

| 评分标准 | 评分 |
|-----------------------------|----|
| (1) 真实性和可用性：数据是真实可用的、具有可用性； | |
| (2) 感知性和可行性：易于感知，方案可行； | |
| (3) 艺术性和设计性：布局合理、数据展示美观； | |
| (4) 原创性和有价值：实用性、内容新颖详实； | |
| (5) 论文内容合理、书写工整。 | |
| 总分 | |

题目： MLB 打击数据的可视化与交互

学号：20123112 姓名：孙梁梁

摘要：本项目从Kaggle平台获取美国职业棒球大联盟（MLB）从1960年到2015年的数据，全部基于前端，使用Java Script进行数据读入和处理，使用ECharts进行可视化展现与交互，成果为HTML网页。网页通过柱状图、折线图、散点图等图表清晰直观地展示了MLB的打击数据，可以帮助爱好者及球队工作人员了解球队的发展情况和定位。

关键词：Kaggle、美国职业棒球大联盟、前端、ECharts、可视化、交互

1 引言

随着互联网的发展，越来越多的人因为使用互联网而产生了海量的数据。这些海量的数据难以进行人工分析，需要借助计算机和可视化的帮助。因此，数据可视化是计算机技术中很重要的一环。本项目通过数据可视化对共计 47249 条 MLB 打击数据进行分析。

经过搜索，发现棒球作为一项在中国较为冷门的小众运动并没有网站能够直观地展现每年或者每个队伍的数据。因此，本项目选取了棒球运动员最高平台的 MLB 的打击数据进行了可视化与交互。项目基于 Java Script 和 HTML 进行开发，适用于 Chrome 和 Edge 平台。使用者可以点击 site 文件夹中的 index.html 进入目录网页，在右上角上传 batting.xlsx 后，将会显示图表。batting.xlsx 内的数据已进行过人工筛选去除缺少详细数据的部分。上传的数据将会在进行处理后通过 localStorage 进行网页间传输和读取。使用者可以点击顶部的“Home”、“Year”和“Team”进行网页跳转。

在“Year”页面内，使用者可以通过输入年份后点击“输入”按钮更改图表展示的年份。在“Team”页面内，使用者可以通过选择队伍更改图表展示的队伍数据。本项目也支持在“Home”页面中上传格式相同的自制数据集。

2 MLB 打击数据可视化的设计和实现

图 1 展示了可视化是常规流程，本文遵循改流程进行设计。

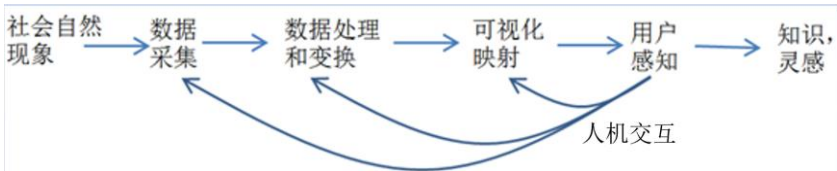


图 1. 可视化常规流程

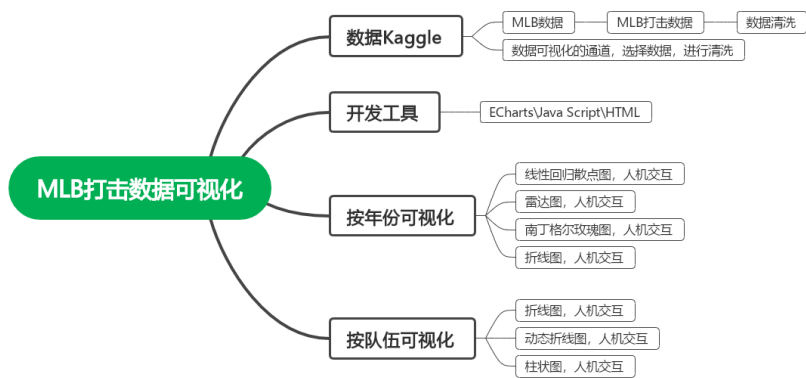


图 2. 系统设计的思维导图



图 3. Home 的原型页面



图 5. Year 的原型页面

2.1 源代码存放地址为：

<https://github.com/hisunll/dataVisualize>

2.2 数据 Kaggle:

