VISUALIZATION MINI PROJECT I REPORT
SAKSHI GUPTA-112552239

DATASET

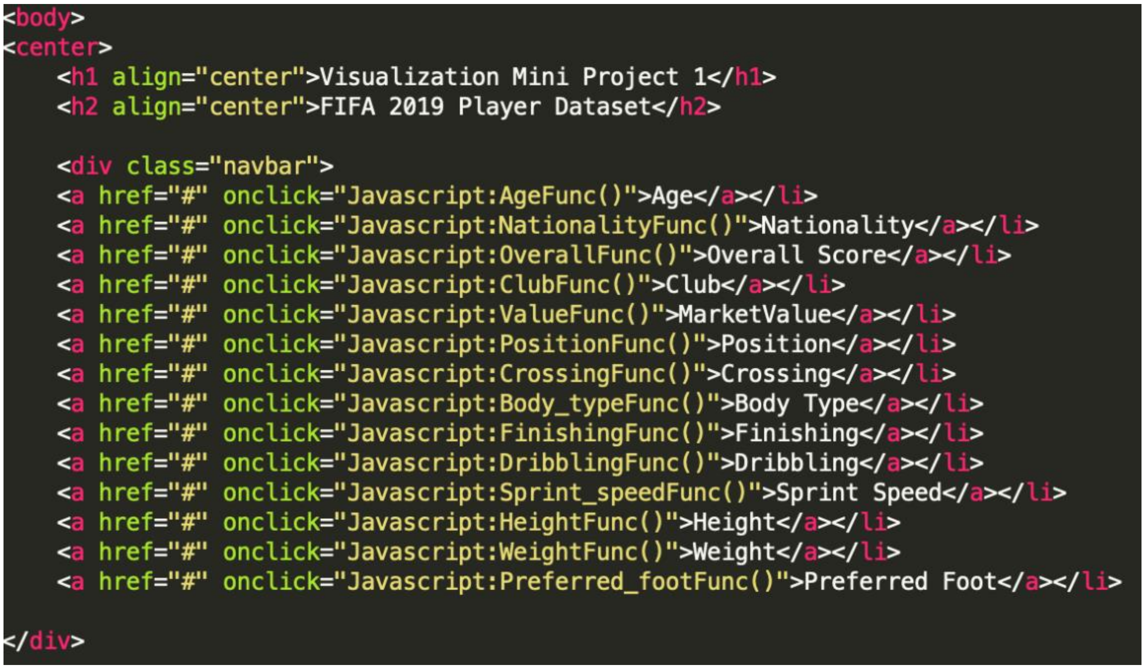
The dataset that I chose for my assignment is the FIFA 2019 Player Dataset. This dataset contains all the relevant information about European Football players that participated in FIFA World Cup 2019. After combining and filtering columns based on relevance to the assignment, I picked out the final 16 attributes which were:

1. Player ID
2. Player Name
3. Age
4. Nationality
5. Overall Rating/Score of the player
6. Football Club that the player belongs to
7. Market Value of the player
8. Position at which a player plays
9. Crossing score
10. Body Type
11. Finishing score
12. Dribbling score
13. Sprint Speed score
14. Height
15. Weight
16. Preferred foot

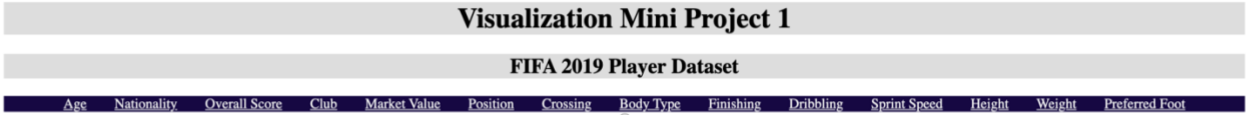
Out of these I left out ID and Name as their plot wasn't showcasing any important information.
Categorical attributes: Nationality, Football Club, Position, Body Type, Height, Preferred foot.
Numerical attributes: Age, Overall Score, Market Value, Crossing, Finishing, Dribbling, Sprint Speed, Weight.
I have used JavaScript, HTML, CSS and d3 version 4 for this assignment. Also, I used XAMPP as the server to host the website.

