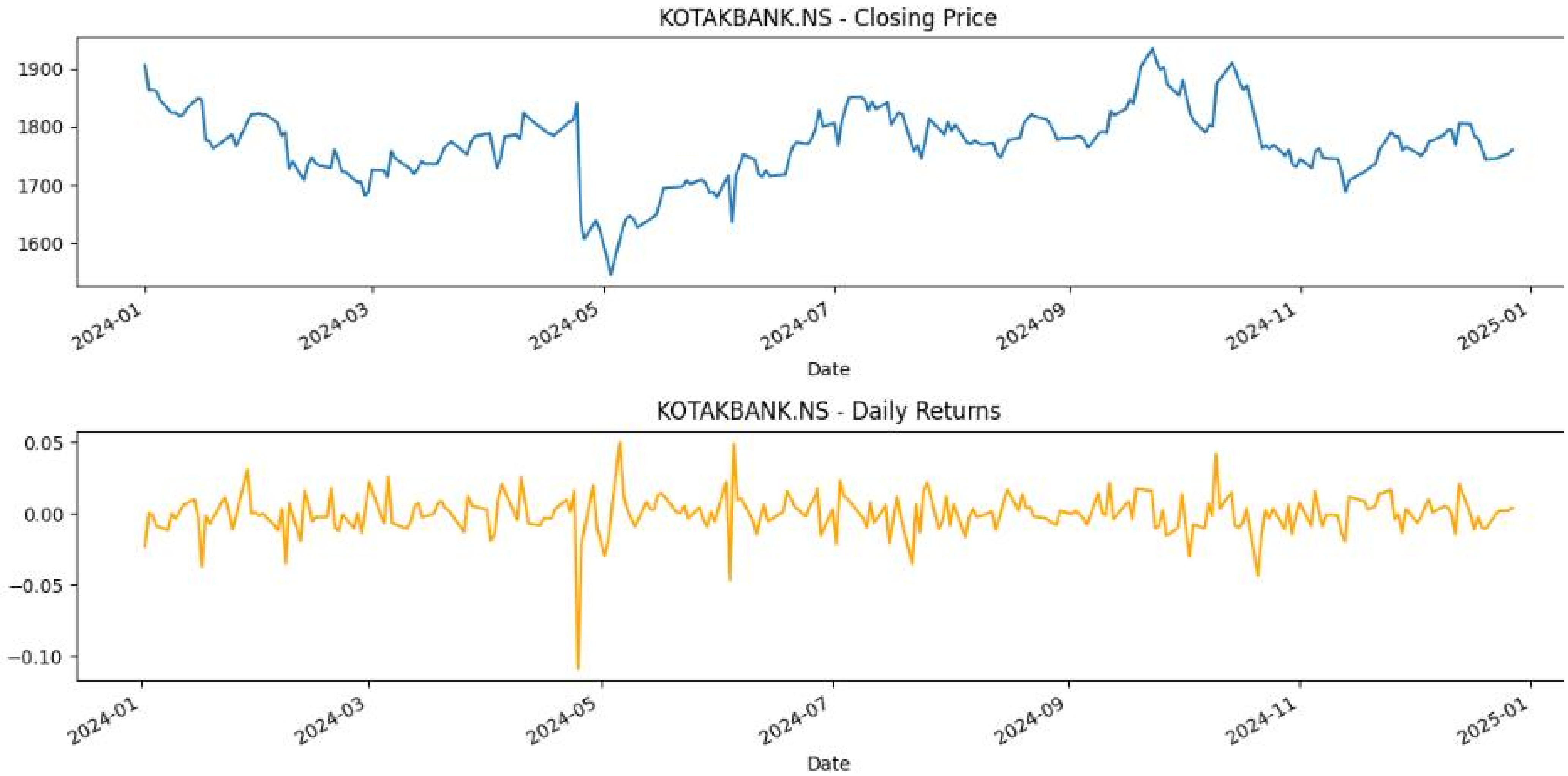
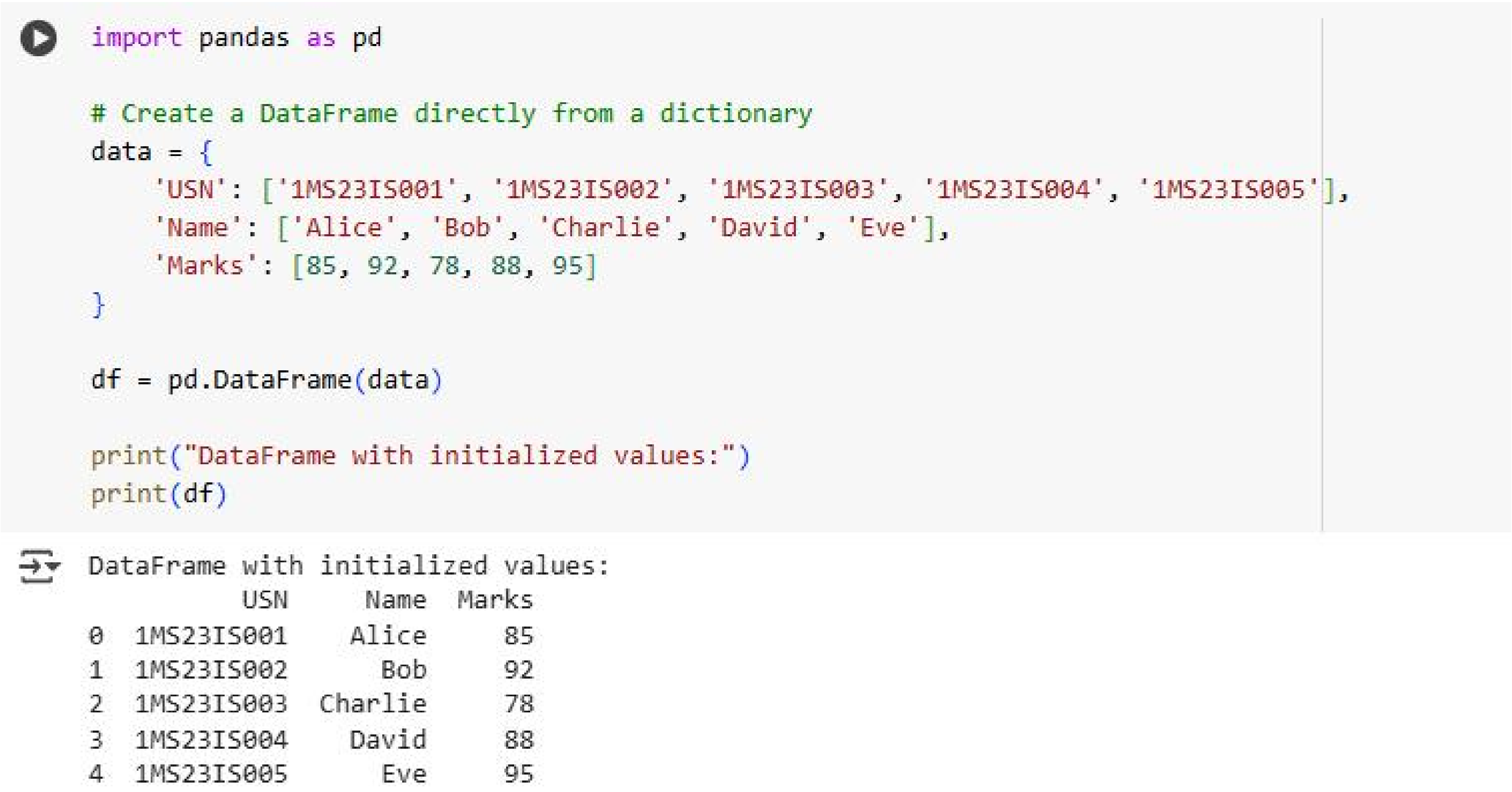


KOTAKBANK.NS - Closing Price

