**Course Project Report**

**Task 1:**

**复现教材中以下未完成的章节案例：P39《酒鬼漫步》、P77《北京高考分数线统计》、P115《预处理部分地区信息》、P141《运动员信息的分组与聚合》、P180《火兔分析某年旅游景点数据》、P242《第九章-数据分析实战：北京租房数据统计分析》.**

1.酒鬼漫步：

import numpy as np  
steps = 2000  
draws = np.random.randint(0,2,size = steps)  
direction\_steps = np.where(draws > 0,1,-1)  
distance = direction\_steps.cumsum()  
print(distance.max())  
print(distance.min())  
steps = 15 / 0.5  
print((np.abs(distance >= steps).argmax()))

文本

描述已自动生成

2.北京高考分数线统计：

import pandas as pd  
import numpy as np  
df\_obj = pd.read\_excel(r'C:\Users\94506\Desktop\Files\Data analysis\pythonProject1\scores.xlsx',header=[0,1])  
print(df\_obj)  
  
sorted\_obj = df\_obj.sort\_index(ascending = False)  
# print(sorted\_obj)  
  
print('\n')  
print(sorted\_obj.max())  
print(sorted\_obj.min())  
result1 = sorted\_obj['一本分数线','文科']  
result1 = np.ptp(result1,axis = 0)  
print(result1)  
result2 = sorted\_obj['一本分数线','理科']  
result2 = np.ptp(result2,axis = 0)  
print(result2)  
result3 = sorted\_obj['二本分数线','文科']  
result3 = np.ptp(result3,axis = 0)  
print(result3)  
result4 = sorted\_obj['二本分数线','理科']  
result4 = np.ptp(result4,axis = 0)  
print(result4)  
  
print('\n')  
ser\_obj1 = sorted\_obj['一本分数线','文科']  
print(ser\_obj1[0] - ser\_obj1[1])  
ser\_obj2 = sorted\_obj['一本分数线','理科']  
print(ser\_obj2[0] - ser\_obj2[1])  
ser\_obj3 = sorted\_obj['二本分数线','文科']  
print(ser\_obj3[0] - ser\_obj3[1])  
ser\_obj4 = sorted\_obj['二本分数线','文科']  
print(ser\_obj4[0] - ser\_obj4[1])  
  
print('\n')  
print(sorted\_obj.describe())

文本

低可信度描述已自动生成

电脑截图

描述已自动生成

3.预处理部分地区信息

import pandas as pd  
import matplotlib.pyplot as plt  
import warnings  
warnings.filterwarnings("ignore")  
  
plt.rcParams['font.sans-serif'] = ['Kaitt', 'SimHei']  
plt.rcParams['axes.unicode\_minus'] = False  
  
file\_path\_bj = open(r'C:\Users\94506\Desktop\Files\Data analysis\pythonProject1\北京地区信息.csv',encoding = 'gbk')  
file\_data\_bjinfo = pd.read\_csv(file\_path\_bj)  
print(file\_data\_bjinfo)  
file\_path\_tj = open(r'C:\Users\94506\Desktop\Files\Data analysis\pythonProject1\天津地区信息.csv',encoding = 'gbk')  
file\_data\_tjinfo = pd.read\_csv(file\_path\_tj)  
print('\n',file\_data\_tjinfo)  
  
print()  
print(file\_data\_bjinfo.duplicated())  
print('\n',file\_data\_bjinfo.duplicated())  
  
file\_data\_bjinfo = file\_data\_bjinfo.drop\_duplicates()  
print('\n',file\_data\_bjinfo)  
  
print()  
print(file\_data\_tjinfo.isnull())  
  
print()  
population = float("{:.2f}".format(file\_data\_tjinfo['常住人口（万人）'].mean()))  
values = {'常住人口（万人）':population}  
file\_data\_tjinfo = file\_data\_tjinfo.fillna(value = values)  
print(file\_data\_tjinfo)  
  
file\_data\_bjinfo.boxplot(column = ['行政面积（K㎡）','户籍人口（万人）','男性','女性','GDP（亿元）','常住人口（万人）'])  
plt.show()  
file\_data\_tjinfo.boxplot(column = ['行政面积（K㎡）','户籍人口（万人）','男性','女性','GDP（亿元）','常住人口（万人）'])  
plt.show()  
  
print()  
print(pd.concat([file\_data\_bjinfo,file\_data\_tjinfo],ignore\_index = True))

