课程专业实践报告

实验题目	综合实践

一．程序代码

第一问

```python
# 画出600276股票近五年营业总收入和净利润的直方图
import pandas as pd
import matplotlib.pyplot as plt

df = pd.read_excel('week13/data/transformed/600276_main_report.xls')
df = df[['科目\时间','净利润(元)','营业总收入(元)']]
df = pd.DataFrame(df)
df['Date'] = pd.to_datetime(df['科目\时间'])
df1 = df[df['Date'] >= pd.to_datetime('2018-01-01')]

# 绘制 营业总收入(元) 直方图
plt.figure()
plt.bar(df1['Date'],df1['营业总收入(元)'],width=30,label='Total operating income')
plt.bar(df1['Date'],df1['净利润(元)'],width=30,label='Net profit')

plt.title('Total operating income and net profit for the last five years')
plt.xlabel('Date')
plt.ylabel('Total Revenue(RMB)')
plt.xticks(rotation=45)  # 旋转x轴标签，避免重叠
plt.legend()

# 显示图表
plt.tight_layout()
plt.show()
```

第二问:

```python
import os
import pandas as pd

# 找出近五年平均利润增长率最高的股票
source_path = 'week13/data/transformed/'

MAX = -1000
for filename in os.listdir(source_path):
    if filename.endswith('.xls'):
        # 构建完整的文件路径
        file_path = os.path.join(source_path, filename)
        # 读取Excel文件
        df = pd.read_excel(file_path)
        df = df[['科目\时间','净利润同比增长率']]
        df = pd.DataFrame(df)
        # 筛选近五年记录
        df['科目\时间'] = pd.to_datetime(df['科目\时间'])
        df = df[df['科目\时间'] >= pd.to_datetime('2018-01-01')]
        # 删除--记录
        df = df[df['净利润同比增长率'] != '--']
        # 字符串转换为numeric数据类型
        df['净利润同比增长率'] =  df['净利润同比增长率'].str.rstrip('%').astype('float') / 100
        # 求平均值
        Mean = df['净利润同比增长率'].mean()
        print(filename,'净利润同比增长率为: ',Mean)
        if Mean > MAX :
            MAX = Mean
            Excel = filename

print('--------------------------------')
print('for ended')
print('近五年平均利润增长率最高的股票为: ',filename,',其净利润同比增长率为: ',MAX,'%')
```

第三问:

```python
import os
import pandas as pd

# 找出近五年平均净利润率最高的股票
source_path = 'week13/data/transformed/'

MAX = -1000
for filename in os.listdir(source_path):
    if filename.endswith('.xls'):
        # 构建完整的文件路径
        file_path = os.path.join(source_path, filename)
        # 读取Excel文件
        df = pd.read_excel(file_path)
        df = df[['科目\时间','销售净利率']]
        df = pd.DataFrame(df)
        # 筛选近五年记录
        df['科目\时间'] = pd.to_datetime(df['科目\时间'])
        df = df[df['科目\时间'] >= pd.to_datetime('2018-01-01')]
        # 删除--记录
        df = df[df['销售净利率'] != '--']
        # 字符串转换为numeric数据类型
        df['销售净利率'] =  df['销售净利率'].str.rstrip('%').astype('float') / 100
        # 求平均值
        Mean = df['销售净利率'].mean()
        print(filename,'销售净利率为: ',Mean)
        if Mean > MAX :
            MAX = Mean
            Excel = filename
print('--------------------------------')
print('for ended')
print('近五年平均净利润最高的股票为: ',filename,',其平均净利润率为: ',MAX,'%')

```

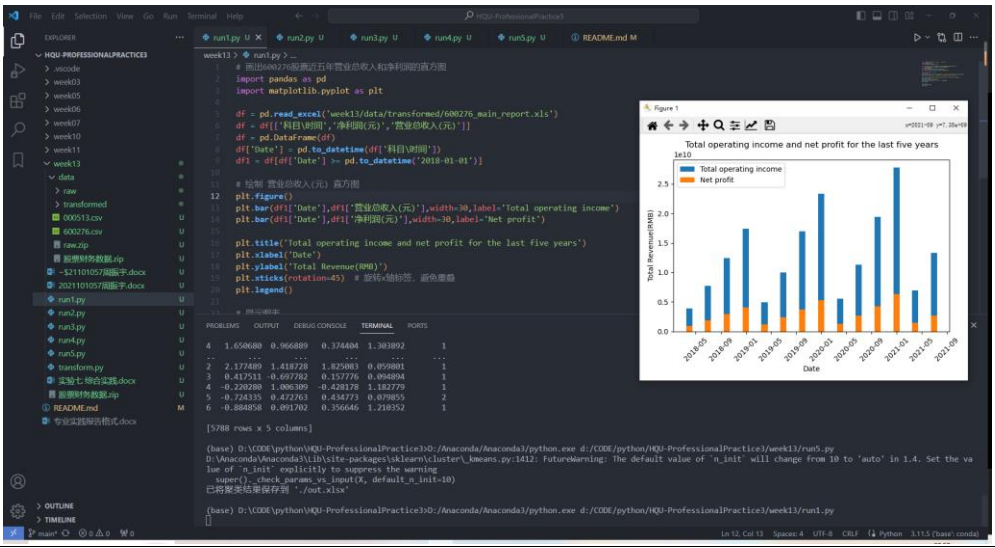
第四问:

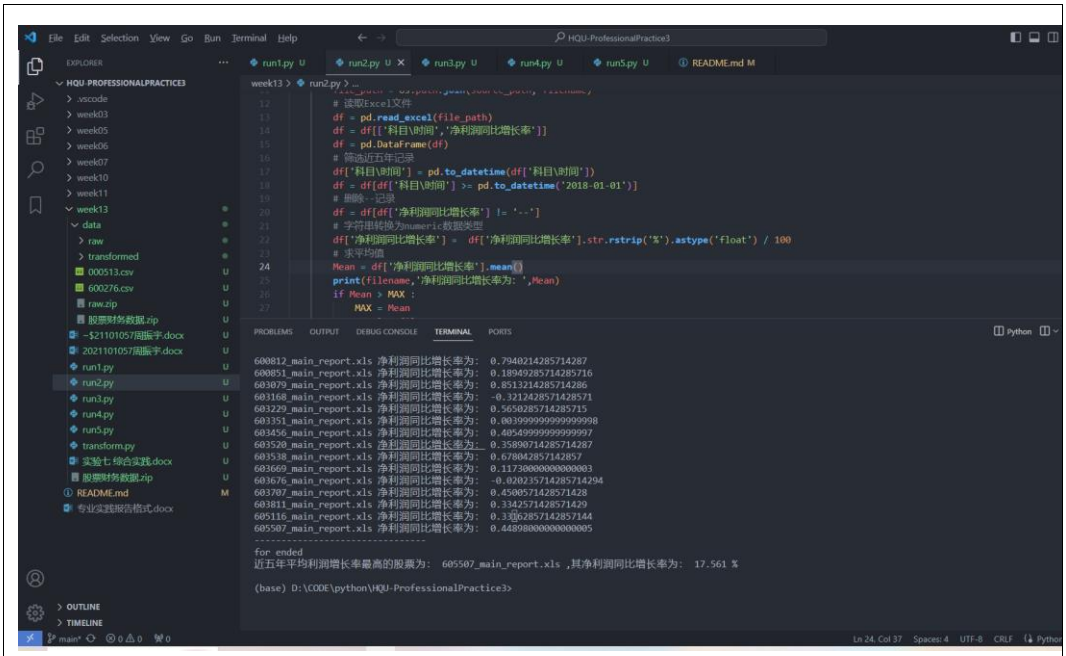
```python
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import warnings

# 预测000513股票2021年9月20-24日股票价格
# 读取股票数据
file_path = 'week13/data/000513.csv'  # 替换为您的文件路径
df = pd.read_csv(file_path, encoding='GBK')
df['Date'] = pd.to_datetime(df['日期'])
df = df.sort_values(by='Date', ascending=True)

# 使用 '收盘价' 数据来训练模型
df = df['收盘价']

# 拟合ARIMA模型,这里使用(5,1,0)作为模型参数
warnings.filterwarnings("ignore")
model = ARIMA(df, order=(5,1,0))
model_fit = model.fit()

# 预测接下来5天的价格
forecast = model_fit.forecast(steps=10)
out = forecast.tolist()

# 打印预测结果
for i in [2,3,4,5,6]:
    print('2021年9月', 18 + i,'日股票价格(收盘价)为: ',out[i],'元')
```

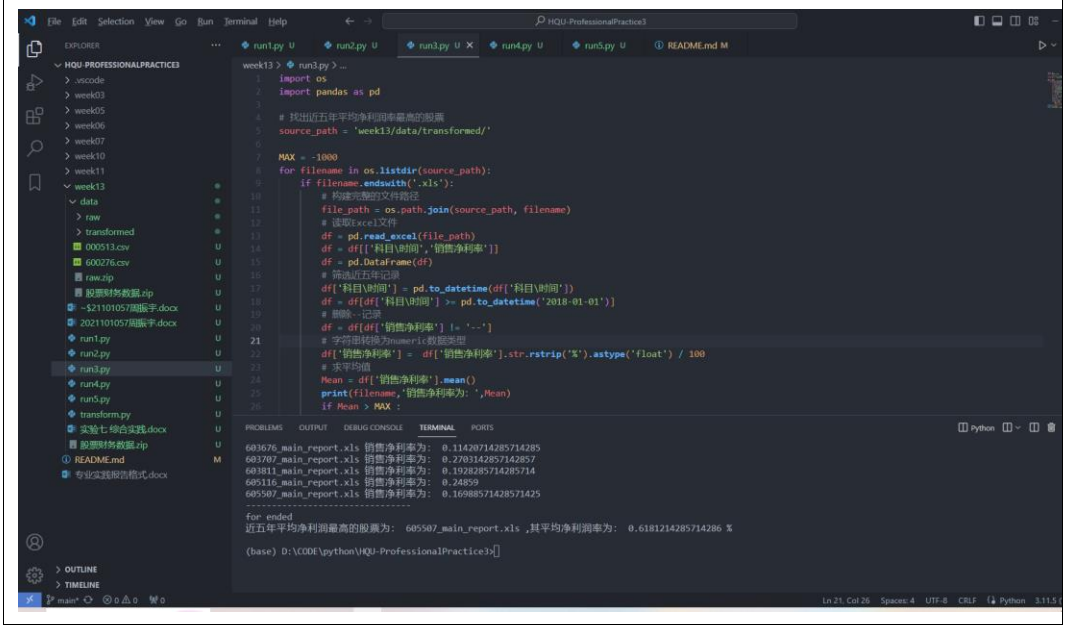
第五问:

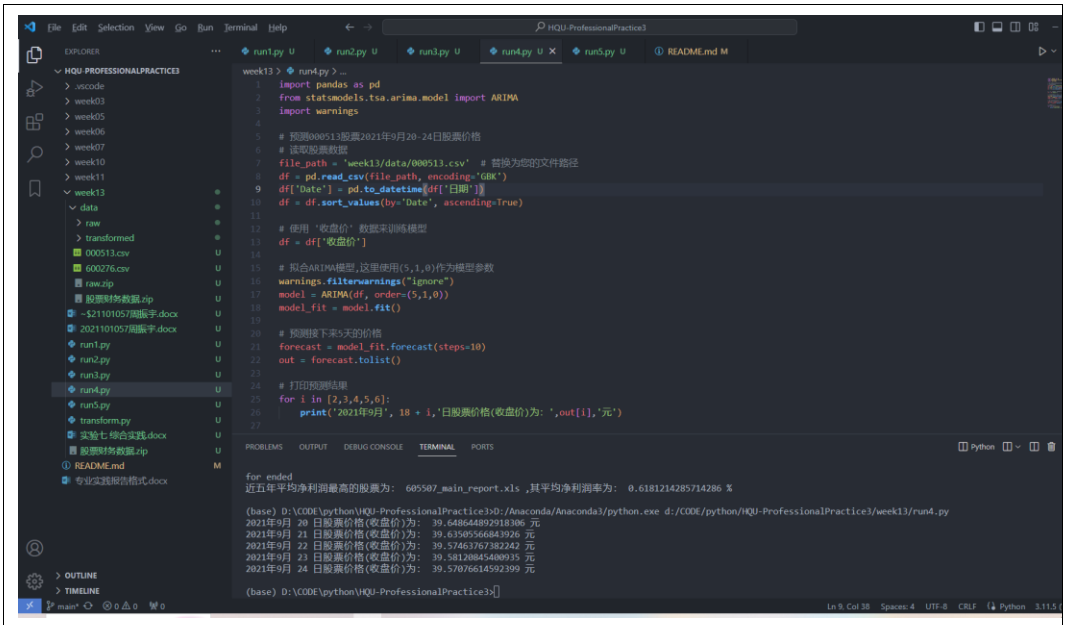
```python
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import os
import numpy as np
# 应该选举哪些财务指标，选用哪种合适的聚类分析方法将股票分成三种类型。
source_path = 'week13/data/transformed/'
df = pd.DataFrame()
data = pd.DataFrame()

# 选择使用的财务指标
def fun(data):
    features = ['净利润(元)', '营业总收入(元)', '净资产收益率-摊薄', '资产负债比率']
    data = data[features]
    data = data[data['净利润(元)'] != '--']
    data = data[data['营业总收入(元)'] != '--']
    data = data[data['净资产收益率-摊薄'] != '--']
    data = data[data['资产负债比率'] != '--']
    return data

# 数据预处理：转换百分比和标准化
def pre(data):
    data['净资产收益率-摊薄'] = data['净资产收益率-摊薄'].str.rstrip('%').astype('float') / 100.0
    data['资产负债比率'] = data['资产负债比率'].str.rstrip('%').astype('float') / 100.0
    scaler = StandardScaler()
    data_scaled = scaler.fit_transform(data)
    data_scaled_df = pd.DataFrame(data_scaled, columns=data.columns)  # 将NumPy数组转换回DataFrame
    return data_scaled_df

# 示例：读取和合并多个Excel文件
for filename in os.listdir(source_path):
    if filename.endswith('.xls'):
        file_path = os.path.join(source_path, filename)
        data = pd.read_excel(file_path)
        data = fun(data)
        data = pre(data)
        df = pd.concat([df, data])

# 应用K-Means聚类
kmeans = KMeans(n_clusters=3, random_state=0)
clusters = kmeans.fit_predict(df)
# print(clusters)
df['Cluster'] = pd.Series(clusters)

# 输出到Excel文件
output_path = './week13/out.xlsx'  # 您可以根据需要更改文件名和路径
df.to_excel(output_path, index=False)

print(f"已将聚类结果保存到 '{output_path}'")
```

