

```
#TO DO -1
import pandas as pd

data = {
    'USN':['1BM22CS226','1BM22CS300','1BM22CS400','1BM22CS500','1BM22CS600'],
    'Name': ['Alice', 'Bob', 'Charlie', 'David','Tom'],
    'Marks': [25, 30, 35, 40,50],
}

df = pd.DataFrame(data)

print("Sample data:")

print(df.head())
```

↗ Sample data:

	USN	Name	Marks
0	1BM22CS226	Alice	25
1	1BM22CS300	Bob	30
2	1BM22CS400	Charlie	35
3	1BM22CS500	David	40
4	1BM22CS600	Tom	50

+ Code

+ Text

```
from sklearn.datasets import load_diabetes

diabetes = load_diabetes()

df = pd.DataFrame(diabetes.data, columns=diabetes.feature_names)

df['target'] = diabetes.target

print("Sample data:")

print(df.head())
```

↗ Sample data:

	age	sex	bmi	bp	s1	s2	s3 \
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142

	s4	s5	s6	target
0	-0.002592	0.019907	-0.017646	151.0
1	-0.039493	-0.068332	-0.092204	75.0
2	-0.002592	0.002861	-0.025930	141.0
3	0.034309	0.022688	-0.009362	206.0
4	-0.002592	-0.031988	-0.046641	135.0

```
import pandas as pd

# Load the dataset from a CSV file
data = pd.read_csv('/content/sample_sales_data.csv')

# Display the first few rows of the dataset
print(data.head())

# Get some basic information about the dataset
print(data.info())
```

↗

	Product	Quantity	Price	Sales	Region
0	Laptop	5	1000	5000	North
1	Mouse	15	20	300	west
2	keyboard	10	50	500	East
3	Monitor	8	200	1600	south
4	Laptop	12	950	11400	north

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8 entries, 0 to 7

```
Data columns (total 5 columns):
#   Column      Non-Null Count  Dtype
---  -
0   Product     8 non-null      object
1   Quantity     8 non-null      int64
2   Price        8 non-null      int64
3   Sales        8 non-null      int64
4   Region       8 non-null      object
dtypes: int64(3), object(2)
memory usage: 452.0+ bytes
None
```

```
df = pd.read_csv('/content/Dataset of Diabetes .csv', encoding='latin-1')

print("Sample data:")

print(df.head())
```

↗ Sample data:

	ID	No_Patien	Gender	AGE	Urea	Cr	HbA1c	Chol	TG	HDL	LDL	VLDL	\
0	502	17975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	
1	735	34221	M	26	4.5	62	4.9	3.7	1.4	1.1	2.1	0.6	
2	420	47975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	
3	680	87656	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	
4	504	34223	M	33	7.1	46	4.9	4.9	1.0	0.8	2.0	0.4	

	BMI	CLASS
0	24.0	N
1	23.0	N
2	24.0	N
3	24.0	N
4	21.0	N

```
#TO DO-2
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]

data = yf.download(tickers, start="2024-01-01", end="2024-12-30", group_by='ticker')

print("First 5 rows of the dataset:")
print(data.head())
```

↗ YF.download() has changed argument auto_adjust default to True
[*****100%*****] 3 of 3 completed

```
First 5 rows of the dataset:
```

Ticker	KOTAKBANK.NS						\
Price	Open	High	Low	Close	Volume		
Date							
2024-01-01	1906.909954	1916.899006	1891.027338	1907.059814	1425902		
2024-01-02	1905.911108	1905.911108	1858.063525	1863.008179	5120796		
2024-01-03	1861.959234	1867.952665	1845.627158	1863.857178	3781515		
2024-01-04	1869.451068	1869.451068	1858.513105	1861.559692	2865766		
2024-01-05	1863.457575	1867.852782	1839.383985	1845.577148	7799341		

Ticker	ICICIBANK.NS						\
Price	Open	High	Low	Close	Volume		
Date							
2024-01-01	983.086778	996.273246	982.541485	990.869812	7683792		
2024-01-02	988.490253	989.134730	971.883221	973.866150	16263825		
2024-01-03	976.295294	979.567116	966.777197	975.650818	16826752		
2024-01-04	977.980767	980.707295	973.519176	978.724365	22789140		
2024-01-05	979.567084	989.779158	975.402920	985.218445	14875499		

