

# 可视化实验一报告

201900161140 张文浩

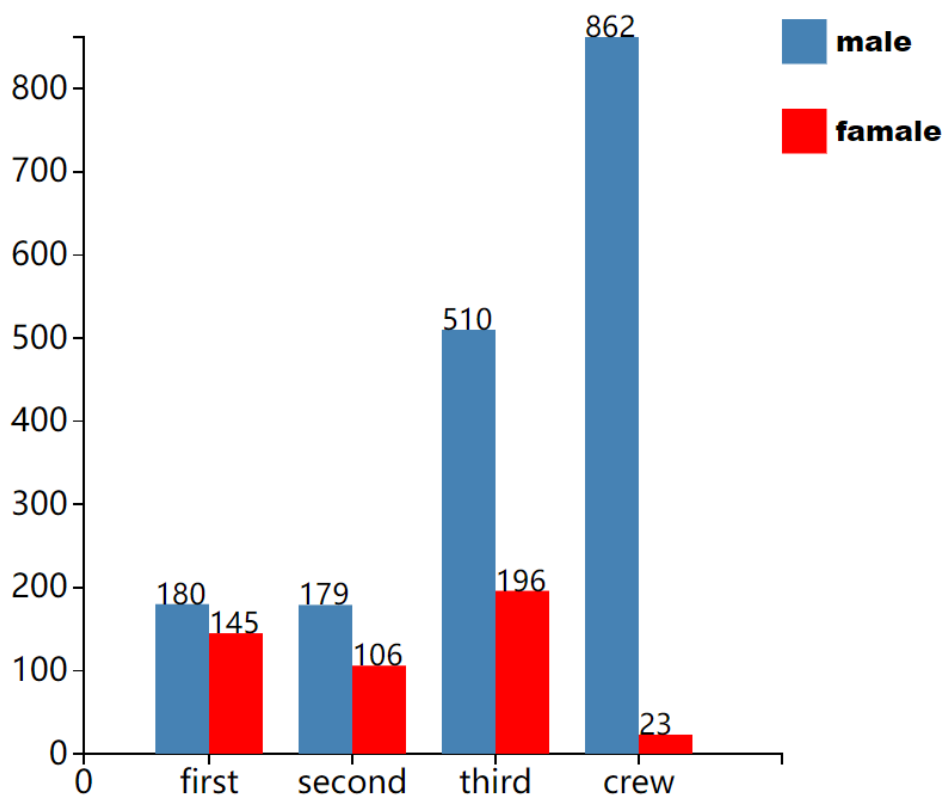
实验完成时间：9.17

软件环境：vscode

## 一、可视化效果

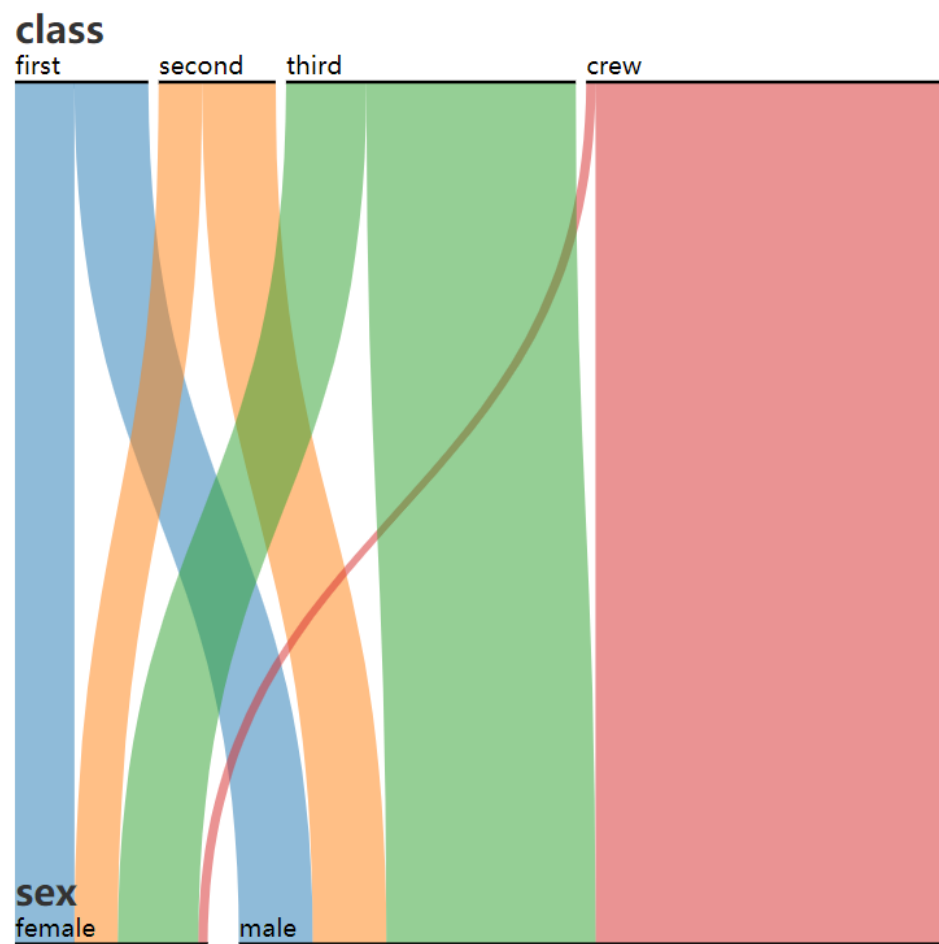
1.bar chart 柱状图

### 1.bar chart



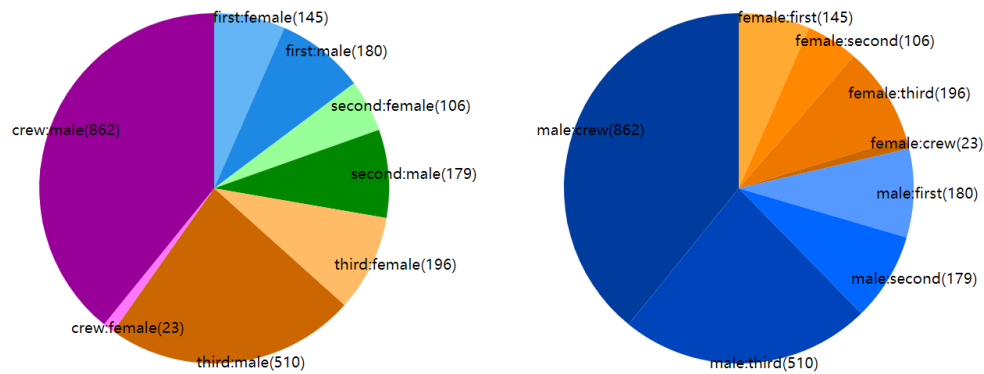
2.parallel set 平行坐标系分类图

## 2.parallel set



### 3. pie plot 饼图

#### 3.pie



## 二、实现方法

### 1.bar chart 柱状图

柱状图使用的库是

```
<script src="http://d3js.org/d3.v3.min.js" charset="utf-8"></script>
```

关键步骤如下：

设置画布

```
var width = 800
var height = 450
var svg1 = d3.select('body')
    .append('svg')
    .attr('width',width)
    .attr('height',height)
```

代入数据集

```
var dataset1 = [180,179,510,862]//柱形图数据集
var dataset2 = [145,106,196,23]
var maxx = Math.max(d3.max(dataset2),d3.max(dataset1))
```

设置“柱子”的比例尺、x 轴 y 轴坐标的比例尺，定义 x 轴 y 轴

```
//柱状图比例尺
var linear = d3.scale.linear().domain([0, maxx]).range([0,400
])

//x 轴的比例尺
var xScale = d3.scale.ordinal()
    .domain(['0','first','second','third','crew'])
    .range([60,130,210,290,370,450])
//y 轴的比例尺
var yScale = d3.scale.linear()
    .domain([0,maxx])
    .range([400, 0]);
//定义 x 轴
var xAxis = d3.svg.axis()
    .scale(xScale)
    .orient("bottom")
//定义 y 轴
