Our product only has an index.html page. In the page, Javascript code would be loaded in the index.html page. “css/style.css” is responsible for the styles. “data/data_long.csv” is the data used for visualization. The code inside the “js” folder is most of the work. The “resource” folder includes the d3 library file. The exploratory. R was used for converting the initial data into the long data. The reference.txt provided an overview of the code reference information.

# 1 Classes

The following classes created: “BaseChart”, “TwoAxisTimeChart”, “BarChart1”, “BarChart2”, “LineChart”, “AreaChart”, “TimeInterval”.

“BaseChart” is an abstract class. The “TwoAxisTimeChart”, “BarChart1” and “BarChart2” classes are inherited from the “BaseChart” class. The “LineChart” and “AreaChart” class are inherited from the “TwoAxisTimeChart” class.

## 1.1 BaseChart

This is an abstract class. It’s in the baseChart.js file.

**Knowledge**: this class would maintain a config file and a data object.

**Responsibility**:

- initVis: It will initialize the width, height, of the parent svg, and the drawing area.
- updateVis: This method is not implemented.
- renderVis: This method is not implemented.

## 1.2 BarChart1 extends BaseChart

This is a chart that would render the context bar chart with a time brusher. When an instance is created, intiVis method of the instance would be called. It’s in the barChart1.js file.

**Knowledge**: This class have all the knowledge that the parent class (BaseChart) has, with additionally knowledge include two callback functions which would be called when the time brusher is moving or ends moving.

**Responsibility**:

- initVis: Like its parent class, it will also initialize the width, height, of the parent svg, and the drawing area. Additionally, it would create the scale of the x and y axes, the d3 axis object of X and Y, and the axis group. Moreover, a d3 x brush and a brush group would also be added as the attribute of the instance. Then it would call it’s updateVis method.
- updateVis: It would update the x1, x2, and Y value getter. It will set the domain of the x axis become the data_long global variable and set the

min and max value of the domain of y become the data's domain. Then it would call it's renderVis method.

- renderVis: This method would also render the bars for the data. This method would render the axis on the axis group svg element.
- Brushed: This method will be called when the brush is on the state of "brush", it would extract the start time and the end time and use to call the brushedCallback function.
- Brushed: This method will be called when the brush is on the state of "end", it would extract the start time and the end time and use them to call the brushedendCallback function.

## 1.3 BarChart2 extends BaseChart

This is a chart that would render the composite bar chart. When an instance is created, intiVis method of the instance would be called. It's in the barChart2.js file.

**Knowledge**: This class have all the knowledge that the parent class (BaseChart) has, with additionally knowledge include the title element and the legend element in the config, and the value type of the major and minor axis as well as the information about how to sort the bars.

**Responsibility**:

- initVis: Like its parent class, it will also initialize the width, height, of the parent svg, and the drawing area. Additionally, it would create the scale of the x and y axes, the d3 axis object of X and Y, and the axis group. Moreover, two drawing area groups for the major bar and the minor bars would de appended. Then, it would sort by the length of the major bars. Finally, it would call it's updateVis method.
- updateVis: It would update the major x, minor x, and the y value getter. It will update the domain scale based on the order information.. Then it would call it's renderVis method.
- renderVis: This method would render the axes, major bars and minor bars.
- setMajor: This method would set the major axis of the chart.
- setMinor: This method would set the minor axis of the chart.
- setData: This method would set the data of the chart.
- getMajorData: This method would return the major data of the chart.
- getMinorData: This method would return the minor data of the chart.
- Sort: this method would sort the chart according to the value of the attribute and desc parameter.

## 1.4 TwoAxisTimeChart extends BaseChart

This is an abstract class. It in the baseChart.js file.

**Knowledge**: This class have all the knowledge that the parent class (BaseChart) has, with additionally knowledge include a time object which has the start time and the end time of the instance of the class. The time information is used for controlling the domain of the time axis, so that it will not be controlled by the data's minimum time.

**Responsibility**:

- initVis: Like its parent class, it will also initialize the width, height, of the parent svg, and the drawing area. Additionally, it would create the scale of the x and y axes, the d3 axis object of X and Y, and the axis group.
- renderVis: This method would render the axis on the axis group svg element.
- setTime(time): This method would change the start time and the end time of the chart.

## 1.5 LineChart extends TwoAxisTimeChart

This class can render a line chart with multiple lines and marks on the given svg, it can also render tooltips on the given tooltip. It can also highlight a rendered line.

**Knowledge**: This class have all the knowledge that the parent class (TwoAxisTimeChart) has, with additionally knowledge include the tooltip, legend, and the highlighted location.

**Responsibility**:

- initVis: Like its parent class, it will also initialize the width, height, of the parent svg, and the drawing area, and create the scale of the x and y axes, the d3 axis object of X and Y, and the axis group. Additionally, it would create a colour domain method and a tooltip tracker.
- updateVis: It would set the x, y, and location getter. It would set the colour, x, and y domain. It will create the line generator. Finally, it would call the renderVis method.
- renderVis: This method would render the lines, markers, tooltips and markers, and update the legends of the line chart. It would highlight a location if set.
- colourScale: This is a private method
- getLocations: Return the rendered locations of the chart.
- highLightLocation: Given a location, high light the location.
- disHighLightLocation: Render all the location in an ordinary way.
- renderTooltip: render the tooltip of the data.
- renderMaker: render the data of the marker.

setTime(time): This method would change the start time and the end time of the chart.

### 1.6 AreaChart extends TwoAxisTimeChart

This class would render an area chart. It is in the areaChart.js

**Knowledge**: This class have all the knowledge that the parent class (TwoAxisTimeChart) has, with additionally knowledge include the tooltip, legend, and the location of the data.

**Responsibility**:

- initVis: Like its parent class, it will also initialize the width, height, of the parent svg, and the drawing area, and create the scale of the x and y axes, the d3 axis object of X and Y, and the axis group. Additionally, it would create a colour scale.
- updateVis: It would set the x, and y domain, as well as the colour scale. Finally, it would render the chart.
- renderVis: This method would render the stacked area and axes, also update the title.
- setLocation: This method would update the location attribute, but it will not update the chart
- getLocation: return the location of the chart.

### 1.7 TimeInterval

This class is a time interval. It has the knowledge of the start of the interval, end of the interval and if the interval includes each side. It provides is inside method, which takes a time and check if the time is inside the interval. It also provide toString method which return a string object represent the time interval.

## 2 Independent functions

There are some independent functions in the main.js: changeLineChart, getLineChartData, changeAreaChart, getAreaChartData, changeBarChart2, getBarChart2Data, barChart1brushedCallback, barChart1brushedendCallback, changeBarChart1, getBarChart1Data.

There are another three functions in the legend.js: updateLineChartLegends, toggleLineChartLocations, and updateAreaChartLegends

- changeLineChart
    - change the time range and the locations of the line chart, given the start time, end time, and locations.
- getLineChartData
    - return the data for the line chart from sorted long data, given the start time and the end time.

- changeAreaChart
    - change the time range and the location of the area chart, given the start time, end time and locations.
- getAreaChartData
    - return the data for the area chart from sorted long data, given the start time and the end time.
- changeBarChart2
    - change the time range and the axes and the ordering of the composite bar chart, given the start time, end time, major axis value, minor axis value and sorting axis (should be “major”, “minor”, or “location”), and order.
- getBarChart2Data
    - return the data for the composite bar chart, given the sorted long data, start time, end time and the data type, the data type should be “all” by default.
- barChart1brushedCallback
    - a call back function that takes 2 time objects and would be called when the time brusher is moving. This function does nothing now because the algorithm for change charts is too slow.
- barChart1brushedendCallback
    - a call back function that takes 2 time objects, it would be called when the time brusher stopped moving. It would now change the time range of the line chart, the area chart and the composite bar chart. It would also change the title of the composite bar chart.
- changeBarChart1
    - change the length of the time, data type, location, facility, of the context bar chart, given the time length in minutes, data type, location, facility.
- getBarChart1Data
    - get the data for the context bar chart, given the long data, start time, end time, the length of time intervals, data type, location, and facility.
- updateLineChartLegends
    - render the legends of the line chart, it would use the data from the linechart.
- toggleLineChartLocations
    - change the rendered location lines of the line chart, given the location information.
- updateAreaChartLegends
    - render the legends of the area chart, it would use the data from the legends.

## 3 Constant

There are many some constant in the config:

- DEVMODE
  - True to do some assertion, False to avoid the assertion.
- VALID_LOCATIONS
  - An array of locations that is allowed to be selected by the user.
- VALID_FACILITIES
  - An array of facilities that allowed to be selected by the user.
- VALID_FACILITIES_INDEX
  - Used for generating stack data in the stacked area chart.
- VALID_BAR_CHART_X
  - An array of valid value type in the composite bar chart.
- MAX_LINE_CHART_LOCATION
  - The maximum number of lines that can be displayed on the line chart.
- EMPTY_COLOUR_DOMAIN_VALUE
  - Used for representing that there is no data in the line chart's colour domain.
- NON_HIGHLIGHTED_COLOUR
  - In the line chart, when highlight a line, other lines would become this colour.
- LINECHART_CONFIG
  - Define the svg, size, margin, tooltip, legend of the line chart.
- AREACHART_CONFIG
  - Define the svg, size, margin, tooltip, legend of the area chart.
- BARCHART1_CONFIG
  - Define the svg, size, and the margin of the bar chart.
- BARCHART2_CONFIG
  - Define the svg, size, margin, tooltip, legend of the area chart
- DEFAULT_BAR_CHART_1_INTERVAL_LENGTH
  - Define the default interval length of the context bar chart.
- DEFAULT_MEANDAMAGEVALUE_FOR_BAR_CHART_2
  - Define the default mean damage value for the missing data, it should be 0.
- DEFAULT_COUNT_FOR_BAR_CHART_2
  - Define the default count value for the missing data, it should be 0.
- DEFAULT_STD_FOR_BAR_CHART_2
  - Define the default deviation value for the missing data, it should be 0.

## 4 Input HTML elements

There are a set of HTML elements that is used for controlling the charts.

## 4.1 Line chart control elements

For the line chart, there is a group of legends, they are inside the line-chart-legends div element. The event listener of these elements will be added on the element inside them in the updateLineChartLegends method. These inside elements can set the line chart's location, highlight a line or set the location of the area chart.

## 4.2 Context bar chart control elements

There are 4 controllers of the context bar chart, they are in the barChartValueType, barChartLocationFilter, barChartFacilityFilter, barChartIntervalLength elements. The event listener is added in the main.js file from line 65 to line 97.

## 4.3 Composite bar chart control elements

There are 8 controllers of the composite bar chart including bar2GroupNameDescending, bar2GroupNameAscending, bar2MajorDescending, bar2MajorAscending, bar2MinorDescending, bar2MinorAscending, bar2Major, bar2Minor. The event listener are added in the main.js file from line 101 to line 135.