

CSE 564 Visualization, Spring 2020, Lab 1

Tianao Wang 112819772

February 20, 2020

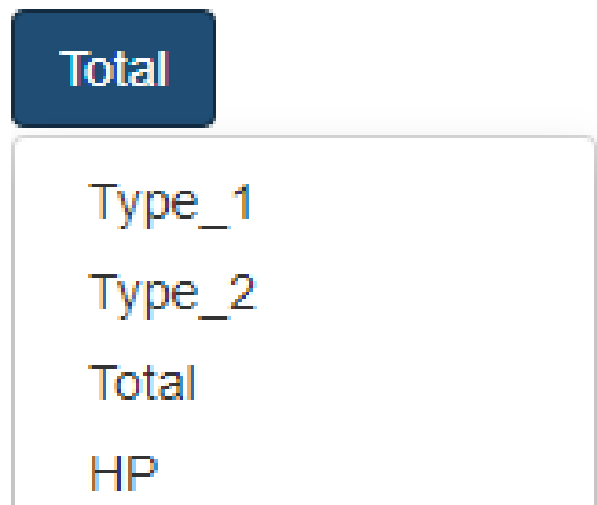
1 Data Preprocessing

The data I use is all Pokémon stats until generation 6 which is a famous IP in Nintendo. The data file has 22 columns and 721 rows. I do some data preprocessing to make the output graph more beautiful.

- (1) I delete some empty data when processing the data. Some Pokémon have more than one type. So for 'Type_2' column, some Pokémon have secondary type but some are empty. If I record 'null' or 'others' as a bar chart bin, the height of 'null' will be very high and this will make other real data be looked shorter and have almost same heights. I want to show the real type_2 element and don't want to be disturbed by these empty data. For this reason, I don't count the empty element when processing this data.
- (2) I give up some data which are not suitable for barchart or histogram. For example, the ID and the Name which are all unique. It's meaningless to draw a barchart or histogram to these data.

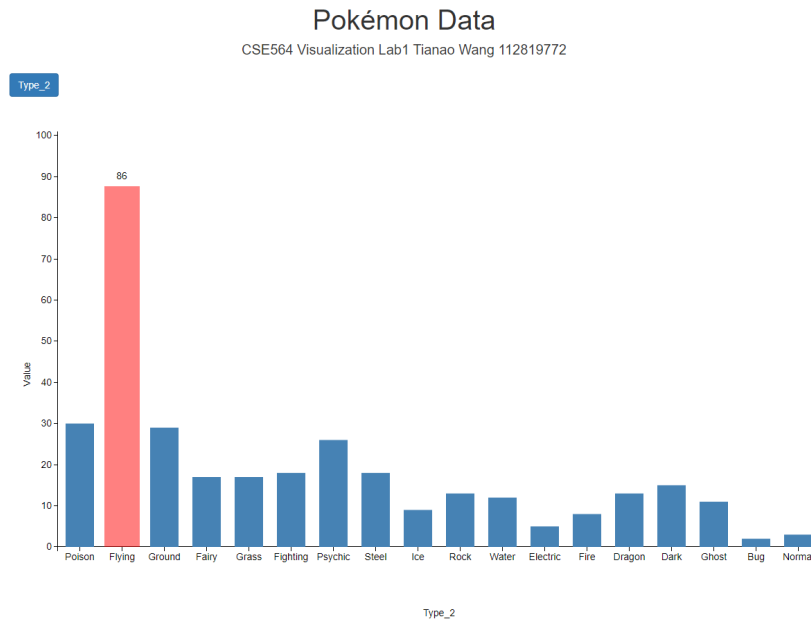
2 Program's Capabilities

- (1) A dropdown menu to switch which data variable to show.
The reason why I use dropdown is to try my best to re-use the code. I can dynamically add choices to the dropdown menu. Even if I have 20 or 30 variables, the web page will be the same.



