

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

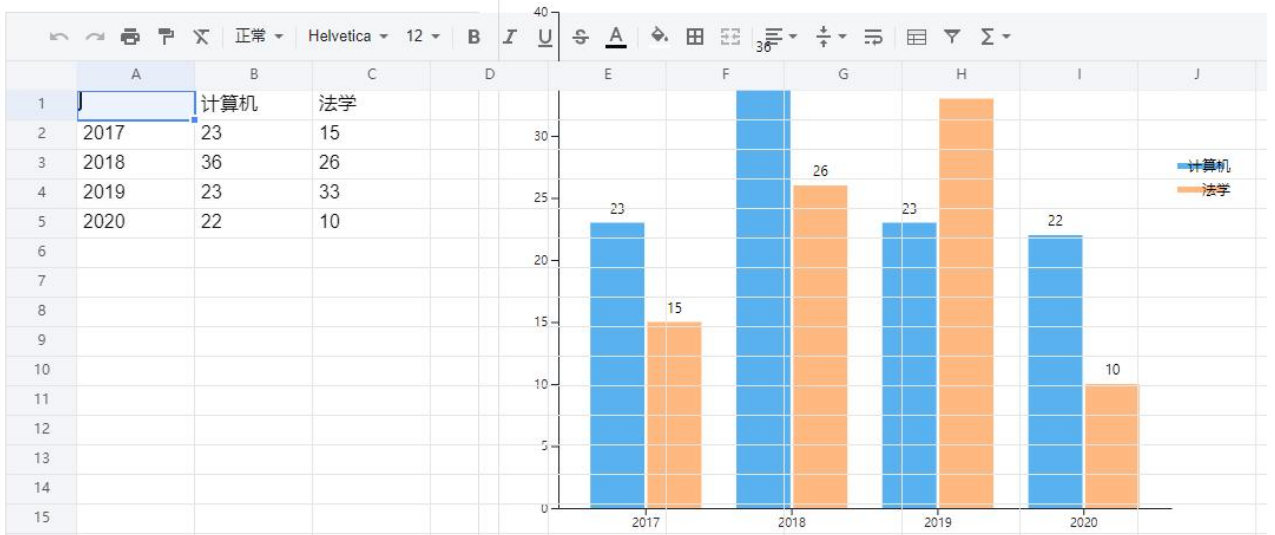
学号：202300130090	姓名：杨笑语	班级：数据班
实验题目：人机交互实验 3		
实验学时：2	实验日期：2025/10/10	
实验要求：基于开源电子表格 x-data-spreadsheet 新增一个可视化功能：勾选复选框后，将电子表格中的数据以分组柱状图（grouped bar chart）可视化展示，取消勾选则移除图表。在编辑单元格或切换复选框时，图表能够自动更新。		
硬件环境： 计算机一台		
软件环境： Windows, Vs Code, 浏览器 前端库：x-data-spreadsheet@1.1.5（含 zh-cn 本地化）, d3.v6		
实验步骤与内容： 1. 页面结构与依赖 网页布局左侧为电子表格容器 xspreadsheet，上方有复选框控制是否可视化；右侧为可视化区域 my_dataviz。通过 CDN 引入所需 CSS/JS。  2. 数据约定 第 1 行（从第 2 列开始）为列标题，即 X 轴子组（如“计算机”“法学”）。 第 1 列（从第 2 行开始）为行标题，即分组（如“2017”“2018”）。 （交叉区域必须为数值）  3.测试步骤 （1）打开 index.html，确认加载成功（控制台无报错）。 （2）在表格中录入数据，第 1 行第 2 列起录入列标题（如“计算机”“法学”）；第 1 列第 2 行起录入行标题（如“2017”“2018”“2019”“2020”）；交叉区域录入数值，确保是数字。 （3）勾选`barchart`，触发`update`收集标题与数据矩阵，进行空值与非数值校验，使用 d3 绘制分组柱状图，添加数值标签与图例。 （4）修改任意单元格或复选框状态，观察图表实时更新或清空。  4. 结果展示 （1）未勾选 barchart，只有数据没有图表		

☐ barchart

	A	B	C	D	E	F	G	H	I	J
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										
10										
11										

