Stock_Information

September 29, 2021

1 Stock Information

[1]:
```
# Libraries
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

import sys
import warnings
warnings.filterwarnings("ignore")

from pandas_datareader import data as pdr
import yfinance as yf
yf.pdr_override()
```

[2]:
```
# Check versions of modules used
print("numpy: {}".format(np.__version__))
print("pandas: {}".format(pd.__version__))
print("matplotlib: {}".format(matplotlib.__version__))
print("seaborn: {}".format(sns.__version__))
print("yahoo_finance: {}".format(yf.__version__))
print("python: {}".format(sys.version))
```

```
numpy: 1.17.2
pandas: 0.24.2
matplotlib: 3.0.3
seaborn: 0.9.0
yahoo_finance: 0.1.52
python: 3.5.5 | packaged by conda-forge | (default, Jul 24 2018, 01:52:17) [MSC
v.1900 64 bit (AMD64)]
```

[3]:
```
stock = 'AMD'
start = '2015-01-01'
end = '2018-01-01'
data = pdr.get_data_yahoo(stock, start, end)
```

```
[**********100%**********]  1 of 1 completed
```

[4]:
```
# Inspect the index
data.index
```

[4]:
```
DatetimeIndex(['2015-01-02', '2015-01-05', '2015-01-06', '2015-01-07',
               '2015-01-08', '2015-01-09', '2015-01-12', '2015-01-13',
               '2015-01-14', '2015-01-15',
               …
               '2017-12-15', '2017-12-18', '2017-12-19', '2017-12-20',
               '2017-12-21', '2017-12-22', '2017-12-26', '2017-12-27',
               '2017-12-28', '2017-12-29'],
              dtype='datetime64[ns]', name='Date', length=755, freq=None)
```

[5]:
```
# Inspect the columns
data.columns
```

[5]:
```
Index(['Adj Close', 'Close', 'High', 'Low', 'Open', 'Volume'], dtype='object')
```

[6]:
```
# Type of data
type(data)
```

[6]:
```
pandas.core.frame.DataFrame
```

[7]:
```
data = data.reset_index() # Date has a column
```

[8]:
```
data.head() # First 5 rows
```

[8]:

	Date	Adj Close	Close	High	Low	Open	Volume
0	2015-01-02	2.67	2.67	2.67	2.67	2.67	0
1	2015-01-05	2.66	2.66	2.70	2.64	2.67	8878200
2	2015-01-06	2.63	2.63	2.66	2.55	2.65	13912500
3	2015-01-07	2.58	2.58	2.65	2.54	2.63	12377600
4	2015-01-08	2.61	2.61	2.65	2.56	2.59	11136600

[9]:
```
data.tail() # Last 5 rows
```

[9]:

	Date	Adj Close	Close	High	Low	Open	Volume
750	2017-12-22	10.54	10.54	10.77	10.20	10.75	50744500
751	2017-12-26	10.46	10.46	10.58	10.34	10.38	20437900
752	2017-12-27	10.53	10.53	10.74	10.40	10.45	22921800
753	2017-12-28	10.55	10.55	10.64	10.43	10.57	18609400
754	2017-12-29	10.28	10.28	10.58	10.27	10.57	26678900

[10]:
```
data.describe() # Statistics
```

[10]:
```
         Adj Close       Close        High         Low        Open  \
count  755.000000  755.000000  755.000000  755.000000  755.000000
mean     6.658503    6.658503    6.803576    6.512450    6.660927
std      4.559978    4.559978    4.650876    4.472189    4.572615
min      1.620000    1.620000    1.690000    1.610000    1.620000
25%      2.360000    2.360000    2.430000    2.315000    2.360000
50%      5.100000    5.100000    5.190000    5.000000    5.100000
75%     11.300000   11.300000   11.510000   11.125000   11.300000
max     15.200000   15.200000   15.650000   14.520000   15.450000

             Volume
count  7.550000e+02
mean   3.646362e+07
std    3.476847e+07
min    0.000000e+00
25%    1.146880e+07
50%    2.751560e+07
75%    4.981880e+07
max    2.683365e+08
```

[11]:
```
prices = data['Adj Close']
features = data.drop(['Date','Adj Close', 'Close'], axis = 1)
```

[12]:
```
features.tail()
```

[12]:
```
      High    Low   Open    Volume
750  10.77  10.20  10.75  50744500
751  10.58  10.34  10.38  20437900
752  10.74  10.40  10.45  22921800
753  10.64  10.43  10.57  18609400
754  10.58  10.27  10.57  26678900
```

[13]:
```
print("Stock dataset has {} data points with {} variables each.".format(*data.
 ↪shape))
```

```
Stock dataset has 755 data points with 7 variables each.
```

[14]:
```
# TODO: Minimum price of the data
minimum_price = np.min(prices)

# TODO: Maximum price of the data
maximum_price = np.max(prices)

# TODO: Mean price of the data
mean_price = np.mean(prices)

# TODO: Median price of the data
```

```
median_price = np.median(prices)

# TODO: Standard deviation of prices of the data
std_price = np.std(prices)

# Show the calculated statistics
print("Statistics for Stock dataset:\n")
print("Minimum price: ${:,.2f}".format(minimum_price))
print("Maximum price: ${:,.2f}".format(maximum_price))
print("Mean price: ${:,.2f}".format(mean_price))
print("Median price ${:,.2f}".format(median_price))
print("Standard deviation of prices: ${:,.2f}".format(std_price))
```

```
Statistics for Stock dataset:

Minimum price: $1.62
Maximum price: $15.20
Mean price: $6.66
Median price $5.10
Standard deviation of prices: $4.56
```