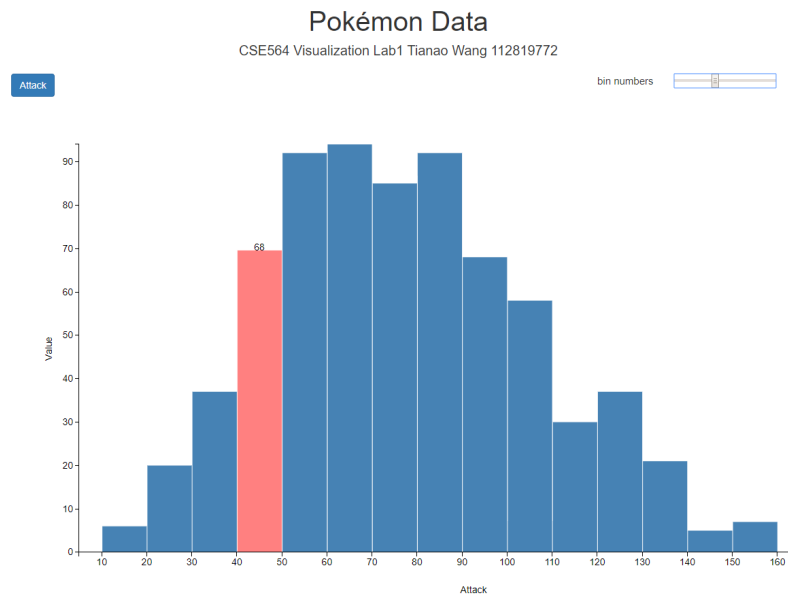
- (2) Draw a bar chart for categorical variable.

If the selected variable is a categorical variable, the website will generate a bar chart to show the datas. The graph labels axes, axes and categories.



- (3) Draw a histogram for numerical variable.

If the selected variable is a numerical variable, the website will generate a histogram to show the datas.

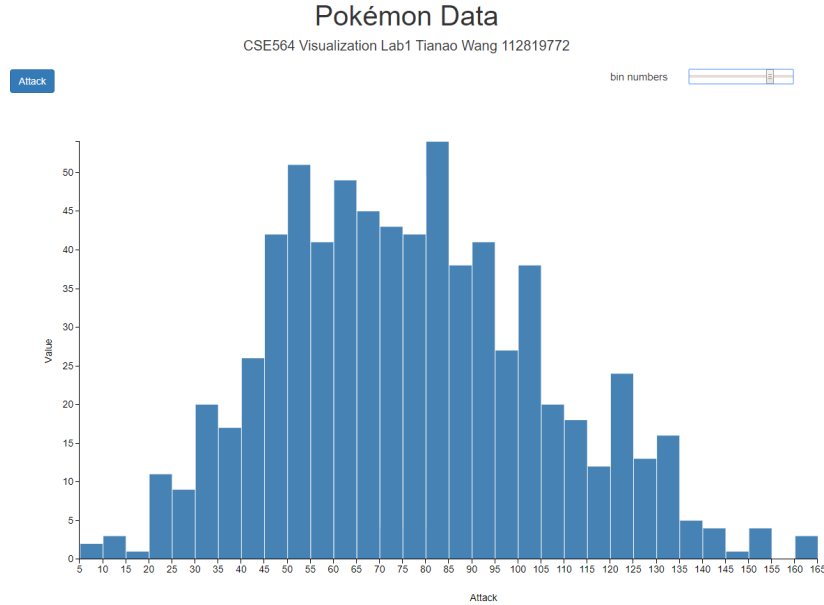


- (4) Move mouse over and out.

If the mouse moves on the bar, it will become wider and higher. Also it will change color and show the value of this bar. These changes aim to make this bar different from other bars and make it more attractive to users. After the mouse moves out, everything will go back.

(5) A slider.

User can use a slider to change the bin numbers of histogram and show the new histogram. This function is aimed to help user to view the data in different scale. Also, if user change the bin numbers, the bars still almostly full the graph to make sure the beauty of the graph.



3 Implementation Details

For this assignment, I use d3.js, bootstrap, jquery to programme. I put the data on github because d3.js can't directly read files by FTP and only support Http to do it. If you want to run my code, please make sure that your computer is connecting to the Internet because I include some css and js package url and the csv data from the Internet.

