

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：202320130203	姓名：李姿含	班级：数据 23
实验题目：电子表格实践 1		
实验学时：2	实验日期：2025	
<p>实验目标：</p> <p>基于开源电子表格代码 x-spreadsheet 添加新的可视化功能，实现表格数据与柱状图可视化的联动，掌握利用 d3.js 进行数据可视化的基本方法，熟悉前端数据处理与可视化的完整流程。</p>		
<p>作品描述（实验背景、数据集来源、描述思路、实验背景）：</p> <p>① 实验背景：</p> <p>在数据处理场景中，电子表格是常用的数据录入与管理工具，但原生表格缺乏直观的可视化展示功能，难以快速呈现数据规律。本实验旨在通过扩展开源电子表格的功能，实现数据录入与可视化的实时联动，提升数据解读效率。</p> <p>② 数据集来源：</p> <p>实验使用自定义的模拟数据集，记录了 2017 年至 2020 年计算机专业与法学专业的相关统计数据，包含年份、计算机专业数据、法学专业数据三个维度。</p> <p>③ 描述思路：</p> <p>1、搭建基础环境，先导入 x-spreadsheet 的样式文件与核心脚本，同时引入用于本地化支持的 zh-cn.js 脚本，以及实现可视化的 d3.v6.js 库，为电子表格功能与可视化功能提供依赖支持；</p> <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> <p>2、设计交互控件，在电子表格容器内添加复选框及“barchart”标签，作为可视化功能的触发开关，用户通过勾选或取消勾选控制图表的显示与隐藏；</p> <pre><div> <input type="checkbox" class="checkbox" value="barchart" /><label>barchart</label> </div></pre> <p>3、配置电子表格参数，设置表格的编辑模式、工具栏、网格线、上下文菜单的显示状态，定义表格视图的宽高计算方式、行与列的数量及尺寸，以及单元格的默认样式，并预设 2017-2020 年的年份数据及对应计算机、法学专业数据，完成初始数据填充后重新渲染表格；</p>		

```

//设置初值
xs.on('cell-edited',update)
xs.cellText(0, 1, "计算机").cellText(0, 2, "法学").reRender();
xs.cellText(1, 0, "2017")
.cellText(1, 1, "23")
.cellText(1, 2, "15")
.reRender();
xs.cellText(2, 0, "2018")
.cellText(2, 1, "36")
.cellText(2, 2, "26")
.reRender();
xs.cellText(3, 0, "2019")
.cellText(3, 1, "23")
.cellText(3, 2, "33")
.reRender();
xs.cellText(4, 0, "2020")
.cellText(4, 1, "22")
.cellText(4, 2, "10")
.reRender();
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];
}

```

- 4、编写数据处理逻辑，创建 getColor 函数定义可视化图表的颜色 palette（包含 20 种不同颜色，通过索引取余确保颜色循环使用），再编写核心的 update 更新函数，该函数首先判断复选框状态，若勾选则执行数据处理流程：先遍历表格行数据，从第 1 行开始读取每行第 0 列的年份信息，直到遇到空值或未定义值停止，记录有效行数并存储年份至 ytitle 数组，同时初始化数据存储数组；接着遍历表格列数据，从第 1 列开始读取第 0 行的专业名称信息，遇到空值或未定义值停止，记录有效列数并存储专业名称至 xtitle 数组；然后遍历有效行列范围内的单元格数据，验证数据是否为空、未定义或非数值类型，若存在无效数据则在控制台打印信息并终止数据处理，若数据有效则转换为数值类型存入数据数组；之后将处理后的 data、xtitle、ytitle 数据存储到 window.localStorage 全局变量中，实现数据的临时保存与共享；再从 localStorage 中读取数据并进行格式转换，将字符串形式的数组拆分并转换为符合 d3.js 处理要求的结构化数据，同时计算数据中的最大值用于设置纵轴刻度范围；

```

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

```

```

function update() {
const a=d3.select('.checkbox');
if(a.property("checked")){
var data = [];
var ytitle = [];
var xtitle = [];
var col = 0;
var rows;
for (var i= 1;i< 20;i++) {
if (xs.cell(i, 0) === null ||xs.cell(i, 0).text === undefined ||xs.cell(i, 0).text === "") {
rows=i;
break;
}
data.push([]);
ytitle.push(xs.cell(i, 0).text);
}
for (var i= 1;i< 20;i++) {
if (
xs.cell(0,i) === null ||xs.cell(0,i).text === undefined ||xs.cell(0,i).text === "") {
col=i;
break;
}
xtitle.push(xs.cell(0,i).text);
}
for (var i= 1;i< rows;i++) {
for (var j= 1; j < col;j++) {
if (
xs.cell(i, j) === null ||
xs.cell(i, j).text === undefined ||

```

- 5、实现可视化渲染，利用 d3.js 先清除页面中已存在的 svg 元素（避免重复渲染），然后创建新的 svg 容器，设置容器的宽高（包含边距）并调整位置；定义分组（groups）为年份、子分组（subgroups）为专业名称，分别创建横轴（x 轴，基于年份的分类轴）和纵轴（y 轴，基于数据最大值的线性轴），并添加到 svg 容器中；设置颜色比例尺，将不同专业名称映射到预设的颜色 palette；创建子分组比例尺，用于在每个年份分组内分配不同专业柱子的位置；接着绘制柱状图，先按年份分组，再在每个分组内按专业绘制矩形，设置柱子的 x 坐标（基于年份分组与子分组位置）、y 坐标（基于数据值对应的纵轴位置）、宽度（子分组带宽）、高度（纵轴底部到数据值位置的距离）及填充色（对应专业颜色）；然后为每个柱子添加数据标签，标签位置在柱子顶部居中，显示对应的数据值；最后创建图例，遍历专业名称与对应颜色，绘制矩形色块与文字标签，设置图例在 svg 容器中的位置（右侧区域，按专业顺序垂直排列），确保图例与柱状图颜色一一对应；若复选框未勾选，则清除页面中所有 svg 元素，停止可视化展示；
- 6、绑定事件监听，将 update 函数分别绑定到表格的“cell-edited”事件（表格数据修改时触发，实现可视化实时更新）与复选框的“change”事件（勾选状态变化时触发，控制可视化的显示与隐藏），确保表格数据与可视化图表的联动效果。

