Art of Data Visualization

이진호, 이해진, 이동구

Project Proposal

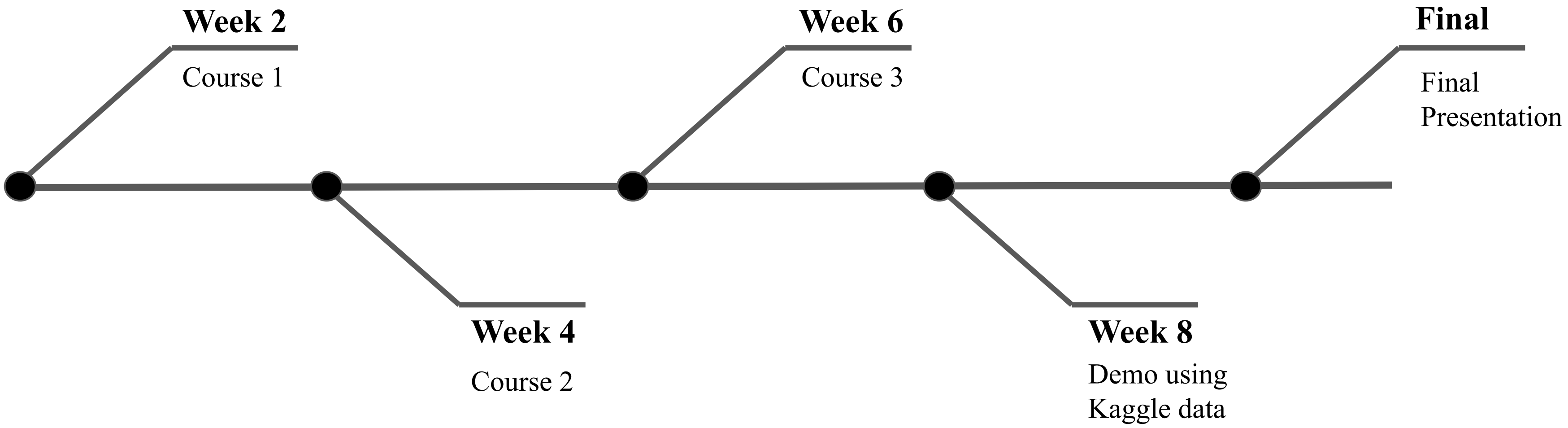
Topic Data Visualization

Description Coursera 와 Kaggle 에 있는 Data Visualization 관련 수업과 자료들을 빠르게 학습하며 파이썬의 여러가지 라이브러리들을어떻게 효율적으로 다루어 시각화 자료를 만드는지 배우는 기회를 가져보려고 합니다.

Expected Duration 8 weeks

Team Member 이진호, 이해진, 이동구

Timeline



Week 2

Course 1

Week 4

Course 2

Week 6

Course 3

Week 8

Demo using
Kaggle data

Final

Final
Presentation

Data Visualization with Python

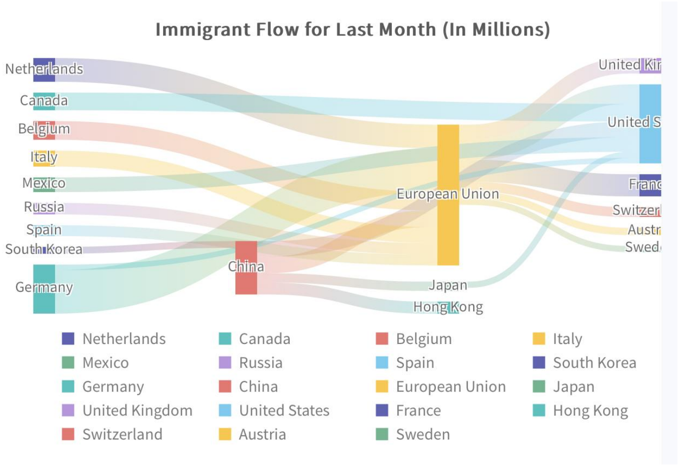
- Visualization of geospatial data
- Maps with markers
- Choropleth Maps
- Plotly & Dash introduction

Advanced Visualization - Sankey Chart

- Sankey charts are like flow diagrams wherein the width represents the quantity of the flow.

- Sankey charts are immensely useful when we are supposed to represent large amount of data with lots of inter correlation amongst the values (nodes).

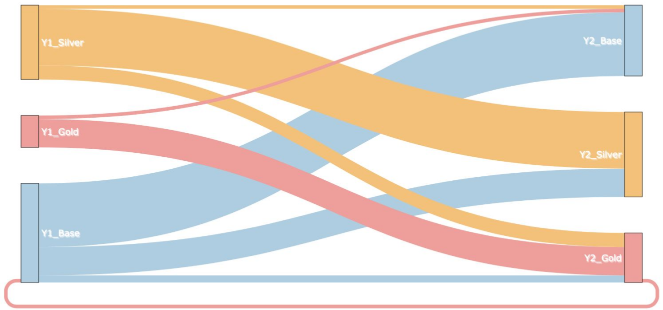
- Sankey charts are usually used for visualization of material, cost or energy flows.

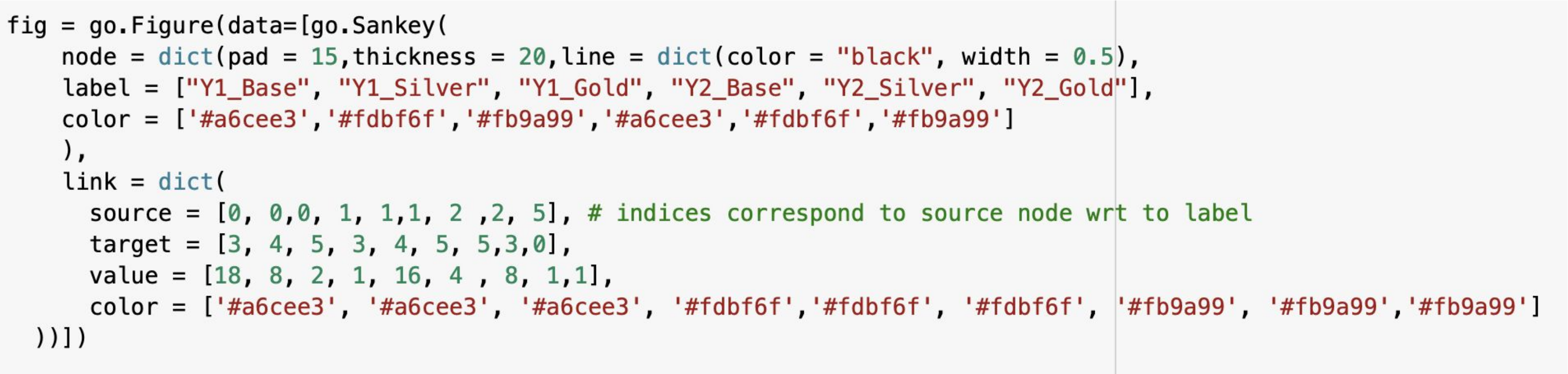


Immigrant Flow for Last Month (In Millions)

Demo 1

- It show the flow of customers from Base,Silver and Gold tiers in the first year and the second year.





