

Problem Statement

Context

Investors face market risk, arising from asset price fluctuations due to economic events, geopolitical developments, and investor sentiment changes. Understanding and analyzing this risk is crucial for informed decision-making and optimizing investment strategies.

Objective

The objective of this analysis is to conduct Market Risk Analysis on a portfolio of Indian stocks using Python. It uses historical stock price data to understand market volatility and riskiness. Using statistical measures like mean and standard deviation, investors gain a deeper understanding of individual stocks' performance and portfolio variability.

Through this analysis, investors can aim to achieve the following objectives:

1. Risk Assessment: Analyze historical volatility of individual stocks and the overall portfolio.
2. Portfolio Optimization: Use Market Risk Analysis insights to enhance risk-adjusted returns.
3. Performance Evaluation: Assess portfolio management strategies' effectiveness in mitigating market risk.
4. Portfolio Performance Monitoring: Monitor portfolio performance over time and adjust as market conditions and risk preferences change.

Data Dictionary

The dataset contains weekly stock price data for 5 Indian stocks over an 8-year period. The dataset enables us to analyze the historical performance of individual stocks and the overall market dynamics.

```
In [26]: # Libraries to help with reading and manipulating data
import pandas as pd
import numpy as np

# Libraries to help with data visualization
import matplotlib.pyplot as plt
import seaborn as sns

import warnings
warnings.filterwarnings('ignore')
```

Understanding the structure of data

```
In [27]: df = pd.read_csv('Market_Risk_Data.csv') # Importing the data
```

```
In [28]: df.head() # Returns first 5 rows
```

```
Out[28]:
```

	Date	ITC Limited	Bharti Airtel	Tata Motors	DLF Limited	Yes Bank
0	28-03-2016	217	316	386	114	173
1	04-04-2016	218	302	386	121	171
2	11-04-2016	215	308	374	120	171
3	18-04-2016	223	320	408	122	172
4	25-04-2016	214	319	418	122	175

Number of rows and columns in the dataset

```
In [29]: # checking shape of the data
```

```
rows = str(df.shape[0])  
columns = str(df.shape[1])
```

```
print(f"There are {rows} rows and {columns} columns in the dataset.")
```

There are 418 rows and 6 columns in the dataset.

Datatypes of the different columns in the dataset

```
In [30]: df.info() # Concise summary of dataset
```

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 418 entries, 0 to 417  
Data columns (total 6 columns):  
#   Column          Non-Null Count  Dtype  
---  ---  
0   Date            418 non-null   object  
1   ITC Limited     418 non-null   int64  
2   Bharti Airtel   418 non-null   int64  
3   Tata Motors     418 non-null   int64  
4   DLF Limited     418 non-null   int64  
5   Yes Bank        418 non-null   int64  
dtypes: int64(5), object(1)  
memory usage: 19.7+ KB
```

There are 6 columns in the dataset. Out of which 5 have integer data type and 1 has object data type.

Check duplicate records

```
In [31]: df.duplicated().sum() # Check duplicate records
```

Out[31]: 0

There are no duplicate records in the dataset.

Statistical summary of the data

In [32]: `df.describe().T`

Out[32]:

	count	mean	std	min	25%	50%	75%	max
ITC Limited	418.0	278.964115	75.114405	156.0	224.25	265.5	304.00	493.0
Bharti Airtel	418.0	528.260766	226.507879	261.0	334.00	478.0	706.75	1236.0
Tata Motors	418.0	368.617225	182.024419	65.0	186.00	399.5	466.00	1035.0
DLF Limited	418.0	276.827751	156.280781	110.0	166.25	213.0	360.50	928.0
Yes Bank	418.0	124.442584	130.090884	11.0	16.00	30.0	249.75	397.0

Observations and Insights:

- Maximum and Minimum Stock Price of Bharti Airtel is highest in given period.
- Maximum and Minimum Stock Price of Yes Bank is lowest in given period.
- Maximum and Minimum Stock Price of Tata Motors, DLF Limited and ITC Limited is greater than Yes Bank but lower than Bharti Airtel.

Finding missing values in the dataset

In [33]: `df.isna().sum()` *# Count NaN values in all columns of dataset*

Out[33]:

Date	0
ITC Limited	0
Bharti Airtel	0
Tata Motors	0
DLF Limited	0
Yes Bank	0

dtype: int64

There are no missing values in the dataset.

Change Date column data type to datetime

In [34]: *# Convert object to DateTime*

