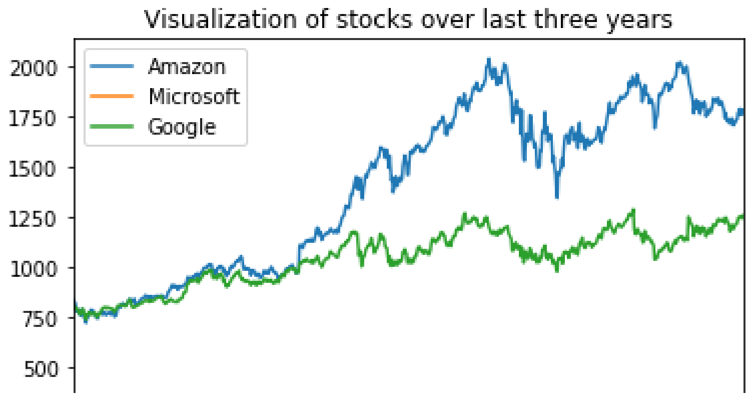
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller, kpss
from statsmodels.tsa.stattools import acf, pacf
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima_model import ARMA
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score


from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_r
```

```
amazon_data=pd.read_csv('AMZN.csv')
print(amazon_data.head())

microsoft_data=pd.read_csv('MSFT.csv')
print(microsoft_data.head())

google_data=pd.read_csv('GOOG.csv')
print(google_data.head())
```

```
         Date        Open        High  ...       Close   Adj Close    Volume
0  2016-10-24  824.950012  838.299988  ...  838.090027  838.090027   4060900
1  2016-10-25  839.299988  843.090027  ...  835.179993  835.179993   3248400
2  2016-10-26  832.760010  833.440002  ...  822.590027  822.590027   3998100
3  2016-10-27  831.239990  831.719971  ...  818.359985  818.359985   7406400
4  2016-10-28  782.000000  789.489990  ...  776.320007  776.320007  10841100

[5 rows x 7 columns]
         Date       Open       High        Low      Close  Adj Close    Volume
0  2016-10-24  59.939999  61.000000  59.930000  61.000000  57.605263  54067000
1  2016-10-25  60.849998  61.369999  60.799999  60.990002  57.595825  35137200
2  2016-10-26  60.810001  61.200001  60.470001  60.630001  57.255852  29911600
3  2016-10-27  60.610001  60.830002  60.090000  60.099998  56.755356  28479900
4  2016-10-28  60.009998  60.520000  59.580002  59.869999  56.538151  33574700
         Date        Open        High  ...       Close   Adj Close   Volume
0  2016-10-24  804.900024  815.179993  ...  813.109985  813.109985  1697500
1  2016-10-25  816.679993  816.679993  ...  807.669983  807.669983  1576400
2  2016-10-26  806.340027  806.979980  ...  799.070007  799.070007  1647700
```

```python
amazon_stocks_data=amazon_data['Close']
microsoft_stocks_data=microsoft_data['Close']
google_stocks_data=google_data['Close']


