Learning_Python_AlphaVantage

December 9, 2020

```
[1]: import pandas as pd
     import yfinance as yf
```

```
[2]: tickerSymbol = 'HD'
```

```
[3]: hd_df = yf.Ticker(tickerSymbol)
```

```
[4]: dt_range = hd_df.history(period='1d', start='2020-12-3', end='2020-12-4')
```

```
[5]: hd_df.recommendations
```

```
[5]:                                  Firm    To Grade  From Grade Action
     Date
     2012-02-22 09:06:00      Jefferies       Hold              main
     2012-02-22 11:12:00            UBS        Buy              main
     2012-03-20 06:30:00  Deutsche Bank       Hold              main
     2012-03-21 06:10:00  Credit Suisse Outperform              main
     2012-03-26 08:48:00 Canaccord Genuity    Hold              init
     …                               …          …          …    …
     2020-08-19 13:58:34 B of A Securities     Buy    Neutral     up
     2020-09-18 15:11:12     Oppenheimer   Perform Outperform  down
     2020-10-07 11:33:40 Morgan Stanley Overweight              main
     2020-11-12 12:30:33 Gordon Haskett        Buy Accumulate     up
     2020-12-04 14:55:12 Morgan Stanley Overweight              main
     [217 rows x 4 columns]
```

```
[6]: # quandl api key = Quandl_API_KEY
     # alpha_vantage key = ALPHA_API_KEY
```

```
[7]: import pandas as pd
     from alpha_vantage.timeseries import TimeSeries
     import time
```

```
[8]: api_key = 'API_KEY'
```

```
[9]: ts = TimeSeries(key=api_key, output_format='json')
```

```
[10]: data, meta_data = ts.get_intraday(symbol='HD', interval = '1min', outputsize =
       ↪'full')
```

```
[11]: print(data)
```

'272.6216', '4. close': '272.7608', '5. volume': '6338'}, '2020-11-25 09:41:00': {'1. open': '272.9895', '2. high': '273.2282', '3. low': '272.9348', '4. close': '273.0114', '5. volume': '8730'}, '2020-11-25 09:40:00': {'1. open': '272.6216', '2. high': '272.9895', '3. low': '272.5967', '4. close': '272.9895', '5. volume': '7186'}, '2020-11-25 09:39:00': {'1. open': '272.8951', '2. high':
'272.9895', '3. low': '272.6216', '4. close': '272.6216', '5. volume': '5955'}, '2020-11-25 09:38:00': {'1. open': '272.7672', '2. high': '272.9697', '3. low':
