将csv数据导入MySQL 的命令

1. 先创建数据所需要的结构表

 DROP TABLE IF EXISTS `tgame`;  
CREATE TABLE tgame(  
  date varchar(20) ,  
  ms text ,  
  name varchar(50) ,  
  score int(10),  
  size int(20),  
  title varchar(150),  
  type varchar(100),  
  url varchar(60),  
  comment int(30)  
) ENGINE = InnoDB CHARACTER SET = utf8;

engine = InnoDB character set = utf8;

1. 执行导入命令

绿色可以不要，只要将数据库编码设置好之后就可以不用再指定了（alter database 库名 character set utf8）

load data local infile 'E:/testgame.csv' into table tgame character set utf8 fields terminated by ',' optionally enclosed by '"' escaped by '"' lines terminated by '\n';

可视化Echarts精简结构

Option= {

Title:{}, //标题

Tooltip:{}, //鼠标拖动显示数据

Legend:{}, //显示图例

xAxis:{},

yAxis:{},

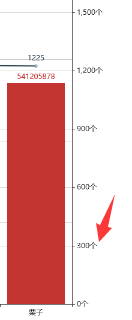
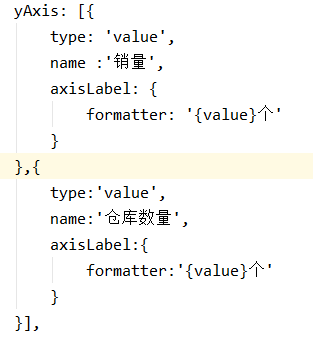
Series:{}, //数据装载

}

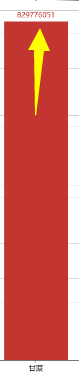
柱状图bar和折线图同理

数据类型一维数组 x y

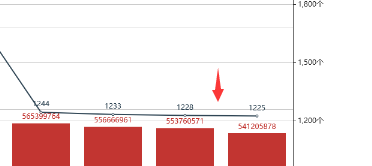
1. 使用精简结构，设置 type:’bar’
2. 需要设置x y轴标题，在xAxis里面设置name，特殊要求需要设置axisLabel:{}



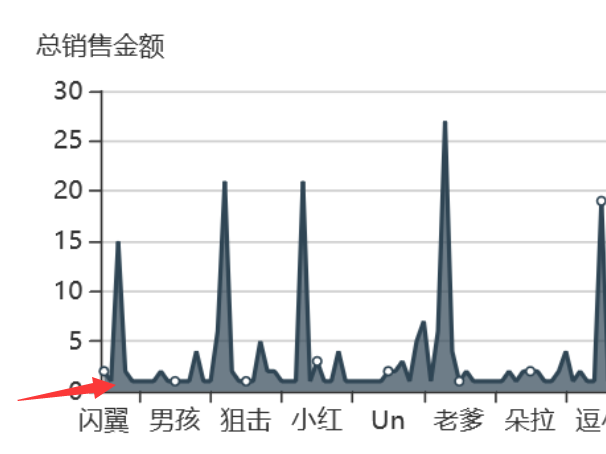
1. 需要显示值在柱子上面，在seires的data下面设置label:{ show:true, position:’top’ }

