

PaperPass旗舰版检测报告

简明打印版

比对结果(相似度):

总 体:	11%	(总体相似度是指本地库、互联网的综合对比结果)
本地库:	8%	(本地库相似度是指论文与学术期刊、学位论文、会议论文、大学生联合比对库、互联网的比对结果)
期刊库:	4%	(期刊库相似度是指论文与学术期刊库的比对结果)
学位库:	4%	(学位库相似度是指论文与学位论文库的比对结果)
会议库:	1%	(会议库相似度是指论文与会议论文库的比对结果)
图书库:	1%	(图书库相似度是指论文与图书库的比对结果)
联合库:	6%	(联合库相似度是指论文与大学生联合比对库的比对结果)
互联网:	4%	(互联网相似度是指论文与互联网资源的比对结果)

报告编号: 6294BC169761092ZP

检测版本: 旗舰版

论文题目: python 百度指数 爬虫 可视化分析

论文作者: 1

论文字数: 9299

段落个数: 255

句子个数: 355

提交时间: 2022-5-30 20:44:06

比对范围: 学术期刊、学位论文、会议论文、书籍数据、大学生联合比对库、互联网资源

查询真伪: <https://www.paperpass.com/check>

句子相似度分布图:



本地库相似资源列表(学术期刊、学位论文、会议论文、书籍数据、大学生联合比对库):

1. 相似度: 5.5%
来源: 大学生联合比对库



互联网相似资源列表:

1. 相似度: 1.1% 标题: 《直方图 - t180互动问答网》
<https://t180.cn/article/166075>
2. 相似度: 0.8% 标题: 《数据可视化的重要性及常见用例 - 简书》
<https://www.jianshu.com/p/a250a8862e55>
3. 相似度: 0.8% 标题: 《只用Python就可以制作的简单词云_p...》
<http://www.cppcns.com/jiaoben/python/399454.html>

基于 Python 的疫情与心理健康的百度指数数据挖掘与可视化分析

摘要：随着新冠继续影响世界，有数据表明，疫情不仅带来经济上的灾难，也带来了人类心理健康上的灾难，越来越多的人，因为疫情防控，被封在家，或者感染后被他人歧视等原因，带来了不小的心理疾病，比如抑郁症，抑郁症的发病数量明显随着疫情的扩大而增高，通过 python 在百度指数上的爬虫对数据挖掘和利用 python 图形库制作可视化的图，可以很清晰地了解到疫情和抑郁症的关系，也将帮助政府和国家对民众心理进行及时的指导和干预。

Data mining and visual analysis of Baidu Index on epidemic situation and mental health based on Python

Abstract:: As the new crown continues to affect the world, some data show that the epidemic has not only brought about economic disasters,  but also brought about disasters in human mental health. More and more people have brought about many psychological diseases, such as depression, because of epidemic prevention and control,  being closed at home, or being discriminated against by

others after infection. The incidence of depression has obviously increased with the expansion of the epidemic, We can clearly understand the relationship between epidemic and depression through data mining by Python crawlers on Baidu Index and making visual graphs by using Python graphics library.

0 引言

51% 随着互联网技术的高速发展，大数据时代的来临，数据规模极具膨胀，但数据存在着难以高效率地获取和分析的挑战。数据科学，俨然成了关乎科技进步的核心基础设施。在国外，著名的搜索引擎谷歌利用大数据分析并预测人们在搜索框内的输入，在国内，像购物 app 淘宝、娱乐 app 抖音，都广泛地利用大数据建立每个用户的个人画像，52% 推送更被人们喜欢的广告或短视频。数据科学的发展，给人们的生活带来了极大的便利；数据科学，已经得到了人民的认可，也得到了政府和党的大力支持，习近平总书记强调，51% 国家要大力实施大数据战略，发展数字经济和数字中国的建设。

目前，鉴于新冠疫情的严峻形式，疫情防控依然是我国的首要工作之一，根据全球著名医学杂志《柳叶刀》的最新报告（Global prevalence and burden of depressive and 52% anxiety disorders in 204 countries and territories in 2020 due to the COVID-19 pandemic）显示，疫情带给人们不仅是经济上的损失，更多的是身心健康上的损害。在政府给予民众必要的生活物资的同时，还需要更多地关心民众在身