amazon_stocks_data.plot(label='Amazon')
microsoft_stocks_data.plot(label='Microsoft')
google_stocks_data.plot(label='Google')
plt.legend(loc='best')
plt.title('Visualization of stocks over last three years')
plt.show()
```



Analysis:

Amazon: We see that Amazon has the highest stock values of all the three companies and there has been a steep rise in the stock value in the middle of the year 2017. Since, the beginning of May 2019 however, we see that there is a steady decline in the stock values of Amazon.

Google: Google stocks are relatively high as compared to Microsoft but are lower compared to Amazon. However, there is a gradual increase in the stock values from the year 2019 onwards and we see that if this upward trend continues and the downward trend of Amazon continues, it might overtake the stock values of Amazon soon.

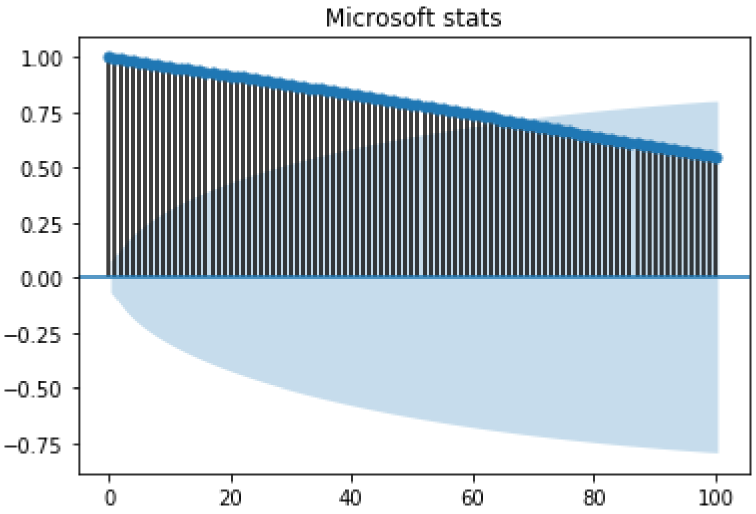
Microsoft: The stock values are very low as compared to the other two companies, we also notice that there is a relatively stagnant trend in the sense that the slope of the upward increase of the stock values is very low. Hence, the Microsoft stock values are relatively stable as compared to the other two companies.

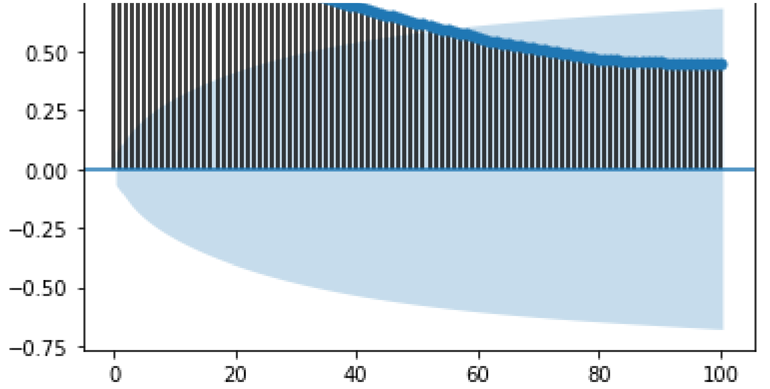
In terms of stability, on analysing the stocks, we clearly see that Microsoft is the most stable of stocks followed by Google and then Amazon. Amazon stocks are by far the most fluctuating.

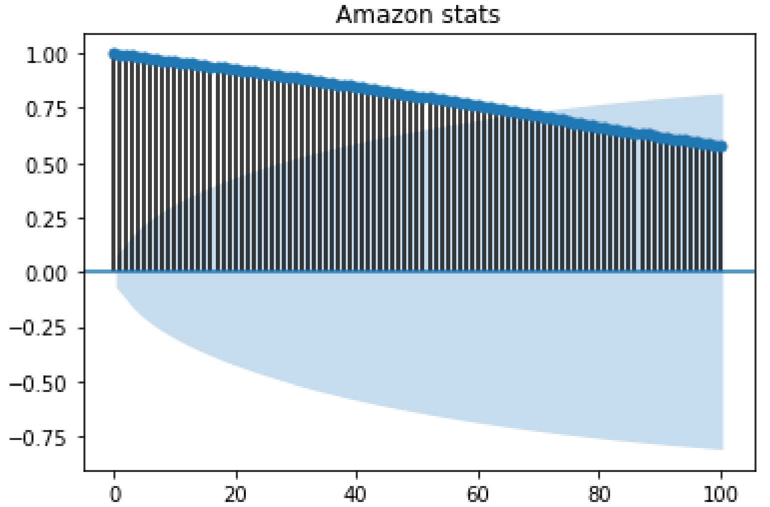
```
def assess_correlation_and_plot(close_stocks, name):
  data=close_stocks.values
  hyp=adfuller(data)
  plot_data=close_stocks.tolist()

  dfs=hyp[0]
  p_val=hyp[1]
  critical_vals=hyp[4]

  print('Dickey-Fuller statistic:', str(hyp[0]))
  print('P Value statistic:', str(hyp[1]))
```

```
    for key in critical_vals:
      print(key+' critical value:', str(hyp[4][key]))

    plot_acf(data, lags=100)
    plt.title(name)

  print('Stats for microsoft')
  assess_correlation_and_plot(microsoft_stocks_data, 'Microsoft stats')
  print()

  print('Stats for google')
  assess_correlation_and_plot(google_stocks_data, 'Google stats')
  print()

  print('Stats for amazon')
  assess_correlation_and_plot(amazon_stocks_data, 'Amazon stats')
  print()
```

```
Stats for microsoft
Dickey-Fuller statistic: 0.01603315321271087
P Value statistic: 0.9598381019798699
1% critical value: -3.439134355513998
5% critical value: -2.865416893922985
10% critical value: -2.56883447171999

Stats for google
Dickey-Fuller statistic: -1.3567493925501937
P Value statistic: 0.6028265448690051
1% critical value: -3.4391698996357687
5% critical value: -2.8654325580580204
10% critical value: -2.568842816582842

Stats for amazon
Dickey-Fuller statistic: -1.1852777719873548
P Value statistic: 0.6798965643031646
1% critical value: -3.4390409569041207
5% critical value: -2.865375732701395
10% critical value: -2.568812543748081
```







The analysis is as follows:

Part 2 - Assessing Autocorrelations:

We see that the lag are all greater than 0. Hence, the autocorrelations between the the current value and past values (lags) are very high with large confidence. One more interesting observation is that as the lag indices go more into the past, the autocorrelation values tend to decrease. This is because the current value is more dependent on recent values of stock prices than on the past values. However, this is just a small decrease in the absolute value and overall the autocorrelations over all lags are high.

We notice that for all three companies the autocorrelation values tend to stagnate around 0.6 at a lag of 100. This is still a very high autocorrelation.

Part 3 - Assessing Stationarity

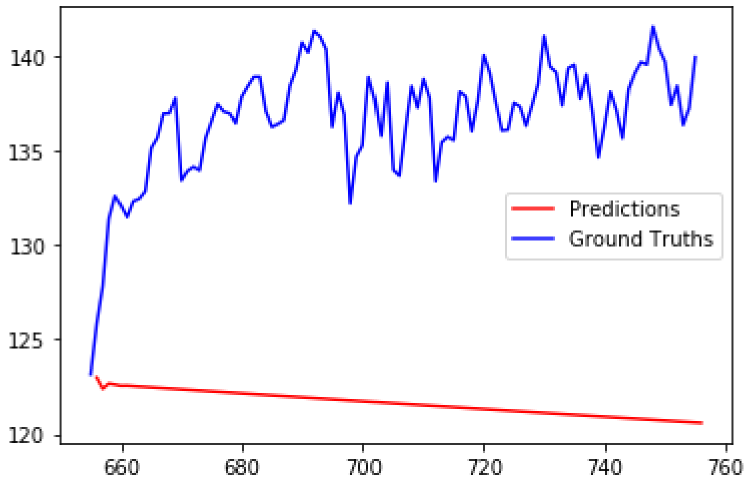
To assess stationarity, the Dickey-Fuller test is performed. The value returned by the DF test is the p-value of whether or not we can reject the null hypothesis that the time series is non stationary.

The analysis for the different companies is:

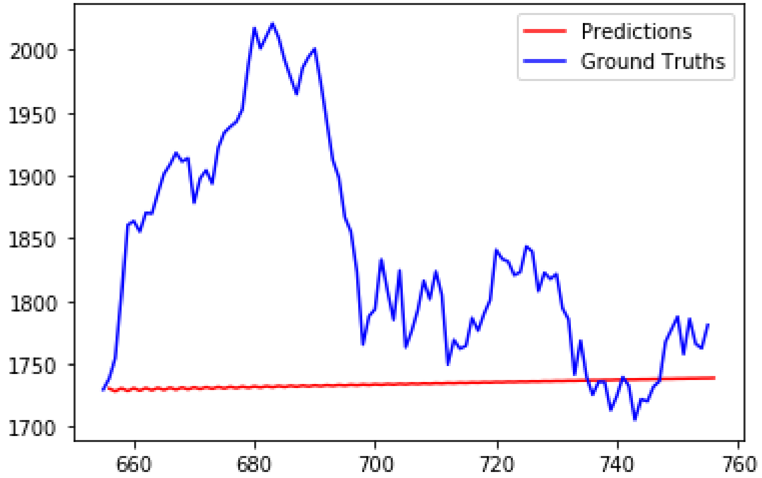
We clearly see that the P-Value for Microsoft is the highest and hence we cannot reject the null hypothesis. Therefore, we can say with a high probability that the Microsoft stock values follow a non stationary time series. We also see that the DF statistic is positive which implies that there is a high chance that the Microsoft stocks are non stationary.

For stocks of Google and Amazon, the p-value is relatively low but on an absolute scale it is still quite high. This implies that the probability of these stocks being a non stationary time series is also quite high but not as high as that of Microsoft.

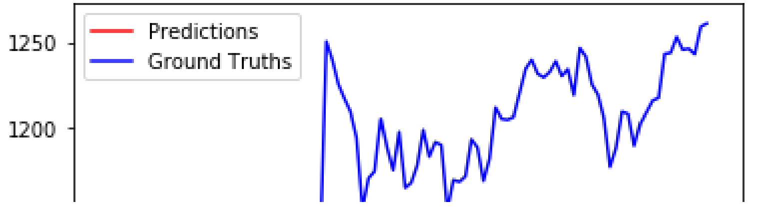
```
def fit_auto_regressive_moving_average(close_stocks, test_order, order, name):
  train_data=close_stocks[:-1*test_order]
  arma=ARMA(train_data, order=order).fit()

