

CSE564: Visualization LAB Assignment 1  
Spring 21

Name- Mayank Jain  
SBU Id- 113263864

Demo Video Link: <https://drive.google.com/file/d/1ICIm0OQPdCvTgmlU-GKxPYD7Krp7QKGr/view?usp=sharing>

Tasks to be implemented:

1. Present a menu to allow users to select a variable and update chart.
2. Draw a bar chart if a categorical variable is selected.
3. Draw a histogram if a numerical variable is selected.
4. On mouse-over display the bar's value above the bar in red color.
5. On mouse-over also color the bar in red to highlight it.
6. Mouse (with left mouse button down) move left (right) should decrease (increase) bin width/size (for numerical variables only).
7. Produce a scatterplot of two selected variables (use a toggle to determine which of the two variable axes is to be loaded).

## 1. Dataset:

The dataset that I have chosen to work on this assignment is the FIFA 21 player's data which contains every possible detail of a player which could be of interest to this sport's or game's fans.

In the original data there are more than 17k players which have been analyzed based on total of 107 parameters. But, for this assignment I have chosen top **799** players according to their overall rating and total **18** attributes to plot on the graphs. As we were asked to have mix of both categorical and numerical variables, so I have picked :

4 Categorical attributes- Nationality, Playing Position, Club and Foot

14 Numerical Variables- Overall Rating, Age, Height, Weight, Pace, Shot, Pass, Defend, Physical, Goalkeeping, Penalties, Crossing, Finishing and Dribbling.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
1	ID	Name	Age	OVA	Nationality	Club	PlayingPosition	Height	Weight	foot	Value	Wage	Pace	Shot	Pass	Defend	Physical	Goalkeeping	Penalties	Crossing	Finishing	Dribbling	
2	0	158023	L. Messi	33	93	Argentina	FC Barcelona	RW	5.58	159	Left	67000000	560000	85	92	91	38	65	54	75	85	95	96
3	1	20801	Cristiano Ronaldo	35	92	Portugal	Juventus	ST	6.17	183	Right	46000000	220000	89	93	81	35	77	58	84	84	95	88
4	2	188545	R. Lewandowski	31	91	Poland	FC Bayern München	ST	6.0	176	Right	80000000	240000	78	91	78	43	82	51	88	71	94	85
5	3	190871	Neymar Jr	28	91	Brazil	Paris Saint-Germain	LW	5.75	150	Right	90000000	270000	91	85	86	36	59	59	92	85	87	95
6	4	192985	K. De Bruyne	29	91	Belgium	Manchester City	CAM	5.92	154	Right	87000000	370000	76	86	93	64	78	56	84	94	82	88
7	5	200389	J. Obiakor	27	91	Slovenia	Atlético Madrid	GK	6.17	192	Right	75000000	125000	87	92	78	52	90	437	11	13	11	12
8	6	192448	M. ter Stegen	28	90	Germany	FC Barcelona	GK	6.17	187	Right	69000000	260000	88	85	88	45	88	439	25	18	14	21
