**《数据可视化原理与应用》实验报告**

实验题目：实验六 综合实验

专业班级： 数据科学与大数据技术1802

学生学号： 541807220111

学生姓名： 刁亚冲

指导教师： 张静

成 绩：

评 语：

1、实验目的

能够根据实际应用，对数据进行分析，然后利用所学的绘图工具进行，柱状图，饼状图和折线图等基本图形的绘制。

2、实验内容及要求

根据给定的乘用车销售数据绘制如下图。

1. 绘制SUV车型在各个省份从2016年1月到2017年12月间的销量折线图（也可以仅绘制其中任意5个省份分销量）。
2. 绘制河南省2016年各种车型销售量占比的饼状图。
3. 绘制河南省2016年1月到6月间，SUV，Sedan和MPV三种车型的销量柱状图。
4. 绘制河南省2016年1月到6月间和2017年1月到6月间SUV车型和Sedan的销售散点图，（提示：可以用横坐标表示2016年销量，纵坐标表示2017年销量，不同颜色表示不同车型）。
5. 绘制中国地图，根据2016年各个省份汽车销售总量，销量高的省份颜色较深。（根据个人情况可以选做）
6. 把以上图形绘制在一个页面上,中间是中国地图，左右两侧各两个图，可以自己设计布局。（根据个人情况可以选做）

3、实验环境

硬件环境：微型计算机一台

软件环境：Window XP/7/8/10操作系统、web开发环境

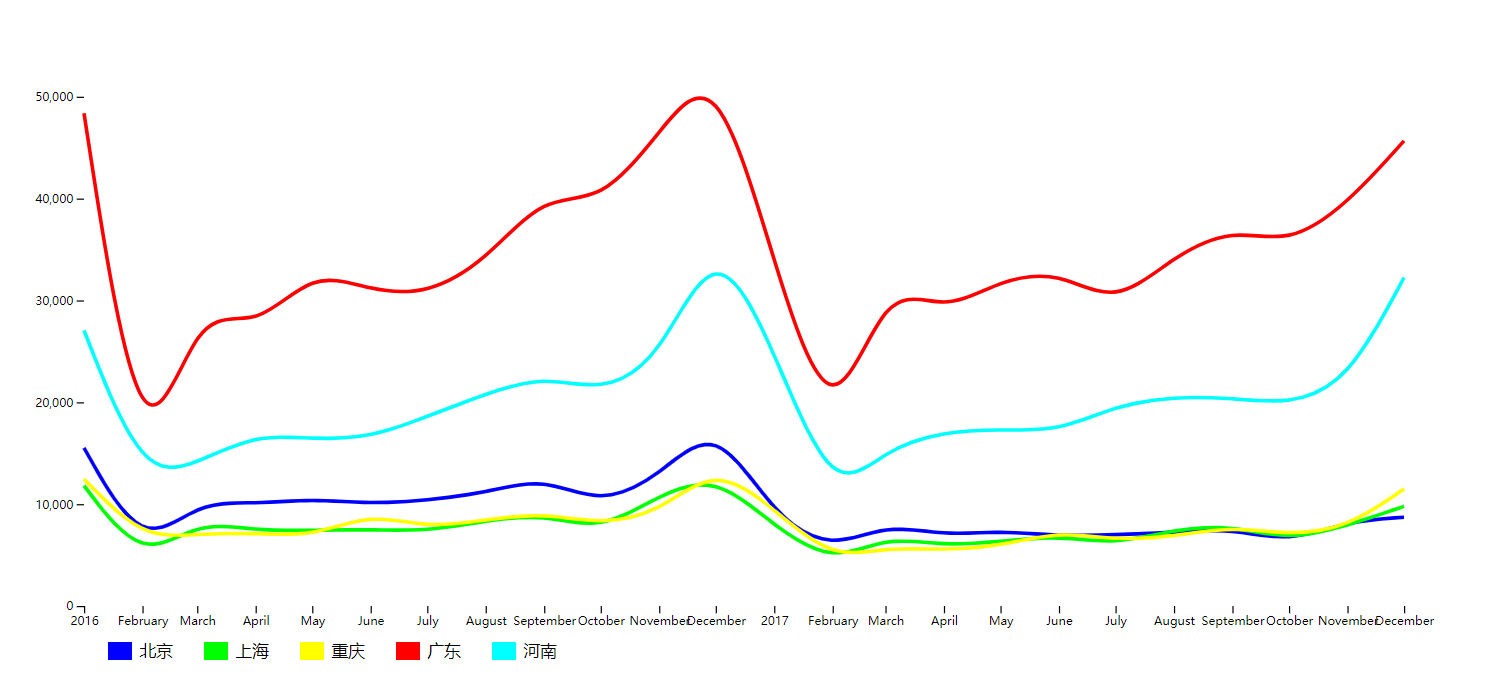
4、程序核心代码和运行效果

1. 绘制SUV车型在各个省份从2016年1月到2017年12月间的销量折线图（也可以仅绘制其中任意5个省份分销量）。

核心代码：

<html>  
<head>  
<meta charset="UTF-8">  
<title></title>  
<script src="../js/d3.min.js"></script>  
</head>  
<body>  
<script type="text/javascript">  
  
d3.json(&apos;SUV销售数据.json&apos;)  
.then(function (dataset) {  
var width = 1200;  
var height = 600;  
var svg = d3.select("body").append("svg")  
.attr("width", width)  
.attr("height", height);  
var padding = {  
top: 50,  
right: 50,  
bottom: 50,  
left: 50  
}; //外边框  
//计算salesVolume的最大值  
var salemax = 0;  
for(var i = 0; i < dataset.length; i++) {  
var currsale = d3.max(dataset[i].salesVolume, function(d) {  
return d[1];  
})  
if(currsale > salemax) {  
salemax = currsale;  
}  
}  
console.log(salemax)  
var xScale = d3.scaleTime()  
.domain([new Date(2016,0),new Date(2017,11)])  
.range([0,width - padding.left - padding.right])  
// .domain([201601,201602,201603,201604,201605,201606,201607,201608,201609,201610,201611,201612,  
//       201701,201702,201703,201704,201705,201706,201707,201708,201709,201710,201711,201712])  
// .range([0, 45, 90, 135, 180, 225, 270, 315, 360, 405, 450, 495, 540, 585, 630, 675, 720, 765, 810, 855, 900, 945, 990, 1035, 1080]);  
var yScale = d3.scaleLinear()  
.domain([0, salemax \* 1.1])  
.range([height - padding.top - padding.bottom, 0])  
  
//创建一个直线生成器  
var linePath = d3.line()  
.x(function(d) {  
//console.log(new Date(d[0]/100,d[0]%100));  
return xScale(new Date(d[0]/100,d[0]%100-1))  
})  
.y(function(d) {  
return yScale(d[1])  
}).curve(d3.curveBasis)  
//定义两个颜色（蓝色，绿色）  
var colors = [d3.rgb(0, 0, 255), d3.rgb(0, 255, 0), d3.rgb(255, 255, 0), d3.rgb(255, 0, 0), d3.rgb(0, 255, 255)];  
//添加路径  
//console.log(dataset[0].gdp)  
svg.selectAll("path")  
.data(dataset)  
.enter()  
.append("path")  
.attr("transform", "translate(" + padding.left + "," + padding.top + ")")  
.attr("d", function(d) {return linePath(d.salesVolume)})  
.attr("fill", "none")  
.attr("stroke-width", 3)  
.attr("stroke", function(d, i) {  
return colors[i]  
})  
//x轴  
//var xAxis = d3.svg.axis().scale(xScale).ticks(5).orient("bottom");  
  
