

金陵数据互动可视化

1. 实验环境

1.1 安装 python

- a) 下载地址: <https://www.python.org/downloads/release/>

1.2 语言选择

- a. Python: 用于词云生成及数据表格的格式转换以生成所需的 csv 文件
b. Html、js、css: 用于可视化网页的制作以及交互功能的实现

2. 数据集分析

2.1 数据下载

- a) 从 ChinaVis2023 可视化项目比赛官网 <https://chinavis.org/2023/challenge.html> 上下载金陵数据集

作品说明文档推荐模板和数据集下载

(1) 赛道1: 「数析时空」城市路口多维数据可视化

- 数据字段解释 (2023年5月16日更新) [Download](#)
- 作品文档答卷模板 [Download](#)
- 数据下载链接 <https://pan.baidu.com/s/1NZbHKzpywiNXIGk>
- 数据Q&A [Download](#)

(2) 赛道2: 「数观千年」人文可视化

- 金陵文脉数据集下载链接 [Download](#)

- b) 通过设置条件筛选, 在“金陵数据.xlsx”文件中的地点表中找出符合条件的数据:
地名 id、地名、繁体地名、别名、地名类别、景观类型、地理位置、朝代、时期、
经度、纬度、相关作品、相关人物、相关地点、相关事件、地名描述

其中要确保没有经纬度、地点等

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	地名id	地名	繁体地名	别名	地名类别	景观类型	地理位置	朝代	时期	经度	纬度	相关作品	相关人物	相关地点	相关事件	地名描述	
																	李白于唐代集空位

需要被选取的主要空单元格没有空。

c) 数据集分析

金陵数据集中共有四个数据表，分别是：作品表、人物表、事件表、地点表。

在作品表中，有以下几项数据：作品 id、作品名、人物 id、名字、朝代、时期、国家、文体、写作时间、创作背景、内容简介、相关作品、相关人物、相关地点、相关事件、诗歌全文。

在人物表中，有以下几项数据：作品 id、作品名、名字、朝代、时期、生年、卒年、基本信息、文学身份、生平事迹、文学作品、文学成就_古人评价、文学成就_今人评价、所著作品、相关作品、相关人物、相关地点、相关事件、作品数。

在事件表中，有以下几项数据：事件 id、事件名称、朝代、时期、年号、年份、事件简介、事件内容、检索词、事件类别、特征类别、一级地点、二级地点、相关人物、相关作品。

在地点表中，有以下几项数据：地名 id、地名、繁体地名、别名、地名类别、景观类别、地理位置、朝代、时期、经度、纬度、相关作品、相关人物、相关地点、相关事件、地名描述。

在这个数据集中，本项目需要筛选的数据有以下几项：地名 id、地名、地名描述、相关事件、相关作品、朝代、经度、纬度等。我们期望通过获取地点的经纬度将其标注在地图上，然后分别记录该地点在不同朝代发生过的相关事件以及出现的相关作品，将其展现在地图上。

3. 可视化实例代码分析

3.1 抓取 csv 文件中的数据

```
d3.csv("./src/data.csv").then(function(data) {
  data.forEach(function(d) {
    var location = {
      dynasty: d.朝代,
      name: d.地名,
      longitude: parseFloat(d.经度),
      latitude: parseFloat(d.纬度),
      description: d.地名描述,
      relatedEvent: d.相关事件,
      relatedWorks: d.相关作品,
    };
    locations.push(location);
    //console.log(locations);
  });
});
```

3.2 创造朝代刷

```

var brush = d3.brushX()
    .extent([[0, 0], [1700, 20]]) // 请根据实际需求修改坐标范围
    .on("brush", brushed);

var svg = d3.select("body").append("svg")
    .attr("width", 1700)
    .attr("transform", "translate(76, 0)")
    .attr("height", 50);

svg.append("g")
    .attr("class", "x axis")
    .call(xAxis);

svg.append("g")
    .attr("class", "brush")
    .call(brush);

svg.select(".brush")
    .selectAll(".overlay")
    .style("fill", "rgba(138, 109, 53, 0.5)") // 设置填充颜色
    .style("stroke", "yellow") // 设置边框颜色
    .style("cursor", "default"); // 设置鼠标样式

function brushed() {
    var selectedExtent = d3.brushSelection(this);
    if (selectedExtent) {
        var selectedDynastyId = Math.round(x.invert(selectedExtent[0]));
        var dynastyLabels = ["六朝", "唐代", "宋代", "元代", "明代", "清朝", "近代"];

        var selectedDynasty = dynastyLabels[selectedDynastyId]
        updateMap(selectedDynasty);
    }
}
//updateMap('唐代');
}).catch(function(error) {
    console.error(error);
});

```

使刷取到的朝代对应地点能在地图上显现

```

function updateMap(selectedDynasty) {
    var filteredLocations = locations.filter(function(d) { return d.dynasty == selectedDynasty; });
    console.log(filteredLocations);
    map.eachLayer(function(layer) {
        if (layer instanceof L.Marker) {
            map.removeLayer(layer);
        }
    });

    filteredLocations.forEach(function(location) {
        var marker = L.marker([location.latitude, location.longitude])
            .addTo(map)
            .bindPopup(location.name)
            .on('click', function() { showDescription(location); })
            .openPopup();
    });
}

```

3.2 创造地点描述容器

```
function showDescription(location) {
    var description = location.description;
    var name = location.name;
    var relatedEvent = location.relatedEvent;
    var relatedWorks = location.relatedWorks;

    // 创建描述信息的HTML元素
    var descriptionDiv = document.createElement('div');
    descriptionDiv.style.color = 'black';
    descriptionDiv.style.border = 'none';
    descriptionDiv.style.border = '1px solid transparent';
    descriptionDiv.textContent = '地名: ' + name + ' 地名描述: ' + description;

    // 创建相关事件按钮的HTML元素
    var eventButton = document.createElement('button');
    eventButton.textContent = '相关事件';
    eventButton.style.backgroundColor = 'yellowgreen';
    eventButton.style.border = 'none';
    eventButton.style.color = 'white';
    eventButton.style.fontSize = '14px';
    eventButton.addEventListener('click', function() { showRelatedEvent(relatedEvent); });

    // 创建相关作品按钮的HTML元素
    var worksButton = document.createElement('button');
    worksButton.textContent = '相关作品';
    worksButton.style.backgroundColor = 'yellowgreen';
    worksButton.style.border = 'none';
    worksButton.style.color = 'white';
    worksButton.style.fontSize = '14px';
    worksButton.addEventListener('click', function() { showRelatedWorks(relatedWorks); });
}
```

使点击地图上标点时，会有对应信息显示，并且通过点击相关作品及相关事件能获取该地点在当前朝代下的相关事件及相关作品

3.3 视觉美化

地图选择在线资源，因此需要挂 vpn 才能显示。同时字体也是在线资源，但不需要挂梯子，需要等网页加载一段时间。

```
<title>Map of JinLing</title>
<script src="./src/leaflet/leaflet.js" charset="utf-8"></script>
<link rel="stylesheet" href="./src/leaflet/leaflet.css" />
```

```
<style>
@font-face{font-family:'ZoomlaJianmingLi-A047';/*逐浪新隶体*/
src: url('https://code.z01.com/font/ZoomlaJianmingLi-A047.eot?#iefix'); /* IE9 */
src: url('https://code.z01.com/font/ZoomlaJianmingLi-A047.eot?#iefix') format("embedded-opentype"),
url('https://code.z01.com/font/ZoomlaJianmingLi-A047.woff') format("woff"), /* chrome、firefox */
url('https://code.z01.com/font/ZoomlaJianmingLi-A047.ttf') format("truetype"), /* chrome、firefox、safari */
url('https://code.z01.com/font/ZoomlaJianmingLi-A047.svg#ZoomlaJianmingLi-A047') format("svg");
font-style: normal; font-weight: normal;}
</style>
```

3.4 python 代码

词云生成

```

import numpy as np

from PIL import Image

import re

import jieba

from wordcloud import WordCloud, ImageColorGenerator, STOPWORDS
import matplotlib.pyplot as plt

with open('content.txt', 'r', encoding='utf-8') as f:

    word= (f.read())

    f.close()


image=np.array(Image.open('yuncai.jpg'))

font=r'C:\\Windows\\fonts\\msyh.ttf'

resultword=re.sub("[A-Za-z0-9\\[\\`~\\!\\@\\#\\$\\^\\&\\*\\(\\)\\=\\|\\{\\}\\:\\;\\'\\\"\\[\\]\\.\\<\\>\\|\\?\\~\\. \\@\\#\\|\\&\\*\\%]", "",word)

wordlist_after_jieba = jieba.cut(resultword)

wl_space_split = " ".join(wordlist_after_jieba)


sw = set(STOPWORDS)

sw.add("")

sw.add("")

sw.add("")

```

```

my_wordcloud =
WordCloud(scale=4,font_path=font,mask=image,stopwords=sw,bac
kground_color='white', max_words = 100,max_font_size =
60,random_state=20).generate(wl_space_split)

plt.imshow(my_wordcloud)

plt.axis("off")

plt.show()

my_wordcloud.to_file('wordcloud.jpg')

```

Csv 格式转换与生成

```

import pandas as pd

# 读取 Excel 文件

chaodai = ''

df = pd.read_excel('./total.xlsx', sheet_name='Sheet1')

# 筛选出某一列数据为特定值的行

selected_rows = df[df['朝代'] == chaodai]

# 输出筛选结果

selected_rows.to_csv('./' + chaodai + '.csv', encoding='utf-8', index =
True)

```

4. 成员贡献

具体分工在 `member.html` 中

其中南京民国历史地名演变部分由薛婷文同学独立完成。艾力下提·依沙克完成了地图的交互和网页框架搭建，蒋明杰同学完成了其余数据处理、网页的制作与美化以及实验文档和小组任务分工交流。

唐正同学完成了词云生成以及数据预处理，闫晓迈同学完成了数据预处理及文档数据集分析部分。郭天悦同学整理了南京古代的历史文化数据。