

```
from sklearn.datasets import load_iris
import pandas as pd

iris = load_iris()
df = pd.DataFrame(iris.data, columns=iris.feature_names)
df['target'] = iris.target
print("sample data: ")
df.head()
```

↗ sample data:

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

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
# method 1

data = {
    'USN': ['A001', 'A002', 'A003', 'A004'],
    'Name': ['Amar', 'Akbar', 'Anthony', 'Bob'],
    'MarksInMaths': [34, 30, 31, 32]
}

df2 = pd.DataFrame(data)
df2
```

↗

	USN	Name	MarksInMaths
0	A001	Amar	34
1	A002	Akbar	30
2	A003	Anthony	31
3	A004	Bob	32

Next steps:

[Generate code with df2](#)

[View recommended plots](#)

[New interactive sheet](#)

```
# method 2

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: ")
df.head()
```

↗ sample data:

	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

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
# method 3
```

```
# Load data from a CSV file (replace 'data.csv' with your file path)
file_path = '/content/industry.csv' # Ensure the file exists in the same directory
df2 = pd.read_csv(file_path)
print("Sample data:")
df2.head()
```

Sample data:

	Industry
0	Accounting/Finance
1	Advertising/Public Relations
2	Aerospace/Aviation
3	Arts/Entertainment/Publishing
4	Automotive

Next steps:

Generate code with df2

View recommended plots

New interactive sheet

```
# method 4

file_path = '/content/Dataset of Diabetes .csv'
data3 = pd.read_csv(file_path)
df3 = pd.DataFrame(data3)
```

df3

	ID	No_Pation	Gender	AGE	Urea	Cr	HbA1c	Cho1	TG	HDL	LDL	VLDL	BMI	CLASS
0	502	17975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
1	735	34221	M	26	4.5	62	4.9	3.7	1.4	1.1	2.1	0.6	23.0	N
2	420	47975	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
3	680	87656	F	50	4.7	46	4.9	4.2	0.9	2.4	1.4	0.5	24.0	N
4	504	34223	M	33	7.1	46	4.9	4.9	1.0	0.8	2.0	0.4	21.0	N
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
995	200	454317	M	71	11.0	97	7.0	7.5	1.7	1.2	1.8	0.6	30.0	Y
996	671	876534	M	31	3.0	60	12.3	4.1	2.2	0.7	2.4	15.4	37.2	Y
997	669	87654	M	30	7.1	81	6.7	4.1	1.1	1.2	2.4	8.1	27.4	Y
998	99	24004	M	38	5.8	59	6.7	5.3	2.0	1.6	2.9	14.0	40.5	Y
999	248	24054	M	54	5.0	67	6.9	3.8	1.7	1.1	3.0	0.7	33.0	Y

1000 rows × 14 columns

Next steps:

Generate code with df3

View recommended plots

New interactive sheet

```
#Using the code given in the above slides, do the exercise of the “Stock Market Data Analysis”, considering the follwoing
# 1. HDFC Bank Ltd. , ICICI Bank Ltd , Kotak Mahindra Bank Ltd.
# tickers = ["HDFCBANK.NS", "ICICIBANK.NS", "KOTAKBANK.NS"]
# 2. Start date: 2024-01-01, End date: 2024-12-30

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:")
data.head()
```

YF.download() has changed argument auto\_adjust default to True  
[\*\*\*\*\*100%\*\*\*\*\*] 3 of 3 completed  
First 5 rows of the dataset:

ICICIBANK.NS						HDFCBANK.NS					KOTAKBANK.NS	
Price	Open	High	Low	Close	Volume	Open	High	Low	Close	Volume	Open	
Date												
2024-01-01	983.086778	996.273246	982.541485	990.869812	7683792	1683.017598	1686.125187	1669.206199	1675.223999	7119843	1906.909954	
2024-01-02	988.490253	989.134730	971.883221	973.866150	16263825	1675.914685	1679.860799	1665.950651	1676.210571	14621046	1905.911108	
2024-01-03	976.295294	979.567116	966.777197	975.650818	16826752	1679.071480	1681.735059	1646.466666	1650.363525	14194881	1861.959234	