ASSIGNMENT TASKS

- **Present a menu to allow users to select a variable and update chart:**
  For this task I made a Navigation bar at the top below the headers with my 14 column names. Once the user clicks on any one button on the bar, the corresponding chart is displayed.



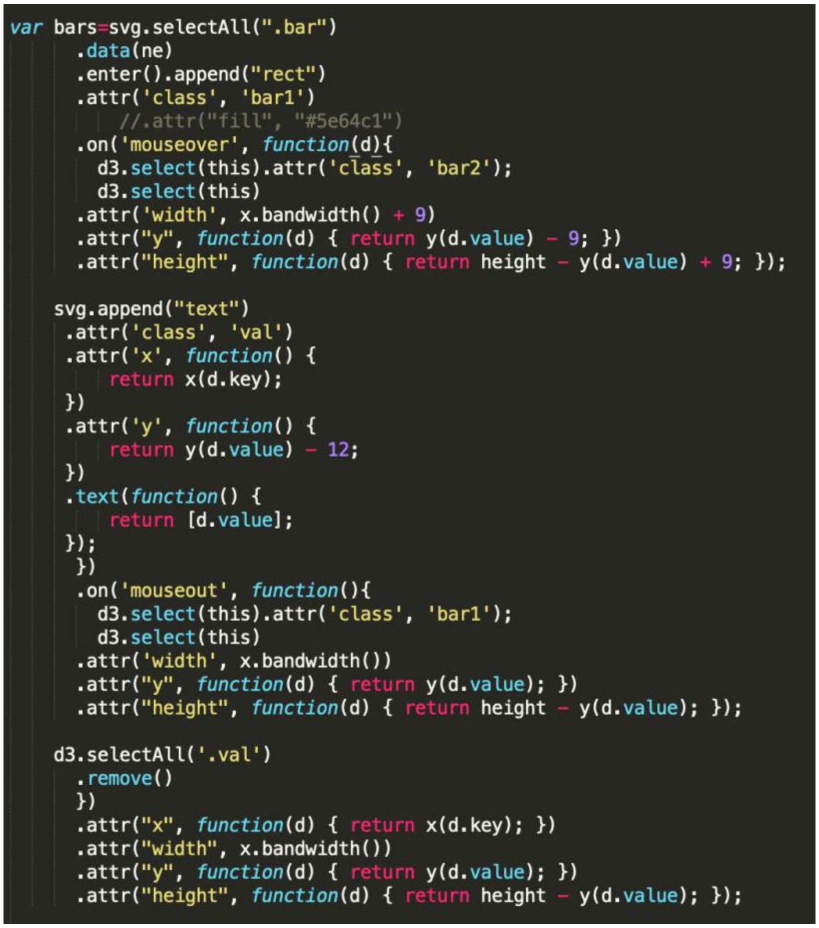
```
<body>
<center>
    <h1 align="center">Visualization Mini Project 1</h1>
    <h2 align="center">FIFA 2019 Player Dataset</h2>

    <div class="navbar">
    <a href="#" onclick="Javascript:AgeFunc()">Age</a></li>
    <a href="#" onclick="Javascript:NationalityFunc()">Nationality</a></li>
    <a href="#" onclick="Javascript:OverallFunc()">Overall Score</a></li>
    <a href="#" onclick="Javascript:ClubFunc()">Club</a></li>
    <a href="#" onclick="Javascript:ValueFunc()">MarketValue</a></li>
    <a href="#" onclick="Javascript:PositionFunc()">Position</a></li>
    <a href="#" onclick="Javascript:CrossingFunc()">Crossing</a></li>
    <a href="#" onclick="Javascript:Body_typeFunc()">Body Type</a></li>
    <a href="#" onclick="Javascript:FinishingFunc()">Finishing</a></li>
    <a href="#" onclick="Javascript:DribblingFunc()">Dribbling</a></li>
    <a href="#" onclick="Javascript:Sprint_speedFunc()">Sprint Speed</a></li>
    <a href="#" onclick="Javascript:HeightFunc()">Height</a></li>
    <a href="#" onclick="Javascript:WeightFunc()">Weight</a></li>
    <a href="#" onclick="Javascript:Preferred_footFunc()">Preferred Foot</a></li>

</div>
```