var yAxis = d3.svg.axis()
    .scale(yScale)
    .orient("left");
//添加 x 轴
svg1.append("g")
    .attr("class","axis")
    .attr("transform","translate(" + 0 + "," + (height-
30) + ")")
    .call(xAxis);
//添加 y 轴
svg1.append("g")
    .attr("class","axis")
    .attr("transform","translate(" + 60 + "," + 20 + ")")
    .call(yAxis);
```

添加“柱子”，因为我想讲 male 与 female 用两种不同的颜色表示，我的方法是构

造两个 rec 分别画 male 和 female 的“柱子”

```
var rectWidth =30
var rec1 = svg1.selectAll("rect1")
    .data(dataset1)
    .enter()//选中只有数据没有元素的部分
    .append('rect')//由于没有元素所以要添加元素
    .attr('fill','steelblue')//设置颜色
    .attr('x',(d,i)=>i*(rectWidth + 50)+100)//设置
横坐标,+3 是设置柱形图之间的间隙
```

```

        .attr('y',(d)=>height-linear(d)-30)//反转
        .attr('width',rectWidth)
        .attr('height',(d)=>linear(d))
    var rec2 = svg1.selectAll("rect2")
        .data(dataset2)
        .enter();//选中只有数据没有元素的部分
        .append('rect')//由于没有元素所以要添加元素
        .attr('fill','red')//设置颜色
        .attr('x',(d,i)=>i*(rectWidth + 50)+130)//设置
横坐标,+3 是设置柱形图之间的间隙
        .attr('y',(d)=>height-linear(d)-30)//反转
        .attr('width',rectWidth)
        .attr('height',(d)=>linear(d))