3.1 Read data

I put the csv file on github and read it by d3.js.

```
var x = [];  
var y = [];  
d3.csv("https://raw.githubusercontent.com/wangTianAo/CSE564/master/hm1/pokemon_attack.csv",  
  function(error) {  
    if(error) {  
      console.log(error);  
    }  
  
    var data = {};  
    for(var i = 0; i < csvdata.length; ++i) {  
      if(csvdata[i][DataCate] == "") {  
        continue;  
      }  
      if(csvdata[i][DataCate] in data) {  
        data[csvdata[i][DataCate]] += 1;  
      } else {  
        data[csvdata[i][DataCate]] = 0;  
      }  
    }  
  
    for(var key in data) {  
      x.push(key);  
      y.push(data[key]);  
    }  
  
    tempData = {x, y};  
    console.log(tempData);  
  })
```

3.2 A dropdown menu

I directly use a dropdown button in bootstrap. Once loading the csv file, it will dynamically generate all the variable choices. Each choice will be binded with a click function which will call a function to draw the charts.

```
for(var key in csvdata[0]){
  var li=document.createElement("li");
  li.setAttribute("aria-labelledby","dropdownMenu1");
  li.setAttribute("selected",true);
  var a=document.createElement("a");
  if(first==""){
    console.log(key)
    first = key;
  }
  a.innerHTML=key;
  a.href="#";
  li.appendChild(a);
  document.getElementById("drop").appendChild(li);
}
$("#dropdownMenu1").text("Total");
$("#drop").children("li").click(function(e){
$(this).addClass("active").siblings("li").removeClass("active");
$("#dropdownMenu1").text($(this).text())
drawChart($(this).text(),binNum);
})
})
```

3.3 bar charts

I use a map to store the keys and values. Then I use d3.scale.ordinal() to record the scale of x and d3.scale.linear() to record the scale of y. After this, I add a svg and use the api of d3 to draw the bar chart.

```
svg.selectAll(".rect")
.data(tempData.y)
.enter()
.append("rect")
.attr("class","bar")
.attr("x",function(d,i){
return xScale(tempData.x[i])+ rectPadding*6;
})
.attr("y",function(d,i){
return yScale(d)
})
.attr("width", xScale.rangeBand() - rectPadding*2 )
.attr("height", function(d){
return max_height - padding.top - padding.bottom - yScale(d);
})
```

3.4 Draw a histogram

I store all the values and use d3.scale.linear() to record the scale of x. Then I use d3.layout.histogram() to generate a histogram and get the data. With the data, I can use the api to draw the histogram.

```

var x_scale = d3.scale.linear()
    .domain([d3.min(tempData), d3.max(tempData)])
    .range([0, width - 100])

var histogram = d3.layout.histogram()
    .range([d3.min(tempData), d3.max(tempData)])
    .bins(x_scale.ticks(binNum))
    .frequency(true)

var data = histogram(tempData);
console.log(data);

```

3.5 Move mouse over and out

Use `on("mouseover",...)` and `on("mouseout",...)` functions to make the bar wider and higher and show the values by text.

```

.on("mouseover", function(d, i) {
    d3.select(this).attr("opacity", 0.5)
        .attr("height", max_height - y_scale(d.y) + 10)
        .attr("y", -10)
    bar.append("text")
        .attr("x", (x_scale(data[0].dx) - x_scale(0)) / 2)
        .attr("dy", -10)
        .attr("text-anchor", "middle")
        .text(function(d, j) {
            if(i == j)
                return formatCount(d.y);
        })
});

```

3.6 Axis

Use `d3.svg.axis()` to generate the axis and draw graph and texts in svg.

```

var xAxis = d3.svg.axis()
    .scale(x_scale)
    .orient("bottom")
    .tickFormat(d3.format(".0f"))
    .ticks(binNum)

var yAxis = d3.svg.axis()
    .scale(y_scale)
    .orient("left")

```

```

//draw x axis
svg.append("g")
  .attr("class","axis")
  .attr("transform", "translate(0," + max_height + ")")
  .call(xAxis)

//draw y axis
svg.append("g")
  .attr("class","axis")
  .call(yAxis)

//draw x axis text
svg.append("text")
  .attr("text-anchor", "middle")
  .attr("transform", "translate("+ (width/2) + ","+(max_height+3*padding.top)+")")
  .text(DataCate);

svg.append("text")
  .attr("text-anchor", "middle")
  .text("Value")
  .attr("transform", "translate("+ (padding.left*(-2)) + ","+(max_height/2)+")ro

```

3.7 A slider

I use a range input to record the value of bin number. Once the user move the input left or right, it will change the value of bin number and the programme will generate the new histogram.

```

function drawChart1(bin_num){
  d3.select("svg").html("");
  binNum = bin_num;
  DrawHistogram(DataCate);
}

```

4 Youtube Link

[Youtube Link](#)