QF600 - Assignment 1

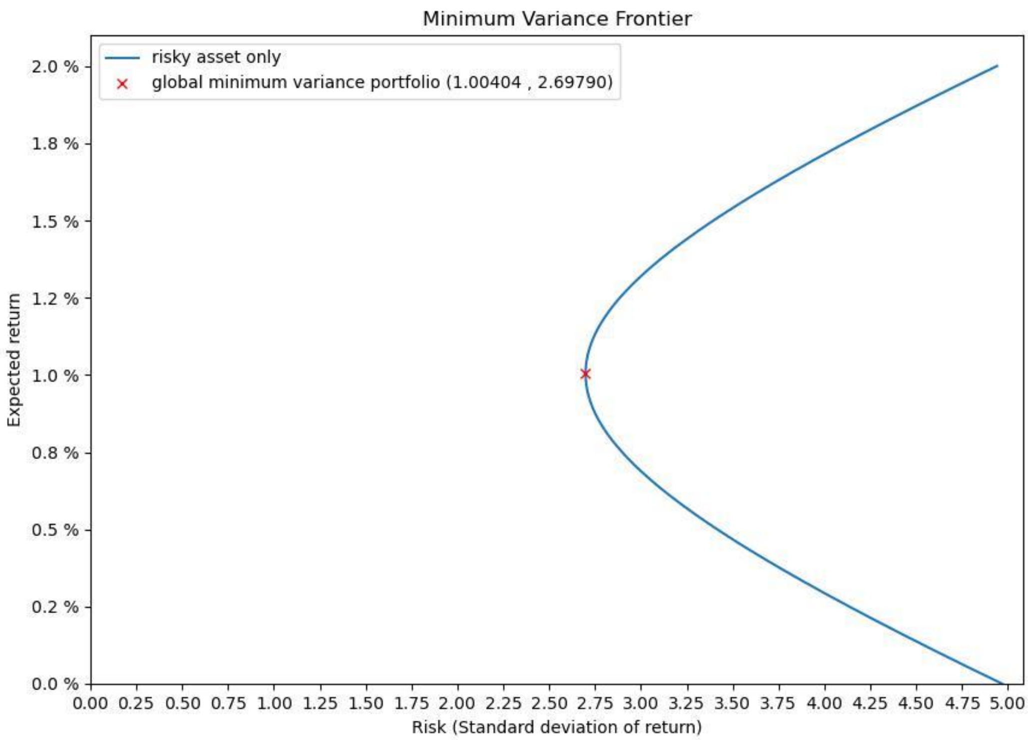
Joseph Taruna Adhika 01474863

1. Create a table showing the mean return and standard deviation of return for the ten industry portfolios. Industry_Portfolios.xlsx contains monthly nominal (net) returns for ten industry portfolios (expressed as percentages, but without "%"), over the ten-year period from Jan 2004 through Dec 2013.

	mean	std dev
NoDur	0.902833	3.345657
Durbl	0.733333	8.361852
Manuf	1.012833	5.310270
Enrgy	1.231167	6.081524
HiTec	0.766250	5.381191
Telcm	0.881417	4.448284
Shops	0.916333	4.093786
Hlth	0.783833	3.787172
Utils	0.907167	3.701763
Other	0.489083	5.582452

Table 1 : Mean and Standard Deviation of each industry portfolios

2. Plot the minimum-variance frontier (without the riskless asset) generated by the ten industry portfolios.

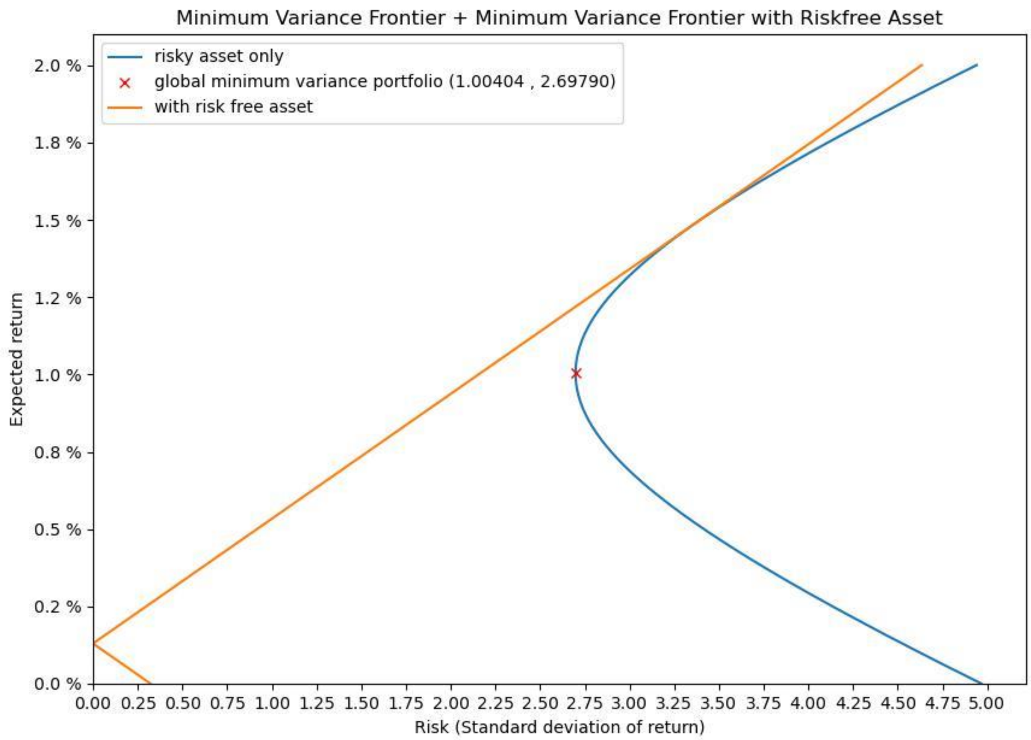


3. Briefly explain the economic significance and relevance of the minimum-variance frontier to an investor.

For a given expected return, there exists a portfolio where the volatility (represented by the standard deviation) can be minimized. Plotting these minimized-volatility portfolios on the graph for varying levels of expected return, then connecting them together will result in the **minimum-variance frontier curve.** This curve is relevant to the risk-averse investor as the investor can choose to invest in a portfolio (point on the curve) at their preferred volatility point knowing the volatility has been minimized for a certain expected return.

The global minimum variance portfolio is the point where the standard deviation is the lowest. At this point, the portfolio has the lowest amount of minimized risk (red x point)

4. Plot the efficient frontier (with the riskless asset, rf= 0.13%) on the same graph as the minimum-variance frontier generated by the ten industry portfolios.
   -



5. Briefly explain the economic significance and relevance of the efficient frontier to an investor.

The upper half of the minimum-variance frontier curve relative to the global minimum variance portfolio point is known as the efficient frontier. The shape of the efficient frontier depends on whether it includes a risk-free asset or not.

For a specific volatility value (i.e. a specific standard deviation), a risk-averse investor should choose to invest in portfolios that lie on the efficient frontier because it gives the highest expected return compared to choosing the portfolio which has the same standard deviation that lies on the lower half of the minimum-variance frontier curve.

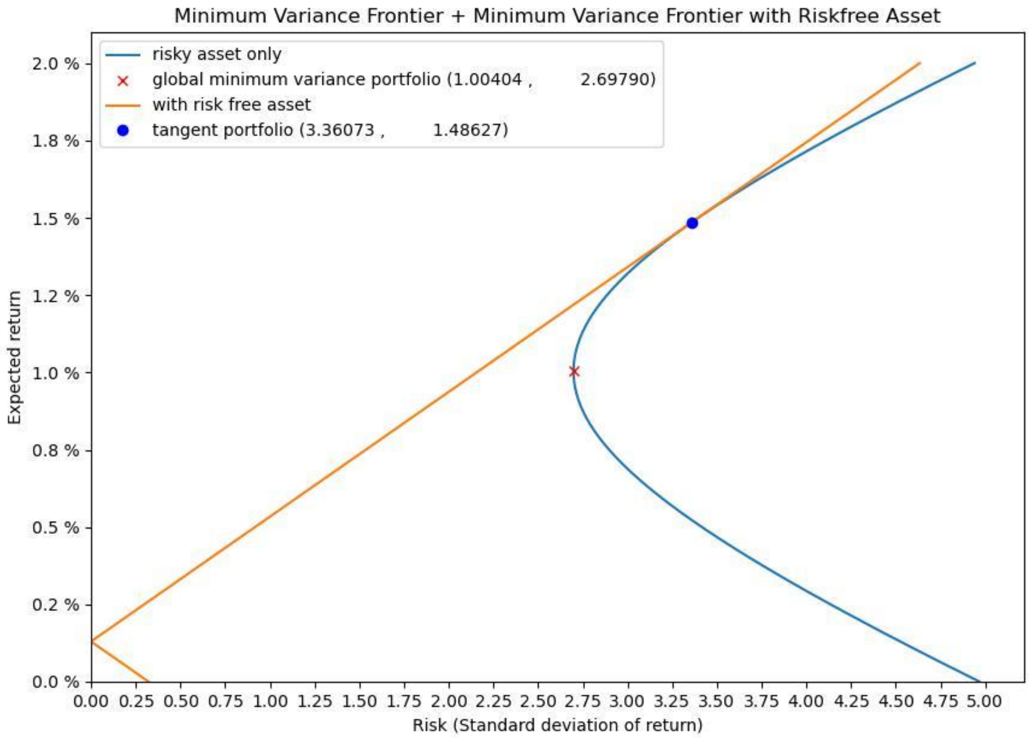
6. Calculate the Sharpe ratio for the tangency portfolio, and also the tangency portfolio weights for the ten industry portfolios.

	0
return	1.486274
var	11.294481
std_dev	3.360726
sharpe_ratio	0.403566

	Industries	Weight
0	NoDur	0.567972
1	Durbl	-0.214073
2	Manuf	0.714105
3	Enrgy	0.104087
4	HiTec	-0.363438
5	Telcm	-0.095463
6	Shops	0.991647
7	Hlth	0.075570
8	Utils	0.132643
9	Other	-0.913051

Table 2 : Sharpe ratio and weightage of the tangent portfolio

7. Briefly explain the economic significance and relevance of the tangency portfolio to an investor.



When the risk-free asset is included to the portfolio, the efficient frontier becomes a straight line which is tangent to the hyperbola curve at **the tangent portfolio**. This line (also known as the Capital Market Line / CML) intersects the Y-axis at the risk-free rate. This is to signify the inclusion of a risk-free asset (e.g. bonds, deposits) with a certain risk-free rate return and 0 standard deviation (hence risk free).

The tangency portfolio is relevant to the investor due as this is the portfolio where 100% of the investor's wealth allocated into risky assets (i.e. a portfolio that is comprised of weighted risky assets) but still has the best return-risk ratio (also known as the Sharpe ratio). Assuming that risk-free rates apply to both lending and borrowing, portfolios on the CML line above and below the tangency portfolio already involves including some form of risk-free asset but still has the same Sharpe ratio.

APPENDIX: Python Code

In [ ]:
```
import numpy as np
import scipy as sp
import math
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
```

In [ ]:
```
rawdata = pd.read_excel('Industry_Portfolios.xlsx')
workdata = rawdata.drop(['Date'],axis =1)
```

In [ ]:
```
R = workdata.mean().to_numpy()
V = workdata.cov().to_numpy()
stdev = workdata.std()
```

In [ ]:
```
resultdf = pd.concat([workdata.mean(),workdata.std()],axis=1).rename(columns = {0:'mean',1:'std dev'})
resultdf
```

In [ ]:
```
class mva():
    def __init__ (self,R,V):
        self.R = R #mean
        self.V = V #covariance matrix
        self.size =\
            len(self.R)
        self.V_inv =\
            np.linalg.inv(self.V)
        self.alpha =\
            np.dot(np.dot(self.R.T,self.V_inv),np.ones(self.size))
        self.zeta =\
            np.dot(np.dot(self.R.T,self.V_inv),self.R)
        self.delta =\
            np.dot(np.dot(np.ones(self.size),self.V_inv),np.ones(self.size))
        self.R_mv =\
            self.alpha/self.delta

    def var_mva(self,R_p):
        return \
            (1/self.delta) + (R_p- self.R_mv)**2 *(self.delta / (self.zeta * self.delta - self.alpha**2))

    def var_mva_with_riskfree(self,R_p,rf):
        return \
            ((R_p -rf)**2)/(self.zeta - 2*self.alpha*rf + self.delta * rf**2)

    def weights_calc(self,R_p):
        _multiplier_1 = ((self.delta * R_p - self.alpha)/(self.zeta * self.delta - self.alpha ** 2))
        _multiplier_2 = ((self.zeta - self.alpha * R_p )/(self.zeta * self.delta - self.alpha ** 2))
        return _multiplier_1 * (np.dot(self.V_inv,R)) +\
            _multiplier_2 * (np.dot(self.V_inv,np.ones(self.size)))

    def tangent_portfolio(self, R_p_list,rf):
        R_tg = (self.alpha * rf - self.zeta)/(self.delta * rf - self.alpha)
        var_tg = (self.zeta - 2 * self.alpha *rf + self.delta * rf**2) / (self.delta ** 2 * (rf- self.R_mv) ** 2)
        sharpe_ratio = (R_tg -rf) / np.sqrt(var_tg)
        tangent_weights = self.weights_calc(R_tg)
        return {'return': R_tg,
                'var':var_tg,
                'std_dev': np.sqrt(var_tg),
                'sharpe_ratio': sharpe_ratio,
                'tangent_weights' : tangent_weights}
```

In [ ]:
```
newmva = mva(R,V)
```

In [ ]:
```
possible_rp  = np.arange(0,2.01,0.01)
print(len(possible_rp))
var_list = []
for x in possible_rp :
    var_list.append(newmva.var_mva(x))
f1, ax1 = plt.subplots(1)
f1.set_figheight(7)
f1.set_figwidth(10)
ax1.plot(np.sqrt(var_list), possible_rp,label = "risky asset only")
ax1.plot(np.sqrt(newmva.var_mva(newmva.R_mv)),newmva.R_mv,'rx',\
         label = f"global minimum variance portfolio ({newmva.R_mv:.5f} ,\
         {np.sqrt(newmva.var_mva(newmva.R_mv)):.5f})")
ax1.set_ylim(ymin=0)
ax1.set_xlim(xmin=0)
ax1.set_xlabel("Risk (Standard deviation of return)")
ax1.set_ylabel("Expected return")
ax1.xaxis.set_major_locator(plt.MultipleLocator(0.25))
ax1.xaxis.set_major_locator(plt.MultipleLocator(0.25))
ax1.set_title("Minimum Variance Frontier")
ax1.yaxis.set_major_formatter(ticker.FormatStrFormatter("%.1f %%"))
ax1.legend()
plt.show(f1)
f1.savefig('assignment1_f1.jpg')
```

In [ ]:

```
rf = 0.13
var_list_riskfree = []
for x in possible_rp :
    var_list_riskfree.append(newmva.var_mva_with_riskfree(x,rf))
f2, ax2 = plt.subplots(1)
f2.set_figheight(7)
f2.set_figwidth(10)
ax2.plot(np.sqrt(var_list), possible_rp,label = "risky asset only")
ax2.plot(np.sqrt(newmva.var_mva(newmva.R_mv)),newmva.R_mv,'rx',\
         label = f"global minimum variance portfolio ({newmva.R_mv:.5f} , {np.sqrt(newmva.var_mva(newmva.R_mv)):.5f})")
ax2.plot(np.abs(np.sqrt(var_list_riskfree)), possible_rp, label = "with risk free asset")
ax2.set_ylim(ymin=0)
ax2.set_xlim(xmin=0)
ax2.set_xlabel("Risk (Standard deviation of return)")
ax2.set_ylabel("Expected return")
ax2.xaxis.set_major_locator(plt.MultipleLocator(0.25))
ax2.set_title("Minimum Variance Frontier + Minimum Variance Frontier with Riskfree Asset")
ax2.yaxis.set_major_formatter(ticker.FormatStrFormatter("%.1f %%"))
ax2.legend()
plt.show(f2)
f2.savefig('f2.jpg')
```

In [ ]:

```
tangent_portfolio = newmva.tangent_portfolio(possible_rp,0.13)
```

In [ ]:

```
noweights = newmva.tangent_portfolio(possible_rp,0.13)
noweights.pop('tangent_weights')
pd.DataFrame.from_dict(noweights, orient ='index')
```

In [ ]:

```
rf = 0.13
var_list_riskfree = []
for x in possible_rp :
    var_list_riskfree.append(newmva.var_mva_with_riskfree(x,rf))

f3, ax3 = plt.subplots(1)
f3.set_figheight(7)
f3.set_figwidth(10)
ax3.plot(np.sqrt(var_list), possible_rp,label = "risky asset only")
ax3.plot(np.sqrt(newmva.var_mva(newmva.R_mv)),newmva.R_mv,'rx',\
         label = f"global minimum variance portfolio ({newmva.R_mv:.5f} ,\
         {np.sqrt(newmva.var_mva(newmva.R_mv)):.5f})")
ax3.plot(np.sqrt(var_list_riskfree), possible_rp, label = "with risk free asset")
ax3.plot(tangent_portfolio['std_dev'],tangent_portfolio['return'],'bo', \
         label = f"global minimum variance portfolio ({tangent_portfolio['std_dev']:.5f} ,\
         {tangent_portfolio['return']:.5f})")
ax3.set_ylim(ymin=0)
ax3.set_xlim(xmin=0)
ax3.set_xlabel("Risk (Standard deviation of return)")
ax3.set_ylabel("Expected return")
ax3.xaxis.set_major_locator(plt.MultipleLocator(0.25))
ax3.set_title("Minimum Variance Frontier + Minimum Variance Frontier with Riskfree Asset")
ax3.yaxis.set_major_formatter(ticker.FormatStrFormatter("%.1f %%"))
ax3.legend()
plt.show(f3)
f3.savefig('f3.jpg')
```

In [ ]:

```
tangent_portfolio_weights_df = pd.concat( [pd.Series(resultdf.index), pd.Series(tangent_portfolio['tangent_weights'])],axis = 1)
    .rename(columns = {0:'Industries',1:'Weight'})
```

In [ ]:

```
tangent_portfolio_weights_df
```

In [ ]:

```
vars(newmva)
```