```
df['Date'] = pd.to_datetime(df['Date']) ## Complete the code to convert Date column
```

In [35]: `df.info()` *# Concise summary of dataset*

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 6 columns):
 #   Column          Non-Null Count  Dtype
---  -
 0   Date            418 non-null   datetime64[ns]
 1   ITC Limited     418 non-null   int64
 2   Bharti Airtel   418 non-null   int64
 3   Tata Motors     418 non-null   int64
 4   DLF Limited     418 non-null   int64
 5   Yes Bank        418 non-null   int64
dtypes: datetime64[ns](1), int64(5)
memory usage: 19.7 KB

```

Stock Price Analysis

```
In [36]: df_ts = pd.read_csv('Market_Risk_Data.csv', parse_dates=True, index_col=0) # Importin
```

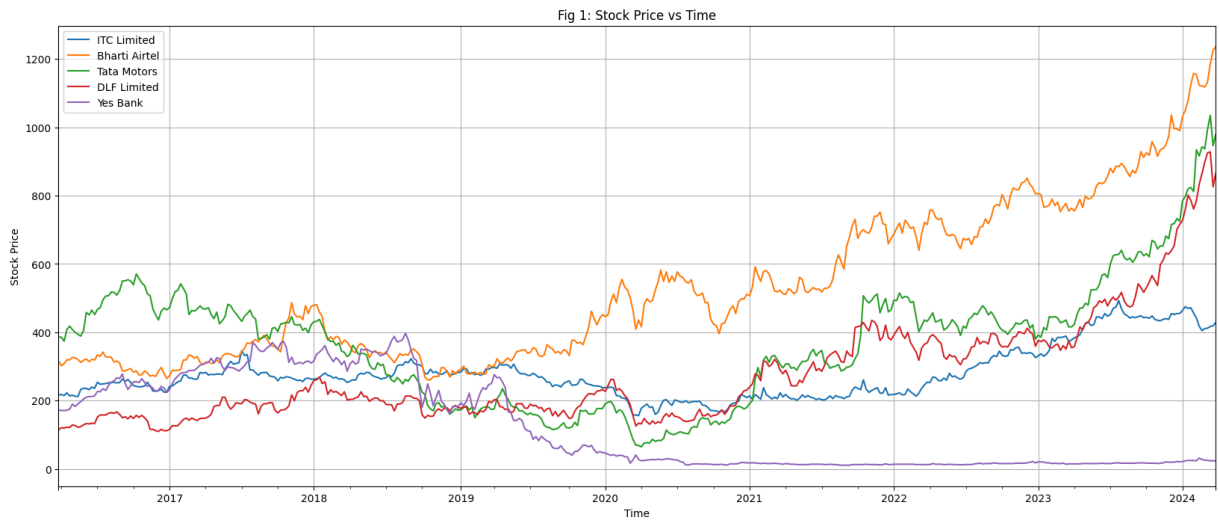
```
In [37]: df_ts.index.name = 'Time' # Renaming index name
df_ts.head() # Returns first 5 rows
```

```
Out[37]:
```

	ITC Limited	Bharti Airtel	Tata Motors	DLF Limited	Yes Bank
Time					
2016-03-28	217	316	386	114	173
2016-04-04	218	302	386	121	171
2016-04-11	215	308	374	120	171
2016-04-18	223	320	408	122	172
2016-04-25	214	319	418	122	175