Next steps: [Generate code with data](#) [View recommended plots](#) [New interactive sheet](#)

```
print("\nShape of the dataset:")
print(data.shape)
print("\nColumn names:")
print(data.columns)
hdfc_data = data['HDFCBANK.NS']
print("\nSummary statistics for Reliance Industries:")
print(hdfc_data.describe())
hdfc_data['Daily Return'] = hdfc_data['Close'].pct_change()

# icici bank
icici_data = data['ICICIBANK.NS']
print(hdfc_data.describe())
icici_data['Daily Return'] = icici_data['Close'].pct_change()

# Kotak bank
kotak_data = data['KOTAKBANK.NS']
print(hdfc_data.describe())
kotak_data['Daily Return'] = kotak_data['Close'].pct_change()
```

Shape of the dataset:  
(244, 15)

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

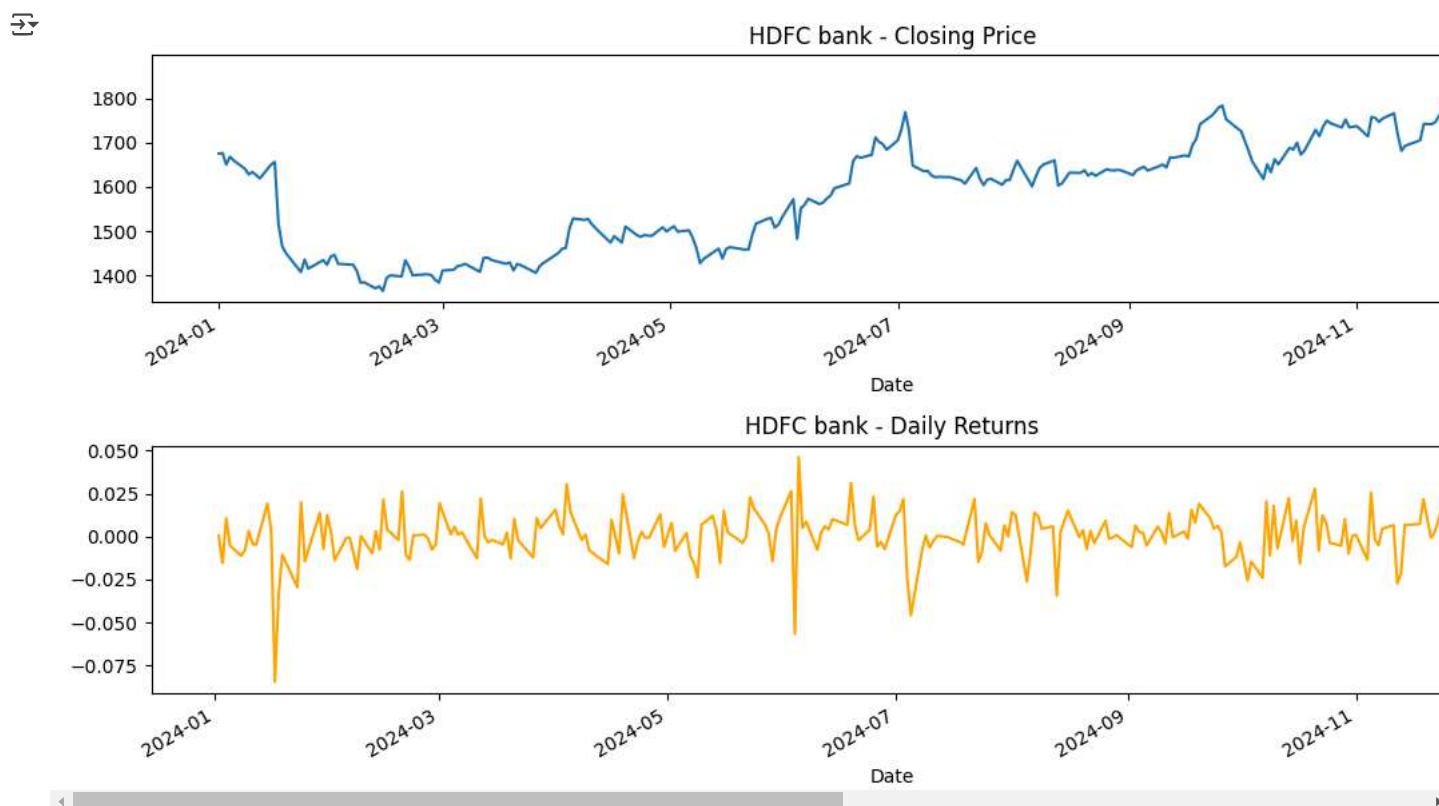
Summary statistics for Reliance Industries:					
Price	Open	High	Low	Close	Volume
count	244.000000	244.000000	244.000000	244.000000	2.440000e+02
mean	1601.375295	1615.443664	1588.221245	1601.898968	2.119658e+07
std	134.648125	134.183203	132.796819	133.748372	2.133860e+07
min	1357.463183	1372.754374	1345.180951	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
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

