```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import minimize
```

Creating data frames with statistical information

```
data = {
    'AAPL': [0.044857944, 0.009320453, -0.33517305, 5.034092588],
    'AMZN': [0.010361979, 0.009556763, -0.312865226, 4.729806047],
    'NFLX': [-0.029674829, 0.01340882, -3.101653581, 41.96801484],
    'META': [-0.006934285, 0.011581856, -1.936505944, 23.28923563],
    'GOOG': [0.029035134, 0.00837577, -0.197699834, 4.310171108]
     }


df_stats = pd.DataFrame(data,index=['Mean','St Dev','Skew','Kurt'])
print("df_stats :")
print(df_stats)
```

```
df_stats :
            AAPL      AMZN       NFLX       META      GOOG
Mean    0.044858  0.010362  -0.029675  -0.006934  0.029035
St Dev  0.009320  0.009557   0.013409   0.011582  0.008376
Skew   -0.335173 -0.312865  -3.101654  -1.936506 -0.197700
Kurt    5.034093  4.729806  41.968015  23.289236  4.310171
```

```
data = {
        'AAPL': [1, 0.660738586, 0.460041141, 0.594784944, 0.698431255],
        'AMZN': [0.660738586, 1, 0.593812086, 0.626197627, 0.679322691],
        'NFLX': [0.460041141, 0.593812086, 1, 0.515892088, 0.492697083],
        'META': [0.594784944, 0.626197627, 0.515892088, 1, 0.668023785],
        'GOOG': [0.698431255, 0.679322691, 0.492697083, 0.668023785, 1]
        }


df_corr = pd.DataFrame(data,index=['AAPL','AMZN','NFLX','META','GOOG'])
print("df_corr :")
print(df_corr)
```

```
df_corr :
          AAPL      AMZN      NFLX      META      GOOG
AAPL  1.000000  0.660739  0.460041  0.594785  0.698431
AMZN  0.660739  1.000000  0.593812  0.626198  0.679323
NFLX  0.460041  0.593812  1.000000  0.515892  0.492697
META  0.594785  0.626198  0.515892  1.000000  0.668024
GOOG  0.698431  0.679323  0.492697  0.668024  1.000000
```

Building the covariance matrix

```
df_cov= pd.DataFrame(np.dot(np.diag(df_stats.loc['St Dev'].values), np.dot(df_corr, np.dia
                     columns=['AAPL', 'AMZN', 'NFLX','META','GOOG'], index=['AAPL', 'AMZN'
print("df_cov :")
print(df_cov)
```

```
df_cov :
          AAPL      AMZN      NFLX      META      GOOG
AAPL  0.000087  0.000059  0.000057  0.000064  0.000055
AMZN  0.000059  0.000091  0.000076  0.000069  0.000054
NFLX  0.000057  0.000076  0.000180  0.000080  0.000055
META  0.000064  0.000069  0.000080  0.000134  0.000065
GOOG  0.000055  0.000054  0.000055  0.000065  0.000070
```

Creating the function to compute portfolio return and variance

```
def get_portfolio_return(array_return, weights):
    return np.sum(array_return*weights)
def get_portfolio_variance(weights, df_cov):
    return np.dot(weights.T, np.dot(df_cov, weights))
```

Now, Computing for each portfolio !

For Portfolio A-

```
weights = np.array([1.5,0,0.5, 0, 0 ])
print(f"weights : {weights}")
print(f"weights sum : {np.sum(weights)} ")
print(f"portfolio return : {get_portfolio_return(df_stats.loc['Mean'].values,weights)}")
print(f"portfolio variance : {get_portfolio_variance(weights, df_cov)}")
print(f"portfolio std dev : {np.sqrt(get_portfolio_variance(weights, df_cov))}")
print(f"portfolio range return : [{get_portfolio_return(df_stats.loc['Mean'].values,weight
print(f"Portfolio Coefficient of variation : {np.sqrt(get_portfolio_variance(weights, df_c
```

```
weights : [1.5 0.  0.5 0.  0. ]
weights sum : 2.0
portfolio return : 0.0524495015
portfolio variance : 0.0003266498560541807
portfolio std dev : 0.01807345722473099
portfolio range return : [0.03437604427526901,0.070522958724731]
Portfolio Coefficient of variation : 0.3445877788701383
```

For Portfolio B-

```
weights = np.array([0.6,0,0.15,0.15,0.1])
print(f"weights : {weights}")
print(f"weights sum : {np.sum(weights)} ")
print(f"portfolio return : {get_portfolio_return(df_stats.loc['Mean'].values,weights)}")
print(f"portfolio variance : {get_portfolio_variance(weights, df_cov)}")
print(f"portfolio std dev : {np.sqrt(get_portfolio_variance(weights, df_cov))}")
print(f"portfolio range return : [{get_portfolio_return(df_stats.loc['Mean'].values,weight
print(f"Portfolio Coefficient of variation : {np.sqrt(get_portfolio_variance(weights, df_c
```

```
weights : [0.6  0.   0.15 0.15 0.1 ]
weights sum : 1.0
portfolio return : 0.0243269127
portfolio variance : 7.469688026198356e-05
portfolio std dev : 0.008642735693169354
portfolio range return : [0.015684177006830648, 0.032969648393169355]
Portfolio Coefficient of variation : 0.3552746622537661
```

For Portfolio C -

```
weights = np.array([1.4, 0.1, -0.3, -0.3, 0.1] )
print(f"weights : {weights}")
print(f"weights sum : {np.sum(weights)} ")
print(f"portfolio return : {get_portfolio_return(df_stats.loc['Mean'].values,weights)}")
print(f"portfolio variance : {get_portfolio_variance(weights, df_cov)}")
print(f"portfolio std dev : {np.sqrt(get_portfolio_variance(weights, df_cov))}")
print(f"portfolio range return : [{get_portfolio_return(df_stats.loc['Mean'].values,weight
print(f"Portfolio Coefficient of variation : {np.sqrt(get_portfolio_variance(weights, df_c
```

```
weights : [ 1.4  0.1 -0.3 -0.3  0.1]
weights sum : 0.9999999999999999
portfolio return : 0.07772356709999999
portfolio variance : 0.0001292297784982216
portfolio std dev : 0.01136792762548309
portfolio range return : [0.06635563947451689, 0.08909149472548308]
Portfolio Coefficient of variation : 0.14626101258138335
```

```
m = np.random.random(5)*2-1
m
```

```
array([-0.70446341, -0.5290876 ,  0.58824672,  0.95361076, -0.86008477])
```

```
m /= m.sum()
m
```

```
array([ 1.2767146 ,  0.95887715, -1.06609253, -1.72824984,  1.55875063])
```

```
m.sum()
```

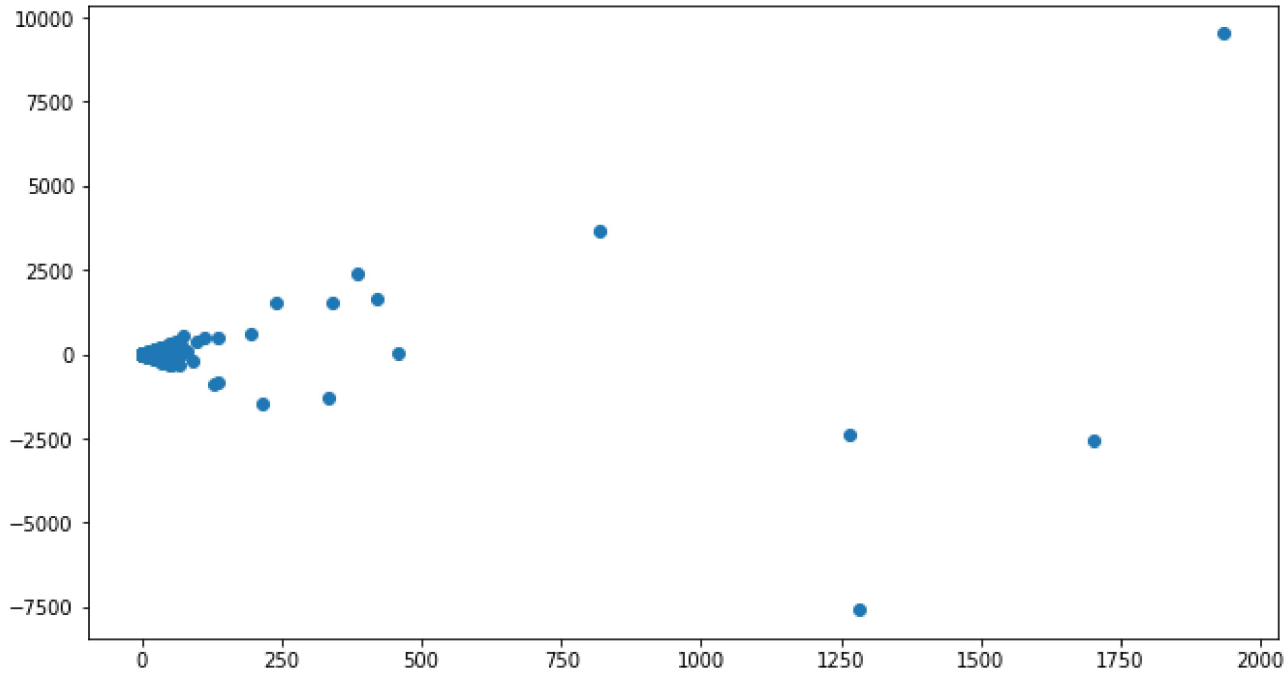
```
1.0
```

```
arr_weights = []
arr_returns = []
arr_stddev = []

for i in range(1000000):
  m = np.random.random(5)*2-1
  m /= m.sum()
  if m.sum()==1:
    arr_weights.append(m)
    arr_returns.append(get_portfolio_return(df_stats.loc['Mean'].values,m))
    arr_stddev.append(np.sqrt(get_portfolio_variance(m, df_cov)))



plt.rcParams['figure.figsize'] = (11, 6)
plt.scatter(arr_stddev, arr_returns)
plt.show()
```



Colab paid products - Cancel contracts here

✓ 0s completed at 12:51 AM