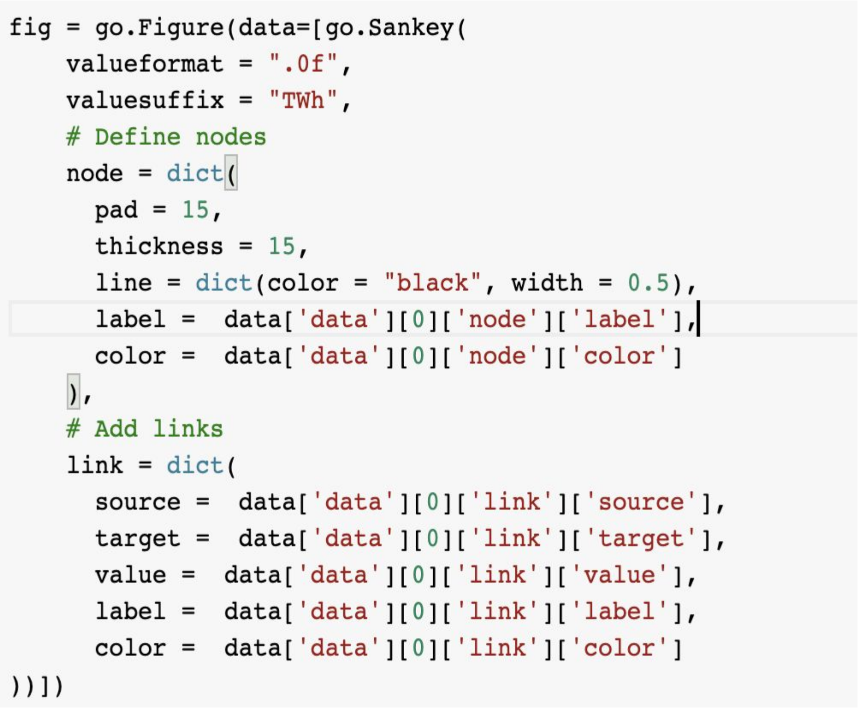
```
fig = go.Figure(data=[go.Sankey(
    node = dict(pad = 15,thickness = 20,line = dict(color = "black", width = 0.5),
    label = ["Y1_Base", "Y1_Silver", "Y1_Gold", "Y2_Base", "Y2_Silver", "Y2_Gold"],
    color = ['#a6cee3','#fdbf6f','#fb9a99','#a6cee3','#fdbf6f','#fb9a99']
    ),
    link = dict(
      source = [0, 0,0, 1, 1,1, 2 ,2, 5], # indices correspond to source node wrt to label
      target = [3, 4, 5, 3, 4, 5, 5,3,0],
      value = [18, 8, 2, 1, 16, 4 , 8, 1,1],
      color = ['#a6cee3', '#a6cee3', '#a6cee3', '#fdbf6f','#fdbf6f', '#fdbf6f', '#fb9a99', '#fb9a99','#fb9a99']
  ))])
```

Demo 2

- Data: Energy Conversion - JSON format

  https://raw.githubusercontent.com/plotly/plotly.js/master/test/image/mocks/sankey_energy.json

- JSON: Follows object literal grammar
- For Object-Structured data

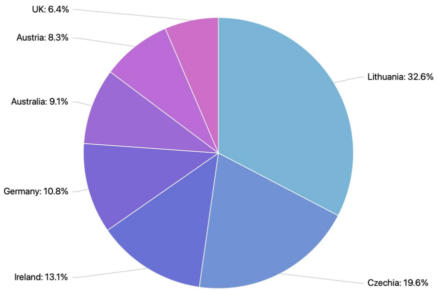


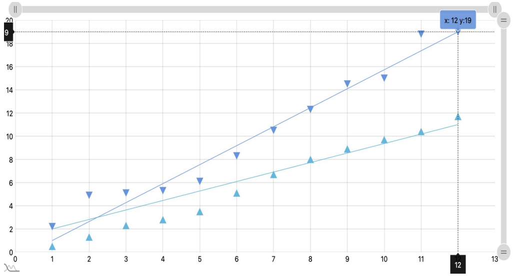
```
fig = go.Figure(data=[go.Sankey(
    valueformat = ".0f",
    valuesuffix = "TWh",
    # Define nodes
    node = dict(
      pad = 15,
      thickness = 15,
      line = dict(color = "black", width = 0.5),
      label =  data['data'][0]['node']['label'],
      color =  data['data'][0]['node']['color']
    ),
    # Add links
    link = dict(
      source =  data['data'][0]['link']['source'],
      target =  data['data'][0]['link']['target'],
      value =  data['data'][0]['link']['value'],
      label =  data['data'][0]['link']['label'],
      color =  data['data'][0]['link']['color']
))])
```

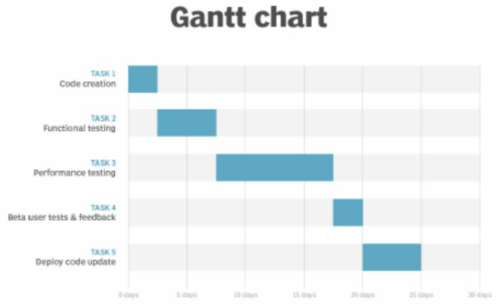
Advanced Visualization - amChart

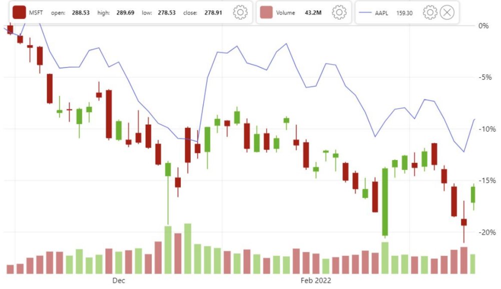
- amCharts is a set of programming libraries and tools for all your data visualization needs on the web.

- amCharts JavaScript product lineup includes classic charts like Line, Area, Column, Bar, Pie, XY, Scatter, Candlestick and OHLC as well as more “exotic” ones like Gauges, Funnels, GANTT, Stock Chart to display large amounts of financial and other date-based data, and fully interactive adaptive world or local country maps.









Data

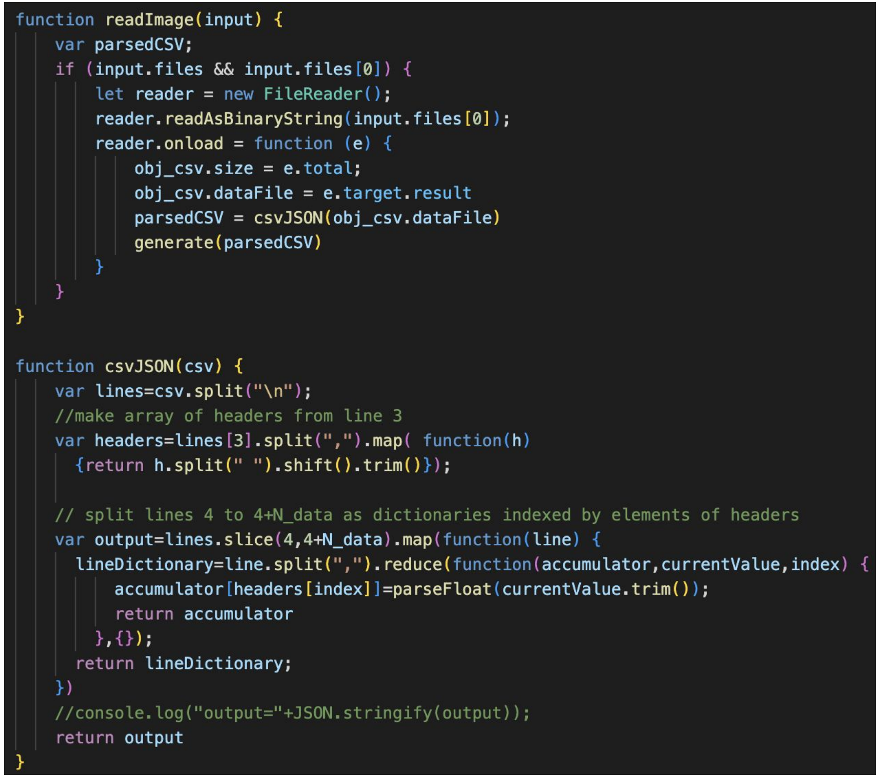
Crop harvest data

Columns: Distance, VRYieldVolume, Wetmass, Moisture

Change of these values over time

Normalized by column

Code : Preprocessing



```
function readImage(input) {
    var parsedCSV;
    if (input.files && input.files[0]) {
        let reader = new FileReader();
        reader.readAsBinaryString(input.files[0]);
        reader.onload = function (e) {
            obj_csv.size = e.total;
            obj_csv.dataFile = e.target.result
            parsedCSV = csvJSON(obj_csv.dataFile)
            generate(parsedCSV)
        }
    }
}

function csvJSON(csv) {
    var lines=csv.split("\n");
    //make array of headers from line 3
    var headers=lines[3].split(",").map( function(h)
      {return h.split(" ").shift().trim()});

    // split lines 4 to 4+N_data as dictionaries indexed by elements of headers
    var output=lines.slice(4,4+N_data).map(function(line) {
      lineDictionary=line.split(",").reduce(function(accumulator,currentValue,index) {
        accumulator[headers[index]]=parseFloat(currentValue.trim());
        return accumulator
      },{});
      return lineDictionary;
    })
    //console.log("output="+JSON.stringify(output));
    return output
}
```

readImage()

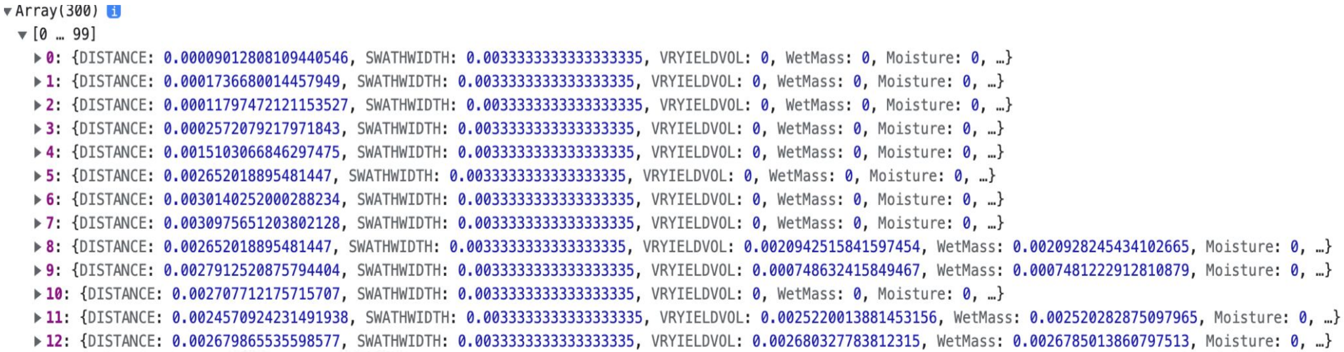
trigger function

onload event -> parse and preprocess the csv file

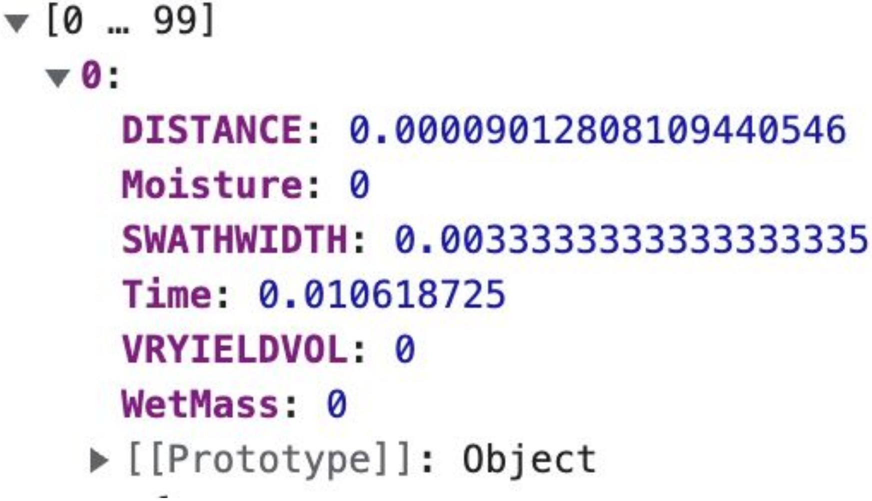
csvJSON()

Convert csv file to JSON object

Csv to JSON

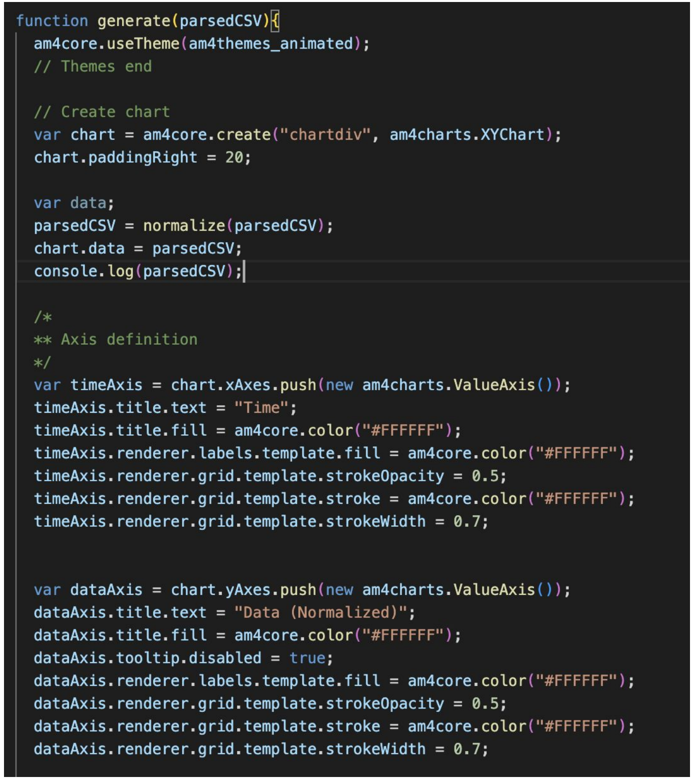


```
▼ Array(300)
  ▼ [0 … 99]
    ▶ 0: {DISTANCE: 0.000090128081094405546, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 1: {DISTANCE: 0.0001736680014457949, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 2: {DISTANCE: 0.00011797472121153527, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 3: {DISTANCE: 0.0002572079217971843, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 4: {DISTANCE: 0.0015103066846297475, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 5: {DISTANCE: 0.002652018895481447, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 6: {DISTANCE: 0.0030140252000288234, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 7: {DISTANCE: 0.0030975651203802128, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 8: {DISTANCE: 0.002652018895481447, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0.0020942515841597454, WetMass: 0.0020928245434102665, Moisture: 0, …}
    ▶ 9: {DISTANCE: 0.0027912520875794404, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0.0007486324158494467, WetMass: 0.0007481222912810879, Moisture: 0, …}
    ▶ 10: {DISTANCE: 0.002707712175715707, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0, WetMass: 0, Moisture: 0, …}
    ▶ 11: {DISTANCE: 0.0024570924231491938, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0.0025220013881453156, WetMass: 0.002520282875097965, Moisture: 0, …}
    ▶ 12: {DISTANCE: 0.002679865535598577, SWATHWIDTH: 0.0033333333333333335, VRYIELDVOL: 0.002680327783812315, WetMass: 0.0026785013860797513, Moisture: 0, …}
```



```
▼ [0 … 99]
  ▼ 0:
      DISTANCE: 0.000090128081094405546
      Moisture: 0
      SWATHWIDTH: 0.0033333333333333335
      Time: 0.010618725
      VRYIELDVOL: 0
      WetMass: 0
    ▶ [[Prototype]]: Object
```

