Crypto_Analysis

October 11, 2021

[ ]:
```
# imports

import panel as pn
pn.extension('plotly')
import plotly.express as px
import pandas as pd
import hvplot.pandas
import matplotlib.pyplot as plt
import numpy as np
import os
from pathlib import Path
from dotenv import load_dotenv
#from data_collection import get_crypo_from_API
import plotly.graph_objects as go

import warnings
warnings.filterwarnings('ignore')
pd.options.display.float_format = '{:.2f}'.format
```

0.1 Crypto Data

[ ]:
```
#Read the crypto data into a Pandas DataFrame

Euthereum_data = pd.read_csv(Path("Resources/ETH-USD.csv"), index_col='Date')
Doge_data = pd.read_csv(Path("Resources/DOGE-USD.csv"), index_col='Date')
Bitcoin_data = pd.read_csv(Path("Resources/BTC-USD.csv"), index_col='Date')
Sushi_data = pd.read_csv(Path("Resources/SUSHI-USD.csv"), index_col='Date')
Tether_data = pd.read_csv(Path("Resources/USDT-USD.csv"), index_col='Date')

Euthereum_data['ticker']='ETH'
Doge_data['ticker']='DOGE'
Bitcoin_data['ticker']='BTC'
Sushi_data['ticker']='SUSHI'
Tether_data['ticker']='USDT'
```

```
volume_data=pd.
↪concat([Euthereum_data,Bitcoin_data,Doge_data,Sushi_data,Tether_data],axis='rows').
↪loc[:,['Volume','ticker']].dropna()
volume_data.reset_index(inplace=True)
volume_data.head()
```

```
[ ]:          Date          Volume ticker
     0  2020-10-06 11497841885.00    ETH
     1  2020-10-07 10537119715.00    ETH
     2  2020-10-08 11511016796.00    ETH
     3  2020-10-10 13618484324.00    ETH
     4  2020-10-11 12584512533.00    ETH
```

0.2 Data Cleaning

```
[ ]: Euthereum_data.columns = ['ETH Open', 'ETH High', 'ETH Low', 'ETH Close', 'ETH␣
      ↪Adj Close', 'ETH Volume','ticker']
     Doge_data.columns = ['DOGE Open', 'DOGE High', 'DOGE Low', 'DOGE Close', 'DOGE␣
      ↪Adj Close', 'DOGE Volume','ticker']
     Bitcoin_data.columns = ['BTC Open', 'BTC High', 'BTC Low', 'BTC Close', 'BTC␣
      ↪Adj Close', 'BTC Volume','ticker']
     Sushi_data.columns = ['SUSHI Open', 'SUSHI High', 'SUSHI Low', 'SUSHI Close',␣
      ↪'SUSHI Adj Close', 'SUSHI Volume','ticker']
     Tether_data.columns = ['USDT Open', 'USDT High', 'USDT Low', 'USDT Close',␣
      ↪'USDT Adj Close', 'USDT Volume','ticker']
     Tether_data.head()
```

```
[ ]:             USDT Open  USDT High  USDT Low  USDT Close  USDT Adj Close  \
     Date
     2020-10-06       1.00       1.01      1.00        1.00            1.00
     2020-10-07       1.00       1.01      1.00        1.00            1.00
     2020-10-08       1.00       1.01      0.99        1.00            1.00
     2020-10-09        NaN        NaN       NaN         NaN             NaN
     2020-10-10       1.00       1.00      1.00        1.00            1.00

                   USDT Volume ticker
     Date
     2020-10-06 36772723041.00   USDT
     2020-10-07 28509871425.00   USDT
     2020-10-08 33458865269.00   USDT
     2020-10-09            NaN   USDT
     2020-10-10 41298643279.00   USDT
```

```
[ ]: Bitcoin_data['Total Traded'] = Bitcoin_data['BTC Open'] * Bitcoin_data['BTC␣
      ↪Volume']
     Bitcoin_data.dropna(inplace=True)
     #Bitcoin_data.drop(columns='ticker', inplace=True)
```

```
Euthereum_data['Total Traded'] = Euthereum_data['ETH Open'] *␣
↪Euthereum_data['ETH Volume']
Euthereum_data.dropna(inplace=True)
#Euthereum_data.drop(columns='ticker', inplace=True)

Doge_data['Total Traded'] = Doge_data['DOGE Open'] * Doge_data['DOGE Volume']
Doge_data.dropna(inplace=True)
#Doge_data.drop(columns='ticker', inplace=True)

Sushi_data['Total Traded'] = Sushi_data['SUSHI Open'] * Sushi_data['SUSHI␣
↪Volume']
Sushi_data.dropna(inplace=True)
#Sushi_data.drop(columns='ticker', inplace=True)

Tether_data['Total Traded'] = Tether_data['USDT Open'] * Tether_data['USDT␣
↪Volume']
Tether_data.dropna(inplace=True)
#Tether_data.drop(columns='ticker', inplace=True)

Trade_data=pd.
↪concat([Euthereum_data,Bitcoin_data,Doge_data,Sushi_data,Tether_data],axis='rows').
↪loc[:,['Total Traded','ticker']].dropna()
Trade_data.reset_index(inplace=True)
Trade_data.head()
```

