# Financial Instruments Coursework

**Sahil Singh**

**Candidate Number**:243655

## Q1

### a

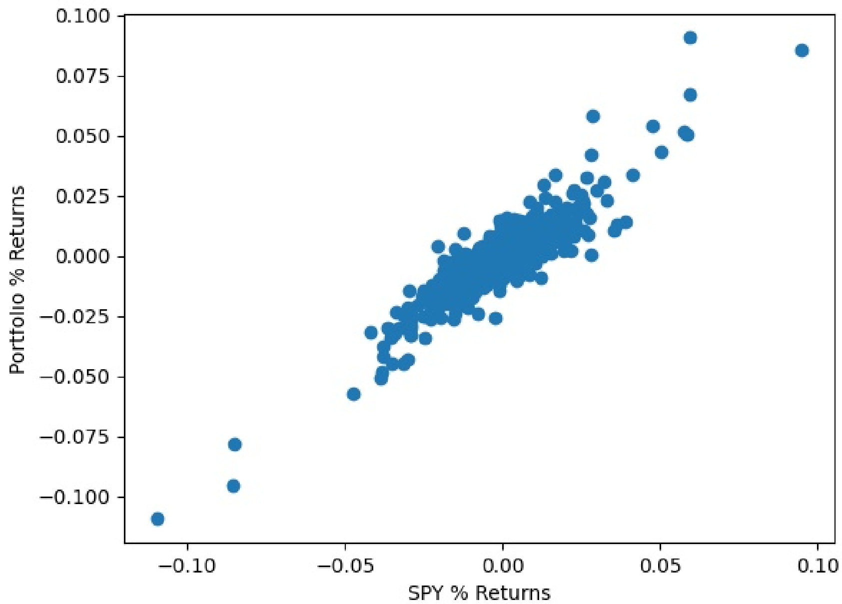


Figure 1: Scatter plot of Portfolio Returns against S&P 500

We see that there seems to be a linear relationship between returns of S&P500 and the returns of the portfolio.Also, the alpha of this portfolio seems to be zero as the intercept seems to be 0. The code written to do the calculations can be found in appendix.

### b

Table 1: Mean of all Stock Returns

| | ln_ret_DBE_mean | ln_ret_JPM_mean | ln_ret_MA_mean | ln_ret_NFLX_mean | ln_ret_SPY_mean | ln_ret_WMT_mean | portfolio_log_returns_mean |
|---|---|---|---|---|---|---|---|
| 0 | 0.000209 | 0.000648 | 0.001037 | 0.001297 | 0.000622 | 0.000681 | 0.000774 |

Table 2: Covariance Matrix

| | ln_ret_DBE | ln_ret_JPM | ln_ret_MA | ln_ret_NFLX | ln_ret_SPY | ln_ret_WMT | portfolio_log_returns |
|---|---|---|---|---|---|---|---|
| ln_ret_DBE | 0.000294 | 0.000128 | 0.000115 | 0.000069 | 0.000090 | 0.000030 | 0.000127 |
| ln_ret_JPM | 0.000128 | 0.000369 | 0.000224 | 0.000098 | 0.000177 | 0.000073 | 0.000178 |
| ln_ret_MA | 0.000115 | 0.000224 | 0.000360 | 0.000185 | 0.000187 | 0.000083 | 0.000193 |
| ln_ret_NFLX | 0.000069 | 0.000098 | 0.000185 | 0.000588 | 0.000145 | 0.000089 | 0.000206 |
| ln_ret_SPY | 0.000090 | 0.000177 | 0.000187 | 0.000145 | 0.000146 | 0.000080 | 0.000136 |
| ln_ret_WMT | 0.000030 | 0.000073 | 0.000083 | 0.000089 | 0.000080 | 0.000191 | 0.000093 |
| portfolio_log_returns | 0.000127 | 0.000178 | 0.000193 | 0.000206 | 0.000136 | 0.000093 | 0.000160 |

Table 3: Beta Calculated From Covariance Matrix

| | DBE_beta | JPM_beta | MA_beta | NFLX_beta | SPY_beta | portfolio_returns_beta |
|---|---|---|---|---|---|---|
| 0 | 0.614786 | 1.213278 | 1.278208 | 0.990476 | 1.0 | 0.928908 |

$\beta_s = \frac{\text{Cov}[R_m, R_s]}{\text{Var}[R_m]}$

where $\beta_s$ is Beta of the stock, $\text{Cov}[R_m, R_s]$ is the covariance of the stock and market index, $\text{Var}[R_m]$ is the variance of the market index.

**c**

Table 4: Portfolio Return Info

| | portfolio_annual_mean_returns | portfolio_std_returns |
|---|---|---|
| 0 | 0.193622 | 0.19971 |

$\mu_{1y} = \mu_{1d} * 250$

$\sigma_{1y} = \sigma_{1d} * \sqrt{250}$

where,

$\mu_{1y}$ and $\mu_{1d}$ are annual and daily log-returns respectively and

$\sigma_{1y}$ and $\sigma_{1d}$ are annual and daily standard deviation of log-returns respectively

# Appendix(Python Code)

## All the neccesary imports

```
import pandas as pd
import os
import statsmodels.api as sm
import yfinance as yf
```

```
import numpy as np
import matplotlib.pyplot as plt
```

## Downloading Data

```
# Downloading Data of certain stocks and S&P500 from Yahoo Finance
k = yf.download('SPY DBE JPM MA NFLX WMT',start='2016-12-31',end='2021-09-30')
k=k[[i for i in k.columns if 'Adj Close' in i]] # Remove all columns except adjusted Clos
k=k.set_axis(k.columns.map('_'.join), axis=1, inplace=False)
```

## Calculating Returns and Saving Figure

```
for i in k.columns:
    logcolname = f"ln_ret_{i[10:]}"
    ord_ret = f"%ret_{i[10:]}" # Column name for percentage returns
    k[logcolname] = np.log(k[i]/k[i].shift(1)) # Calculating log returns
    k[ord_ret] = np.exp(k[logcolname])-1 # calculating percentage returns from log returns
print(k)
# Extracting Columns Names of only stocks
per_ret=[i for i in k.columns if "%ret" in i and 'SPY' not in i]
log_ret=[i for i in k.columns if "ln_ret" in i and 'SPY' not in i]
k['portfolio_returns_percentage'] = k[per_ret].mean(axis=1) #Calculating Portfolio Returns
k['portfolio_log_returns'] = k[log_ret].mean(axis=1 # Calculating log portfolio returns
plt.scatter(k.portfolio_returns_percentage,k['%ret_SPY']) #Plotting Scatter Plot
plt.xlabel("SPY % Returns")
plt.ylabel("Portfolio % Returns")
plt.savefig("ScatterPlot.jpg")
```

## Calculating Means

```
return_info1 = pd.DataFrame() # For calculating Mean of all series
for i in cov_mat_cols:
    return_info1[f"{i}_mean"] = [k[i].mean()]
```

## Calculating Covariance and Beta

```
cov_mat_cols =[i for i in k.columns if 'ln_ret' in i or 'log_returns' in i]#Extracting Colu
cov_mat = k[cov_mat_cols].cov() # covariance matrix of markets and portfolio
info = pd.DataFrame()
for i in cov_mat.columns[:-2]:

    info[f"{i[7:]}_beta"]=[cov_mat.loc['ln_ret_SPY',i]/cov_mat.loc['ln_ret_SPY','ln_ret_SPY'
# Computing beta for portfolio returns
info[f"portfolio_returns_beta"]=[
        cov_mat.loc['ln_ret_SPY','portfolio_log_returns']/cov_mat.loc['ln_ret_SPY','ln_ret_S
print(info)
```

### Calculating Annualized Standard Deviation

```
return_info = pd.dataFrame()
return_info['portfolio_annual_mean_returns'] = [k.portfolio_log_returns.mean()*250]
return_info['portfolio_std_returns'] = [k.portfolio_log_returns.std()*250**.5]
```