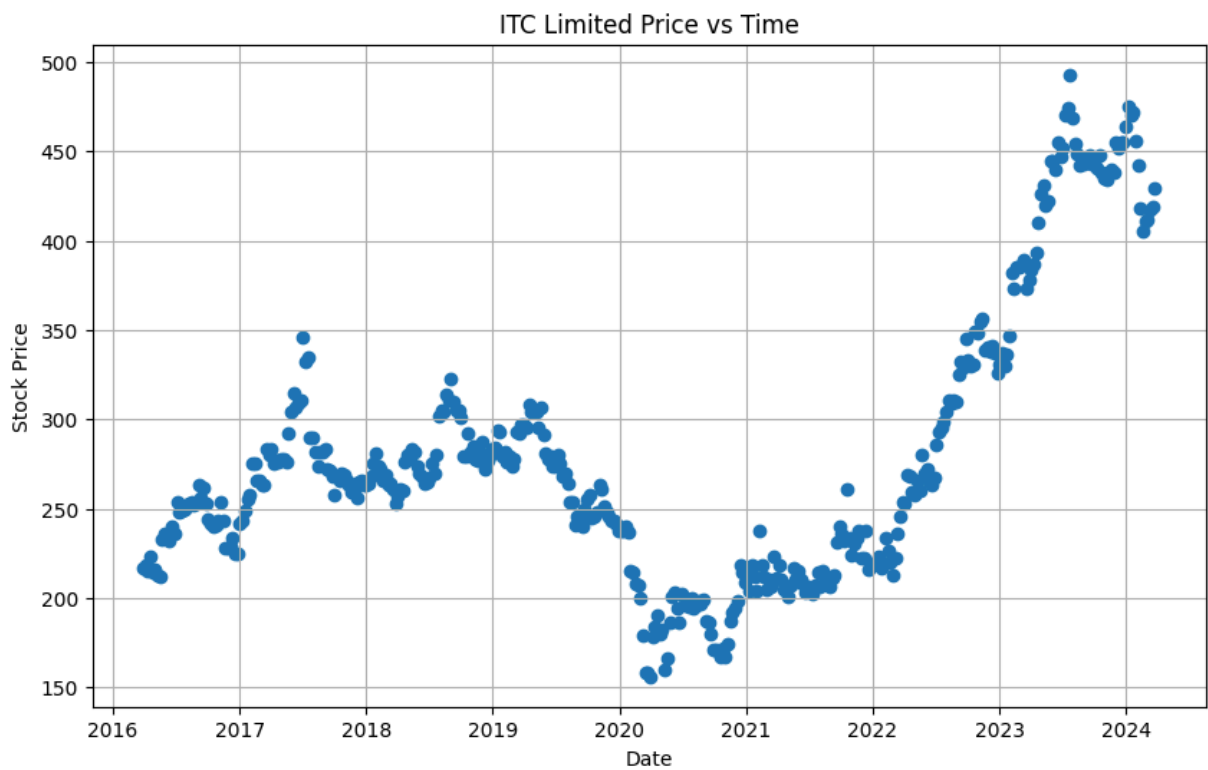
```
In [39]: # To find trend of data

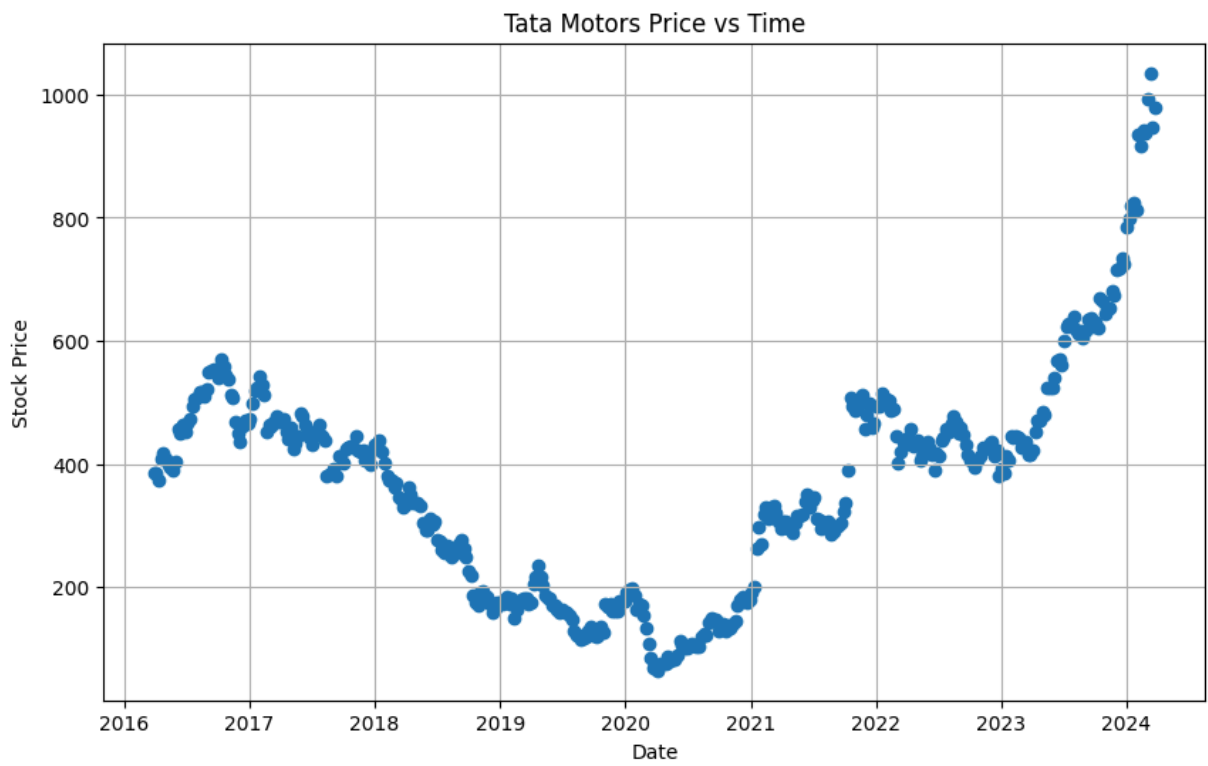
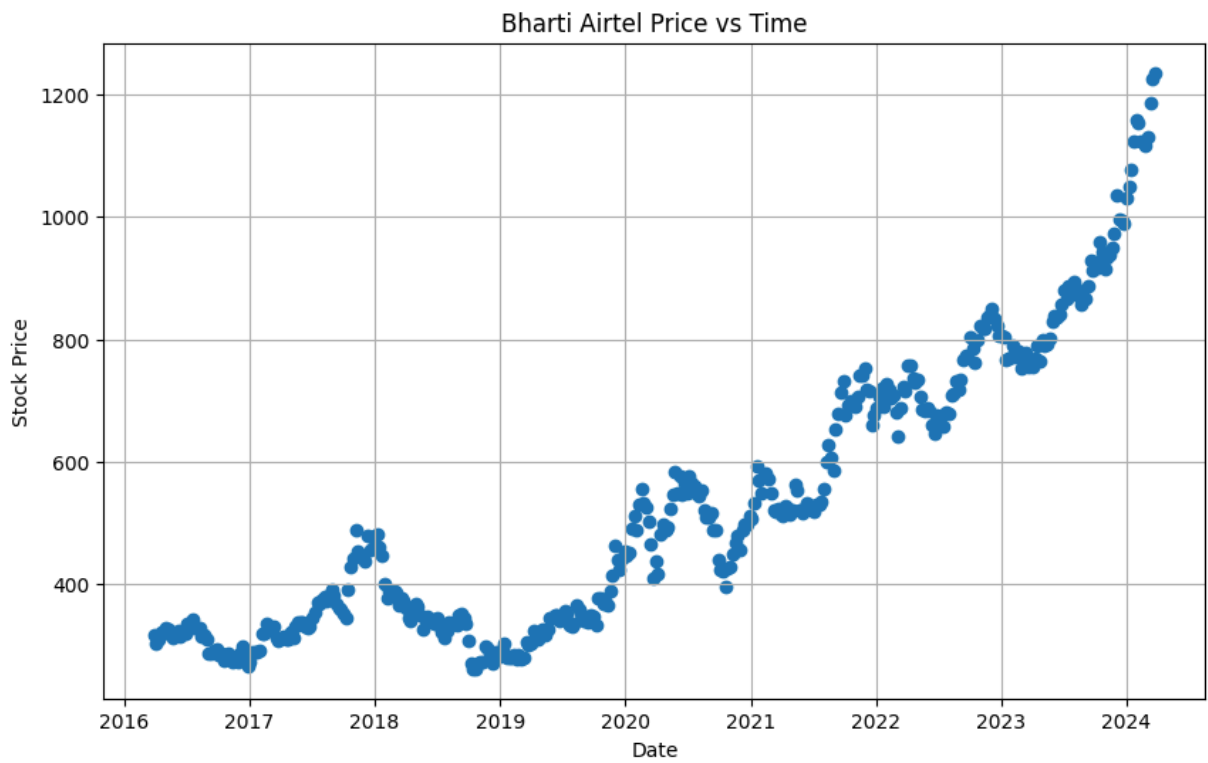
df_ts.plot(figsize=(20,8))
plt.title('Fig 1: Stock Price vs Time')
plt.xlabel('Time')
plt.ylabel('Stock Price')
plt.grid()
plt.show()
```

