"""

Robert Gorman

Week 5 Homework

"""

import pandas as pd

import yfinance as yf

import numpy as np

import statsmodels.api as sm

#stock portfolio

tickers = ['AAPL', 'MSFT', 'GOOGL', 'AMZN', 'WMT']

start\_date = '2017-01-01'

end\_date = '2023-01-01'

stock\_data = pd.concat([yf.download(ticker, start=start\_date, end=end\_date)['Adj Close'] for ticker in tickers], axis=1, keys=tickers)

pe\_ratio = stock\_data / stock\_data.shift(1)

pe\_ratio = pe\_ratio.fillna(0.0)

#ranking stocks based on pe ratio - #1

pe\_ratio\_rank = pe\_ratio.groupby([pe\_ratio.index.year, pe\_ratio.index.month]).rank(pct=True)

#find the stocks with >= 90th or <=10th percentile

top\_portfolio = pe\_ratio\_rank >= 0.9

bottom\_portfolio = pe\_ratio\_rank <= 0.1

#equal-weighted portfolio returns

port\_returns = (stock\_data.pct\_change() \* np.logical\_xor(top\_portfolio, bottom\_portfolio)).sum(axis=1)

sp500\_data = yf.download('^GSPC', start=start\_date, end=end\_date)['Adj Close']

excess\_returns = port\_returns - sp500\_data.pct\_change()

print('\nMonthly Return of Equal-Weight Portfolio')

print('\n',port\_returns)

#drop/fill missing values so # of observations are the same

portfolio\_returns\_clean = port\_returns.dropna()

excess\_returns\_clean = excess\_returns.fillna(0)

#print to confirm

print (len(portfolio\_returns\_clean))

print (len(excess\_returns\_clean))

#regression for average portfolio return

model\_portfolio = sm.OLS(portfolio\_returns\_clean, sm.add\_constant(np.ones\_like(portfolio\_returns\_clean)))

results\_portfolio = model\_portfolio.fit().get\_robustcov\_results(cov\_type= 'HC0')

#regression for average excess portfolio return in excess of sp500 index returns

model\_excess = sm.OLS(excess\_returns\_clean, sm.add\_constant(np.ones\_like(excess\_returns\_clean)))

results\_excess = model\_excess.fit().get\_robustcov\_results(cov\_type= 'HC0')

# Print or analyze the regression results

print("\nRegression for Average Portfolio Return:")

print(results\_portfolio.summary())

print("\nRegression for Average Excess Portfolio Return:")

print(results\_excess.summary())

pe\_ratio\_rank\_reset = pe\_ratio\_rank.stack().reset\_index().rename(columns={0: 'ret\_top'})

print(pe\_ratio\_rank\_reset.columns)

#convert year/month to date

pe\_ratio\_rank\_reset['Year'] = pe\_ratio\_rank\_reset['Date'].dt.year

pe\_ratio\_rank\_reset['Month'] = pe\_ratio\_rank\_reset['Date'].dt.month

#download factor data

Factor = pd.read\_excel(r"C:\Users\rdg83\OneDrive - Rutgers University\Course Investment Portfolio Management\Week 5 Stuff\Factors-1.xlsx")

Factor['Year'] = Factor['Year'].astype(int)

Factor['Month'] = Factor['Month'].astype(int)

#print to see exact column names

print(Factor.columns)

#merge with right column names

prank\_top\_factor1 = pd.merge(pe\_ratio\_rank\_reset, Factor, left\_on=['Year', 'Month'], right\_on=['Year', 'Month'])

#excess returns

prank\_top\_factor1['ret\_top\_rf'] = prank\_top\_factor1['ret\_top'] - prank\_top\_factor1['RF']

#add constant

prank\_top\_factor1 = sm.add\_constant(prank\_top\_factor1)

reg\_top\_factor1 = sm.OLS(prank\_top\_factor1['ret\_top\_rf'],

prank\_top\_factor1[['const', 'MktRF', 'SMB', 'HML', 'MOM', 'RMW', 'CMA']]

).fit().get\_robustcov\_results(cov\_type='HC0')

print("\nRegression for portfolio excess of the risk-free rate on the six factors:")

print('\n',reg\_top\_factor1.summary())

“””

Screenshots of my outputs are below in the next two pages

“””