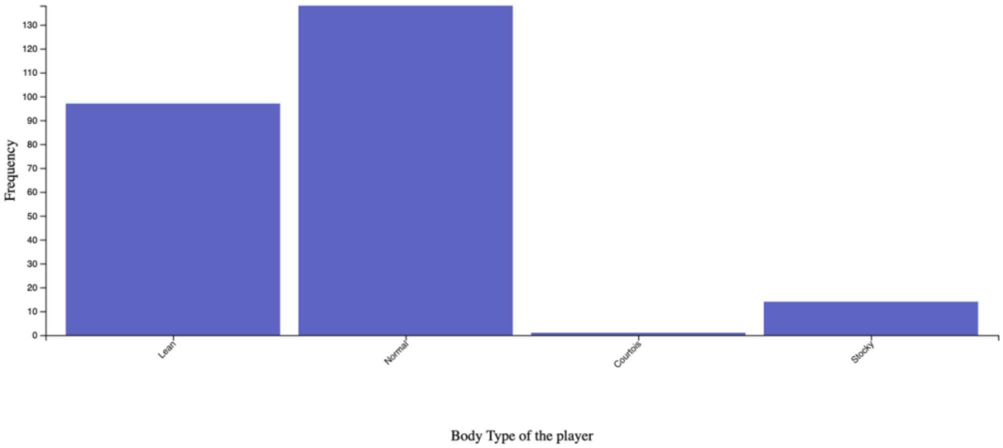
Visualization Mini Project 1

FIFA 2019 Player Dataset

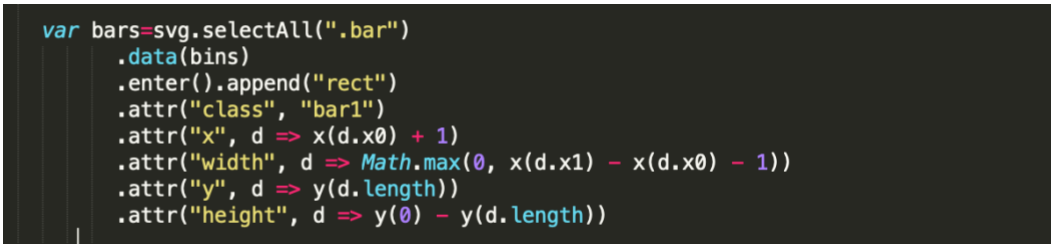
- **Draw a bar chart if a categorical variable is selected:**
  When the user selects any of the 6 categorical attributes, a bar graph is displayed on the screen. For this purpose, I wrote a separate bar JavaScript which I call for these 6 categorical values. For e.g., if the user clicks on Body Type, then the following graph is displayed. I plotted the different values of a particular column on the x axis and used d3.nest() to group the data together and count the frequency of each value over the dataset which I plotted on the y-axis.