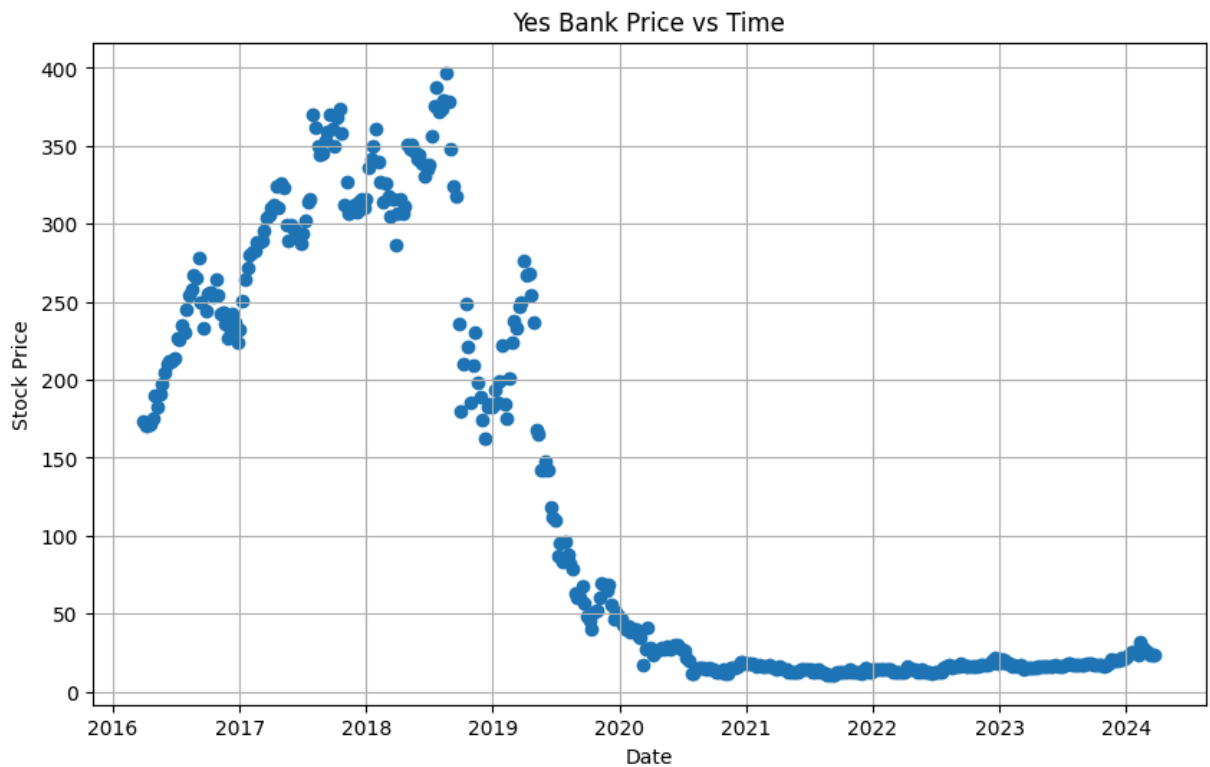
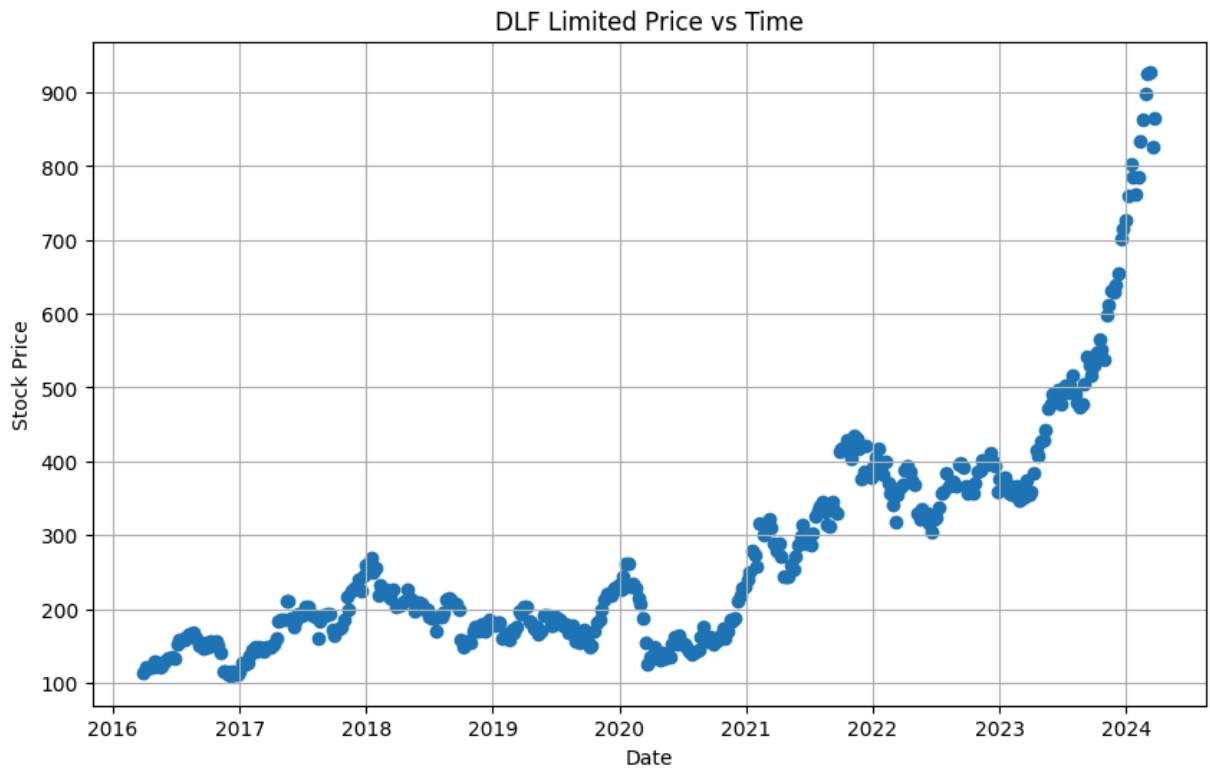


```
In [40]: numeric_columns = df.select_dtypes(np.number) # To get all the numerical columns i

for i, stock in enumerate (numeric_columns):
    plt.figure(figsize=(10, 6))
    plt.scatter(df['Date'],df[stock])
    plt.title(f'{stock} Price vs Time')
    plt.xlabel('Date')
    plt.ylabel('Stock Price')
    plt.grid(True)
    plt.show()
```







