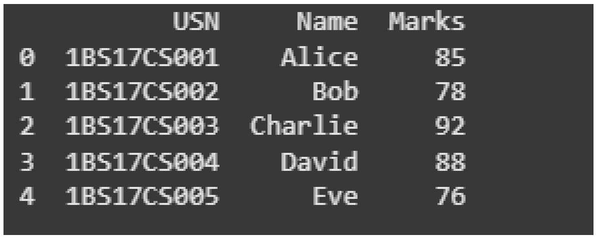
LAB 0

Data collection and preprocessing

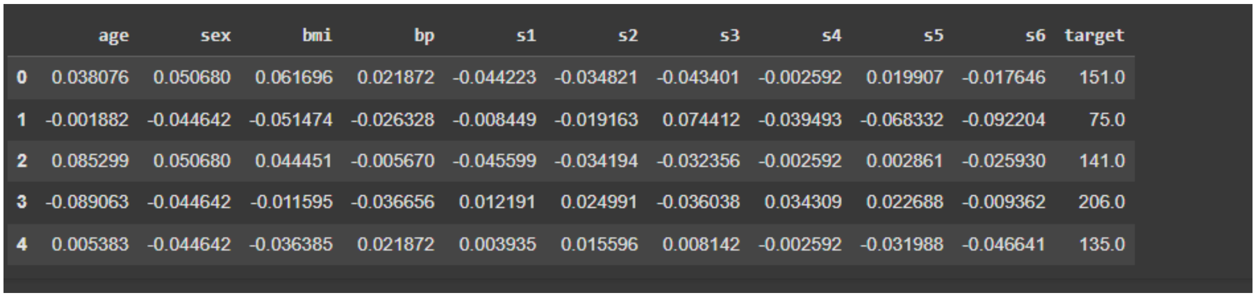
```
#Method 1

import pandas as pd

# Initializing values directly into DataFrame
data = {
    'USN': ['1BS17CS001', '1BS17CS002', '1BS17CS003', '1BS17CS004', '1BS17CS005'],
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Marks': [85, 78, 92, 88, 76]
}

# Create DataFrame
df = pd.DataFrame(data)

print(df)
```



```
          USN     Name  Marks
0  1BS17CS001    Alice     85
1  1BS17CS002      Bob     78
2  1BS17CS003  Charlie     92
3  1BS17CS004    David     88
4  1BS17CS005      Eve     76
```

```
#Method 2

import pandas as pd
from sklearn.datasets import load_diabetes

# Load the diabetes dataset
diabetes_data = load_diabetes()

# Convert the dataset into a DataFrame for better readability
df = pd.DataFrame(diabetes_data.data, columns=diabetes_data.feature_names)

# Adding the target values (diabetes progression) to the DataFrame
df['target'] = diabetes_data.target
```

# Display the DataFrame
df.head()



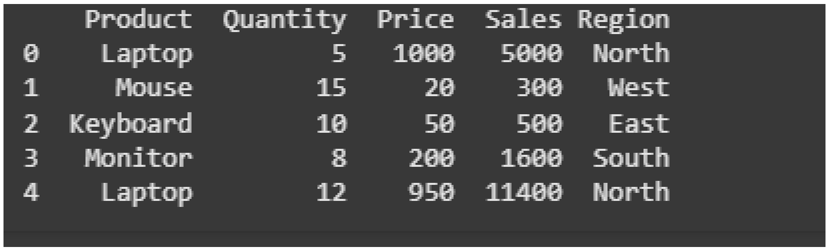
	age	sex	bmi	bp	s1	s2	s3	s4	s5	s6	target
0	0.038076	0.050680	0.061696	0.021872	-0.044223	-0.034821	-0.043401	-0.002592	0.019907	-0.017646	151.0
1	-0.001882	-0.044642	-0.051474	-0.026328	-0.008449	-0.019163	0.074412	-0.039493	-0.068332	-0.092204	75.0
2	0.085299	0.050680	0.044451	-0.005670	-0.045599	-0.034194	-0.032356	-0.002592	0.002861	-0.025930	141.0
3	-0.089063	-0.044642	-0.011595	-0.036656	0.012191	0.024991	-0.036038	0.034309	0.022688	-0.009362	206.0
4	0.005383	-0.044642	-0.036385	0.021872	0.003935	0.015596	0.008142	-0.002592	-0.031988	-0.046641	135.0

