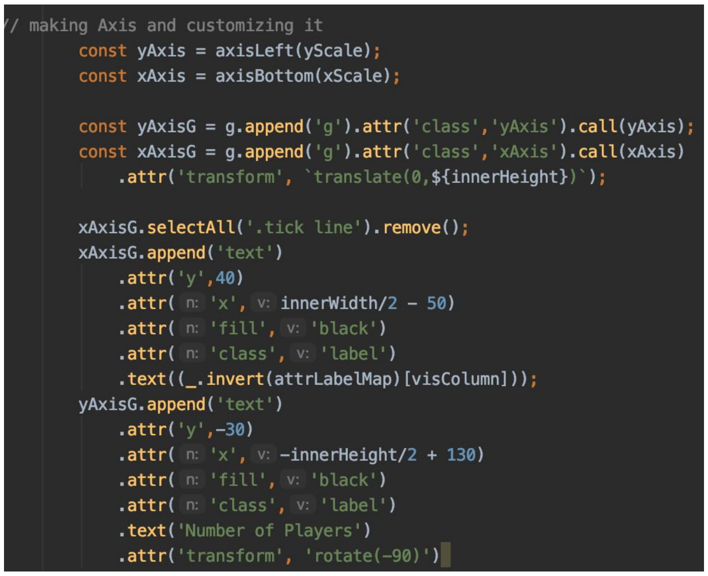
- We have chosen width as 1300 px and height as 600 px
- We have also added a div with id = ‘dd-container’ to place our dropdown



```
<body>
<div id='dd-container'></div>
<svg width="1300" height="600"></svg>
<script type="text/javascript" src="main.js"></script></body>
</html>
```

Update Bars

- Function which handles the selection changes on our drop down
- Update bars creates a Bar chart or Histogram depending on attribute chosen
- Makes x-axis scales, y-axis scales and build axis
- Customize axes Labels to show correct attributes



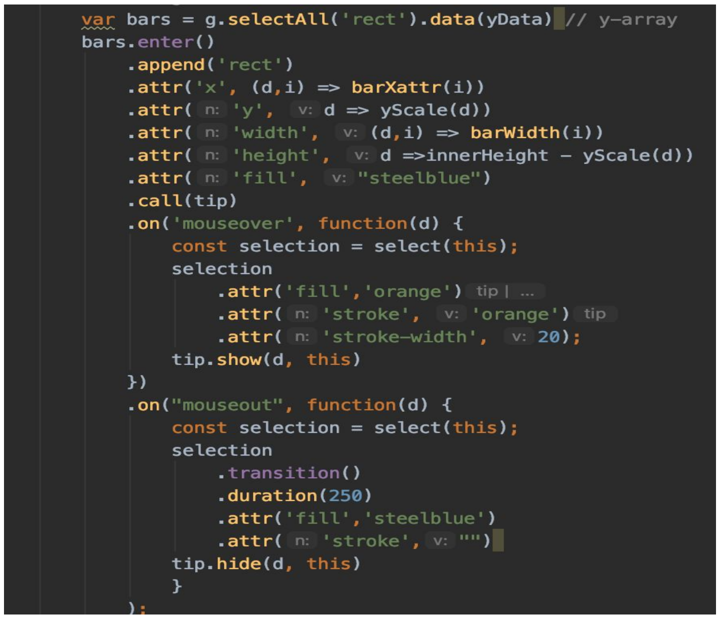
```
// making Axis and customizing it
      const yAxis = axisLeft(yScale);
      const xAxis = axisBottom(xScale);

      const yAxisG = g.append('g').attr('class','yAxis').call(yAxis);
      const xAxisG = g.append('g').attr('class','xAxis').call(xAxis)
         .attr('transform', `translate(0,${innerHeight})`);

      xAxisG.selectAll('.tick line').remove();
      xAxisG.append('text')
         .attr('y',40)
         .attr( n: 'x', v: innerWidth/2 - 50)
         .attr( n: 'fill', v: 'black')
         .attr( n: 'class', v: 'label')
         .text((_.invert(attrLabelMap)[visColumn]));
      yAxisG.append('text')
         .attr('y',-30)
         .attr( n: 'x', v: -innerHeight/2 + 130)
         .attr( n: 'fill', v: 'black')
         .attr( n: 'class', v: 'label')
         .text('Number of Players')
         .attr('transform', 'rotate(-90)')
```

D3 Rect Elements

- We have added attributes height, width, fill for our rect elements
- Mouse events ‘mouseover’ and ‘mouseout’ change fill and zoom in on the bar
- Upon Hover, we are showing the value of the bar



```
    var bars = g.selectAll('rect').data(yData) // y-array
    bars.enter()
        .append('rect')
        .attr('x', (d,i) => barXattr(i))
        .attr( n: 'y',  v: d => yScale(d))
        .attr( n: 'width',  v: (d,i) => barWidth(i))
        .attr( n: 'height',  v: d =>innerHeight - yScale(d))
        .attr( n: 'fill',  v: "steelblue")
        .call(tip)
        .on('mouseover', function(d) {
            const selection = select(this);
            selection
                .attr('fill','orange') tip | ...
                .attr( n: 'stroke',  v: 'orange') tip
                .attr( n: 'stroke-width',  v: 20);
            tip.show(d, this)
        })
        .on("mouseout", function(d) {
            const selection = select(this);
            selection
                .transition()
                .duration(250)
                .attr('fill','steelblue')
                .attr( n: 'stroke', v: "")
            tip.hide(d, this)
            }
        );
```