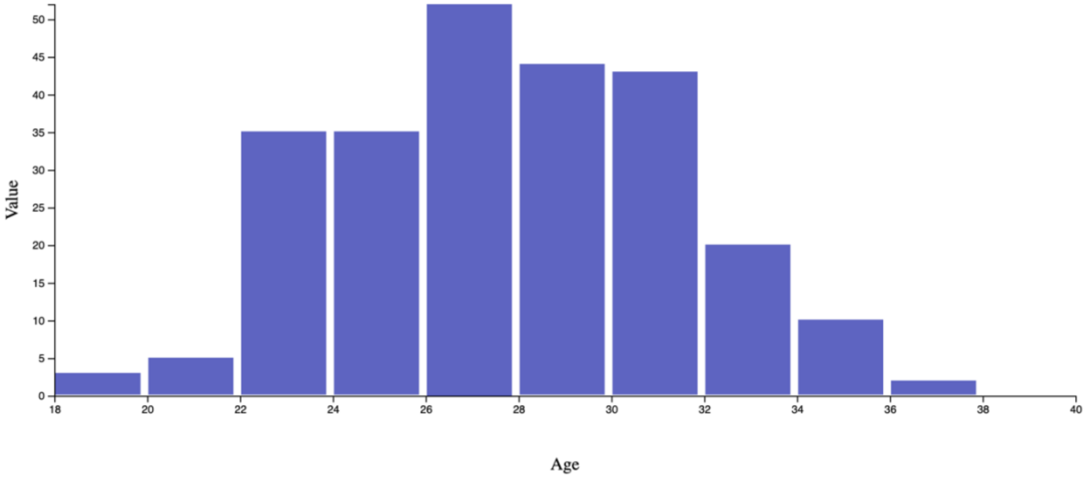
```
var bars=svg.selectAll(".bar")
      .data(ne)
      .enter().append("rect")
      .attr('class', 'bar1')
          //.attr("fill", "#5e64c1")
      .on('mouseover', function(d){
        d3.select(this).attr('class', 'bar2');
        d3.select(this)
      .attr('width', x.bandwidth() + 9)
      .attr("y", function(d) { return y(d.value) - 9; })
      .attr("height", function(d) { return height - y(d.value) + 9; });

    svg.append("text")
     .attr('class', 'val')
     .attr('x', function() {
         return x(d.key);
     })
     .attr('y', function() {
         return y(d.value) - 12;
     })
     .text(function() {
         return [d.value];
     });
      })
      .on('mouseout', function(){
        d3.select(this).attr('class', 'bar1');
        d3.select(this)
      .attr('width', x.bandwidth())
      .attr("y", function(d) { return y(d.value); })
      .attr("height", function(d) { return height - y(d.value); });

    d3.selectAll('.val')
      .remove()
      })
      .attr("x", function(d) { return x(d.key); })
      .attr("width", x.bandwidth())
      .attr("y", function(d) { return y(d.value); })
      .attr("height", function(d) { return height - y(d.value); });
```



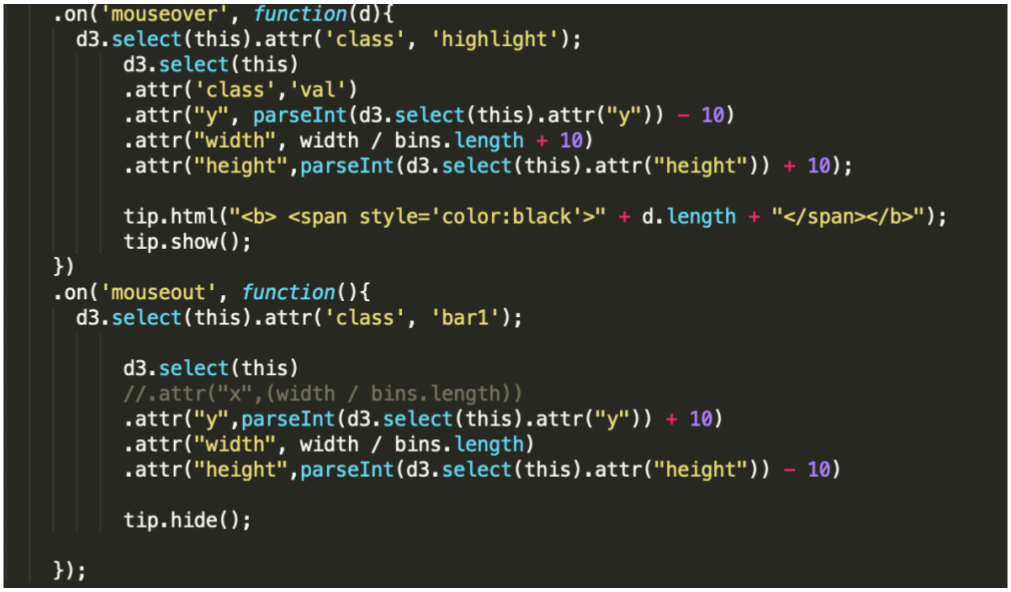
- **Draw a histogram if a numerical variable is selected:**
  When the user selects any of the 8 numerical attributes, a bar graph is displayed on the screen. For this purpose, I wrote a separate hist JavaScript which I call for these 8 numerical values. For e.g., if the user clicks on Age, then the following graph is displayed. I plotted the bins that I created (default=15) on the x-axis and I used the length() to count the frequency count of the each bin which I plotted on the y-axis.