心健康上的问题，并给予及时和有效的治疗。如何获取疫情与身心健康的相关信息？每日的新冠阳性数量的增加是否和抑郁症数量的增加呈正相关？急需心理治疗的人数是否因为疫情有所增加？

42%
网络爬虫是一种高效地从互联网获取数据的技术手段，依据几个特定的关键词，可以爬取指定日期、指定地区，甚至特定的性格人群的计算机程序。利用网络爬虫，依据不同的需求，制定相应的程序规则，自动地爬取想要的信息，并将数据持久化地保存在计算机上。

本文以百度指数为爬取平台，制定的关键词有“新增阳性”、“抑郁症”、“地区”、“抑郁症治疗”、“疫情”等，基于 python 语言设计的一套网络爬虫程序，完成对目标关键词百度指数信息的获取，存储数据后，再通过 python 的数据库和绘图库对数据做量化分析和模型的建立，并以可视化的形式更直观地展现出来。

53%
综上所述，我们可以利用爬虫技术获取疫情与心理健康的数据，54%
并利用数据分析技术建立一套相关性的模型，给出强用的证据展示疫情和心理疾病的强关联性，帮助政府预测和治疗因为疫情导致的心理疾病的人群。

1 价值

（一）本课题的学术价值

- （1）目前学术界很少人利用百度指数做数据的分析和预测，本课题可以通过数据爬取、分析，建立一套通用的大数

据预测模型。

- (2) 目前学术界很少有人研究疫情与心理疾病的关系，并给出良好的应对措施，本课题可以分析出疫情与心理疾病变化的关系。

(二) 本课题的应用价值

- (1) 44% 基于百度指数爬取各种的数据信息，建立疫情风险预测模型。
- (2) 基于百度指数的爬取的数据信息，建立疫情与心理疾病的防范机制。

2 课题研究对象

54% 本课题的研究对象是疫情与心理健康的关系。随着疫情的扩大，心理健康出现问题的人数是否相应的增多，以及需要接受心理治疗的人数的变化关系。

3 主要内容

3.1 python 介绍

87% Python 是一种具有动态语义的解释型、面向对象的高级编程语言。它的高级内置数据结构，结合动态类型和动态绑定，使其对快速应用程序开发非常有吸引力，也可以用作脚本或胶水语言将现有组件连接在一起。Python 简单易学的语法强调可读性，因此降低了程序维护

的成本。Python 支持模块和包，这鼓励程序模块化和代码重用。

3.2 爬虫介绍

3.2.1 爬虫原理

^{68%} 网络爬虫：网络爬虫（也称为网络蜘蛛，网络机器人），它是一个经验法则，自动从万维网上抓取信息的程序或脚本。^{54%} 一个请求网站并提取数据的自动化程序，可以理解为蜘蛛在互联网上爬行，^{58%} 互联网可以比作一张大网，爬虫在这个大网上爬行，遇到一些有趣的网站资源，可以模拟一个浏览器并抓取它，然后将其放入 CSV 数据库等。

3.2.2 爬虫分类

^{77%} 根据实现的技术和结构，网络爬虫可以分为一般网络爬虫、聚焦网络爬虫、增量网络爬虫和深度网络爬虫。

3.3.3 爬虫的基本工作流程

(1) 获取初始 URL。^{41%} 初始 URL 是网络爬虫的入口点，链接到需要爬取的网页。

(2) 在抓取网页时，我们需要获取页面的 HTML 内容，然后对其进行解析以获取链接到该页面的所有页面的 URL。

(3) 将这些 URL 放入队列中。

(4) 循环遍历队列，从队列中逐一读取 URL，对于每个 URL，爬取对应的网页，然后重复上面的爬取过程。

64%

(5) 检查是否满足停止条件。如果没有设置停止条件，爬虫会一直爬到无法获取到新的 URL。

3.3 数据挖掘

3.3.1 获取 COOKIE

百度指数需要登录才能爬取，它使用 COOKIE 进行身份验证，所以需要先获取 COOKIE，打开 Chrome 浏览器，访问百度指数的首页并登录，按 F12 调出开发者模式，点击状态栏的“网络”，并勾选保留日志的功能，选中 "Fetch/XHR"，再次刷新网页，找到其中的 GET 请求，其中某些请求包含 COOKIE 参数，点击复制 COOKIE 的 value

