1) Create a table showing the intercept and slope coefficients for the ten industry portfolios.

	intercept coefficient	slope coefficient
NoDur	0.369443	0.652647
Durbl	-0.415599	1.64854
Manuf	0.159771	1.16985
Enrgy	0.501719	0.96985
HiTec	-0.0640195	1.13297
Telcm	0.194691	0.900729
Shops	0.275492	0.826492
Hlth	0.237841	0.673036
Utils	0.444585	0.538086
Other	-0.387135	1.20731

2) Briefly explain the economic significance of the intercept and slope coefficients.

The slope coefficients is also known as Beta. It is the market risk premium and shows the risk return trade off at a given time. Higher the market risk premium steeper the slope and vice-versa.

The intercept of the SML line theoretically is equal to the risk-free rate. But the SML line used by us is a regression thus it is pricing error of the industry portfolio relative to the CAPM model which is also known as Alpha.

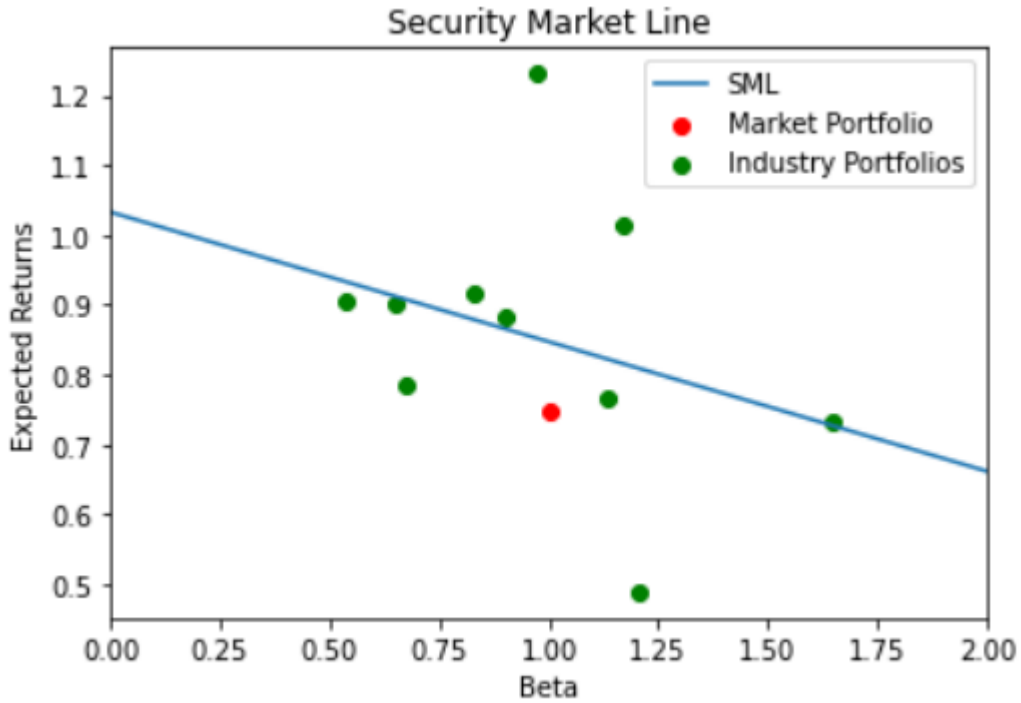
3) Calculate the mean monthly return for each of the ten industry portfolios, as well as the market portfolio.

	Mean Returns
NoDur	0.902833
Durbl	0.733333
Manuf	1.01283
Enrgy	1.23117
HiTec	0.76625
Telcm	0.881417
Shops	0.916333
Hlth	0.783833
Utils	0.907167
Other	0.489083
Market	0.748083

4) Regress the mean monthly returns of the ten industry portfolios and the market portfolio on the corresponding β's. This will give you the intercept and slope coefficients for the SML. (Warning: the results may be very different from what you would expect!)

```
sml intecept coefficient: 1.0327683682657058
sml slope coefficient: -0.18546745836573275
```

5) Use the estimated intercept and slope coefficients for the SML to plot the SML in the range of β from zero to two on the horizontal axis. Also plot the positions of the ten industry portfolios and the market portfolio. (You are NOT required to label the individual portfolios.)



6) Briefly explain the economic significance of the SML.

The purpose of SML is to decide which investment product to include in a portfolio. When the security is plotted above the SML, it is undervalued as it offers greater return compared to its risk. If the security is plotted below the SML line, it is considered overvalued as it offers a lower return to its risk. Thus, the SML can determine if an

investment portfolio gives a favourable expected return compared to its level of risk. The security market line is a representation of Capital Asset Pricing Model. It shows the expected rate of return of a security against its systematic, non-diversifiable risk. The risk refers to volatility of itself rather than the market portfolio. If market is in perfect equilibrium or with Efficient Market Theory, all securities should be on the SML. If beta is negative, then it can act as recession insurance as part of a well-diversified portfolio. The SML line consist of risk-free rate, Expected Return and Beta of the security to market. X-axis refers to beta while Y-axis refers to expected return.

Appendix

```
# -*- coding: utf-8 -*-
"""
Created on Thu Sep 15 17:27:24 2022
@author: XuebinLi
"""
import warnings
warnings.simplefilter("ignore", UserWarning)
import glob
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.dates import DateFormatter, MinuteLocator
import datetime
from datetime import date
from datetime import timedelta
import pandas as pd
from tabulate import tabulate
from numpy.linalg import inv
import math
import asset_pricing_efficient_frontier_enhanced as AP_project1
from sklearn.linear_model import LinearRegression
pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 500)
pd.set_option('display.width', 1000)
#set up dataframes
df_industry = pd.read_excel('C:\\Users\\lixue\\OneDrive\\Desktop\\smu\\MQF\\Asset
Pricing\\lesson3\\Industry_Portfolios.xlsx',index_col=0,header=0)
df_market = pd.read_excel('C:\\Users\\lixue\\OneDrive\\Desktop\\smu\\MQF\\Asset
Pricing\\lesson3\\Market_Portfolio.xlsx',index_col=0,header=0)
#df = pd.read_excel('C:\\Users\\XuebinLi\\OneDrive - Linden Shore
LLC\\Desktop\\python\\asset_pricing_project\\Project2\\Industry_Portfolios.xlsx')
#df2 = pd.read_excel('C:\\Users\\XuebinLi\\OneDrive - Linden Shore
LLC\\Desktop\\python\\asset_pricing_project\\Project2\\Market_Portfolio.xlsx')
#Reload modules: asset_pricing_efficient_frontier_enhanced
AP_project1
#declaration
#risk-free rate
rf_rate = AP_project1.rf_rate
def market_model(data_industry,data_mkt,rf):
#first part is capm model
#portfolio risk premium
RP_minus_RF = df_industry- rf_rate
#market risk premium
RM_minus_RF = df_market - rf_rate
#regress portfolio premium over market premium
```

```
reg = LinearRegression().fit(RM_minus_RF, RP_minus_RF)
#industry beta which is the slope
industry_beta = reg.coef_
#intercept which is the alpha
industry_alpha = reg.intercept_
#create table of alpha and beta of 10 industries.
market_coefficient =
pd.DataFrame(np.concatenate((industry_alpha.reshape(1,10),industry_beta.reshape(1,10))),
index = ['intercept coefficient','slope coefficient'],
columns = data_industry.columns)
print(market_coefficient)
#this part is security market line
#average return of each of the 10 industrial portfolios
mean_industry_returns = df_industry.mean()
#average return of the market returns
mean_market_returns = df_market.mean()
#total average returns of portfolio and market
mean_industry_plus_mkt_returns = np.concatenate((mean_industry_returns,
mean_market_returns),axis=0)
#market beta= 1
market_beta = [[1]]
#10 industry beta + 1 market beta
mean_industry_plus_market_beta = np.concatenate((industry_beta, market_beta),axis=0)
#regress average portfolio + market returns over average portfolio + market betas
reg_sml = LinearRegression().fit(mean_industry_plus_market_beta,
mean_industry_plus_mkt_returns)
#beta of sml
reg_sml_coefficient_slope = reg_sml.coef_
#alpha of sml
reg_sml_intercept = reg_sml.intercept_
print("sml intecept:",reg_sml_intercept)
print("sml slope:",reg_sml_coefficient_slope[0])
#plot sml
#for loop over 0-2 x axis with constant of m and c
def my_range(start, end, step):
while start <= end:
yield start
start += step
yaxis = []
xaxis = []
for x in my_range(0, 2.1, 0.1):
stdplot = reg_sml_coefficient_slope*x + reg_sml_intercept
xaxis += [x]
yaxis += [stdplot]
plt.plot(xaxis,yaxis)
plt.xlabel("Beta")
plt.ylabel("Returns")
plt.title("Security Market Line")
plt.scatter(market_beta,mean_market_returns,c='r',label='Industry Portfolios')
plt.scatter(industry_beta,mean_industry_returns,c='g',label='Market Portfolio')
market_model(df_industry,df_market,rf_rate)
```