'272.4078', '4. close': '272.9697', '5. volume': '11220'}, '2020-11-25 09:37:00': {'1. open': '273.0459', '2. high': '273.1387', '3. low': '272.6464', '4. close': '272.6763', '5. volume': '9317'}, '2020-11-25 09:36:00': {'1. open':
'272.6415', '2. high': '273.1288', '3. low': '272.6216', '4. close': '272.9398',
'5. volume': '9684'}, '2020-11-25 09:35:00': {'1. open': '272.9597', '2. high':
'272.9597', '3. low': '272.5917', '4. close': '272.6962', '5. volume': '8295'}, '2020-11-25 09:34:00': {'1. open': '272.5619', '2. high': '273.0144', '3. low':
'272.4664', '4. close': '272.9299', '5. volume': '8993'}, '2020-11-25 09:33:00': {'1. open': '272.5619', '2. high': '272.6912', '3. low': '272.2138', '4. close': '272.5448', '5. volume': '13251'}, '2020-11-25 09:32:00': {'1. open': '272.5221', '2. high': '272.7906', '3. low': '272.4923', '4. close': '272.7310',
'5. volume': '4618'}, '2020-11-25 09:31:00': {'1. open': '271.9752', '2. high':
'272.5520', '3. low': '271.8160', '4. close': '272.5122', '5. volume':
'141661'}, '2020-11-25 09:30:00': {'1. open': '271.8558', '2. high': '271.8558',
'3. low': '271.8558', '4. close': '271.8558', '5. volume': '260'}, '2020-11-25
09:23:00': {'1. open': '272.0945', '2. high': '272.0945', '3. low': '272.0945', '4. close': '272.0945', '5. volume': '1839'}, '2020-11-25 09:11:00': {'1. open':
'272.3730', '2. high': '272.3730', '3. low': '272.3730', '4. close': '272.3730', '5. volume': '487'}, '2020-11-25 08:53:00': {'1. open': '272.4923', '2. high':
'272.4923', '3. low': '272.4923', '4. close': '272.4923', '5. volume': '420'},
'2020-11-25 08:48:00': {'1. open': '272.4923', '2. high': '272.4923', '3. low': '272.4923', '4. close': '272.4923', '5. volume': '280'}, '2020-11-25 08:35:00':
{'1. open': '272.5221', '2. high': '273.4768', '3. low': '272.5221', '4. close': '273.4768', '5. volume': '713'}, '2020-11-25 08:15:00':

```
{'1. open': '273.4673', '2. high': '273.4673', '3. low': '273.4673',
'4. close': '273.4673', '5. volume': '104'}, '2020-11-25 08:12:00':
{'1. open': '273.0890', '2. high': '273.0890', '3. low': '273.0890',
'4. close': '273.0890', '5. volume': '106'}, '2020-11-25 08:01:00':