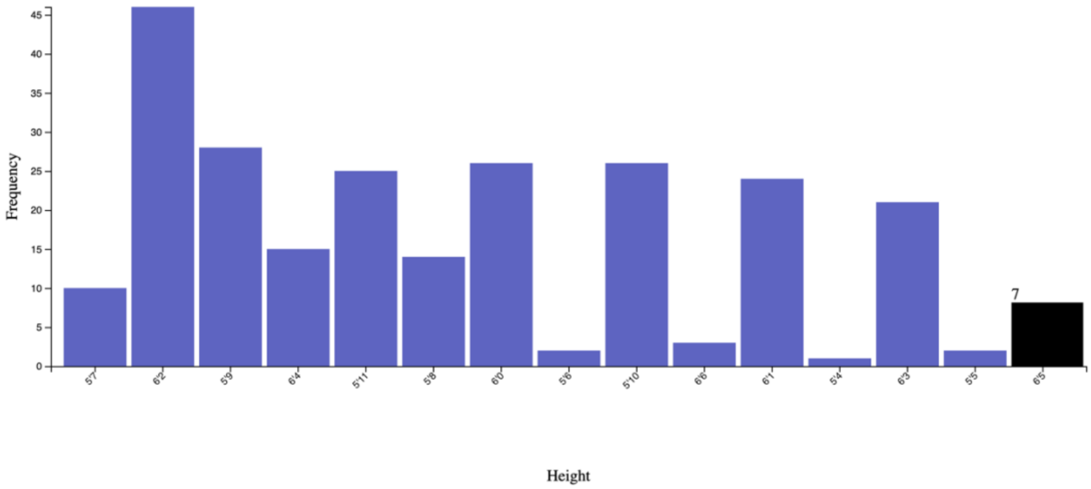
```
var bars=svg.selectAll(".bar")
        .data(bins)
        .enter().append("rect")
        .attr("class", "bar1")
        .attr("x", d => x(d.x0) + 1)
        .attr("width", d => Math.max(0, x(d.x1) - x(d.x0) - 1))
        .attr("y", d => y(d.length))
        .attr("height", d => y(0) - y(d.length))
```



- **On mouse-over display the value of the bar on top of the bar:**
  When the user hovers the mouse cursor over a bar, then the frequency count or value of that particular attribute value is displayed on top of the bar. For this purpose, the ‘mouseover’ and ‘mouseout’ mouse handling events are used.

