

课程内容

Canvas、D3、threeJs、antV、Echarts

何为数据可视化

把数据经过一定转换之后变成图形显示的操作

数据类型

- 定量数据

- 连续性变量
- 离散变量

- 定性数据

- 文本描述
- 有序变量和无序变量

数据	数据类型	说明
11010、10086 、10000	定量数据/离散数值	取值来自于有限或可数的集合
篮球、足球、排球	定性数据/无序分类	数据没无序且不同数据无关联
及格、不及格、良好、优秀	定性数据/有序分类	数据有序且不同数据有比较关系
100m、200m、300m	定量数据/连续性数据	数据可以是实数域的任意值
拉勾教育	文本数据	任意格式文本、视情况分类
Jan. 1 1970, 00:00:00 am	日期/时间	特定的日期或时间

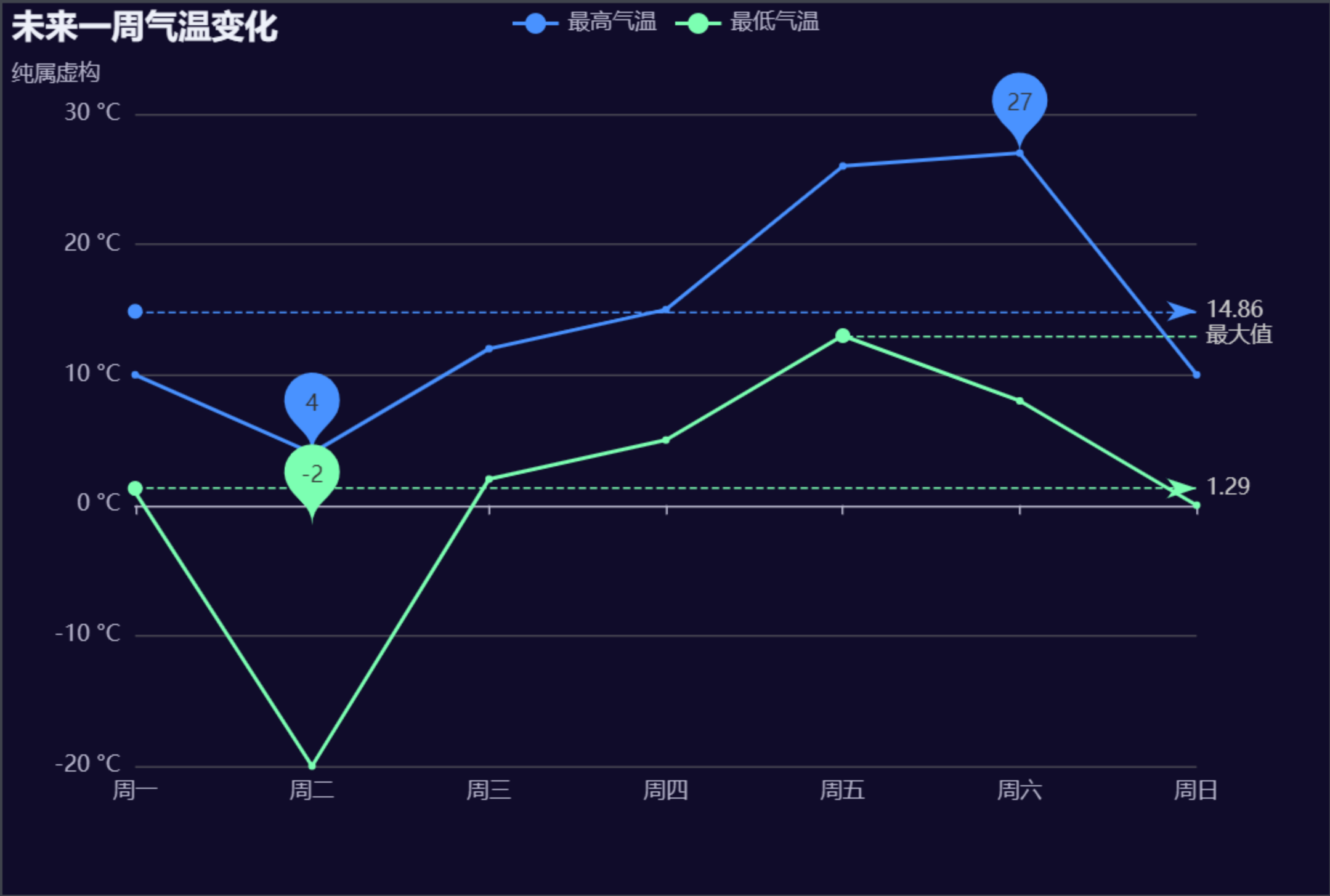
转换图形



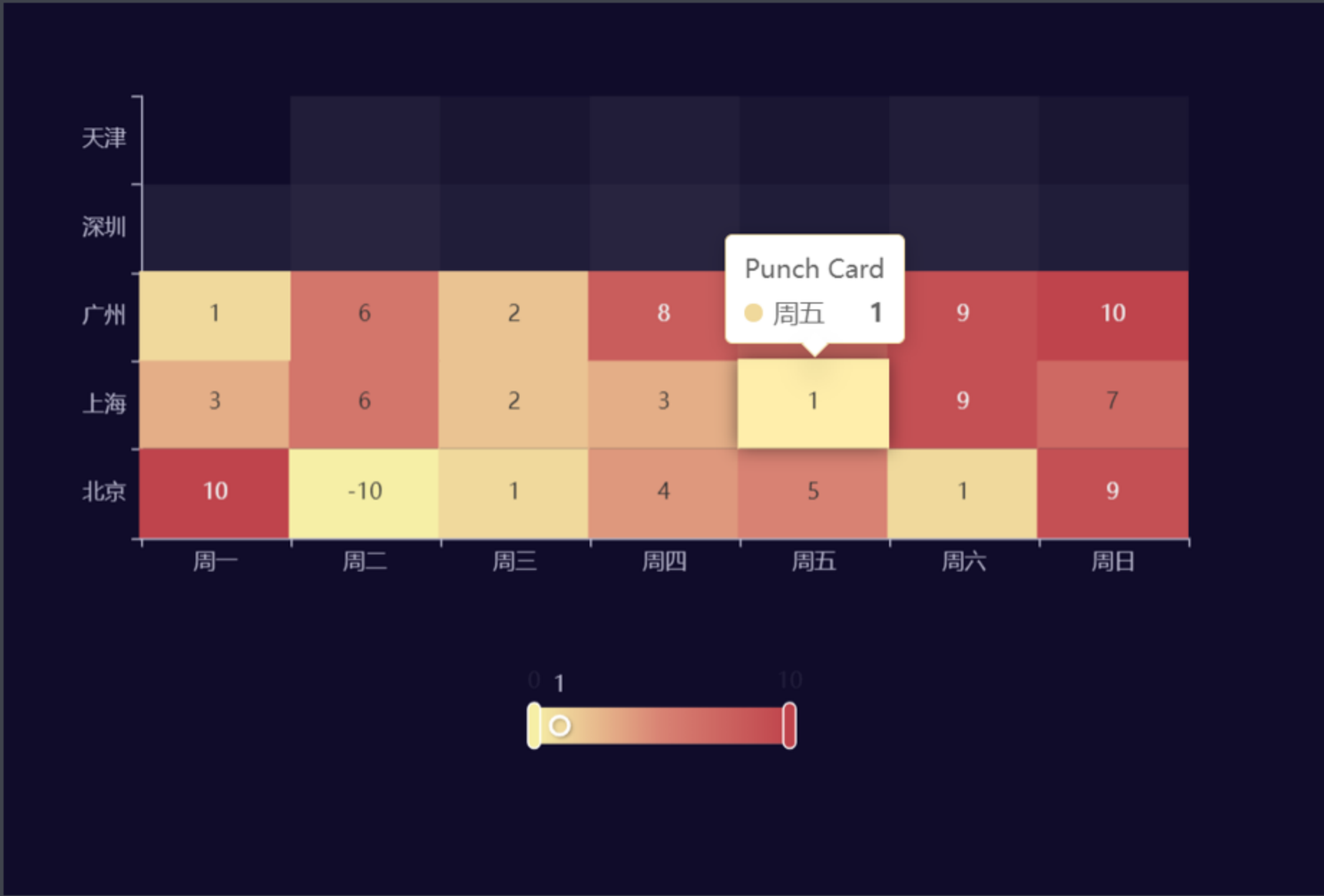
```
[  
  {  
    name: '最低气温',  
    data: [1, -20, 2, 5, 13, 8, 0]  
  },  
  {  
    name: '最高气温',  
    data: [10, 4, 12, 15, 26, 27, 10]  
  },  
]
```


