

```
In [1]: #import libraries and Load data

import matplotlib.pyplot as plt
import yfinance as yf
data = yf.download('AAPL', '2016-01-01', '2018-01-01')
data
```

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

Out[1]:

	Open	High	Low	Close	Adj Close	Volume
Date						
2015-12-31	107.010002	107.029999	104.820000	105.260002	98.129669	40912300
2016-01-04	102.610001	105.370003	102.000000	105.349998	98.213585	67649400
2016-01-05	105.750000	105.849998	102.410004	102.709999	95.752419	55791000
2016-01-06	100.559998	102.370003	99.870003	100.699997	93.878586	68457400
2016-01-07	98.680000	100.129997	96.430000	96.449997	89.916473	81094400
...
2017-12-22	174.679993	175.419998	174.500000	175.009995	169.466919	16349400
2017-12-26	170.800003	171.470001	169.679993	170.570007	165.167572	33185500
2017-12-27	170.100006	170.779999	169.710007	170.600006	165.196625	21498200
2017-12-28	171.000000	171.850006	170.479996	171.080002	165.661407	16480200
2017-12-29	170.520004	170.589996	169.220001	169.229996	163.869995	25999900

504 rows × 6 columns

In [2]: *#Considering Adj Close data for further analysis*

```
data1=data[['Adj Close']]  
data1
```

Out[2]:

	Adj Close
Date	
2015-12-31	98.129669
2016-01-04	98.213585
2016-01-05	95.752419
2016-01-06	93.878586
2016-01-07	89.916473
...	...
2017-12-22	169.466919
2017-12-26	165.167572
2017-12-27	165.196625
2017-12-28	165.661407
2017-12-29	163.869995

504 rows × 1 columns

In [3]: *#Use the pandas rolling function for calculating the moving average, and add the result as a new column to the dataframe*

```
data1['100MA'] = data1['Adj Close'].rolling(window=100).mean()
data1
```

C:\job\lib\site-packages\ipykernel_launcher.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy

This is separate from the ipykernel package so we can avoid doing imports until

Out[3]:

	Adj Close	100MA
Date		
2015-12-31	98.129669	NaN
2016-01-04	98.213585	NaN
2016-01-05	95.752419	NaN
2016-01-06	93.878586	NaN
2016-01-07	89.916473	NaN
...
2017-12-22	169.466919	157.916578
2017-12-26	165.167572	158.073095
2017-12-27	165.196625	158.222021
2017-12-28	165.661407	158.352337
2017-12-29	163.869995	158.452533

504 rows × 2 columns

In [4]: *# The 99 first moving average values show as NaN.
This is because we needed the first 100 Adj Close observations to calculate the MA (Moving Average).*

In [5]: *#Plotting Adj Close and Moving Average*

```
fig = plt.figure(figsize=(20,12))
ax1 = plt.plot(data1['Adj Close'])
ax1 = plt.plot(data1['100MA'])
ax1 = plt.title('Daily Adj Close Report', fontsize=22)
ax1 = plt.xlabel('Date', fontsize=18)
ax1 = plt.ylabel('Adj Close', fontsize=18)
ax1 = plt.legend(['Adj Close', '100 day SMA'],prop={'size':20}, loc='upper left')
plt.grid(True)
plt.show()
```