3.3.2.导入第三方相关库

```
import time, random, json, pandas as pd
from qdata.baidu_index import get_search_index
from qdata.baidu_index.common import split_keywords
```

3.3.3 准备 cookie 和关键词

```
keywords_list = [["抑郁症"], ["疫情"]]
cookies = ""
```

3.3.4 核心代码如下

```
63%  
def spider(keywords_list, start_date, end_date, cookies):  
  
    print("爬虫开始工作")  
  
    all_list = []  
  
    # 计时  
  
    tic = time.time()  
  
    # 遍历关键词  
  
    for keywords in split_keywords(keywords_list):  
  
        for index in get_search_index(  
  
            keywords_list=keywords,  
  
            start_date=start_date,  
  
            end_date=end_date,  
  
            cookies=cookies,  
  
        ):  
  
            if index["type"] == "all":  
  
                s = {  
  
                    "keyword": index["keyword"][0],  
  
                    "date": index["date"],  
  
                    "index_num": int(index["index"]),  
  
                }  
  
                all_list.append(s)  
  
                print("正在爬取： %s" % index)
```



```

        # 随机睡眠, 防止被屏蔽

        time.sleep(random.uniform(3, 5))

    toc = time.time()

    shijian = toc - tic

    print("耗时%.2f 秒,爬取完成! \n 开始写入 json"%shijian)

```

3.3.5 将数据写入 json 文件中

```

with open("temp.json", "w") as f:

    f.write(json.dumps(all_list))

print("写入完成")

```

3.3.6 将数据写入 excle 文件中

```

print("json 转 csv")

# 写入 csv

df = pd.read_json(".\\data\\temp.json")

df.to_csv(".\\data\\temp.csv", index=None)

print("转 csv 完成! ")

```

3.4 数据处理

爬取好的数据已经按照日期依次保存 keyword、date 、index 的 key 和 value 在 json 数组的文件中, 由于我们的时间范围跨度过大, 从疫

情开始到现在，所以我们需要将数据合并处理，以月份为单位合并的代码如下：

```
def analyse(keywords_length):
    print("开始分析数据")
    df = pd.read_csv(".\\data\\temp.csv")
    # keyword 分组 合算 每天到每月 的 index_num 数值，并重新将分组后的数据放入新的 dataframe 中
    df["date"] = pd.to_datetime(df["date"])
    df
    =
df.set_index("date").groupby("keyword").resample("m").sum().reset_index()

# 保存
df.to_csv(".\\data\\sum.csv")

# 转化为 list
list_df = pd.read_csv(".\\data\\sum.csv")
index_list_df = list_df.groupby("keyword")["index_num"].apply(list)
# 保存为 json
index_list = []

for i in range(keywords_length):
    index_json = {
```

```

        "keyword": index_list_df.index[i],
        "index_sum": index_list_df.values[i],
    }
    index_list.append(index_json)

with open(".\\data\\index_list.json", "w") as f:
    f.write(json.dumps(index_list))

print("分析完成！ ")

os.remove(".\\data\\temp.csv")
os.remove(".\\data\\temp.json")

```

3.4 数据可视化

79% 数据可视化是将信息转换为视觉环境（如地图或图形）的实践，以使数据更易于人脑理解和获取见解。数据可视化的主要目标是更容易识别大型数据集中的模式、趋势和异常值。

可视化之前，我们首先需要引入必须的第三方库：

```

import matplotlib.pyplot as plt
import sys, json, pandas as pd
import numpy as np

sys.path.append(".")

```

59% 然后将百度指数数据放入一个数组中：

```
def get_index_list(keyword):
    with open(".\\data\\index_list.json", "r") as f:
        date_list_json = json.load(f)

    for i in range(len(keyword)):
        if date_list_json[i]["keyword"] == keyword:
            return date_list_json[i]["index_sum"]
```