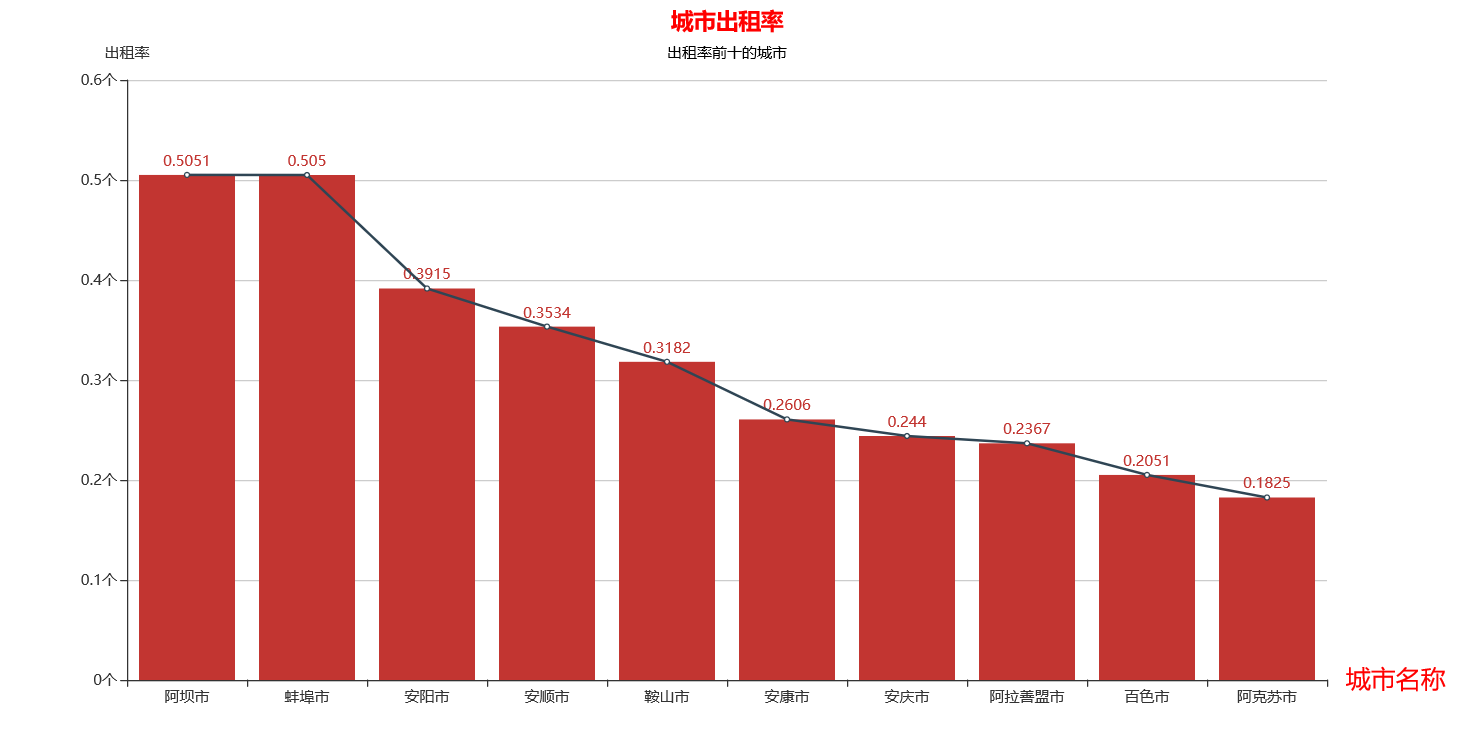


1. 双图（柱状图和折线图双拼）需要将折线图置顶的设置 yAxisIndex:1



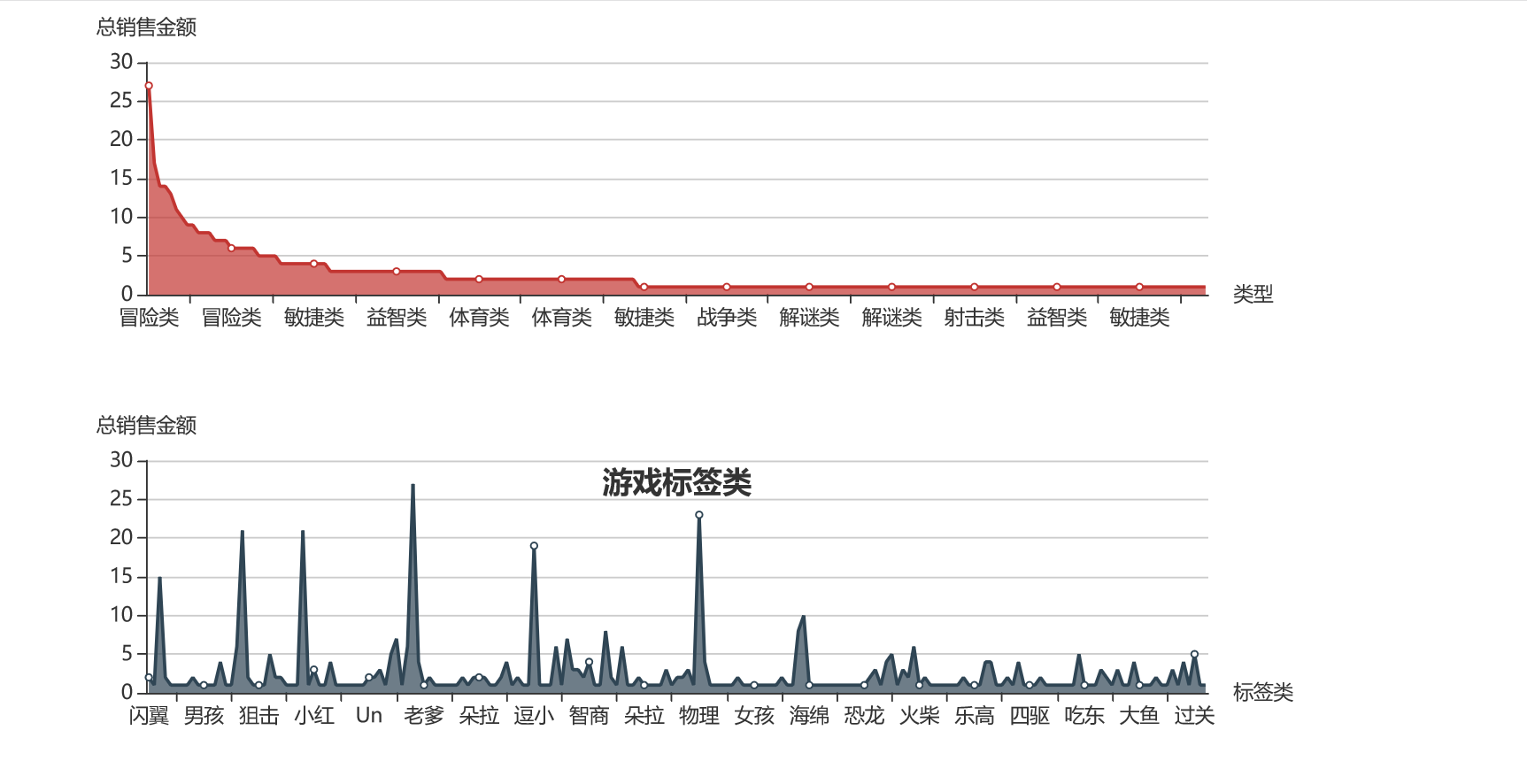
1. 添加阴影，areaStyle,区域阴影

  
console.log(eval({{ t\_na | safe}}))  
 console.log(eval({{ t\_va | safe}}))  
 op = {  
 title:[{  
 text:**'城市出租率'**,  
 textStyle:{  
 color:**'red'**,  
 fontWeight:**'bold'**, <!--默认加粗-->  
 },  
 subtext:**'出租率前十的城市'**,  
 subtextStyle:{  
 color:**'black'**,  
 },  
 x:**'center'**,  
 }],  
 legend:{},  
 xAxis:[{  
 name:**'城市名称'**,  
 data:eval({{ t\_na | safe}}),  
 nameTextStyle:{  
 color:**'red'**,  
 fontSize:20,  
 },  
 axisLabel:{  
 formatter:**'{value}市'** },  
 gridIndex:0,  
 }],  
 yAxis:[{  
 name:**'出租率'**,  
 axisLabel:{  
 formatter:**'{value}个'** },  
 gridIndex:0,  
 }],  
 series:[{  
 type:**'bar'**,  
 data:eval({{ t\_va | safe}}),  
 label:{  
 normal:{  
 show:true,  
 position:**'top'** },  
 },  
  
 },{  
 type:**'line'**,  