Axis and Series Configuration



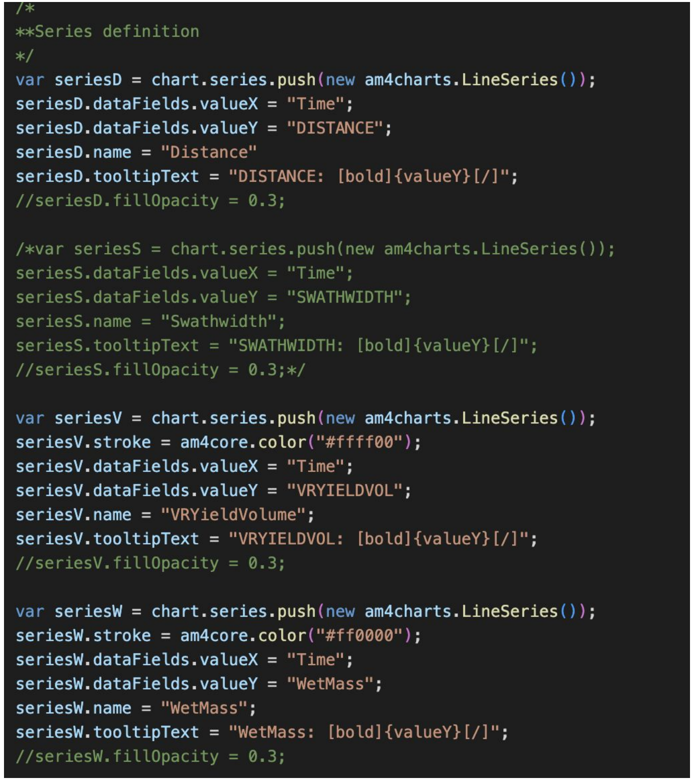
```
function generate(parsedCSV){
    am4core.useTheme(am4themes_animated);
    // Themes end

    // Create chart
    var chart = am4core.create("chartdiv", am4charts.XYChart);
    chart.paddingRight = 20;

    var data;
    parsedCSV = normalize(parsedCSV);
    chart.data = parsedCSV;
    console.log(parsedCSV);

    /*
    ** Axis definition
    */
    var timeAxis = chart.xAxes.push(new am4charts.ValueAxis());
    timeAxis.title.text = "Time";
    timeAxis.title.fill = am4core.color("#FFFFFF");
    timeAxis.renderer.labels.template.fill = am4core.color("#FFFFFF");
    timeAxis.renderer.grid.template.strokeOpacity = 0.5;
    timeAxis.renderer.grid.template.stroke = am4core.color("#FFFFFF");
    timeAxis.renderer.grid.template.strokeWidth = 0.7;


    var dataAxis = chart.yAxes.push(new am4charts.ValueAxis());
    dataAxis.title.text = "Data (Normalized)";
    dataAxis.title.fill = am4core.color("#FFFFFF");
    dataAxis.tooltip.disabled = true;
    dataAxis.renderer.labels.template.fill = am4core.color("#FFFFFF");
    dataAxis.renderer.grid.template.strokeOpacity = 0.5;
    dataAxis.renderer.grid.template.stroke = am4core.color("#FFFFFF");
    dataAxis.renderer.grid.template.strokeWidth = 0.7;
```



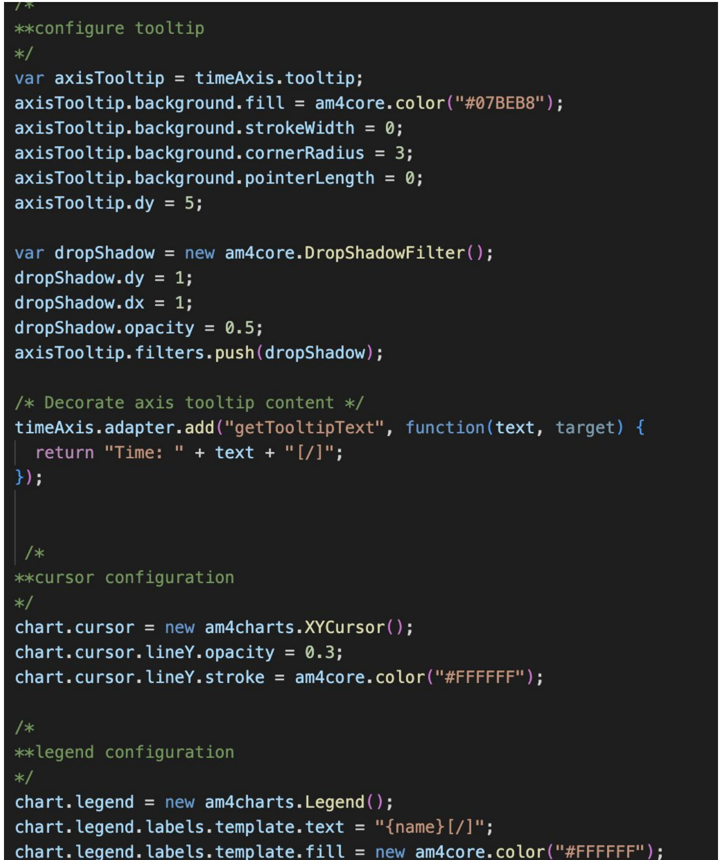
```
/*
**Series definition
*/
var seriesD = chart.series.push(new am4charts.LineSeries());
seriesD.dataFields.valueX = "Time";
seriesD.dataFields.valueY = "DISTANCE";
seriesD.name = "Distance"
seriesD.tooltipText = "DISTANCE: [bold]{valueY}[/]";
//seriesD.fillOpacity = 0.3;

/*var seriesS = chart.series.push(new am4charts.LineSeries());
seriesS.dataFields.valueX = "Time";
seriesS.dataFields.valueY = "SWATHWIDTH";
seriesS.name = "Swathwidth";
seriesS.tooltipText = "SWATHWIDTH: [bold]{valueY}[/]";
//seriesS.fillOpacity = 0.3;*/

var seriesV = chart.series.push(new am4charts.LineSeries());
seriesV.stroke = am4core.color("#ffff00");
seriesV.dataFields.valueX = "Time";
seriesV.dataFields.valueY = "VRYIELDVOL";
seriesV.name = "VRYieldVolume";
seriesV.tooltipText = "VRYIELDVOL: [bold]{valueY}[/]";
//seriesV.fillOpacity = 0.3;

var seriesW = chart.series.push(new am4charts.LineSeries());
seriesW.stroke = am4core.color("#ff0000");
seriesW.dataFields.valueX = "Time";
seriesW.dataFields.valueY = "WetMass";
seriesW.name = "WetMass";
seriesW.tooltipText = "WetMass: [bold]{valueY}[/]";
//seriesW.fillOpacity = 0.3;
```