3.4.1 折线图

折线图是在直线上绘制数据点的一种方法。通常，它用于显示趋势数据，或两个数据集的比较。

实现折线图的代码如下：

```
def draw_mulite_line(list):
    print("开始绘制折线多图")
    date_list = get_date_list()
    x = 1
    for i in list:
        plt.rcParams['font.sans-serif']=['Microsoft YaHei']
        plt.xlabel("时间")
        plt.ylabel("指数")
        plt.subplot(1,2,x)
        x=x+1
```

```

plt.xticks(rotation=30)

plt.plot(date_list, get_index_list(i))

plt.grid()

plt.title("%s"%i)

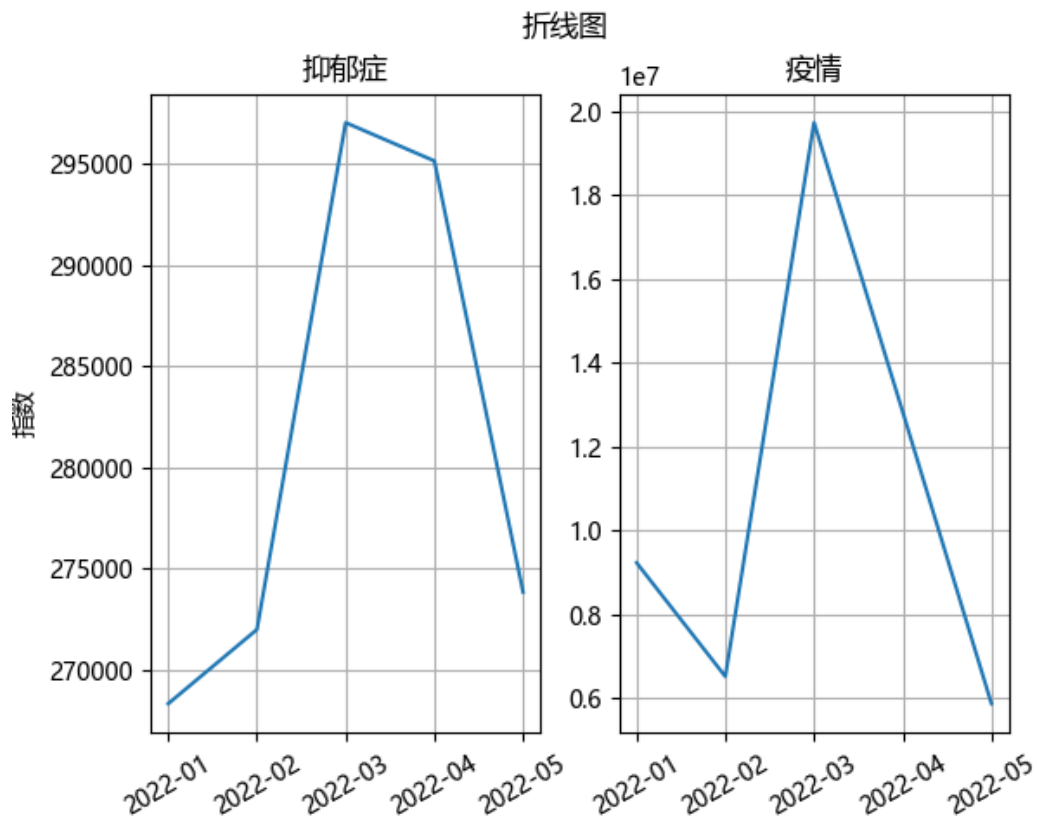
plt.suptitle("折线图")

plt.savefig('.\\data\\mulite_line.png')

plt.close()

print("绘制完毕!")

