

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: 202300130113	姓名: 丁正旸	班级: 23 数据		
实验题目: 电子表格实践 I				
实验学时: 2	实验日期: 2025. 10. 25			
实验目标:				
Add a new vis function based on the open source spreadsheet codes: https://github.com/myliang/x-spreadsheet				
实验过程与内容:				
步骤 1: 环境搭建与依赖导入 在 HTML 文档<head>标签中, 通过 CDN 方式导入 x-spreadsheet 的 CSS 和 JS 文件(含中文语言包), 以及 D3.js v6 库, 确保无需本地安装即可加载依赖:				
<pre><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> <script src="https://d3js.org/d3.v6.js"></script></pre> 定义 HTML 结构, 包含电子表格容器 (#xspreadsheet)、图表容器 (#my_dataviz) 及图表控制复选框, 设置基础 CSS 样式(固定表格和图表尺寸, 优化元素显示效果)。				
步骤 2: 电子表格初始化与数据设置 调用 x_spreadsheet.locale("zh-cn") 设置表格中文环境, 通过 x_spreadsheet() 方法创建表格实例, 配置编辑模式、行列数、单元格样式等参数:				
<pre>var xs = x_spreadsheet("#xspreadsheet", { mode: 'edit', row: { len: 15, height: 25 }, col: { len: 8, width: 100 }, // 其他样式配置... });</pre> 使用 cellText(row, col, text) 方法设置初始数据, 模拟 2017-2020 年 “计算机” “法学” 两个专业的人数统计(行标题为年度, 列标题为专业), 并通过 reRender() 刷新表格显示。				
步骤 3: 数据读取与格式校验 在 update() 函数中, 分三步读取表格数据:				
读取行标题: 遍历左侧第一列(从第 1 行开始), 获取年度信息(2017-2020), 直到遇到空单元格; 读取列标题: 遍历顶部第一行(从第 1 列开始), 获取专业信息(计算机、法学), 直到遇到空单元格; 读取数值数据: 遍历有效行和列, 校验数据格式(非空且为数字), 将文本转为数值存入二维数组。 数据校验: 若某单元格为空或非数字, 在控制台打印错误信息并终止流程, 避免无效数据导致图表绘制异常。				
步骤 4: 柱状图绘制与联动更新 比例尺定义:				
主 X 轴(年度): 使用 d3.scaleBand 映射年度分类数据到图表宽度, 设置间距 padding(0.2); Y 轴(人数): 使用 d3.scaleLinear 映射数值数据(0 到最大值)到图表高度, 通过 nice() 优化刻度为整数; 子 X 轴(专业): 在每个年度的 X 轴范围内, 为不同专业分配宽度, 设置子柱子间距 padding(0.05)。 图表元素绘制:				
坐标轴: 通过 d3.axisBottom(x) 和 d3.axisLeft(y) 绘制 X 轴(底部) 和 Y 轴(左侧), 隐藏 X 轴首尾刻度				

线；

柱状图：绑定数据后，使用 `rect` 元素绘制柱子，通过 `getColor()` 函数为不同专业分配颜色；

数值标签：在每个柱子顶部添加 `text` 元素，显示具体人数，确保标签水平居中且位于柱子上方 `10px` 处；

图例：在图表右侧绘制颜色块和专业名称，通过纵向排列区分不同专业。

联动更新：

复选框事件：勾选 “显示柱状图” 时绘制图表，取消勾选时清除所有 `SVG` 元素；

表格编辑事件：修改表格数据后，自动触发 `update()` 函数，重新读取数据并更新图表，实现 “编辑即更新”。

(三) 关键代码说明

颜色调色板函数：定义 20 种颜色，通过索引模实现循环使用，确保不同专业的柱子颜色唯一且美观：

```
function getColor(idx) {  
    var palette = ['#5ab1ef', '#ffb980', '#d87a80', ...]; // 20 种颜色  
    return palette[idx % palette.length];  
}
```

数据缓存优化：使用 `localStorage` 存储数据（需通过 `JSON.stringify` 和 `JSON.parse` 处理数组），避免页面刷新后数据丢失：

```
window.localStorage.data = JSON.stringify(data); // 存储  
var list_data = JSON.parse(window.localStorage.data).flat(); // 读取并展平
```

图表清除逻辑：绘制新图表前，通过 `d3.selectAll('svg').remove()` 清除旧图表，避免重复绘制导致重叠。

结论与体会：

本次实验通过 `x-spreadsheet` 和 `D3.js` 实现了 “数据编辑 – 可视化联动” 的完整流程，不仅掌握了具体技术的使用方法，更理解了大数据可视化的核心逻辑——“以数据为核心，以用户理解为目标”。后续可在现有基础上扩展功能，优化体验，为更复杂的大数据分析场景（如多维度数据对比、实时流数据可视化）奠定基础。