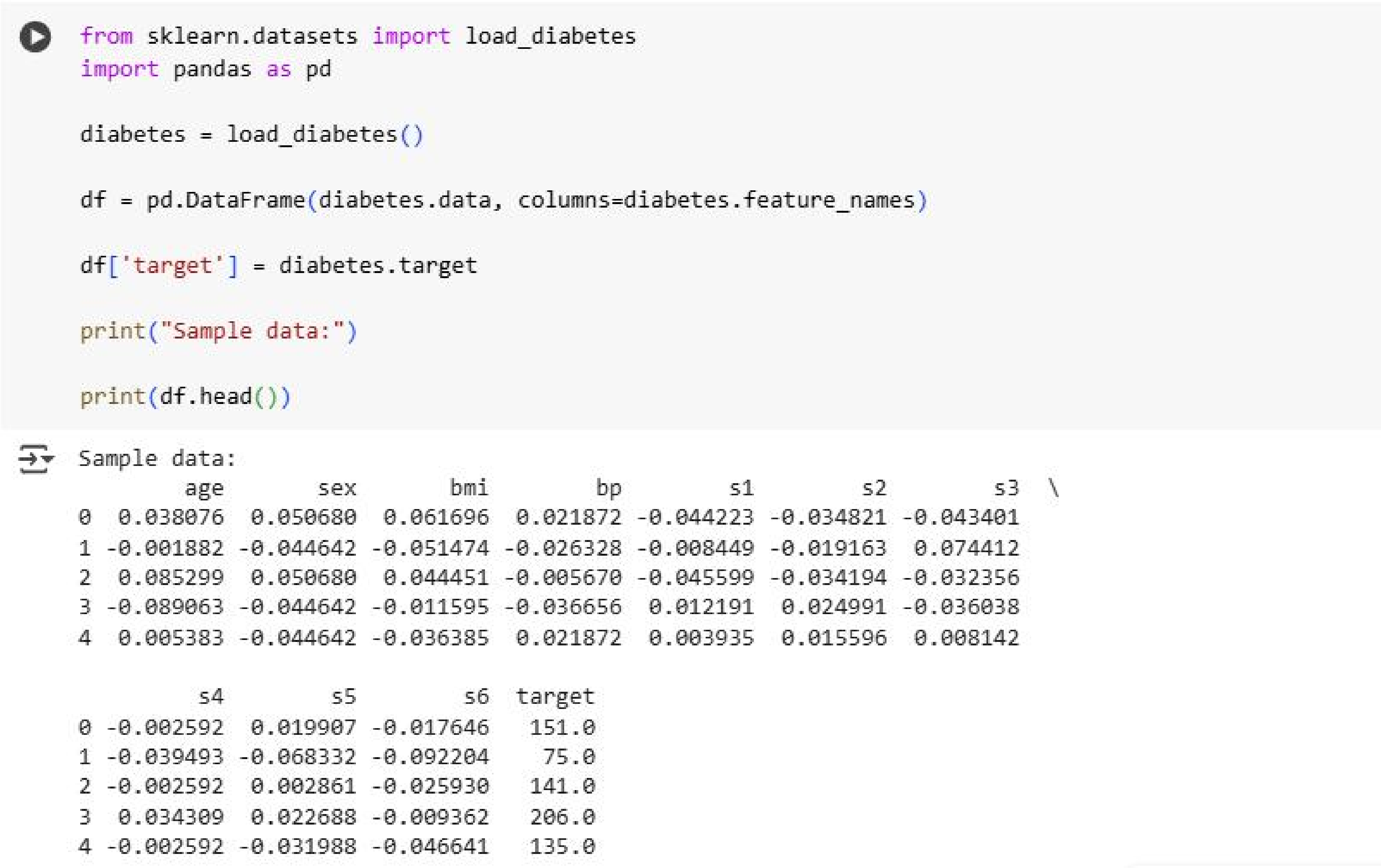
KOTAKBANK.NS - Daily Returns

```
KOTAKBANK.NS stock data saved to 'KOTAKBANK.NS_stock_data.csv'.
```



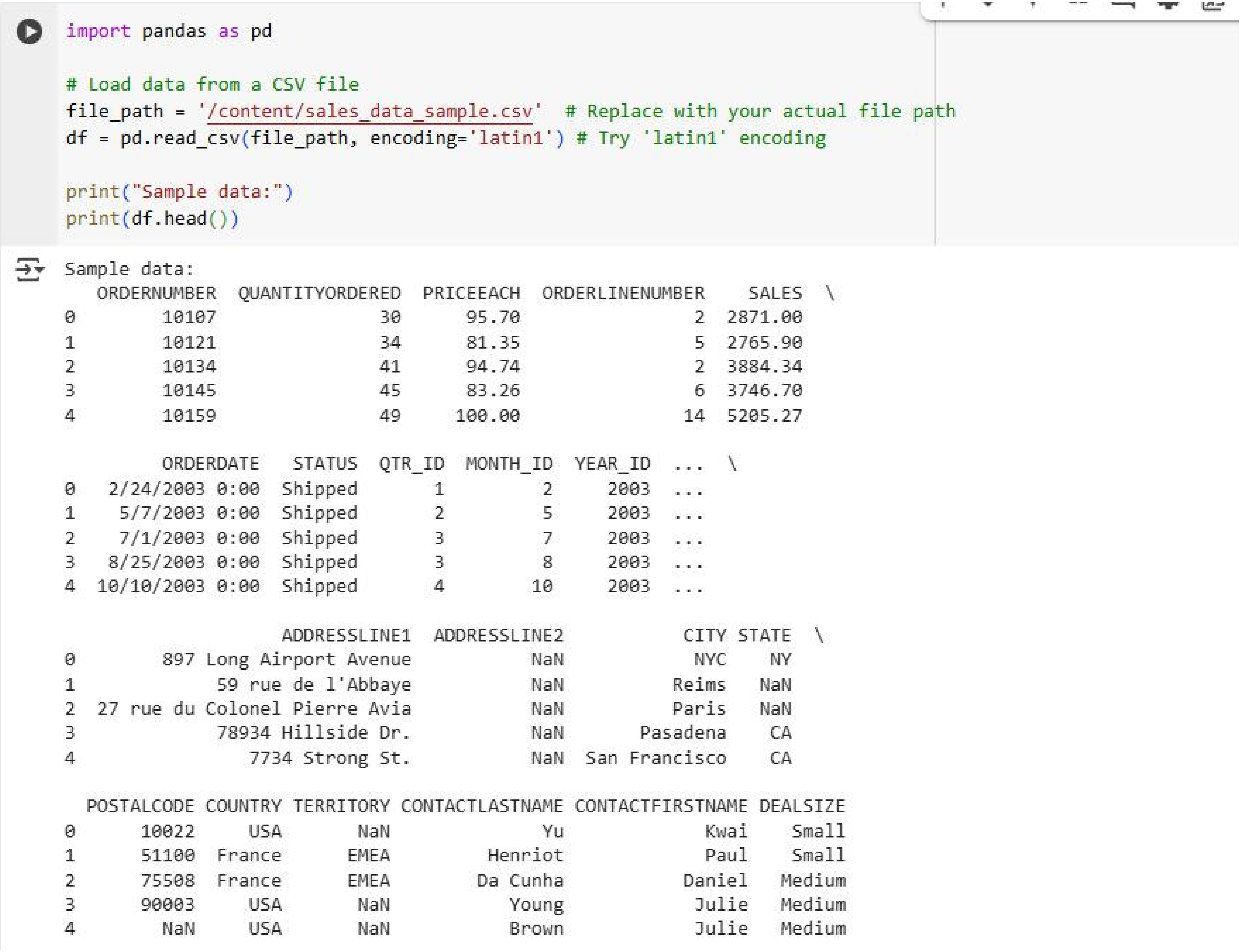
```python
import pandas as pd

# Create a DataFrame directly from a dictionary
data = {
    'USN': ['1MS23IS001', '1MS23IS002', '1MS23IS003', '1MS23IS004', '1MS23IS005'],
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Marks': [85, 92, 78, 88, 95]
}

df = pd.DataFrame(data)

print("DataFrame with initialized values:")
print(df)
```

```
DataFrame with initialized values:
          USN     Name  Marks
0  1MS23IS001    Alice     85
1  1MS23IS002      Bob     92
2  1MS23IS003  Charlie     78
3  1MS23IS004    David     88
4  1MS23IS005      Eve     95
```



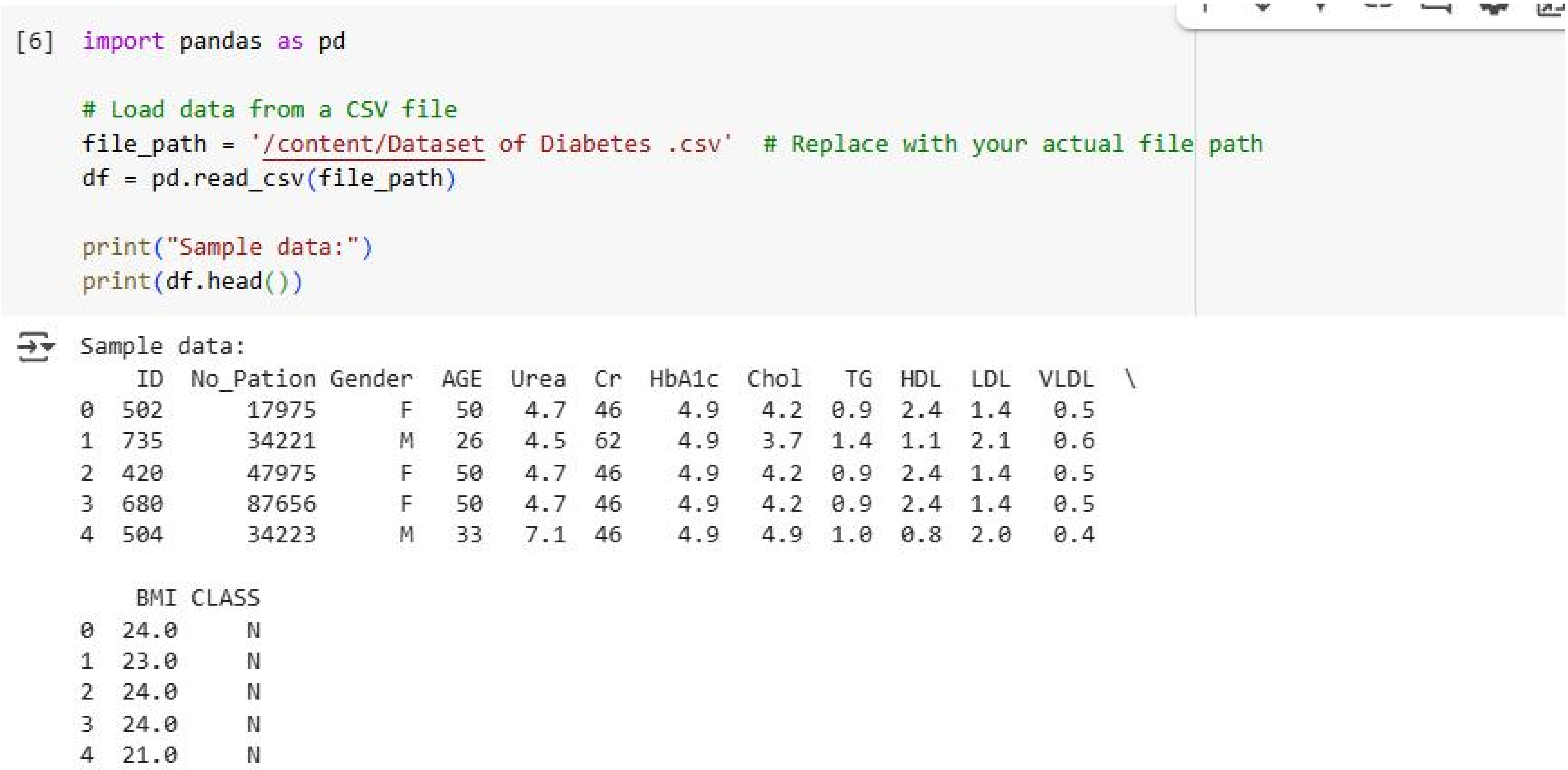
```python
from sklearn.datasets import load_diabetes
import pandas as pd

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 data from a CSV file
file_path = '/content/sales_data_sample.csv'  # Replace with your actual file path