数据转换图形步骤

- 执行映射将数据与不同标度对应
- 选择合适的图形形状将映射体现出来



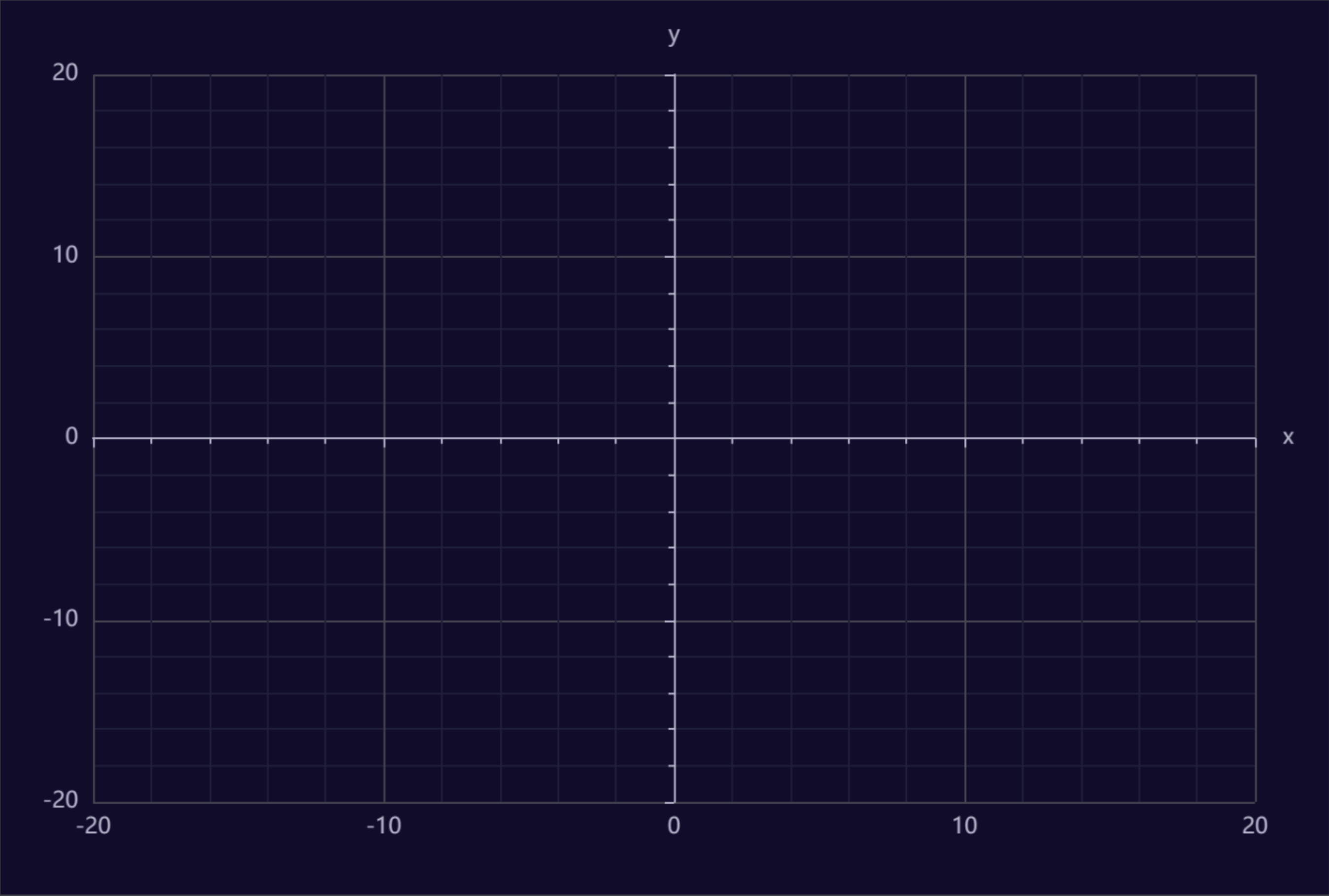
```
[
  {
    '北京': {
      name: '最低气温',
      data: [1, -20, 2, 5, 13, 8, 0]
    },
    {
      name: '最高气温',
      data: [10, 4, 12, 15, 26, 27, 10]
    }
  },
  {
    '上海': {
      name: '最低气温',
      data: [1, -20, 2, 5, 13, 8, 0]
    },
    {
      name: '最高气温',
      data: [10, 4, 12, 15, 26, 27, 10]
    }
  },
  {
    '广州': {
      name: '最低气温',
      data: [1, -20, 2, 5, 13, 8, 0]
    },
    {
      name: '最高气温',
      data: [10, 4, 12, 15, 26, 27, 10]
    }
  }
],
]
```



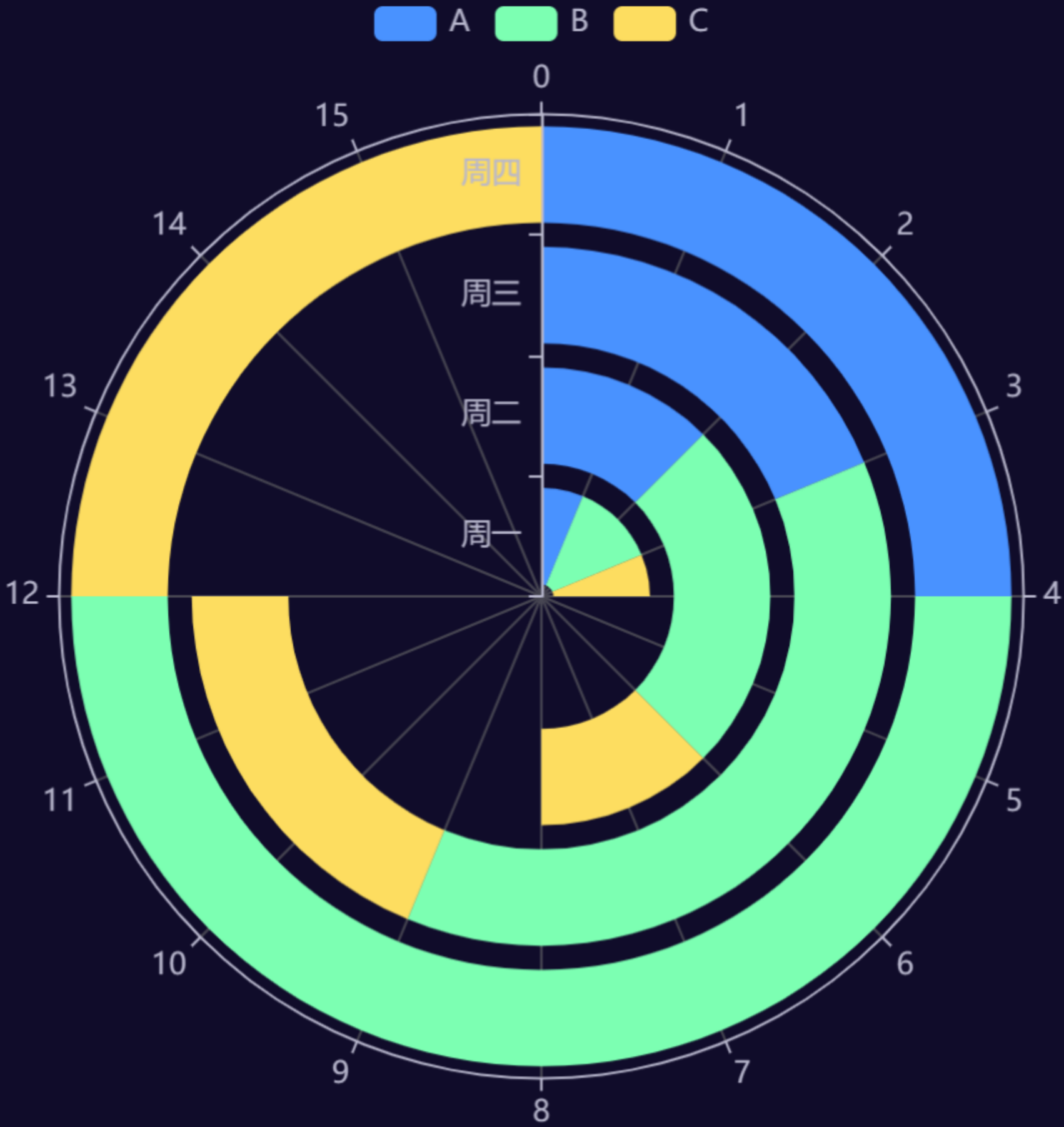
数据可视化

- 数据可视化就是将数据转换为图形展示
- 被展示的数据是分类型的
- 转换图形一般有二个步骤
 - 将数据与不同的标度对应，做好映射
 - 选择合适的图形来体现我们设计的映射

