Math 507, Fall 2024.

Homework 1

Due: Tue, Sept 24, 2024, NO LATER than 11:30am.

For each ticker in the file 'TechTickers.scv', download its adjusted closing price for each business day between Jan 1, 2013 and Jan 1, 2022. This is your sample. All questions listed below must be answered using this sample. All mean returns and variances/covariances of the returns must be annualized assuming 250 days in a year. Assume the existence of a riskless return 0.01. All exercises must be solved using only the code provided at the bottom of the homework. You are required to submit your code as a Jupyter notebook (.ipynb).

1. In this question, you construct the efficient frontier. For parts (a)–(e), you need to use only stock returns, without a riskless asset.

   (a) 3 pts Produce the estimated vector of mean returns (sample mean returns) and the estimated covariance matrix (sample covariance matrix) and print them.

   (b) 5 pts Compute the weights of the minimal-variance portfolio and print them.

   (c) 5 pts Compute the weights of the optimal mean-variance portfolio (i.e., maximizing a linear combination of mean and variance) with the coefficient of risk aversion $\gamma = 2$. Plot the portfolio weights on a graph. Print the mean and the standard deviation of the resulting portfolio.

   (d) 6 pts Compute the weights of the optimal mean-variance portfolio in the robust setting, assuming that the true mean returns of the basic assets are within one standard deviation (the standard deviation is estimated from the sample) away from their sample means. Plot the weights of the resulting optimal portfolio and compare this graph to the one produced in part (c). Print the standard deviation of the optimal portfolio return, as well as its worst- and best-case mean return (according to the chosen intervals of possible mean returns of the basic assets). Compare these means and the standard deviation to those produced in part (c) and explain the difference between the two results.

   (e) 6 pts Compute the efficient frontier and plot it as a set $\{(f(\mu), \mu)\}$, where μ changes over a grid of 100 equidistant points in $[0, 2]$, and $f(\mu)$ is the standard deviation of the efficient portfolio with mean return μ. On the same plot, show the pairs $(\sqrt{\Sigma_{ii}}, \mu^i)$ corresponding to the standard deviations and the means of the returns of individual basic assets. Comment on where the latter pairs lie relative to the efficient frontier and why.

   (f) 6 pts Add a riskless asset to the set of available ones. Compute the weights of the market portfolio (i.e., of the optimal mutual fund), as well as the mean, standard deviation and Sharpe ratio of its return, and print them. Compute the efficient frontier for the extended market and plot it in the same coordinates and for the same values of μ as in part (e). Plot the efficient frontier from part (e) on the same graph and comment on the relationship between the two.

2. In this question, you investigate the regression interpretation of CAPM. You need to use the riskless return $R = 0.01$.

   (a) 3 pts Compute the "beta" for each basic asset, according to the CAPM formula (using the part of the formula that expresses beta through the weights of the market portfolio, which you computed in 1.f), and print the results.

(b) 6 pts Use the (ordinary least-square) linear regression model, to regress the excess returns of the individual basic assets on the excess return of the market portfolio. Recall that we denote the mean returns of the hedged assets by $\{a^i\}_i$ ("hedged" means that we subtract $\beta^i(R^M - R)$ from the return). Print the resulting $\{(a^i, \beta^i)\}$. Comment on the magnitude of $\{a^i\}$ and compare the resulting $\{\beta^i\}$ to those obtained in part (a).

```
import pandas as pd
import numpy as np
import csv
import matplotlib.pyplot as plt
import math
import yfinance as yf
import scipy.optimize
from scipy import stats

#sample optimization
def my_obj(x):
    return np.sum([y**2 for y in x])
def my_constr(x):
    return np.sum(x) - 1
constr = {'type': 'eq', 'fun': my_constr}
x0 = [1,0]
opt1 = scipy.optimize.minimize(my_obj, x0, constraints=constr, options={'maxi
print(opt1)

#sample graphs
x = [0.1*i for i in range(100)]
y = [math.exp(x[i]) for i in range(len(x))]
plt.plot(x,y)
plt.scatter([2,6],[5000,5000])
plt.show()

#sample regression
x = np.random.random(1000)
y = np.random.random(1000)
slope, intercept, r_value, p_value, std_err = stats.linregress(x,y)
print(intercept)
print(slope)
print(p_value)
print(r_value)
plt.scatter(x,y)

#sample data from yahoo finance
#read the list of tickers from a csv file and print them out
tickers_file = 'TechTickers.csv'
tickers = [];
f = open(tickers_file,"r",encoding='utf-8-sig')
for line in csv.reader(f):
    tickers.append(str(line[0]))
f.close
tickers_str = tickers[0]
```

```
for s in tickers[1:]: tickers_str=tickers_str+" "+s
## data for INTC is not available
#tickers.remove('INTC')
#downoad the prices and volumes for the previously read list of tickers
start_date = '2013-01-01'
end_date = '2022-01-01'
stock_data = yf.download(tickers_str, start=start_date, end=end_date)
```