 data:eval({{ t\_va | safe}}),  
 xAxisIndex:0,  
 }]  
 }  
 echarts.init(document.getElementById(**'a'**)).setOption(op)

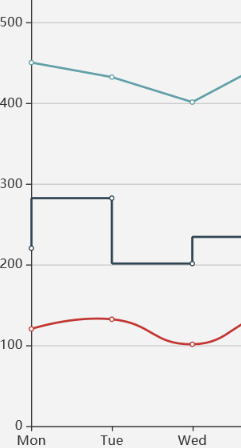


子图系统

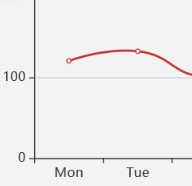
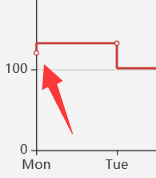
option = {  
title:[{  
 text:'游戏类型',  
 x:'center',  
},{  
 text:'游戏标签类',  
 x:'center',  
 top:'60%',  
}],  
tooltip:{  
 trigger:'axis'  
},  
legend:{},  
xAxis:[{  
 name:'类型',  
 gridIndex:0,  
 data:na,  
 },{  
 name:'标签类',  
 gridIndex:1,  
 data:na1,  
  
 }],  
yAxis:[{  
 gridIndex:0,  
 name:'总销售金额',  
},{  
 gridIndex:1,  
 name:'总销售金额',  
}],  
grid:[{  
 bottom:'60%',   
},{  
 top:'60%'  
}],  
series:[{  
 type:'line',  
 data:va,  
 areaStyle:true,  
},{  
 type:'line',  
 data:va1,  
 xAxisIndex:1,  
 yAxisIndex:1,  
 areaStyle:true, 区域阴影  
}]  
}  
var myChart = echarts.init(document.getElementById('a')).setOption(option);



