

山东大学计算机科学与技术学院

大数据分析与实践实验报告

学号：202300130214	姓名：于博雅	班级：23级数据班		
实验题目：电子表格实践I				
实验学时：2	实验日期：2025. 10. 26			
实验目标： Add a new vis function based on the open source spreadsheet codes: https://github.com/myliang/x-spreadsheet				
实验环境： Window, vscode, 利用x-spreadsheet进行表格操作, 利用d3进行可视化				
实验步骤与结果： <h3>1. 导入官方依赖库</h3> <p>通过 CDN 引入 x-spreadsheet 表格库、d3.js 可视化库及中文语言包，为实验提供基础工具支持。</p> <pre><!-- 导入官方库（保持版本一致，调整CDN顺序） --> <script src="https://d3js.org/d3.v6.js"></script> <link rel="stylesheet" href="https://unpkg.com/x-data-spreadsheet@1.1.5/dist/xspreadsheet.css" /> <script src="https://unpkg.com/x-data-spreadsheet@1.1.5/dist/xspreadsheet.js"></script> <script src="https://unpkg.com/x-data-spreadsheet@1.1.9/dist/locale/zh-cn.js"></script></pre> <h3>2. 创建表格容器、图表控件区域及可视化展示区域，通过 CSS 优化布局与视觉效果。</h3> <pre><!-- 页面结构 --> <h1 class="page-title">电子表格与多图表联动分析</h1> <div class="content-container"> <!-- 表格与图表控件区域 --> <div id="xspreadsheet"> <div class="chart-controls"> <h3>请选择可视化类型</h3> <h3>（每次只可选择一种，更换前请取消之前的勾选）</h3> <label><input type="checkbox" class="chart-checkbox" value="bar" />柱状图</label> <label><input type="checkbox" class="chart-checkbox" value="line" />折线图</label> <label><input type="checkbox" class="chart-checkbox" value="pie" />饼图</label> <label><input type="checkbox" class="chart-checkbox" value="scatter" />散点图</label> </div> </div> <!-- 可视化结果展示区域 --> <div id="my_dataviz"></div> </div> <!-- CSS样式 --> <style> /* 全局样式优化：统一字体、增加页面间距 */ * {</pre>				

```
margin: 0;
padding: 0;
box-sizing: border-box;
font-family: 'Microsoft YaHei', 'Helvetica', sans-serif;
}

body {
    padding: 20px;
    background-color: #f8f9fa;
}

/* 标题样式：增加视觉层级 */

.page-title {
    font-size: 24px;
    color: #2c3e50;
    margin-bottom: 20px;
    padding-bottom: 10px;
    border-bottom: 2px solid #3498db;
}

/* 容器布局：采用弹性盒，优化响应式 */

.content-container {
    display: flex;
    gap: 30px;
    flex-wrap: wrap;
}

/* 表格容器美化：增加阴影、圆角 */

#xspreadsheet {
    width: 420px;
    height: 520px;
    padding: 15px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.08);
    flex-shrink: 0;
}

/* 可视化容器优化：增加内边距、阴影 */

#my_dataviz {
    flex: 1;
    min-width: 800px;
    height: 900px;
    padding: 20px;
    background-color: #fff;
    border-radius: 8px;
    box-shadow: 0 2px 8px rgba(0, 0, 0, 0.08);
}

/* 图表控件区域美化：调整配色、增加间距 */

.chart-controls {
    margin-bottom: 18px;
    padding: 12px;
    border: 1px solid #e0e0e0;
    border-radius: 6px;
    background-color: #f5f7fa;
}

.chart-controls h3 {
    margin: 0 0 12px 0;
    font-size: 16px;
```

```

        color: #34495e;
        font-weight: 600;
    }
    .chart-controls label {
        display: inline-block;
        margin: 0 15px 8px 0;
        font-size: 14px;
        color: #2c3e50;
        cursor: pointer;
        transition: color 0.2s;
    }
    .chart-controls label:hover {
        color: #3498db;
    }
    .chart-controls input[type="checkbox"] {
        margin-right: 6px;
        accent-color: #3498db;
    }
    /* 图表元素样式优化：坐标轴、标签字体调整 */
    .axis text {
        font-size: 12px;
        color: #4a4a4a;
    }
    .axis path {
        stroke: #ddd;
    }
    .chart-label {
        font-size: 11px;
        font-weight: 500;
        fill: #2c3e50;
    }
    /* 图例样式优化 */
    .legend text {
        font-size: 12px;
        color: #4a4a4a;
    }
}
</style>

```

3. 初始化电子表格

配置 `x-spreadsheet` 的编辑模式、行列尺寸、样式等属性，启用工具栏与网格线。

4. 预设表格数据

在表格中填入 2017-2020 年计算机、法学专业的样本数据，并渲染显示。

```

// 2. 预设表格数据（保持数据一致，调整赋值顺序）
const initTableData = () => {
    // 年份列（行1-行4）
    spreadsheet.cellText(1, 0, "2017").reRender();
    spreadsheet.cellText(2, 0, "2018").reRender();
    spreadsheet.cellText(3, 0, "2019").reRender();
    spreadsheet.cellText(4, 0, "2020").reRender();
    // 专业列（列1-列2）
    spreadsheet.cellText(0, 1, "计算机").reRender();
    spreadsheet.cellText(0, 2, "法学").reRender();
}

```

```

// 数据值
spreadsheet.cellText(1, 1, "23").cellText(1, 2, "15").reRender();
spreadsheet.cellText(2, 1, "36").cellText(2, 2, "26").reRender();
spreadsheet.cellText(3, 1, "23").cellText(3, 2, "33").reRender();
spreadsheet.cellText(4, 1, "22").cellText(4, 2, "10").reRender();
};

initTableData();

```

5. 定义工具函数

实现颜色配置（图表配色）与表格数据提取（数据校验、格式转换）功能。

6. 实现多图表渲染函数

通过ChartRenderer对象封装柱状图、折线图、饼图、散点图的渲染逻辑，包含坐标轴、图例、交互效果（hover）。

7. 实现核心更新函数

整合数据提取与图表渲染逻辑，根据复选框状态触发对应图表的显示 / 隐藏，避免多图表重叠。

```

// 6. 核心更新函数：整合数据提取与图表渲染（逻辑流程调整）
const handleDataUpdate = () => {
    // 清除现有图表
    d3.select("#my_dataviz").selectAll("svg").remove();

    // 提取表格数据
    const tableData = extractTableData();
    if (!tableData) return; // 数据无效时终止

    const { yearList, majorList, formattedData, dataMatrix, maxValue } = tableData;
    const container = d3.select("#my_dataviz");

    // 检查各图表复选框状态，按需渲染
    const isBarChecked = d3.select('input[value="bar"]').property("checked");
    const isLineChecked = d3.select('input[value="line"]').property("checked");
    const isPieChecked = d3.select('input[value="pie"]').property("checked");
    const isScatterChecked = d3.select('input[value="scatter"]').property("checked");

    // 多图表布局：按类型排列（避免重叠）
    let chartTop = 0;
    const chartSpacing = 20;

    // 渲染柱状图
    if (isBarChecked) {
        const barContainer = container.append("div")
            .attr("transform", `translate(0, ${chartTop})`);
        ChartRenderer.renderBar(barContainer, formattedData, yearList, majorList, maxValue);
        chartTop += 580 + chartSpacing; // 累加高度，避免后续图表重叠
    }

    // 渲染折线图
    if (isLineChecked) {
        const lineContainer = container.append("div")
    }
}

```

```

        .attr("transform", `translate(0, ${chartTop})`);

    ChartRenderer.renderLine(lineContainer, formattedData, yearList, majorList, maxValue);
    chartTop += 580 + chartSpacing;
}

// 渲染饼图（居中显示）
if (isPieChecked) {
    const pieContainer = container.append("div")
        .attr("transform", `translate(150, ${chartTop})`);

    ChartRenderer.renderPie(pieContainer, dataMatrix, majorList);
    chartTop += 480 + chartSpacing;
}

// 渲染散点图
if (isScatterChecked) {
    const scatterContainer = container.append("div")
        .attr("transform", `translate(0, ${chartTop})`);

    ChartRenderer.renderScatter(scatterContainer, dataMatrix, yearList, majorList, maxValue);
    chartTop += 580 + chartSpacing;
}

};


```

8. 绑定交互事件并初始化

为表格编辑事件与复选框状态变化事件绑定更新函数，确保数据修改与图表联动，并初始触发一次更新。

```

// 7. 绑定事件：表格编辑、复选框变化（事件绑定方式调整）
spreadsheet.on('cell-edited', handleDataUpdate);
d3.selectAll(".chart-checkbox").on("change", handleDataUpdate);

// 8. 初始触发一次更新（确保页面加载时默认显示空图表区域）
handleDataUpdate();

```

结果展示：

初始界面，可以进行选择可视化类型：

The screenshot shows a web-based data analysis tool. At the top, there's a modal dialog titled "请选择可视化类型" (Select Visualization Type) with the instruction "(每次只可选择一种，更换前请取消之前的勾选)" (Only one type can be selected at a time, please uncheck before changing). It contains four options: "柱状图" (Bar Chart), "折线图" (Line Chart), "饼图" (Pie Chart), and "散点图" (Scatter Plot), each with an unchecked checkbox. Below the dialog is a spreadsheet interface with a toolbar above it. The spreadsheet has columns labeled A, B, C, D, and E. The data in rows 1 through 5 is as follows:

	A	B	C	D	E
1	计算机	法学			
2	2017	23	15		
3	2018	36	26		
4	2019	23	33		
5	2020	22	10		
6					
7					
8					
9					

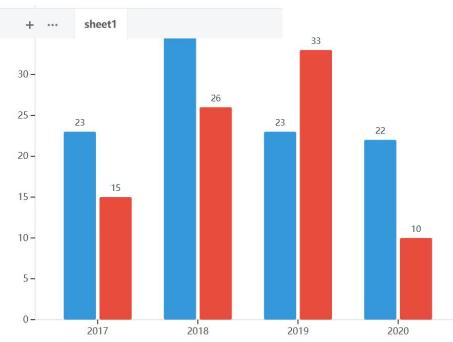
勾选柱状图：

电子表格与多图表联动分析

请选择可视化类型
(每次只可选择一种, 更换前请取消之前的勾选)

柱状图 折线图 饼图 散点图

A	B	C	D	E
1	2017	计算机	法学	
2	2018	23	15	
3	2019	36	26	
4	2020	23	33	
5	2020	22	10	



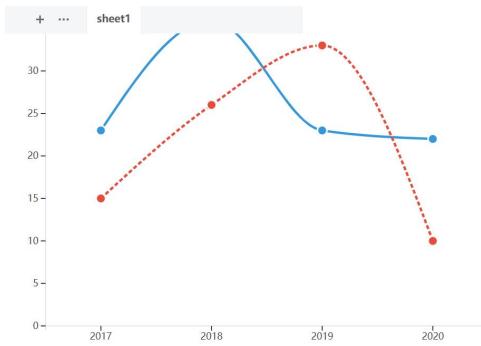
勾选折线图：

电子表格与多图表联动分析

请选择可视化类型
(每次只可选择一种, 更换前请取消之前的勾选)

柱状图 折线图 饼图 散点图

A	B	C	D
1	2017	计算机	法学
2	2018	23	15
3	2019	36	26
4	2020	23	33
5	2020	22	10



勾选饼图：

电子表格与多图表联动分析

请选择可视化类型
(每次只可选择一种, 更换前请取消之前的勾选)

柱状图 折线图 饼图 散点图

A	B	C
1	2017	计算机
2	2018	23
3	2019	36
4	2020	23
5	2020	22



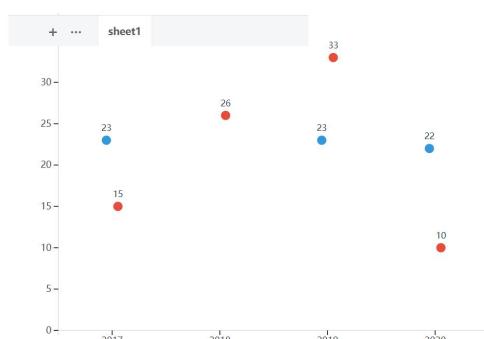
勾选散点图：

电子表格与多图表联动分析

请选择可视化类型
(每次只可选择一种, 更换前请取消之前的勾选)

柱状图 折线图 饼图 散点图

A	B	C	D
1	2017	计算机	法学
2	2018	23	15
3	2019	36	26
4	2020	23	33
5	2020	22	10



结论分析与体会：

基于开源电子表格工具 x-spreadsheet 扩展了可视化功能，实现了表格数据与柱状图的联动展示。

同时技术整合效果良好，通过 CDN 引入第三方库，高效完成了表格编辑与数据可视化的技术融合，d3.js 的实际应用，清晰掌握了从数据提取、格式转换到图形绘制、图例添加的完整流程，体会了“数据驱动视图”的核心思想。