```



42%

很明显地看出， 抑郁症的发病率和疫情热度呈正相关

3.4.2 柱状图

99%

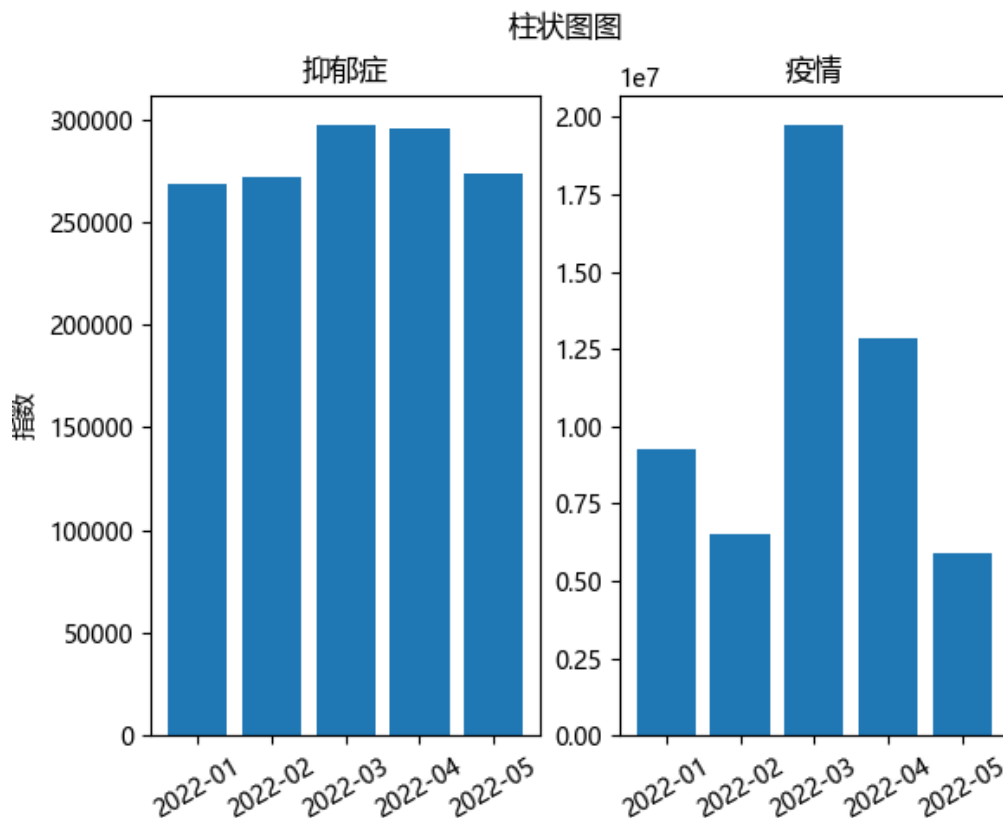
直方图是将一组数据点组织到用户指定范围的图形表示。柱状图在外

观上类似于条形图，它通过获取许多数据点并将它们分组到逻辑范围或数据箱中，将数据序列压缩为易于解释的视觉效果。

实现代码如下：

```
def draw_mulite_bar(list):  
    print("开始绘制柱状多图")  
    x=1  
    for i in list:  
        plt.rcParams['font.sans-serif']=['Microsoft YaHei']  
        plt.xlabel("时间")  
        plt.ylabel("指数")  
        plt.subplot(1,2,x)  
        x=x+1  
        plt.xticks(rotation=30)  
        plt.bar(get_date_list(), get_index_list(i))  
        plt.title("%s"%i)  
        plt.suptitle("柱状图图")  
    plt.savefig('.\\data\\mulite_bar.png')  
    plt.close()  
    print("绘制完毕")
```

效果如下：



在这张图中，抑郁症相对变化不是很明显，是由于其本身的庞大基数造成的

3.4.3 词云

词云（也称为文本云或标签云）的工作方式很简单：特定词在文本数据源（如演讲、博客文章或数据库）中出现的次数越多，它在词云。词云是用不同大小描绘的词的集合或集群。^{51%}单词出现的越大越粗，它在给定文本中被提及的次数越多，它就越重要。也称为标签云或文本云，这些是提取文本数据最相关部分（从博客文章到数据库）的理想方法。它们还可以帮助业务用户比较和对比两段不同的文本，以找出两者之间的措辞相似之处。

代码实现如下：

```
import wordcloud, os, jieba

import numpy as np

from PIL import Image

pwd = os.getcwd()

pic = Image.open(r"C:\Users\fanyq\Desktop\baidu\code\libs\pikaqiu.jpg")

shape = np.array(pic)

wc = wordcloud.WordCloud(
    mask=shape, font_path="simkai.ttf", background_color="white",
    max_font_size=100
)

text = open(pwd + "\\data\\ci.txt", "r", encoding="UTF-8").read()

cut_text = jieba.cut(text)

result = " ".join(cut_text)

wc.generate(result)

wc.to_file(pwd + "\\data\\cloud.jpg")
```