Ticker	HDFCBANK.NS						\
Price	Open	High	Low	Close	Volume		
Date							
2024-01-01	1683.017598	1686.125187	1669.206199	1675.223999	7119843		
2024-01-02	1675.914685	1679.860799	1665.950651	1676.210571	14621046		
2024-01-03	1679.071480	1681.735059	1646.466666	1650.363525	14194881		
2024-01-04	1655.394910	1672.116520	1648.193203	1668.071777	13367028		
2024-01-05	1664.421596	1681.932477	1645.628180	1659.538208	15944735		

```
tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]
```

```
data = yf.download(tickers, start="2024-01-01", end="2024-12-30", group_by='ticker')
```

```
# Access HDFC Bank data using MultiIndex
```

```
hdfc_data = data[('HDFCBANK.NS',)] # Accessing with a tuple for MultiIndex
```

```
print("\nShape of the HDFC Bank data:")
```

```
print(hdfc_data.shape)
```

```
print("\nColumn names:")
```

```
print(hdfc_data.columns)
```

```
print("\nSummary statistics for HDFC Bank:")
```

```
print(hdfc_data.describe())
```

```
hdfc_data['Daily Return'] = hdfc_data['Close'].pct_change()
```

```
[*****100%*****] 3 of 3 completed
Shape of the HDFC Bank data:
(244, 5)
```

```
Column names:
```

```
Index(['Open', 'High', 'Low', 'Close', 'Volume'], dtype='object', name='Price')
```

```
Summary statistics for HDFC Bank:
```

	Price	Open	High	Low	Close	Volume
count	244.000000	244.000000	244.000000	244.000000	244.000000	2.440000e+02
mean	1601.375295	1615.443664	1588.221245	1601.898968	2.119658e+07	2.133860e+07
std	134.648125	134.183203	132.796819	133.748372	2.133860e+07	2.133860e+07
min	1357.463183	1372.754374	1345.180951	1365.404785	8.798460e+05	8.798460e+05
25%	1475.316358	1494.072805	1460.259509	1474.564087	1.274850e+07	1.274850e+07
50%	1627.724976	1638.350037	1616.000000	1625.950012	1.686810e+07	1.686810e+07
75%	1696.474976	1711.425018	1679.250000	1697.062531	2.295014e+07	2.295014e+07
max	1877.699951	1880.000000	1858.550049	1871.750000	2.226710e+08	2.226710e+08

```
<ipython-input-26-bf418c32754a>:6: PerformanceWarning: indexing past lexsort depth may impact performance.
```

```
hdfc_data = data[('HDFCBANK.NS',)] # Accessing with a tuple for MultiIndex
```

```
<ipython-input-26-bf418c32754a>:17: SettingWithCopyWarning:
```

```
A value is trying to be set on a copy of a slice from a DataFrame.
```

```
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user\_guide/indexing.html#returning-a-view-versus-a-copy
```

```
hdfc_data['Daily Return'] = hdfc_data['Close'].pct_change()
```

```
plt.figure(figsize=(14, 12))
```

```
# Calculate Daily Returns for all tickers within the 'data' DataFrame
```

```
for ticker in tickers:
```

```
    data[ticker, 'Daily Return'] = data[ticker, 'Close'].pct_change()
```

```
# Plot Closing Prices
```

```
plt.subplot(3, 1, 1)
```

```
data['HDFCBANK.NS']['Close'].plot(title="HDFC Bank - Closing Price", color='blue')
```

```
data['ICICIBANK.NS']['Close'].plot(label="ICICI Bank", color='green')
```

```
data['KOTAKBANK.NS']['Close'].plot(label="Kotak Bank", color='red')
```

```
plt.legend()
```

```
# Plot Daily Returns using the calculated column within 'data'
```

```
plt.subplot(3, 1, 2)
```

```
data['HDFCBANK.NS', 'Daily Return'].plot(title="HDFC Bank - Daily Returns", color='blue')
```

```
data['ICICIBANK.NS', 'Daily Return'].plot(label="ICICI Bank", color='green')
```