9	7	203376	V. van Dijk	28	90	Netherlands	Liverpool	CB	6.33	203	Right	75000000	210000	76	60	71	91	86	58	62	53	52	70
10	8	208722	S. Mané	28	90	Senegal	Liverpool	LW	5.75	152	Right	78000000	250000	94	85	80	44	76	56	71	76	90	91
11	9	209331	M. Salah	28	90	Egypt	Liverpool	RW	5.75	157	Left	78000000	250000	93	86	81	45	75	62	83	79	91	90
12	10	212831	Alisson	27	90	Brazil	Liverpool	GK	6.25	201	Right	62000000	160000	86	88	85	51	91	439	23	17	13	27
13	11	231747	K. Mbappé	21	90	France	Paris Saint-Germain	ST	5.83	161	Right	105000000	160000	96	86	78	39	76	42	70	78	91	92
14	12	153079	S. Agüero	32	89	Argentina	Manchester City	ST	5.67	154	Right	53000000	300000	78	90	77	33	73	59	75	70	94	88
15	13	155862	Sergio Ramos	34	89	Spain	Real Madrid	CB	6.0	181	Right	24000000	300000	71	70	76	88	85	46	92	66	65	65
16	14	165153	K. Benzema	32	89	France	Real Madrid	CF	6.08	179	Right	53000000	350000	74	85	81	40	76	41	84	75	88	87
17	15	167495	M. Neuer	34	89	Germany	FC Bayern München	GK	6.33	203	Right	29000000	125000	87	87	91	57	86	440	47	15	13	30
18	16	192119	T. Courtois	28	89	Belgium	Real Madrid	GK	6.5	212	Left	56000000	250000	84	89	74	48	85	420	27	14	14	13
19	17	200145	Casemiro	28	89	Brazil	Real Madrid	CDM	6.08	185	Right	59000000	310000	65	73	76	86	91	67	66	58	64	69
20	18	121939	P. Lahn	32	88	Germany	FC Bayern München	RWB	5.58	146	Right	29000000	200000	67	56	82	86	64	47	69	84	47	82
21	19	162835	S. Handanović	35	88	Slovenia	Inter	GK	6.33	203	Right	16000000	100000	88	85	73	53	89	424	23	12	10	18
22	20	182521	T. Kroos	30	88	Germany	Real Madrid	CM	6.0	168	Right	55000000	310000	54	81	91	71	69	51	73	88	76	80
23	21	183277	E. Hazard	29	88	Belgium	Real Madrid	LW	5.75	163	Right	58000000	350000	88	82	83	35	66	45	87	77	82	93
24	22	201024	K. Koulibaly	29	88	Senegal	Napoli	CB	6.17	196	Right	50000000	140000	75	28	55	89	86	43	33	30	22	69
25	23	202126	H. Kane	26	88	England	Tottenham Hotspur	ST	6.17	196	Right	71000000	220000	68	91	80	47	83	54	90	75	94	80
26	24	202652	R. Sterling	25	88	England	Manchester City	LW	5.58	152	Right	72000000	270000	93	81	79	45	67	63	69	78	85	90
27	25	210257	Ederson	26	88	Brazil	Manchester City	GK	6.17	190	Left	53000000	195000	86	82	93	63	86	435	17	20	14	23
28	26	211110	R. Dybala	26	88	Argentina	Juventus	CAM	5.83	165	Left	71000000	190000	85	85	84	43	63	26	86	82	84	91
29	27	212622	J. Kimmich	25	88	Germany	FC Bayern München	CDM	5.75	161	Right	65000000	145000	71	72	86	81	79	60	44	91	66	83
30	28	215914	N. Kanté	29	88	France	Chelsea	CDM	5.5	154	Right	51000000	190000	77	66	76	86	82	54	54	68	65	79
31	29	488	O. Kahn	38	87	Germany	FC Bayern München	GK	6.17	198	Right	0	0	83	88	62	60	92	346	21	21	21	21
32	30	138956	G. Chiellini	35	87	Italy	Juventus	CB	6.17	187	Left	15000000	94000	66	46	58	90	79	15	50	54	33	59

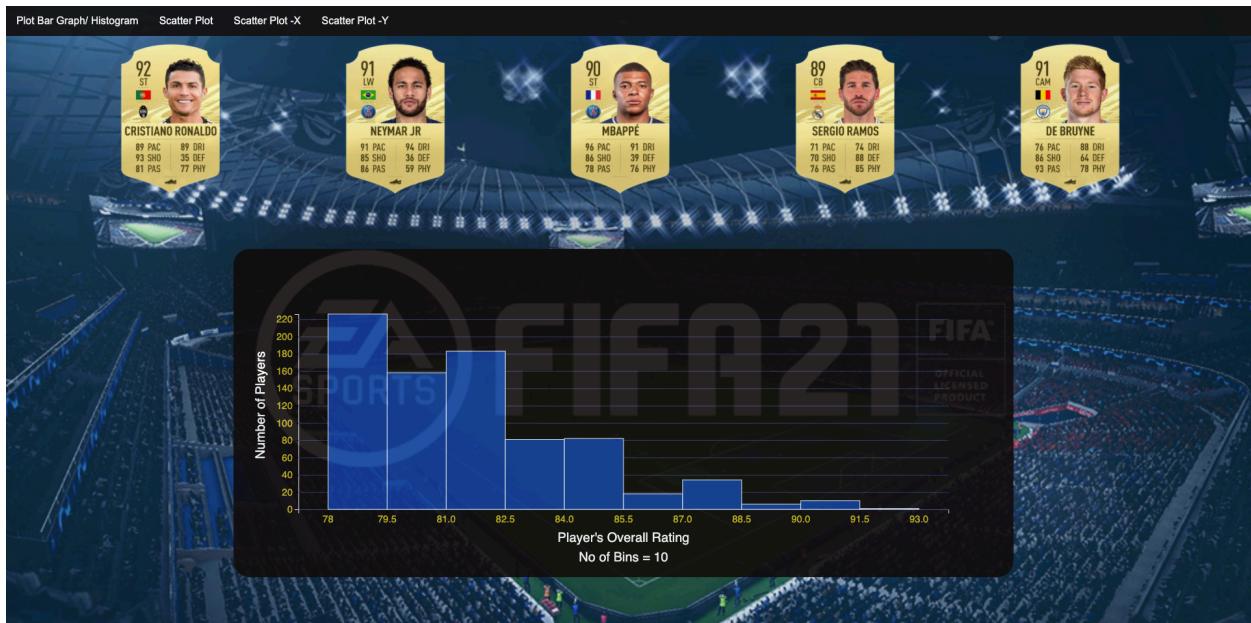
## 2. Data Preprocessing:

The data did not had any missing values and could be used directly so not much of data preprocessing was involved. I just had to do some filtering in the excel sheet and little bit of python scripting to be able to read the CSV file in my javascript code.

Python Script used: dataclean.py

Dataset CSV file name: data.csv

### 3. Web page design:



To make the UI for plotting the graphs and scatter plot I used HTML and CSS.

The page design consists of one menu bar provided on top of the webpage which could be used to plot various charts by clicking on the buttons and the drop-downs provided.

By default when the page is loaded there is a Histogram present which is showing Count of players with certain ratings.

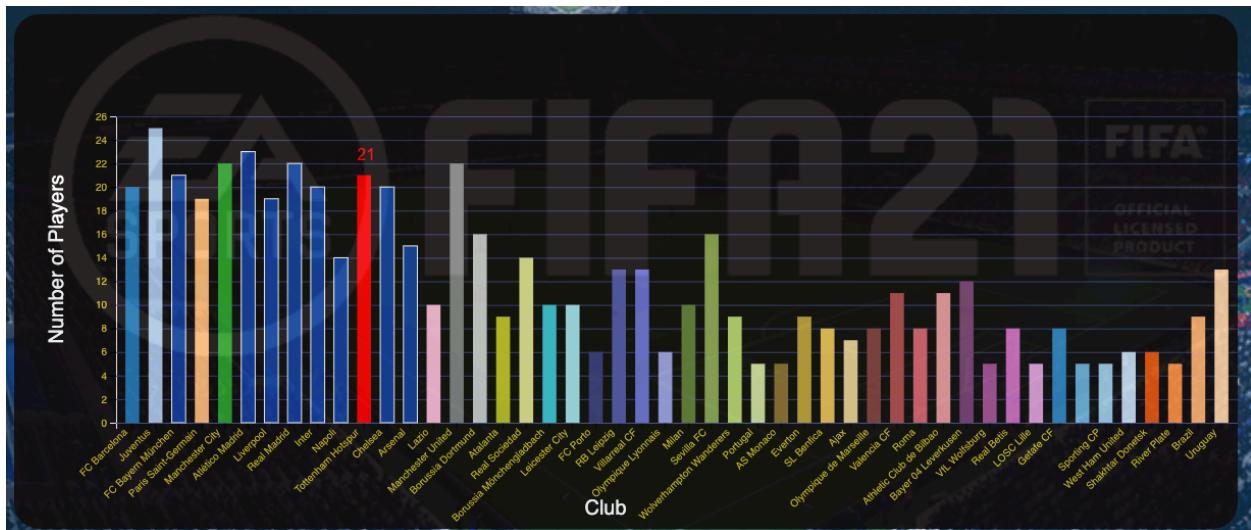
From the first drop-down labelled as “Plot Bar Graph/ Histogram”, if the user selects a categorical variable then the BarChart is drawn and if the user selects a numerical variable then the Histogram is visible on the screen.

There is one more button present in the top navbar which consists of one button labelled as “Scatter Plot” which when clicked will present a default scatter plot between Age and Pace of a player. Then to select the variables which the user wants to plot, two more options are provided to select X axis and Y axis attribute.

All the plots that are made are built in a svg container which is present in the centre of the screen using d3.js version 4 functions.

File names: Homepage.html, chart.css and style.css

#### 4. Bar Graph :



From the drop down menu if the categorical variable is selected then the Bar-graph appears on the screen. This is basically to show the number of players having that attribute, like the above screenshot represents number of players belonging to a specific club present in Fifa 21.

Code flow: Once the user clicks on a specific attribute, then a function call is triggered specific to that attribute which is present in the helper.js file. This function in turn calls main drawBarGraph(data,toPlot) present in BarGraph\_categorical.js file which takes the input attribute name and the data corresponding to that attribute in the form of an array.

This function first converts the input array to a Json object which will have key as the attribute name and value as the count of players belonging to that class. After we have the Json object we can use it to plot the bar graph using d3 functions.

Main code snippets responsible for plotting the graph:

```

g.selectAll(".bar")
    .data(res).enter().append("rect")
        .on("mouseover", onMouseOver)
        .on("mouseout", onMouseOut)
    .attr("x", function(d, i) { return x(d.xval); })
        .attr("width", x.bandwidth())
    .transition()
        .ease(d3.easeLinear)
    .duration(200)
    .delay(function (d, i) {
        return i * 50;
    })
    .attr("height", function(d, i) { return height -y(d.yval); })
    .attr("x", function(d, i) { return x(d.xval); })
    .attr("y", function(d, i) { return y(d.yval); })
    .attr("fill", function(d, i) { return colors[i]; });

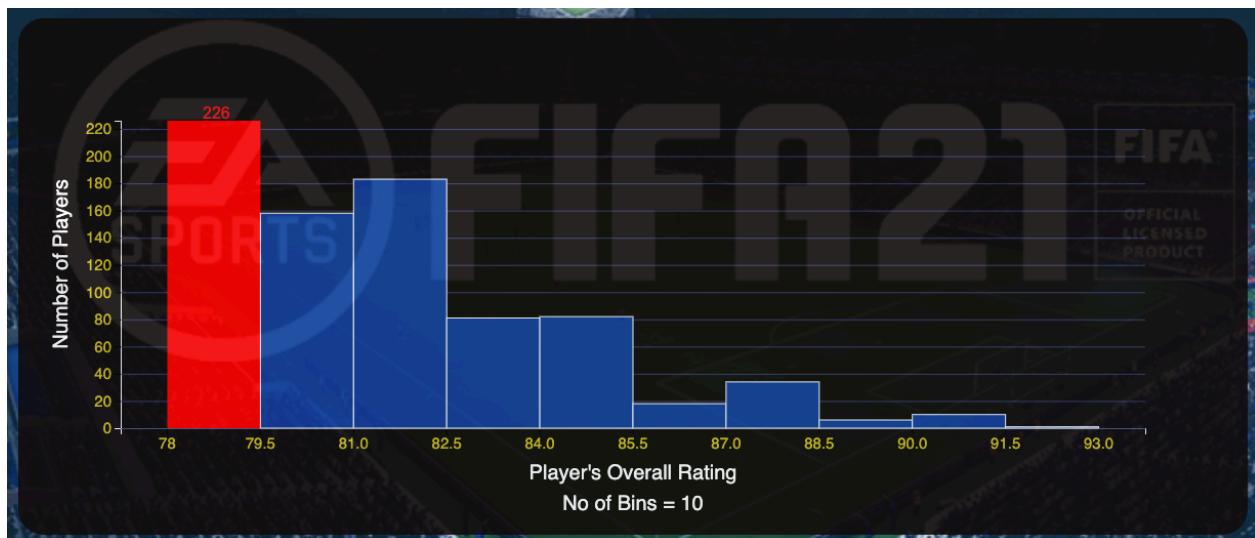
```

Functionalities implemented:

- Draw a bar chart if a categorical variable is selected.
- Highlight the bar in red color on MouseOver.
- Highlight the count in red color of that bar.

Files involved in the flow: helper.js, BarGraph\_categorical.js

## 5. Histogram:



From the drop down menu if the numerical variable is selected then the Histogram appears on the screen. This is basically to show the number of players with a certain attribute belonging in a specific range. By default the number of bins will be 10.

The number of bins can be increased or decreased dynamically by clicking the mouse in the chart area and dragging to left or right with mouse down. Also, to see the value of that bin the user can move the mouse to that bin and it will be highlighted in red color with its value present above the bin.

Code flow: Once the user clicks on a numerical attribute, then a function call is triggered specific to that attribute which is present in the helper.js file. This function in turn calls the main drawHistogram(data, noOfBins, toPlot) function present in Histogram\_Numerical.js file which takes the input attribute name, array data and default number of bins.

This function first calculates the bin width, the range of that attribute to plot in each bin and calculates the count of players belonging in that range.

After that similar to Bar-graph, first I plot the axes and the values which will be present on them and then plot the histogram.

Main code snippets responsible for plotting the graph:

```
g.selectAll(".bar")
  .data(Range_Xaxis)
  .enter().append("rect")
  .attr("class", "bar")
  .on("mouseover", onMouseOver)
  .on("mouseout", onMouseOut)
  .attr("x", function(d) { return x(d)+(x.bandwidth()/2) ; })
  .attr("y", height)
  .attr("height", 0)
  .attr("width", x.bandwidth())
  .transition()
  .ease(d3.easeLinear)
  .duration(600)
  .attr("y", function(d,i) { return y(Value_Yaxis[i]); })
  .attr("height", function(d,i) { return height - y(Value_Yaxis[i]); });


```

After the histogram is plotted, I worked on implementing mousedown and mousemove functionality to change the number of bins and replot the graph (I have kept minimum bins as 1 and maximum number of bins as 25) :

```
d3.select(".hist").on("mousedown", function() {
  console.log("Mouse Down")
  console.log( d3.event.pageX, d3.event.pageY )
  var mouseisdown = true;
  var xOnDown = d3.event.pageX;
  //InitialNoOfBins = noOfBins;

  d3.select(window)
    .on("mousemove", mousemove)
    .on("mouseup", mouseup);

  d3.event.preventDefault();

  function mousemove() {
    //console.log("Mouse Moving");

    if(Boolean(mouseisdown) == true)
    {   var xOnMove = d3.event.pageX;

      if(xOnMove - xOnDown > 20)
      {
        d3.selectAll("svg > *").remove();
        noOfBins--;
        if(noOfBins < 1)
        {
          noOfBins = 1;
        }
        drawHistogram(data, noOfBins, toPlot);
        xOnDown = xOnMove
      }
      else if(xOnDown - xOnMove > 20)
      {
        d3.selectAll("svg > *").remove();
        console.log("Increasing bins");
        noOfBins++;
        if(noOfBins >20 )
        {
          noOfBins = 20;
        }
      }
    }
  }
});
```

```

        drawHistogram(data, noOfBins, toPlot);
        xOnDown = xOnMove
    }
}

function mouseup() {
//console.log("Mouse up");
//console.log( d3.event.pageX, d3.event.pageY );
mouseisdown = false;

}
);

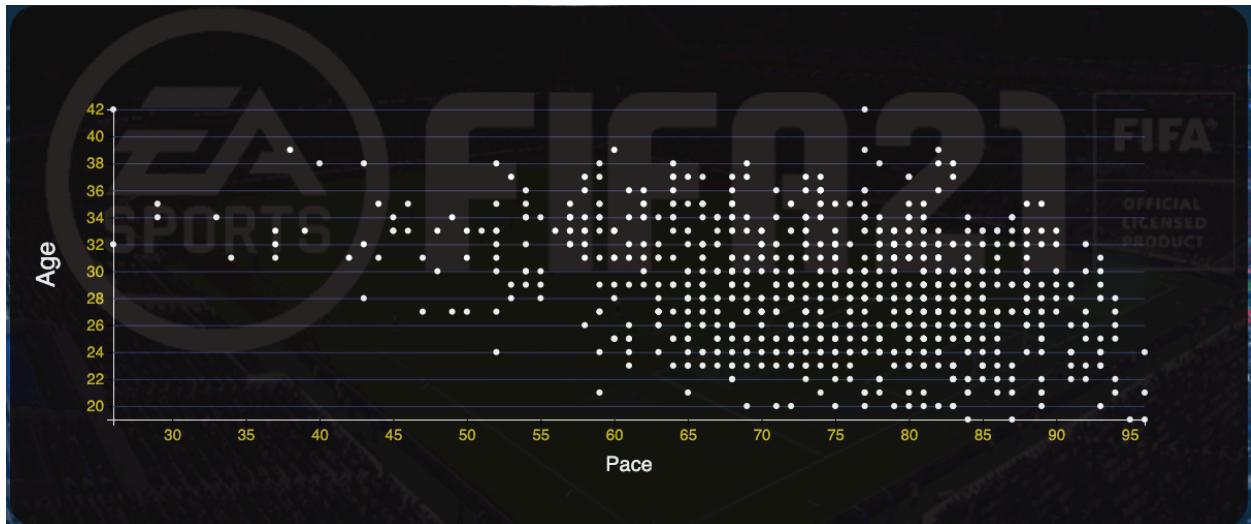
```

Functionalities implemented:

- Draw a Histogram if a numerical variable is selected.
- Highlight the bar in red color on MouseOver.
- Highlight the count in red color of that bar.
- Decrease/Increase number of bins on mouse down and drag right/left event.

Files involved in the flow: helper.js, Histogram\_Numerical.js

## 6. Scatter Plot:



When the user clicks on ScatterPlot button present on top of the menu, by default a Scatter plot will appear on the page having the attribute - Pace on X axis and Age on Y axis. If the user wants to change the variables present on X and Y axis, then they have the option to select the variables from the drop down to be plotted on the respective axes.

Code flow: Once the user clicks on Scatter plot button and choose what variables to plot on which axis, a function call is triggered specific to those attributes which is present in the helper.js file. This function in turn calls the main drawScatter(data1, data2, toPlot1, toPlot2)

function present in ScatterPlot.js file which takes the input attribute names, arrays of data for x and y axis.

For the scatter plot also, first we have to plot the axes and which variables will go on them similar to bar graph and histogram and then plot the scatter chart for them.

Main code snippets responsible for plotting the chart:

```
g.append("g")
.selectAll("dot")
.data(myDATA)
.enter()
.append("circle")
.attr("cx", function (d) { return x(d[0]); })
.attr("cy", function (d) { return y(d[1]); })
.transition()
.ease(d3.easeLinear)
.duration(0.5)
.delay(function (d, i) {
    return i * 1;
})
.attr("r", 3)
.style("fill", "white");
```

Functionalities implemented:

- Produce a scatter plot of two selected variables.
- Provided the option to choose Attributes for X axis and Y axis.

Files involved in the flow: helper.js, ScatterPlot.js