Tooltips Configuration



```
/*
**configure tooltip
*/
var axisTooltip = timeAxis.tooltip;
axisTooltip.background.fill = am4core.color("#07BEB8");
axisTooltip.background.strokeWidth = 0;
axisTooltip.background.cornerRadius = 3;
axisTooltip.background.pointerLength = 0;
axisTooltip.dy = 5;

var dropShadow = new am4core.DropShadowFilter();
dropShadow.dy = 1;
dropShadow.dx = 1;
dropShadow.opacity = 0.5;
axisTooltip.filters.push(dropShadow);

/* Decorate axis tooltip content */
timeAxis.adapter.add("getTooltipText", function(text, target) {
  return "Time: " + text + "[/]";
});



 /*
**cursor configuration
*/
chart.cursor = new am4charts.XYCursor();
chart.cursor.lineY.opacity = 0.3;
chart.cursor.lineY.stroke = am4core.color("#FFFFFF");

/*
**legend configuration
*/
chart.legend = new am4charts.Legend();
chart.legend.labels.template.text = "{name}[/]";
chart.legend.labels.template.fill = new am4core.color("#FFFFFF");
```

timeAxis.tooltip

DropShadowFilter

XYcursor()

legend

Advanced Visualization - Chart.js

- js. is an free JavaScript library for making HTML-based charts. It is one of the simplest visualization libraries for JavaScript.

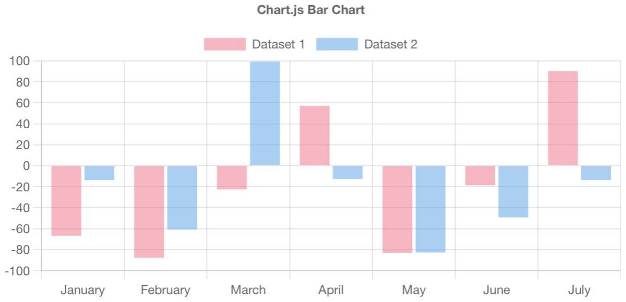
- It is a open source library, which supports 8 different chart types: bar, line, area, pie, bubble, radar, polar and scatter.

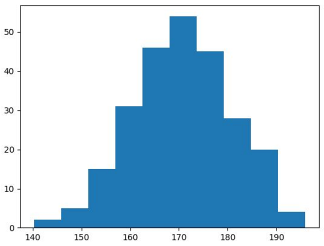
- From scale tracking to animation, it makes graph looks more distinct and better.

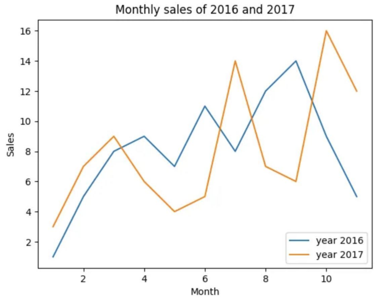
ON
OFF
60
40
20
0
1
2
5
10
20
50
100
A
B
C
D
E
F
G

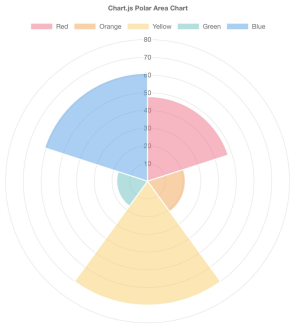
Chart.js

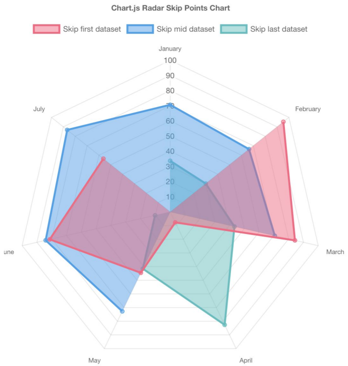
matplotlib

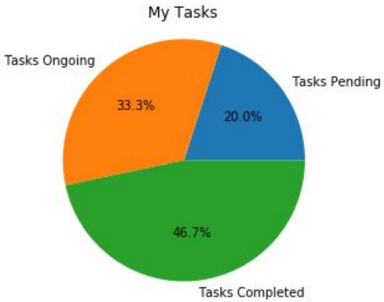




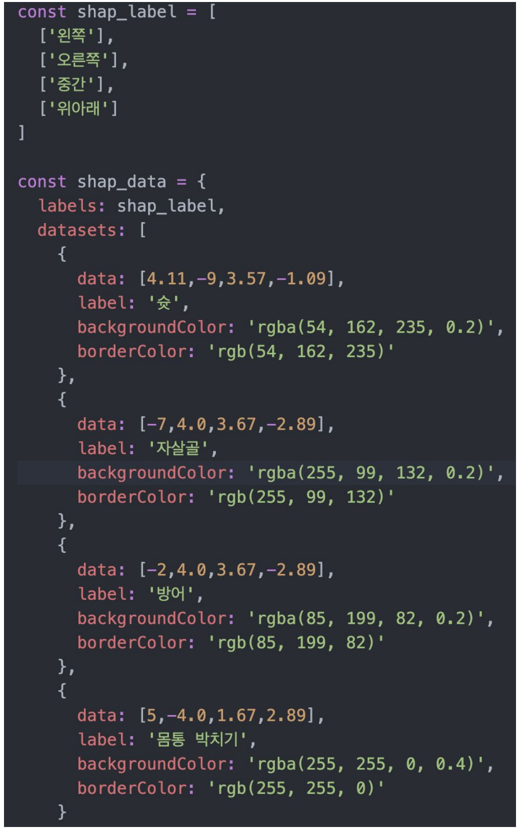






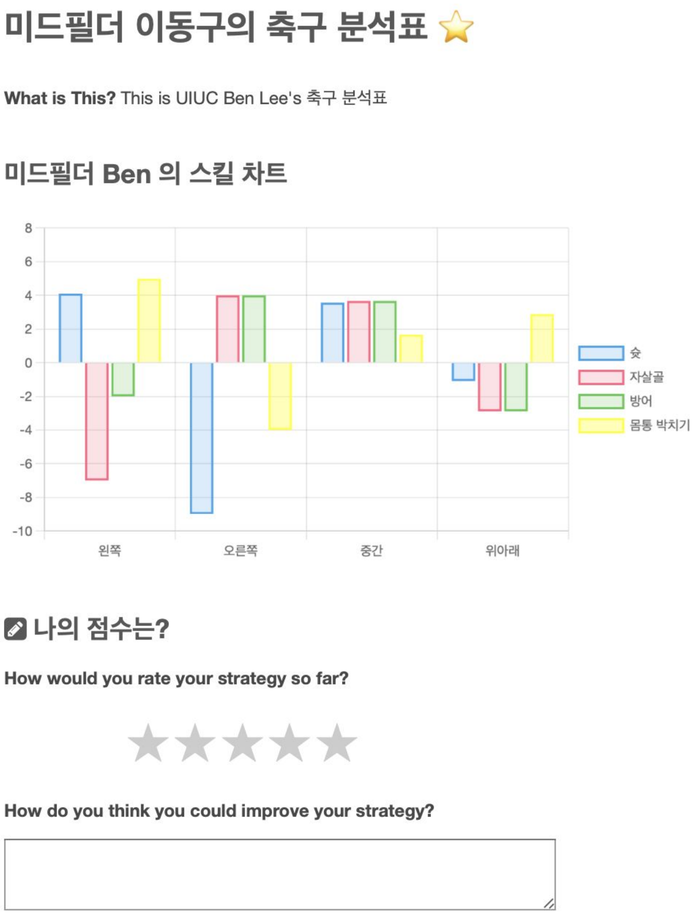


Demo 1



```
const shap_label = [
  ['왼쪽'],
  ['오른쪽'],
  ['중간'],
  ['위아래']
]

const shap_data = {
  labels: shap_label,
  datasets: [
    {
      data: [4.11,-9,3.57,-1.09],
      label: '슛',
      backgroundColor: 'rgba(54, 162, 235, 0.2)',
      borderColor: 'rgb(54, 162, 235)'
    },
    {
      data: [-7,4.0,3.67,-2.89],
      label: '자살골',
      backgroundColor: 'rgba(255, 99, 132, 0.2)',
      borderColor: 'rgb(255, 99, 132)'
    },
    {
      data: [-2,4.0,3.67,-2.89],
      label: '방어',
      backgroundColor: 'rgba(85, 199, 82, 0.2)',
      borderColor: 'rgb(85, 199, 82)'
    },
    {
      data: [5,-4.0,1.67,2.89],
      label: '몸통 박치기',
      backgroundColor: 'rgba(255, 255, 0, 0.4)',
      borderColor: 'rgb(255, 255, 0)'
    }
```



미드필더 이동구의 축구 분석표 ⭐

What is This? This is UIUC Ben Lee's 축구 분석표

미드필더 Ben 의 스킬 차트

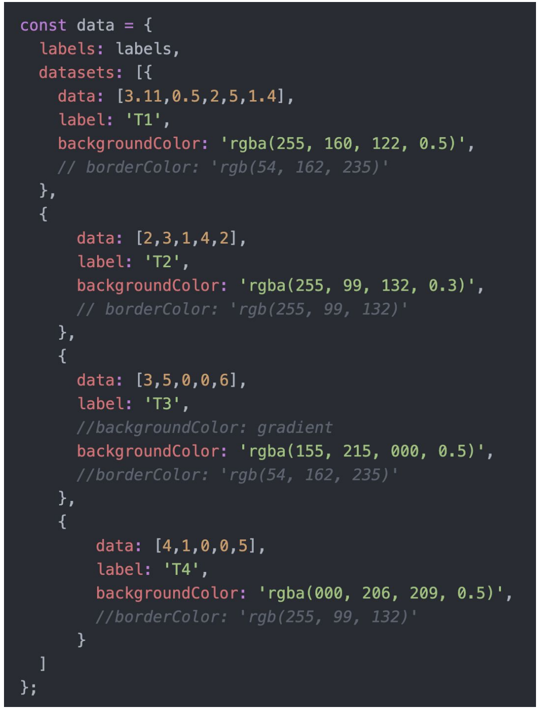
📝 나의 점수는?

How would you rate your strategy so far?

★★★★★

How do you think you could improve your strategy?

Demo 2



```
const data = {
  labels: labels,
  datasets: [{
    data: [3.11,0.5,2,5,1.4],
    label: 'T1',
    backgroundColor: 'rgba(255, 160, 122, 0.5)',
    // borderColor: 'rgb(54, 162, 235)'
  },
  {
      data: [2,3,1,4,2],
      label: 'T2',
      backgroundColor: 'rgba(255, 99, 132, 0.3)',
      // borderColor: 'rgb(255, 99, 132)'
    },
    {
      data: [3,5,0,0,6],
      label: 'T3',
      //backgroundColor: gradient
      backgroundColor: 'rgba(155, 215, 000, 0.5)',
      //borderColor: 'rgb(54, 162, 235)'
    },
    {
        data: [4,1,0,0,5],
        label: 'T4',
        backgroundColor: 'rgba(000, 206, 209, 0.5)',
        //borderColor: 'rgb(255, 99, 132)'
      }
  ]
};
```

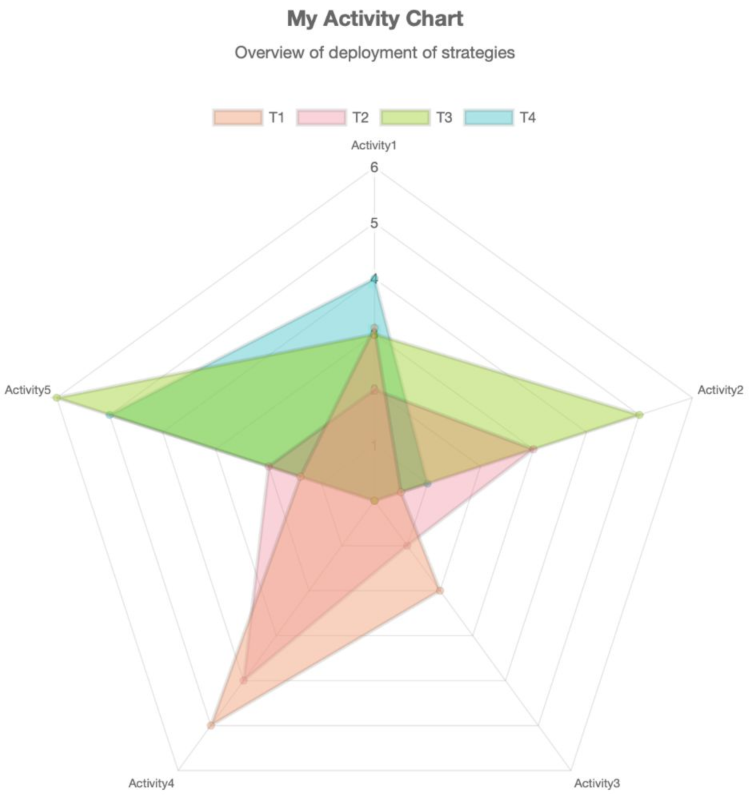
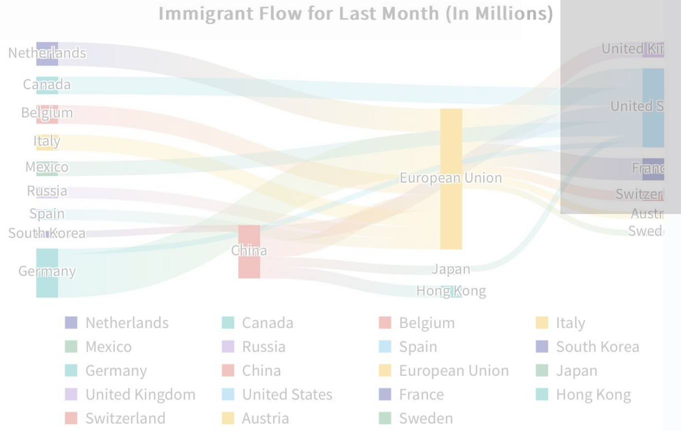


Table of Contents

1. Sankey Chart
2. amChart
3. Chart.js

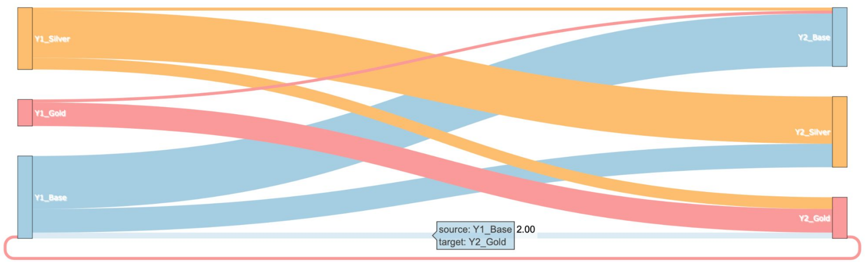


Advanced Visualization - Sankey Chart

- Sankey charts are like flow diagrams wherein the width represents the quantity of the flow.

