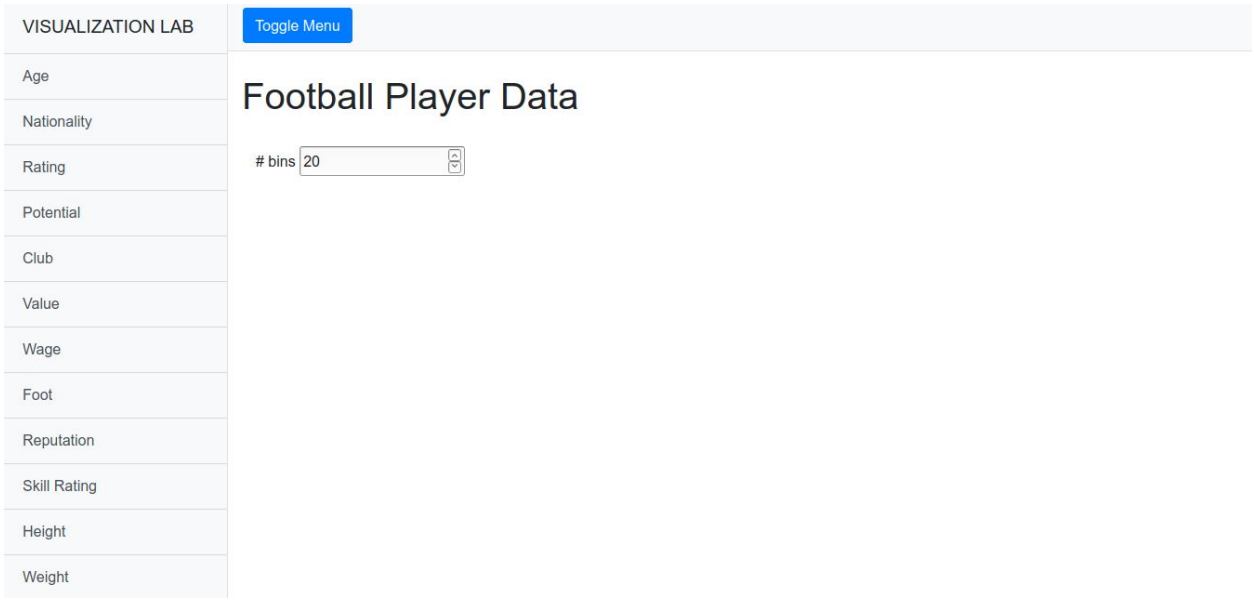
LAB1

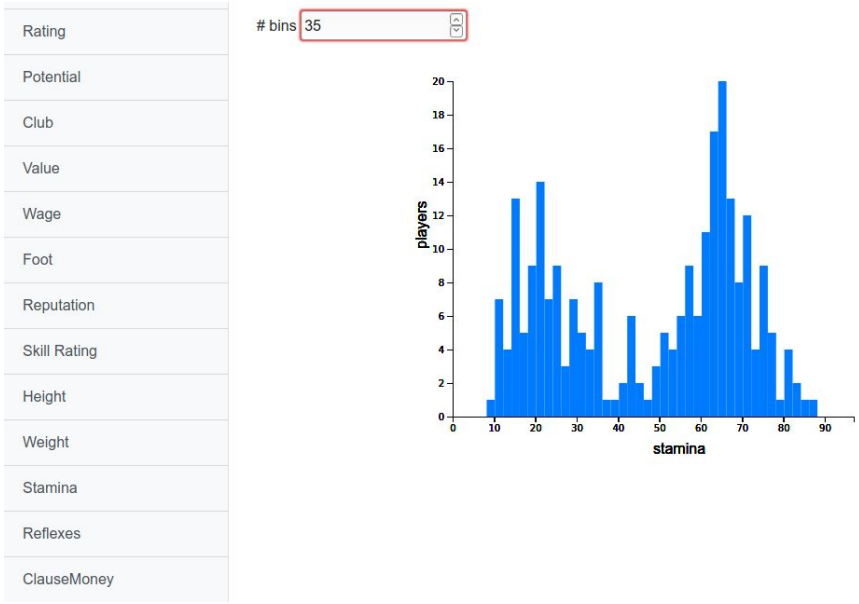
Kushagra Pareek : 112551443
kpareek@cs.stonybrook.edu

I have taken data from kaggle , with 250 data points and 15 columns.