Price	Daily Return
count	243.000000
mean	0.000392
std	0.014151
min	-0.084358
25%	-0.006986
50%	0.000703
75%	0.007460
max	0.046287

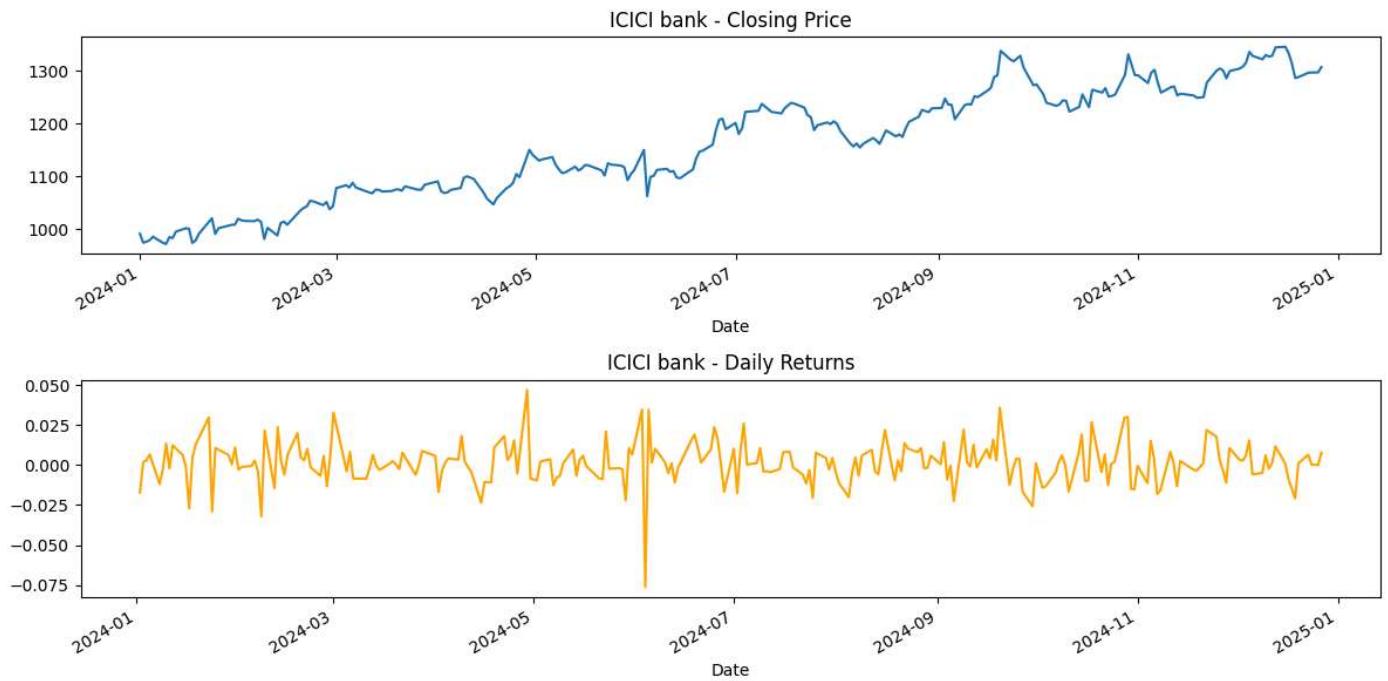
  

Price	Open	High	Low	Close	Volume \
count	244.000000	244.000000	244.000000	244.000000	2.440000e+02
mean	1601.375295	1615.443664	1588.221245	1601.898968	2.119658e+07
std	134.648125	134.183203	132.796819	133.748372	2.133860e+07
min	1357.463183	1372.754374	1345.180951	1365.404785	8.798460e+05
25%	1475.316358	1494.072805	1460.259509	1474.564087	1.274850e+07

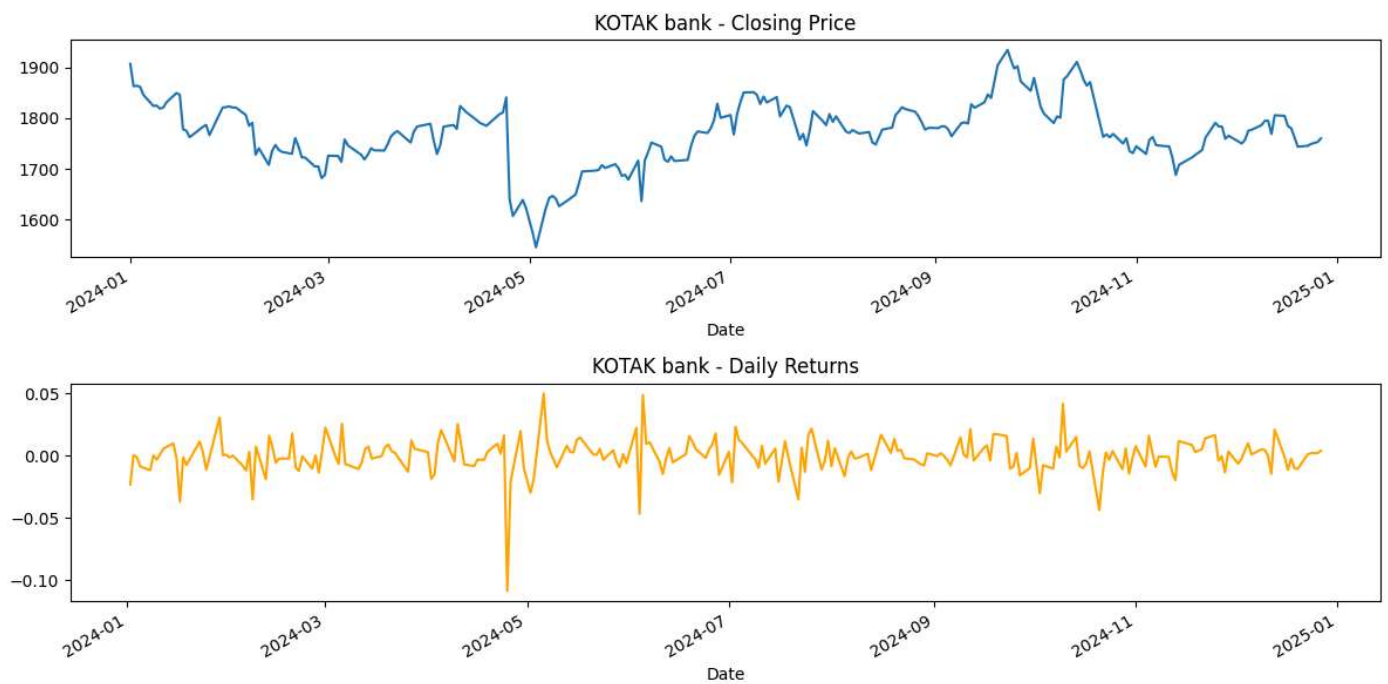
```
plt.figure(figsize=(12, 6))
plt.subplot(2, 1, 1)
hdfc_data['Close'].plot(title="HDFC bank - Closing Price")
plt.subplot(2, 1, 2)
hdfc_data['Daily Return'].plot(title="HDFC bank - Daily Returns", color='orange')
plt.tight_layout()
plt.show()
```



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



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



```
# Save the Reliance data to a CSV file

hdfc_data.to_csv('hdfc_stock_data.csv')
```

```
icici_data.to_csv('icici_stock_data.csv')  
kotak_data.to_csv('kotak_stock_data.csv')
```

Start coding or [generate](#) with AI.