[ ]:

```
         Date     Total Traded ticker
0  2020-10-06 4071820280429.81    ETH
1  2020-10-07 3594123813264.87    ETH
2  2020-10-08 3937860957008.13    ETH
3  2020-10-10 4976227755171.94    ETH
4  2020-10-11 4667953551688.09    ETH
```

[ ]:

```
Crypto_data = pd.concat([Euthereum_data, Doge_data, Bitcoin_data, Sushi_data,␣
↪Tether_data], axis="columns", join="inner")
Crypto_data.head(5)
```

[ ]:

```
            ETH Open  ETH High  ETH Low  ETH Close  ETH Adj Close  \
Date
2020-10-06    354.14    355.50   338.52     341.02         341.02
2020-10-07    341.09    342.59   335.53     342.12         342.12
2020-10-08    342.09    352.80   336.50     351.46         351.46
2020-10-10    365.40    378.27   365.35     370.97         370.97
2020-10-11    370.93    377.25   369.83     375.14         375.14

               ETH Volume  ticker     Total Traded  DOGE Open  DOGE High  …  \
Date                                                                      …
```

```
2020-10-06 11497841885.00     ETH 4071820280429.81       0.00       0.00  …
2020-10-07 10537119715.00     ETH 3594123813264.87       0.00       0.00  …
2020-10-08 11511016796.00     ETH 3937860957008.13       0.00       0.00  …
2020-10-10 13618484324.00     ETH 4976227755171.94       0.00       0.00  …
2020-10-11 12584512533.00     ETH 4667953551688.09       0.00       0.00  …

            ticker  Total Traded  USDT Open  USDT High USDT Low  USDT Close  \
Date
2020-10-06   SUSHI   88604157.04       1.00       1.01     1.00        1.00
2020-10-07   SUSHI   63278301.71       1.00       1.01     1.00        1.00
2020-10-08   SUSHI   70507985.47       1.00       1.01     0.99        1.00
2020-10-10   SUSHI   76103160.01       1.00       1.00     1.00        1.00
2020-10-11   SUSHI   56330917.90       1.00       1.00     1.00        1.00

            USDT Adj Close    USDT Volume  ticker   Total Traded
Date
2020-10-06            1.00 36772723041.00    USDT 36832699352.28
2020-10-07            1.00 28509871425.00    USDT 28557882048.48
2020-10-08            1.00 33458865269.00    USDT 33494197830.72
2020-10-10            1.00 41298643279.00    USDT 41346012822.84
2020-10-11            1.00 36190854082.00    USDT 36224547767.15

[5 rows x 40 columns]
```

