

山东大学 计算机科学与技术 学院

大数据分析与实践 课程实验报告

学号：202300130205	姓名：李尚远	班级：23 级数据班
实验题目：电子表格实践 1		
实验学时：2	实验日期：20251024	
实验目的：基于原有库的基础上新增一种可视化		
硬件环境： 计算机一台		
软件环境： Linux 或 Windows		
实验步骤与内容： 一、 导入官方文件库 <pre><meta charset="UTF-8"> <meta name="viewport" content="width=device-width, initial-scale=1.0"> <title>表格与多图表可视化</title> <!-- 引入 x-spreadsheet 样式 --> <link rel="stylesheet" href="https://unpkg.com/x-data-spreadsheet@1.1.5/dist/xspreadsheet.css" /> <!-- 引入 x-spreadsheet 核心库 --> <script src="https://unpkg.com/x-data-spreadsheet@1.1.5/dist/xspreadsheet.js"></script > <!-- 引入 x-spreadsheet 中文语言包 --> <script src="https://unpkg.com/x-data-spreadsheet@1.1.9/dist/locale/zh-cn.js"></script > <!-- 引入 D3.js v6 --> <script src="https://d3js.org/d3.v6.js"></script></pre> 二、 定义样式 <pre><style> /* 表格容器样式 */ #xspreadsheet { width: 400px; height: 500px; padding: 0px; margin: 0px;</pre>		

```

    }
    /* 图表容器样式 */
    #my_dataviz {
        width: 1000px;
        height: 900px;
        padding: 0px;
        margin: 0px;
    }
    /* 坐标轴文本样式 */
    .ticktext {
        font-size: 20;
        stroke: black;
        stroke-width: 0.05em;
    }
}
</style>

```

三、 定义容器以及表格初始化

```

<body>
  <!-- 表格容器（包含复选框） -->
  <div id="xspreadsheet">
    <input type="checkbox" class="checkbox" value="barchart" />
    <label>柱状图</label>
    <br>
    <input type="checkbox" class="checkbox" value="linechart" />
    <label>折线图</label>
  </div>
  <!-- 图表容器 -->
  <div id="my_dataviz"></div>

  <script>
    // 初始化 x-spreadsheet 中文语言
    x_spreadsheet.locale("zh-cn");

    // 初始化表格实例
    var xs = x_spreadsheet("#xspreadsheet", {
      mode: 'edit', // 编辑模式（可选：edit/read）
      showToolbar: true, // 显示工具栏
      showGrid: true, // 显示网格线
      showContextMenu: true, // 显示右键菜单
      view: {
        height: () => document.documentElement.clientHeight,
        width: () => document.documentElement.clientWidth,
      },
      row: {
        len: 15, // 行数
      }
    });
  </script>

```

```
        height: 25, // 行高
      },
      col: {
        len: 8, // 列数
        width: 100, // 列宽
        indexWidth: 60, // 行索引宽度
        minWidth: 60, // 最小列宽
      },
      style: {
        bgcolor: '#ffffff', // 背景色
        align: 'left', // 水平对齐
        valign: 'middle', // 垂直对齐
        textwrap: false, // 自动换行
        strike: false, // 删除线
        underline: false, // 下划线
        color: '#0a0a0a', // 文字颜色
        font: {
          name: 'Helvetica', // 字体
          size: 10, // 字号
          bold: false, // 加粗
          italic: false, // 斜体
        },
      },
    },
  });
```

// 设置表格初始数据

```
xs.cellText(0, 1, "计算机")
  .cellText(0, 2, "法学")
  .render(); // 重新渲染表格
```

```
xs.cellText(1, 0, "2017")
  .cellText(1, 1, "23")
  .cellText(1, 2, "15")
  .render();
```

```
xs.cellText(2, 0, "2018")
  .cellText(2, 1, "36")
  .cellText(2, 2, "26")
  .render();
```

```
xs.cellText(3, 0, "2019")
  .cellText(3, 1, "23")
  .cellText(3, 2, "33")
  .render();
```

```

xs.cellText(4, 0, "2020")
  .cellText(4, 1, "22")
  .cellText(4, 2, "10")
  .reRender();

// 颜色调色板（用于图表配色）
function getColor(idx) {
  var palette = [
    '#5ab1ef', '#ffb980', '#d87a80', '#2ec7c9', '#b6a2de',
    '#8d98b3', '#e5cf0d', '#97b552', '#95706d', '#dc69aa',
    '#07a2a4', '#9a7fd1', '#588dd5', '#f5994e', '#c05050',
    '#59678c', '#c9ab00', '#7eb00a', '#6f5553', '#c14089'
  ];
  return palette[idx % palette.length];
}

```

四、 核心更新函数，实现表格修改、图绘制

```

// 核心更新函数（表格修改/复选框切换时触发）
function update() {
  // 获取复选框状态
  const barCheckbox = d3.select('input[value="barchart"]');
  const lineCheckbox = d3.select('input[value="linechart"]');

  // 清除现有图表
  d3.selectAll('svg').remove();

  if (barCheckbox.property("checked") ||
lineCheckbox.property("checked")) {
    // 1. 从表格中读取数据
    var data = []; // 存储数值数据
    var ytitle = []; // 存储行标题（年份）
    var xtitle = []; // 存储列标题（专业）
    var rows = 0; // 有效行数
    var col = 0; // 有效列数

    // 读取行标题（第 0 列，从第 1 行开始）
    for (var i = 1; i < 20; i++) {
      const cell = xs.cell(i, 0);
      if (cell === null || cell.text === undefined || cell.text ===
"" ) {
        rows = i; // 记录有效行数
        break;
      }
    }
  }
}

```

```

        data.push([]);
        ytitle.push(cell.text);
    }

    // 读取列标题（第 0 行，从第 1 列开始）
    for (var i = 1; i < 20; i++) {
        const cell = xs.cell(0, i);
        if (cell === null || cell.text === undefined || cell.text ===
"" ) {

            col = i; // 记录有效列数
            break;
        }
        xtitle.push(cell.text);
    }

    // 读取数值数据（排除标题行/列）
    for (var i = 1; i < rows; i++) {
        for (var j = 1; j < col; j++) {
            const cell = xs.cell(i, j);
            // 验证数据格式（必须为数字）
            if (cell === null || cell.text === undefined ||
isNaN(+cell.text)) {
                console.log('无效数据: 行', i, '列', j, '值', cell);
                return;
            }
            data[i - 1][j - 1] = +cell.text; // 转为数字存入
        }
    }

    // 2. 存储数据到 localStorage（临时缓存）
    window.localStorage.data = data;
    window.localStorage.xTitle = xtitle;
    window.localStorage.yTitle = ytitle;

    // 3. 从 localStorage 读取并格式化数据
    var xTitle = Array.from(window.localStorage.xTitle.split(","));
    var yTitle = Array.from(window.localStorage.yTitle.split(","));
    var list_data = window.localStorage.data.split(",");
    var pos = 0; // 数据索引指针

    // 重组数据格式（二维数组）
    var data1 = [];
    for (var i = 0; i < yTitle.length; i++) {
        let tmp = [];

```

```

        for (var j = 0; j < xTitle.length; ++j) {
            tmp.push(+list_data[pos++]);
        }
        data1.push(tmp);
    }

    // 计算数值最大值（用于 Y 轴定义域）
    var max = 0;
    var chartData = [];
    // 格式化数据为 D3 可用格式（对象数组）
    for (var i = 0; i < yTitle.length; i++) {
        var item = { group: yTitle[i] };
        chartData.push(item);
        for (var j = 0; j < xTitle.length; j++) {
            if (data1[i][j] > max) max = data1[i][j];
            item[xTitle[j]] = data1[i][j];
        }
    }

    // 4. 初始化 D3 图表
    const margin = { top: 40, right: 30, bottom: 40, left: 50 };
    const width = 600 - margin.left - margin.right;
    const height = 500 - margin.top - margin.bottom;

    // 创建 SVG 容器
    const svg = d3.select("#my_dataviz")
        .append("svg")
        .attr("width", width + margin.left + margin.right + 100) // 预
留图例空间

        .attr("height", height + margin.top + margin.bottom)
        .append("g")
        .attr("transform", `translate(${margin.left},
${margin.top})`);

    // 定义分组（X 轴：年份）
    const groups = yTitle;
    // 定义子分组（柱状图分组：专业）
    const subgroups = xTitle;

    // 5. 定义比例尺
    // X 轴比例尺（年份分组）
    const x = d3.scaleBand()
        .domain(groups)
        .range([0, width])

```

```

        .padding([0.2]);

// X 轴子比例尺（专业分组，每个年份下的柱子）
const xSubgroup = d3.scaleBand()
    .domain(subgroups)
    .range([0, x.bandwidth()])
    .padding([0.05]);

// Y 轴比例尺（数值范围）
const y = d3.scaleLinear()
    .domain([0, max])
    .range([height, 0])
    .nice(); // 优化刻度显示

// 6. 绘制坐标轴
// 底部 X 轴（年份）
svg.append("g")
    .attr("transform", `translate(0, ${height})`)
    .call(d3.axisBottom(x).tickSizeOuter(0));

// 左侧 Y 轴（数值）
svg.append("g")
    .call(d3.axisLeft(y));

if (barCheckbox.property("checked")) {
    // 7. 绘制柱状图
    svg.append("g")
        .selectAll("g")
        .data(chartData)
        .join("g")
        .attr("class", "bar")
        .attr("transform", (d) => `translate(${x(d.group)}, 0)` //
按年份分组平移

        .selectAll("rect")
        .data((d) => {
            // 为每个年份分组，生成对应专业的数值数据
            return subgroups.map((key) => ({ key: key, value:
d[key] }));

        })
        .join("rect")
        .attr("x", (d) => xSubgroup(d.key)) // 子分组 X 位置
        .attr("y", (d) => y(d.value)) // Y 位置（从顶部向下）
        .attr("width", xSubgroup.bandwidth()) // 柱子宽度
        .attr("height", (d) => height - y(d.value)) // 柱子高度

```

```

        .attr("fill", (d, i) => getColor(i)); // 柱子颜色

// 8. 绘制数值标签（柱子顶部）
svg.append("g")
    .selectAll("g")
    .data(chartData)
    .join("g")
    .attr("class", "bar")
    .attr("transform", (d) => `translate(${x(d.group)}, 0)` )
    .selectAll("text")
    .data((d) => {
        return subgroups.map((key) => ({ key: key, value:
d[key] }));
    })
    .join("text")
    .attr("x", (d) => xSubgroup(d.key) + xSubgroup.bandwidth()
* 0.5) // 水平居中

    .attr("y", (d) => y(d.value) - 10) // 位于柱子顶部上方 10px
    .text((d) => d.value) // 显示数值
    .attr('text-anchor', 'middle'); // 文本居中

// 9. 绘制图例
var legendData = xTitle.map((name, idx) => ({
    name: name,
    color: getColor(idx)
}));

const legend = svg.selectAll(".legend")
    .data(legendData)
    .enter()
    .append("g")
    .attr("class", "legend")
    .attr("transform", (d, i) => `translate(30, ${i * 20 +
120})`); // 图例位置

// 图例矩形
legend.append("rect")
    .attr("x", width - 25)
    .attr("y", 8)
    .attr("width", 40)
    .attr("height", 5)
    .style("fill", (d) => d.color);

// 图例文本

```



```
        legend.append("text")
            .attr("x", width + 20)
            .attr("y", 15)
            .style("text-anchor", "end")
            .text((d) => d.name);
    }

    if (lineCheckbox.property("checked")) {
        // 准备折线图数据
        const lineData = subgroups.map((subgroup, idx) => {
            return {
                name: subgroup,
                values: chartData.map(d => ({ year: d.group, value:
d[subgroup] })))
            };
        });

        // 定义折线生成器
        const line = d3.line()
            .x(d => x(d.year))
            .y(d => y(d.value));

        // 绘制折线
        const lines = svg.selectAll(".line")
            .data(lineData)
            .enter()
            .append("g")
            .attr("class", "line");

        lines.append("path")
            .attr("d", d => line(d.values))
            .attr("fill", "none")
            .attr("stroke", (d, i) => getColor(i))
            .attr("stroke-width", 2);

        // 绘制折线点
        lines.selectAll("circle")
            .data(d => d.values)
            .enter()
            .append("circle")
            .attr("cx", d => x(d.year))
            .attr("cy", d => y(d.value))
            .attr("r", 4)
            .attr("fill", (d, i, nodes) =>
```

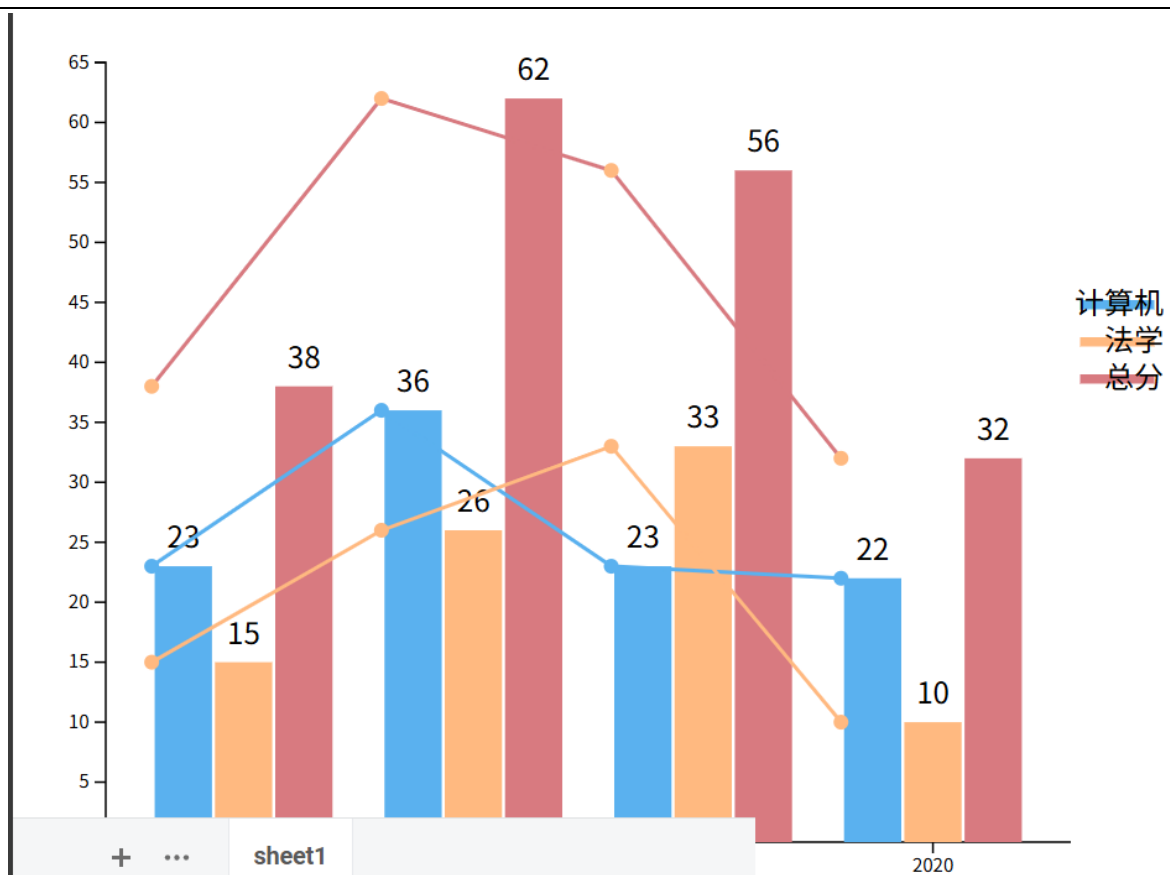
```
getColor(nodes[i].parentNode.__data__.name === subgroups[0] ? 0 : 1));

    // 折线图例（复用柱状图图例，若需单独图例可调整）
    }
  }
}
```

五、效果展示：

☒ 柱状图
 ☒ 折线图

	A	B	C	D	E	F	G	H
1		计算机	法学	总分				
2	2017	23	15	38				
3	2018	36	26	62				
4	2019	23	33	56				
5	2020	22	10	32				
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								



结论分析与体会：