SURIYA S

225229140 DVA LAB 8

In [1]:
```
#importing libraries
import pandas as pd
from dateutil.parser import parse
import matplotlib.pyplot as plt
```

In [ ]:

In [6]:
```
#importing data
data=pd.read_csv('amazon_stock.csv')
data.head()
```

Out[6]:
```
   None  ticker       Date     Open     High      Low    Close   Volume  Adj_Close
0     0    AMZN  3/27/2018  1572.40  1575.96  1482.32  1497.05  6793279    1497.05
1     1    AMZN  3/26/2018  1530.00  1556.99  1499.25  1555.86  5547618    1555.86
2     2    AMZN  3/23/2018  1539.01  1549.02  1495.36  1495.56  7843966    1495.56
3     3    AMZN  3/22/2018  1565.47  1573.85  1542.40  1544.10  6177737    1544.10
4     4    AMZN  3/21/2018  1586.45  1590.00  1563.17  1581.86  4667291    1581.86
```

In [10]:
```
data.drop(['None','ticker'],axis=1)
```

```
 ...         ...     ...       ...       ...       ...       ...       ...
1286   2/13/2013  261.53  269.9600  260.3000  269.4700   5293000  269.4700
1287  02-12-2013  259.19  260.1600  257.0000  258.7000   2943700  258.7000
1288  02-11-2013  263.20  263.2500  256.6000  257.2100   3403700  257.2100
1289  02-08-2013  261.40  265.2500  260.5550  261.9500   3879200  261.9500
1290  02-07-2013  264.10  264.1000  255.1100  260.2300   3975700  260.2300
1291  02-06-2013  265.16  266.8900  261.1100  262.2200   2770400  262.2200
1292  02-05-2013  262.00  268.0300  261.4600  266.8900   4012900  266.8900
1293  02-04-2013  262.78  264.6840  259.0700  259.9800   3723600  259.9800
1294  02-01-2013  268.93  268.9300  262.8000  265.0000   6115000  265.0000
1295   1/31/2013  271.04  275.9400  263.6991  265.5000   6772100  265.5000
1296   1/30/2013  283.00  284.2000  267.1100  272.7640  13075400  272.7640
```

In [18]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1316 entries, 0 to 1315
Data columns (total 9 columns):
None         1316 non-null int64
ticker       1316 non-null object
Date         1316 non-null object
Open         1316 non-null float64
High         1316 non-null float64
Low          1316 non-null float64
Close        1316 non-null float64
Volume       1316 non-null int64
Adj_Close    1316 non-null float64
dtypes: float64(5), int64(2), object(2)
memory usage: 92.6+ KB
```

In [19]:
```
data=pd.read_csv('amazon_stock.csv',parse_dates=['Date'])
```

In [22]:
```
data.drop(['None','ticker'], axis=1, inplace=True)
```

In [23]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1316 entries, 0 to 1315
Data columns (total 7 columns):
Date         1316 non-null datetime64[ns]
Open         1316 non-null float64
High         1316 non-null float64
Low          1316 non-null float64
Close        1316 non-null float64
Volume       1316 non-null int64
Adj_Close    1316 non-null float64
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 72.0 KB
```

In [24]:
```
data.head()
```

Out[24]:

	Date	Open	High	Low	Close	Volume	Adj_Close
0	2018-03-27	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
1	2018-03-26	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2	2018-03-23	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
3	2018-03-22	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
4	2018-03-21	1586.45	1590.00	1563.17	1581.86	4667291	1581.86