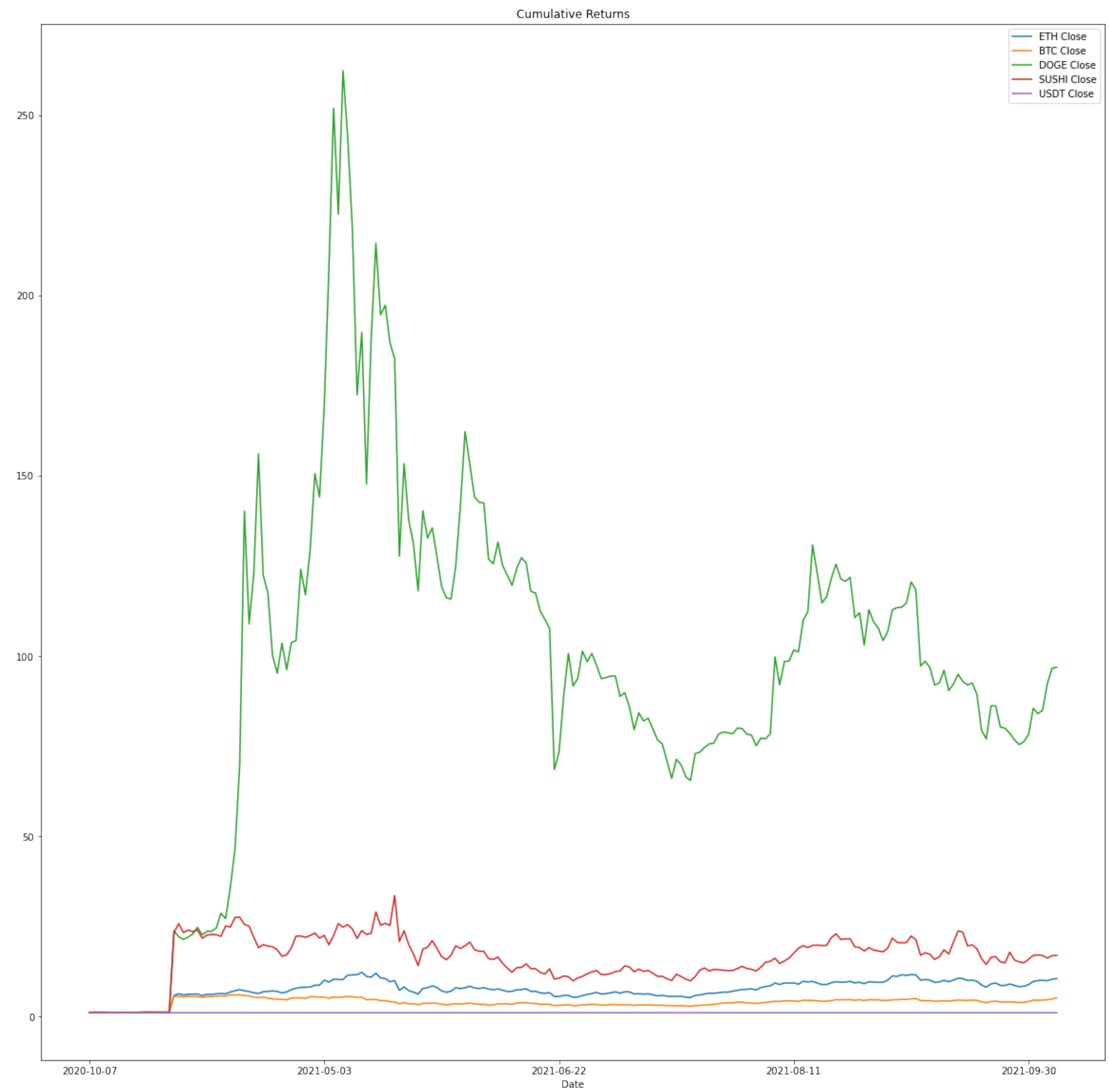
[ ]:
```
Crypto_data = Crypto_data.dropna()
Crypto_data.head(2)
```

[ ]:
```
            ETH Open  ETH High  ETH Low  ETH Close  ETH Adj Close  \
Date
2020-10-06    354.14    355.50   338.52     341.02         341.02
2020-10-07    341.09    342.59   335.53     342.12         342.12

               ETH Volume  ticker     Total Traded  DOGE Open  DOGE High  …  \
Date                                                                      …
2020-10-06 11497841885.00     ETH 4071820280429.81       0.00       0.00  …
2020-10-07 10537119715.00     ETH 3594123813264.87       0.00       0.00  …

            ticker  Total Traded  USDT Open  USDT High USDT Low  USDT Close  \
Date
2020-10-06   SUSHI   88604157.04       1.00       1.01     1.00        1.00
2020-10-07   SUSHI   63278301.71       1.00       1.01     1.00        1.00

            USDT Adj Close    USDT Volume  ticker   Total Traded
Date
2020-10-06            1.00 36772723041.00    USDT 36832699352.28
2020-10-07            1.00 28509871425.00    USDT 28557882048.48
```