//y轴  
//var yAxis = d3.svg.axis().scale(yScale).orient("left");  
var xAxis = d3.axisBottom(xScale) //指定x坐标轴和比例尺  
.ticks(24); //指定刻度的数量  
var yAxis = d3.axisLeft(yScale) //指定y坐标轴和比例尺  
.ticks(8); //指定刻度的数量  
svg.append("g")  
.attr("class", "axis")  
.attr("transform", "translate(" + padding.left + "," + (height - padding.bottom) + ")")  
.call(xAxis);  
svg.append("g")  
.attr("class", "axis")  
.attr("transform", "translate(" + padding.left + "," + padding.top + ")")  
.call(yAxis);  
  
//添加矩形图例和文字  
svg.selectAll("rect")  
.data(dataset)  
.enter()  
.append("rect") //添加矩形  
.attr("width", 20)  
.attr("height", 15)  
.attr("fill", function(d, i) {  
return colors[i]  
})  
.attr(&apos;x&apos;, function(d, i) {  
return padding.left + 80 \* i;  
})  
.attr("y", height - padding.bottom)  
.attr("transform", "translate(20,30)");  
//添加标签文字  
svg.selectAll(".text")  
.data(dataset)  
.enter()  
.append("text") //添加矩形边上的文字  
.attr("font-size", "14px")  
.attr("text-anchor", "middle")  
.attr("fill", "#000")  
.attr(&apos;x&apos;, function(d, i) {  
return padding.left + 80 \* i;  
})  
.attr("dx", "40px")  
.attr("dy", "0.9em")  
.attr("y", height - padding.bottom)  
.attr("transform", "translate(20,30)")  
.text(function(d) { //哪个国家的信息  
return d.province;  
})  
}).catch(error => console.log(error))  
</script>  
</body>  
</html>

运行效果：



1. 绘制河南省2016年各种车型销售量占比的饼状图。

核心代码：

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<script src="https://cdn.jsdelivr.net/npm/d3@6.5.0/dist/d3.min.js"></script>

<title>Document</title>

<style>

text{

font-size: 16px;

}

</style>

</head>

<body>

<div>

<svg id='svg6'></svg>

</div>

<script>

d3.csv('数据处理结果.csv', function (item) {

return {

x: item.bodyType,

y: item.salesVolume

}

}).then(function (oriData) {

const [width, height] = [960, 640];

let svg = d3.select('#svg6')

.attr('width', width)

.attr('height', height)

let g = svg.append('g')

.attr('transform', 'translate( 140, 40 )')

//设置饼图的半径

let radius = Math.min(width, height) \* 0.8 / 2

let arc = d3.arc()

.innerRadius(70)

.cornerRadius(10)

//饼图与文字相连的曲线起点

let pointStart = d3.arc()

.innerRadius(radius)

.outerRadius(radius)

//饼图与文字相连的曲线终点

let pointEnd = d3.arc()

.innerRadius(radius + 20)

.outerRadius(radius + 20)

let drawData = d3

.pie()

.value(function (d) {

return d.y

})

.sort(null)

.sortValues(null)

.startAngle(0)

.endAngle(Math.PI \* 2)

.padAngle(0.05)(oriData)

console.log(drawData)

let colorScale = d3

.scaleOrdinal()

.domain(d3.range(0, oriData.length))

.range(d3.schemeSet1);

g.append('g')

.attr('transform', 'translate( ' + radius + ', ' + radius + ' )')

.attr('stroke', 'steelblue')

.attr('stroke-width', 1)

.selectAll('path')

.data(drawData)

.enter()

.append('path')

.attr('fill', function (d) {

return colorScale(d.index)

})

.attr('d', function (d) {

d.outerRadius = radius;

return arc(d)

})

.on('mouseover', arcTween(radius + 20, 0))

.on('mouseout', arcTween(radius, 150))

.transition()

.duration(2000)

.attrTween('d', function (d) {

//初始加载时的过渡效果

let fn = d3.interpolate({

endAngle: d.startAngle

}, d)

return function (t) {

return arc(fn(t))

}

})

function arcTween(outerRadius, delay) {

// 设置缓动函数,为鼠标事件使用

return function () {

d3.select(this)

.transition()

.delay(delay)

.attrTween('d', function (d) {

let i = d3.interpolate(d.outerRadius, outerRadius)

return function (t) {

d.outerRadius = i(t)

return arc(d)

}

})

}

}

//文字层

let sum = d3.sum(oriData, d => d.y);

console.log(typeof radius)

svg.append('g')

.attr('transform', 'translate( ' + (radius + 140) + ', ' + (radius + 50) + ' )')

.selectAll('text')

.data(drawData)

.enter()

.append('text')

.attr('transform', function (d) {

return 'translate(' + arc.centroid(d) + ') ' +

'rotate(' + (-90 + (d.startAngle + (d.endAngle - d.startAngle) / 2) \* 180 / Math

.PI) + ')'

})

.attr('text-anchor', 'middle')

.attr('dominant-baseline', 'central')

.attr('font-size', '12px')

.text(function (d) {

return (d.data.y / sum \* 100).toFixed(2) + '%';

})

let legend = g.append('g')

.attr('transform', 'translate( ' + radius \* 2.5 + ', 0 )')

.selectAll('g')

.data(drawData)

.enter()

.append('g')

.attr('transform', function (d, i) {

return 'translate(0,' + i \* 20 + ')'

});

legend

.append('rect')

.attr('width', 40)

.attr('height', 25)

.attr('fill', function (d) {

return colorScale(d.index)

});

legend

.append('text')

.text(function (d) {

return d.data.x

})

.style('font-size', 12)

.attr('y', '1em')

.attr('x', '3.5em')

.attr('dy', 3)

//曲线层

g.append('g')

.attr('transform', 'translate( ' + radius + ', ' + radius + ' )')

.selectAll('path')

.data(drawData)

.enter()

.append('path')

.attr('d',

d3

.linkHorizontal()

.source(function (d) {

return pointStart.centroid(d)

})

.target(function (d) {

return pointEnd.centroid(d)

})

)

.style('stroke', '#999')

.style('stroke-width', 2)

.attr('fill', 'none')

//饼图外面的文字

g.append('g')

.attr('transform', 'translate( ' + radius + ', ' + radius + ' )')

.selectAll('path')

.data(drawData)

.enter()

.append('text')

.text(function (d) {

return d.data.x

})

.attr('x', function (d) {

return pointEnd.centroid(d)[0]

})

.attr('y', function (d) {

return pointEnd.centroid(d)[1]

})

.style('font-size', 16)

.attr('text-anchor', function (d) {

if (d.startAngle > Math.PI) {

return 'end'

}

})

.attr('dominant-baseline', function (d) {

if (d.index === 4) {

return 'hanging'

}

})