1. 折线图直线（默认），平滑，方形，分别step:true，smooth:true，

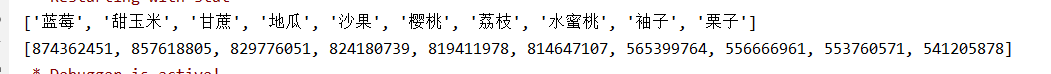


1. 折线图从0开始，设置 boundarGap:0

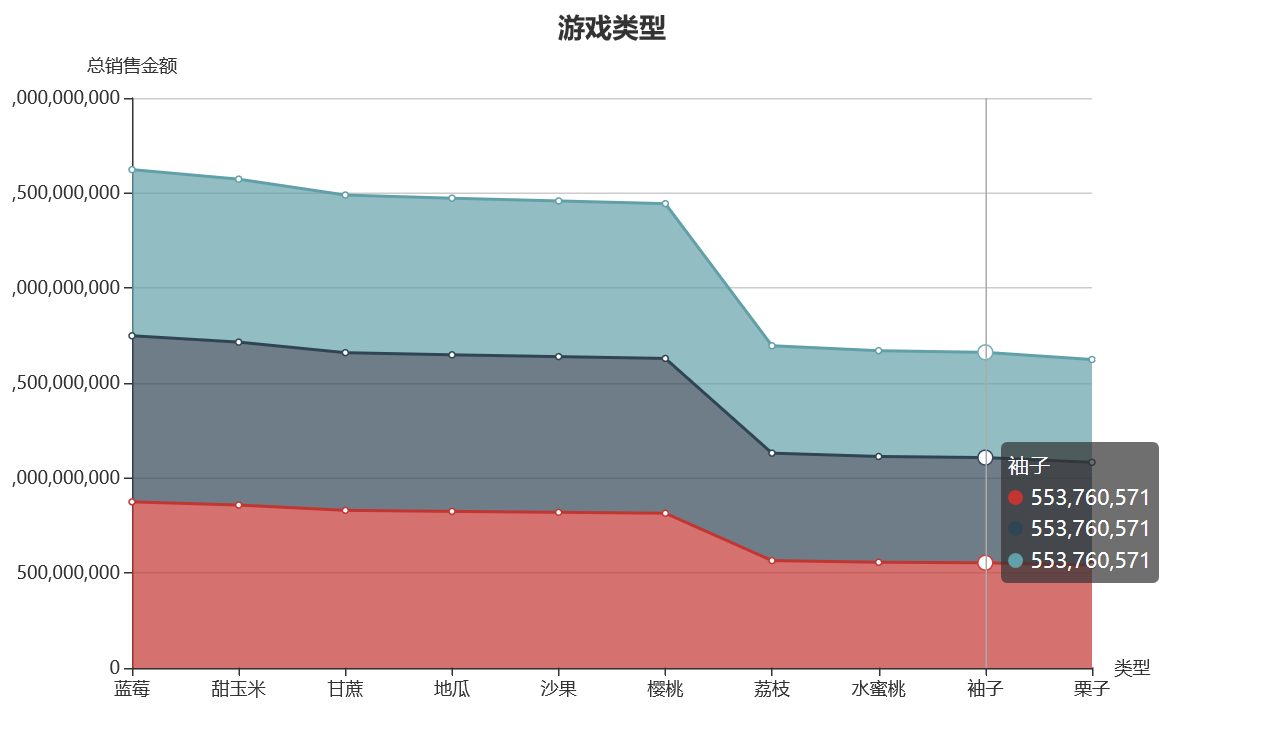
 ---> 

堆叠图

子图

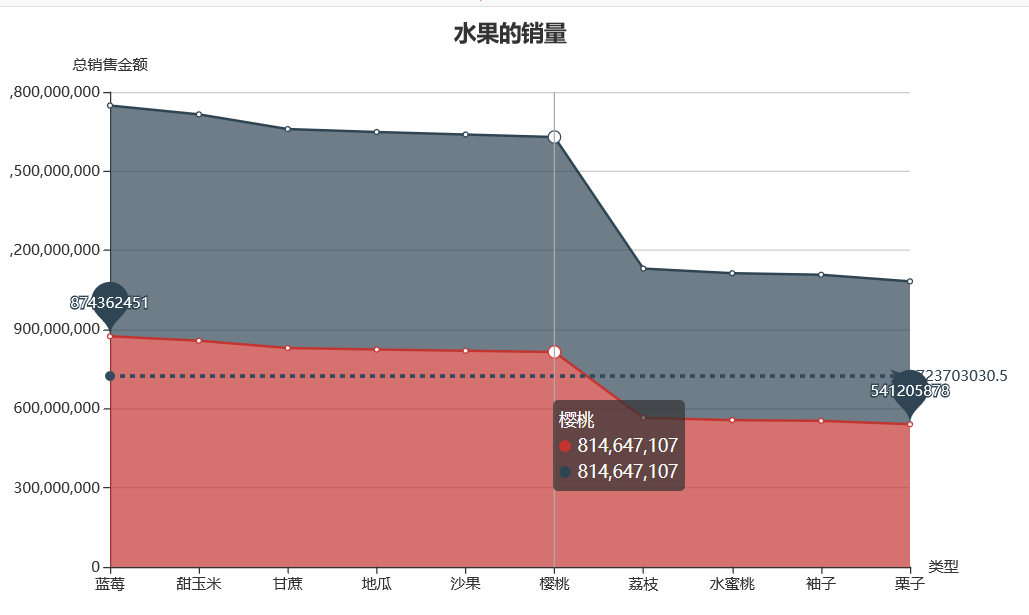


option = {  
 title:[{  
 text:'游戏类型',  
 x:'center',  
 }],  
 tooltip:{  
 trigger:'axis'  
 },  
 legend:{},  
 xAxis:[{  
 boundaryGap:false,  
 name:'类型',  
 data:na  
 }],  
 yAxis:[{  
 name:'总销售金额',  
 }],  
 series:[{  
 type:'line',  
 data:va,  
 stack:'a',  
 areaStyle:true,  
  
 },{  
 type:'line',  
 data:va,  
 stack:'a',  
 areaStyle:true,  
  
 },{  
 type:'line',  
 data:va,  
 stack:'a',  
 areaStyle:true,  
<!-- label:{-->  
<!-- normal:{-->  
<!-- show:true,-->  
<!-- position:'insideRight'-->  
<!-- }-->  
<!-- }-->  
 }]  
 }  
 var myChart = echarts.init(document.getElementById('a')).setOption(option);



option = {  
title:[{  
 text:**'游戏类型'**,  
 x:**'center'**,  
}],  