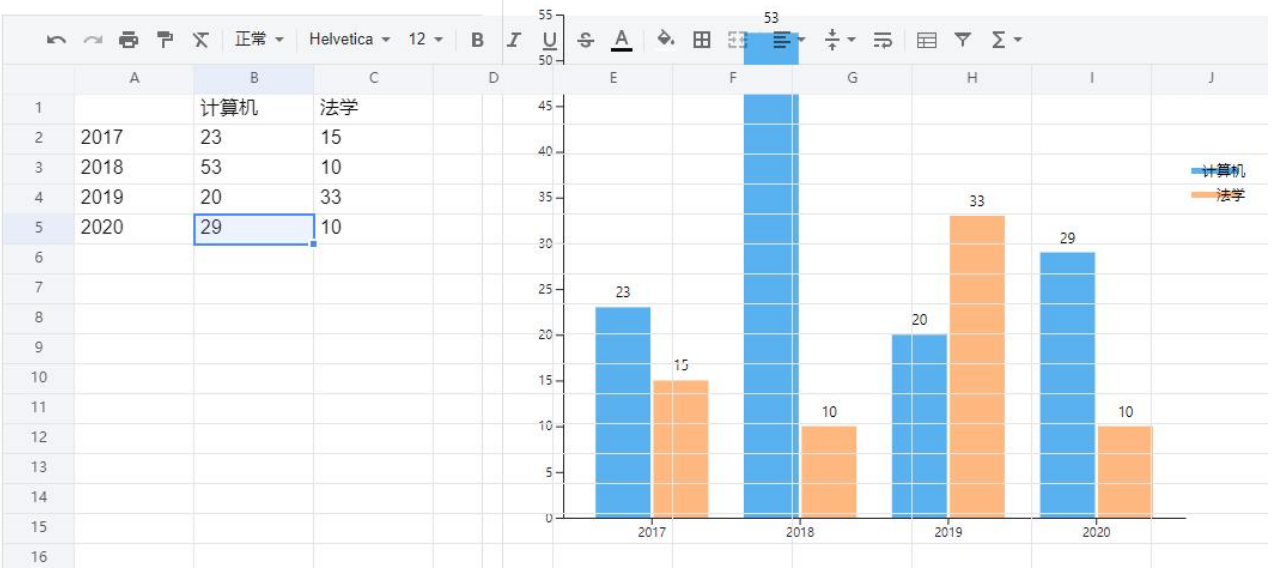
(2) 勾选 barchart，数据、图表都出现

☒ barchart

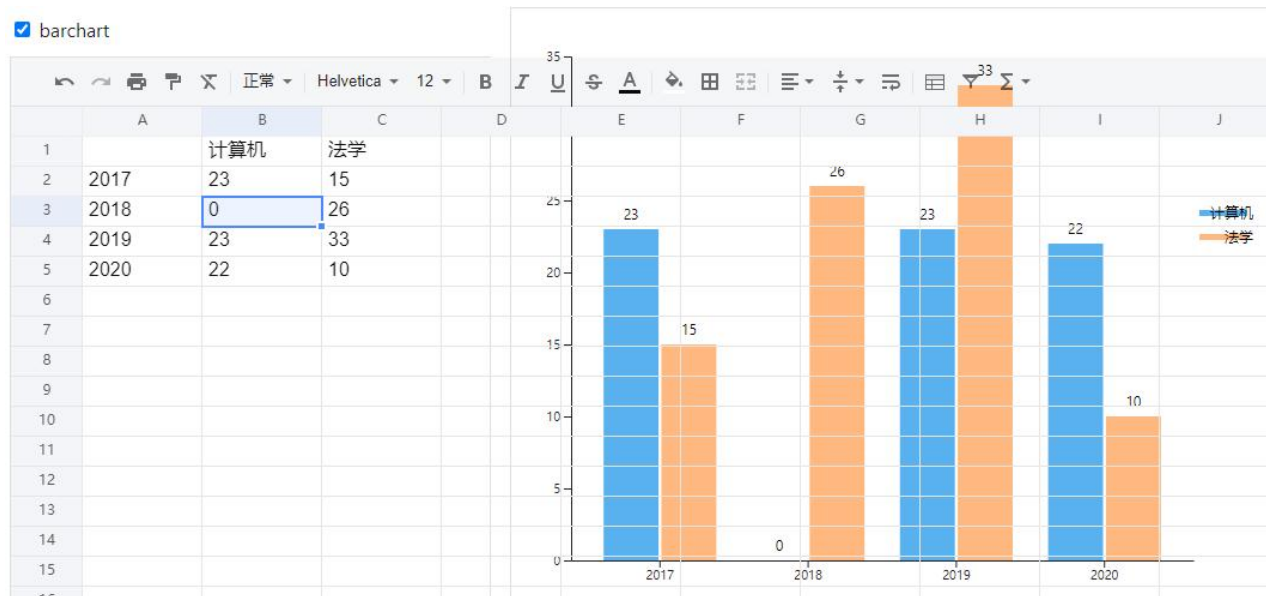


(3) 修改计算机 '53'、法学 '10' 等多个数据后图表实时更新

☒ barchart



#### (4) 数据取 0 后柱形消失



#### 5. 关键代码（节选） - 依赖引入与容器：

```
``1:9:index.html
```

```
<!doctype html>
```

```
<html lang="zh-CN">
```

```
<head>
```

```
<meta charset="utf-8" />
```

```
<meta name="viewport" content="width=device-width, initial-scale=1" />
```

```
<title>电子表格实践 I - x-spreadsheet + d3 可视化</title>
```

```
<link rel="stylesheet" href="https://unpkg.com/x-data-spreadsheet@1.1.5/dist/xspreadsheet.css" />
```

```
...
```

```
``47:59:index.html
```

```
<body>
```

```
<div class="page">
```

```
<div>
```

```
<div class="controls">
```

```
<label>
```

```
<input type="checkbox" class="checkbox" value="barchart" />
```

```
barchart
```

```
</label>
```

```
</div>
```

```
<div id="xspreadsheet"></div>
```

```
</div>
```

```
<div id="my_dataviz"></div>
```

---

```
</div>
```

```
...
```

- 初始化电子表格与初值:

```
```69:89:index.html
```

```
const xs = x_spreadsheet('#xspreadsheet', {
  mode: 'edit',
  showToolbar: true,
  showGrid: true,
  showContextmenu: true,
  row: { len: 50, height: 25 },
  col: { len: 20, width: 100, indexWidth: 60, minWidth: 60 },
  style: {
    bgcolor: '#ffffff', align: 'left', valign: 'middle', textwrap: false,
    strike: false, underline: false, color: '#0a0a0a',
    font: { name: 'Helvetica', size: 12, bold: false, italic: false }
  }
});
```

```
// 初值：第一行做列标题（从第 2 列起），第一列做行标题（从第 2 行起），其余为数值
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();
```

```
...
```

- 数据收集与校验:

```
```101:134:index.html
```

```
function collectTableData() {
  // 收集 X 轴子组（第一行第 2 列起）与 Y 轴分组（第一列第 2 行起）
  const xTitle = [];
  const yTitle = [];
  let colCount = 0;
  let rowCount = 0;

  for (let c = 1; c < 1000; c += 1) {
    const cell = xs.cell(0, c);
    if (!cell || cell.text === undefined || cell.text === '') { colCount = c; break; }
    xTitle.push(String(cell.text));
  }
}
```

```

for (let r = 1; r < 1000; r += 1) {
  const cell = xs.cell(r, 0);
  if (!cell || cell.text === undefined || cell.text === "") { rowCount = r; break; }
  yTitle.push(String(cell.text));
}

if (xTitle.length === 0 || yTitle.length === 0) return null;

const dataMatrix = Array.from({ length: yTitle.length }, () => Array(xTitle.length).fill(0));
for (let r = 1; r < rowCount; r += 1) {
  for (let c = 1; c < colCount; c += 1) {
    const cell = xs.cell(r, c);
    if (!cell || cell.text === undefined || cell.text === "") return null;
    const value = Number(cell.text);
    if (Number.isNaN(value)) return null;
    dataMatrix[r - 1][c - 1] = value;
  }
}

return { xTitle, yTitle, dataMatrix };
}

```

- 可视化更新（核心）：

```

136:246:index.html
function update() {
  const checked = d3.select('.checkbox').property('checked');
  // 清理画布
  d3.select('#my_dataviz').selectAll('*').remove();

  if (!checked) return;

  const table = collectTableData();
  if (!table) return; // 数据不完整或格式不对

  // 将数据存为全局变量（亦可持久化到 localStorage）
  window.xTitle = table.xTitle.slice();
  window.yTitle = table.yTitle.slice();
  window.tableData = table.dataMatrix.map(row => row.slice());

  // 计算最大值
  const maxValue = d3.max(window.tableData.flat());

```

```
// 分组柱状图 (grouped bar chart)
const margin = { top: 40, right: 30, bottom: 40, left: 50 };
const width = 600 - margin.left - margin.right;
const height = 500 - margin.top - margin.bottom;

const svg = d3
  .select('#my_dataviz')
  .append('svg')
  .attr('width', width + margin.left + margin.right + 100)
  .attr('height', height + margin.top + margin.bottom)
  .append('g')
  .attr('transform', `translate(${margin.left},${margin.top})`);

const groups = window.yTitle;
const subgroups = window.xTitle;

const x = d3.scaleBand().domain(groups).range([0, width]).padding(0.2);
svg.append('g')
  .attr('transform', `translate(0, ${height})`)
  .call(d3.axisBottom(x).tickSizeOuter(0));

const y = d3.scaleLinear().domain([0, maxValue]).range([height, 0]).nice();
svg.append('g').call(d3.axisLeft(y));

const xSubgroup = d3.scaleBand().domain(subgroups).range([0, x.bandwidth()]).padding(0.05);

// 将数据结构转换为 [{ group: '2017', A: 23, B: 15 }, ...]
const series = groups.map((g, gi) => {
  const row = { group: g };
  subgroups.forEach((k, ki) => { row[k] = window.tableData[gi][ki]; });
  return row;
});

// 柱
svg.append('g')
  .selectAll('g')
  .data(series)
  .join('g')
  .attr('class', 'bar')
  .attr('transform', d => `translate(${x(d.group)}, 0)`)
  .selectAll('rect')
  .data(d => subgroups.map(key => ({ key, value: d[key] })))
  .join('rect')
  .attr('x', d => xSubgroup(d.key))
```

```
.attr('y', d => y(d.value))
.attr('width', xSubgroup.bandwidth())
.attr('height', d => height - y(d.value))
.attr('fill', (d, i) => getColor(i));

// 数值标签
svg.append('g')
  .selectAll('g')
  .data(series)
  .join('g')
  .attr('transform', d => `translate(${x(d.group)}, 0)`)
  .selectAll('text')
  .data(d => subgroups.map(key => ({ key, value: d[key] })))
  .join('text')
  .attr('x', d => xSubgroup(d.key) + xSubgroup.bandwidth() * 0.5)
  .attr('y', d => y(d.value) - 8)
  .attr('text-anchor', 'middle')
  .style('font-size', '12px')
  .text(d => d.value);

// 图例
const legendData = subgroups.map((name, i) => ({ name, color: getColor(i) }));
const legend = svg
  .selectAll('.legend')
  .data(legendData)
  .enter()
  .append('g')
  .attr('class', 'legend')
  .attr('transform', (d, i) => `translate(30, ${i * 20 + 120})`);

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

legend.append('text')
  .attr('x', width + 20)
  .attr('y', 15)
  .style('text-anchor', 'end')
  .style('font-size', '12px')
  .text(d => d.name);
}
```

---

```
// 事件绑定
xs.on('cell-edited', update);
d3.selectAll('.checkbox').on('change', update);
```

#### 结论分析与体会：

我在本次实验中通过事件驱动将数据录入与可视化联动起来，显著提升教学实验的直观性与互动性。我注意到简洁的数据契约（首行/首列为维度，交叉为度量）有利于约束输入、降低错误率，统一不同模型的标准。同时，在复杂场景可以加入更明确的校验反馈。d3 有强大的可组合性，后续可扩展堆叠柱状、折线、热力图，并加入轴标题、缩放、tooltip 等交互增强用户体验。