  train_size=len(train_data)
  test_size=len(list(close_stocks))-train_size

  predictions=arma.predict(train_size, train_size+test_size)
  ground_truths=close_stocks[-1*test_order-1:]

  plt.plot(predictions, color='red', label='Predictions')
  plt.plot(ground_truths, color='blue', label='Ground Truths')
  plt.legend(loc='best')
  plt.title(name)
  plt.show()

  print("MSE loss:", mean_squared_error(predictions, ground_truths))
  print("R2 score:", r2_score(predictions, ground_truths))

fit_auto_regressive_moving_average(microsoft_stocks_data, 100, (3, 1), 'Microsoft stats')
fit_auto_regressive_moving_average(amazon_stocks_data, 100, (3, 1), 'Amazon stats')
fit_auto_regressive_moving_average(google_stocks_data, 100, (3, 1), 'Google stats')
```

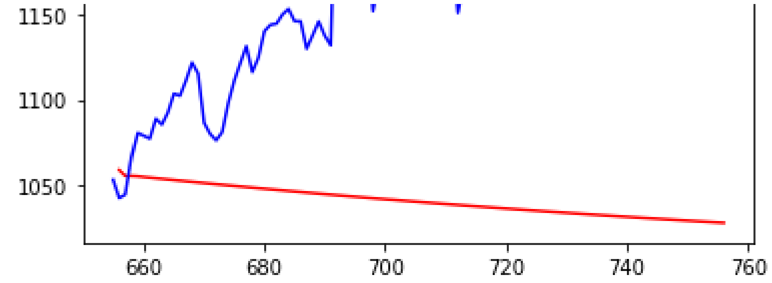



```
MSE loss: 239.17548253149621
R2 score: -644.6824815446543
/usr/local/lib/python3.6/dist-packages/statsmodels/base/model.py:492: HessianInversionWarning: Inverting hessian fa
  'available', HessianInversionWarning)
```




```
MSE loss: 18220.306623520697
R2 score: -2318.469232612349
```






```
MSE loss: 21310.35948977023
R2 score: -304.7679946452651
```

Part 3 - Predictability

For the different companies, the analysis is:

Microsoft:

We clearly see that the directional accuracy is very poor since the trends of the predictions and true values are clearly in opposite directions. However, the RMSE loss value is quite low due to the absolute values of the difference in the predictions. Hence, although loss is low, the predictability of these stocks is very poor.

Amazon:

Here, we see that the means of the predictions and the ground truths are somewhat similar but the absolute values are very high and hence the RMSE loss values are very high. Therefore, the predictability of these stocks is also very poor.

Google:

In the case of google, the directional accuracy is again very poor. We clearly see that this trend leads to high RMSE and hence low predictability.

Therefore we see that for all three companies the predictability is very low. One possible reason is that ARMA is usually used for stationary series but by the DF tests, we have seen that the time series we are working with have non stationary tendencies.