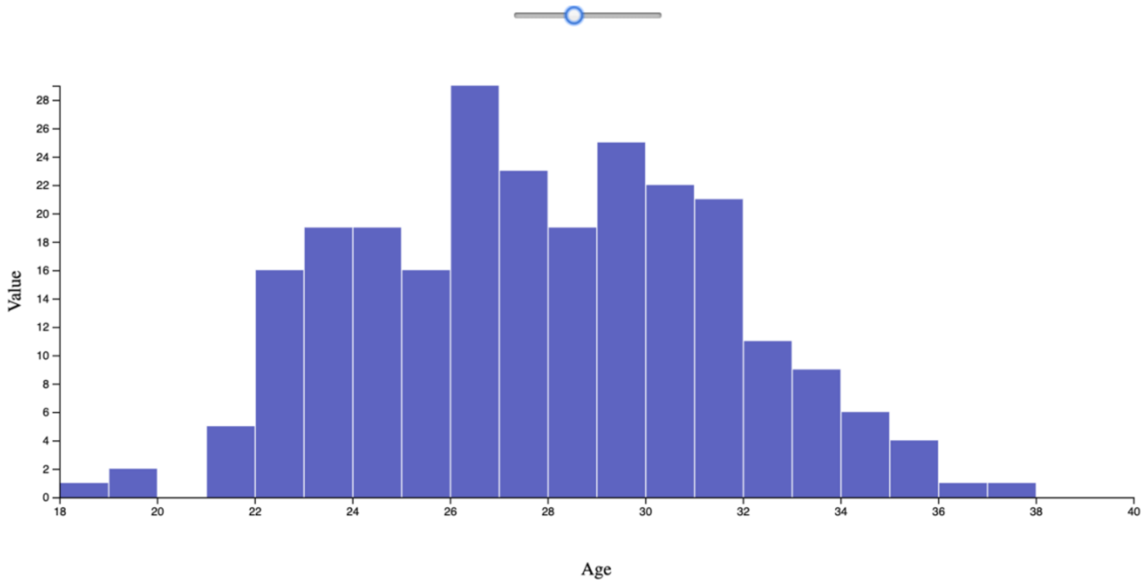


```
    .on('mouseover', function(d){
      d3.select(this).attr('class', 'highlight');
          d3.select(this)
          .attr('class','val')
          .attr("y", parseInt(d3.select(this).attr("y")) - 10)
          .attr("width", width / bins.length + 10)
          .attr("height",parseInt(d3.select(this).attr("height")) + 10);

          tip.html("<b> <span style='color:black'>" + d.length + "</span></b>");
          tip.show();
    })
    .on('mouseout', function(){
      d3.select(this).attr('class', 'bar1');

          d3.select(this)
          //.attr("x",(width / bins.length))
          .attr("y",parseInt(d3.select(this).attr("y")) + 10)
          .attr("width", width / bins.length)
          .attr("height",parseInt(d3.select(this).attr("height")) - 10)

          tip.hide();

    });
```



- **On mouse-over also make the bar wider and higher to focus on it:**
  When the user hovers the mouse cursor over a bar in the graph then the bar becomes more focused and its color changes to make it more prominent for the user to see. This is also done using event handlers by increasing the width and height attributes of the ‘svg’ element in the ‘mouseover’ event handler and changing it back to the original in the ‘mouseout’ event handler. Example is shown above.

- **Mouse (with left mouse button down) move left (right) should decrease (increase) bin width/size:**
  For this I made a slider for the range of bins that I kept from 0-30. The selected value is displayed on the screen and when the user slides, the number of bins change as per the selected value. The slider works only for the histograms i.e. only for the numerical values. In my software, dragging the slider to the left increases the number of bins i.e. decreases the bin width and dragging it to the right decreases the number of bins i.e. increases the bin width.



YOUTUBE LINK

https://youtu.be/O-40hkKb51Q

ZIP FILE CONTENTS

- Index.html
- Index.js
- Bar.js
- Hist.js
- changeBins.js
- index_css.css
- new.csv
- Report PDF

REFERENCES

- www.w3schools.com
- www.stackoverflow.com
- www.tutorialsteacher.com