

```
import pandas as pd

# Create a DataFrame directly from a dictionary

data = {

'Name': ['Alice', 'Bob', 'Charlie', 'David'],

'Age': [25, 30, 35, 40],

'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']

}

df = pd.DataFrame(data)

print("Sample data:")

print(df.head())
```

Sample data:

	Name	Age	City
0	Alice	25	New York
1	Bob	30	Los Angeles
2	Charlie	35	Chicago
3	David	40	Houston

```
from sklearn.datasets import load_iris

iris = load_iris()

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

df['target'] = iris.target

print("Sample data:")

print(df.head())
```

Sample data:

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	\
0	5.1	3.5	1.4	0.2	
1	4.9	3.0	1.4	0.2	
2	4.7	3.2	1.3	0.2	
3	4.6	3.1	1.5	0.2	
4	5.0	3.6	1.4	0.2	

	target
0	0
1	0
2	0
3	0
4	0

```
file_path = 'data.csv' # Ensure the file exists in the same directory

df = pd.read_csv(file_path)

print("Sample data:")

print(df.head())

print("\n")
```

Sample data:

	ID	Name	Age	City
0	1	Alice	25	New York
1	2	Bob	30	Los Angeles
2	3	Charlie	35	Chicago
3	4	David	40	Houston
4	5	Eva	28	Phoenix

```
df = pd.read_csv('/content/Mobiles Dataset (2025).csv', encoding='latin-1') # or 'iso-8859-1' or any other suitable encoding
```

```
df = pd.read_csv('content/mobiles_dataset_2023.csv', encoding='latin1', # or 'ISO-8859-1' or any other suitable encoding
```

```
print("Sample data:")
```

```
print(df.head())
```

```

Sample data:
  Company Name      Model Name Mobile Weight  RAM Front Camera \
0      Apple      iPhone 16 128GB        174g   6GB        12MP
1      Apple      iPhone 16 256GB        174g   6GB        12MP
2      Apple      iPhone 16 512GB        174g   6GB        12MP
3      Apple  iPhone 16 Plus 128GB        203g   6GB        12MP
4      Apple  iPhone 16 Plus 256GB        203g   6GB        12MP

  Back Camera  Processor Battery Capacity Screen Size \
0      48MP   A17 Bionic    3,600mAh  6.1 inches
1      48MP   A17 Bionic    3,600mAh  6.1 inches
2      48MP   A17 Bionic    3,600mAh  6.1 inches
3      48MP   A17 Bionic    4,200mAh  6.7 inches
4      48MP   A17 Bionic    4,200mAh  6.7 inches

  Launched Price (Pakistan) Launched Price (India) Launched Price (China) \
0          PKR 224,999          INR 79,999          CNY 5,799
1          PKR 234,999          INR 84,999          CNY 6,099
2          PKR 244,999          INR 89,999          CNY 6,499
3          PKR 249,999          INR 89,999          CNY 6,199
4          PKR 259,999          INR 94,999          CNY 6,499

  Launched Price (USA) Launched Price (Dubai)  Launched Year
0          USD 799          AED 2,799          2024
1          USD 849          AED 2,999          2024
2          USD 899          AED 3,199          2024
3          USD 899          AED 3,199          2024
4          USD 949          AED 3,399          2024

```

```
import pandas as pd
```

```
# Create a DataFrame directly from a dictionary
```

```

data = {
    'USN': ['101', '102', '103', '104'],
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Marks': [25, 30, 35, 40],
}

```

```
df = pd.DataFrame(data)
```

```
print("Sample data:")
```

```
print(df.head())
```

```

Sample data:
  USN      Name  Marks
0  101    Alice     25
1  102     Bob     30
2  103  Charlie     35
3  104    David     40

```

```
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  0.005299  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

```

```
file_path = 'sample_sales_data.csv' # Ensure the file exists in the same directory
```

```
df = pd.read_csv(file_path)
```

```
print("Sample data:")
```

```
print(df.head())
```

```
print("\n")
```

```

Sample data:
   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

