

# 山东大学 计算机科学与技术 学院

## 大数据分析实践 课程实验报告

学号：202300130236	姓名：陈德康	班级：数据 23
-----------------	--------	----------

实验题目：电子表格实践 I

实验学时：2

实验日期：2025.10.10

实验目的：

Add a new vis function based on the open source spreadsheet

硬件环境：

计算机一台

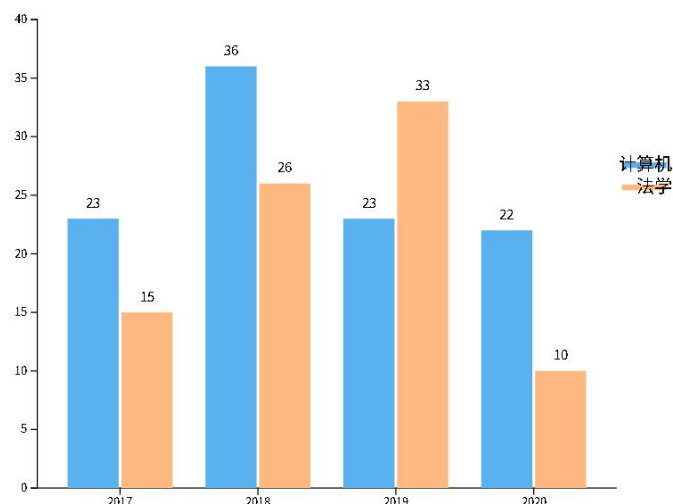
软件环境：

Windows11

Visual Studio Code

实验步骤与内容：

首先，跑通示例的 index.html 文件，最后可视化得到的柱状图如下图：



我尝试在此基础上实现一个功能，同样给出一个选择框，勾选之后能给每一列的数据都绘制一个饼状图，思路如下：

-实现一个新的函数 `function drawPieCharts(data, xtitle, ytitle)`，为每一列数据生成一个独立的饼状图，先定义了饼图尺寸、边距和布局规则，计算并创建合适大小的 SVG 容器；之后遍历每列的标题（如计算机），提取对应列的数据，并格式化为包含名称、数值和颜色的饼图数据；接着根据列索引计算每个饼图的位置，创建独立分组并添加专业名称作为标题

-在每个饼状图中，为每个扇区添加标签，从而清晰地显示对应的年份和数值，从而生成饼状图

代码如下:

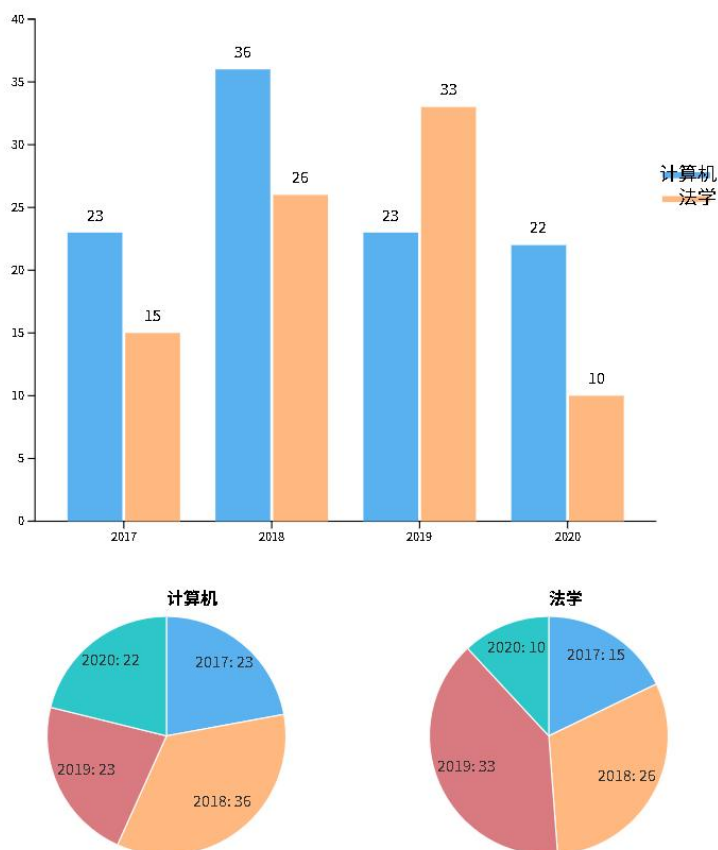
```
function drawPieCharts(data, xtitle, ytitle) {
  const pieSize = 200;
  const margin = { top: 40, right: 60, bottom: 40, left: 60 };
  const chartsPerRow = 3;
  const gap = 40;
  const totalWidth = (pieSize + margin.left + margin.right) * Math.min
(chartsPerRow, xtitle.length) - gap;
  const totalHeight = Math.ceil(xtitle.length / chartsPerRow) * (pieSi
ze + margin.top + margin.bottom);
  const svgContainer = d3.select("#my_dataviz")
    .append("svg")
    .attr("width", totalWidth)
    .attr("height", totalHeight);
  xtitle.forEach((major, colIndex) => {
    const pieData = ytitle.map((year, rowIndex) => ({
      name: year,
      value: data[rowIndex][colIndex],
      color: getColor(rowIndex)
    }));
    const row = Math.floor(colIndex / chartsPerRow);
    const col = colIndex % chartsPerRow;
    const xOffset = col * (pieSize + margin.left + margin.right) + mar
gin.left;
    const yOffset = row * (pieSize + margin.top + margin.bottom) + mar
gin.top;
    const pieGroup = svgContainer.append("g")
      .attr("transform", `translate(${xOffset + pieSize/2}, ${yOffset
+ pieSize/2})`);
    pieGroup.append("text")
      .attr("class", "pie-title")
      .attr("y", -pieSize/2 - 10)
      .attr("font-size", "14px")
      .attr("font-weight", "bold")
      .text(major);
    const pie = d3.pie()
      .value(d => d.value)
      .sort(null);
    const arc = d3.arc()
      .innerRadius(0)
      .outerRadius(pieSize / 2);
    pieGroup.selectAll("path")
      .data(pie(pieData))
      .join("path")
```

```

    .attr("d", arc)
    .attr("fill", d => d.data.color)
    .attr("stroke", "#fff")
    .attr("stroke-width", 1);
    const labelArc = d3.arc()
    .innerRadius(pieSize / 4)
    .outerRadius(pieSize / 2);
    pieGroup.selectAll("text.pie-sector-label")
    .data(pie(pieData))
    .join("text")
    .attr("class", "pie-label")
    .attr("transform", d => `translate(${labelArc.centroid(d)})`)
    .text(d => `${d.data.name}: ${d.data.value}`);
  });
}

```

结果如下图，同时勾选了柱状图和饼状图的绘制：



最终代码在 ex3.html 中

---

#### 结论分析与体会：

通过本次实验，我成功跑通了示例的 `index.html` 文件，文件实现了一个表格，勾选选项可以生成一个柱状图进行可视化；并且，我在此基础上编写了一个函数，为每一列的数据生成饼状图，从而丰富可视化的内容，让数据可以进行横向和纵向的对比；并且，我也体会到了可视化需要结合具体的需求进行设计，才能让数据呈现更加直观易懂，且第三方库的灵活运用能大幅提升开发效率，这为我进行进一步的学习和进行更复杂的实验奠定基础