tooltip:{  
 trigger:**'axis'**},  
legend:{},  
xAxis:[{  
 boundaryGap:false,  
 name:**'类型'**,  
 data:na  
 }],  
yAxis:[{  
 name:**'总销售金额'**,  
}],  
series:[{  
 type:**'line'**,  
 data:va,  
 stack:**'a'**,  
 areaStyle:true,  
},{  
 type:**'line'**,  
 data:va,  
 stack:**'a'**,  
 areaStyle:true,

markPoint:{  
 data:[  
 {type:'max',name:'最大值'},  
 {type:'min',name:'最小值'},  
 ]  
 },  
 markLine:{  
 data:[  
 {type:'average',name:'平均值'}  
 ]  
 },  
}]  
}  
var myChart = echarts.init(document.getElementById('a')).setOption(option);



饼图

数据类型:

data:[

{value:335, name:'直接访问'},

{value:310, name:'邮件营销'},

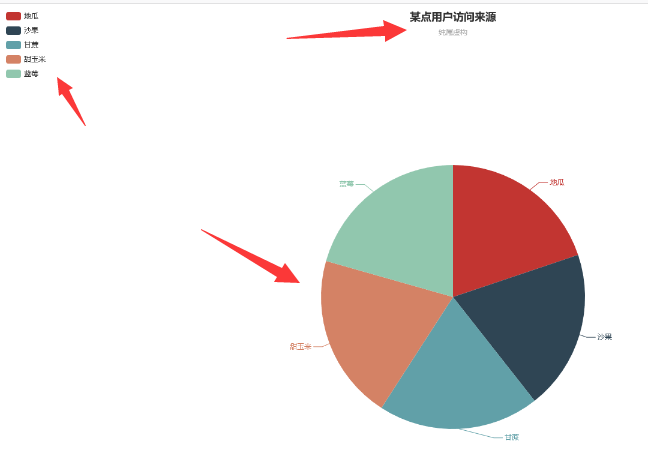
{value:234, name:'联盟广告'},

{value:135, name:'视频广告'},

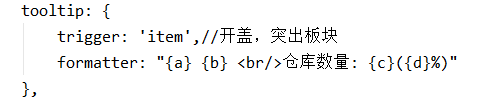
{value:1548, name:'搜索引擎'}

],

1. 使用精简结构，设置 type:’pie’，不需要xAxis和yAxis
2. 特殊设置：设置标题居中 x:’center’
3. 图例左边 left:true (legend:标签中)
4. 设置方向为锤直 orient:’vertical’ (legend:标签中)

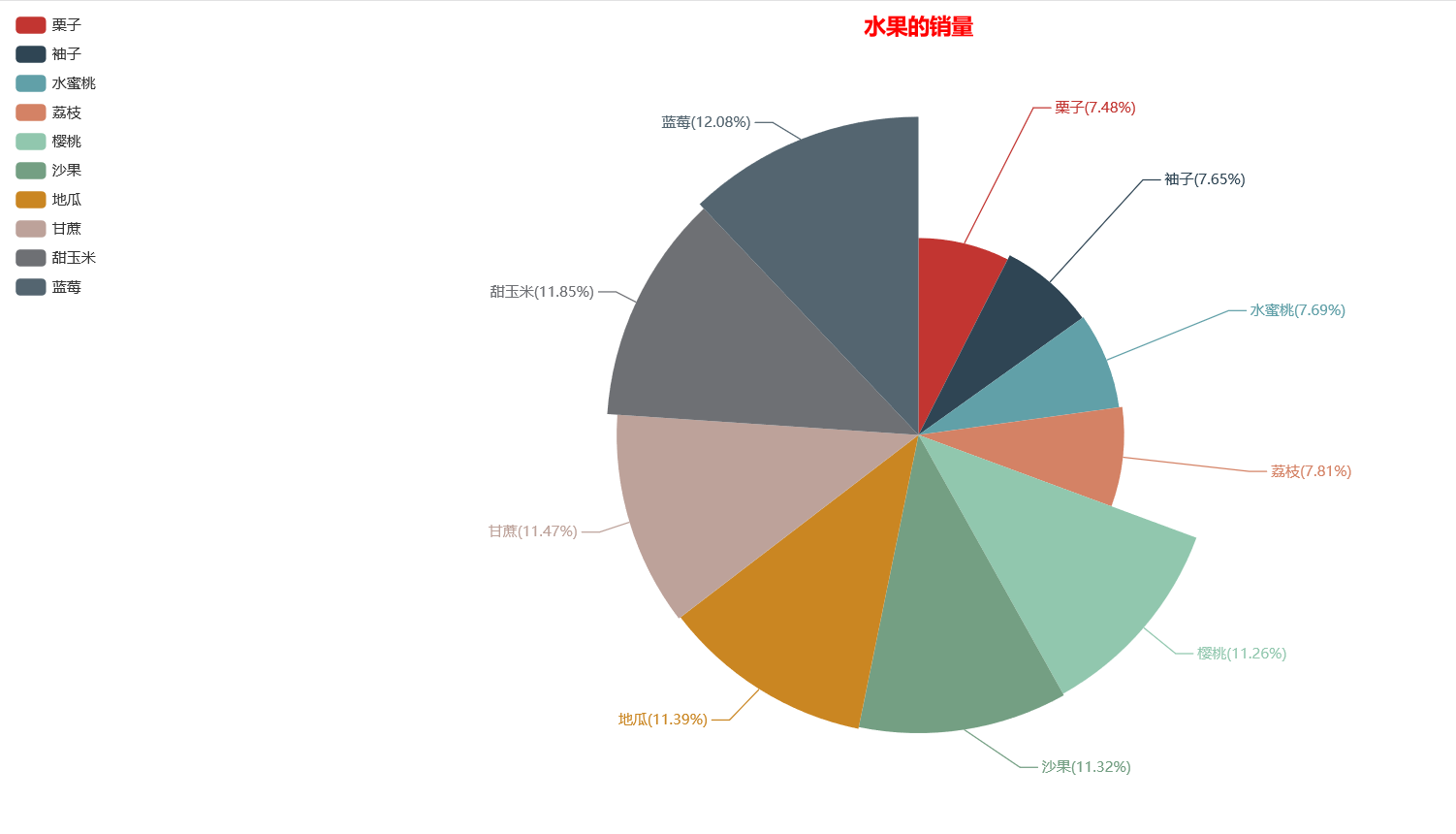


1. 需要显示每个的直在tooltip这是formatter:



1. 设置饼图的大小 radius:’50%’ 整体半径变一半
2. 设置饼图的位置 center:[‘50%’,’60%’]

option = {  
 backgroundColor: **'white'**,  
 title: [{  
 text: **'水果的销量'**,  
 x: **'center'**,  
 textStyle: {  
 color: **'red'** },  
 }],  
 tooltip: {  
 trigger: **'item'**,  
 formatter: **'({d}%)'** },  
 legend: {  
 left: **true**, 使图例右  
 orient: true 使图列竖排  
 },  
 series: {  
 type: **'pie'**,  
 data: data.sort(function(a, b){  
**return** a.value - b.value}),  
roseType: true,   
label: {  
 formatter: **'{b}({d}%)'**},  
}  
}  
var  
myChart = echarts.init(document.getElementById(**'a'**)).setOption(option);



气泡图

数据类型：二维数组 [[20,30,560,’蓝莓’],...]

1. 使用精简结构 type:’scatter’
2. 类型少只需要设置气泡（球）的大小 symbolSize:function(data){ return data[2] / 10000 } （没有normal:{}）
3. 类型多需要设置两种:

symbolSize:function(data){ //\*\*\*\*\*

return data[2] / 1000

}

label:{

emphasis:{ //\*\*\*\*\*

show:true, //\*\*\*\*\*

formatter:function(da){

return da.data[3]

}

}

}

雷达图

数据格式：3组数据

1. data: [

{ name: '销售（sales）', max: 6500},

{ name: '管理（Administration）', max: 16000},

{ name: '信息技术（Information Techology）', max: 30000},

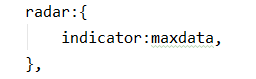
{ name: '客服（Customer Support）', max: 38000},

{ name: '研发（Development）', max: 52000},

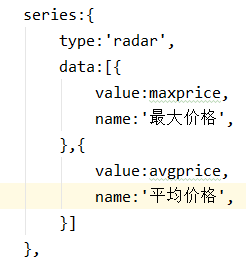
{ name: '市场（Marketing）', max: 25000}

]

1. 最大值:一维数据
2. 最小值:一维数据
3. 使用精简结构上需要添加 radar标签数据为：最大值和name值



1. 在series中添加最大值和最小值 data:[{},{}]



词云图





1. 数据类型：data: [

{ name: '销售', value: 6500},

]

1. 使用精简数据结构 type:’wordCloud’
2. 需要设置rgb，

textStyle:{

normal:{

color:function(){

return ‘rgb(’+[

Math.round(Math.random()\*200),

].join()+’)’

}

}

}

热力图

三种数据类型：data : [[0,0,5]...]，x y

1. 需要设置x y轴的数据，将x y一维数据赋值给(xAxis) data
2. 热力图需要设置visualMap，里面设置大小值，还有calculable:true（拖拽）
3. Series里面需要显示每个热力的值的话，需要设置label : { show : true }



**def** heatMap():  
 cur = mysqlDB().lsw()  
 sql = **"""  
 select   
 month(`发货日期`)as yue,day(`发货日期`)as tian,sum(`小计`)as number  
 from `按季度汇总销售额`  
 where YEAR(`发货日期`) = '1997'  
 group by yue,tian  
 """** cur.execute(sql)  
 data = list(cur.fetchall())  
 temp = []  
 **for** i **in** data:  
 t = []  
 t.append(i[0])  
 t.append(i[1])  
 t.append(int(i[2]))  
 temp.append(t)  
 **return** temp

<!DOCTYPE **html**>  
<**html lang="en"**>  
<**head**>  
 <**meta charset="UTF-8"**>  
 <**title**>Title</**title**>  
 <**script src="{{ url\_for ('static',filename='echarts.js')}}"**></**script**>  
  
</**head**>  
<**body**>  
<**div id="a" style="**height:650px;width:1200px**"**></**div**>  
<**script**>

var data= eval({{h | safe}})  
 console.log(data)  
 var month = []  
 var day = []  
 **for**(var i=0;i<13;i++){  
 month.push(i)  
 }  
 **for**(var j=0;j<32;j++){  
 day.push(j)  
 }  
 console.log(month)  
 console.log(day)  
 option = {  
 title:{},  
 legend:{},  
 tooltip:{},  
 xAxis:{  
 name:**'每月'**,  
<!-- boundaryGap: false,-->  
 data:month,  
 },  
 yAxis:{  
 name:**'每天'**,  
 data:day  
 },  
 visualMap:{  
 max:5000,  
 min:0,  
 calculable:true  
 },  
 series:{  
 type:**'heatmap'**,  
 data:data,  
  
 label:{  
 show:true  
 }  
 },  
 }  
 echarts.init(document.getElementById(**'a'**)).setOption(option)

</**script**>

</**body**>  
</**html**>

中国地图

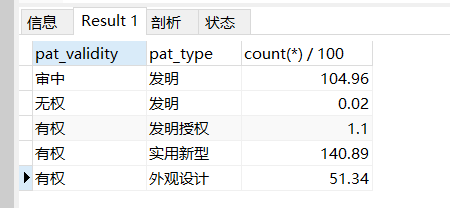


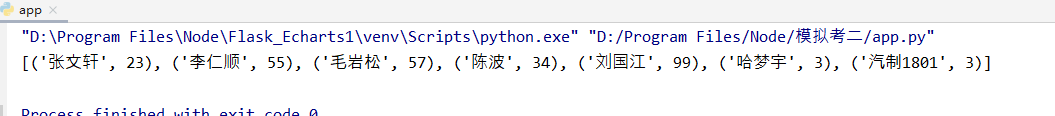
**def** task1\_china():  
 cur = mysqlDB().shop()  
 sql= **"select left(apply\_address,2)as address,count(\*)as number from test1 where apply\_address like '%省%' group by address order by number desc"** cur.execute(sql)  
 data = cur.fetchall()  
  
 name , value = [],[]  
 **for** i **in** data:  
 **if** i[0] == **'黑龙'**:  
 name.append(**'黑龙江'**)  
 **elif** i[0] == **'内蒙'**:  
 name.append(**'内蒙古'**)  
 **elif** i[0] != **'黑龙' and** i[0] != **'内蒙'**:  
 name.append(i[0])  
 value.append(int(i[1]))  
 temp = dict(zip(name,value))  
 **return** temp

<div id=**"a"** style=**"height:650px;width:1500px"**></div>  
<script>  
 var keyname = eval({{ w\_temp | safe }})  
 var data = []  
 **for**(var i **in** keyname){  
 data.push({  
 name:i,  
 value:keyname[i]  
 })  
 }  
 console.log(data)  
 option = {  
 titlt:{},  
 tooltip:{},  
 series:{  
 type:**'wordCloud'**,  
 data:data,  
 textStyle:{  
 normal:{  
 color:function(){  
 **return 'rgb('**+[  
 Math.round(Math.random()\*200),  
 Math.round(Math.random()\*200),  
 Math.round(Math.random()\*200),  
 ].join()+**')'** }  
 }  
 }  
 },  
  
 }  
 echarts.init(document.getElementById(**'a'**)).setOption(option)  
</script>

关系图

数据的依赖关系





**def** over(shopcur):  
 sql = **'select 班级,名字,期末成绩 from `行为数据` limit 3000'** shopcur.execute(sql)  
 datas = shopcur.fetchall()  
 data1, data2, data3, links = [], [], [], []  
 **for** i **in** datas:  
 t, t1 = [], []  
 t.append(i[0])  
 t.append(i[1])  
 links.append(t)  
  
 data1.append(i[0])  
 data2.append(i[1])  
 data3.append(i[2])  
 *#* a = dict(zip(data2, data3))  
 a1 = list(a.keys())  
 a2 = list(a.values())  
 a = list(zip(a1,a2))  
  
 b = dict(zip(data1,data3))  
 b1 = list(b.keys())  
 b2 = list(b.values())  
 b = list(zip(b1,b2))  
 data = a + b  
  
  
 **return** data, links

<**script src="{{url\_for('static',filename='echarts.js')}}"**></**script**>

<**div id="a" style="**height:1600px;width:3200px**"**></**div**>

<**script**>

var e = eval({{ eee |tojson}})  
 var d = eval({{ d |tojson}})  
 var data = []  
 var ty1 = []  
 **for**(var i **in** d){  
 data.push({  
 name:d[i][0],  
 symbolSize:d[i][1],  
 categories:1,  
 })  
 }  
 **for**(var j **in** e){  
 ty1.push({  
 source:e[j][0],  
 target:e[j][1],  
 })  
 }  
 console.log(data)  
 console.log(ty1)

option = {  
 title:{},  
 tooltip:{},  
 legend:{},  
 series:{  
 type:**'graph'**,  
 layout:**'force'**,  
 symbolSize:10,  
 edgeSymbol:[**'s'**,**'arrow'**],  
 draggable:true,  
 label:{  
 normal:{show:true}  
 },  
 force:{  
 repulsion:2000 *#设置线的长度* },  
 lineStyle:{  
 normal:{  
 width:2,  
 color:**'red'** }  
 },  
 data:ty,  
 links:re,  
 categories:[{ *#可以不要* name:**'aa'** },{  
 name:**'bb'** },{  
 name:**'cc'** }]  
  
 }  
}  
echarts.init(document.getElementById(**'main'**)).setOption(option)

</**script**>