In [25]:
```
data.set_index('Date',inplace=True)
```

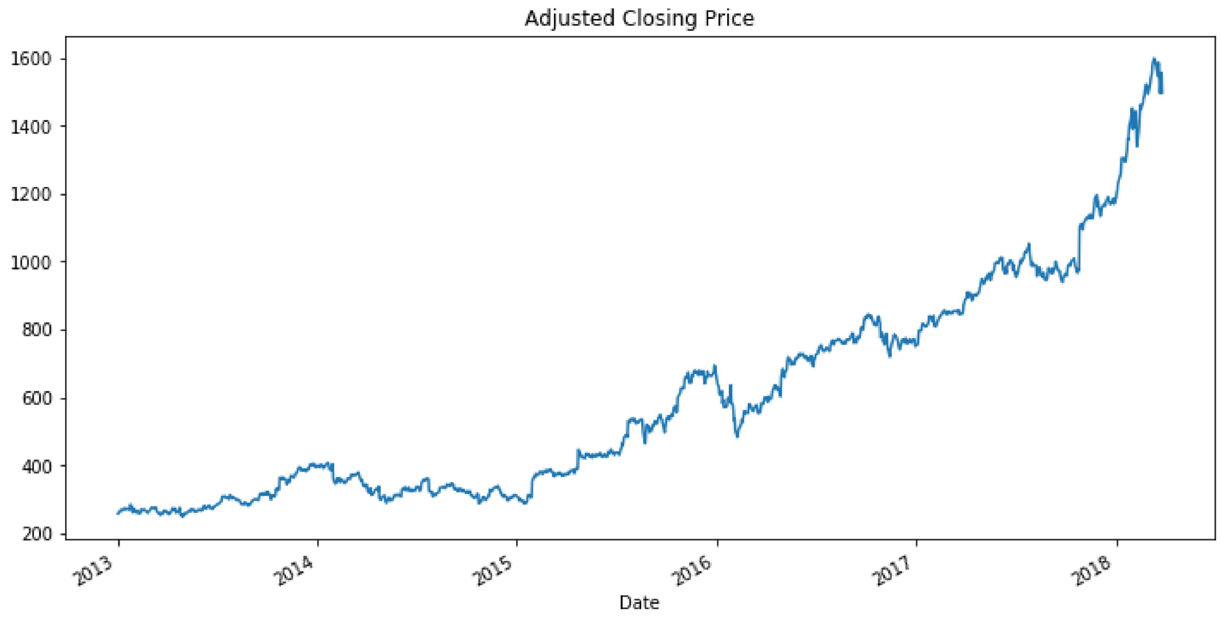
In [26]:
```
data.head()
```

Out[26]:

	Open	High	Low	Close	Volume	Adj_Close
Date						
2018-03-27	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
2018-03-26	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2018-03-23	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
2018-03-22	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
2018-03-21	1586.45	1590.00	1563.17	1581.86	4667291	1581.86

In [28]:
```
data['Adj_Close'].plot(figsize=(12,6),title='Adjusted Closing Price')
```

Out[28]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2c4e99ea2b0>
```



In [29]:
```
from datetime import datetime
my_year=2020
my_month=5
my_day=1
my_hour=13
my_minute=36
my_second=45
test_date=datetime(my_year,my_month,my_day)
test_date
```

Out[29]:
```
datetime.datetime(2020, 5, 1, 0, 0)
```

In [30]:

```
test_date=datetime(my_year,my_month,my_day,my_hour,my_minute,my_second)
print("The day is :",test_date.day)
print("The hour is :",test_date.hour)
print("The month is :",test_date.month)
```

```
The day is : 1
The hour is : 13
The month is : 5
```

In [31]:

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 1316 entries, 2018-03-27 to 2013-01-02
Data columns (total 6 columns):
Open         1316 non-null float64
High         1316 non-null float64
Low          1316 non-null float64
Close        1316 non-null float64
Volume       1316 non-null int64
Adj_Close    1316 non-null float64
dtypes: float64(5), int64(1)
memory usage: 72.0 KB
```

In [33]:

```
print(data.index.max())
print(data.index.min())
```

```
2018-03-27 00:00:00
2013-01-02 00:00:00
```

In [34]:

```
data.index.argmin()
```

Out[34]:

```
1315
```

In [35]:

```
data.index.argmax()
```

Out[35]:

```
0
```

In [36]:

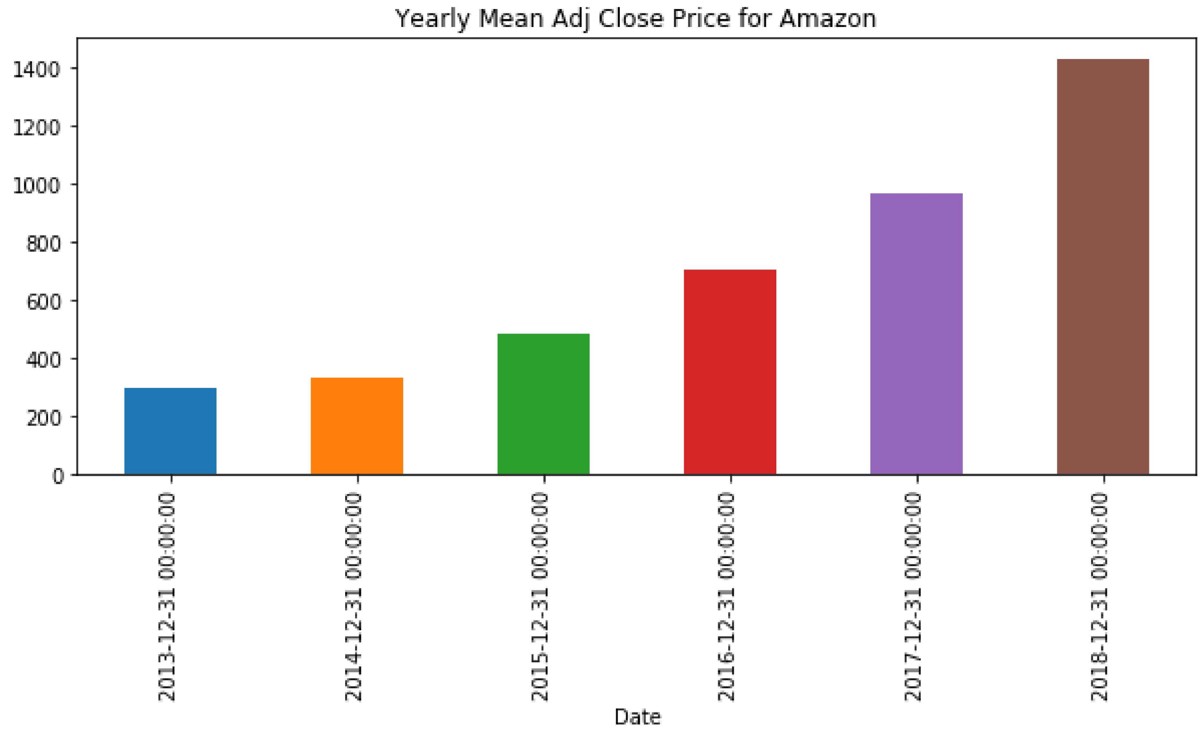
```
data.resample('Y').mean()
```

Out[36]:

