1. PnL curve drawing:

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

df = pd.read\_excel('PnL.xlsx')

# Suppose there is daily rate of return data, expressed in list form

CSI300\_return = df.iloc[i, n1] # i refers to date index, n1 refers to corresponding column

date = df.iloc[i, n2] # n2 refers to corresponding column

# Calculate the cumulative return for each day

CSI300\_cum\_simple\_returns = df.iloc[i, n3]

strategy\_cum\_simple\_returns = df.iloc[i, n4]

# Plotting the Cumulative Yield Curve

plt.figure(figsize=(20, 6))

plt.plot(date, CSI300\_cum\_simple\_returns, linestyle='-', color='b', label='benchmark')

plt.plot(date, strategy\_cum\_simple\_returns, linestyle='-', color='r', label='strategy')

# range(1, len(CSI300\_cum\_simple\_returns) + 1)

# plt.plot(range(1, len(cum\_return) + 1), cum\_return, linestyle='-', color='r')

plt.xlabel('Days')

plt.ylabel('Cumulative Simple Returns')

plt.title('Cumulative Simple Returns Over Time')

plt.legend()

plt.grid(True)

plt.show()

2. Portfolio Daily Yield Matching

This code is used to change to match the stock name and the corresponding yield in the two excel sheets.

import pandas as pd

# Import two tables and identify the corresponding stock position

df1 = pd.read\_excel('stock\_portfolio.xlsx')

df2 = pd.read\_excel('stock\_return\_table.xlsx')

date = df1.iloc[:, 0]

date1 = df1.iloc[:, 0][1]

table1 = []

table2 = []

for i in range(len(date)):

portfolio = df1.iloc[i, 1:]

for n in range(100):

stock = portfolio[n]

table1.append(stock)

daily\_return = df2[stock][i]

table2.append(daily\_return)

print(table2)

# Convert the corresponding list into a dataframe

import pandas as pd

def list\_to\_dataframe(data\_list, num\_per\_row):

# Slice the list into chunks of length num\_per\_row

data\_chunks = [data\_list[i:i+num\_per\_row] for i in range(0, len(data\_list), num\_per\_row)]

# Create a DataFrame using a list builder

df = pd.DataFrame((chunk + [None]\*(num\_per\_row - len(chunk))) for chunk in data\_chunks)

return df

num\_per\_row = 100

result\_df = list\_to\_dataframe(table2, num\_per\_row)

print(result\_df)

# output file to desktop

import os

import pandas as pd

desktop\_path = os.path.join(os.path.expanduser("~"), "Desktop")

output\_file\_name = "return\_ratio.xlsx"

output\_file\_path = os.path.join(desktop\_path, output\_file\_name)

result\_df.to\_excel(output\_file\_path, index=False)

3. Four-factor assignment

# PEG convert

import pandas as pd

df = pd.read\_excel('/Users/wadechen/Desktop/data/PEG.xlsx')

def map\_values(value):

if value < 0:

return 5

elif 0 <= value < 0.5:

return 1

elif 0.5 <= value < 1:

return 2

elif 1 <= value < 2:

return 3

elif 2 <= value < 3:

return 4

else:

return 5

def modify\_dataframe\_values(df):

modified\_df = df.applymap(map\_values)

return modified\_df

modified\_df = modify\_dataframe\_values(df)

import os

desktop\_path = os.path.join(os.path.expanduser("~"), "Desktop")

output\_file\_name = "PEG\_convert.xlsx"

output\_file\_path = os.path.join(desktop\_path, output\_file\_name)

modified\_df.to\_excel(output\_file\_path, index=False)

import pandas as pd

rsi = pd.read\_excel('/Users/wadechen/Desktop/data/RSI.xlsx')

def map\_values(value):

if 0 < value <= 10 :

return 5

elif 10 < value <= 30:

return 4

elif 30 < value <= 50:

return 3

elif 50 < value <= 70:

return 2

elif 70 < value <= 100:

return 1

else:

return 0

def modify\_dataframe\_values(df):

modified\_df = df.applymap(map\_values)

return modified\_df

modified\_df1 = modify\_dataframe\_values(rsi)

import os

desktop\_path = os.path.join(os.path.expanduser("~"), "Desktop")

output\_file\_name = "RSI\_convert.xlsx"

output\_file\_path = os.path.join(desktop\_path, output\_file\_name)

modified\_df1.to\_excel(output\_file\_path, index=False)

import pandas as pd

beta = pd.read\_excel('/Users/wadechen/Desktop/data/beta.xlsx')

def map\_values(value):

if 1.5 <= value:

return 5

elif 1.2 <= value < 1.5:

return 4

elif 1 <= value < 1.2:

return 3

elif 0.8 <= value < 1:

return 2

else:

return 1

def modify\_dataframe\_values(df):

modified\_df = df.applymap(map\_values)

return modified\_df

modified\_df2 = modify\_dataframe\_values(beta)

import os

desktop\_path = os.path.join(os.path.expanduser("~"), "Desktop")

output\_file\_name = "beta\_convert.xlsx"

output\_file\_path = os.path.join(desktop\_path, output\_file\_name)

modified\_df2.to\_excel(output\_file\_path, index=False)

import pandas as pd

EI = pd.read\_excel('/Users/wadechen/Desktop/data/EI.xlsx')

def map\_values(value):

if 0 < value < 0.75:

return 1

elif 0.75 <= value < 1:

return 2

elif 1 <= value < 1.5:

return 5

elif 1.5 <= value < 2:

return 4

elif 2 <= value < 2.5:

return 3

else:

return 1

def modify\_dataframe\_values(df):

modified\_df = df.applymap(map\_values)

return modified\_df

modified\_df3 = modify\_dataframe\_values(EI)

import os

desktop\_path = os.path.join(os.path.expanduser("~"), "Desktop")

output\_file\_name = "EI\_convert.xlsx"

output\_file\_path = os.path.join(desktop\_path, output\_file\_name)

modified\_df3.to\_excel(output\_file\_path, index=False)