```

```
df = pd.read_csv('/content/Dataset of Diabetes .csv', encoding='latin-1') # or 'iso-8859-1' or any other suitable encoding
```

```
print("Sample data:")
```

```
print(df.head())
```

```

Sample data:
   ID  No_Patient  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

```

```
import yfinance as yf
```

```
import pandas as pd
```

```
import matplotlib.pyplot as plt
```

```
tickers = ["RELIANCE.NS", "TCS.NS", "INFY.NS"]
```

```
# Fetch historical data for the last 1 year
```

```
data = yf.download(tickers, start="2022-10-01", end="2023-10-01",
group_by='ticker')
```

```
# Display the first 5 rows of the dataset
```

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

```
print(data.head())
```

```
print("\nShape of the dataset:")
```

```
print(data.shape)
```

```
# Check column names

print("\nColumn names:")

print(data.columns)

# Summary statistics for a specific stock (e.g., Reliance)

reliance_data = data['RELIANCE.NS']

print("\nSummary statistics for Reliance Industries:")

print(reliance_data.describe())

# Calculate daily returns

reliance_data['Daily Return'] = reliance_data['Close'].pct_change()
plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)

reliance_data['Close'].plot(title="Reliance Industries - Closing Price")

plt.subplot(2, 1, 2)

reliance_data['Daily Return'].plot(title="Reliance Industries - Daily Returns", color='orange')

plt.tight_layout()

plt.show()
reliance_data.to_csv('reliance_stock_data.csv')

print("\nReliance stock data saved to 'reliance_stock_data.csv'.")
```

```
[*****100%*****] 3 of 3 completed
<ipython-input-13-dd0e3ed665f7>:38: 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

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

First 5 rows of the dataset:

Ticker	INFY.NS				
Price	Open	High	Low	Close	Volume
2022-10-03	1326.433859	1326.433859	1302.009439	1309.289795	4943169
2022-10-04	1333.667406	1345.456933	1328.312866	1342.779663	6631341
2022-10-06	1357.434054	1371.337230	1356.588570	1366.968994	6180672
2022-10-07	1358.702230	1369.505339	1352.877991	1363.258301	3994466
2022-10-10	1339.914233	1376.222095	1339.914233	1374.014526	5274677

Ticker	TCS.NS				
Price	Open	High	Low	Close	Volume
2022-10-03	2836.902592	2861.245917	2817.011122	2827.383057	1763331
2022-10-04	2870.007907	2934.465796	2863.424823	2927.977295	2145875
2022-10-06	2946.779844	2959.093614	2929.208995	2938.207520	1790816
2022-10-07	2933.897460	2941.096371	2896.672174	2903.113037	1939879
2022-10-10	2851.111135	2961.935056	2846.375070	2953.931152	3064063

Ticker	RELIANCE.NS				
Price	Open	High	Low	Close	Volume
2022-10-03	1092.351874	1103.976472	1079.334126	1082.303101	11852723
2022-10-04	1095.229421	1104.456029	1091.735131	1102.263550	8948850
2022-10-06	1109.480507	1119.072468	1104.524564	1106.328735	13352162
2022-10-07	1102.925982	1116.286369	1102.925982	1111.010742	7714340
2022-10-10	1098.518193	1104.273467	1090.753238	1098.883667	6329527

Shape of the dataset:
(247, 15)

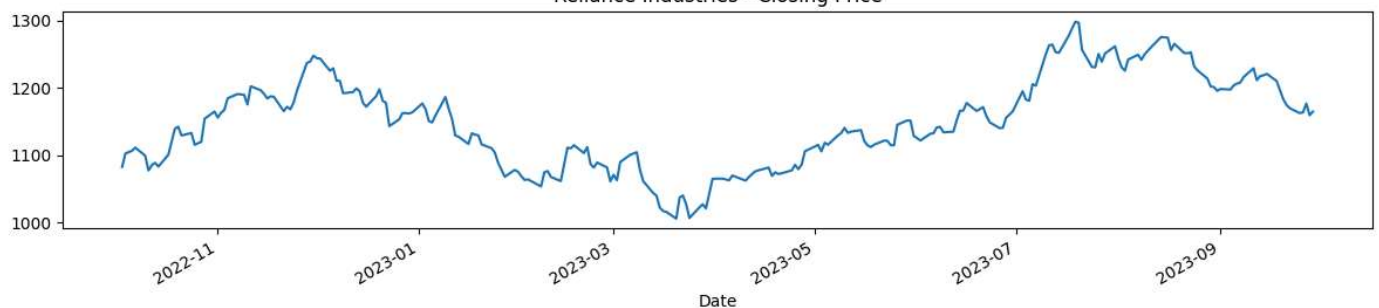
Column names:

```
MultiIndex([( 'INFY.NS', 'Open'),
( 'INFY.NS', 'High'),
( 'INFY.NS', 'Low'),
( 'INFY.NS', 'Close'),
( 'INFY.NS', 'Volume'),
( 'TCS.NS', 'Open'),
( 'TCS.NS', 'High'),
( 'TCS.NS', 'Low'),
( 'TCS.NS', 'Close'),
( 'TCS.NS', 'Volume'),
( 'RELIANCE.NS', 'Open'),
( 'RELIANCE.NS', 'High'),
( 'RELIANCE.NS', 'Low'),
( 'RELIANCE.NS', 'Close'),
( 'RELIANCE.NS', 'Volume')],
names=['Ticker', 'Price'])
```

Summary statistics for Reliance Industries:

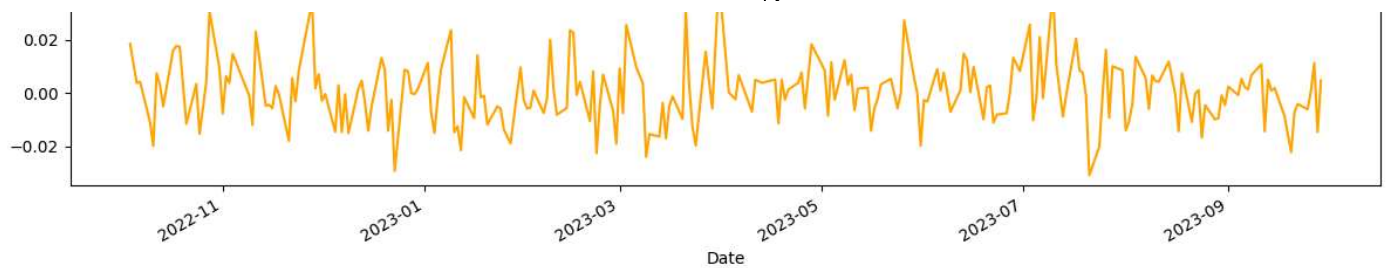
	Price	Open	High	Low	Close	Volume
count	247.000000	247.000000	247.000000	247.000000	247.000000	2.470000e+02
mean	1151.113731	1159.809204	1140.728176	1150.085766	1150.085766	1.316652e+07
std	65.667213	66.649930	65.532728	66.499556	66.499556	6.754099e+06
min	1011.732921	1014.016738	995.746138	1005.452393	1005.452393	3.370033e+06
25%	1102.777486	1107.310807	1088.640578	1101.247253	1101.247253	8.717141e+06
50%	1151.502745	1159.130787	1142.824226	1151.320068	1151.320068	1.158959e+07
75%	1198.585310	1204.999150	1189.185821	1197.370239	1197.370239	1.530302e+07
max	1292.642908	1304.518807	1277.569690	1298.055542	1298.055542	5.708188e+07

Reliance Industries - Closing Price



Reliance Industries - Daily Returns





Reliance stock data saved to 'reliance_stock_data.csv'.

