Code:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as py
```
Code:
```
df = pd.read_csv("/content/stocks.csv")
```
Code:
```
df.head()
```
Output:
```
Ticker Date Open High Low Close \
0 AAPL 2023-02-07 150.639999 155.229996 150.639999 154.649994
1 AAPL 2023-02-08 153.880005 154.580002 151.169998 151.919998
2 AAPL 2023-02-09 153.779999 154.330002 150.419998 150.869995
3 AAPL 2023-02-10 149.460007 151.339996 149.220001 151.009995
4 AAPL 2023-02-13 150.949997 154.259995 150.919998 153.850006
Adj Close Volume
0 154.414230 83322600
1 151.688400 64120100
2 150.639999 56007100
3 151.009995 57450700
4 153.850006 62199000
```
Code:
```
df.tail()
```
Output:
```
Ticker Date Open High Low Close \
243 GOOG 2023-05-01 107.720001 108.680000 107.500000 107.709999
244 GOOG 2023-05-02 107.660004 107.730003 104.500000 105.980003
245 GOOG 2023-05-03 106.220001 108.129997 105.620003 106.120003
246 GOOG 2023-05-04 106.160004 106.300003 104.699997 105.209999
247 GOOG 2023-05-05 105.320000 106.440002 104.738998 106.214996
Adj Close Volume
243 107.709999 20926300
244 105.980003 20343100
245 106.120003 17116300
246 105.209999 19780600
247 106.214996 20705300
```
Code:
```
# checking Null Values or Missing Values
df.isnull().sum()
```
Output:
```
Ticker 0
Date 0
Open 0
High 0
Low 0
Close 0
Adj Close 0
Volume 0
dtype: int64
```
Code:
```
df.duplicated()
```
Output:
```
0 False
1 False
2 False
3 False
4 False
```

```
...
243 False
244 False
245 False
246 False
247 False
Length: 248, dtype: bool
```
Code:
```
df.info()
```
Output:
```

RangeIndex: 248 entries, 0 to 247
Data columns (total 8 columns):
# Column Non-Null Count Dtype
--- ------ -------------- -----
0 Ticker 248 non-null object
1 Date 248 non-null object
2 Open 248 non-null float64
3 High 248 non-null float64
4 Low 248 non-null float64
5 Close 248 non-null float64
6 Adj Close 248 non-null float64
7 Volume 248 non-null int64
dtypes: float64(5), int64(1), object(2)
memory usage: 15.6+ KB
```

Code:
```
df.describe()
```
Output:
```
Open High Low Close Adj Close \
count 248.000000 248.000000 248.000000 248.000000 248.000000
mean 215.252093 217.919662 212.697452 215.381674 215.362697
std 91.691315 92.863023 90.147881 91.461989 91.454750
min 89.540001 90.129997 88.860001 89.349998 89.349998
25% 135.235004 137.440004 134.822495 136.347498 136.347498
50% 208.764999 212.614998 208.184998 209.920006 209.920006
75% 304.177505 307.565002 295.437500 303.942505 303.942505
max 372.410004 373.829987 361.739990 366.829987 366.829987
Volume
count 2.480000e+02
mean 3.208210e+07
std 2.233590e+07
min 2.657900e+06
25% 1.714180e+07
50% 2.734000e+07
75% 4.771772e+07
max 1.133164e+08
```
Code:
```
# convert date into datetime
df['Date']= pd.to_datetime(df['Date'])
```
Code:
```
# use date as index
df.set_index(df["Date"])
```
Output:
```
Ticker Date Open High Low Close \
Date
```

```
2023-02-07 AAPL 2023-02-07 150.639999 155.229996 150.639999 154.649994
2023-02-08 AAPL 2023-02-08 153.880005 154.580002 151.169998 151.919998
2023-02-09 AAPL 2023-02-09 153.779999 154.330002 150.419998 150.869995
2023-02-10 AAPL 2023-02-10 149.460007 151.339996 149.220001 151.009995
2023-02-13 AAPL 2023-02-13 150.949997 154.259995 150.919998 153.850006
... ... ... ... ... ... ...
2023-05-01 GOOG 2023-05-01 107.720001 108.680000 107.500000 107.709999
2023-05-02 GOOG 2023-05-02 107.660004 107.730003 104.500000 105.980003
2023-05-03 GOOG 2023-05-03 106.220001 108.129997 105.620003 106.120003
2023-05-04 GOOG 2023-05-04 106.160004 106.300003 104.699997 105.209999
2023-05-05 GOOG 2023-05-05 105.320000 106.440002 104.738998 106.214996
Adj Close Volume
Date
2023-02-07 154.414230 83322600
2023-02-08 151.688400 64120100
2023-02-09 150.639999 56007100
2023-02-10 151.009995 57450700
2023-02-13 153.850006 62199000
... ... ...
2023-05-01 107.709999 20926300
2023-05-02 105.980003 20343100
2023-05-03 106.120003 17116300
2023-05-04 105.209999 19780600
2023-05-05 106.214996 20705300
[248 rows x 8 columns]
```
Code:
```
# Crete a Daily Change Column
df['Daily_Change'] = (df['Close'] - df['Open'])/df['Open']*100
```
Code:
```
df['Daily_Change']
```
Code:
```
# filter days where Close Amt > Previous Close Amt
df['Close'].shift(1)
df['Change_Close'] = df['Close'].shift(0) > df['Close'].shift(1)
df['Change_Close']
```
Output:
```
0 False
1 False
2 False
3 True
4 True
...
243 False
244 False
245 True
246 False
247 True
Name: Change_Close, Length: 248, dtype: bool
```
Code:
```
# find the highest and lowest Adj Close price for each company
result = df.groupby("Ticker")['Adj Close'].agg(Highest_Adj_Close = "max" , Lowest_Adj_Close =
"min").reset_index()
result
```
Output:
```
Ticker Highest_Adj_Close Lowest_Adj_Close
0 AAPL 173.570007 145.309998
```

```
1 GOOG 109.459999 89.349998
2 MSFT 310.649994 246.270004
3 NFLX 366.829987 292.760010
```

Code:

```
# calculate avg Adj Close price for each price
Avg_Adj_Price = df.groupby('Ticker')['Adj Close'].mean()
Avg_Adj_Price
```

Output:

```
Ticker
AAPL 158.229397
GOOG 100.631532
MSFT 274.975182
NFLX 327.614677
Name: Adj Close, dtype: float64
```

Code:

Code:

Code:

Code: