

数据集成可视化部分

前端部分

1. 环境

考虑到只需要进行简单的图像展示，并且没有动态交互，故前端采用简单的html+js完成，并使用Echarts组件对数据进行可视化。

要运行前端项目需要输入以下指令下载echarts包

```
npm install echarts --save
```

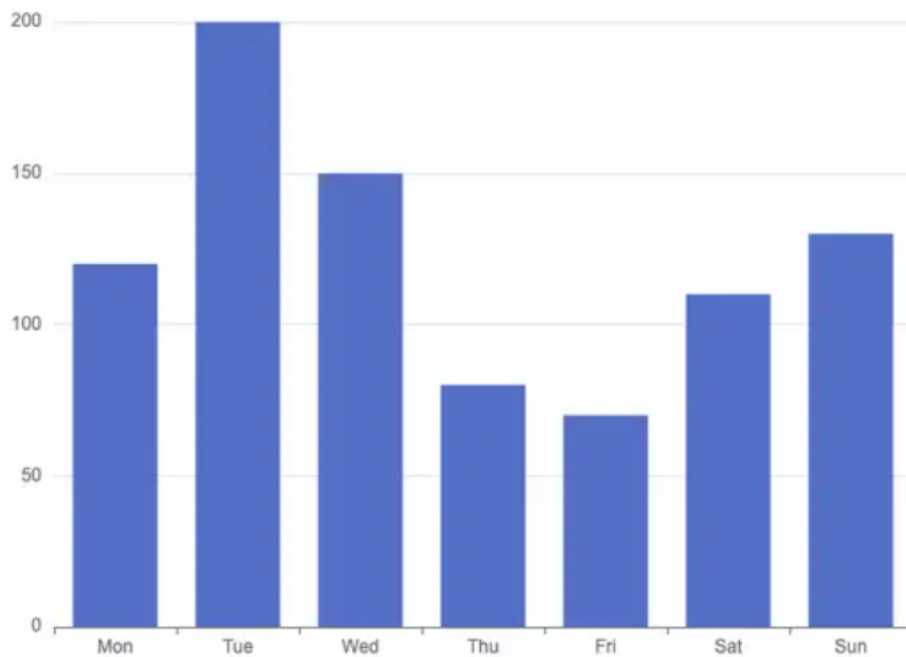
```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- 1. 安装echarts后, 引入js文件 -->
  <script src="../../node_modules/echarts/dist/echarts.js"></script>
  <title>Echarts</title>
</head>
<body>
  <!-- 定义显示图标的div -->
```

在html文件的head中引入echarts

2. 可视化方案

Echarts可以绘制包括但不限于：折线图、柱状图、饼图、散点图、日历坐标系、数据集等常用图像，考虑到作业二的数据可视化需要表现数据与时间的关系，且后端数据库发送来的数据为以用户区分的包含时间和消费信息的JSON数据，我们选取折线图、柱状图、日历坐标系等来进行数据展示。

2.1 柱状图



基础柱状图

JS

TS

Basic Bar

数据以用户为单位区分，所以图表展示出的数据集是某用户在该段时间内的消费行为。其中柱状图用以展示横轴为日期的数据，而纵轴为该用户在该日的消费金额。在数据量较少时，其可视化常规柱状图。当数据量很大时，横轴会缩放为时间段是不是具体某一天的时间点，并展示所有数据。在生成的柱状图下方会有所有数据的数据总览，通过在图表上滚动鼠标滚轮或者在总览条上拖拽显示区间可以查看时间划分更为详细的局部的数据。

```

chartData={
  title: {
    text: '该用户一周内消费柱状图'
  },
  toolbox:{
    feature:{
      dataZoom:{
        yAxisIndex:false
      },
      saveAsImage:{
        pixelRatio:2
      }
    }
  },
  tooltip: {},
  grid:{bottom:90},
  dataZoom:[
    {
      type:'inside'
    },
    {
      type:'slider'
    }
  ],
  legend: {
  },
  dataset:[{
    source:[]
  },],
  xAxis: {type:'category'},
  yAxis: {},
  series: []
};

```

代码如下。

使用数据集方式将数据放在dataset中的source里，并设置dataZoom实现数据时间尺度上的缩放

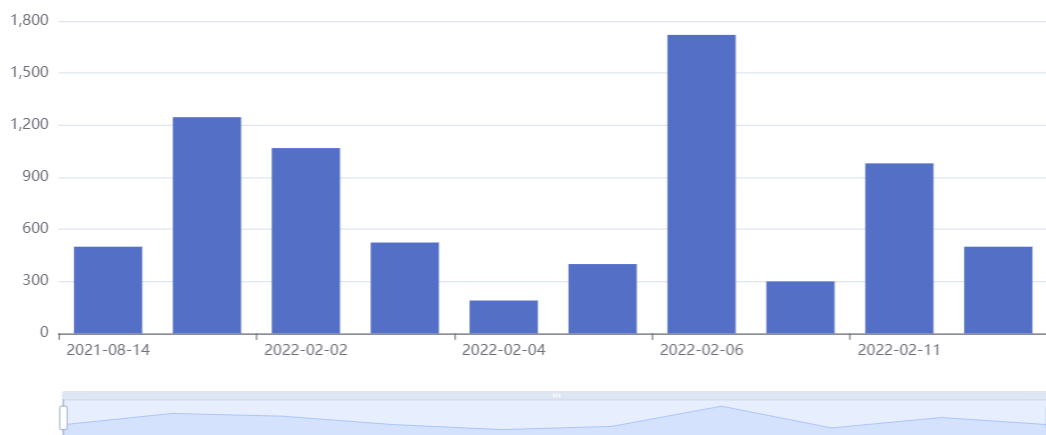
```

for(let date in result){
  chartData.dataset[0].source.push([date,parseInt(result[date])]);
}
for(let i=0;i<chartData.dataset[0].source.length-1;i++){
  for(let j=0;j<chartData.dataset[0].source.length-i-1;j++){
    if((chartData.dataset[0].source[j])[0]>(chartData.dataset[0].source[j+1])[0]){
      let temp=chartData.dataset[0].source[j];
      chartData.dataset[0].source[j]=chartData.dataset[0].source[j+1];
      chartData.dataset[0].source[j+1]=temp;
    }
  }
}

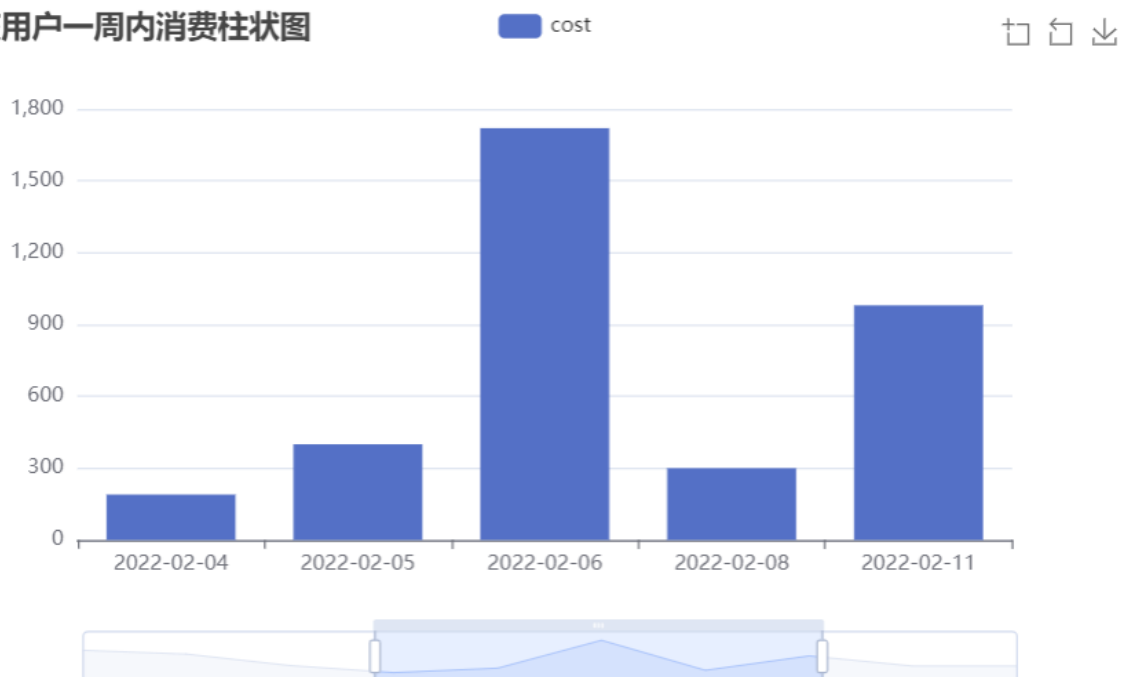
```

将每天的数据逐个压入数据集中，并进行排序以确保数据的展示是按照时间顺序（echarts中的柱状图默认是按照数据集中的数据顺序生成）

该用户一周内消费柱状图

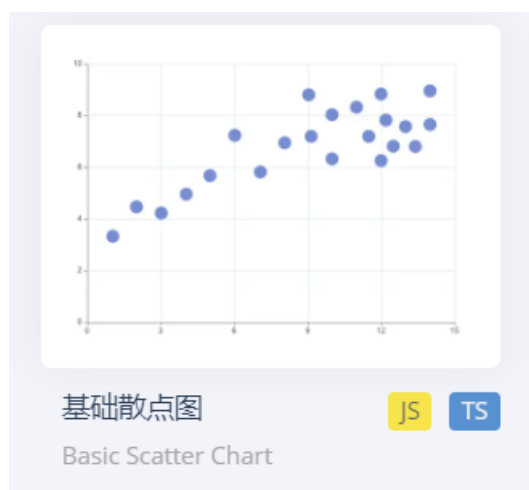


该用户一周内消费柱状图



实际效果图如上

2.2 散点图



散点图便于展示在一周的尺度内某用户的交易金额情况。通过横坐标为周的散点图可以直观的看出，该用户在每周几会进行交易、周几进行的交易金额较大等等

```

chartData={
  title: {
    text: '该用户消费散点图'
  },
  tooltip: {},
  legend: {
  },
  dataset:[{
    source:[
      ['day','cost'],
    ]
  },
  {
    transform:[],
  }
],
  xAxis: {type:'category'},
  yAxis: {},
  series: []
};

```

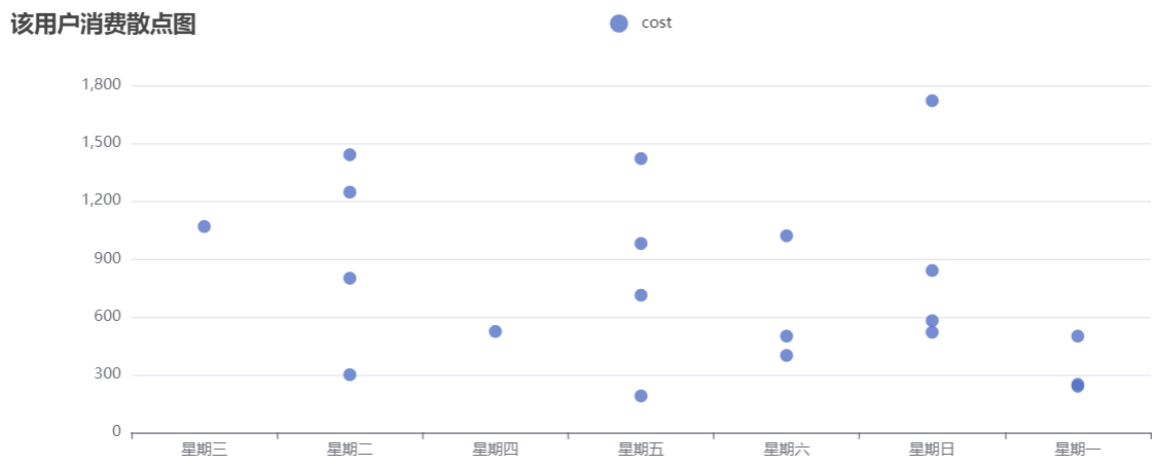
数据集的格式与柱状图类似

```

let weekArray=["星期日","星期一","星期二","星期三","星期四","星期五","星期六"]
for(let date in result){
  let week=weekArray[new Date(date).getDay()];
  chartData.dataset[0].source.push([week,parseInt(result[date])]);
}
chartData.series.push({type:'scatter'})
lineChart.setOption(chartData)

```

这里需要额外做的是将日期转换为星期中的某一天，利用getDay方法实现。



实际效果图如上。

2.3 日历热力图



日历热力图能非常直观的展示一年中每天该用户是否进行交易以及交易金额的大小。类似于github中的活跃度

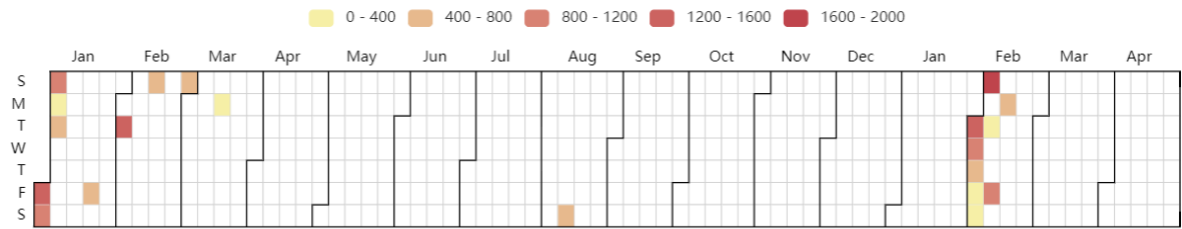
```
var option = {
  title: {
    top: 30,
    text: '该用户消费日历热力图'
  },
  tooltip: {},
  visualMap: {
    min: 0,
    max: 2000,
    type: 'piecewise',
    orient: 'horizontal',
    left: 'center',
    top: 65
  },
  calendar: {
    top: 120,
    left: 30,
    right: 30,
    cellSize: ['auto', 18],
    range: ['2021-01-01', '2022-04-30'],
    itemStyle: {
      borderWidth: 0.5
    }
  },
}
```

这里设置热力图显示的数据范围（图示中是2000），以及时间范围（图示中是2021-01-01到2022-4-30）

```
let data=[];
for(let date in result){
  data.push([date,parseInt(result[date])]);
}
```

数据的处理与柱状图类似

该用户消费日历热力图



生成的效果图如上