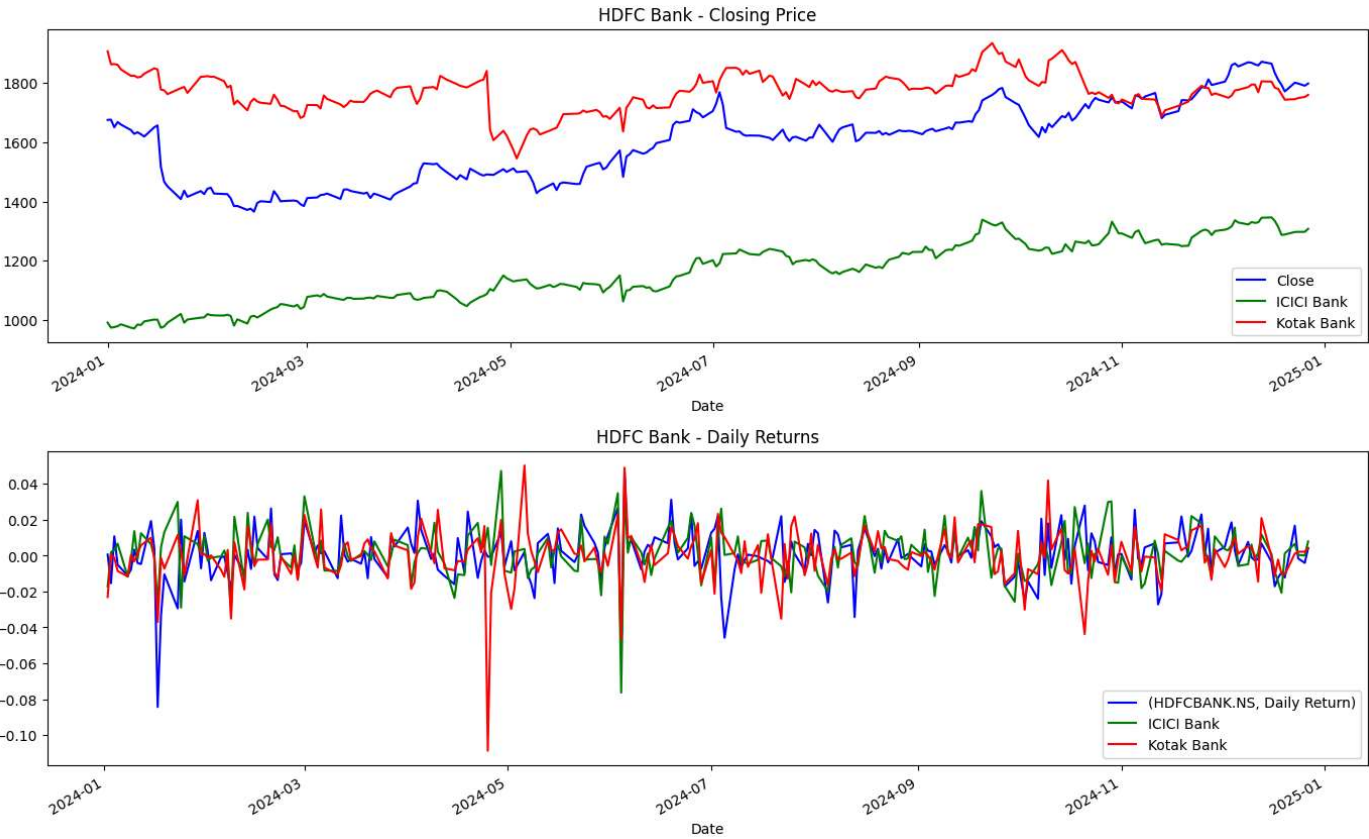
```
data['KOTAKBANK.NS', 'Daily Return'].plot(label="Kotak Bank", color='red')
```

```
plt.legend()
```

```
# Adjust layout
```

```
plt.tight_layout()
```

```
plt.show()
```



```
import yfinance as yf
import pandas as pd

# Define the ticker for HDFC Bank
ticker = "HDFCBANK.NS"

# Download historical data for 2024
hdfc_data = yf.download(ticker, start="2024-01-01", end="2024-12-30")

# Calculate daily returns for HDFC Bank
hdfc_data['Daily Return'] = hdfc_data['Close'].pct_change()

# Display the first few rows to ensure the data looks correct
print(hdfc_data.head())

# Save the HDFC Bank data with Daily Return to a CSV file
hdfc_data.to_csv('hdfc_stock_data_2024.csv')

print("\nHDFC Bank stock data saved to 'hdfc_stock_data_2024.csv'.")
```



[*****100%*****] 1 of 1 completed						Price	High	Low	Open
Ticker	HDFCBANK.NS	HDFCBANK.NS	HDFCBANK.NS	HDFCBANK.NS	HDFCBANK.NS				
Date									
2024-01-01	1675.223999	1686.125187	1669.206199	1683.017598	7119843				
2024-01-02	1676.210571	1679.860799	1665.950651	1675.914685	14621046				
2024-01-03	1650.363525	1681.735059	1646.466666	1679.071480	14194881				
2024-01-04	1668.071777	1672.116520	1648.193203	1655.394910	13367028				
2024-01-05	1659.538208	1681.932477	1645.628180	1664.421596	15944735				
Price	Daily Return								
Ticker									

```
Date
2024-01-01      NaN
2024-01-02    0.000589
2024-01-03   -0.015420
2024-01-04    0.010730
2024-01-05   -0.005116
```

```
HDFC Bank stock data saved to 'hdfc_stock_data_2024.csv'.
```