df = pd.read_csv(file_path, encoding='latin1') # Try 'latin1' encoding

print("Sample data:")
print(df.head())
```

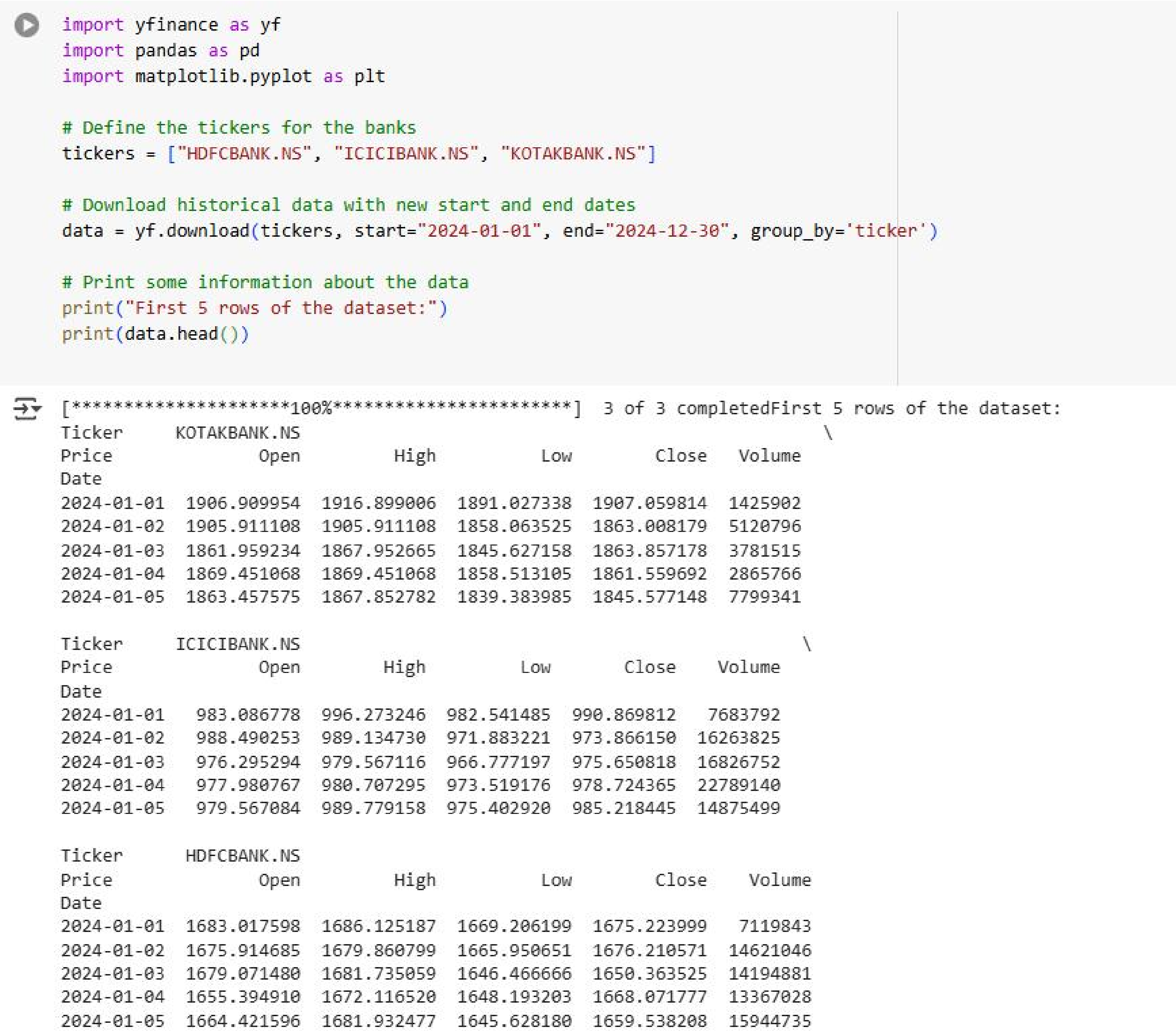
```
Sample data:
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  ...  \
0   2/24/2003 0:00  Shipped       1         2     2003  ...
1    5/7/2003 0:00  Shipped       2         5     2003  ...
2    7/1/2003 0:00  Shipped       3         7     2003  ...
3   8/25/2003 0:00  Shipped       3         8     2003  ...
4  10/10/2003 0:00  Shipped       4        10     2003  ...

                    ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN              Yu             Kwai    Small
1      51100  France      EMEA         Henriot             Paul    Small
2      75508  France      EMEA        Da Cunha           Daniel   Medium
3      90003     USA       NaN           Young            Julie   Medium
4        NaN     USA       NaN           Brown            Julie   Medium
```



[6]

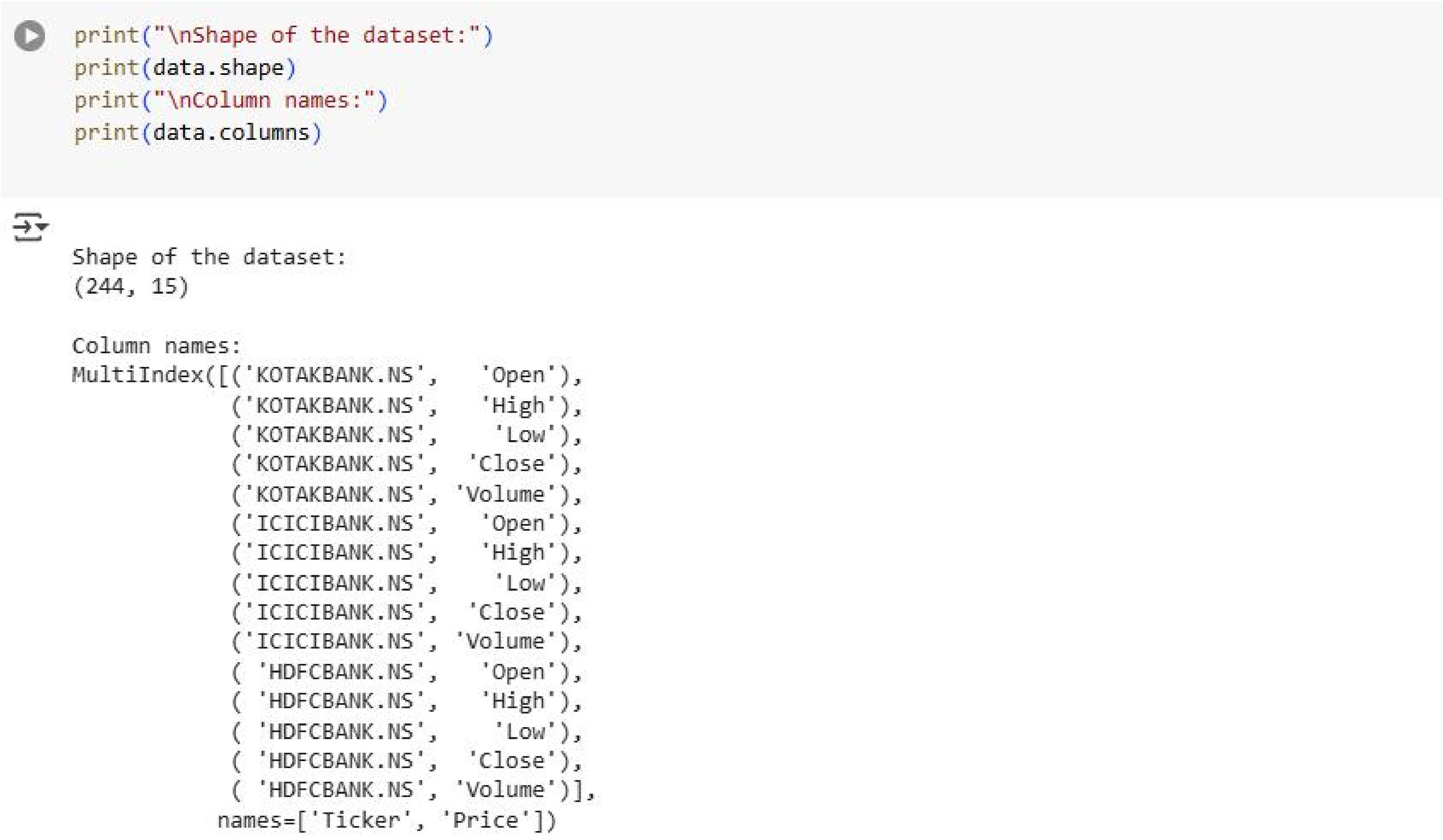
```
import pandas as pd

# Load data from a CSV file
file_path = '/content/Dataset of Diabetes .csv'  # Replace with your actual file path
df = pd.read_csv(file_path)

print("Sample data:")
print(df.head())
```

```
Sample data:
    ID  No_Pation 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
```



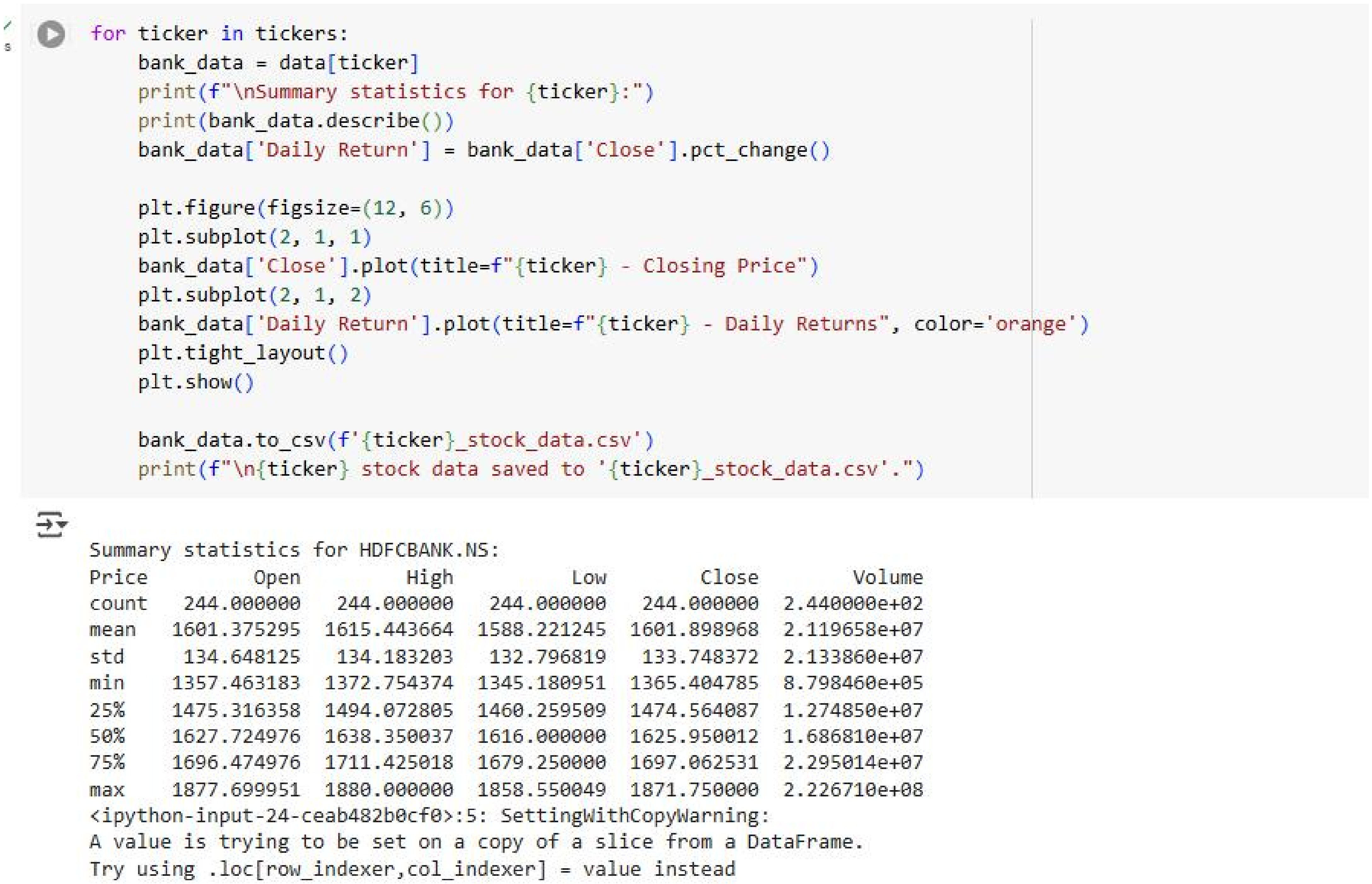
```python
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

# Define the tickers for the banks
tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]

# Download historical data with new start and end dates
data = yf.download(tickers, start="2024-01-01", end="2024-12-30", group_by='ticker')

# Print some information about the data
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  
```



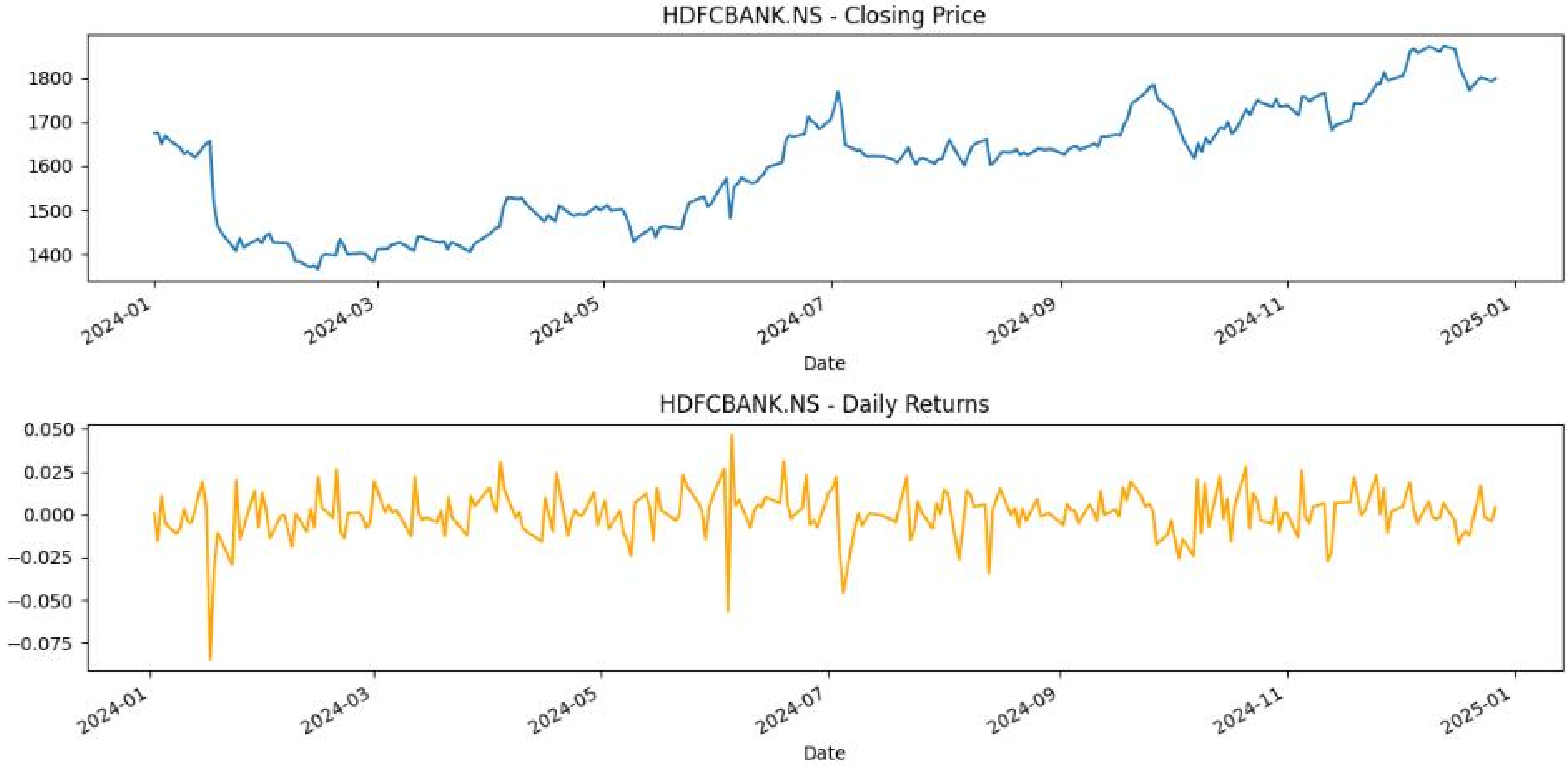
```python
print("\nShape of the dataset:")
print(data.shape)
print("\nColumn names:")
print(data.columns)
```

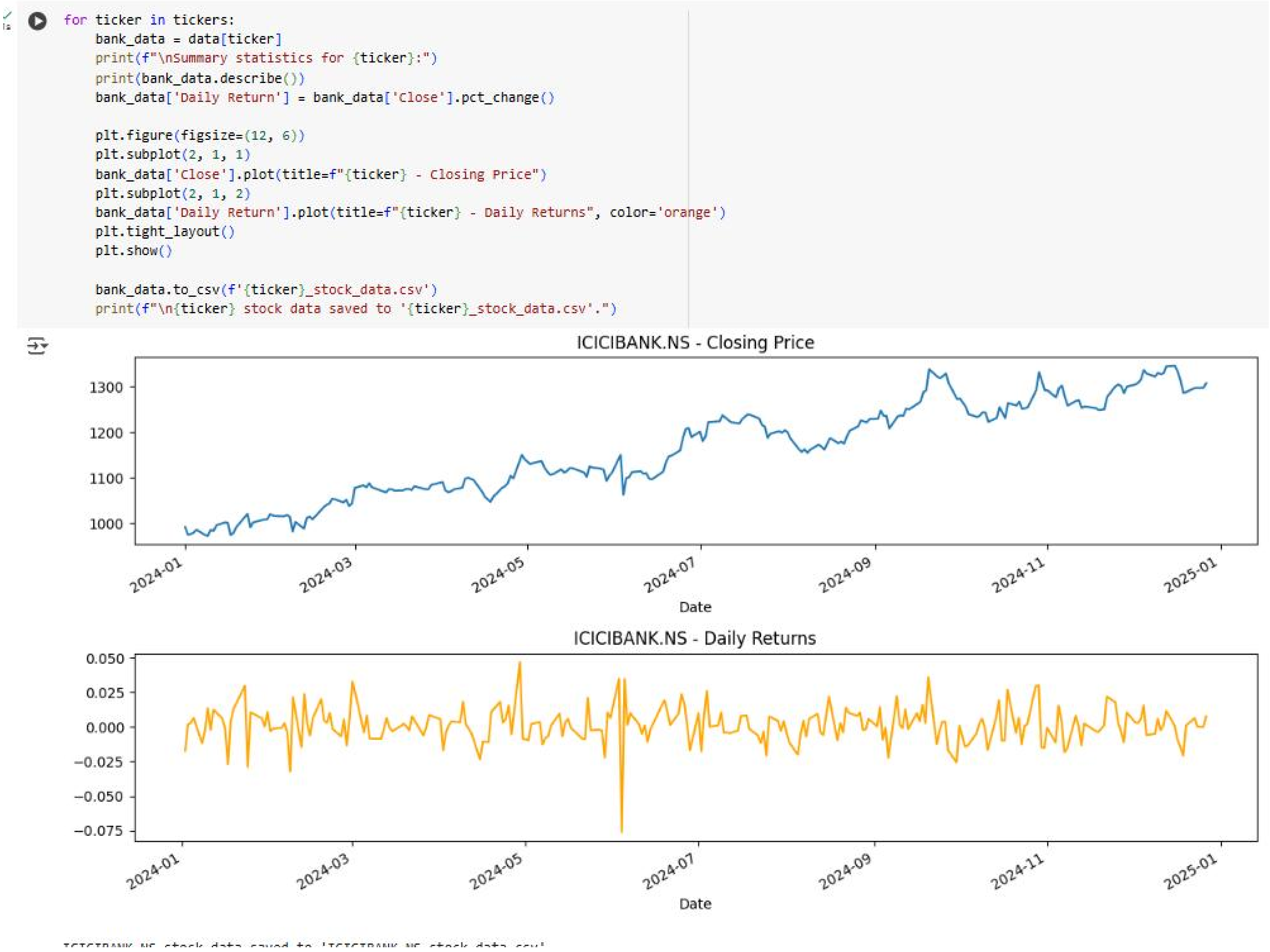
```
Shape of the dataset:
(244, 15)

Column names:
MultiIndex([('KOTAKBANK.NS',   'Open'),
            ('KOTAKBANK.NS',   'High'),
            ('KOTAKBANK.NS',    'Low'),
            ('KOTAKBANK.NS',  'Close'),
            ('KOTAKBANK.NS', 'Volume'),
            ('ICICIBANK.NS',   'Open'),
            ('ICICIBANK.NS',   'High'),
            ('ICICIBANK.NS',    'Low'),
            ('ICICIBANK.NS',  'Close'),
            ('ICICIBANK.NS', 'Volume'),
            ( 'HDFCBANK.NS',   'Open'),
            ( 'HDFCBANK.NS',   'High'),
            ( 'HDFCBANK.NS',    'Low'),
            ( 'HDFCBANK.NS',  'Close'),
            ( 'HDFCBANK.NS', 'Volume')],
           names=['Ticker', 'Price'])
```



```python
for ticker in tickers:
    bank_data = data[ticker]
    print(f"\nSummary statistics for {ticker}:")
    print(bank_data.describe())
    bank_data['Daily Return'] = bank_data['Close'].pct_change()

    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1)
    bank_data['Close'].plot(title=f"{ticker} - Closing Price")
    plt.subplot(2, 1, 2)
    bank_data['Daily Return'].plot(title=f"{ticker} - Daily Returns", color='orange')
    plt.tight_layout()
    plt.show()

    bank_data.to_csv(f'{ticker}_stock_data.csv')
    print(f"\n{ticker} stock data saved to '{ticker}_stock_data.csv'.")
```

```
Summary statistics for HDFCBANK.NS:
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-24-ceab482b0cf0>:5: 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
```


HDFCBANK.NS - Closing Price
1800
1700
1600
1500
1400
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
Date
HDFCBANK.NS - Daily Returns
0.050
0.025
0.000
−0.025
−0.050
−0.075
2024-01
2024-03
2024-05
2024-07
2024-09
2024-11
2025-01
Date



```
for ticker in tickers:
    bank_data = data[ticker]
    print(f"\nSummary statistics for {ticker}:")
    print(bank_data.describe())
    bank_data['Daily Return'] = bank_data['Close'].pct_change()

    plt.figure(figsize=(12, 6))
    plt.subplot(2, 1, 1)
    bank_data['Close'].plot(title=f"{ticker} - Closing Price")
    plt.subplot(2, 1, 2)
    bank_data['Daily Return'].plot(title=f"{ticker} - Daily Returns", color='orange')
    plt.tight_layout()
    plt.show()

    bank_data.to_csv(f'{ticker}_stock_data.csv')
    print(f"\n{ticker} stock data saved to '{ticker}_stock_data.csv'.")
```

```
ICICIBANK.NS stock data saved to 'ICICIBANK.NS_stock_data.csv'
```