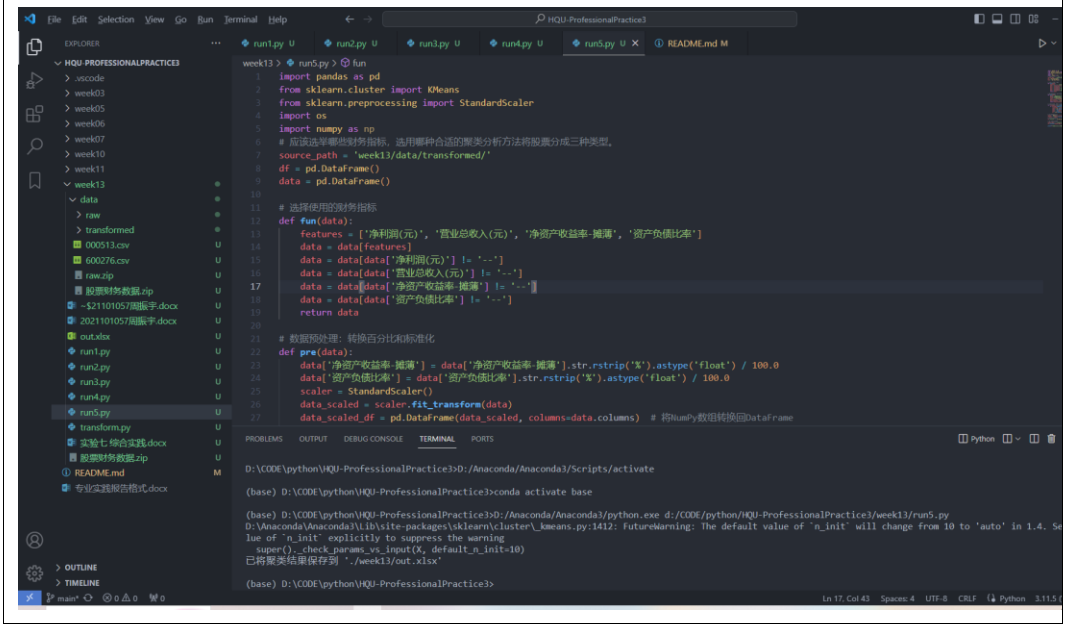
二．运行结果

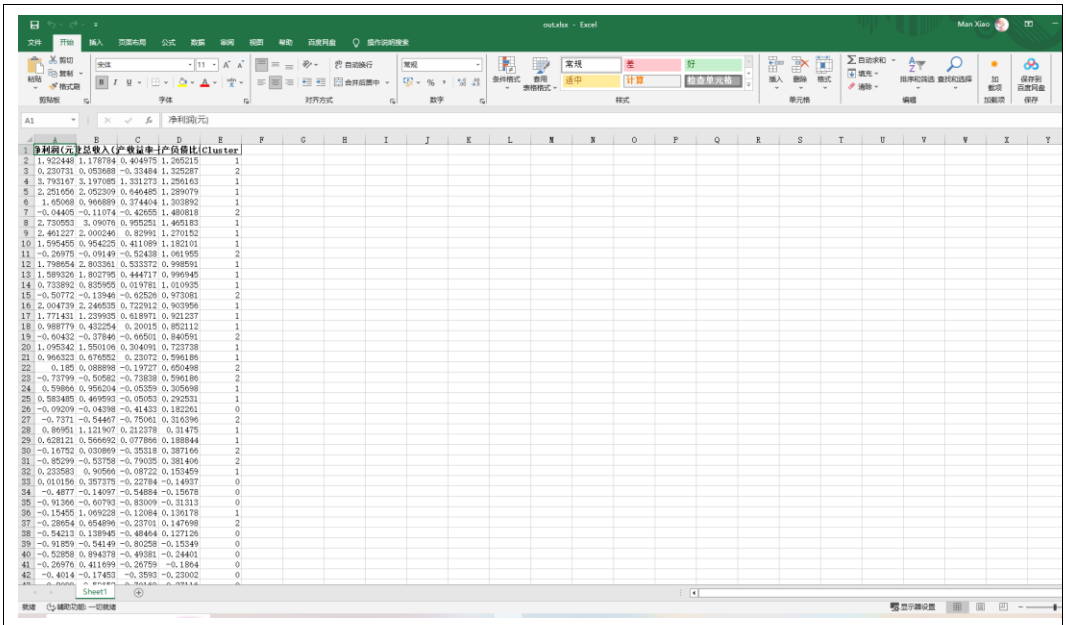



File Edit Selection View Go Run Terminal Help
HQU-ProfessionalPractice3
EXPLORER
HQU-PROFESSIONALPRACTICE3
run1.py U
run2.py U
run3.py U
run4.py U
run5.py U
README.md M
week13 > run2.py > ...
# 读取Excel文件
df = pd.read_excel(file_path)
df = df[['科目\时间','净利润同比增长率']]
df = pd.DataFrame(df)
# 筛选近五年记录
df['科目\时间'] = pd.to_datetime(df['科目\时间'])
df = df[df['科目\时间'] >= pd.to_datetime('2018-01-01')]
# 删除--记录
df = df[df['净利润同比增长率'] != '--']
# 字符串转换为numeric数据类型
df['净利润同比增长率'] = df['净利润同比增长率'].str.rstrip('%').astype('float') / 100
# 求平均值
Mean = df['净利润同比增长率'].mean()
print(filename,'净利润同比增长率为: ',Mean)
if Mean > MAX :
MAX = Mean
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
600812_main_report.xls 净利润同比增长率为: 0.7940214285714287
600851_main_report.xls 净利润同比增长率为: 0.1894928571428571
603079_main_report.xls 净利润同比增长率为: 0.8513214285714286
603168_main_report.xls 净利润同比增长率为: -0.3212428571428571
603229_main_report.xls 净利润同比增长率为: 0.5650285714285715
603351_main_report.xls 净利润同比增长率为: 0.003999999999999998
603456_main_report.xls 净利润同比增长率为: 0.40549999999999997
603520_main_report.xls 净利润同比增长率为: 0.35890714285714287
603538_main_report.xls 净利润同比增长率为: 0.6780428571428571
603669_main_report.xls 净利润同比增长率为: 0.11730000000000003
603676_main_report.xls 净利润同比增长率为: -0.020235714285714294
603707_main_report.xls 净利润同比增长率为: 0.4500571428571428
603811_main_report.xls 净利润同比增长率为: 0.3342571428571429
605116_main_report.xls 净利润同比增长率为: 0.33062857142857144
605507_main_report.xls 净利润同比增长率为: 0.44898000000000005
for ended
近五年平均利润增长率最高的股票为: 605507_main_report.xls ,其净利润同比增长率为: 17.561 %
(base) D:\CODE\python\HQU-ProfessionalPractice3>
OUTLINE
TIMELINE
main*
Ln 24, Col 37 Spaces: 4 UTF-8 CRLF


File Edit Selection View Go Run Terminal Help
HQU-ProfessionalPractice3
EXPLORER
HQU-PROFESSIONALPRACTICE3
run1.py U
run2.py U
run3.py U
run4.py U
run5.py U
README.md M
week13 > run3.py > ...
import os
import pandas as pd
# 找出近五年平均净利润率最高的股票
source_path = 'week13/data/transformed/'
MAX = -1000
for filename in os.listdir(source_path):
if filename.endswith('.xls'):
# 构建完整的文件路径
file_path = os.path.join(source_path, filename)
# 读取Excel文件
df = pd.read_excel(file_path)
df = df[['科目\时间','销售净利率']]
df = pd.DataFrame(df)
# 筛选近五年记录
df['科目\时间'] = pd.to_datetime(df['科目\时间'])
df = df[df['科目\时间'] >= pd.to_datetime('2018-01-01')]
# 删除--记录
df = df[df['销售净利率'] != '--']
# 字符串转换为numeric数据类型
df['销售净利率'] = df['销售净利率'].str.rstrip('%').astype('float') / 100
# 求平均值
Mean = df['销售净利率'].mean()
print(filename,'销售净利率为: ',Mean)
if Mean > MAX :
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python
603676_main_report.xls 销售净利率为: 0.11420714285714285
603707_main_report.xls 销售净利率为: 0.2703142857142857
603811_main_report.xls 销售净利率为: 0.1928285714285714
605116_main_report.xls 销售净利率为: 0.24859
605507_main_report.xls 销售净利率为: 0.16988571428571425
for ended
近五年平均净利润最高的股票为: 605507_main_report.xls ,其平均净利润率为: 0.6181214285714286 %
(base) D:\CODE\python\HQU-ProfessionalPractice3>
OUTLINE
TIMELINE
main*
Ln 21, Col 26 Spaces: 4 UTF-8 CRLF Python 3.11.5


run4.py
import pandas as pd
from statsmodels.tsa.arima.model import ARIMA
import warnings
# 预测000513股票2021年9月20-24日股票价格
# 读取股票数据
file_path = 'week13/data/000513.csv' # 替换为您的文件路径
df = pd.read_csv(file_path, encoding='GBK')
df['Date'] = pd.to_datetime(df['日期'])
df = df.sort_values(by='Date', ascending=True)
# 使用 '收盘价' 数据来训练模型
df = df['收盘价']
# 拟合ARIMA模型,这里使用(5,1,0)作为模型参数
warnings.filterwarnings("ignore")
model = ARIMA(df, order=(5,1,0))
model_fit = model.fit()
# 预测接下来5天的价格
forecast = model_fit.forecast(steps=10)
out = forecast.tolist()
# 打印预测结果
for i in [2,3,4,5,6]:
print('2021年9月', 18 + i,'日股票价格(收盘价)为: ',out[i],'元')
for ended
近五年平均净利润最高的股票为: 605507_main_report.xls ,其平均净利润率为: 0.6181214285714286 %
(base) D:\CODE\python\HQU-ProfessionalPractice3>D:/Anaconda/Anaconda3/python.exe d:/CODE/python/HQU-ProfessionalPractice3/week13/run4.py
2021年9月 20 日股票价格(收盘价)为: 39.648644892918306 元
2021年9月 21 日股票价格(收盘价)为: 39.63505566843926 元
2021年9月 22 日股票价格(收盘价)为: 39.57463767382242 元
2021年9月 23 日股票价格(收盘价)为: 39.58120845400935 元
2021年9月 24 日股票价格(收盘价)为: 39.57076614592399 元
(base) D:\CODE\python\HQU-ProfessionalPractice3>


run5.py
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import os
import numpy as np
# 应该选举哪些财务指标，选用哪种合适的聚类分析方法将股票分成三种类型。
source_path = 'week13/data/transformed/'
df = pd.DataFrame()
data = pd.DataFrame()
# 选择使用的财务指标
def fun(data):
features = ['净利润(元)', '营业总收入(元)', '净资产收益率-摊薄', '资产负债比率']
data = data[features]
data = data[data['净利润(元)'] != '--']
data = data[data['营业总收入(元)'] != '--']
data = data[data['净资产收益率-摊薄'] != '--']
data = data[data['资产负债比率'] != '--']
return data
# 数据预处理：转换百分比和标准化
def pre(data):
data['净资产收益率-摊薄'] = data['净资产收益率-摊薄'].str.rstrip('%').astype('float') / 100.0
data['资产负债比率'] = data['资产负债比率'].str.rstrip('%').astype('float') / 100.0
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
data_scaled_df = pd.DataFrame(data_scaled, columns=data.columns) # 将NumPy数组转换回DataFrame
D:\CODE\python\HQU-ProfessionalPractice3>D:/Anaconda/Anaconda3/Scripts/activate
(base) D:\CODE\python\HQU-ProfessionalPractice3>conda activate base
(base) D:\CODE\python\HQU-ProfessionalPractice3>D:/Anaconda/Anaconda3/python.exe d:/CODE/python/HQU-ProfessionalPractice3/week13/run5.py
D:\Anaconda\Anaconda3\Lib\site-packages\sklearn\cluster_kmeans.py:1412: FutureWarning: The default value of `n_init` will change from 10 to 'auto' in 1.4. Se
lue of `n_init` explicitly to suppress the warning
super()._check_params_vs_input(X, default_n_init=10)
已将聚类结果保存到 './week13/out.xlsx'
(base) D:\CODE\python\HQU-ProfessionalPractice3>



净利润(元)	营业总收入(元)	净资产收益率	户均持比	Cluster
1.922448	1.178784	0.404975	1.265215	1
0.230731	0.053688	-0.33484	1.325287	2
3.793167	3.197085	1.331273	1.256163	1
2.251656	2.052309	0.646485	1.289079	1
1.65068	0.966889	0.374404	1.303892	1
-0.04405	-0.11074	-0.42655	1.480818	2
2.730553	3.09076	0.955251	1.465183	1
2.461227	2.000246	0.82991	1.270152	1
1.595455	0.954225	0.411089	1.182101	1
-0.26975	-0.09149	-0.52438	1.061955	2
1.798654	2.803361	0.533372	0.998591	1
1.589326	1.802795	0.444717	0.996945	1
0.733892	0.835955	0.019781	1.010935	1
-0.50772	-0.13946	-0.62526	0.973081	2
2.004739	2.246535	0.722912	0.903956	1
1.771431	1.239935	0.618971	0.921237	1
0.988779	0.432254	0.20015	0.852112	1
-0.60432	-0.37846	-0.66501	0.840591	2
1.095342	1.550106	0.304091	0.723738	1
0.966323	0.676552	0.23072	0.596186	1
0.185	0.088898	-0.19727	0.650498	2
-0.73799	-0.50582	-0.73838	0.596186	2
0.59866	0.956204	-0.05359	0.305698	1
0.583485	0.469593	-0.05053	0.292531	1
-0.09209	-0.04398	-0.41433	0.182261	0
-0.7371	-0.54467	-0.75061	0.316396	2
0.86951	1.121907	0.212378	0.31475	1
0.628121	0.566692	0.077866	0.188844	1
-0.16752	0.030869	-0.35318	0.387166	2
-0.85299	-0.53758	-0.79035	0.381406	2
0.233583	0.90566	-0.08722	0.153459	1
0.010156	0.357375	-0.32784	-0.14937	0
-0.4877	-0.14097	-0.54884	-0.15678	0
-0.91366	-0.60793	-0.83009	-0.31313	0
-0.15455	1.069228	-0.12084	0.136178	1
-0.28654	0.654896	-0.23701	0.147698	2
-0.54213	0.138945	-0.48464	0.127126	0
-0.91859	-0.54149	-0.80258	-0.15349	0
-0.52858	0.894378	-0.49381	-0.24401	0
-0.26976	0.411699	-0.26759	-0.1864	0
-0.4014	-0.17453	-0.3593	-0.23002	0