坐标轴



直角坐标系分为线性和非线性



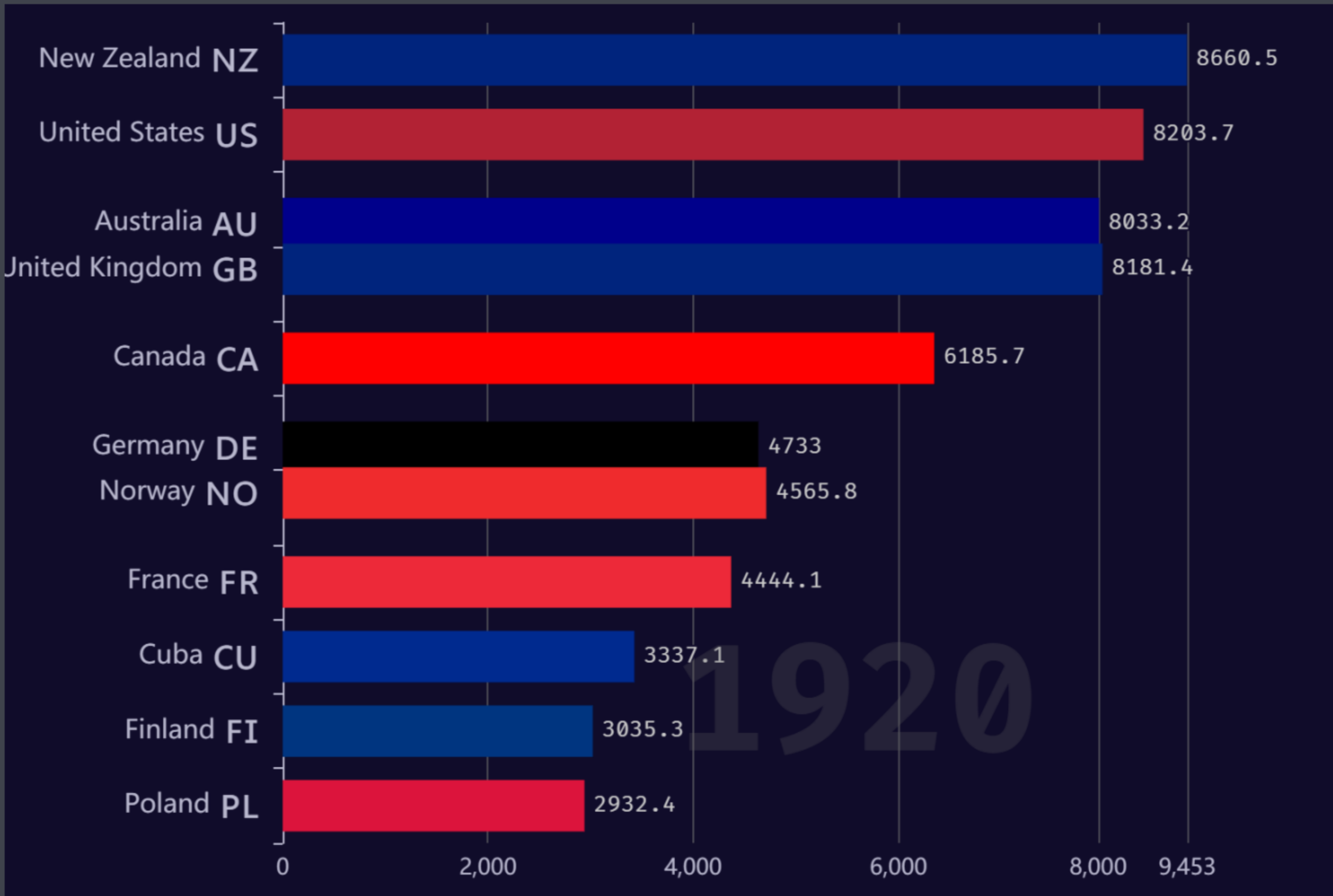
总结

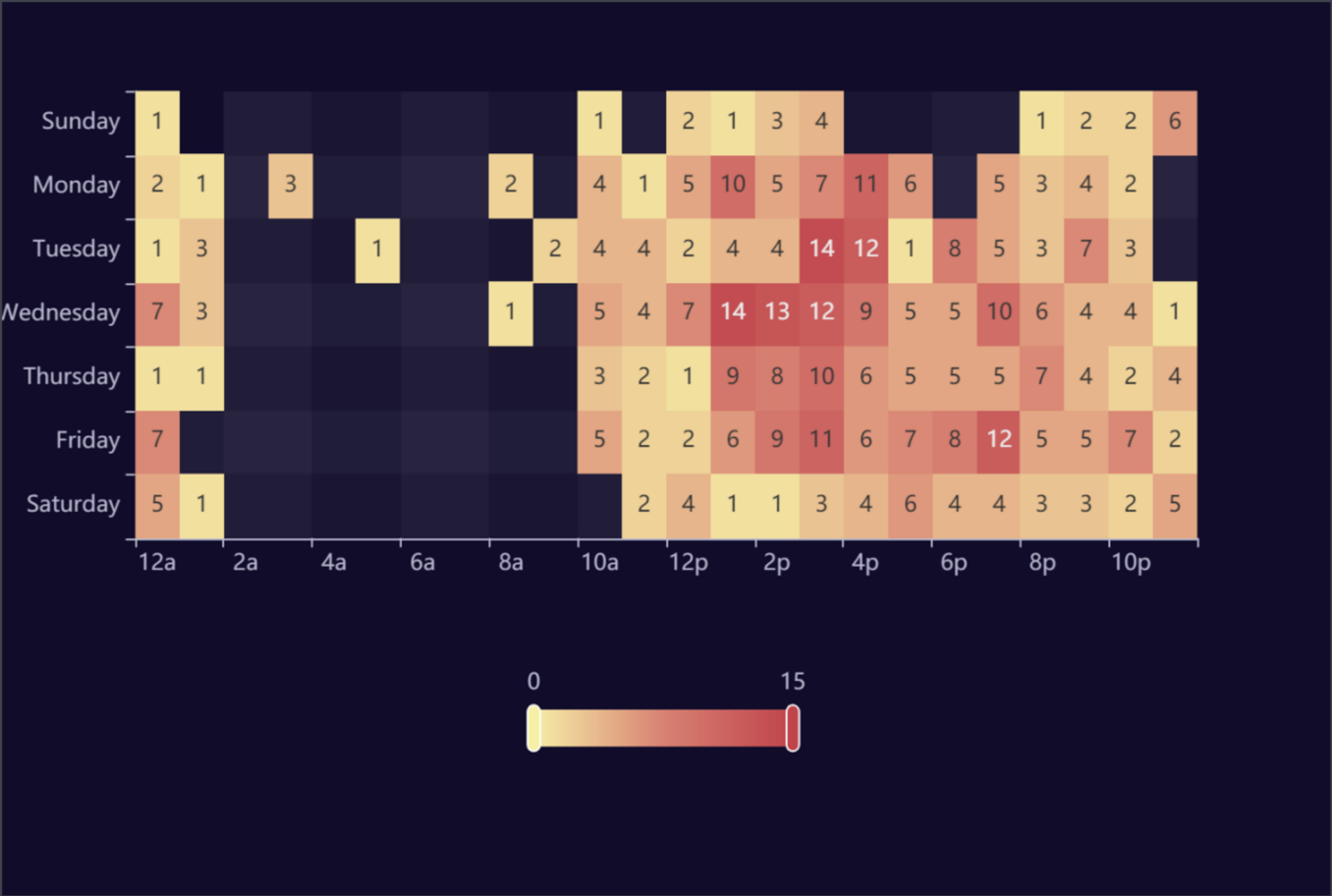
- 坐标轴是可视化数据的第一步，不同轴向展示数据标度
- 常见坐标轴是直角坐标轴
 - 线性直角坐标轴
 - 非线性直角坐标轴
- 极坐标轴

颜色标度

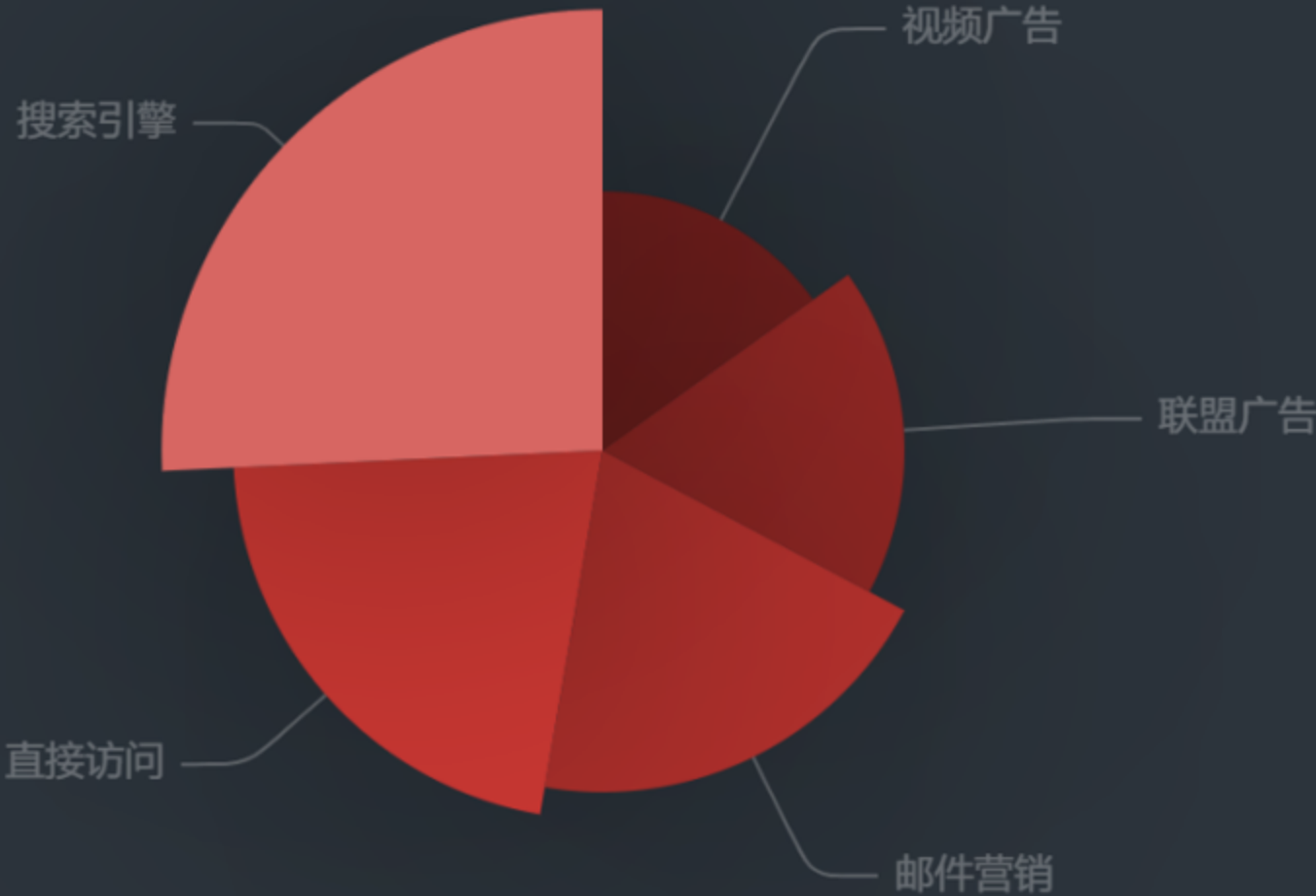
颜色标度作用

- 数据分组
- 不同颜色对应数据
- 指定数据突出显示





Customized Pie



总结

- 颜色是可视化当中重要的一个数据展示标度
- 颜色可以实现数据分组、展示、突出展示

可视化解决方案

Highcharts

Echarts

AntV

D3

zrender

three.js

HTML

Svg

Canvas

WebGL

浏览器(chrome)

SKia

OpenGL

Skia 是chrome 及安卓底层2D绘图引擎

OpenGL 是2D、3D图形渲染库

canvas绘制直线

canvas高清绘制

canvas绘制坐标系

canvas绘制直方图

canvas绘制圆形

Canvas 绘制文字

碰撞检测

Canvas 动画

Canvas 绘制关系图

myCharts 工程化配置

myCharts 初始化实现

myCharts 动画函数实现

进度圆环初始化

进度圆环绘制

折现图坐标系绘制

折线图展示数据绘制

折线图数据连线绘制

折线图刻度连线绘制

直方图绘制

Svg 绘制图形

像素图与矢量图

像素图由一个个像素块组成

矢量图由多个数学公式绘制曲线组成

矢量图由多个数学公式绘制曲线组成

D3.js 底层采用 svg 完成图形绘制

D3使用

D3 操作 svg

D3数据绑定

D3的三种选择集

D3 绘制直方图

线性比例尺使用

D3 常见比例尺

比例尺与坐标轴

D3 过渡

D3绘制动态直方图

D3 交互

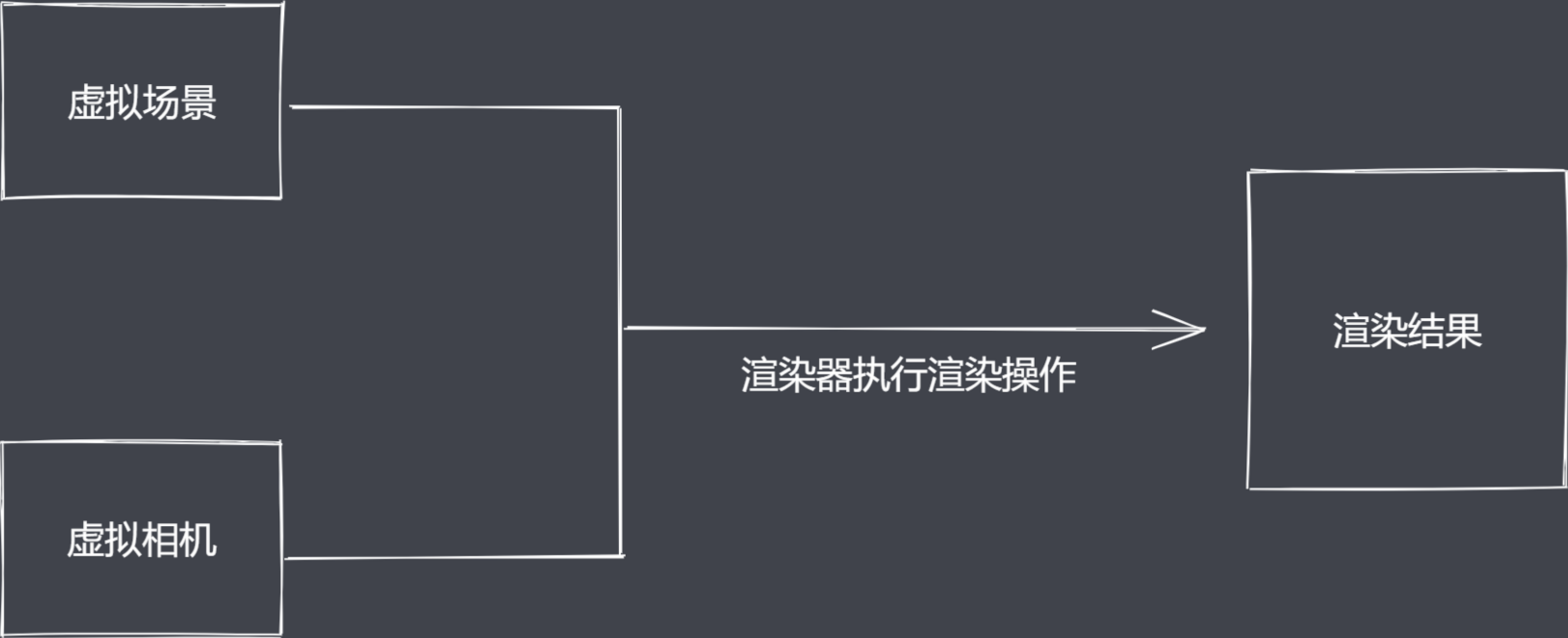
WebGL 与 ThreeJS

浏览器端的 3D 呈现技术

将 javascript 与 openGL ES2 结合操作显 卡

ThreeJS 是采用 javascript 编写的类
库

场景、相机、渲染、几何体



场景就是一个显示呈现的舞台

浏览器端呈现的内容都是相机拍摄

相机

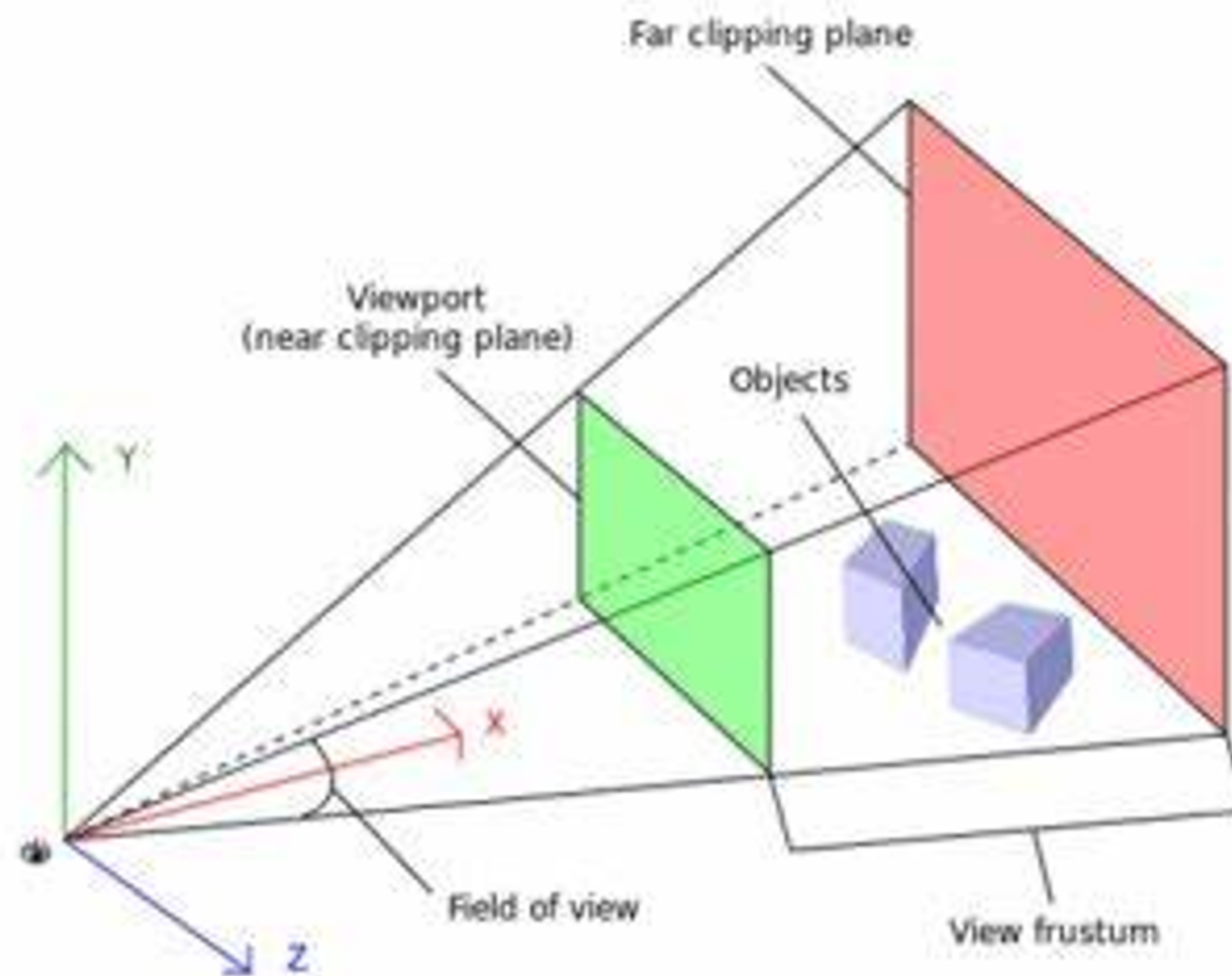
- 正投影相机

- 远处近处的内容做同等大小呈现处理

- 透视相机

- 符合心理习惯，近大远小，离视点近则大，远即小，灭点处消失

- `THREE.PerspectiveCamera = function(fov, aspect, near, far) {}`



渲染器决定了内容如何呈现至屏幕

ThreeJS 绘制立方体

材质与相机控制

场景添加光源

精灵材质与交互操作

VR 全景几何体实现

全景地标添加

全景看房实现