- Sankey charts are immensely useful when we are supposed to represent large amount of data with lots of inter correlation amongst the values (nodes).

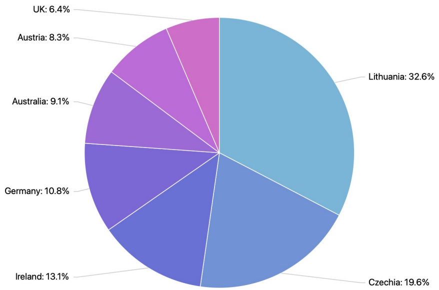
- Sankey charts are usually used for visualization of material, cost or energy flows.

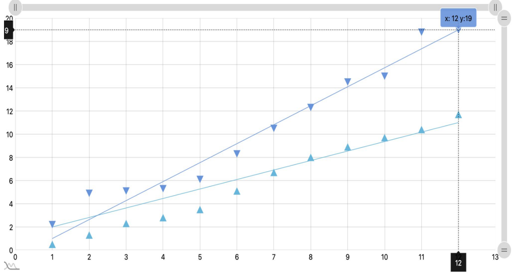


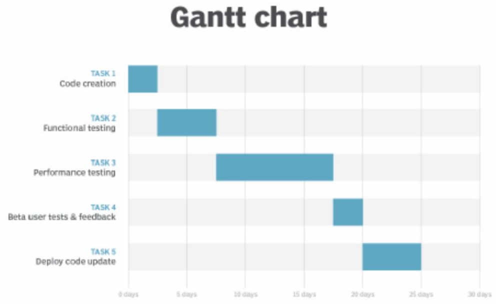
Advanced Visualization - amChart

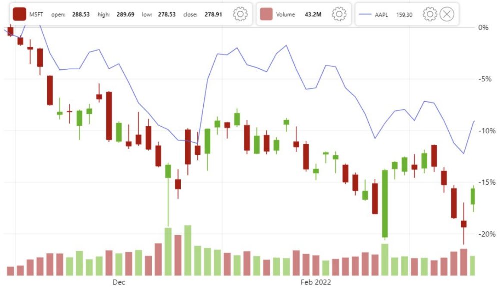
- amCharts is a set of programming libraries and tools for all your data visualization needs on the web.

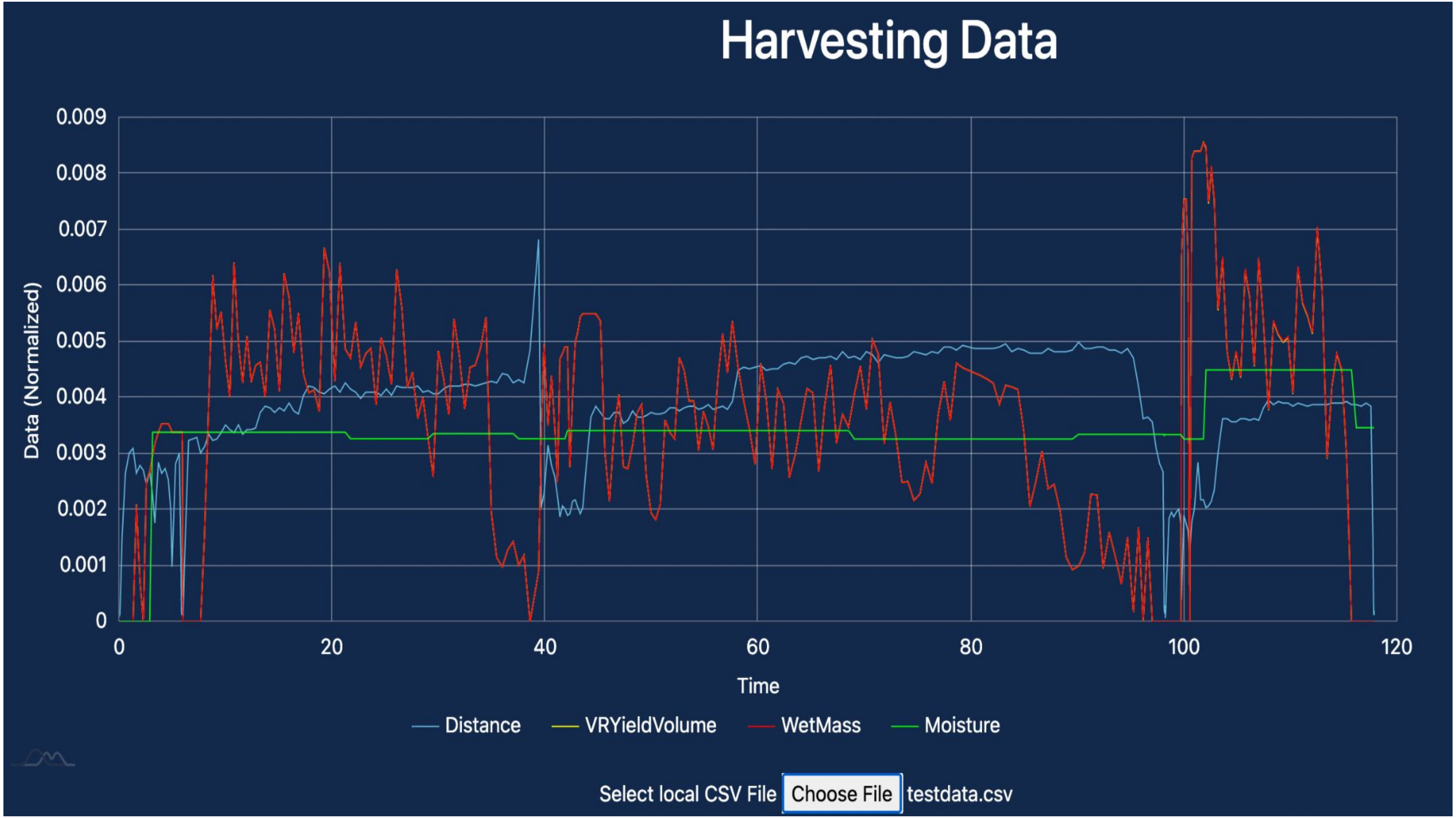
- amCharts JavaScript product lineup includes classic charts like Line, Area, Column, Bar, Pie, XY, Scatter, Candlestick and OHLC as well as more “exotic” ones like Gauges, Funnels, GANTT, Stock Chart to display large amounts of financial and other date-based data, and fully interactive adaptive world or local country maps.