效果如图：



3.4.4 饼图

饼图（或圆图）是一种圆形统计图形，它被划分为多个部分以说明数字比例。在饼图中，每个切片的弧长（以及相应的中心角和面积）与其表示的数量成比例。虽然它的名字是因为它像一个被切成薄片的馅饼，但它的呈现方式也有不同。

实现代码如下：

```
def draw_mulite_pie(list):
```

```
print("开始绘制饼多图")

x=1

for i in list:

    plt.rcParams['font.sans-serif']=['Microsoft YaHei']

    plt.subplot(1,2,x)

    x=x+1

    plt.xticks(rotation=30)

plt.pie( get_index_list(i),labels=get_date_list(),autopct='%.2f%%')

    plt.title("%s"%i)

    plt.suptitle("饼图")

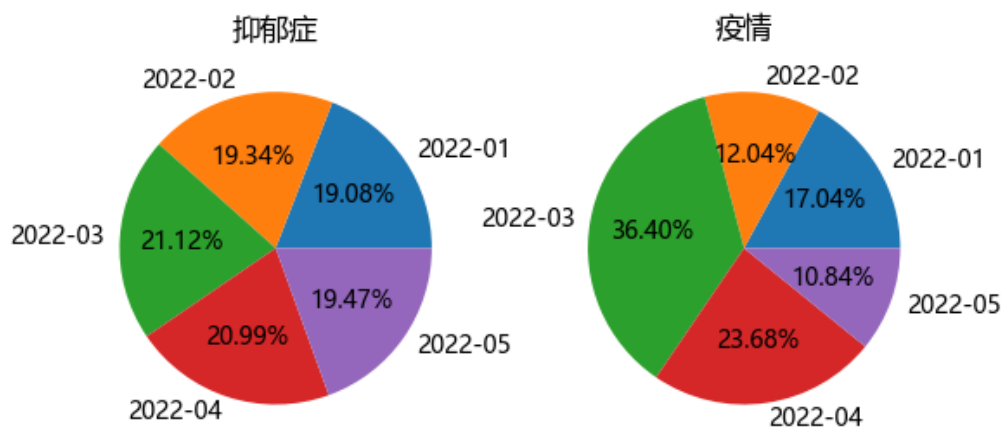
plt.savefig('.\\data\\mulite_pie.png')

plt.close()

print("绘制完毕")
```

效果如下：

饼图



从这张图中，我们能看出每个月百度指数的百分比，由于疫情集中于三月份开始，所以三月份热度占比最大，同理，抑郁症的热度也稍微有所增加。

3.4.5 散点图

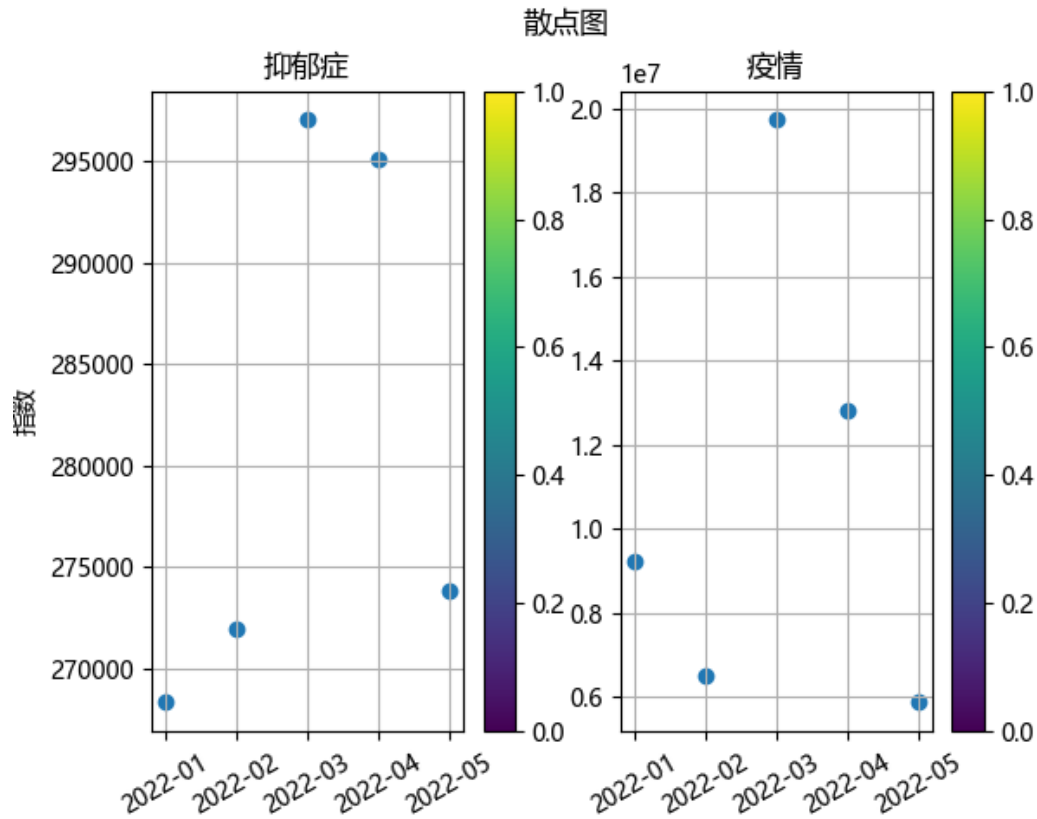
散点图是一组数据的图形表示，其中变量对的值绘制在坐标系上。^{41%}该工具广泛应用于统计学和其他科学与工程领域，用于表示数据关系。

实现代码如下：

```
def draw_mulite_matrix(list):
    print("开始绘制散点多图")
    x=1
    for i in list:
```

```
plt.rcParams['font.sans-serif']=['Microsoft YaHei']  
  
plt.xlabel("时间")  
  
plt.ylabel("指数")  
  
plt.subplot(1,2,x)  
  
x=x+1  
  
plt.xticks(rotation=30)  
  
plt.scatter(get_date_list(), get_index_list(i))  
  
plt.grid()  
  
plt.colorbar()  
  
plt.title("%s"%i)  
  
plt.suptitle("矩阵点图")  
  
plt.savefig('.\\data\\mulite_scatter.png')  
  
plt.close()  
  
print("绘制完毕")
```

效果如下：

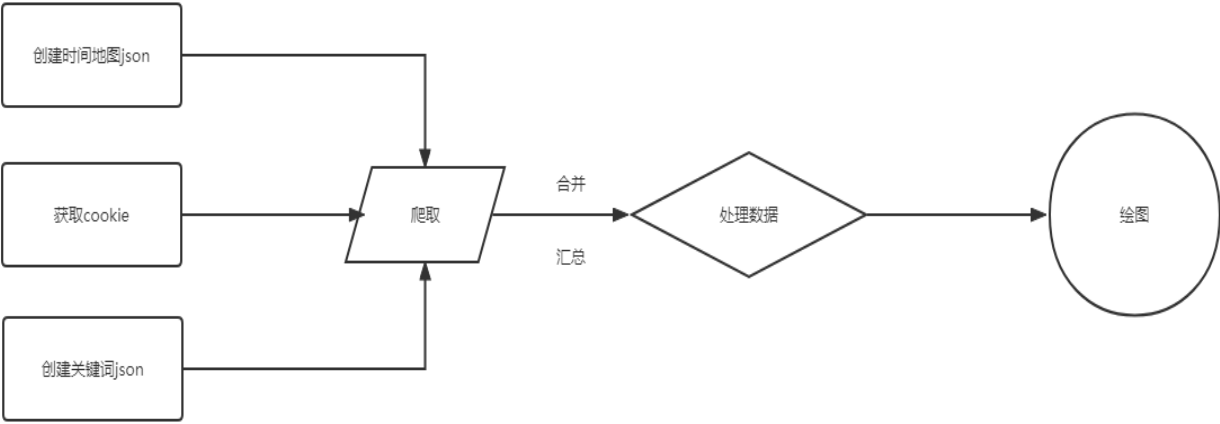


散点图能详细的说明百度指数的具体数值，更具体的表现了疫情和抑郁症的关系

3.5 结果分析

本文通过爬取疫情和抑郁症的百度指数, 获得 2020 一月份到 2022 五月份的 CSV 文件, 并用 Pandas 模块进行数据处理, matplotlib 模块绘制两者的饼图、柱状图、散点图、折线图。然后将有效的数据进行提取, 绘制词云。再将疫情和抑郁症的百度指数进行统计, 最后以 Png 文件的形式对天气数据可视化。进一步可通过该案例分析每月的百度指数的变化关系, 得出疫情和抑郁症的百度指数呈正相关, 政府有必要根据此, 做出提前的预备方案, 诱导民众的心理向好的方面发展。

4 总体框架



5 思路方法

(一) 抽样法：

在此基础上，从集合中选出一些具有代表性的单位，通过对这些选定单位的研究得出结论。在日常生活中，我们在为家庭购买小麦、大米等物品时，不会对包装中的每一块小麦或大米进行检查，而是对一些小麦或大米进行抽样，并以此为依据决定购买小麦或大米。通过这种方法可以节省时间和金钱。在这种方法中保持极端谨慎是很重要的，否则可能会出现得出错误结论的可能性。样本或范式政策基于三个被称为抽样原则的原则。^{56%}三大原理是概率原理、统计规律性原理和大数惯性原理。

在此文中，我们将天数合并到月数，并随机抽选一部分作为调查

（二）归因法

^{82%} 归因模型是一个规则或一组规则，用于确定如何将销售和转化功劳分配给转化路径中的接触点。通过分析各种可视化的图形，将增长和下降的部分分别与因子相连接，可以分析出疫情与抑郁症存在正相关性。例如，Analytics 中的 Last Interaction 模型将 100% 的功劳分配给紧接在销售或转化之前的最终接触点（即点击）。^{62%} 相比之下，首次交互模型将 100% 的功劳分配给启动转化路径的接触点。

（三）调查法

调查方法是一种过程、工具或技术，您可以通过向预定义的人群提问来收集研究中的信息。通常，它有助于研究参与者与进行研究的个人或组织之间的信息交流。

6 创新之处

（一）学术思想特色和创新

本研究通过百度指数将疫情和抑郁症的变化关系通过可视化的方式展现出来，强有力地说明了抑郁症受疫情影响变化大，减少疫情的恐慌、做好民众的心理疏通和治疗，也是后疫情时代必要的行政措施，对国家发展、民众身心健康起到促进作用。

（二）学术观点特色和创新

本研究提出了将百度指数作为数据支撑的方法，百度指数来源于大数据的统计，具有一定的客观性、真实性、普遍性，从另一个维度反映了社会舆论、心理的变化，开通了一条探索数据支撑、数据挖掘、数据分析、数据可视化的新道路。

(三) 学术方法特色和创新

1. 研究视角和内容创新,本研究以百度指数为切入点,对疫情和抑郁症进行理论研究和实证方法,体现了新的视角。
2. 研究方法创新,利用 python 爬虫对数据进行挖掘,再利用 pandas 进行矩阵分析,最后通过 matplotlib 绘制可视化的图形,一套完整的数据挖掘和分析的流程充分说明了两者的联系。

7 参考文献

- 【1】 姜岩主编. PYTHON 程序设计基础=PYTHON PROGRAMMING BASICS. 北京: 机械工业出版社, 2021.01.
- 【2】 刘金花作. 文本挖掘与 Python 实践. 成都: 四川大学出版社, 2021.08.
- 【3】 郭洪伟编. 数据分析方法与应用. 北京: 首都经济贸易大学出版社, 2021.03.
- 【4】 宋万清, 杨寿渊, 陈剑雪, 高永彬编著. 数据挖掘. 北京: 中国铁道出版社, 2019.01.
- 【5】 (美) 大卫·洛辛 (David Loshin) 著; 尚慧萍, 鲍忠贵译. 大数据分析. 北京: 国防工业出版社, 2020.01.
- 【6】 (日) 松本健太郎著, 田中景译. 大数据. 杭州: 浙江人民出版社, 2020.06.
- 【7】 李伊. 数据可视化. 北京: 首都经济贸易大学出版社, 2020.03.
- 【8】 赵涵原. 基于 Python 爬虫的书籍数据可视化分析[J]. 电子技术与软件工程, 2021 (14) : 178-179.
- 【9】 吕云翔编著. 数据结构. 北京: 机械工业出版社, 2020.07.
- 【10】 A Survey: How Python Pitches in IT-World
- 【11】 Hongnian Wang. IEEE .Learning Deep Features for Giant Panda Gender Classification using Face Images