```

最后加上必要的文字注释，代码见源代码附件

## 2.parallel set 平行坐标系分类图

第二个平行坐标轴的图使用的是网上的 d3.parsets.js 和 d3.parsets.css 库，然后自己实现 html 文件中的内容。d3.parsets.js 中封装了一些绘制平行坐标轴的函数，比较方便美观。但由于我再网上没有找到相关的函数接口的使用方式说明，只好自己去读 d3.parsets.js 中的函数，尝试着使用，下面是具体步骤：

第一步还是绘制画布并构建数据集，为方便后面函数调用，这里构造的数据集是字典的形式，而非第一个柱状图的数组形式，不做赘述。

关键步骤就是用 d3.parsets()函数，调节合适的参数，绑定数据集，绘制图像

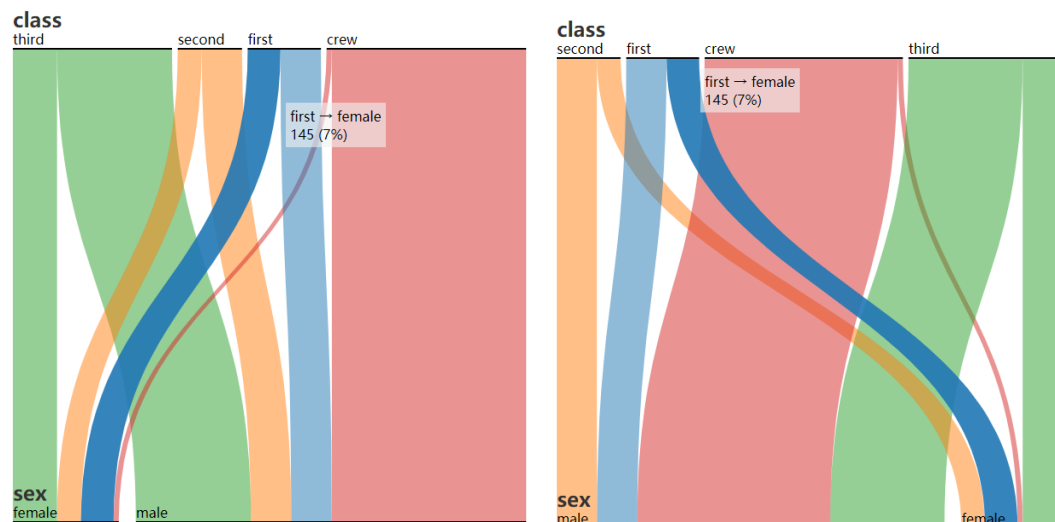
```

var chart = d3.parsets()
    .width(600)
    .tension(0.5)
    //.dimensions(["class","sex"])
    svg2.datum(dataset3).call(chart)

```

可以通过调整函数的参数调整线条的韧性，同时实现了鼠标的交互操作，例如将鼠标移动到任意线条上，即可显示这个数据集的大小和百分比信息，还可以随意

调节图像的形状每个数据集的位置，如下图所示



### 3. pie plot 饼图

饼图我是用的是 v4 版本的 d3

```
<script src="http://d3js.org/d3.v4.js" charset="utf-8"></script>
```

我的想法是画两张饼图，一张基于 sex 分类，另一张基于 class 分类

两张图的共同的参数和函数

```
//声明 margin, 半径的大小, 颜色
var margin = {top: 20, right: 20, bottom: 20, left: 20},
width = 600 - margin.right - margin.left,
height = 600 - margin.top - margin.bottom,
radius = width/3;
```

定义弧度生成器、饼图生成器、以及产生动画的函数

```
//生成弧度生成器-src generator
var arc = d3.arc()
    .outerRadius( radius)
    .innerRadius(0);

var labelArc = d3.arc()
    .outerRadius(radius + 50)
    .innerRadius(radius - 50)

//声明饼图生成器
var pie = d3.pie()
    .sort(null)
```

```

        .value(function(d) {
            //console.log(d[1]);
            return d[1]; });
    function pieTween(b){
        b.innerRadius = 0;
        var i = d3.interpolate({startAngle:0,endAngle:0},b);
        return function(t) {return arc(i(t));};
    }

```

定义颜色

```

    var color3 = d3.scaleOrdinal()
        .range(["#64B5F6", "#1E88E5", "#99FF99", "#008800", "#FFBB66",
            "#CC6600", "#FF77FF", "#990099"]);
    var color4 = d3.scaleOrdinal()
        .range(["#FFAA33", "#FF8800", "#EE7700", "#CC6600", "#5599FF",
            "#0066FF", "#0044BB", "#003C9D"]);

```

定义画布，装入数据，画饼图、添加文字，下面是画第一张饼图的实现，第二张同理

```

//定义画布 define svg
var svg3 = d3.select("body")
    .append("svg")
    .attr("width",width)
    .attr("height",height)
    .append("g")
    .attr("transform","translate("+ width/2 +"," + height/2 +
        ")");

//装入数据
var data3 = [['first:female',145],['first:male',180],['second:female',106],['second:male',179],['third:female',196],['third:male',510],['crew:female',23],['crew:male',862]];
//append g elements (arc) 相当于给 g 元素添加类属性
var g3 = svg3.selectAll(".arc")
    .data(pie(data3))
    .enter().append("g")
    .attr("class","arc");

//给 arc 添加通道 path-相当于填充颜色并且给与动作，pieTween 函数就是产生动作的函数
g3.append("path")
    .attr("d",arc)
    .style("fill",function(d) {
        return color3(d.data[0]);
    });

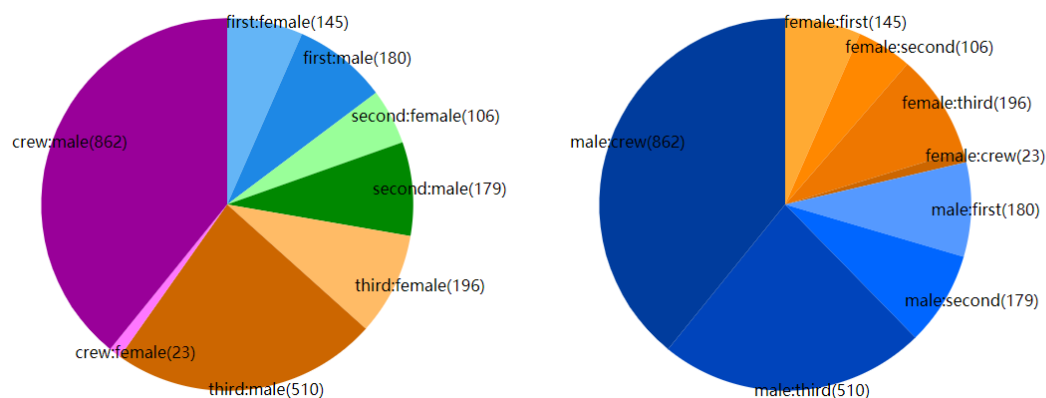
```

```

    })
    .transition()
    .ease(d3.easeLinear)
    .duration(2000)
    .attrTween("d",pieTween);
    //添加文字- 以圆为中心进行位置的寻找并且显示文字
    g3.append("text")
    .transition()
    .ease(d3.easeLinear)
    .duration(2000)
    .attr("transform", function(d) { return "translate(" + labelArc.centroid(d) + ")"; })
    .attr("dx",-40)
    .attr("dy",".35em")
    .text(function(d){
    console.log(d);
    return d.data[0]+"("+d.data[1]+")";});

```

最后效果



这张图的颜色设定比较重要，便于区分不同的类别，比如第一张图是基于 class 分类的，所以每个 class 有个及色调，再根据 sex 区分，male 的总是比 female 的颜色深一些。第二张图是先基于 sex 分类，male 以蓝色为主基调，female 以橙色为主基调。



### 三、实验总结

本次实验我通过绘制三种不同的图像对数据进行了可视化, 在过程中遇到了很多问题并尝试解决, 从中学到了很多。我清楚地认识到了每种不同的图像能够表达数据的不同的信息:

第一张图柱状图就可以很好地表达每种 class 中 sex 的数值大小; 第二张 parallel set 比较美观, 而且即能表达 class 中 sex 的分布情况, 也能表达 sex 中 class 的分布情况。同时实现了很好地交互效果, 可以根据需要进行美化。并能优美地展示更多的数据; 第三个饼状图, 第一张饼状图表达了 class 本身的分布情况和每种 class 内 sex 的分布情况。第二张饼状图表达了 sex 本身的分布情况和每种 sex 内 class 的分布情况, 在绘制饼状图的时候我也感受到了颜色选择的正确性, 合适的颜色选择可以让数据可视化更加直观。

同时, 本次是二维数据, 如果换做是维度更高的数据, 这三张图中我认为 parallel set 能够更加从容的处理高维数据, 当然, 还有更多更好的表达方式待我去学习。