[2 rows x 40 columns]

[ ]:
```
Crypto_Daily_Returns=Crypto_data[['ETH Close','BTC Close','DOGE Close','SUSHI␣
↪Close','USDT Close']].pct_change().dropna()
(1+Crypto_Daily_Returns).cumprod().plot(figsize=(20, 20), title="Cumulative␣
↪Returns")
```

[ ]: <AxesSubplot:title={'center':'Cumulative Returns'}, xlabel='Date'>



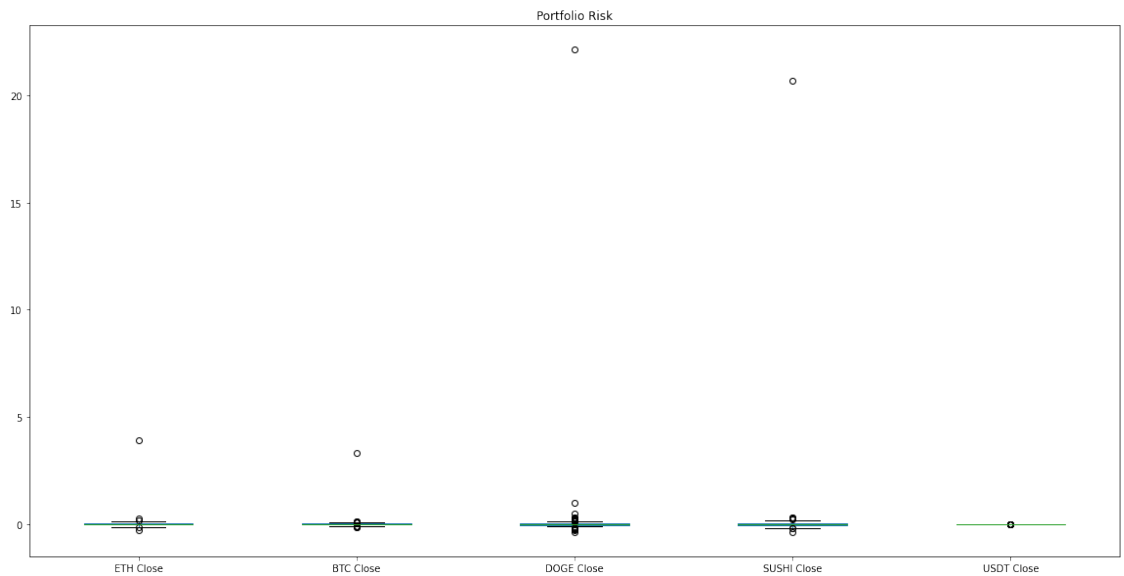
[ ]:
```

```

[ ]:
```
Crypto_Daily_Returns.std().sort_values(ascending=False)
```

[ ]: DOGE Close   1.54
     SUSHI Close   1.44
     ETH Close   0.28
     BTC Close   0.23
     USDT Close   0.00
     dtype: float64

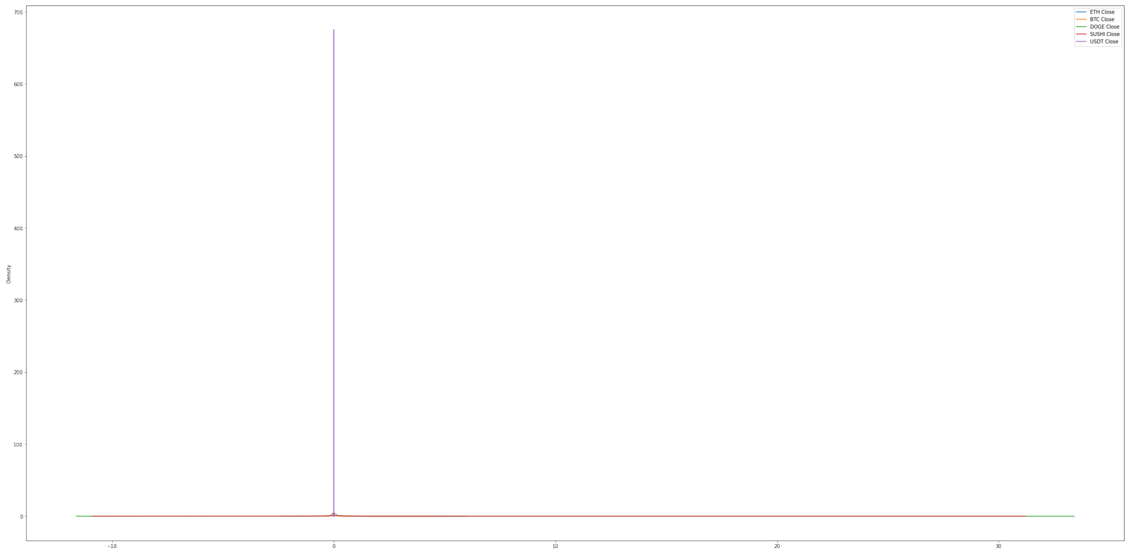
[ ]:
```
Crypto_Daily_Returns.plot.box(figsize=(20, 10), title="Portfolio Risk")
#doge  has the highest level of volatilty and USDT has the lowest one
```

[ ]: <AxesSubplot:title={'center':'Portfolio Risk'}>



[ ]:
```
Crypto_Daily_Returns.plot.density(figsize=(40, 20))
```

[ ]: <AxesSubplot:ylabel='Density'>




```
[ ]: annual_std = Crypto_Daily_Returns.std()* np.sqrt(365)
     sharpe_ratios=(Crypto_Daily_Returns.mean()*365)/annual_std
     sharpe_ratios
     #comment
```

```
[ ]: ETH Close      1.67
     BTC Close      1.45
     DOGE Close     1.49
     SUSHI Close    1.36
     USDT Close    -0.10
     dtype: float64
```

```
[ ]: #df.hvplot.line(x=x_var,y=y_var,xlabel =x_label,ylabel␣
      ↪=y_label,title=title,groupby=groupby)

     test=volume_data#.groupby(['Date','ticker']).sum()
     test['year-month']=test['Date'].str.slice(0,7,1)
     test['year']=test['Date'].str.slice(0,4,1)
     test

     test.hvplot.
      ↪line(x='Date',y='Volume',xlabel='Date',ylabel='Volume',title='Intraday␣
      ↪Volume',by='ticker',figsize=(200,100),groupby='year')
     #test2.hvplot.
      ↪line(x='year-month',y='Volume',xlabel='Date',ylabel='Volume',title='Intraday␣
      ↪Volume',by='ticker',figsize=(200,100))
```

```
BokehModel(combine_events=True, render_bundle={'docs_json':␣
 ↪{'3dde8e31-57e8-48c0-a898-59be2fd54862': {'defs': …
```

[ ]:
```
:DynamicMap   [year]
   :NdOverlay   [ticker]
      :Curve   [Date]   (Volume)
```

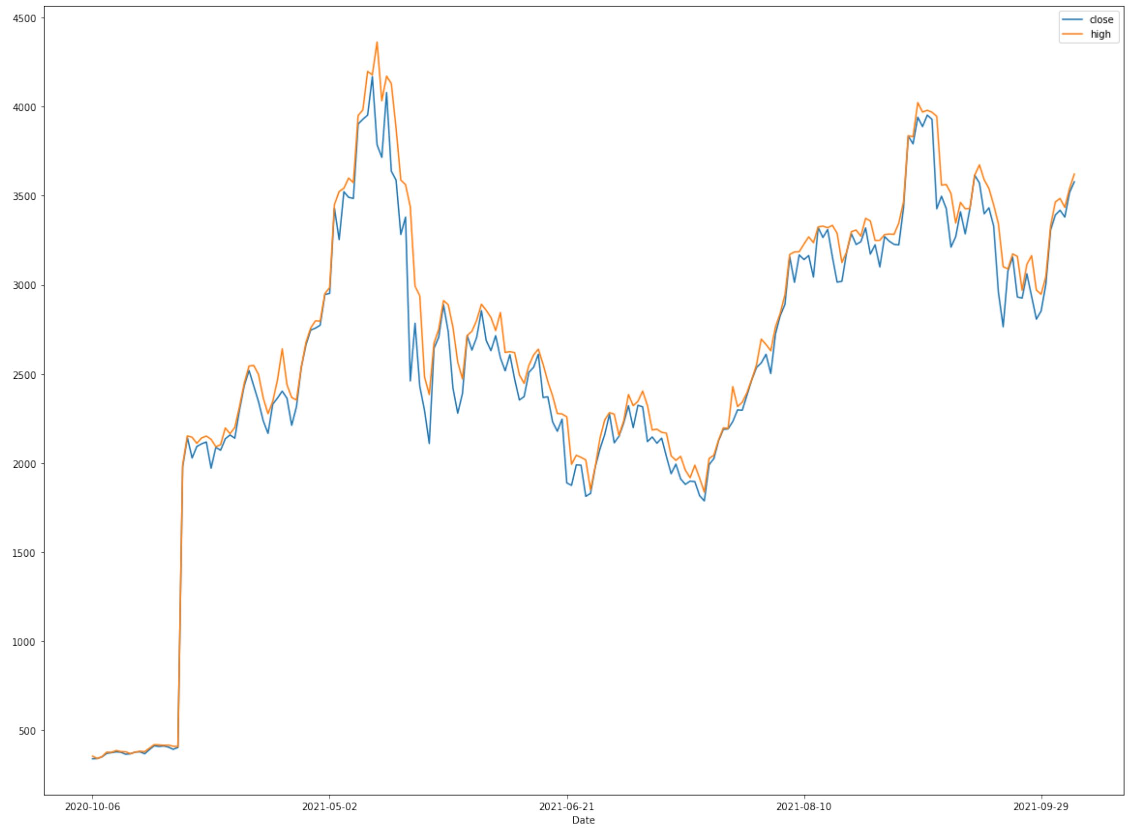
[ ]:
```
Trade_data.head()
Trade_data['year']=test['Date'].str.slice(0,4,1)

Trade_data.hvplot.line(x='Date',y='Total Traded',xlabel='Date',ylabel='Daily␣
 ↪Traded Total',title='Intraday␣
 ↪Traded',by='ticker',figsize=(200,100),groupby='year')
```

```
BokehModel(combine_events=True, render_bundle={'docs_json':␣
 ↪{'33e2dce5-ac96-49e7-bcd2-26c43b671cda': {'defs': …
```

[ ]:
```
:DynamicMap   [year]
   :NdOverlay   [ticker]
      :Curve   [Date]   (Total Traded)
```

0.3 we can see that the volatility is related to the traded volume. the bigger coins have highest trading activities

[ ]:
```
s_test= Crypto_data.loc[:,['ETH Close','ETH High']].dropna()
plt.figure(figsize=(20,15))

s_test['ETH Close'].plot(label='close')
s_test['ETH High'].plot(label='high')
plt.legend(loc='upper right')
plt.show()
```



1 What is the optimal Portfolio for reducing exposure to volatility or to risk

in this section we compute the daily returns of the close prices for each asset and annualized the covariance matrix

[ ]:
```
col=['ETH Close','BTC Close','DOGE Close','USDT Close','SUSHI Close']
df=Crypto_data[col]
df.rename(columns={'ETH Close':'ETH','BTC Close':'BTC','DOGE Close':
 ↪'DOGE','USDT Close':'USDT','SUSHI Close':'SUSHI'},inplace=True)
df.index=pd.DatetimeIndex(df.index)

daily_returns=df.pct_change().dropna()
variance_matrix=len(daily_returns.index)*daily_returns.cov()
```

we need to loop through multiple combinations of portfolio and store the returns and volatility encounterd in each scenario

[ ]:
```
#create empty list to store all returns, volatility and weights
port_returns=[]
port_volatility=[]
```

```
port_weights=[]

#find the number of assets to assign weight to
num_assets=len(daily_returns.columns)

#find the number of scenarios
num_portfolios=10000

#compute the expected return which is the mean of the retuns
individual_returns=df.pct_change().mean()#df[(df.index=='2020-10-06')|(df.
↪index=='2021-10-06')].pct_change().mean()*100
individual_returns
```

[ ]:
```
ETH      0.02
BTC      0.02
DOGE     0.12
USDT    -0.00
SUSHI    0.10
dtype: float64
```

[ ]:
```
#we loop through each scenarios to find the weights, returns and volatility␣
↪encountered
for port in range(num_portfolios):
    weights=np.random.random(num_assets)
    weights=weights/np.sum(weights)
    port_weights.append(weights)
    returns= np.dot(weights,individual_returns)
    port_returns.append(returns)

    var=variance_matrix.mul(weights,axis=0).mul(weights,axis=1).sum().sum()
    sd=np.sqrt(var)

    ann_sd=sd*np.sqrt(len(daily_returns.index))
    port_volatility.append(ann_sd)
```

[ ]:
```
data ={'returns':port_returns,'Volatility':port_volatility}
for counter,ticker in enumerate(df.columns.to_list()):
    data[ticker+' weight'] = [w[counter] for w in port_weights]
[print(f'lenght of {len(data[item])} for {item} list') for item in list(data.
↪keys())]
```

```
lenght of 10000 for returns list
lenght of 10000 for Volatility list
lenght of 10000 for ETH weight list
lenght of 10000 for BTC weight list
lenght of 10000 for DOGE weight list
lenght of 10000 for USDT weight list
lenght of 10000 for SUSHI weight list
```

[ ]: [None, None, None, None, None, None, None]

[ ]:
```
portfolio=pd.DataFrame(data)
```

[ ]:
```
#minimum volatility:
min_vol_port=portfolio.iloc[[portfolio['Volatility'].idxmin()]]

#highest sharpe ratio

optimal_sharpe_portfolio=portfolio.loc[[((portfolio['returns']-0)/
 ↪portfolio['Volatility']).idxmax()]]
```

[ ]:
```
min_weight_df=pd.DataFrame(min_vol_port.iloc[:,2:].unstack()).replace('␣
 ↪weight','').reset_index().rename(columns={'level_0':'ticker',0:'weight'})#.
 ↪drop('level_1',axis=1)
min_weight_df['ticker']= min_weight_df['ticker'].str.replace(' weight','')

optimal_weight_df=pd.DataFrame(optimal_sharpe_portfolio.iloc[:,2:].unstack()).
 ↪replace(' weight','').reset_index().rename(columns={'level_0':'ticker',0:
 ↪'weight'})#.drop('level_1',axis=1)
optimal_weight_df['ticker']= optimal_weight_df['ticker'].str.replace('␣
 ↪weight','')
optimal_weight_df
```

[ ]:

	ticker	level_1	weight
0	ETH	4958	0.57
1	BTC	4958	0.14
2	DOGE	4958	0.01
3	USDT	4958	0.27
4	SUSHI	4958	0.01

[ ]:
```
fig=px.pie(data_frame=min_weight_df,names='ticker',values='weight')
fig.update_layout(
    title='Weight of minimum volatility portfolio',
    font=dict(size=18 ))
```

[ ]:
```
fig=px.pie(data_frame=optimal_weight_df,names='ticker',values='weight')
fig.update_layout(
    title='Weight of portfolio with highest sharpe ratio',
    font=dict(size=18 ))
```

[ ]:
```
p1=portfolio.hvplot.
 ↪scatter(x='Volatility',y='returns',xlabel='Volatility',ylabel='Expected␣
 ↪return',legend='top',height=500,width=500)
p2=optimal_sharpe_portfolio.hvplot.scatter(x='Volatility',y='returns')
p3=min_vol_port.hvplot.scatter(x='Volatility',y='returns')
p1*p2*p3
```

[ ]:
```
:Overlay
   .Scatter.I   :Scatter   [Volatility]   (returns)
   .Scatter.II  :Scatter   [Volatility]   (returns)
   .Scatter.III :Scatter   [Volatility]   (returns)
```

[ ]:
```
portfolio.hvplot(x='')
```

[ ]:
```
        ETH   BTC   DOGE  USDT  SUSHI
ETH   15.82 13.33  86.70  0.00  81.27
BTC   13.33 11.42  74.07  0.00  69.28
DOGE  86.70 74.07 493.28  0.00 458.52
USDT   0.00  0.00   0.00  0.00   0.01
SUSHI 81.27 69.28 458.52  0.01 429.27
```