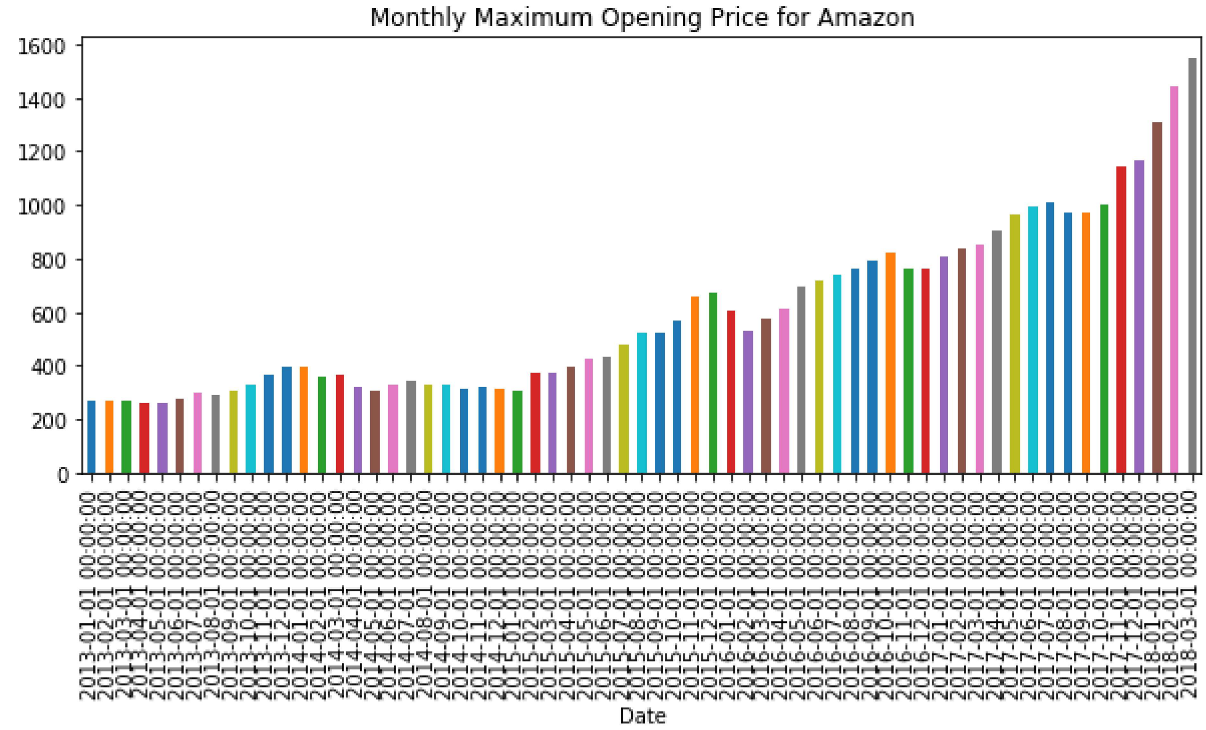
	Open	High	Low	Close	Volume	Adj_Close
Date						
2013-12-31	297.877223	300.925966	294.656658	298.032235	2.967880e+06	298.032235
2014-12-31	332.798433	336.317462	328.545440	332.550976	4.083223e+06	332.550976
2015-12-31	478.126230	483.248272	472.875443	478.137321	3.797801e+06	478.137321
2016-12-31	699.669762	705.799103	692.646189	699.523135	4.122043e+06	699.523135
2017-12-31	967.565060	973.789752	959.991826	967.403996	3.466207e+06	967.403996
2018-12-31	1429.770000	1446.701017	1409.469661	1429.991186	5.586829e+06	1429.991186

In [42]:

```
data['Adj_Close'].resample('A').mean().plot(kind='bar', figsize=(10,4))
plt.title('Yearly Mean Adj Close Price for Amazon')
plt.show()
```

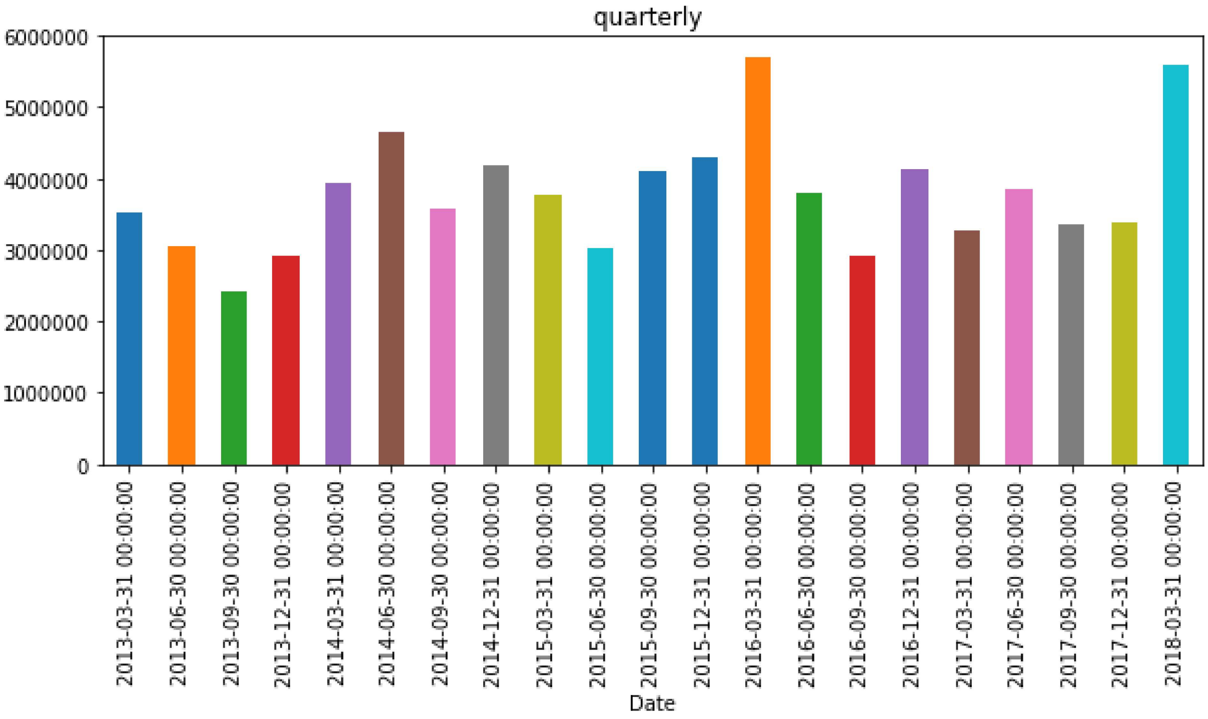


```
In [46]: data['Adj_Close'].resample('MS').mean().plot(kind='bar', figsize=(10,4))
         plt.title('Monthly Maximum Opening Price for Amazon')
         plt.show()
```



In [59]:

```
data['Volume'].resample('Q').mean().plot(kind='bar', figsize=(10,4))
plt.title('quarterly ')
plt.show()
```



In [52]:
```
data.head()
```

Out[52]:
```
               Open     High      Low    Close   Volume  Adj_Close
      Date
2018-03-27  1572.40  1575.96  1482.32  1497.05  6793279    1497.05
2018-03-26  1530.00  1556.99  1499.25  1555.86  5547618    1555.86
2018-03-23  1539.01  1549.02  1495.36  1495.56  7843966    1495.56
2018-03-22  1565.47  1573.85  1542.40  1544.10  6177737    1544.10
2018-03-21  1586.45  1590.00  1563.17  1581.86  4667291    1581.86
```

In [53]:
```
data.shift(1, axis=0).head(5)
```

Out[53]:
```
               Open     High      Low    Close     Volume  Adj_Close
      Date
2018-03-27      NaN      NaN      NaN      NaN        NaN        NaN
2018-03-26  1572.40  1575.96  1482.32  1497.05  6793279.0    1497.05
2018-03-23  1530.00  1556.99  1499.25  1555.86  5547618.0    1555.86
2018-03-22  1539.01  1549.02  1495.36  1495.56  7843966.0    1495.56
2018-03-21  1565.47  1573.85  1542.40  1544.10  6177737.0    1544.10
```

In [54]:
```
data.shift(-1, axis=0).head(5)
```

Out[54]:
```
               Open     High      Low    Close     Volume  Adj_Close
      Date
2018-03-27  1530.00  1556.99  1499.25  1555.86  5547618.0    1555.86
2018-03-26  1539.01  1549.02  1495.36  1495.56  7843966.0    1495.56
2018-03-23  1565.47  1573.85  1542.40  1544.10  6177737.0    1544.10
2018-03-22  1586.45  1590.00  1563.17  1581.86  4667291.0    1581.86
2018-03-21  1550.34  1587.00  1545.41  1586.51  4507049.0    1586.51
```

In [55]:

```
data.shift(periods=3, freq='M').head()
```

Out[55]:

	Open	High	Low	Close	Volume	Adj_Close
Date						
2018-05-31	1572.40	1575.96	1482.32	1497.05	6793279	1497.05
2018-05-31	1530.00	1556.99	1499.25	1555.86	5547618	1555.86
2018-05-31	1539.01	1549.02	1495.36	1495.56	7843966	1495.56
2018-05-31	1565.47	1573.85	1542.40	1544.10	6177737	1544.10
2018-05-31	1586.45	1590.00	1563.17	1581.86	4667291	1581.86

In [56]:

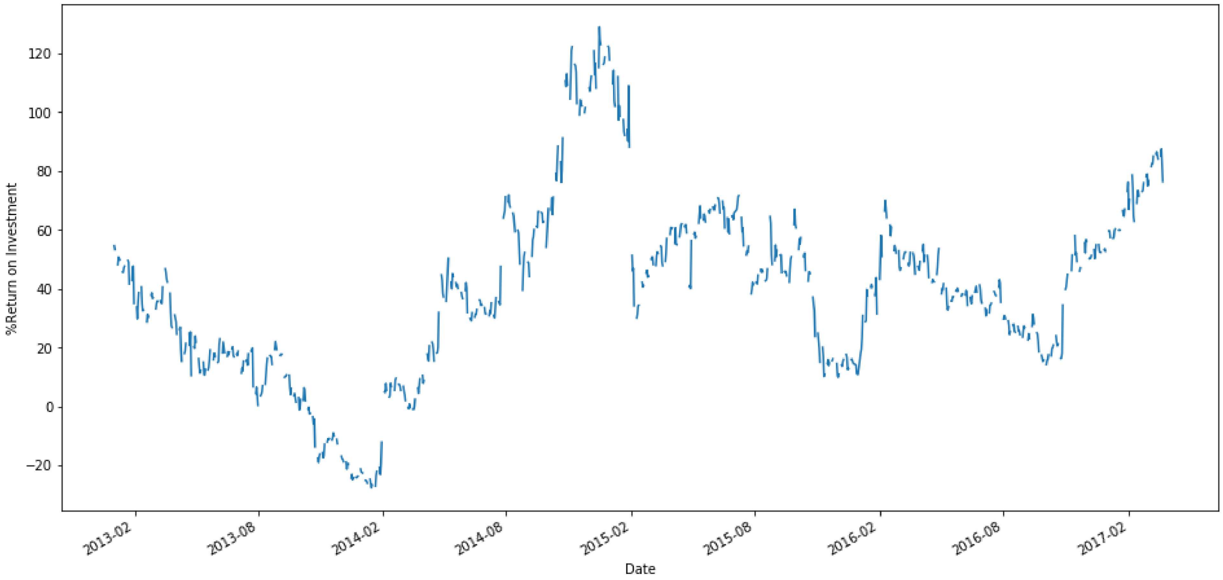
```
import warnings
warnings.filterwarnings('ignore')
```

In [57]:

```
ROI = 100*(data['Adj_Close'].tshift(periods=-365, freq='D')/data['Adj_Close']-1)
ROI.plot(figsize=(16,8))
plt.ylabel('%Return on Investment')
```

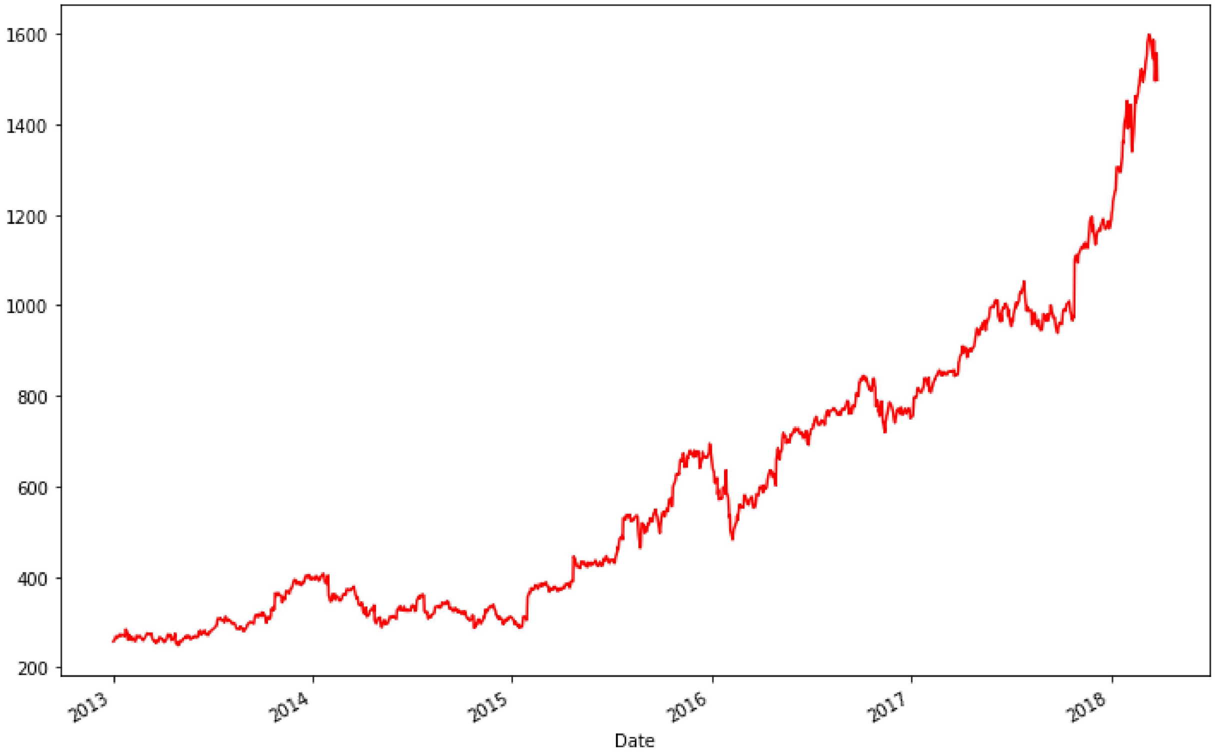
Out[57]:

```
Text(0,0.5,'%Return on Investment')
```



In [60]:
```
data['Adj_Close'].plot(figsize=(12,8), color='red')
```

Out[60]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2c4ed3d7dd8>
```



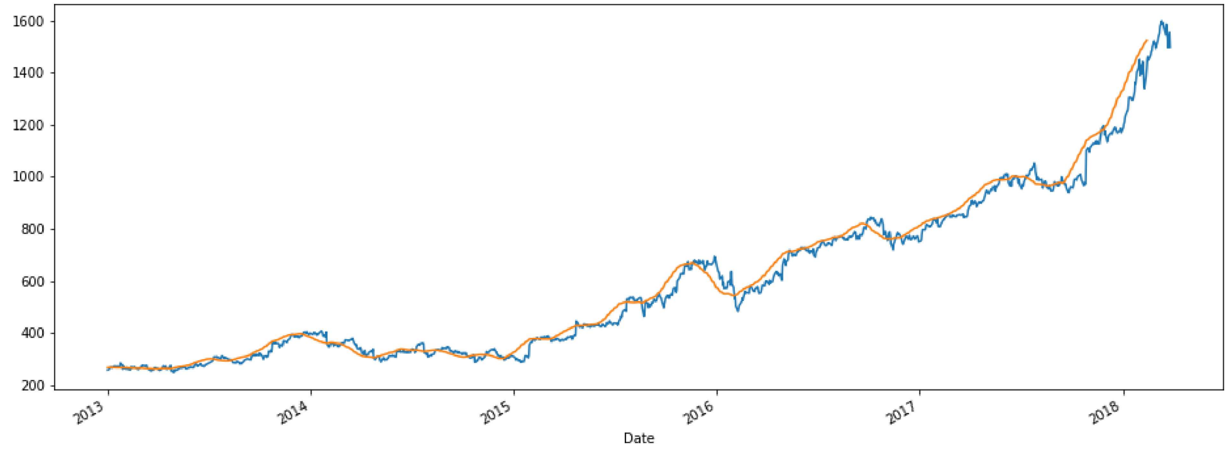
In [61]:
```
data.rolling(7).mean().head(10)
```

Out[61]:

Date	Open	High	Low	Close	Volume	Adj_Close
2018-03-27	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-26	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-23	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-22	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-21	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-20	NaN	NaN	NaN	NaN	NaN	NaN
2018-03-19	1556.885714	1570.640000	1521.894286	1543.695714	5.987651e+06	1543.695714
2018-03-16	1558.464286	1572.565714	1534.062857	1554.357143	5.752191e+06	1554.357143
2018-03-15	1567.750000	1578.268571	1545.328571	1558.137143	5.534923e+06	1558.137143
2018-03-14	1576.034286	1586.471429	1558.975714	1571.771429	5.009270e+06	1571.771429

In [62]:
```
data['Adj_Close'].plot()
data.rolling(window=30).mean()['Adj_Close'].plot(figsize=(16,6))
```

Out[62]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x2c4e9f504a8>
```



In [ ]: