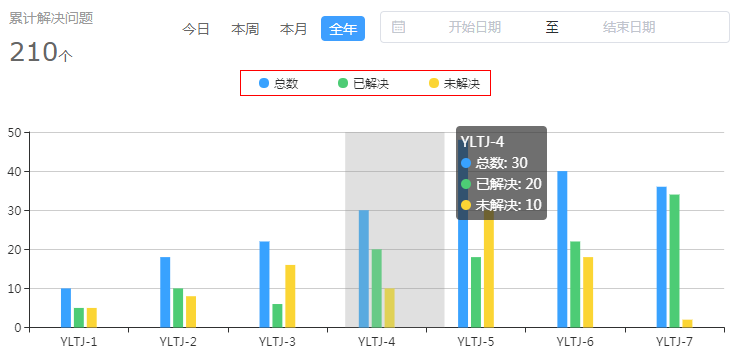
查看echarts版本：console.log(echarts.version)

# 一、echarts设置图标图例legend为圆，长方形，扇形等



legend: {

data: ["总数", "已解决", "未解决"],

icon: "circle",   //  这个字段控制形状  类型包括 circle，rect ，roundRect，triangle，diamond，pin，arrow，none

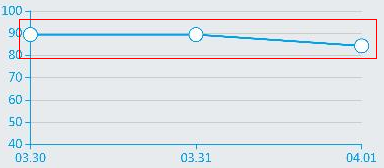
itemWidth: 10,  // 设置宽度

itemHeight: 10, // 设置高度

itemGap: 40 // 设置间距

}

# 二、设置图表标记series为圆，长方形，扇形等



series: [

{

name: ‘综合评分‘,

type: ‘line‘,

stack: ‘总量‘,

data: [<volist name="list" id="vv">{$vv.score},</volist>],

symbol:"circle", // 这个字段控制形状  类型包括 circle，rect ，roundRect，triangle，diamond，pin，arrow，none

symbolSize: 14, // 图表的点的大小

itemStyle: {normal: {color: ‘#03a1ea‘}}

}

]

# 三、echarts-formatter 数据格式化小结

formatter格式化方法的参数说明：(下面一段话引用自https://www.cnblogs.com/ys-wuhan/p/6149265.html)

{string}，模板（Template），其变量为：

{a} | {a0}

{b} | {b0}

{c} | {c0}

{d} | {d0} （部分图表类型无此项）

多值下则存在多套{a1}, {b1}, {c1}, {d1}, {a2}, {b2}, {c2}, {d2}, ...

其中变量a、b、c在不同图表类型下代表数据含义为：

折线（区域）图、柱状（条形）图: a（系列名称），b（类目值），c（数值）, d（无）

散点图（气泡）图 : a（系列名称），b（数据名称），c（数值数组）, d（无）

饼图、雷达图 : a（系列名称），b（数据项名称），c（数值）, d（百分比）

弦图 : a（系列名称），b（项1名称），c（项1-项2值），d（项2名称)， e(项2-项1值)

力导向图 :

节点 : a（类目名称），b（节点名称），c（节点值）

边 : a（系列名称），b（源名称-目标名称），c（边权重）， d（如果为true的话则数据来源是边）

{Function}，传递参数列表如下：

<Array> params : 数组内容同模板变量，[[a, b, c, d], [a1, b1, c1, d1], ...]

<String> ticket : 异步回调标识

<Function> callback : 异步回调，回调时需要两个参数，第一个为前面提到的ticket，第二个为填充内容html

## 一、tooltip的数据格式化

tooltip : {

trigger: 'axis',

formatter:function(a,b,c){

return a+'<br/>'+b+c;

}

},

## 二、坐标轴上的数据格式化

1、Y轴

yAxis : [{

type : 'value',

axisLabel : {

formatter: function(value){

return value+"单位";}

}

}],

2、X轴

xAxis : [{

type : 'category',

axisLabel : {

formatter: function(value){

return value+"单位";}

}

}],

## 三、series的数据格式化

series: [

{

name: '',

type: 'pie',

label: {

normal: {

show: false,

position: 'center',

formatter:function(a,b,c){

// 数据据格式化：33,454

a.data.value = a.data.value.toString().replace(/(\d)(?=(?:\d{3})+$)/g, '$1,');