{'1. open': '273.4868', '2. high': '273.4868', '3. low':
'273.4868', '4. close': '273.4868', '5. volume': '1329'}}
```

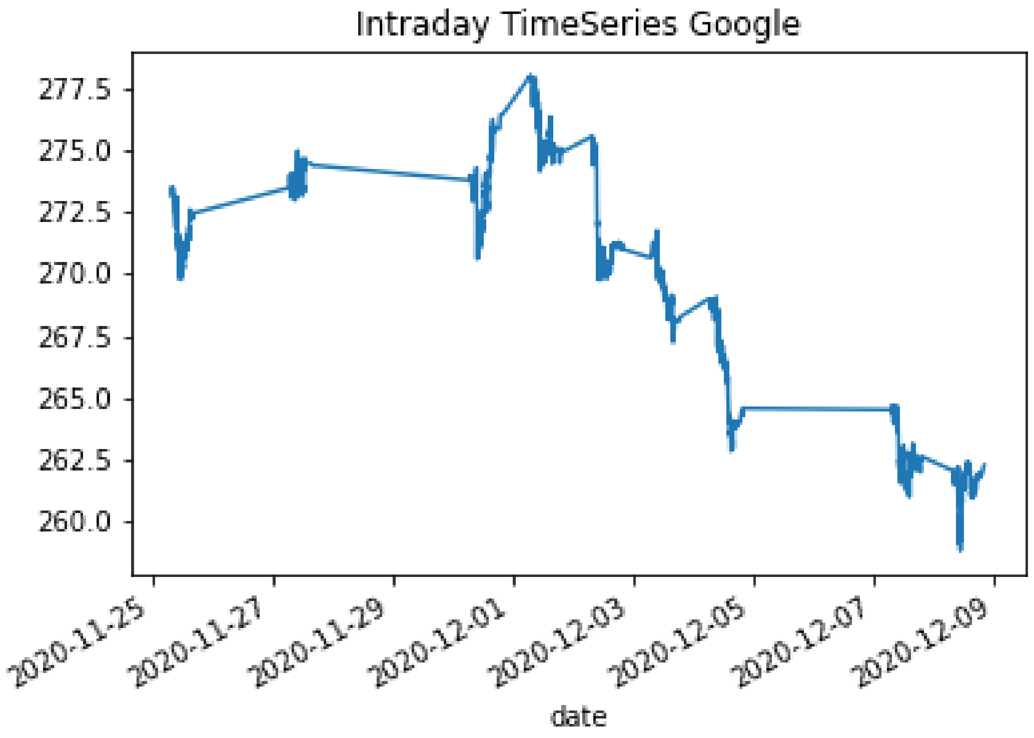
[12]:
```
import pandas as pd
from alpha_vantage.timeseries import TimeSeries
import time
import matplotlib.pyplot as plt
```

[13]:
```
from alpha_vantage.timeseries import TimeSeries
import matplotlib.pyplot as plt
api_key = 'API_KEY'
ts = TimeSeries(key='api_key',output_format='pandas')
data, meta_data = ts.get_intraday(symbol='HD',interval='1min',
    outputsize='full')
print(data)
```

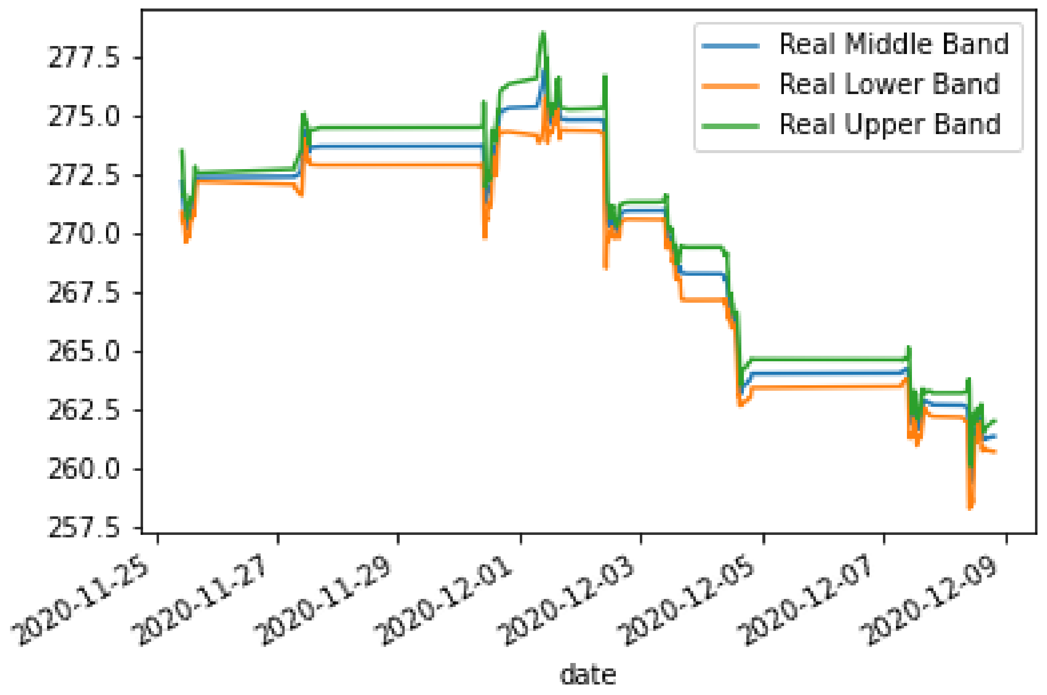
```
                     1. open   2. high    3. low  4. close  5. volume
date

2020-12-08 20:00:00 262.2900 262.2900 262.2900          138.0
262.2900
2020-12-08 18:49:00 261.8600 261.8600 261.8600          315.0
261.8600
2020-12-08 18:40:00 261.8001 261.8001 261.8001          377.0
261.8001
2020-12-08 18:39:00 262.0000 262.0000 262.0000          355.0
262.0000
2020-12-08 18:00:00 261.9500 261.9500 261.9500          156.0
261.9500
…                        …        …        …        …         …
2020-11-25 08:48:00 272.4923 272.4923 272.4923          280.0
272.4923
2020-11-25 08:35:00 272.5221 273.4768 272.5221          713.0
273.4768
2020-11-25 08:15:00 273.4673 273.4673 273.4673          104.0
273.4673
2020-11-25 08:12:00 273.0890 273.0890 273.0890          106.0
273.0890
2020-11-25 08:01:00 273.4868 273.4868 273.4868         1329.0
273.4868
[3655 rows x 5 columns]
```

[14]:
```
data['4. close'].plot()
plt.title('Intraday TimeSeries Google')
plt.show()
```



```
[15]: from alpha_vantage.techindicators import
TechIndicators api_key = 'API_KEY'
tsi = TechIndicators(key='api_key',output_format='pandas') data,
meta_data = tsi.get_bbands(symbol='HD',interval='1min',
time_period=60) data.plot() plt.show()
```

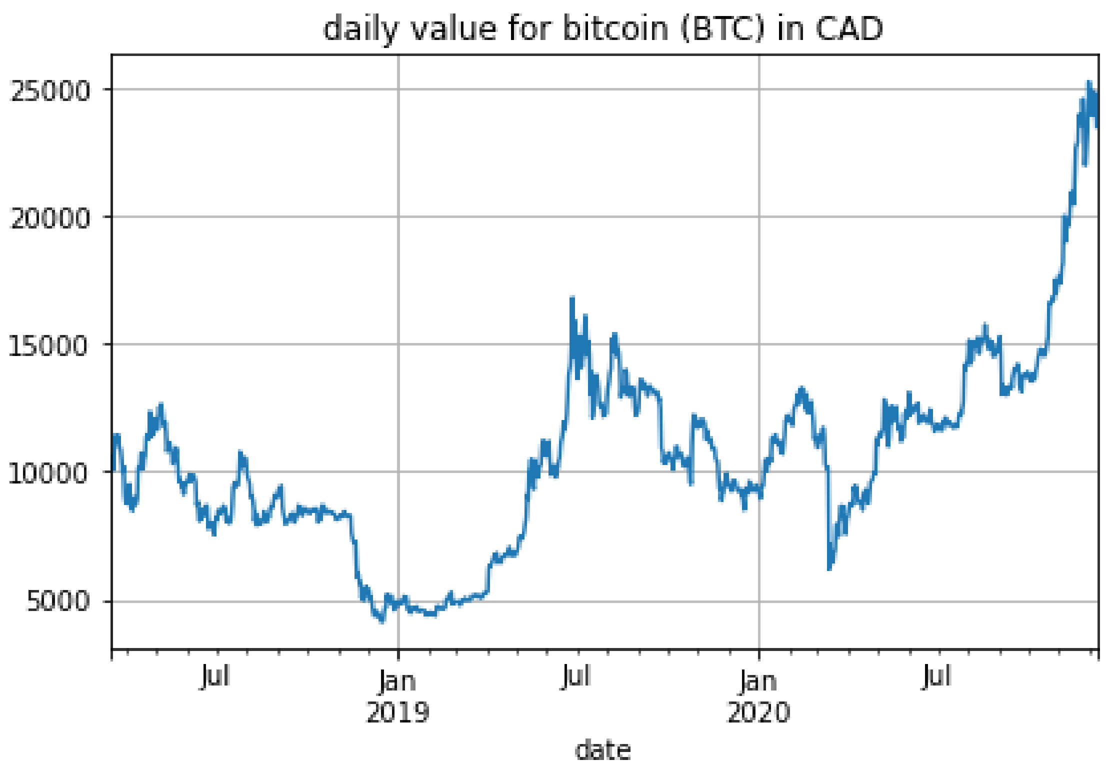


[16]:
```
from alpha_vantage.cryptocurrencies import
CryptoCurrencies import matplotlib.pyplot as plt
api_key = API_KEY'
cc = CryptoCurrencies(key='api_key',output_format='pandas') data,
meta_data = cc.get_digital_currency_daily(symbol='BTC',
market='CAD') print(data) data['1a. open (CAD)'].plot()
