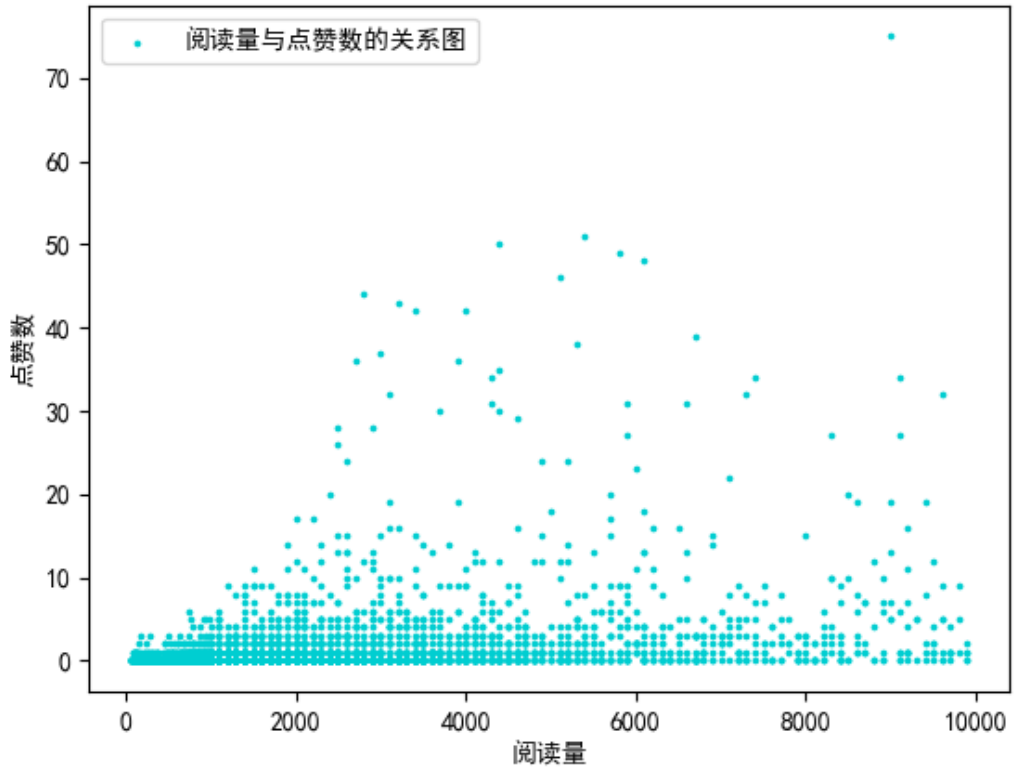
数据分析文档

结论一

阅读量与点赞数没有线性关系，但阅读量越大，文章所获得的平均点赞数越高。



图表说明

大部分文章的点赞数在10以下。但相对来说，阅读量越大，点赞数多的概率更大。从图中可以看出，点赞数在20以上的文章阅读量都超过了2000，而阅读量在2000以下的文章点赞数基本都在10以下。

整体上看，文章点赞数主要集中在0-10的范围内，与阅读量没有明显的线性相关趋势。

代码

```
def fans_read():
    file_list = os.listdir("./data")
    read_list = []
    like_list = []
    for i in range(0, len(file_list)):
        with open(f'./data/{file_list[i]}', 'r', encoding="utf-8") as f:
            data = json.load(f)
        if data['read'][-1] == 'k':
            hot = int(float(data['read'][0:-1]) * 1000)
        else:
            hot = int(data['read'])
        if data['like'][-1] == 'k':
            like = int(float(data['like'][0:-1]) * 1000)
```

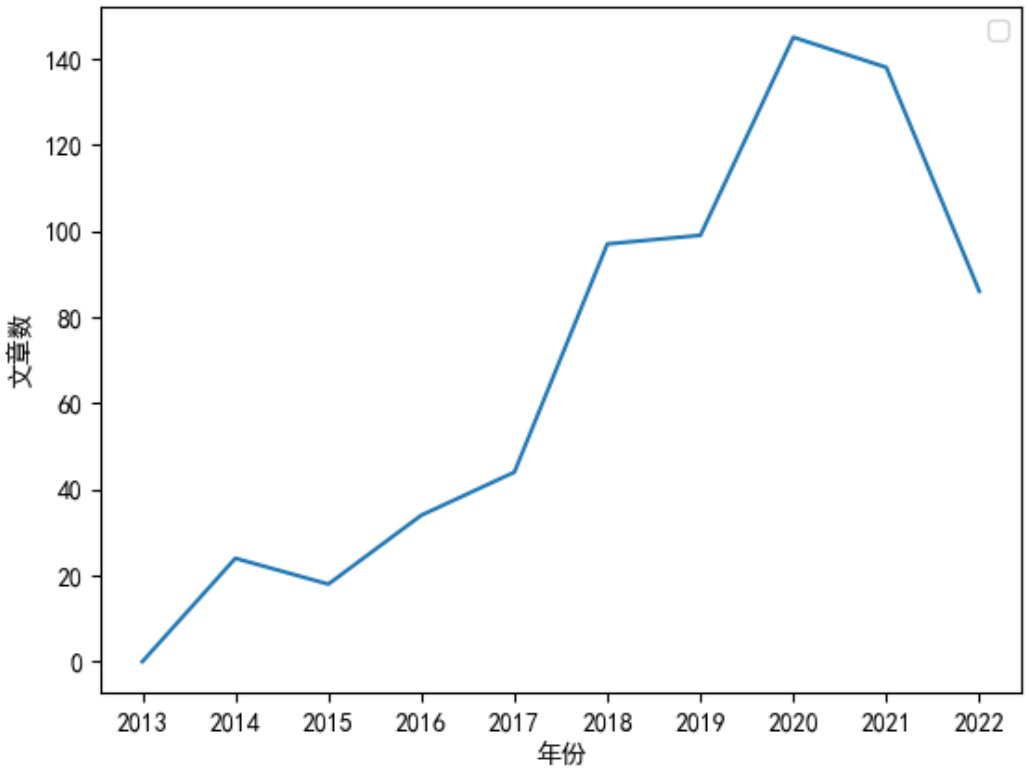
```
            else:
                like = int(data['like'])
            if hot < 10000:
                if like<100:
                    read_list.append(hot)
                    fans_list.append(like)
        print(len(read_list))
        print(len(like_list))
        plt.rcParams['font.sans-serif'] = ['SimHei']
        plt.rcParams['axes.unicode_minus'] = False

        color_dot = '#00CED1'
        area = np.pi * 0.05 ** 0.05  # 点面积
        plt.xlabel('阅读量')
        plt.ylabel('点赞数')

        plt.scatter(read_list, like_list, s=area, c=color_dot, label='阅读量与点赞数的关系图')
        plt.legend()
        plt.show()
```

代码逻辑

先读入所有已爬取的数据，获取每条博客的阅读量和点赞数信息。由于个别博客的点赞数或阅读量过大，造成大部分数据在图表显示过小，所以限制了阅读量的和点赞数的范围，得到要绘制图表的数据。最后设置图标的字体，颜色，点的大小等绘制信息。

结论二

从 2014 —— 2022 年，在论坛上讨论 Python 语言的人越来越多，反映了使用 Python 的人群逐渐增多。



图表说明

从图表中可以看出，从 2013 - 2020 年，在论坛中有关 Python 的文章数目始终处于一个增长趋势，在近2年内文章的数目基本保持不变，处于一个较多的水平，这说明有关 Python 的使用情况、出现的问题有更多的人在讨论，反映了有更多的人在使用这种编程语言。

2022 年出现的文章数目下降应该与时间有关，2022 年还没有结束，根据今年所剩 3 个月的时间估算，2022 年的文章数目与 2021 年大致相同。

代码

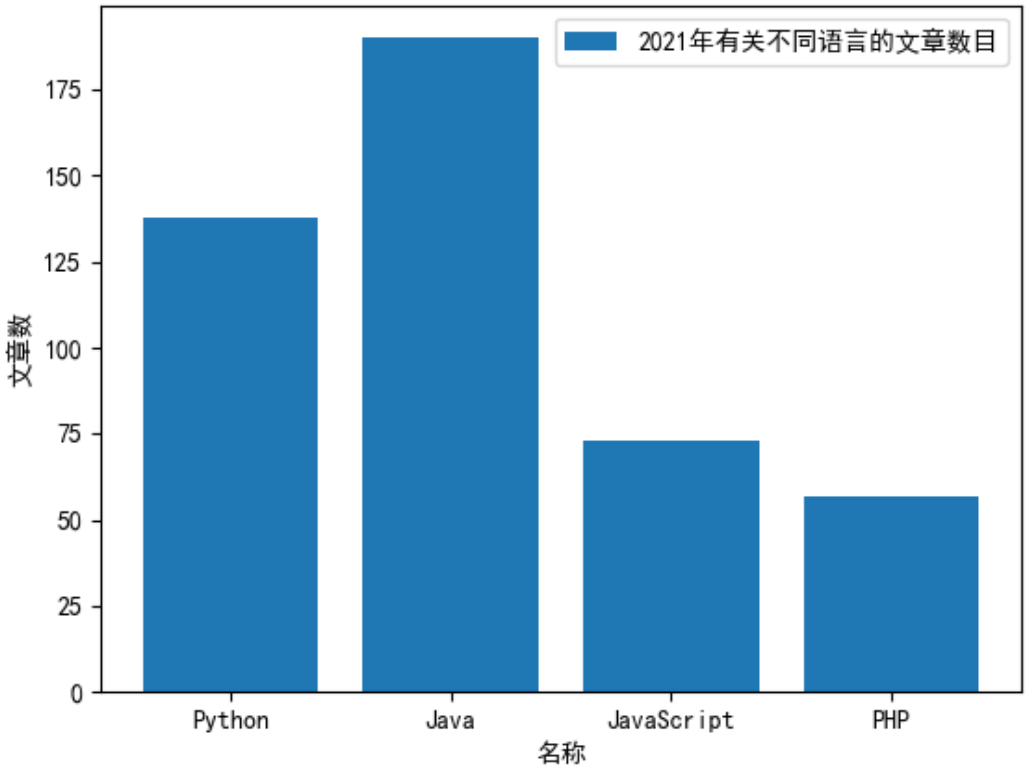
```
def python_article():
    file_list = os.listdir("./data")
    print(len(file_list))
    dic = {}
    for i in range(0, len(file_list)):
        with open(f'./data/{file_list[i]}', 'r', encoding="utf-8") as f:
            data = json.load(f)
        if data['author_pic'][12:18] == 'python':
            year = data['pub_date'][0:4]
            if year in dic:
                tmp = dic[year]
                dic[year] = tmp + 1
            else:
                dic[year] = 0
    print(dic['2019'])
    year_list = []
    num_list = []
    for y, num in dic.items():
        year_list.append(y)
        num_list.append(num)
    print(year_list)
    print(num_list)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False
    plt.xlabel('年份')
    plt.ylabel('文章数')
    plt.plot(year_list, num_list)
    plt.show()
```

代码逻辑

首先读入数据，判断文章是否有关 Python 语言，再获取文章的发表年份，统计每个年份发布的文章数目。最后设置绘图的字体样式，标签等。

结论3

Python 与 Java 的使用人数相较于 JavaScript 和 PHP 更多。



图表说明

从图表中可以看出，有关 Python 和 Java 的文章数目显著多于有关 JavaScript 和 PHP 的文章数目，这说明前两种语言的使用人数更多，产生了更多的内容分享。

此次分析中所使用的每组语言的数据均在 这组数据与 2021 年编程语言排行基本相符。

代码

```
def get_num(file_list, str):
    dic= {}
    num = 0
    cnt = 0
    for i in range(0, len(file_list)):
        with open(f'./data/{file_list[i]}', 'r', encoding="utf-8") as f:
            data = json.load(f)
        if data['author_pic'][12:12+len(str)] == str:
            num = num + 1
            year = data['pub_date'][0:4]
            if year == '2021':
                print(year)
                cnt = cnt + 1
    return cnt


def compare():
    file_list = os.listdir("./data")
    python_num = get_num(file_list, 'python')
    java_num = get_num(file_list, 'Java')
    javascript_num = get_num(file_list, 'JavaScript')
    php_num = get_num(file_list, 'PHP')
    name_list = ['Python', 'Java', 'JavaScript', 'PHP']
```

```
    num_list = [python_num, java_num, javascript_num, php_num]
    print(num_list)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.rcParams['axes.unicode_minus'] = False
    plt.xlabel('名称')
    plt.ylabel('文章数')
    plt.bar(name_list, num_list,label='2021年有关不同语言的文章数目')
    plt.legend()
    plt.show()
```

代码说明

先读取数据，再利用 get_num 函数获取每种语言类型在 2021 年发布的文章数目，最后设置图标的绘图的字体样式，标签等。

get_num 函数根据传入的 str 参数来查找相应编程语言类型的文章，在筛选 2021 年发布的文章，获取文章数目作为返回值。