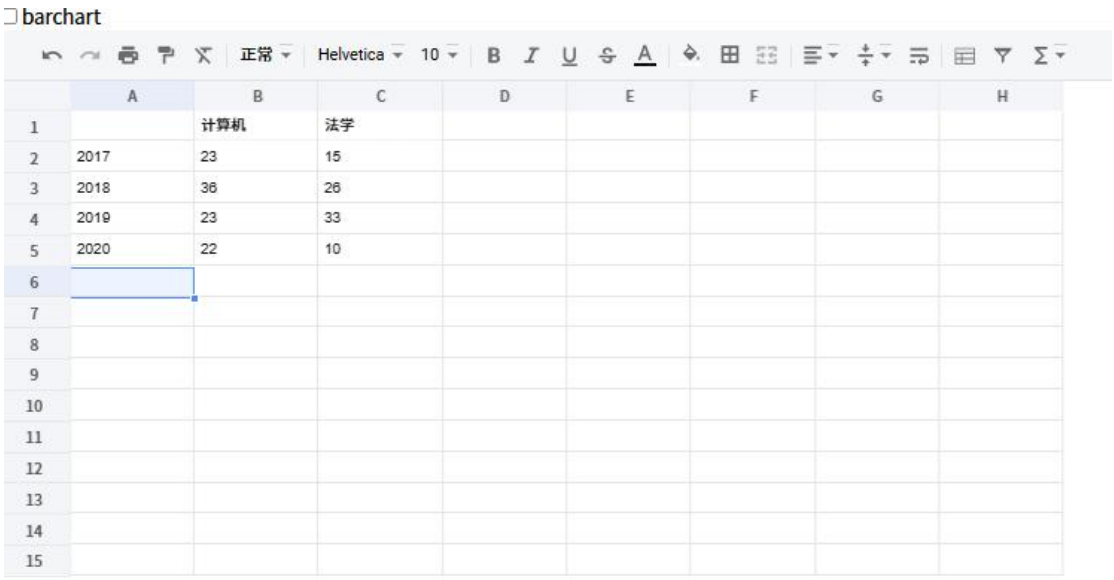
实验三

barchart



	A	B	C	D	E	F	G	H
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								
12								
13								
14								
15								

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>电子表格实践 I</title>
    <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>
    <style>
        #xspreadsheet {
            width: 400px;
            height: 500px;
            padding: 0px;
            margin: 0px;
        }
        #my_dataviz {
            width: 1000px;
            height: 900px;
            padding: 0px;
            margin: 0px;
        }
        .tick text {
            font-size: 20px;
            stroke: black;
            stroke-width: 0.05em;
        }
```

```
    </style>
</head>
<body>
    <div id="xspreadsheet">
        <input type="checkbox" class="checkbox" value="barchart" /><label>barchart</label>
    </div>
    <div id="my_dataviz"></div>

    <script>
        x_spreadsheet.locale("zh-cn");
        var xs = x_spreadsheet("#xspreadsheet", {
            mode: 'edit',
            showToolbar: true,
            showGrid: true,
            showContextmenu: true,
            view: {
                height: () => document.documentElement.clientHeight,
                width: () => document.documentElement.clientWidth,
            },
            row: {
                len: 15,
                height: 25,
            },
            col: {
                len: 8,
                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: 10,
                    bold: false,
                    italic: false,
                },
            },
```

```
});

// 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];
}

function update() {
    const a = d3.select('.checkbox');
    if (a.property("checked")) {
        var data = [];
        var ytitle = [];
        var xtitle = [];
        var col = 0;
        var rows = 0;

        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 ||
                    isNaN(+xs.cell(i, j).text)
                ) {
                    console.log(i, j, xs.cell(i, j));
                    return;
                } else {
                    data[i - 1][j - 1] = +xs.cell(i, j).text;
                }
            }
        }

        window.localStorage.data = data;
        window.localStorage.xTitle = xtitle;
        window.localStorage.yTitle = ytitle;
        console.log(window.localStorage.data);

        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;

        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);
}

var max = 0;
var data = [];
for (var i = 0; i < yTitle.length; i++) {
        var jsd = {};
        jsd["group"] = yTitle[i];
        data.push(jsd);
}

for (var i = 0; i < yTitle.length; i++) {
        for (var j = 0; j < xTitle.length; j++) {
            if (data1[i][j] > max)
                max = data1[i][j];
            data[i][xTitle[j]] = data1[i][j];
        }
}

console.log(data);
console.log(max);

const margin = { top: 40, right: 30, bottom: 40, left: 50 },
        width = 600 - margin.left - margin.right,
        height = 500 - margin.top - margin.bottom;

d3.selectAll('svg').remove();
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 subgroups = xTitle;
const groups = yTitle;

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, max]).range([height, 0]).nice();
svg.append("g").call(d3.axisLeft(y));

const color = d3
        .scaleOrdinal()
        .domain(subgroups)
        .range(["#FF9999", "#0099CC", "#99CC00"]);

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

svg
        .append("g")
        .selectAll("g")
        .data(data)
        .join("g")
        .attr("class", "bar")
        .attr("transform", (d) => `translate(${x(d.group)}, 0)`)
        .selectAll("rect")
        .data(function (d) {
                return subgroups.map(function (key) {
                        return { 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", function (d, i) { return getColor(i) });

// 添加数据标签
svg
        .append("g")
        .selectAll("g")
        .data(data)
        .join("g")
        .attr("class", "bar")
        .attr("transform", (d) => `translate(${x(d.group)}, 0)`)
```

```
        .selectAll("text")
        .data(function (d) {
            return subgroups.map(function (key) {
                return { key: key, value: d[key] };
            });
        })
        .join("text")
        .attr("x", (d) => xSubgroup(d.key) + xSubgroup.bandwidth() * 0.5)
        .attr("y", (d) => y(d.value) - 10)
        .text(d => d.value)
        .attr('text-anchor', 'middle');

// 添加图例
var data_legend = [];
for (var i = 0; i < xTitle.length; i++) {
        var jsd = {};
        jsd["name"] = xTitle[i];
        data_legend.push(jsd);
}
for (var i = 0; i < xTitle.length; i++) {
        data_legend[i]["color"] = getColor(i);
}

var legend = svg
        .selectAll(".legend")
        .data(data_legend)
        .enter()
        .append("g")
        .attr("class", "legend")
        .attr("transform", function (d, i) {
            return "translate(30," + (i * 20 + 120) + ")";
        });

legend
        .append("rect")
        .attr("x", width - 25)
        .attr("y", 8)
        .attr("width", 40)
        .attr("height", 5)
        .style("fill", function (d) {
            return d.color;
        });

legend
```

```
                        .append("text")
                        .attr("x", width + 20)
                        .attr("y", 15)
                        .style("text-anchor", "end")
                        .text(function (d) {
                              return d.name;
                        });
                } else {
                      d3.selectAll('svg').remove();
                }
          }

          d3.selectAll(".checkbox").on("change", update);
    </script>
</body>
</html>
```