plt.tight_layout() plt.title('daily value
for bitcoin (BTC) in CAD') plt.grid()
plt.show()
```

```
           1a. open (CAD)  1b. open (USD)  2a. high (CAD)  2b. high (USD)  \
date
2020-12-09   23476.849732        18324.11    23548.456000        18380.00
2020-12-08   24556.632280        19166.90    24720.549008        19294.84
2020-12-07   24802.328004        19358.67    24882.069892        19420.91
2020-12-06   24531.981992        19147.66    24880.904000        19420.00
2020-12-05   23895.033412        18650.51    24569.572400        19177.00
…                     …              …             …              …
2018-03-20   11011.926812         8595.01    11594.860000         9050.00
2018-03-19   10491.746800         8189.00    11153.140676         8705.23
2018-03-18   10024.121612         7824.01    10656.252880         8317.40
2018-03-17   10582.712000         8260.00    10696.251944         8348.62
2018-03-16   10558.343576         8240.98    11033.233168         8611.64
           3a. low (CAD)  3b. low (USD)  4a. close (CAD)  4b. close (USD)  \
 date
2020-12-09 23215.344000    18120.00   23330.652000       18210.00
2020-12-08 23317.840000    18200.00   23476.849732       18324.11
2020-12-07 24218.369856    18902.88   24556.632280       19166.90
2020-12-06 24159.588400    18857.00   24803.263280       19359.40
2020-12-05 23702.200000    18500.00   24531.981992       19147.66
 …                …           …             …            …
2018-03-20 10608.336000     8280.00   11415.466376        8909.98
2018-03-19 10362.858080     8088.40   11018.320000        8600.00
2018-03-18  9380.946400     7322.00   10493.015188        8189.99
2018-03-17  9893.413588     7721.99   10025.133760        7824.80
2018-03-16 10121.480000     7900.00   10582.712000        8260.00

            5. volume 6. market cap (USD)
 date
 2020-12-09 4338.778160 4338.778160 2020-12-08
 61626.947614 61626.947614
2020-12-07 41372.296293      41372.296293
2020-12-06 37043.091861      37043.091861
2020-12-05 42922.748573      42922.748573
 …               …                …
2018-03-20 44865.105835      44865.105835
2018-03-19 55297.084942      55297.084942
2018-03-18 59488.231711      59488.231711
2018-03-17 33110.206329      33110.206329
2018-03-16 38815.409893      38815.409893

 [1000 rows x 10 columns]
```



```
[17]: from alpha_vantage.timeseries import TimeSeries
import matplotlib.pyplot as plt api_key = 'API_KEY'
ts = TimeSeries(key='api_key',output_format='pandas')
data, meta_data =
ts.get_intraday(symbol='HD',interval='1min',
outputsize='full') print(data)
```

```
                     1. open  2. high   3. low 4. close 5. volume
date
2020-12-08 20:00:00 262.2900 262.2900 262.2900         138.0
262.2900
2020-12-08 18:49:00 261.8600 261.8600 261.8600         315.0
261.8600
2020-12-08 18:40:00 261.8001 261.8001 261.8001         377.0
261.8001
2020-12-08 18:39:00 262.0000 262.0000 262.0000         355.0
262.0000
2020-12-08 18:00:00 261.9500 261.9500 261.9500         156.0
261.9500
…                          …        …        …        …         …
2020-11-25 08:48:00 272.4923 272.4923 272.4923         280.0
272.4923
```

```
2020-11-25 08:35:00 272.5221 273.4768 272.5221          713.0
273.4768
2020-11-25 08:15:00 273.4673 273.4673 273.4673          104.0
273.4673
2020-11-25 08:12:00 273.0890 273.0890 273.0890          106.0
273.0890
2020-11-25 08:01:00 273.4868 273.4868 273.4868         1329.0
273.4868
[3655 rows x 5 columns]
```

```
[19]: close_data = data['4. close']
      percentage_change = close_data.pct_change()

      print(percentage_change)
```

```
date
2020-12-08 20:00:00        NaN
2020-12-08         -
18:49:00           0.001639
2020-12-08         -
18:40:00           0.000229
2020-12-08          0.000764
18:39:00
2020-12-08         -
18:00:00           0.000191
                      …
2020-11-25          0.000000
08:48:00
2020-11-25          0.003613
08:35:00
2020-11-25         -
08:15:00           0.000035
2020-11-25         -
08:12:00           0.001383
2020-11-25          0.001457
08:01:00
Name: 4. close, Length: 3655, dtype: float64
```

```
[20]: change = percentage_change[-1]
```

```
[21]: print(change)
```

```
0.0014566679727121556
```

```
[22]: if abs(change) > 0.0004:
          print("HD Alert: ", change)
```

```
HD Alert: 0.0014566679727121556
```