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

# Define the tickers and date range
tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]
start_date = "2024-01-01"
end_date = "2024-12-30"

# Download historical data using yfinance
data = yf.download(tickers, start=start_date, end=end_date, group_by='ticker')

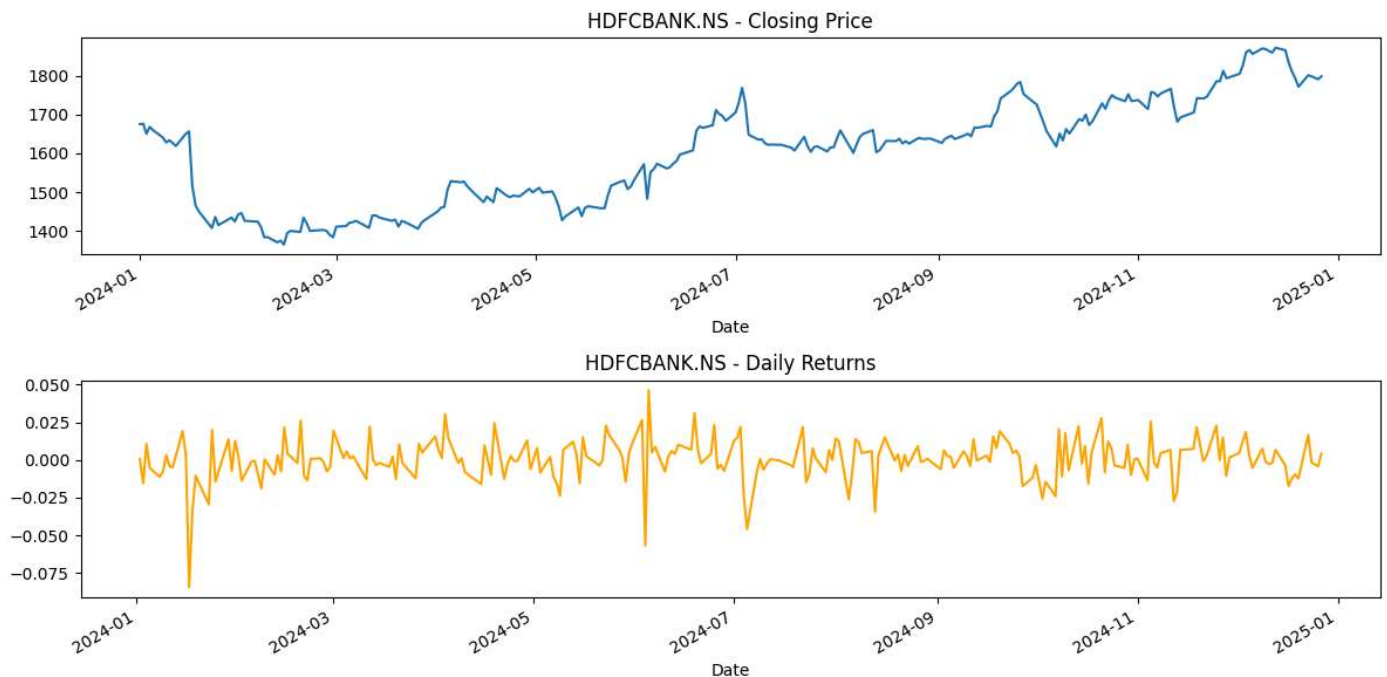
# Plotting closing prices and daily returns for each bank
for ticker in tickers:
    # Extract data for the current ticker
    ticker_data = data[ticker]

    # Calculate daily returns
    ticker_data['Daily Return'] = ticker_data['Close'].pct_change()

    # Create subplots for closing price and daily returns
    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1)
    ticker_data['Close'].plot(title=f"{ticker} - Closing Price")
    plt.subplot(2, 1, 2)
    ticker_data['Daily Return'].plot(title=f"{ticker} - Daily Returns", color='orange')
    plt.tight_layout()
    plt.show()
```

```
[*****100%*****] 3 of 3 completed  
<ipython-input-14-d2e5e8d10ae8>:19: 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
ticker_data['Daily Return'] = ticker_data['Close'].pct_change()



```
<ipython-input-14-d2e5e8d10ae8>:19: 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
ticker_data['Daily Return'] = ticker_data['Close'].pct_change()