图片包含 图形用户界面

描述已自动生成

图片包含 日历

描述已自动生成

表格

中度可信度描述已自动生成 表格

低可信度描述已自动生成

图片包含 日历

描述已自动生成

表格

低可信度描述已自动生成

图片包含 日历

描述已自动生成

图片包含 图形用户界面

描述已自动生成

图表, 箱线图

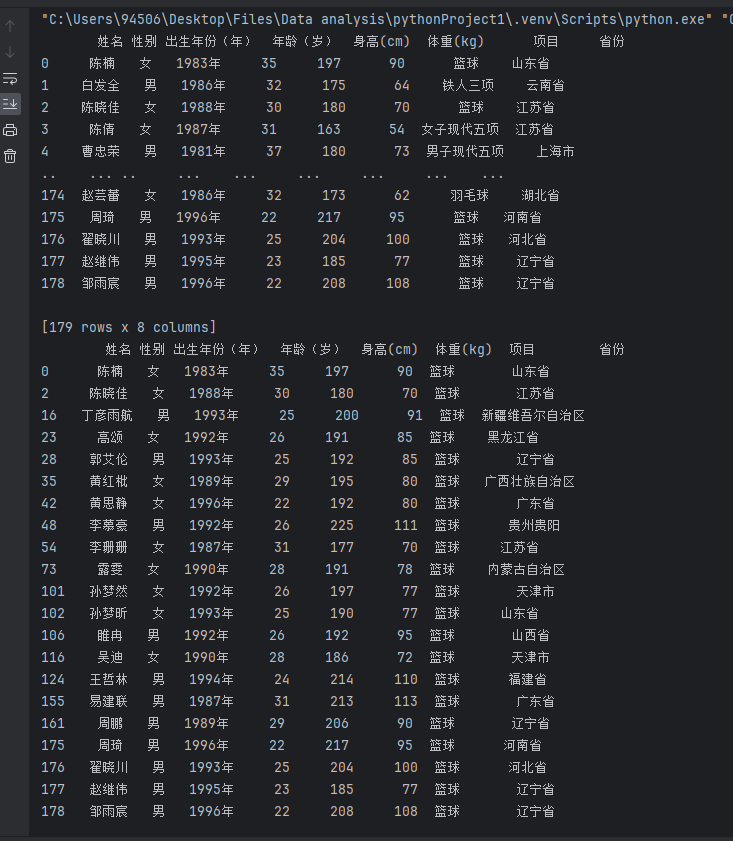
描述已自动生成

图表

描述已自动生成

4.运动员信息的分组与聚合

import pandas as pd  
file\_path = open('运动员信息表.csv')  
df = pd.read\_csv(file\_path)  
print(df)  
  
data\_group = df.groupby('项目')  
df\_basketball = dict([x for x in data\_group])['篮球']  
print(df\_basketball)  
  
print()  
sex = df\_basketball[["年龄（岁）","身高(cm)","体重(kg)"]].groupby(df\_basketball["性别"])  
print(sex.mean())  
  
print()  
info = sex.transform('mean')  
print(info)  
  
print()  
sex = df\_basketball.groupby(df["性别"])  
basketball\_male=dict([x for x in sex])['男']  
print(basketball\_male)  
  
def range\_data\_group(arr):  
 return arr.max()-arr.min()  
print()  
print(basketball\_male.agg({"年龄（岁）":range\_data\_group,  
 "身高(cm)":range\_data\_group,  
 "体重(kg)":range\_data\_group}))  
  
print()  
df\_basketball['体质指数'] = 0  
print(df\_basketball)  
  
def outer(num):  
 def ath\_bmi(sum\_bmi):  
 weight = df\_basketball["身高(cm)"]  
 height = df\_basketball["体重(kg)"]  
 sum\_bmi = weight / (height / 100) \*\* 2  
 return num + sum\_bmi  
 return ath\_bmi  
  
all\_bmi = df\_basketball["体质指数"]  
df\_basketball["体质指数"] = df\_basketball[["体质指数"]].apply(outer(all\_bmi))  
print()  
print(df\_basketball)



文本

低可信度描述已自动生成

图片包含 文本

描述已自动生成

文本

低可信度描述已自动生成

文本, 日历

中度可信度描述已自动生成

5.画图分析某年旅游景点数据

# -\*- coding: utf-8 -\*-  
import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
scenery\_file\_path = open('风景名胜区.csv')  
scenery\_data = pd.read\_csv(scenery\_file\_path)  
print(scenery\_data)  
  
print()  
area = float("{:.1f}".format(scenery\_data['总面积(平方公里)'].mean()))  
tourist = float("{:.1f}".format(scenery\_data['游客量(万人次)'].mean()))  
values = {"总面积(平方公里)":area,"游客量(万人次)":tourist}  
scenery\_data = scenery\_data.fillna(value = values)  
print(scenery\_data)  
  
print()  
data = scenery\_data.groupby("省份")  
hebei\_scenery = dict([x for x in data])['河北']  
print(hebei\_scenery)  
  
plt.rcParams['font.sans-serif'] = ['SimHei']  
plt.rcParams['axes.unicode\_minus'] = False  
area = hebei\_scenery['总面积(平方公里)'].values  
tourist = hebei\_scenery['游客量(万人次)'].values  
plt.figure(figsize = (12,6))  
x\_num = range(0,len(area))  
x\_dis = [i + 0.3 for i in x\_num]  
plt.bar(x\_num,area,color = 'g',width = .3,label = '总面积')  
plt.bar(x\_dis,tourist,color = 'r',width = .3,label = '游客量')  
plt.ylabel('单位：平方千米/万人次')  
plt.title('河北景点面积及游客数量')  
plt.legend(loc = 'upper right')  
plt.xticks(range(0,10),['苍岩山','嶂石岩','西柏坡-天桂山','秦皇岛北戴河','响堂山','娲皇宫','太行大峡谷','崆山白云洞','野三坡','承德避署山庄外八庙'])  
plt.show()  
  
every\_scenery = hebei\_scenery['游客量(万人次)'].values  
all\_scenery = hebei\_scenery['游客量(万人次)'].sum()  
percentage = (every\_scenery / all\_scenery) \* 100  
np.set\_printoptions(precision = 2)  
labels = ['苍岩山','嶂石岩','西柏坡-天桂山','秦皇岛北戴河','响堂山','娲皇宫','太行大峡谷','崆山白云洞','野三坡','承德避署山庄外八庙']  
plt.axes(aspect = 1)  
plt.pie(x = percentage,labels = labels,autopct = '%3.2f %%',shadow = True,labeldistance = 1.2,startangle =90,pctdistance = 0.7)  
plt.legend(loc = 'upper left')  
plt.show()

图片包含 日历

描述已自动生成

电脑的屏幕

描述已自动生成

图表, 条形图

描述已自动生成

图表, 饼图

描述已自动生成

6. 第九章-数据分析实战：北京租房数据统计分析

# -\*- coding: utf-8 -\*-  
import pandas as pd  
import numpy as np  
  
file\_path = open('链家北京租房数据.csv')  
file\_data = pd.read\_csv(file\_path)  
print(file\_data)  
print(file\_data.duplicated())  
file\_data = file\_data.drop\_duplicates()  
print(file\_data)  
  
file\_data = file\_data.dropna()  
print(file\_data)  
  
data\_new = np.array([])  
data = file\_data['面积(㎡)'].values  
for i in data:  
 data\_new = np.append(data\_new, np.array(i[:-2]))  
data = data\_new.astype(np.float64)  
file\_data.loc[:, '面积(㎡)'] = data  
print(file\_data)  
  
housetype\_data = file\_data['户型']  
temp\_list = []  
for i in housetype\_data:  
 new\_info = i.replace('房间', '室')  
 temp\_list.append(new\_info)  
file\_data.loc[:, '户型'] = temp\_list  
print(file\_data)  
  
new\_df = pd.DataFrame({'区域': file\_data['区域'].unique(), '数量': [0] \* 13})  
print(new\_df)  
  
print()  
groupby\_area = file\_data.groupby(by='区域').count()  
new\_df['数量'] = groupby\_area.values  
print(new\_df)  
print('\n', new\_df.sort\_values(by='数量', ascending=False))  
  
file\_data['位置'] = '北京市' + file\_data['区域'].values + '区' + file\_data['小区名称'].values  
print(file\_data)  
  
# # coding = utf-8  
# import requests  
# import time  
# import json  
#  
#  
# class LngLat:  
# def get\_data(self):  
# house\_names = file\_data['位置']  
# house\_names = house\_names.tolist()  
# return house\_names  
#  
# def get\_url(self):  
# url\_temp = "http://api.map.baidu.com/geocoder/v2/?address={}" \  
# "&output = json" \  
# "&ak = gfRHhpMyINfZkCEni0t5Vu05p24QfxMB" \  
# "&callback = showLocation"  
# house\_names = self.get\_data()  
# return [url\_temp.format(i) for i in house\_names]  
#  
# def parse\_url(self, url):  
# while 1:  
# try:  
# r = requests.get(url)  
# except requests.exceptions.ConnectionError:  
# time.sleep(2)  
# continue  
# return r.content.decode('UTF-8')  
#  
# def run(self):  
# li = []  
# urls = self.get\_url()  
# for url in urls:  
# data = self.parse\_url(url)  
# str = data.split("{")[-1].split("}")[0]   
# try:  
# lng = float(str.split(",")[0].split(":")[1])  
# lat = float(str.split(",")[1].split(":")[1])  
# except ValueError:  
# continue  
# dict\_data = dict(lng=lng, lat=lat, count=1)  
# li.append(dict\_data)  
# f = open(r'C:\Users\94506\Desktop\Files\Data analysis\pythonProject1\经纬度信息.txt', 'w')  
# f.write(json.dumps(li))  
# f.close()  
# print('正在写入...')  
# print('写入成功！')  
#  
#  
# if \_\_name\_\_ == '\_\_main\_\_':  
# execute = LngLat()  
# execute.run()  
  
def all\_house(arr):  
 arr = np.array(arr)  
 key = np.unique(arr)  
 result = {}  
 for k in key:  
 mask = (arr == k)  
 arr\_new = arr[mask]  
 v = arr\_new.size  
 result[k] = v  
 return result  
  
house\_array = file\_data['户型']  
house\_info = all\_house(house\_array)  
print(house\_info)  
  
house\_type = dict((key,value) for key,value in house\_info.items() if value > 50)  
show\_houses = pd.DataFrame({'户型':[x for x in house\_type.keys()],'数量':[x for x in house\_type.values()]})  
print(show\_houses)  
  
import matplotlib.pyplot as plt  
import matplotlib  
matplotlib.rcParams['font.sans-serif'] = ['SimHei']  
matplotlib.rcParams['axes.unicode\_minus'] = False  
house\_type = show\_houses['户型']  
house\_type\_num = show\_houses['数量']  
plt.barh(range(11),house\_type\_num,height = 0.7,color = 'steelblue',alpha = 0.8)  
plt.yticks(range(11),house\_type)  
plt.xlim(0,2500)  
plt.xlabel('数量')  
plt.ylabel('户型种类')  
plt.title("北京地区各户型房屋数量")  
for x,y in enumerate(house\_type\_num):  
 plt.text(y + 0.2,x - 0.1,'%s' %y)  
plt.show()  
  
df\_all = pd.DataFrame({'区域':file\_data['区域'].unique(),  
 '房屋总金额':[0] \* 13,  
 '总面积(㎡)':[0] \* 13})  
print(df\_all)  
sum\_price = file\_data['价格(元/月)'].groupby(file\_data['区域']).sum()  
sum\_area = file\_data['面积(㎡)'].groupby(file\_data['区域']).sum()  
df\_all['房屋总金额'] = sum\_price.values  
df\_all['总面积(㎡)'] = sum\_area.values  
print(df\_all)  
  
df\_all['每平方千米租金（元）'] = round(df\_all['房屋总金额'] / df\_all['总面积(㎡)'],2)  
print(df\_all)  
  
df\_merge = pd.merge(new\_df,df\_all)  
print(df\_merge)  
  
import matplotlib.ticker as mtick  
from matplotlib.font\_manager import FontProperties  
num = df\_merge['数量']  
price = df\_merge['每平方千米租金（元）']  
l = [i for i in range(13)]  
plt.rcParams['font.sans-serif'] = ['SimHei']  
lx = df\_merge['区域']  
fig = plt.figure()  
ax1 = fig.add\_subplot(111)  
ax1.plot(l,price,'or-',label = '价格')  
for i,(\_x,\_y) in enumerate(zip(l,price)):  
 plt.text(\_x,\_y,price[i],color = 'black',fontsize = 10)  
ax1.set\_ylim([0,200])  
ax1.set\_ylabel('价格')  
plt.legend(prop = {'family':'SimHei','size':8},loc = 'upper left')  
ax2 = ax1.twinx()  
plt.bar(l,num,alpha = 0.3,color = 'green',label = '数量')  
ax2.set\_ylabel('数量')  
ax2.set\_ylim([0,2000])  
plt.legend(prop = {'family':'SimHei','size':8},loc = 'upper right')  
plt.xticks(l,lx)  
plt.show()  
  
print('房屋最大面积是%d平方米'%(file\_data['面积(㎡)'].max()))  
print('房屋最小面积是%d平方米'%(file\_data['面积(㎡)'].min()))  
print('房屋最高价格为%d元'%(file\_data['价格(元/月)'].max()))  
print('房屋最低价格为%d元'%(file\_data['价格(元/月)'].min()))  
  
area\_divide = [1,30,50,70,90,120,140,160,1200]  
area\_cut = pd.cut(list(file\_data['面积(㎡)']),area\_divide)  
area\_cut\_data = area\_cut.describe()  
print(area\_cut\_data)  
  
area\_percentage = (area\_cut\_data['freqs'].values) \* 100  
np.set\_printoptions(precision = 2)  
labels = ['30平方米以下','30-50平方米','50-70平方米','70-90平方米',  
 '90-120平方米','120-140平方米','140-160平方米','160平方米以上',]  
plt.axes(aspect = 1)  
plt.pie(x = area\_percentage,labels = labels,autopct = "%.2f %%",  
 shadow = True,labeldistance = 1.2,startangle = 90,pctdistance = 0.7)  
plt.legend(loc = 'upper right')  
plt.show()

文本

描述已自动生成

文本

描述已自动生成

屏幕上写着字

描述已自动生成

图片包含 文本

描述已自动生成

文本

描述已自动生成

文本

描述已自动生成

屏幕上有字

中度可信度描述已自动生成

文本

中度可信度描述已自动生成

图形用户界面, 文本

描述已自动生成

文本

描述已自动生成

图表, 条形图

描述已自动生成

图表, 折线图

描述已自动生成

图表, 饼图

描述已自动生成

**Task 2:**

**利用八爪鱼爬虫工具，完成以下任务：**

1. **爬取淘宝或天猫某商品的评价信息，利用第八章文本数据分析所学内容或课外拓展内容，对所有评价进行分析，得出最后该商品的总体评价是好评还是差评。要求该商品的评价数量要超过1000条.**
2. **爬取列表网任意品牌在宁波的二手车信息，利用数据分析和数据可视化技术，对该品牌二手车信息进行汇总并出具分析报告。要求爬取的二手车不少于500条信息，分析报告要求展示车辆的各项信息以及数据分布等内容，具体形式不做硬性要求。**

模板设置：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

爬取多个商品链接，筛选有效数据：

图形用户界面, 文本

中度可信度描述已自动生成

爬取评论信息：

文本

低可信度描述已自动生成

对数据进行预处理：

图片包含 文本

描述已自动生成

代码部分：

1. **import** pandas as pd
2. **from** textblob **import** TextBlob
4. # Load the data
5. file\_path = r'C:\Users\27234\Desktop\AI\BIGDATA\淘宝商品评论.csv'
6. df = pd.read\_csv(file\_path,encoding='gbk')
8. # The review column is named '评论内容'
9. reviews\_column = '评论内容'
11. # Function to analyze sentiment
12. **def** analyze\_sentiment(review):
13. analysis = TextBlob(str(review))
14. **if** analysis.sentiment.polarity > 0:
15. **return** 'positive'
16. **elif** analysis.sentiment.polarity == 0:
17. **return** 'neutral'
18. **else**:
19. **return** 'negative'
21. # Apply sentiment analysis
22. **if** reviews\_column **in** df.columns:
23. df['sentiment'] = df[reviews\_column].apply(analyze\_sentiment)
24. result = df['sentiment'].value\_counts(normalize=True)
25. # Determine overall sentiment
26. total\_reviews = len(df)
27. positive\_reviews = result.get('positive', 0) \* total\_reviews
28. negative\_reviews = result.get('negative', 0) \* total\_reviews
29. overall\_sentiment = ('positive' **if** positive\_reviews > negative\_reviews **else**
30. 'negative' **if** negative\_reviews > positive\_reviews **else**
31. 'neutral')
32. **else**:
33. **raise** Exception(f'Column "{reviews\_column}" does not exist in the provided data.')
35. # The final output would be a tuple of the overall sentiment and the sentiment distribution
36. **print**(overall\_sentiment)

运行结果：

文本

描述已自动生成

二手车：

爬取数据：

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

代码部分：

1. **import** pandas as pd
2. **import** numpy as np
3. **import** matplotlib.pyplot as plt
4. **from** matplotlib.font\_manager **import** FontProperties


8. # 设置中文字体
9. plt.rcParams['font.sans-serif'] = ['SimHei']  # 'SimHei'是黑体的意思
10. plt.rcParams['axes.unicode\_minus'] = False  # 正确显示负号
12. # 加载数据
13. file\_path = r"C:\Users\27234\Desktop\AI\BIGDATA\列表网二手车列表及详情页采集.csv"
14. df = pd.read\_csv(file\_path, encoding="gbk")
16. # 填充缺失数据为"未知"
17. df.fillna("未知", inplace=True)
19. **import** jieba
20. **import** jieba.analyse
21. **import** matplotlib.pyplot as plt
22. **import** pandas as pd
23. **import** re
25. # 设置中文字体
26. plt.rcParams['font.sans-serif'] = ['SimHei']  # 'SimHei'是黑体的意思
27. plt.rcParams['axes.unicode\_minus'] = False  # 正确显示负号
29. # 加载数据
30. file\_path = r"C:\Users\27234\Desktop\列表网二手车列表及详情页采集.csv"
31. df = pd.read\_csv(file\_path, encoding="gbk")
33. # 填充缺失数据为"未知"
34. df.fillna("未知", inplace=True)
36. # 根据'品牌'列进行分组，创建子数据集
37. grouped = df.groupby('品牌')
39. # 创建一个字典，用于存储按品牌分组的子数据集
40. brand\_datasets = {brand: data.reset\_index(drop=True) **for** brand, data **in** grouped}
42. **def** clean\_and\_extract\_number(series):
43. # 排除'面议'和'未知'，并提取数字，这里使用正则表达式
44. **return** series.str.replace('面议|未知', '', regex=True).str.extract(r'(\d+\.?\d\*)', expand=False).astype(float)
46. # 对每个品牌进行数据清洗和提取数字
47. **for** brand, data **in** brand\_datasets.items():
48. data['价格'] = clean\_and\_extract\_number(data['价格'])
49. data['年龄'] = clean\_and\_extract\_number(data['年龄'])
50. data['行驶里程'] = clean\_and\_extract\_number(data['行驶里程'])
52. # 只有当三个字段都不为空时，才能绘制箱型图
53. **if** data['价格'].notna().any() **and** data['年龄'].notna().any() **and** data['行驶里程'].notna().any():
54. # 绘制箱型图
55. fig, axes = plt.subplots(1, 3, figsize=(18, 5))  # 创建一个包含3个子图的图形
56. data.boxplot(column='价格', ax=axes[0])  # 价格的箱型图
57. axes[0].set\_title(f'{brand} 品牌价格箱型图')
58. axes[0].set\_ylabel('价格')
60. data.boxplot(column='年龄', ax=axes[1])  # 年龄的箱型图
61. axes[1].set\_title(f'{brand} 品牌年龄箱型图')
62. axes[1].set\_ylabel('年龄')
64. data.boxplot(column='行驶里程', ax=axes[2])  # 行驶里程的箱型图
65. axes[2].set\_title(f'{brand} 品牌行驶里程箱型图')
66. axes[2].set\_ylabel('行驶里程')
68. plt.suptitle(f'{brand} 品牌数据箱型图')  # 设置总标题
69. plt.tight\_layout(rect=[0, 0, 1, 0.95])  # 调整布局
70. plt.show()  # 展示箱型图
71. # 假设"描述"列包含了文本信息
72. descriptions = df["描述"].tolist()
73. keywords = []
75. # 通过循环提取每个描述的关键字
76. **for** desc **in** descriptions:
77. # 这里设置topK为10，表示提取每个描述中的前10个关键词
78. extracted\_keywords = jieba.analyse.extract\_tags(desc, topK=10)
79. keywords.append(extracted\_keywords)
81. # 如果您有停用词表，可以这样使用它
82. # file\_path = open(r"C:\Users\27234\Desktop\AI\BIGDATA\停用词表.txt", encoding="utf-8")
83. # stop\_word = file\_path.read()
84. # jieba.analyse.set\_stop\_words(stop\_word)
86. **import** matplotlib.pyplot as plt
87. **import** pandas as pd
88. **import** numpy as np
90. **from** wordcloud **import** WordCloud
92. # 将所有关键词合并为一个长字符串，以便生成词云
93. all\_keywords = " ".join([" ".join(kw) **for** kw **in** keywords])
95. # 生成词云
96. wordcloud = WordCloud(
97. font\_path=r"C:\Users\27234\AppData\Local\Microsoft\Windows\Fonts\AaErMoXingShu-2.ttf",
98. width=800,
99. height=400,
100. ).generate(all\_keywords)
102. # 绘制词云
103. plt.figure(figsize=(10, 5))
104. plt.imshow(wordcloud, interpolation="bilinear")
105. plt.axis("off")
106. plt.show()

运行结果：

图示, 日历

描述已自动生成

图表, 箱线图

描述已自动生成

黑板上的涂鸦

描述已自动生成