Observations and Insights:

- Stock Price of ITC Limited, Bharti Airtel, Tata Motors and DLF Limited is increasing over the time period.
- Stock Price of Yes Bank is decreasing over the time period.

Returns and Volatility Analysis

Return Calculation

```
In [41]: # Take log and diff to calculate return from prices
```

```
Return_of_Stocks = np.log(df.drop(['Date'],axis=1)).diff(axis = 0)
```

```
In [45]: print('Shape of the dataset:', Return_of_Stocks.shape) # Shape of the dataset
```

Shape of the dataset: (418, 5)

```
In [46]: Return_of_Stocks.head() # Returns first 5 rows
```

```
Out[46]:
```

	ITC Limited	Bharti Airtel	Tata Motors	DLF Limited	Yes Bank
0	NaN	NaN	NaN	NaN	NaN
1	0.004598	-0.045315	0.000000	0.059592	-0.011628
2	-0.013857	0.019673	-0.031582	-0.008299	0.000000
3	0.036534	0.038221	0.087011	0.016529	0.005831
4	-0.041196	-0.003130	0.024214	0.000000	0.017291

Average Returns

```
In [47]: ## To get the mean for the returns for all stocks
```

```
StockMeans = Return_of_Stocks.mean()  
StockMeans.sort_values()
```

```
Out[47]: Yes Bank      -0.004737  
ITC Limited      0.001634  
Tata Motors      0.002234  
Bharti Airtel    0.003271  
DLF Limited      0.004863  
dtype: float64
```

Volatility

```
In [48]: # To get the std. deviation for the returns for all stocks
```

```
StockStdDev = Return_of_Stocks.std()  
StockStdDev.sort_values()
```

```
Out[48]: ITC Limited      0.035904  
Bharti Airtel    0.038728  
DLF Limited      0.057785  
Tata Motors      0.060484  
Yes Bank         0.093879  
dtype: float64
```


Visualizing Average Returns and Volatility

```
In [49]: ## To get a dataframe for mean and std. deviation for the returns of all stocks

data = pd.DataFrame({'Average Returns': StockMeans, 'Volatility': StockStdDev})
data
```