本平台数据皆来自于 Kaggle 平台。

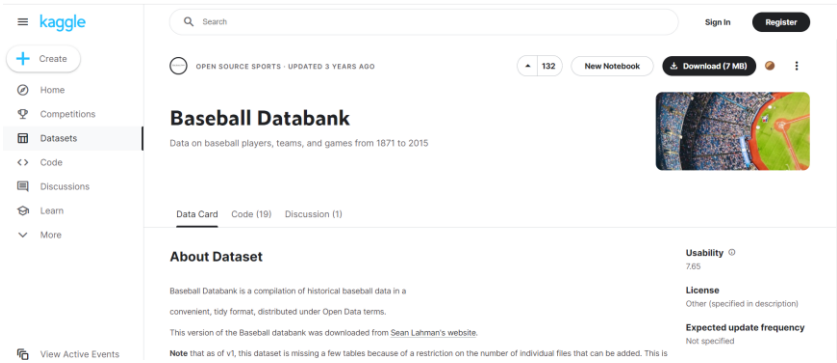


图 6. MLB 打击数据来源网站

上传文件后，将数据读入双层 Map: yearData 和 teamData 中。其读取 excel 文件的代码如下。

```
$('#excel-file').change(function (e) {  
    var files = e.target.files;  
    var fileReader = new FileReader();  
    fileReader.onload = function (ev) {  
        try {  
            var data = ev.target.result  
            var workbook = XLSX.read(data, {  
                type: 'binary'  
            }) // 以二进制流方式读取得到整份 excel 表格对象  
            persons = []; // 存储获取到的数据  
        } catch (e) {  
            console.log('文件类型不正确');  
            return;  
        }  
        var fromTo = '';  
        for (var sheet in workbook.Sheets) {  
            if (workbook.Sheets.hasOwnProperty(sheet)) {  
                fromTo = workbook.Sheets[sheet]['!ref'];  
                persons = persons.concat(XLSX.utils.sheet_to_json(workbook.Sheets[sheet]));  
                break;  
            }  
        }  
    }  
});
```

读入的 Map 结构如图

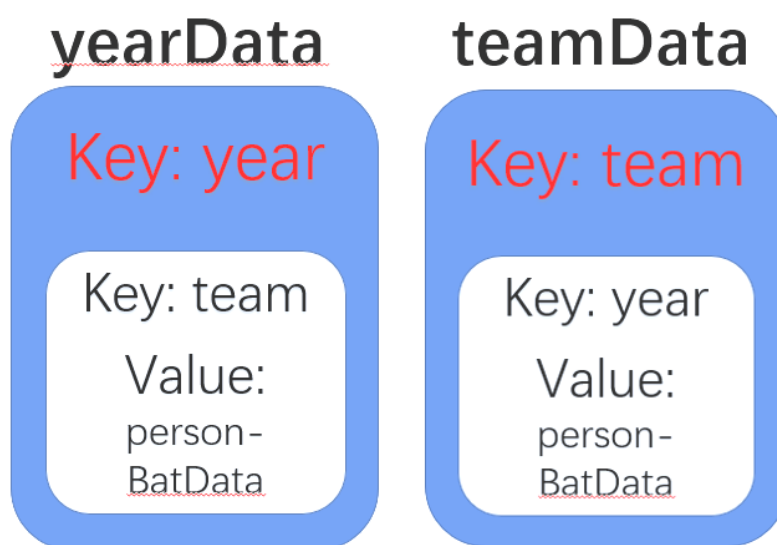


图 7. 双层 Map 结构

由于 Team 数据只需展示不同年份同一队伍的总数据，因此对 teamData 数据进行进一步处理，将 teamData 中的 year 中每个数据进行整合。其代码如下。

```
for(let [teamName, yearBatData] of teamData){  
    var temp = new Map();  
    for(let [year, batData] of yearBatData) {  
        var toSendTeamData = [];  
        toSendTeamData[0] = batData[0][1];  
        for(let i=1;i<11;i++)  
            toSendTeamData[i]=0;  
    }  
}
```

```
for(let i=0;i<batData.length;i++){
  for(let j=1;j<11;j++){
    if(i!= 4)
      toSendTeamData[j] = parseInt(toSendTeamData[j]) +parseInt(batData[i][j+1]);
    else
      toSendTeamData[j] = parseInt(toSendTeamData[j]) +batData[i][j+1];
  }
  temp.set(year,toSendTeamData);
}
toSendTeam.set(teamName, temp);
}
teamData = toSendTeam;
```

页面间数据传送需要将 Map 转换为 Object 后再转为 String，转化后的 String 使用 localStorage 暂存。yearData 的存储部分代码如下。

```
var obj = new Object();
yearData.forEach((item, key, yearData) => {
  var obj2=new Object();
  item.forEach((item2,key2,item) => {
    obj2[key2]=item2
  })
  obj[key]=obj2
})
var setData = JSON.stringify(obj);
localStorage.setItem("key",setData);
```

再指定暂存数据传送到的页面。代码如下

```
window.location.href = 'year.html';
```

读取数据同样使用 localStorage。代码如下。

```
obj = JSON.parse(localStorage.getItem("key"));
yearData = new Map();
for ([key, value] of Object.entries(obj)) {
  var team = new Map()
  for ( [key2, value2] of Object.entries(value))
    team.set(key2,value2)
  yearData.set(key,team)
}
```

2.3 按年份可视化：

2.3.1 年份选择：

通过 input 实现了输入文本框和按钮。用户可以在文本框内输入待查询的年份后点击按钮进行数据查询和图表展示。其关键代码如下。

```
<input id = "toSearchYear" type="text" name = "toSearchYear">
<input id="toSearch"type="button" value = "搜索">
```

对于按钮的判断、输入数据的获取和报错，关键代码如下。

```
document.onclick = function(){
    var obj = event.srcElement;
    if (obj.type == "button"){
        year = document.getElementById("toSearchYear").value
        yearInt=parseInt(year)
        if(year <1965 || year > 2015)
            alert("输入有误")
        else{
            ViewTeamBatData(year);
            ViewTeamRBI(year);
            ViewTeamOtherData(year);
            ViewDay(year);
        }
    }
}
```

2.3.2 线性回归散点图：

本图展现了该年份所有上场打击的选手的打席数量和安打数量，按不同联盟进行线性回归并展示结果以及不同联盟的平均打席与平均安打数，直观地体现了选手的平均水平。数据处理部分将取出所需的值，组成以联盟为 key 的 Map，并将 Map 以联盟分开转换为 Array。代码如下。

```
for(let [team, batData] of yearData.get(nianfen))
    for(let i = 0; i < batData.length; i++){
        var hitNum = batData[i][4];
        var atBatNum = batData[i][3];
        var teamLg = batData[i][1];
        lgHitData.get(teamLg).push([atBatNum, hitNum]);
    }
for (let i=0;i<lgHitData.get("AL").length;i++)
    dataArr.push(lgHitData.get("AL")[i].map(Number));
var dataArr2 = [];
for (let i=0;i<lgHitData.get("NL").length;i++)
    dataArr2.push(lgHitData.get("NL")[i].map(Number));
```

在散点图上显示线性回归结果，需要加载线性回归插件。其代码如下。

```
echarts.registerTransform(ecStat.transform.regression);
```

2.3.3 雷达图：

本图展现了该年份不同队伍的各项打击数据，包括：打率、长打率、本垒打数量、牺牲打数量、四坏上垒次数和盗垒数量。可以直观地展示不同队伍的优势与劣势。

雷达图所需的数据类型为 Array[name: " ",value:[]]。另外为了展现更直观，在数据处理中也计算了各项数据的最大值和最小值。数据求和及转换代码如下。

```
for(let [team, batData] of yearData.get(nianfen)){
    TeamNameLH[num]=team+" ";
    var teamHitSum=0, teamLHSum=0,teamHRSum=0,teamHitRate=0,teamABSum=0;
    var teamSSum=0,teamBBSum=0,teamSBSum=0;
```

```

for(let i = 0; i < batData.length; i++){
    teamHitSum= teamHitSum + parseInt(batData[i][4])
    teamLHSum= teamLHSum+ batData[i][5]
    teamHRSum= teamHRSum + parseInt(batData[i][6])
    teamABSum=teamABSum+parseInt(batData[i][3])
    teamSSum = teamSSum + parseInt(batData[i][11])
    teamSBSum= teamSBSum + parseInt(batData[i][8])
    teamBBSum= teamBBSum + parseInt(batData[i][9])
}
teamHitRate= teamHitSum/teamABSum
teamLHRate= teamLHSum/teamHitSum
toShowData[num] = {name:team+" ",
    value:[teamLHRate,teamHitRate,teamHRSum,teamSSum,teamBBSum,teamSBSum]}
num=num+1;
}

```

为了使数据差距更明显，对展示轴的最大最小值进行了设置。其代码如下。

```

radar: {
    indicator: [
        { text: 'LongHit\nRate', max: maxLHRate,min:minLHRate},
        { text: 'Hit\nRate', max: maxHitRate ,min:minHitRate},
        { text: 'HR', max: maxHR,min:20 },
        { text: 'SF+SH', max: maxS },
        { text: 'Bad\nBall', max: maxBB ,min:100},
        { text: 'Steal\nBase', max: maxSB },
    ]
}

```

2.3.4 南丁格尔玫瑰图：

本图展现了该年份不同队伍的总打点数，直观地体现了不同队伍的打击成果相对好坏。

南丁格尔玫瑰图所需的数据类型与雷达图相似，此处不加赘述。

为了使差距更加明显，对展示的数据进行了减去部分值的操作，并不影响标签上显示值的真实性。其代码如下。

```

data: teamRBI.sort(function (a, b) {
    return a.value - b.value - minRBI*0.9;
})

```

2.3.5 折线图：

本图展现了该年份不同队伍的四坏上垒数、盗垒数和牺牲打数，直观地体现了不同队伍在选取、脚力等小技巧上的不同水平。

折线图的数据要求是对 x 轴传一个作为标签轴的 Array，再对不同线各传一个 Array。数据处理如下。

```

for(let [team, batData] of yearData.get(nianfen)){
    teamSumBB[num] = 0;
    teamSumSB[num] = 0;
    teamSumS[num] = 0;
    teamNameOD[num] = team;
    for(let i = 0; i < batData.length; i++){

```



```
teamSumBB[num] = teamSumBB[num]+parseInt(batData[i][9])
teamSumSB[num] = teamSumSB[num]+parseInt(batData[i][8])
teamSumS[num] = teamSumS[num]+parseInt(batData[i][11])
}
num=num+1;
}
showTeamOtherData(teamNameOD,teamSumBB,teamSumSB,teamSumS)
```

由于数值差距过大，对 y 轴使用了 log 以使折线的变化更明显。代码如下。

```
yAxis: {type: 'log'}
```

2.4 按队伍可视化：

2.4.1 队伍选择：

通过 select 实现了下拉框选择队伍。其代码如下。

```
var AL = document.getElementById("American League");
for (var i = 0; i < ALKey.length; i++) {
    var option = new Option(ALKey[i]+" ", ALKey[i]+" ");
    AL.options[i] = option;
}
<select id="American League" onchange="changeData(this)">American League</select>
```

2.3.6 折线图：

本图展现了该队伍不同年份的打率与总得点的折线图。直观地体现了打率与总得点的关系。

2.3.7 动态折线图：

本图展现了该队伍随时间推移的安打数、长打数、本垒打数、盗垒数和四坏上垒次数的变化。直观地体现了随着时间的变化，队伍各项数据的变化。

动态折线图所需的数据为 Array，为了使变化更明显对 y 轴使用了 log。动画的配置代码如下。

```
animationDuration: 10000
```

2.3.8 柱状图：

本图展现了该队伍随时间推移的安打数、盗垒数与双杀打数量的对比。安打数与双杀打通常成负相关，盗垒数与双杀打关系并不密切。

柱状图所需数据为 Array。

3 MLB 打击数据集和可视化的展示

MLB 打击数据每列由：运动员姓名、年份、服役顺序、队伍、联盟、出场数、打席数、上垒数、安打数、二垒打数、三垒打数、本垒打数、打点、盗垒数、杀盗垒数、四坏上垒数、被三振数、被故意四坏数、触身球数、牺牲触击数、牺牲高飞数和双杀打数组成。

| | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-----------|--------------|-------|--------|------|---|----|---|---|----|----|----|-----|----|----|----|----|-----|-----|----|----|-----|--|--|--|--|--|
| 17 | 20123112 孙梁梁 | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | | | | | |
| playerID | yearID | stint | teamID | lgID | G | AB | R | H | 2B | 3B | HR | RBI | SB | CS | BB | SO | IBB | HBP | SH | SF | GDP | | | | | |
| carlos01 | 1970 | 1 | WS2 | AL | | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| humphbo | 1970 | 1 | WS2 | AL | | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| riddle01 | 1970 | 1 | WS2 | AL | | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| kreutfr01 | 1969 | 1 | WS2 | AL | | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| coxca01 | 1968 | 1 | WS2 | AL | | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| denehbi01 | 1968 | 1 | WS2 | AL | | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| haywobi01 | 1968 | 1 | WS2 | AL | | 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| milesj01 | 1968 | 1 | WS2 | AL | | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| baldwda01 | 1966 | 1 | WS2 | AL | | 4 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| cheneto01 | 1966 | 1 | WS2 | AL | | 3 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | | | | | |
| closta01 | 1966 | 1 | WS2 | AL | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| craigpe01 | 1966 | 1 | WS2 | AL | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| kopliho01 | 1966 | 1 | WS2 | AL | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| narumbu01 | 1966 | 1 | WS2 | AL | | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| durenry01 | 1965 | 2 | WS2 | AL | | 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| mooreba01 | 1965 | 1 | WS2 | AL | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| willhni01 | 1965 | 1 | WS2 | AL | | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| barreaa01 | 2015 | 1 | WAS | NL | | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| carpeda01 | 2015 | 2 | WAS | NL | | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| cedenxa01 | 2015 | 1 | WAS | NL | | 5 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| delosab01 | 2015 | 1 | WAS | NL | | 2 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| gracema01 | 2015 | 1 | WAS | NL | | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| hillta01 | 2015 | 1 | WAS | NL | | 6 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| janssca01 | 2015 | 1 | WAS | NL | | 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| martira04 | 2015 | 1 | WAS | NL | | 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |
| papeljo01 | 2015 | 2 | WAS | NL | | 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | | | |

图 6. MLB 打击数据

3.1 按年份可视化

3.1.1 线性回归散点图：

该图可以进行数据的选择性展示、区域放大缩小和移动鼠标查看具体信息。

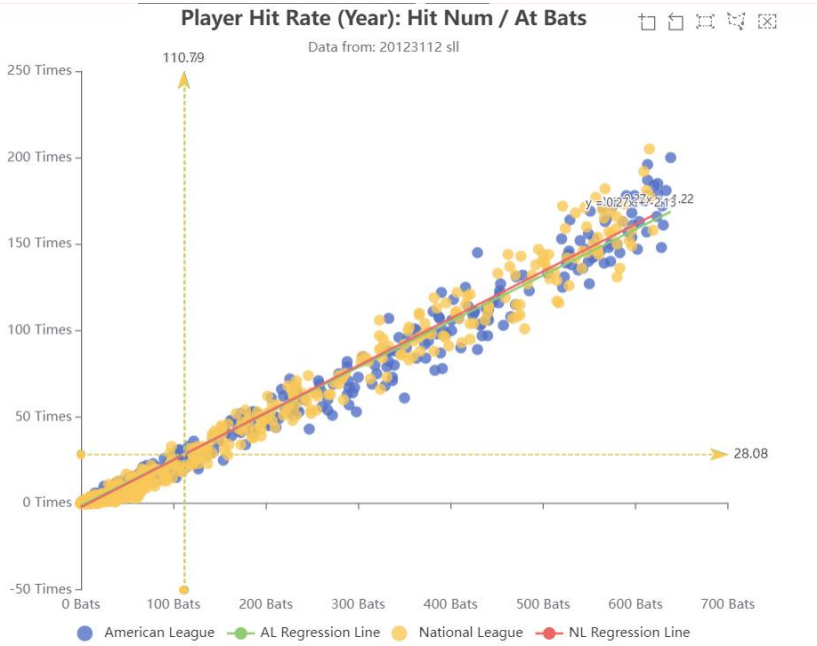


图 7. 安打-打席散点图

3.1.2 雷达图：

该图可以进行数据的选择性展示、数据筛选和移动鼠标查看具体信息。

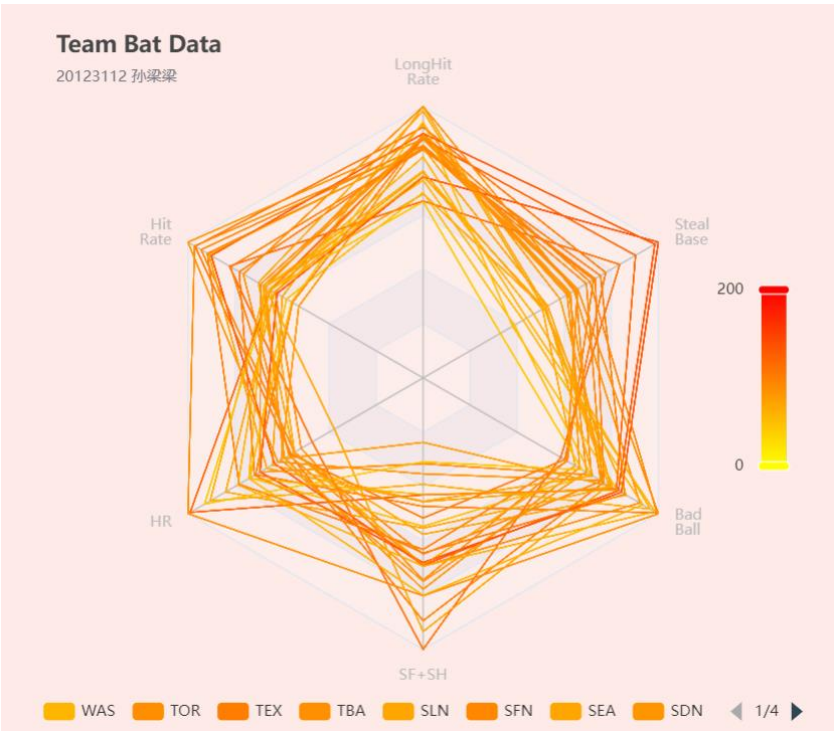


图 8. 各项打击数据雷达图

3.1.3 南丁格尔图：

该图可以移动鼠标查看具体信息。

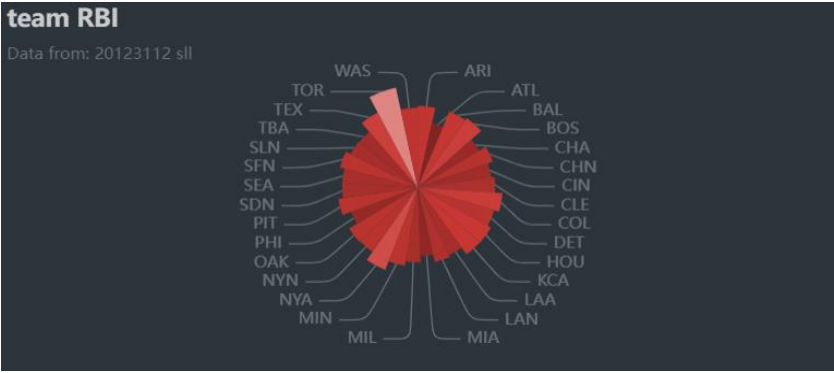


图 8. 得点玫瑰图

3.1.3 折线图：

该图可以进行数据的选择性展示和移动鼠标查看具体信息。

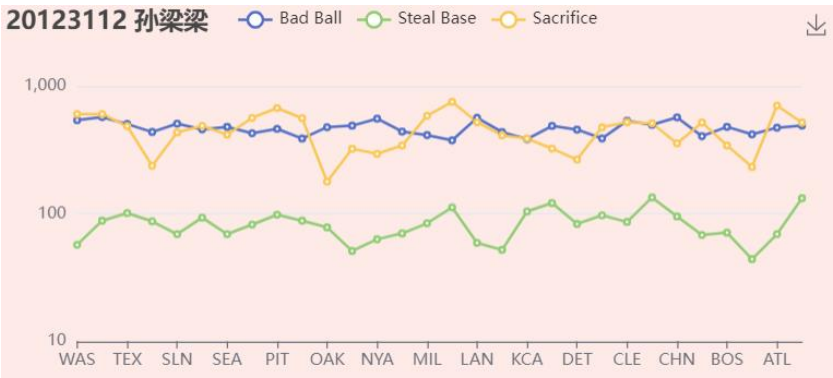


图 9. 四坏、盗垒、牺牲打堆叠折线图

3.2 按队伍可视化

3.2.1 折线图：

该图可以进行数据的区域选择性展示和移动鼠标查看具体信息。

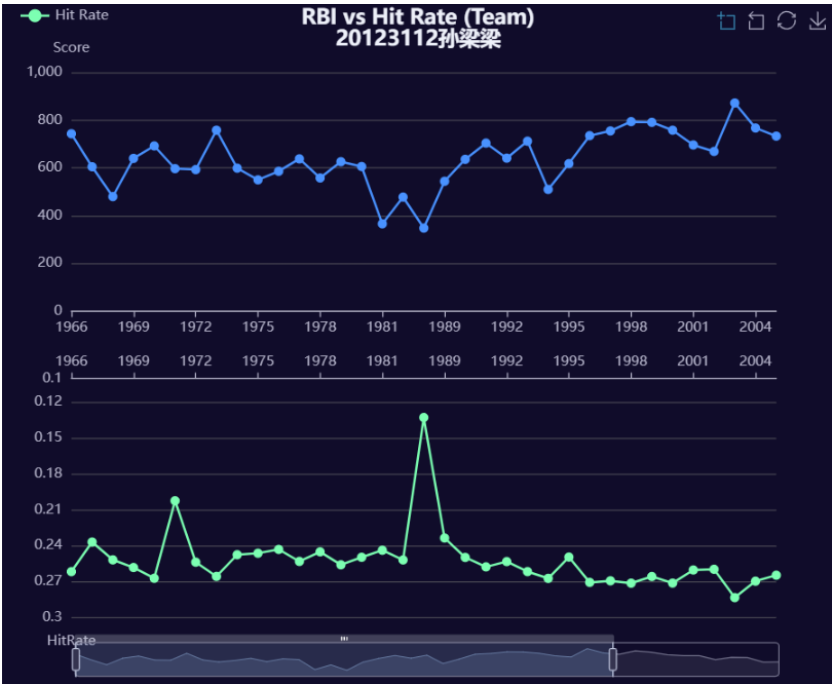


图 10. 得点-打率对比折线图

3.2.2 动态折线图：

该图可以进行数据的选择性展示和移动鼠标查看具体信息。

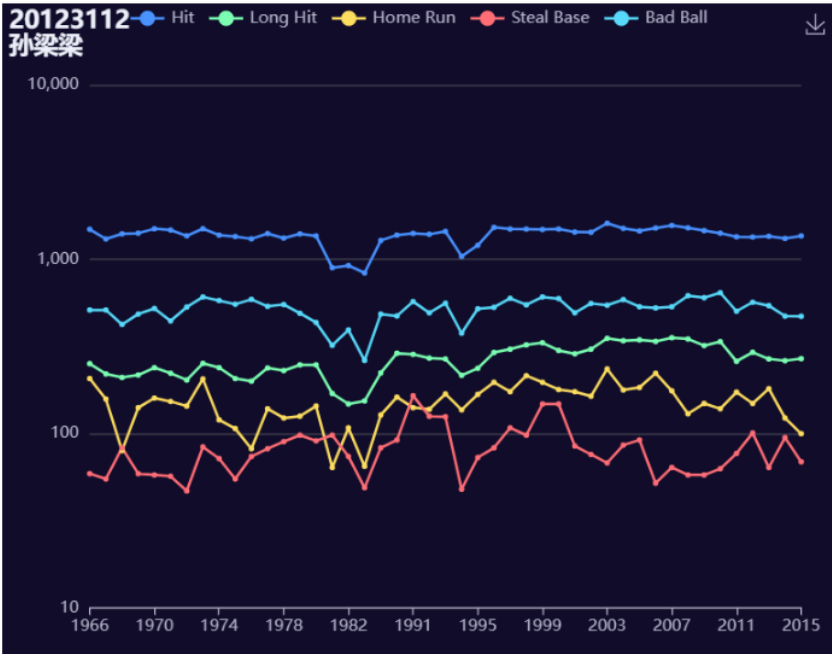


图 12. 各项数据变化动态折线图

3.2.3 动态折线图:

该图可以进行数据的选择性展示和移动鼠标查看具体信息。

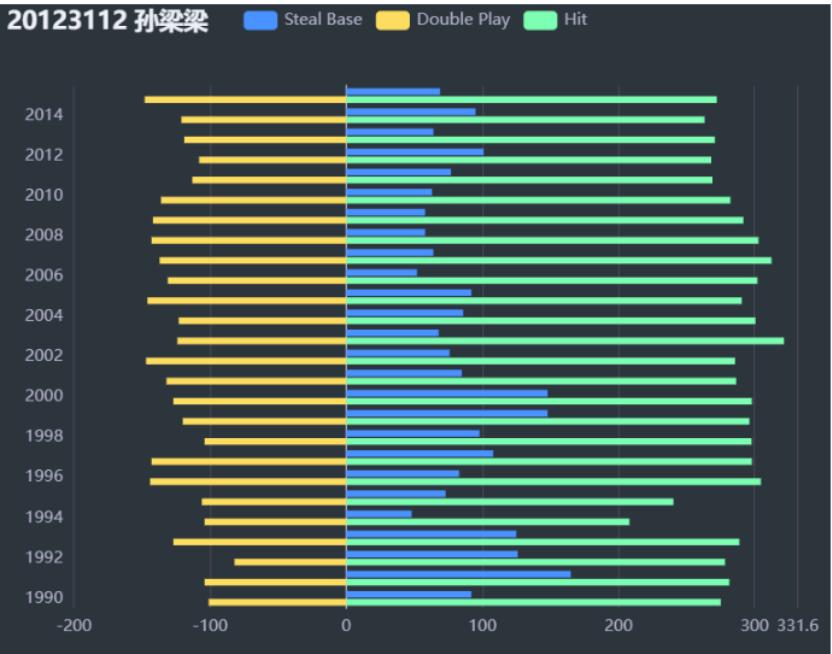


图 12. 安打、盗垒-双杀对比柱状图

4 本设计的特点（例）

（1） 真实性和可用性

本项目所使用的数据集为真实的 MLB 数据，对数据集中存在的空白内容进行了删除。

本项目可以帮助棒球爱好者了解棒球运动整体的打击数据走向、不同队伍的特点以及队伍数据的逐年变化。而对于球队，本项目可以帮助了解自身的定位，近年的发展、制定训练方向和将来的目标。

（2） 感知性和可行性

对于整体数据，使用散点图，很好地体现了上千条数据，而使用其他图表都会产生图表过大，不够直观的问题。

使用雷达图，在比较不同队伍多维度数据上更加直观。另外，采取黄-红的颜色，使过渡自然的同时也足够醒目。

使用南丁格尔玫瑰图而不用饼图是因为存在大于 30 个队伍，柄图的平均角度太小，很难看出数据间的差别。

使用折线图是因为可以更好地展现数据的值。少量数据的对比效果良好。

使用柱状图是因为碍于网页排版，竖排柱状图的展现更清晰。

（3） 艺术性和设计性

对于“Team”和“Year”页面，用户可以选择自己想查看的数据。此外，每一张图都含有交互设计，在图边也含有介绍。用户可以自行上传更新后的数据集。

（4） 原创性和有价值

本项目可以获得每年的平均打率、不同队伍在打击数据的各个项目上以及总体的表现、打率与长打率的关系、双杀与安打的关系、选球能力与打击能力的关系等等。可以帮助帮助棒球爱好者了解棒球运动整体的打击数据走向、不同队伍的特点以及队伍数据的逐年变化。而对于球队，本项目可以帮助了解自身的定位，近年的发展、制定训练方向和将来的目标。比如，如果想要提升打率，可以考虑培养选手的选球能力。

5 结论与展望

本项目实现了适用于 Chrome 和 Edge 平台的 MLB 打击数据可视化。使用者可以点击 site 文件夹

中的 index.html 进入目录网页，在右上角上传 batting.xlsx 后，将会显示图表。batting.xlsx 内的数据已进行过人工筛选去除缺少详细数据的部分。上传的数据将会在进行处理后通过 localStorage 进行网页间传输和读取。使用者可以点击顶部的“Home”、“Year”和“Team”进行网页跳转。

在“Year”页面内，使用者可以通过输入年份后点击“输入”按钮更改图表展示的年份。在“Team”页面内，使用者可以通过选择队伍更改图表展示的队伍数据。对于所有图表，使用者都可以进行交互。本项目也支持在“Home”页面中上传格式相同的自制数据集。

希望在将来，可以增加按选手展示的图表页面以及对投手数据和获奖数据的可视化。也希望在将来可以使用 Vue 框架将本项目改进成前后端分离。