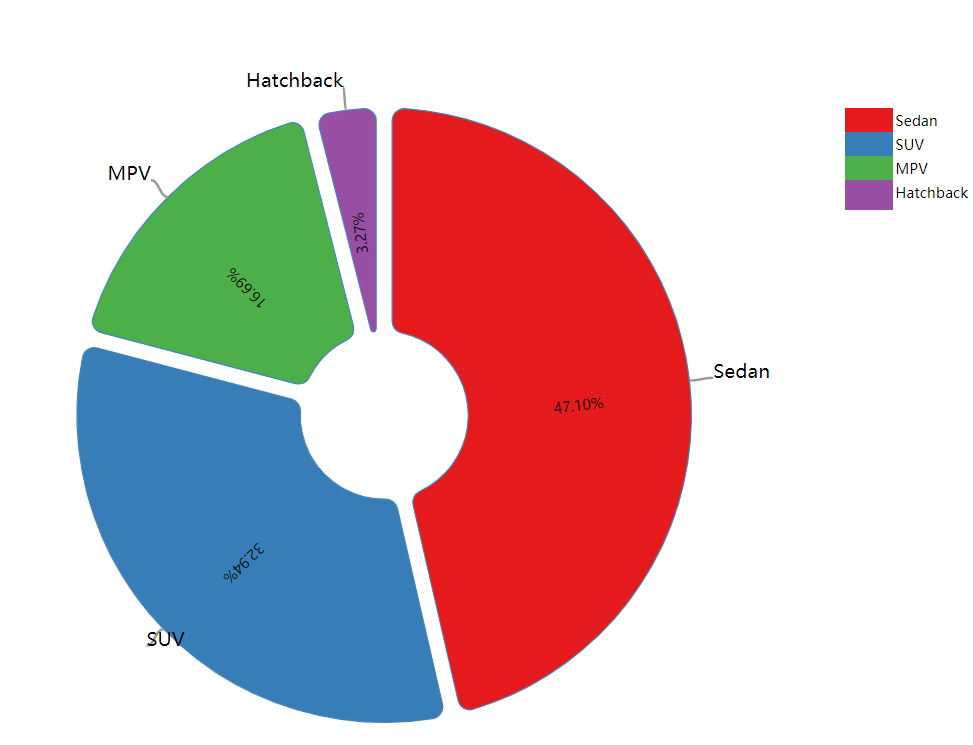
}).catch(error => console.log(error))

</script>

</body>

</html>

运行效果图：



1. 绘制河南省2016年1月到6月间，SUV，Sedan和MPV三种车型的销量柱状图。

核心代码：

<!DOCTYPE html>

<head>

<meta charset="utf-8">

<title>Bar</title>

</head>

<style>

.axis .domain {

display: none;

}

</style>

<svg width="960" height="500"></svg>

<script src="https://cdn.jsdelivr.net/npm/d3@6.5.0/dist/d3.min.js"></script>

<script>

var svg = d3.select("svg"),

margin = {top: 20, right: 20, bottom: 30, left: 80},

width = +svg.attr("width") - margin.left - margin.right,

height = +svg.attr("height") - margin.top - margin.bottom,

g = svg.append("g").attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var x0 = d3.scaleBand()

.rangeRound([0, width])

.paddingInner(0.1);

var x1 = d3.scaleBand()

.padding(0.05);

var y = d3.scaleLinear()

.rangeRound([height, 0]);

var z = d3.scaleOrdinal()

.range(["#98abc5", "#8a89a6", "#7b6888", "#6b486b", "#a05d56", "#d0743c"]);

let datasource=d3.csv("data.csv",

function (columns) {

return {bodyType:columns.bodyType,

one\_month:parseInt(columns.one\_month),

two\_month:parseInt(columns.two\_month),

three\_month:parseInt(columns.three\_month),

four\_month:parseInt(columns.four\_month),

five\_month:parseInt(columns.five\_month),

six\_month:parseInt(columns.six\_month)

}

}).then(function (data){

console.log(data)

var keys = data.columns.slice(1);

x0.domain(data.map(function (d) {

return d.bodyType;

}));

x1.domain(keys).rangeRound([0, x0.bandwidth()]);

y.domain([0, d3.max(data, function (d) {

return d3.max(keys, function (key) {

return d[key];

});

})]).nice();

g.append("g")

.selectAll("g")

.data(data)

.enter().append("g")

.attr("transform", function (d) {

return "translate(" + x0(d.bodyType) + ",0)";

})

.selectAll("rect")

.data(function (d) {

return keys.map(function (key) {

return {key: key, value: d[key]};

});

})

.enter().append("rect")

.attr("x", function (d) {

return x1(d.key);

})

.attr("y", function (d) {

return y(d.value);

})

.attr("width", x1.bandwidth())

.attr("height", function (d) {

return height - y(d.value);

})

.attr("fill", function (d) {

return z(d.key);

});

g.append("g")

.attr("class", "axis")

.attr("transform", "translate(0," + height + ")")

.call(d3.axisBottom(x0));

g.append("g")

.attr("class", "axis")

.call(d3.axisLeft(y).ticks(null, "s"))

.append("text")

.attr("x", 2)

.attr("y", y(y.ticks().pop()) + 0.5)

.attr("dy", "0.32em")

.attr("fill", "#000")

.attr("font-weight", "bold")

.attr("text-anchor", "start")

.text("Sales");

var legend = g.append("g")

.attr("font-family", "sans-serif")

.attr("font-size", 10)

.attr("text-anchor", "end")

.selectAll("g")

.data(keys.slice().reverse())

.enter().append("g")

.attr("transform", function (d, i) {

return "translate(0," + i \* 20 + ")";

});

legend.append("rect")

.attr("x", width - 19)

.attr("width", 19)

.attr("height", 19)

.attr("fill", z);

legend.append("text")

.attr("x", width - 24)

.attr("y", 9.5)

.attr("dy", "0.32em")

.text(function (d) {

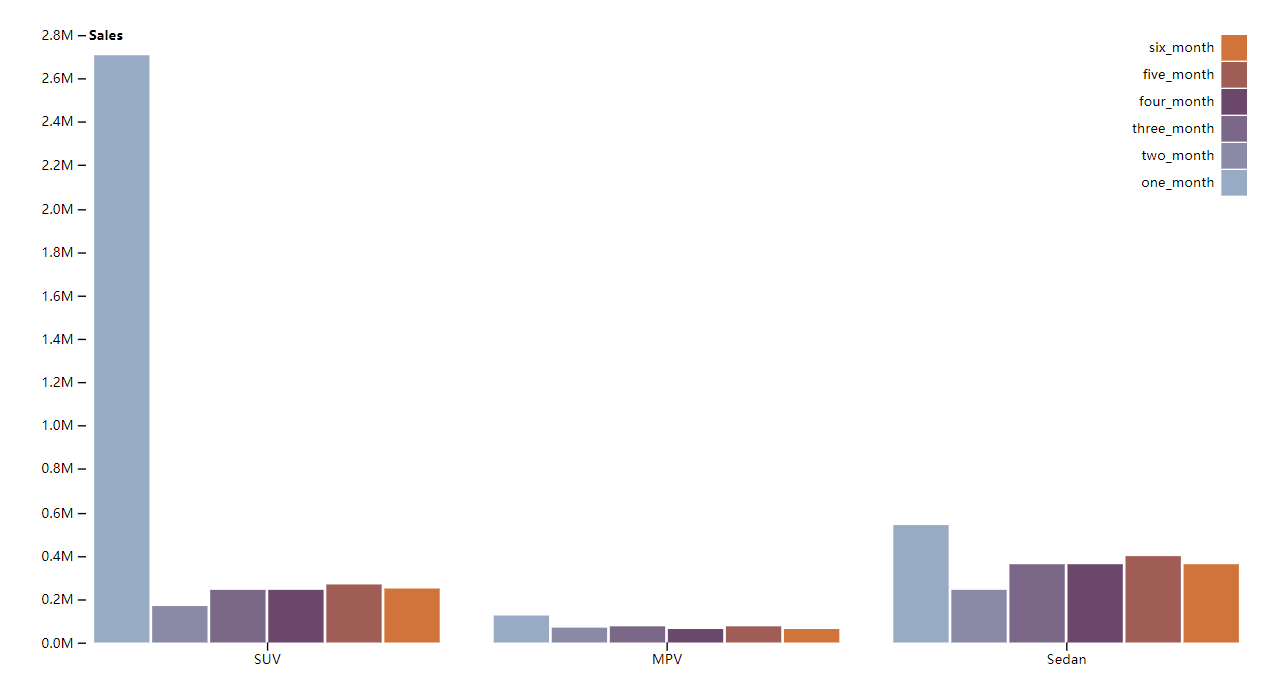
return d;