Harvesting Data

Select local CSV File testdata.csv

Advanced Visualization - Chart.js

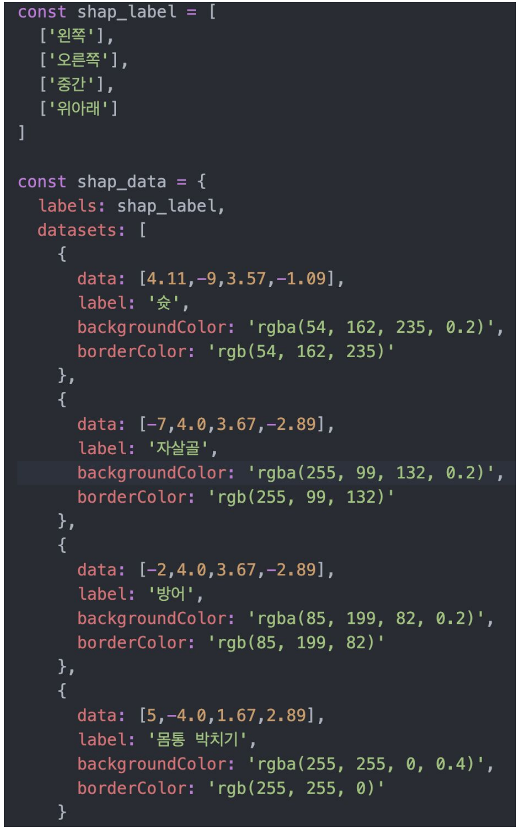
- js. is an free JavaScript library for making HTML-based charts. It is one of the simplest visualization libraries for JavaScript.

- It is a open source library, which supports 8 different chart types: bar, line, area, pie, bubble, radar, polar and scatter.

- From scale tracking to animation, it makes graph looks more distinct and better.

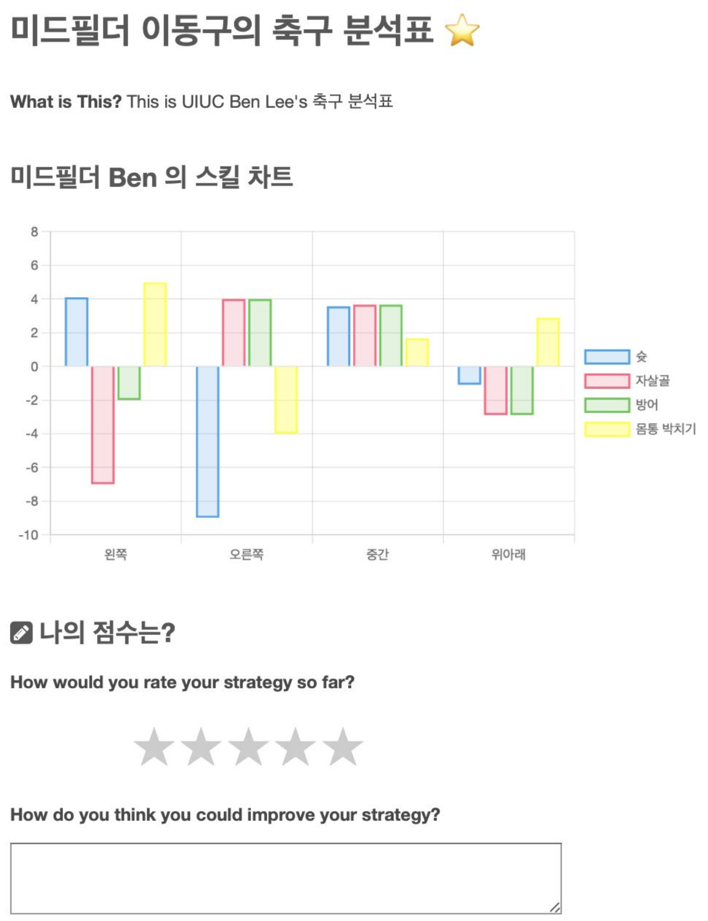
ON
OFF
60
40
20
0
A
B
C
D
E
F
G
1
2
5
10
20
50
100

Demo



```
const shap_label = [
  ['왼쪽'],
  ['오른쪽'],
  ['중간'],
  ['위아래']
]

const shap_data = {
  labels: shap_label,
  datasets: [
    {
      data: [4.11,-9,3.57,-1.09],
      label: '슛',
      backgroundColor: 'rgba(54, 162, 235, 0.2)',
      borderColor: 'rgb(54, 162, 235)'
    },
    {
      data: [-7,4.0,3.67,-2.89],
      label: '자살골',
      backgroundColor: 'rgba(255, 99, 132, 0.2)',
      borderColor: 'rgb(255, 99, 132)'
    },
    {
      data: [-2,4.0,3.67,-2.89],
      label: '방어',
      backgroundColor: 'rgba(85, 199, 82, 0.2)',
      borderColor: 'rgb(85, 199, 82)'
    },
    {
      data: [5,-4.0,1.67,2.89],
      label: '몸통 박치기',
      backgroundColor: 'rgba(255, 255, 0, 0.4)',
      borderColor: 'rgb(255, 255, 0)'
    }
```



미드필더 이동구의 축구 분석표 ⭐

What is This? This is UIUC Ben Lee's 축구 분석표

미드필더 Ben 의 스킬 차트

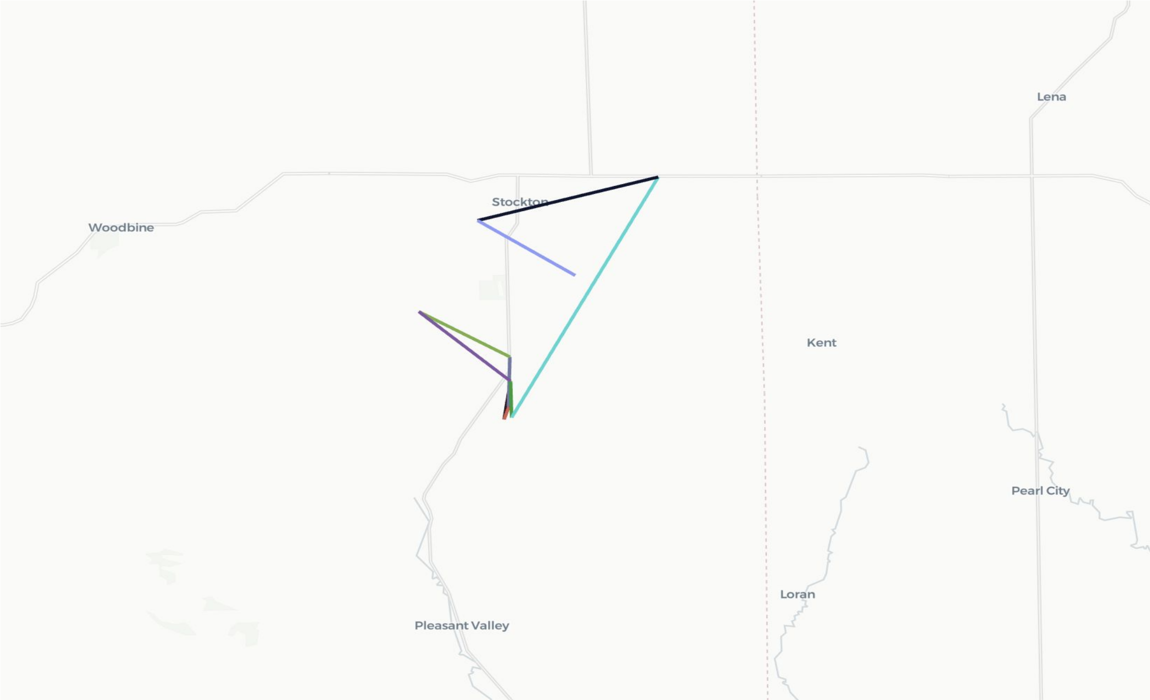
✎ 나의 점수는?

How would you rate your strategy so far?

★★★★★

How do you think you could improve your strategy?

Advanced Visualization - Trips Layer/Path Layer


Lena
Stockton
Woodbine
Kent
Pearl City
Loran
Pleasant Valley