


```
# prompt: name usn marks dataframe with 5 rows

import pandas as pd

# Sample data (replace with your actual data)
data = {
    'USN': [101, 102, 103, 104, 105],
    'Name': ['Alice', 'Bob', 'Charlie', 'David', 'Eve'],
    'Marks': [85, 92, 78, 95, 88]
}

# Create a DataFrame
df = pd.DataFrame(data)

# Print the DataFrame
df
```



	USN	Name	Marks
0	101	Alice	85
1	102	Bob	92
2	103	Charlie	78
3	104	David	95
4	105	Eve	88

Next steps:


[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
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
df
```



	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
...
437	0.041708	0.050680	0.019662	0.059744	-0.005697	-0.002566	-0.028674	-0.002592	0.031193	0.007207	178.0
438	-0.005515	0.050680	-0.015906	-0.067642	0.049341	0.079165	-0.028674	0.034309	-0.018114	0.044485	104.0
439	0.041708	0.050680	-0.015906	0.017293	-0.037344	-0.013840	-0.024993	-0.011080	-0.046883	0.015491	132.0
440	-0.045472	-0.044642	0.039062	0.001215	0.016318	0.015283	-0.028674	0.026560	0.044529	-0.025930	220.0
441	-0.045472	-0.044642	-0.073030	-0.081413	0.083740	0.027809	0.173816	-0.039493	-0.004222	0.003064	57.0

442 rows × 11 columns

Next steps:

[Generate code with df](#)

[View recommended plots](#)

[New interactive sheet](#)

```
file_path= '/content/Dataset of Diabetes .csv'
df2 = pd.read_csv(file_path)
df2
```

	ID	No_Patient	Gender	AGE	Urea	Cr	HbA1c	Chol	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 df2](#) [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 following
#1. HDFC Bank Ltd. , ICICI Bank Ltd , Kotak Mahindra Bank Ltd.
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())

[*****100%*****] 3 of 3 completedFirst 5 rows of the dataset:
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

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

print("\nShape of the dataset:")

print(data.shape)
print("\nColumn names:")

print(data.columns)

HDFC_BANK = data['HDFCBANK.NS']
ICICIBANK = data['ICICIBANK.NS']
KOTAKBANK = data['KOTAKBANK.NS']

print("\nSummary statistics for BANK:")
```

```
print(HDFC_BANK.describe())
```



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 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	1601.898968	2.119658e+07
std	134.648125	134.183203	132.796819	133.748372	133.748372	2.133860e+07
min	1357.463183	1372.754374	1345.180951	1365.404785	1365.404785	8.798460e+05
25%	1475.316358	1494.072805	1460.259509	1474.564087	1474.564087	1.274850e+07
50%	1627.724976	1638.350037	1616.000000	1625.950012	1625.950012	1.686810e+07
75%	1696.474976	1711.425018	1679.250000	1697.062531	1697.062531	2.295014e+07
max	1877.699951	1880.000000	1858.550049	1871.750000	1871.750000	2.226710e+08

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

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

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

```
print("\nSummary statistics for BANK:")
```

```
print(HDFC_BANK.describe())
```

```
print(ICICIBANK.describe())
```

```
print(KOTAKBANK.describe())
```



Summary statistics for 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	1601.898968	2.119658e+07	
std	134.648125	134.183203	132.796819	133.748372	133.748372	2.133860e+07	
min	1357.463183	1372.754374	1345.180951	1365.404785	1365.404785	8.798460e+05	
25%	1475.316358	1494.072805	1460.259509	1474.564087	1474.564087	1.274850e+07	
50%	1627.724976	1638.350037	1616.000000	1625.950012	1625.950012	1.686810e+07	
75%	1696.474976	1711.425018	1679.250000	1697.062531	1697.062531	2.295014e+07	
max	1877.699951	1880.000000	1858.550049	1871.750000	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	244.000000	2.440000e+02	
mean	1161.723560	1173.687900	1151.318979	1162.751791	1162.751791	1.539172e+07	
std	104.905646	105.668229	105.083015	105.520481	105.520481	9.503609e+06	
min	965.637027	979.567116	961.869473	971.387512	971.387512	1.007022e+06	
25%	1073.818215	1085.368782	1067.386038	1075.107086	1075.107086	1.014533e+07	
50%	1169.443635	1178.450012	1157.361521	1165.470703	1165.470703	1.291768e+07	
75%	1248.512512	1261.399994	1236.649963	1250.812531	1250.812531	1.755770e+07	
max	1344.900024	1362.349976	1340.050049	1346.099976	1346.099976	7.325777e+07	

Price Daily Return

count	243.000000
mean	0.001228
std	0.013124
min	-0.076336
25%	-0.004943
50%	0.001207

	75%	0.007810				
max	0.047161					
Price	Open	High	Low	Close	Volume	\
count	244.000000	244.000000	244.000000	244.000000	2.440000e+02	
mean	1771.245907	1787.548029	1754.395105	1770.792347	5.736598e+06	
std	62.189675	61.978802	62.765980	62.594747	5.388927e+06	
min	1581.266899	1586.161558	1542.159736	1545.006592	1.824890e+05	
25%	1733.974927	1754.131905	1719.028421	1736.297058	3.300380e+06	
50%	1769.500000	1789.450012	1758.099976	1773.681030	4.307680e+06	
75%	1809.925018	1826.998164	1789.912506	1808.155670	6.159475e+06	
max	1935.000000	1942.000000	1909.599976	1934.699951	6.617908e+07	

	Price	Daily Return
count	243.000000	
mean	-0.000221	
std	0.014692	
min	-0.108720	
25%	-0.007429	

```
plt.figure(figsize=(12, 6))

plt.subplot(2, 1, 1)

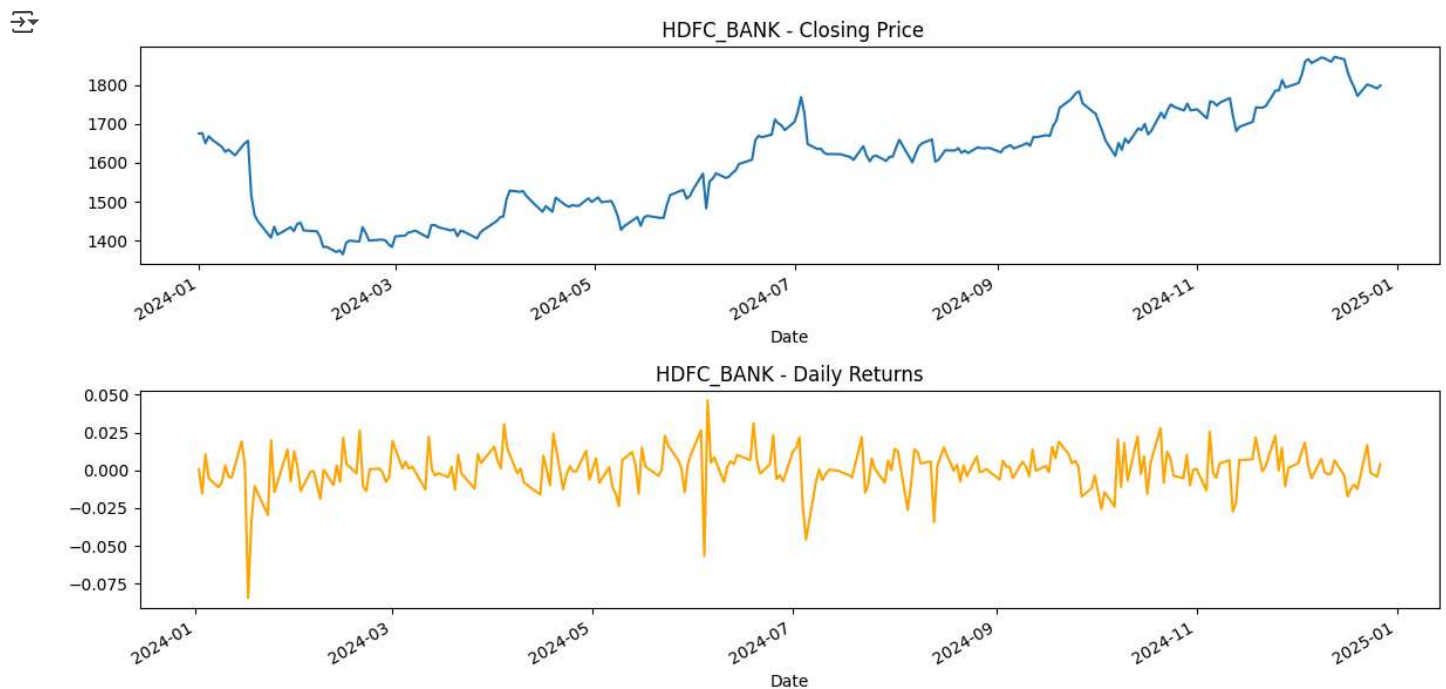
HDFC_BANK['Close'].plot(title=" HDFC_BANK - Closing Price")

plt.subplot(2, 1, 2)

HDFC_BANK['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)

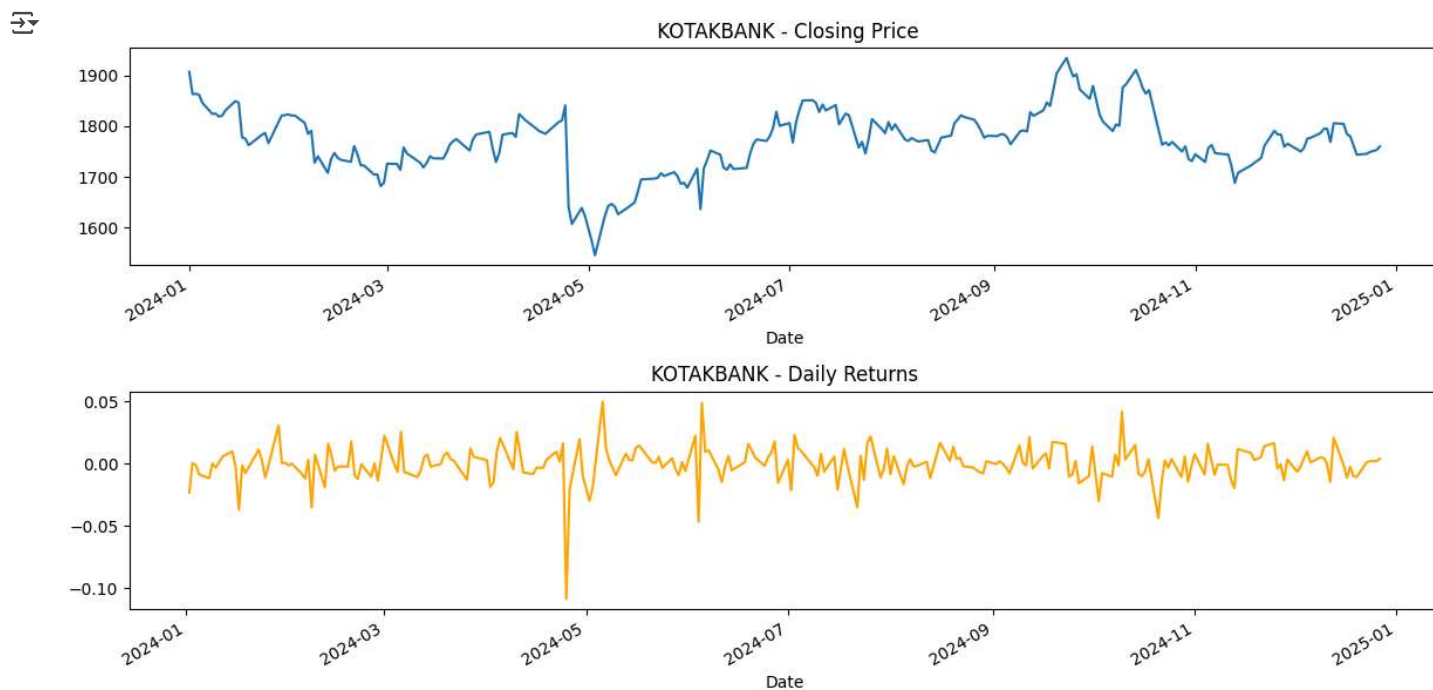
KOTAKBANK['Close'].plot(title=" KOTAKBANK - Closing Price")

plt.subplot(2, 1, 2)

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

plt.tight_layout()
```

```
plt.show()
```



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

```
ICICIBANK['Close'].plot(title=" ICICIBANK - Closing Price")
```

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

```
ICICIBANK['Daily Return'].plot(title="ICICIBANK - Daily Returns", color='orange')
```

```
plt.tight_layout()
```

```
plt.show()
```