});

})

</script>

运行效果图：



1. 绘制河南省2016年1月到6月间和2017年1月到6月间SUV车型和Sedan的销售散点图，（提示：可以用横坐标表示2016年销量，纵坐标表示2017年销量，不同颜色表示不同车型）。

核心代码：

<html lang="en">

<head>

<meta charset="UTF-8" />

<meta http-equiv="X-UA-Compatible" content="IE=edge" />

<meta name="viewport" content="width=device-width, initial-scale=1.0" />

<title>第二次实验</title>

<script src="http://d3js.org/d3.v6.min.js"></script>

<style>

.svg-tooltip {

background-color: rgb(255, 255, 255);

box-shadow: rgba(0, 0, 0, 0.2) 1px 2px 10px;

border-radius: 0.1rem;

display: block;

font: 14px / 21px "Microsoft YaHei";

color: rgb(102, 102, 102);

text-overflow: ellipsis;

white-space: pre;

max-width: 320px;

position: absolute;

z-index: 300;

visibility: hidden;

border-width: 1px;

border-radius: 4px;

border-color: rgb(255, 255, 255);

padding: 10px;

}

</style>

</head>

<body>

<script>

//初始化 svg 画布大小

const width = 960,

height = 640;

// 添加 svg

let svg = d3

.select("body")

.append("svg")

.attr("width", width)

.attr("height", height);

// 设置边距

let margin = {

top: 30,

right: 80,

bottom: 30,

left: 80,

}

let colors = ["#c06f98", "#2c9678", "#2376b7"];

// 初始化tool tip

// Define the div for the hover effect

const tooltip = d3

.select("body")

.append("div")

.attr("name", "tooltip")

.attr("class", "svg-tooltip")

.text("I'm a circle!");

// 读数据

d3.csv("./four.csv")

.then(function (data) {

// 设置自定义数据格式 label 是年份，data 里面是 {x, y}对象

// 可以打印观看效果

console.log()

let suv = {

'label': 'SUV',

'data': data.filter(item => item.bodyType == 'SUV').map(item => {

return {

x: item.salesVolume1,

y: item.salesVolume2,

label: item.bodyType

}

})

}

let sedan = {

'label': 'Sedan',

'data': data.filter(item => item.bodyType == 'Sedan').map(item => {

return {

x: item.salesVolume1,

y: item.salesVolume2,

label: item.bodyType

}

})

}

var data = [suv, sedan];

/\* 生成所有点集 \*/

let Points = [];

data.forEach((item) => {

//遍历的方法 可以打印观看效果

item.data.forEach((item) => {

Points.push({

x: item.x,

y: item.y,

});

});

});

/\* 生成所有点集 \*/

/\* 画x轴 \*/

// x 比例尺

let x = d3

.scaleBand()

// extent 返回数组最大值和最小值

// nice 方法，自动选择最合适的区域

.domain(Points.map((item) => item.x))

//.nice()

.range([margin.left, width - margin.right]);

svg

.append("g")

// 回调函数 selection to call

// 设置 g 分组名字，平移

.call((g) =>

g

.attr("name", "xAxis")

.attr("transform", `translate(${0}, ${height - margin.bottom})`)

)

// 添加 x坐标轴 inner size 和 outer size

.call(d3.axisBottom(x).tickSizeOuter(0).tickSizeInner(10))

// 分割线 需要 clone （克隆） 否则会影响上面刻度

.call((g) =>

g

.selectAll(".tick line")

.clone()

.attr("y2", -(height - margin.top - margin.bottom))

.attr("opacity", 0.1)

)

.call((g) =>

g

/\* 坐标轴文字 \*/

.selectAll(".tick text")

.attr("font-size", 14)

);

/\* 画x轴 \*/

/\* 画y轴 \*/

// 同 x 轴

let y = d3

.scaleBand()

.domain(Points.map((item) => item.y))

//.nice()

.range([height - margin.bottom, margin.top]);

// 同 x 轴

svg

.append("g")

.call((g) =>

g

.attr("name", "yAxis")

.attr("transform", `translate(${margin.left}, ${0})`)

)

.call(d3.axisLeft(y).tickSizeOuter(0).tickSizeInner(10))

.call((g) =>

g

.selectAll(".tick line")

.clone()

.attr("x2", width - margin.right - margin.left)

.attr("opacity", 0.1)

)

.call((g) =>

g

/\* 坐标轴文字 \*/

.selectAll(".tick text")

.attr("font-size", 14)

);

/\* 画y轴 \*/

/\* 画散点 \*/

/\*

此处绑定数据为二次绑定，相当于双重循环，

\*/

svg

.append("g")

.call((g) => g.attr("name", "scatterPoint"))

.selectAll("g")

.data(data)

.join("g")

// 上面绑定 三年 数据， 每一个元素是每一年

.call((g) => g.attr("name", (element) => element.label))

.call((g) =>

g

.selectAll("circle")

.data((element) => element.data)

.join("circle")

// 第二次绑定数据，每一个元素是每一年的数据

.attr("cx", (element) => x(element.x))

.attr("cy", (element) => y(element.y))

.attr("r", 20)

.call((element) =>

element.on("mousemove", function (event, element) {

//console.log(element)

tooltip

.style("top", event.pageY - 10 + "px")

.style("left", event.pageX + 10 + "px")

.text(

`label: ${element.label}\nx: ${element.x}\ny: ${element.y}`

);

})

)

)

//注意此时，使用call不影响原来的selection

//绑定的数据还是三年，每一个元素都是一年整体

//所以直接对每一年整体添加颜色。

.call((g) =>

g

.attr("fill", (element, index) => colors[index])

.attr("fill-opacity", 0.5)

)

//添加图例

.call((g) =>

g

/\* 图例矩形 \*/

.append("rect")

.attr("x", width - margin.right + 10)

.attr("y", (element, index) => index \* 30 + margin.top)

.attr("width", 25)

.attr("height", 25)

)

//添加图例文字

.call((g) =>

g

/\* 图例文字 \*/

.append("text")

.attr("x", width - margin.right + 40)

.attr("y", (element, index) => index \* 30 + margin.top + 18)

.text((item) => item.label)

.attr("font-weight", "bold")

)

//交互效果，数据触摸和移出

.call((g) =>

g

.on("mouseover", function (event, element) {

d3.select(this).attr("fill-opacity", 1);

// Shows the tooltip

tooltip.style("visibility", "visible");

})

.on("mouseout", function (event, element) {

d3.select(this)

.transition()

.duration(500)

.attr("fill-opacity", 0.5);

// Deletes the tool tip (i.e just makes it invisible)...

tooltip.style("visibility", "hidden");

})

);

/\* 画散点 \*/

console.log(Points, data);

})

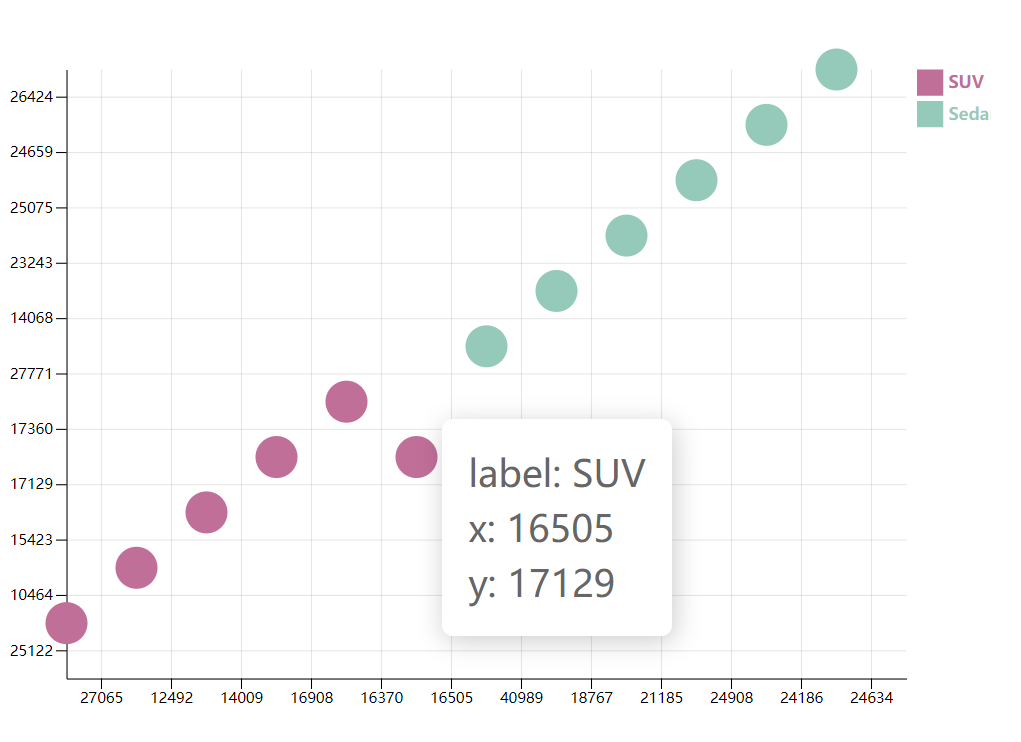
.catch((error) => console.log(error));

</script>

</body>

</html>

运行效果图：



1. 绘制中国地图，根据2016年各个省份汽车销售总量，销量高的省份颜色较深。（根据个人情况可以选做）

核心代码：

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title></title>

<style>

.southchinasea {

stroke: black;

stroke-width: 1px;

fill: red;

}

</style>

</head>

<body>

<script src="https://cdn.jsdelivr.net/npm/d3@6.5.0/dist/d3.min.js"></script>

<script>

let width = 1000;

let height = 1000;

let svg = d3.select("body").append("svg")

.attr("width", width)

.attr("height", height)

.attr("transform", "translate(0,0)");

d3.csv("实验六-5.csv").then(function (sales\_data){

console.log("各省份销售量",sales\_data);

let projection = d3.geoMercator()

.center([107, 31])

.scale(850)

.translate([width/2, height/2]);

let path = d3.geoPath()

.projection(projection);

d3.json("实验六-5.json").then(function(root){

const sales = [];

console.log("数组数据1",root);

root.features.forEach(function(element1) {

let flog = 0;

sales\_data.forEach(function (element2){

if(element1.properties.name == element2.province){

flog = 1;

sales.push(element2.salesVolume);

}

})

if(flog == 0)

sales.push(0);

});

console.log("数组数据",sales);

let color = d3.scaleSequentialSqrt(d3.extent(sales), d3.interpolateReds )

//let color = d3.scaleOrdinal(d3.quantize(d3.interpolateRainbow, 40))

console.log(root)

svg.append("g").selectAll("path")

.data(root.features )

.enter()

.append("path")

.attr("stroke","#000")

.attr("stroke-width",1)

.attr("fill", function(d,i){

return color(sales[d.properties.id-1]);

})

.attr("d", path )

.on("mouseover",function(d,i){

d3.select(this)

.attr("fill","yellow");

})

.on("mouseout",function(d,i){

d3.select(this)

.attr("fill",function(d,i){

return color(sales[d.properties.id-1]);

});

});

let texts = svg.selectAll(".texts")

.data(root.features)

.enter()

.append("text")

.attr("class", "texts")

.attr('fill','#e1e1e1')

.attr("font-size","15px")

.text(function(d){return d.properties.name;})

.attr("transform", function(d) {

let centroid = path.centroid(d),

x = centroid[0]-5,

y = centroid[1];

if((d.properties.name=="河北")||(d.properties.name=="澳门")||(d.properties.name=="安徽"))

y=y+30;

return "translate(" + x + ", " + y + ")";

})

});

//添加南海图例

d3.xml("southchinasea.svg").then( function(xmlDocument) {

console.log(xmlDocument)

svg.html(function(d){

return d3.select(this).html() + xmlDocument.getElementById("southchinasea").outerHTML;

});

d3.select("#southchinasea")

.attr("transform","translate(800,600)scale(0.5)")

.attr("class","southchinasea");

});

})

</script>

</body>

</html>

1. 把以上图形绘制在一个页面上,中间是中国地图，左右两侧各两个图，可以自己设计布局。（根据个人情况可以选做）

核心代码：

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>数据可视化原理实验报告</title>

<link rel="stylesheet" href="./css/index.css">

<link rel="stylesheet" href="./css/globle.css">

<script src="https://cdn.jsdelivr.net/npm/d3@6.5.0/dist/d3.min.js"></script>

<style>

.axis .domain {

display: none;

}

.svg-tooltip {

background-color: rgb(255, 255, 255);

box-shadow: rgba(0, 0, 0, 0.2) 1px 2px 10px;

border-radius: 0.1rem;

display: block;

font: 14px / 21px "Microsoft YaHei";

color: rgb(102, 102, 102);

text-overflow: ellipsis;

white-space: pre;

max-width: 320px;

position: absolute;

z-index: 300;

visibility: hidden;

border-width: 1px;

border-radius: 4px;

border-color: rgb(255, 255, 255);

padding: 10px;

}

</style>

</head>

<body>

<div class="main">

<div class="top\_title">

</div>

<div class="datas">

<div class="data\_left">

<div class="left\_top">

<div id="Business\_type" class="left\_top\_main left\_bottom\_main"></div>

</div>

</div>

<div class="data\_middle">

</div>

<div id="map" class="middle\_bottom"></div>

</div>

<div class="data\_right">

<div class="right\_top">

<div id="Land\_analysis" class="left\_top\_main right\_top\_main"></div>

</div>

<div class="right\_bottom">

<div id="Information\_Delivery" class="left\_top\_main right\_top\_main"></div>

</div>

</div>

</div>

</div>

</body>

</html>

<script src="./js/jquery-1.8.3.min.js"></script>

<script src="https://img.hcharts.cn/highmaps/highmaps.js"></script>

<script src="https://data.jianshukeji.com/geochina/hunan.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/echarts/4.2.1/echarts-en.min.js"></script>

<script>

//初始化 svg 画布大小

const width = 960,

height = 640;

// 添加 svg

let svg = d3

.select("#Business\_type")

.append("svg")

.attr("width", width)

.attr("height", height)

.attr('transform', `translate(-300, 200)scale(0.4)`)

let margin = {

top: 30,

right: 80,

bottom: 30,

left: 80,

};

let colors = ["#c06f98", "#2c9678", "#2376b7"];

// 初始化tool tip

// Define the div for the hover effect

const tooltip = d3

.select("body")

.append("div")

.attr("name", "tooltip")

.attr("class", "svg-tooltip")

.text("I'm a circle!");

// 读数据

d3.csv("./four.csv")

.then(function (data) {

// 设置自定义数据格式 label 是年份，data 里面是 {x, y}对象

// 可以打印观看效果

console.log()

let suv = {

'label': 'SUV',

'data': data.filter(item => item.bodyType == 'SUV').map(item => {

return {

x: item.salesVolume1,

y: item.salesVolume2,

label: item.bodyType

}

})

}

let sedan = {

'label': 'Sedan',

'data': data.filter(item => item.bodyType == 'Sedan').map(item => {

return {

x: item.salesVolume1,

y: item.salesVolume2,

label: item.bodyType

}

})

}

var data = [suv, sedan];

/\* 生成所有点集 \*/

let Points = [];

data.forEach((item) => {

//遍历的方法 可以打印观看效果

item.data.forEach((item) => {

Points.push({

x: item.x,

y: item.y,

});

});

});

/\* 生成所有点集 \*/

/\* 画x轴 \*/

// x 比例尺

let x = d3

.scaleBand()

// extent 返回数组最大值和最小值

// nice 方法，自动选择最合适的区域

.domain(Points.map((item) => item.x))

//.nice()

.range([margin.left, width - margin.right]);

svg

.append("g")

// 回调函数 selection to call

// 设置 g 分组名字，平移

.call((g) =>

g

.attr("name", "xAxis")

.attr("transform", `translate(${0}, ${height - margin.bottom})`)

)

// 添加 x坐标轴 inner size 和 outer size

.call(d3.axisBottom(x).tickSizeOuter(0).tickSizeInner(10))

// 分割线 需要 clone （克隆） 否则会影响上面刻度

.call((g) =>

g

.selectAll(".tick line")

.clone()

.attr("y2", -(height - margin.top - margin.bottom))

.attr("opacity", 0.1)

)

.call((g) =>

g

/\* 坐标轴文字 \*/

.selectAll(".tick text")

.attr("font-size", 14)

);

/\* 画x轴 \*/

/\* 画y轴 \*/

// 同 x 轴

let y = d3

.scaleBand()

.domain(Points.map((item) => item.y))

//.nice()

.range([height - margin.bottom, margin.top]);

// 同 x 轴

svg

.append("g")

.call((g) =>

g

.attr("name", "yAxis")

.attr("transform", `translate(${margin.left}, ${0})`)

)

.call(d3.axisLeft(y).tickSizeOuter(0).tickSizeInner(10))

.call((g) =>

g

.selectAll(".tick line")

.clone()

.attr("x2", width - margin.right - margin.left)

.attr("opacity", 0.1)

)

.call((g) =>

g

/\* 坐标轴文字 \*/

.selectAll(".tick text")

.attr("font-size", 14)

);

/\* 画y轴 \*/

/\* 画散点 \*/

/\*

此处绑定数据为二次绑定，相当于双重循环，

\*/

svg

.append("g")

.call((g) => g.attr("name", "scatterPoint"))

.selectAll("g")

.data(data)

.join("g")

// 上面绑定 三年 数据， 每一个元素是每一年

.call((g) => g.attr("name", (element) => element.label))

.call((g) =>

g

.selectAll("circle")

.data((element) => element.data)

.join("circle")

// 第二次绑定数据，每一个元素是每一年的数据

.attr("cx", (element) => x(element.x))

.attr("cy", (element) => y(element.y))

.attr("r", 20)

.call((element) =>

element.on("mousemove", function (event, element) {

//console.log(element)

tooltip

.style("top", event.pageY - 10 + "px")

.style("left", event.pageX + 10 + "px")

.text(

`label: ${element.label}\nx: ${element.x}\ny: ${element.y}`

);

})

)

)

//注意此时，使用call不影响原来的selection

//绑定的数据还是三年，每一个元素都是一年整体

//所以直接对每一年整体添加颜色。

.call((g) =>

g

.attr("fill", (element, index) => colors[index])

.attr("fill-opacity", 0.5)

)

//添加图例

.call((g) =>

g

/\* 图例矩形 \*/

.append("rect")

.attr("x", width - margin.right + 10)

.attr("y", (element, index) => index \* 30 + margin.top)

.attr("width", 25)

.attr("height", 25)

)

//添加图例文字

.call((g) =>

g

/\* 图例文字 \*/

.append("text")

.attr("x", width - margin.right + 40)

.attr("y", (element, index) => index \* 30 + margin.top + 18)

.text((item) => item.label)

.attr("font-weight", "bold")

)

//交互效果，数据触摸和移出

.call((g) =>

g

.on("mouseover", function (event, element) {

d3.select(this).attr("fill-opacity", 1);

// Shows the tooltip

tooltip.style("visibility", "visible");

})

.on("mouseout", function (event, element) {

d3.select(this)

.transition()

.duration(500)

.attr("fill-opacity", 0.5);

// Deletes the tool tip (i.e just makes it invisible)...

tooltip.style("visibility", "hidden");

})

);

/\* 画散点 \*/

console.log(Points, data);

})

.catch((error) => console.log(error));

</script>

<script>

main()

function main() {

//初始化 svg 画布大小

const width = 1000,

height = 1000;

// 添加 svg

let svg = d3

.select("#Information\_Delivery")

.append("svg")

.attr("width", width)

.attr("height", height)

.attr('transform', `translate(200,-2200) scale(0.5)`)

d3.csv("销售数据-5.csv").then(function (sales\_data) {

console.log("各省份销售量", sales\_data);

let projection = d3.geoMercator()

.center([107, 31])

.scale(850)

.translate([width / 2, height / 2]);

let path = d3.geoPath()

.projection(projection);

d3.json("实验六-5.json").then(function (root) {

const sales = [];

console.log("数组数据1", root);

root.features.forEach(function (element1) {

let flog = 0;

sales\_data.forEach(function (element2) {

if (element1.properties.name == element2.province) {

flog = 1;

sales.push(element2.salesVolume);

}

})

if (flog == 0)

sales.push(0);

});

console.log("数组数据", sales);

let color = d3.scaleSequentialSqrt(d3.extent(sales), d3.interpolateReds)

//let color = d3.scaleOrdinal(d3.quantize(d3.interpolateRainbow, 40))

console.log(root)

svg.append("g").selectAll("path")

.data(root.features)

.enter()

.append("path")

.attr("stroke", "#000")

.attr("stroke-width", 1)

.attr("fill", function (d, i) {

return color(sales[d.properties.id - 1]);

})

.attr("d", path)

.on("mouseover", function (d, i) {

d3.select(this)

.attr("fill", "yellow");

})

.on("mouseout", function (d, i) {

d3.select(this)

.attr("fill", function (d, i) {

return color(sales[d.properties.id - 1]);

});

});

let texts = svg.selectAll(".texts")

.data(root.features)

.enter()

.append("text")

.attr("class", "texts")

.attr('fill', '#e1e1e1')

.attr("font-size", "15px")

.text(function (d) {

return d.properties.name;

})

.attr("transform", function (d) {

let centroid = path.centroid(d),

x = centroid[0] - 5,

y = centroid[1];

if ((d.properties.name == "河北") || (d.properties.name == "澳门") || (d.properties.name == "安徽"))

y = y + 30;

return "translate(" + x + ", " + y + ")";

})

});

//添加南海图例

d3.xml("southchinasea.svg").then(function (xmlDocument) {

console.log(xmlDocument)

svg.html(function (d) {

return d3.select(this).html() + xmlDocument.getElementById("southchinasea").outerHTML;

});

d3.select("#southchinasea")

.attr("transform", "translate(800,600)scale(0.5)")

.attr("class", "southchinasea");

});

})

}

</script>

<script>

main()

function main() {

//初始化 svg 画布大小

const width = 900,

height = 900;

// 添加 svg

let svg = d3

.select("#Land\_analysis")

.append("svg")

.attr("width", width)

.attr("height", height)

.attr('transform', `translate(700,-1500) scale(0.4)`)

d3.csv('数据处理结果.csv', function (item) {

return {

x: item.bodyType,

y: item.salesVolume

}

}).then(function (oriData) {

const [width, height] = [960, 640];

let g = svg.append('g')

.attr('transform', 'translate( 140, 40 )')

//设置饼图的半径

let radius = Math.min(width, height) \* 0.8 / 2

let arc = d3.arc()

.innerRadius(70)

.cornerRadius(10)

//饼图与文字相连的曲线起点

let pointStart = d3.arc()

.innerRadius(radius)

.outerRadius(radius)

//饼图与文字相连的曲线终点

let pointEnd = d3.arc()

.innerRadius(radius + 20)

.outerRadius(radius + 20)

let drawData = d3

.pie()

.value(function (d) {

return d.y

})

.sort(null)

.sortValues(null)

.startAngle(0)

.endAngle(Math.PI \* 2)

.padAngle(0.05)(oriData)

console.log(drawData)

let colorScale = d3

.scaleOrdinal()

.domain(d3.range(0, oriData.length))

.range(d3.schemeSet1);

g.append('g')

.attr('transform', 'translate( ' + radius + ', ' + radius + ' )')

.attr('stroke', 'steelblue')

.attr('stroke-width', 1)

.selectAll('path')

.data(drawData)

.enter()

.append('path')

.attr('fill', function (d) {

return colorScale(d.index)

})

.attr('d', function (d) {

d.outerRadius = radius;

return arc(d)

})

.on('mouseover', arcTween(radius + 20, 0))

.on('mouseout', arcTween(radius, 150))

.transition()

.duration(2000)

.attrTween('d', function (d) {

//初始加载时的过渡效果

let fn = d3.interpolate({

endAngle: d.startAngle

}, d)

return function (t) {

return arc(fn(t))

}

})

function arcTween(outerRadius, delay) {

// 设置缓动函数,为鼠标事件使用

return function () {

d3.select(this)

.transition()

.delay(delay)

.attrTween('d', function (d) {

let i = d3.interpolate(d.outerRadius, outerRadius)

return function (t) {

d.outerRadius = i(t)

return arc(d)

}

})

}

}

//文字层

let sum = d3.sum(oriData, d => d.y);

console.log(typeof radius)

svg.append('g')

.attr('transform', 'translate( ' + (radius + 140) + ', ' + (radius + 50) + ' )')

.selectAll('text')

.data(drawData)

.enter()

.append('text')

.attr('transform', function (d) {

return 'translate(' + arc.centroid(d) + ') ' +

'rotate(' + (-90 + (d.startAngle + (d.endAngle - d.startAngle) / 2) \* 180 / Math

.PI) + ')'

})

.attr('text-anchor', 'middle')

.attr('dominant-baseline', 'central')

.attr('font-size', '12px')

.text(function (d) {

return (d.data.y / sum \* 100).toFixed(2) + '%';

})

let legend = g.append('g')

.attr('transform', 'translate( ' + radius \* 2.5 + ', 0 )')

.selectAll('g')

.data(drawData)

.enter()

.append('g')

.attr('transform', function (d, i) {

return 'translate(0,' + i \* 20 + ')'

});

legend

.append('rect')

.attr('width', 40)

.attr('height', 25)

.attr('fill', function (d) {

return colorScale(d.index)

});

legend

.append('text')

.text(function (d) {

return d.data.x

})

.style('font-size', 12)

.attr('y', '1em')

.attr('x', '3.5em')

.attr('dy', 3)

//曲线层

g.append('g')

.attr('transform', 'translate( ' + radius + ', ' + radius + ' )')

.selectAll('path')

.data(drawData)

.enter()

.append('path')

.attr('d',

d3

.linkHorizontal()

.source(function (d) {

return pointStart.centroid(d)

})

.target(function (d) {

return pointEnd.centroid(d)

})

)

.style('stroke', '#999')

.style('stroke-width', 2)

.attr('fill', 'none')

//饼图外面的文字

g.append('g')

.attr('transform', 'translate( ' + radius + ', ' + radius + ' )')

.selectAll('path')

.data(drawData)

.enter()

.append('text')

.text(function (d) {

return d.data.x

})

.attr('x', function (d) {

return pointEnd.centroid(d)[0]

})

.attr('y', function (d) {

return pointEnd.centroid(d)[1]

})

.style('font-size', 16)

.attr('text-anchor', function (d) {

if (d.startAngle > Math.PI) {

return 'end'

}

})

.attr('dominant-baseline', function (d) {

if (d.index === 4) {

return 'hanging'

}

})

}).catch(error => console.log(error))

}

</script>

<script>

main()

function main() {

//初始化 svg 画布大小

const width = 1200,

height = 600;

// 添加 svg

let svg = d3

.select("#Business\_type")

.append("svg")

.attr("width", width)

.attr("height", height)

.attr('transform', `translate(-350,-810) scale(0.4)`)

d3.json('SUV销售数据.json')

.then(function (dataset) {

var padding = {

top: 50,

right: 50,

bottom: 50,

left: 50

}; //外边框

//计算salesVolume的最大值

var salemax = 0;

for (var i = 0; i < dataset.length; i++) {

var currsale = d3.max(dataset[i].salesVolume, function (d) {

return d[1];

})

if (currsale > salemax) {

salemax = currsale;

}

}

console.log(salemax)

var xScale = d3.scaleTime()

.domain([new Date(2016, 0), new Date(2017, 11)])

.range([0, width - padding.left - padding.right])

// .domain([201601,201602,201603,201604,201605,201606,201607,201608,201609,201610,201611,201612,

// 201701,201702,201703,201704,201705,201706,201707,201708,201709,201710,201711,201712])

// .range([0, 45, 90, 135, 180, 225, 270, 315, 360, 405, 450, 495, 540, 585, 630, 675, 720, 765, 810, 855, 900, 945, 990, 1035, 1080]);

var yScale = d3.scaleLinear()

.domain([0, salemax \* 1.1])

.range([height - padding.top - padding.bottom, 0])

//创建一个直线生成器

var linePath = d3.line()

.x(function (d) {

//console.log(new Date(d[0]/100,d[0]%100));

return xScale(new Date(d[0] / 100, d[0] % 100 - 1))

})

.y(function (d) {

return yScale(d[1])

}).curve(d3.curveBasis)

//定义两个颜色（蓝色，绿色）

var colors = [d3.rgb(0, 0, 255), d3.rgb(0, 255, 0), d3.rgb(255, 255, 0), d3.rgb(255, 0, 0), d3.rgb(0, 255,

255)];

//添加路径

//console.log(dataset[0].gdp)

svg.selectAll("path")

.data(dataset)

.enter()

.append("path")

.attr("transform", "translate(" + padding.left + "," + padding.top + ")")

.attr("d", function (d) {

return linePath(d.salesVolume)

})

.attr("fill", "none")

.attr("stroke-width", 3)

.attr("stroke", function (d, i) {

return colors[i]

})

//x轴

//var xAxis = d3.svg.axis().scale(xScale).ticks(5).orient("bottom");

//y轴

//var yAxis = d3.svg.axis().scale(yScale).orient("left");

var xAxis = d3.axisBottom(xScale) //指定x坐标轴和比例尺

.ticks(24); //指定刻度的数量

var yAxis = d3.axisLeft(yScale) //指定y坐标轴和比例尺

.ticks(8); //指定刻度的数量

svg.append("g")

.attr("class", "axis")

.attr("transform", "translate(" + padding.left + "," + (height - padding.bottom) + ")")

.call(xAxis);

svg.append("g")

.attr("class", "axis")

.attr("transform", "translate(" + padding.left + "," + padding.top + ")")

.call(yAxis);

//添加矩形图例和文字

svg.selectAll("rect")

.data(dataset)

.enter()

.append("rect") //添加矩形

.attr("width", 20)

.attr("height", 15)

.attr("fill", function (d, i) {

return colors[i]

})

.attr('x', function (d, i) {

return padding.left + 80 \* i;

})

.attr("y", height - padding.bottom)

.attr("transform", "translate(20,30)");

//添加标签文字

svg.selectAll(".text")

.data(dataset)

.enter()

.append("text") //添加矩形边上的文字

.attr("font-size", "14px")

.attr("text-anchor", "middle")

.attr("fill", "#000")

.attr('x', function (d, i) {

return padding.left + 80 \* i;

})

.attr("dx", "40px")

.attr("dy", "0.9em")

.attr("y", height - padding.bottom)

.attr("transform", "translate(20,30)")

.text(function (d) { //哪个国家的信息

return d.province;

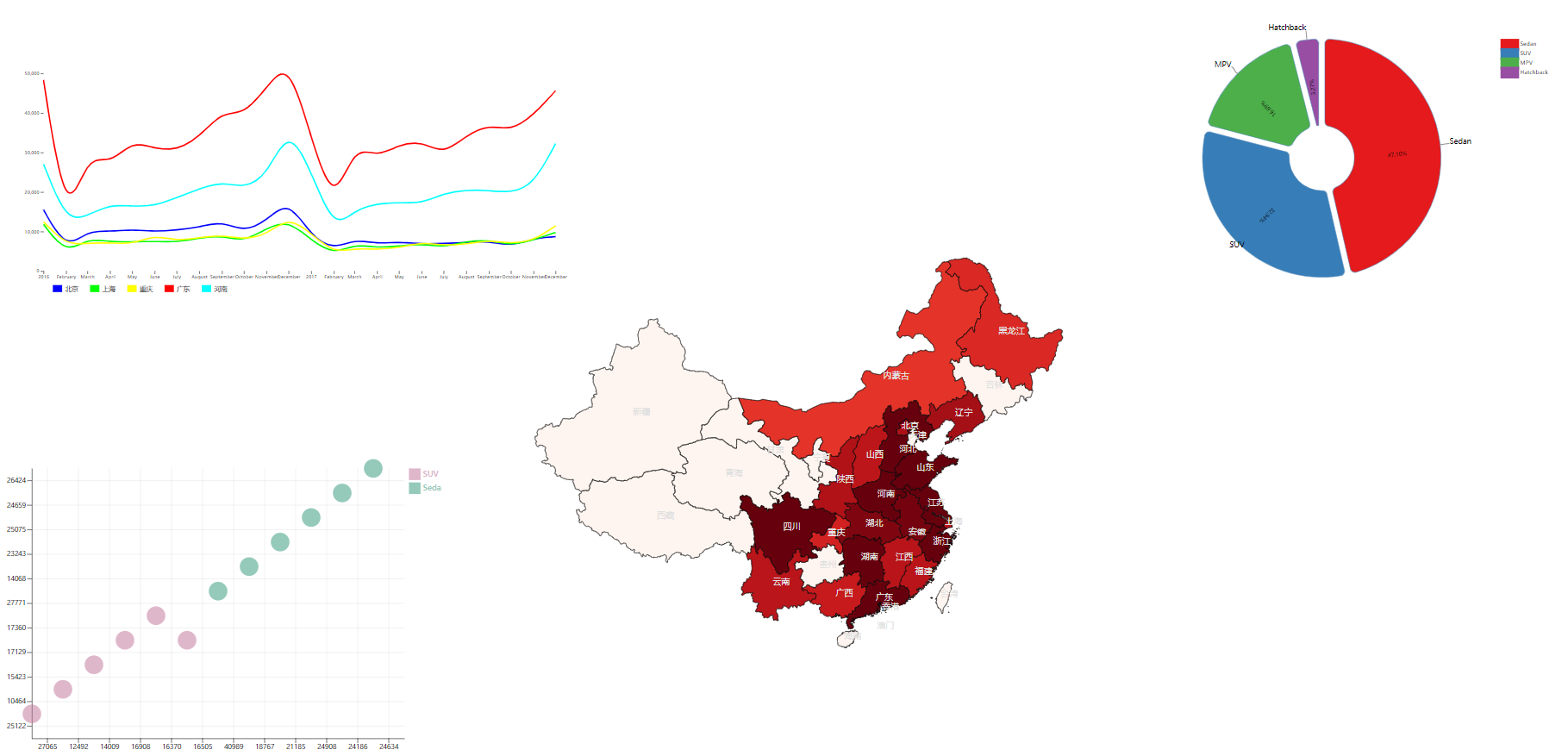
})

}).catch(error => console.log(error))

}

</script>

运行效果图:



5、实验小结

通过本次实验我根据实际应用，对数据进行分析，然后利用所学的绘图工具进行，柱状图，饼状图和折线图等基本图形的绘制。对数据可视化的作用和技术更加的熟悉，数据可视化是指将大型数据集中的数据以图形图像形式表示，并利用数据分析和开发工具发现其中未知信息的处理过程。数据可视化的目的是对数据进行可视化处理，以使能够明确、有效地传递信息。比起枯燥乏味的数值，人类对于大小、位置、浓淡、颜色、形状等能够有更好、更快的认识，经过可视化之后的数据能够加深人类对于数据的理解和记忆。

在本次实验过程中遇到了数据预处理的问题，我的处理方法是将数据csv格式导入到mysql数据库中，通过sql对需要的数据进行查询并导出。这也锻炼了我mysql的能力也加深了sql语言的熟悉程度。遇到的另一个问题是将不同图形放在同一页面中，解决方法是将不同图形放在不同的svg画布中，再根据需求调整坐标位置，最终解决了这个问题。

本次实验收获很多，通过将不可见的数据现象转化为可见的图形符号和图标，能将错综复杂、看起来无法解释和关联的数据，建立起联系和关联，发现规律和特征。