Load Page:



I have chosen a basic Bootstrap Fixed side Navigation bar, the left side contains all data columns, clicking on a data column will result in display of the graph of that specific column.



Histogram (code snippets).
Every data filled of numerical field when clicked , calls their respective function,
The data is cleaned and passed to generic functions.

The functions below are used for creating histogram.

```
function createXScaleNum(data){

      let x = d3.scaleLinear()
                .domain([0,d3.max(data,function(d){return +d}
                  ) + 20 ])
                .range([0,width]);
      return x;
}

function createYScaleNum(data){

    let y = d3.scaleLinear()
              .domain([0,d3.max(data,function(d){return d.length})])
              .range([height,0]);

     return y;
}
```

```
function appendXNumScale(xScale,label){

     svg.append("g").attr("transform", "translate(0,"+ height+")")
                    .call(d3.axisBottom(xScale));
     svg.append("text").attr("transform", "translate("+ (width/2)+","+ (height+40)+")")
                    .text(label);

}

function appendYNumScale(yScale,label){

        svg.append("g").call(d3.axisLeft(yScale));
        svg.append("text").attr("transform", "translate("+-28+","+height/2+")rotate(-90)")
                        .text(label);

}
```

```
function createHistogram(xScale, data){

     let hist  = d3.histogram()
              .domain(xScale.domain())
              .thresholds(xScale.ticks(nBins));

    let bins = hist(data);

       return bins;
}

function appendHistogram(xScale, yScale, bins){


  svg.selectAll("rect")
          .data(bins)
          .enter()
            .append("rect")
            .attr("x", 1)
            .attr("transform", function(d) { return "translate(" + xScale(d.x0) + "," + yScale(d.length) + ")"; })
            .attr("width", function(d) { return xScale(d.x1) - xScale(d.x0)  ; })
            .attr("height", function(d) { return height - yScale(d.length); })
            .style("fill", "rgb(0,123,255)")
            .on("mouseover", showTooltip)
            .on("mousemove", moveTooltip)
            .on("mouseleave", hideTooltip);


}
```

Tooltip code is a generic tooltip template used in D3.

```
  .on("mouseover", showTooltip)
  .on("mousemove", moveTooltip)
  .on("mouseleave", hideTooltip);
```

Bin changer code:

```
d3.select("#page-content-wrapper").on("mousedown", function() {

    let div = d3.select(this)
        .classed("active", true);

    let xPos = d3.mouse(div.node())[0];


      let win = d3.select(window)
      .on("mousemove", mousemove)
      .on("mouseup", function(){
        div.classed("active", false);
          win.on("mousemove", null).on("mouseup", null);
      });

      function mousemove() {
        elem  = document.getElementById("nBin");
        if(d3.mouse(div.node())[0] + 20 < xPos){

            if(elem.value > 1 && elem.value <100){
            elem.value = elem.value - 1;
            elem.dispatchEvent(new Event("input"));
          }else{

              elem.value = 1;
          }

          xPos = d3.mouse(div.node())[0];
        }
        else if(d3.mouse(div.node())[0] - 20 > xPos ){
```

Bar chart: Similar functions are used for bar chart , Instead of linear scale , I have used band scale for x-axis:

```
function createXScale(data){

        let x = d3.scaleBand()
                  .domain(data)
                  .range([0,width]);
        return x;

}
```

And to count the frequency of each element in the columns used the below function:

```
function count(arr) {
  return arr.reduce((prev, curr) => (prev[curr] = ++prev[curr] || 1, prev), {})
}

function createXScaleNum(data){
```