}

}

},

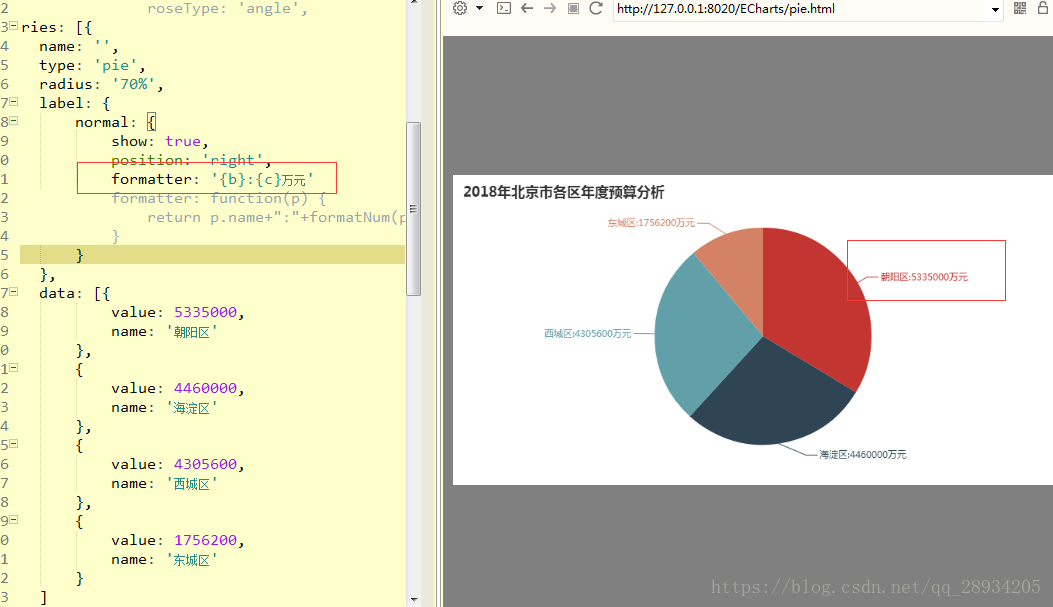
data:projectPercent

}

]

# 四、echarts 让数据千分位显示

原来的效果



千分位展示 效果



代码：

formatter: function(p) {

return p.name+":"+formatNum(p.value)+"万元";

}或

formatter:function(a,b,c){

a.data.value = formatNum(a.data.value);

}

function formatNum(strNum) {

if(strNum.length <= 3) {

return strNum;

}

if(!/^(\+|-)?(\d+)(\.\d+)?$/.test(strNum)) {

return strNum;

}

var a = RegExp.$1,

b = RegExp.$2,

c = RegExp.$3;

var re = new RegExp();

re.compile("(\\d)(\\d{3})(,|$)");

while(re.test(b)) {

b = b.replace(re, "$1,$2$3");

}

return a + "" + b + "" + c;

}

# 五、y轴只显示整数

yAxis: {  
 type: 'value',  
 minInterval: 1, // 最小递增值  
 name: "单位：人"

}

# 六、x轴文本内容太长的几种解决方案

地址：<https://www.jianshu.com/p/b452cbe9be0e>

Echarts 标签中文本内容太长的时候怎么办 ?

* **1对文本进行倾斜**  
  在xAxis.axisLabe中修改rotate的值

xAxis: {

data: ["衬衫11111","羊毛二二","雪纺衫111","裤子111","高跟鞋11","袜子111"],//x轴中的数据

name:"123",//坐标轴名称。

nameLocation:'end',//坐标轴名称显示位置。

axisLabel : {//坐标轴刻度标签的相关设置。

interval:0,

rotate:"45"

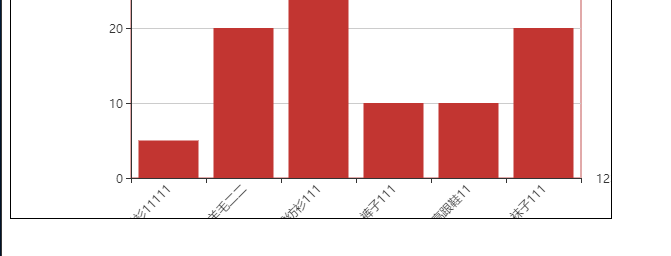
}

},

interval

坐标轴刻度标签的显示间隔(在类目轴中有效哦)，默认会采用标签不重叠的方式显示标签（也就是默认会将部分文字显示不全）  
可以设置为0强制显示所有标签，如果设置为1，表示隔一个标签显示一个标签，如果为3，表示隔3个标签显示一个标签，以此类推

一开始我没设置 因为标签文本过长的原因他就自动不显示全部



被遮挡住就让grid 组件离容器向上移动 把grid中的bottom的值调大一些

grid:{//直角坐标系内绘图网格

show:true,//是否显示直角坐标系网格。[ default: false ]

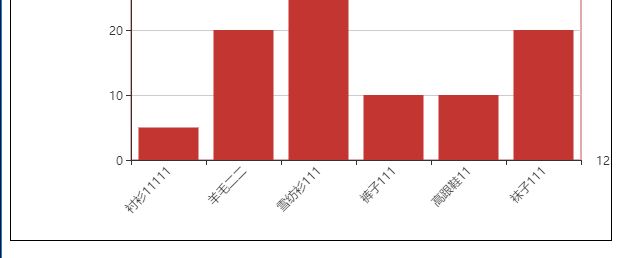
left:"20%",//grid 组件离容器左侧的距离。

right:"30px",

borderColor:"#c45455",//网格的边框颜色

bottom:"20%" //

},



* **2.换行显示**  
  在xAxis.axisLabel中 使用formatter回调函数实现换行

axisLabel : {//坐标轴刻度标签的相关设置。

formatter : function(params){

var newParamsName = "";// 最终拼接成的字符串

var paramsNameNumber = params.length;// 实际标签的个数

var provideNumber = 4;// 每行能显示的字的个数

var rowNumber = Math.ceil(paramsNameNumber / provideNumber);// 换行的话，需要显示几行，向上取整

/\*\*

\* 判断标签的个数是否大于规定的个数， 如果大于，则进行换行处理 如果不大于，即等于或小于，就返回原标签

\*/

// 条件等同于rowNumber>1

if (paramsNameNumber > provideNumber) {

/\*\* 循环每一行,p表示行 \*/

for (var p = 0; p < rowNumber; p++) {

var tempStr = "";// 表示每一次截取的字符串

var start = p \* provideNumber;// 开始截取的位置

var end = start + provideNumber;// 结束截取的位置

// 此处特殊处理最后一行的索引值

if (p == rowNumber - 1) {

// 最后一次不换行

tempStr = params.substring(start, paramsNameNumber);

} else {

// 每一次拼接字符串并换行

tempStr = params.substring(start, end) + "\n";

}

newParamsName += tempStr;// 最终拼成的字符串

}

} else {

// 将旧标签的值赋给新标签

newParamsName = params;

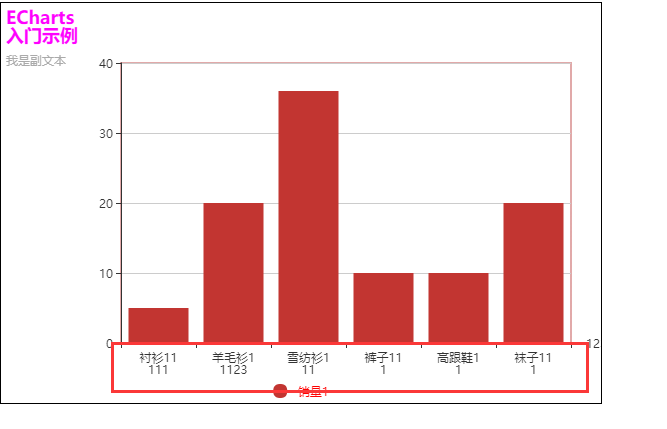
}

//将最终的字符串返回

return newParamsName

}

}



* **3.文字竖直显示**  
  同样和换行一个道理,只是这个是单个文字换行  
  在xAxis.axisLabel中 使用formatter回调函数实现换行

axisLabel: {

interval: 0,

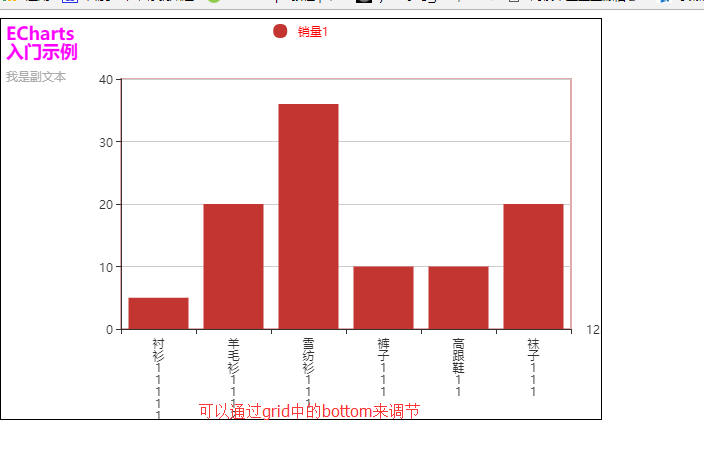
formatter:function(value)

{

return value.split("").join("\n");

}

}



* **4.隔一个换行**  
  在xAxis.axisLabel中 使用formatter回调函数实现换行

axisLabel : {//坐标轴刻度标签的相关设置。

clickable:true,//并给图表添加单击事件 根据返回值判断点击的是哪里

interval : 0,

formatter : function(params,index){

if (index % 2 != 0) {

return '\n\n' + params;

}

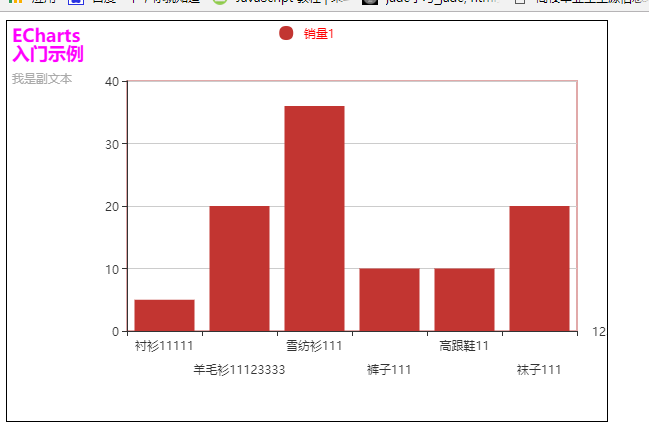
else {

return params;

}

}

}



# 七、添加暂不数据提示

**function** noDataLoadingOption(echarts, top, right) {  
 **var** option = echarts.getOption();  
 **var** series = option.series;  
 **var** flag = 0; // 0：无数据；1：有数据  
  
 **for** (**var** i = 0; i < series.length; i++) {  
 **var** sData = series[i].data;  
 **if**(sData.length > 0){  
 showData(echarts);  
 flag = 1;  
 }**else** { // 给指标添加默认值  
 **var** data = [];  
 **var** xylData = [];  
 **var** type = series[i].type;  
 **if**(type == "bar" || type == "line"){  
 xylData = **typeof**(option.xAxis[0].data) != "undefined" ? option.xAxis[0].data : option.yAxis[0].data;  
 }**else if**(type == "pie" && **typeof**(option.legend[0]) != "undefined"){  
 xylData = option.legend[0].data;  
 }  
 **for** (**var** j = 0; j < xylData.length; j++) {  
 data[j] = 0;  
 }  
 option.series[i].data = data;  
 echarts.setOption(option);  
 }  
 }  
  
 **if**(flag == 0){  
 **for** (**var** i = 0; i < series.length; i++) {  
 option.series[i].data = [];  
 echarts.setOption(option);  
 }  
 hideData(echarts, top, right);  
 }  
}  
  
**function** hideData(echarts, top, right){  
 **var** span = $(echarts.getDom()).find("div").first().find("span");  
 **if**(span.is("span")){  
 span.css("display", "block");  
 }**else**{  
 $(echarts.getDom()).find("div").first().append(getCssSpan(top, right));  
 }  
  
 **var** canvas = $(echarts.getDom()).find("div").first().find("canvas");  
 **if**(canvas.is("canvas")){  
 canvas.css("display", "none");  
 }  
}  
  
**function** showData(echarts){  
 **var** span = $(echarts.getDom()).find("div").first().find("span");  
 **if**(span.is("span")){  
 span.css("display", "none");  
 }  
  
 **var** canvas = $(echarts.getDom()).find("div").first().find("canvas");  
 **if**(canvas.is("canvas")){  
 canvas.css("display", "block");  
 }  
}  
  
**function** getCssSpan(top, right) {  
 **var** top = **typeof**(top) == "undefined" ? 50 : top;  
 **var** right = **typeof**(right) == "undefined" ? 50 : right;  
  
 **var** cssSpan = "<span style='" +  
 "color: #488728;" +  
 "font-size: 15px;" +  
 "position: absolute;" +  
 "top:"+top+"%;" +  
 "right:"+right+"%;" +  
 "transform: translate(65%, -50%);'>" +  
 "暂无数据！" +  
 "</span>";  
  
 **return** cssSpan;  
}

调用noDataLoadingOption(myecharts, 50, 50);