#Method 3

import pandas as pd

file_path='/content/drive/MyDrive/Lab-0/sample_sales_data.csv'

df=pd.read_csv(file_path)
# Display the first few rows of the DataFrame to verify
print(df.head())



```
    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
```

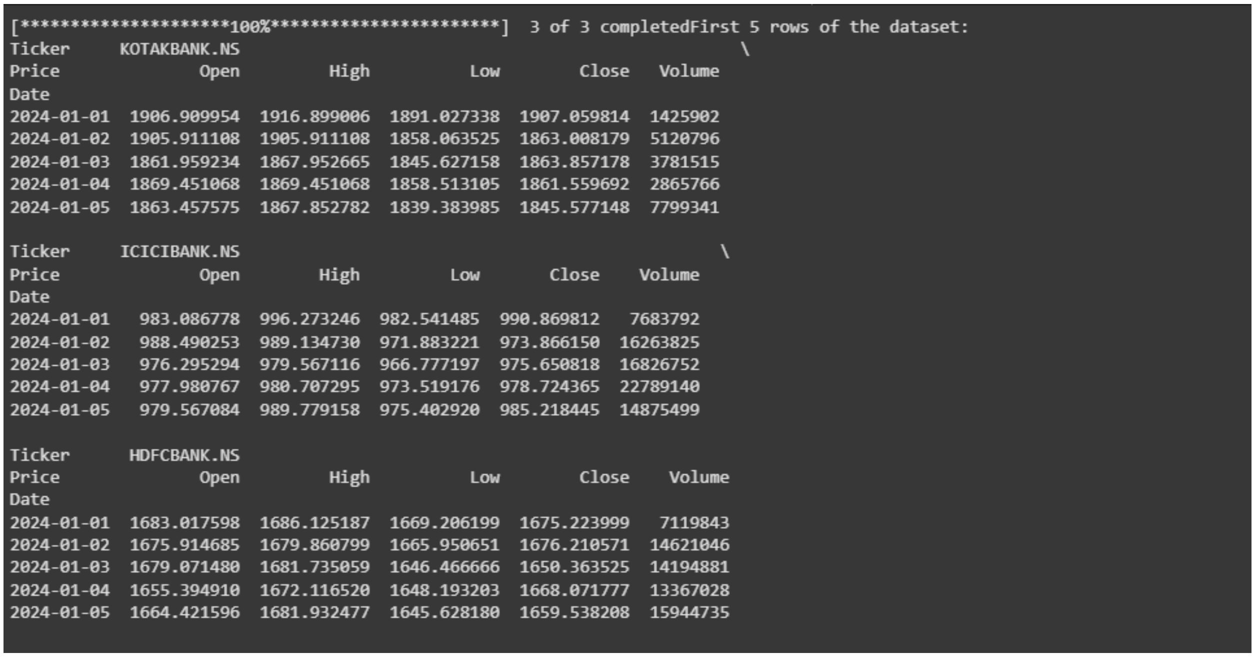
#Method 4

# STOCK MARKET DATA ANALYSIS

# Step 1: Import required libraries
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

# Step 2: Downloading Stock Market Data
# Define the ticker symbols for Indian companies

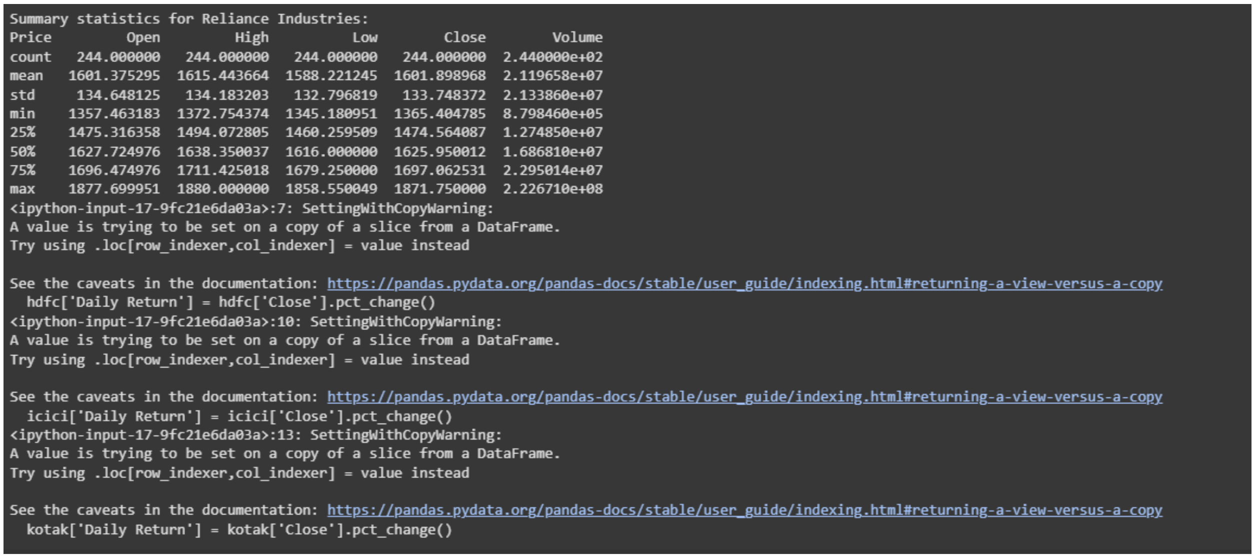
```
tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]

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

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



```
[*********************100%***********************]  3 of 3 completedFirst 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
```

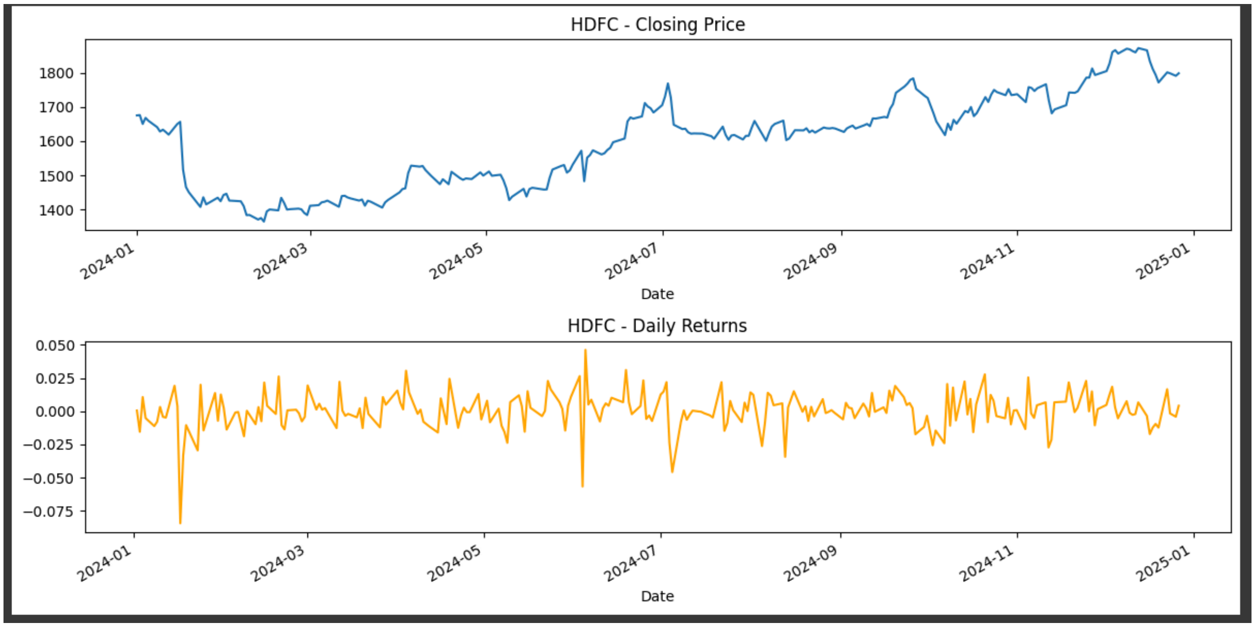
SUMMARY STATISTICS

```
# Summary statistics for a specific stock (e.g., Reliance)
hdfc = data['HDFCBANK.NS']
print("\nSummary statistics for Reliance Industries:")
print(hdfc.describe())

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

icici = data['ICICIBANK.NS']
icici['Daily Return'] = icici['Close'].pct_change()

kotak = data['KOTAKBANK.NS']
kotak['Daily Return'] = kotak['Close'].pct_change()
```

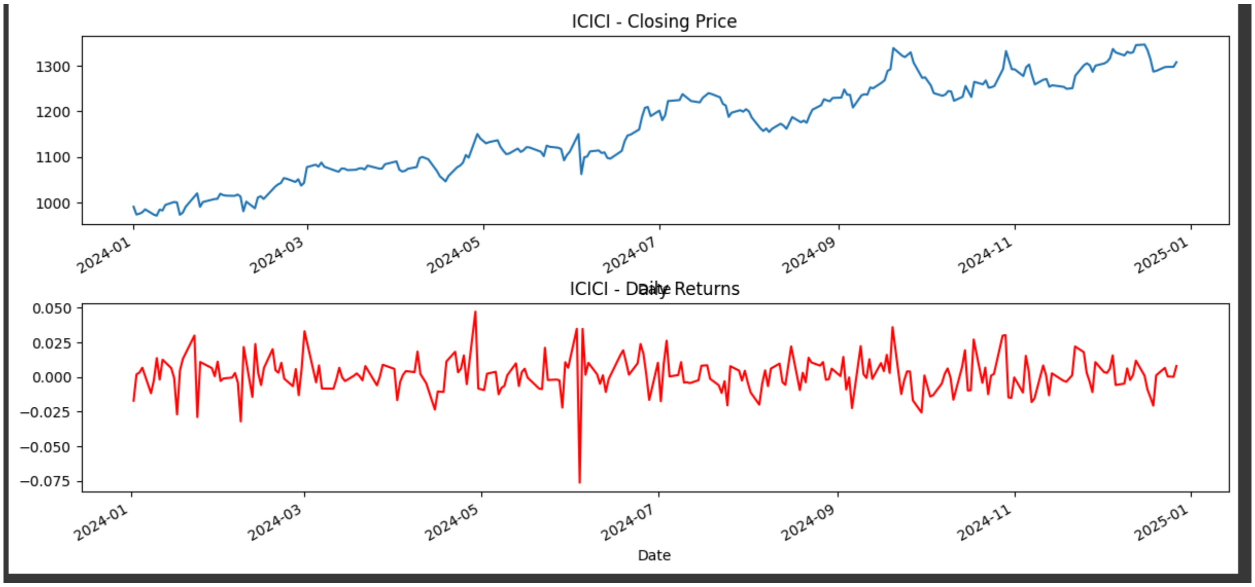


```
Summary statistics for Reliance Industries:
Price         Open         High          Low        Close        Volume
count   244.000000   244.000000   244.000000   244.000000  2.440000e+02
mean   1601.375295  1615.443664  1588.221245  1601.898968  2.119658e+07
std     134.648125   134.183203   132.796819   133.748372  2.133860e+07
min    1357.463183  1372.754374  1345.180951  1365.404785  8.798460e+05
25%    1475.316358  1494.072805  1460.259509  1474.564087  1.274850e+07
50%    1627.724976  1638.350037  1616.000000  1625.950012  1.686810e+07
75%    1696.474976  1711.425018  1679.250000  1697.062531  2.295014e+07
max    1877.699951  1880.000000  1858.550049  1871.750000  2.226710e+08
<ipython-input-17-9fc21e6da03a>:7: 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['Daily Return'] = hdfc['Close'].pct_change()
<ipython-input-17-9fc21e6da03a>:10: 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
  icici['Daily Return'] = icici['Close'].pct_change()
<ipython-input-17-9fc21e6da03a>:13: 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
  kotak['Daily Return'] = kotak['Close'].pct_change()
```

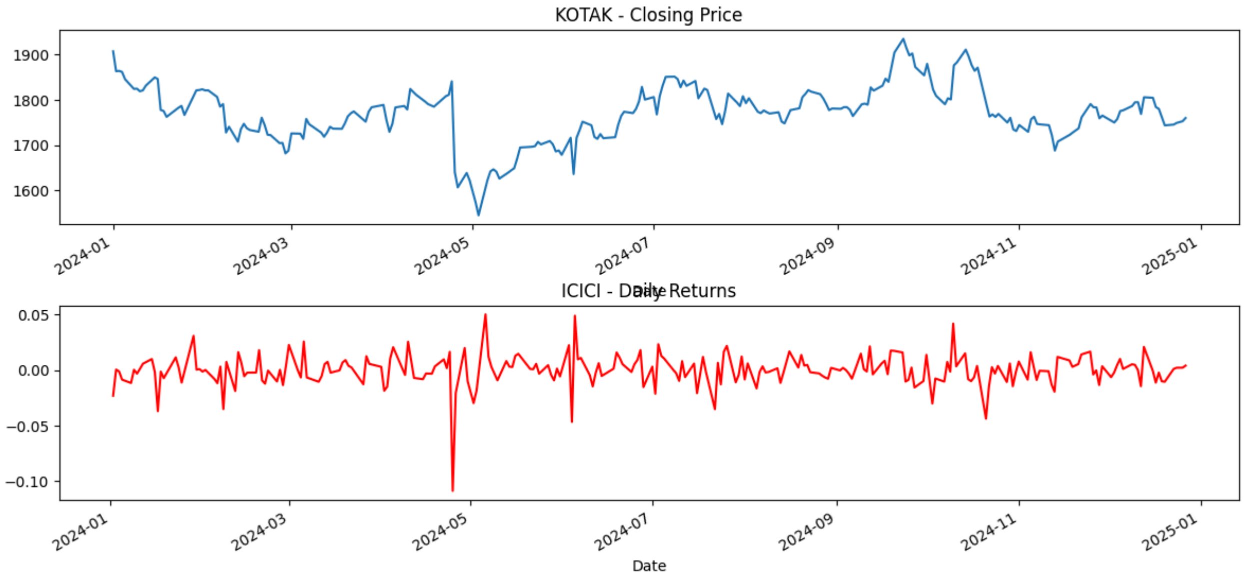
```
# plotting closing price and daily returns - HDFC

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
hdfc['Close'].plot(title="HDFC - Closing Price")
plt.subplot(2, 1, 2)
hdfc['Daily Return'].plot(title="HDFC - Daily Returns", color='orange')
plt.tight_layout()
plt.show()
```



```
#ICICI

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
icici['Close'].plot(title="ICICI - Closing Price")
plt.subplot(2, 1, 2)
plt.tight_layout()
icici['Daily Return'].plot(title="ICICI - Daily Returns", color='red')
plt.show()
```



```
#KOTAK

plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
kotak['Close'].plot(title="KOTAK - Closing Price")
plt.subplot(2, 1, 2)
plt.tight_layout()
kotak['Daily Return'].plot(title="ICICI - Daily Returns", color='red')
plt.show()
```


KOTAK - Closing Price
1900
1800
1700
1600
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
Date
ICICI - Daily Returns
0.05
0.00
−0.05
−0.10
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
Date