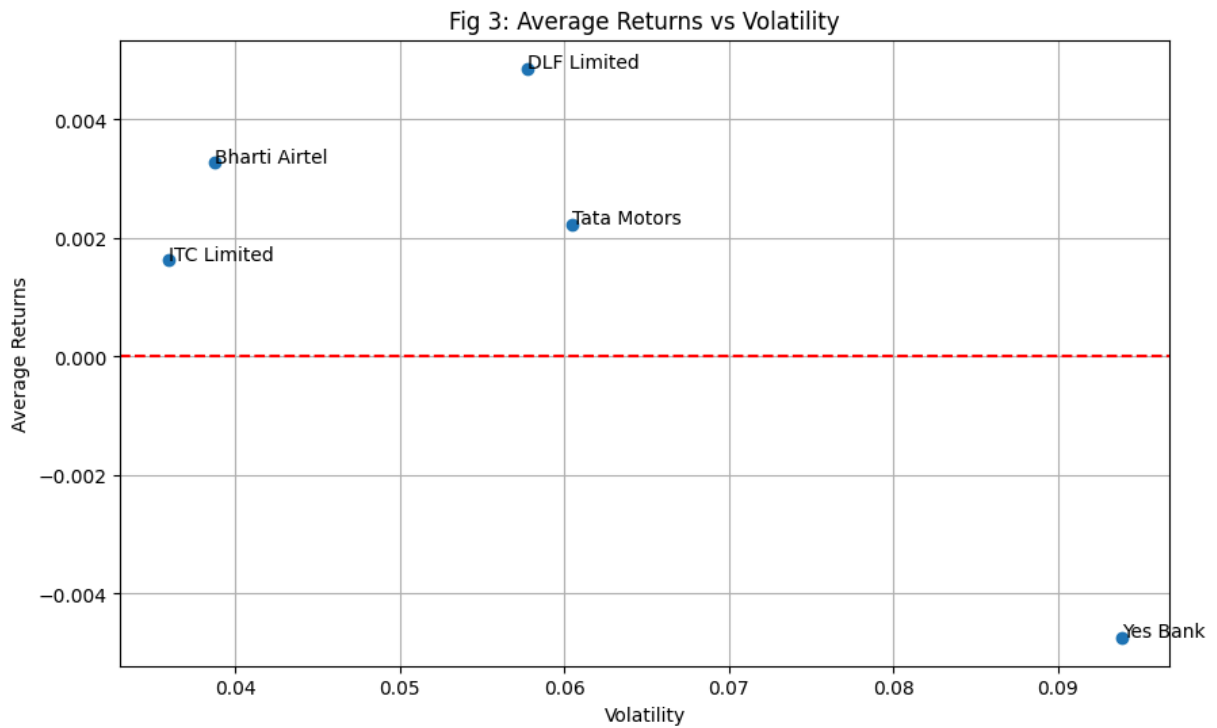
```
Out[49]:
```

	Average Returns	Volatility
ITC Limited	0.001634	0.035904
Bharti Airtel	0.003271	0.038728
Tata Motors	0.002234	0.060484
DLF Limited	0.004863	0.057785
Yes Bank	-0.004737	0.093879

```
In [50]: # To get the plot of returns vs volatility for the returns for all stocks

fig = plt.figure(figsize=(10,6))
ax = fig.add_subplot(111)
plt.scatter(data['Volatility'], data['Average Returns'])
plt.axhline(y=0, linestyle='--', color = "red")

for index, row in data.iterrows():
    ax.text(row['Volatility'], row['Average Returns'], index)
plt.xlabel('Volatility')
plt.ylabel('Average Returns')
plt.title('Fig 3: Average Returns vs Volatility')
plt.grid()
plt.show()
```



Observations and Insights:

- DLF Limited Stock is giving highest average returns followed by Bharti Airtel, Tata Motors, ITC Limited and Yes Bank Stocks.
- Yes Bank Stock has highest volatility followed by Tata Motors, DLF Limited, Bharti Airtel and ITC Limited Stocks.

Actionable Insights:

- Maximum and Minimum Stock Price of Bharti Airtel is highest in given period. Maximum and Minimum Stock Price of Yes Bank is lowest in given period.
- Stock Price of ITC Limited, Bharti Airtel, Tata Motors and DLF Limited is increasing over the time period. Stock Price of Yes Bank is decreasing over the time period.
- ITC Limited Stock is giving good average returns with lowest volatility.
- Bharti Airtel Stock is giving very good average returns however volatility is more than ITC Limited Stock.
- DLF Limited Stock is giving highest average returns however volatility is more than ITC Limited and Bharti Airtel Stocks.
- Tata Motors Stock is giving better average returns than ITC Limited Stock however volatility is more than ITC Limited, Bharti Airtel and DLF Limited Stocks.
- Yes Bank Stock is giving lowest average returns with highest volatility.

Business Recommendations:

- Investors can buy Bharti Airtel Stocks as its price is highest in given period and also increasing over the time period. It is giving very good average returns with low volatility.
- Investors can also buy ITC Limited, Tata Motors and DLF Limited Stocks as their price is increasing over the time period. DLF Limited Stock has highest average returns and ITC Limited Stock has lowest volatility.
- Investors can sell Yes Bank Stocks as its price is lowest in given period and also decreasing over the time period. It is giving lowest average returns with highest volatility.

In []: