In [26]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import datetime
import statsmodels.api as sm
from pandas import Series as Series

path = 'AAPL_2006-01-01_to_2018-01-01.csv'

data = pd.read_csv(path)
data['Date'] = data['Date'].apply(lambda x : datetime.datetime.strptime(x, '%Y-%m-%d'))
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3019 entries, 0 to 3018
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype         
---  ------  --------------  -----         
 0   Date    3019 non-null   datetime64[ns]
 1   Open    3019 non-null   float64       
 2   High    3019 non-null   float64       
 3   Low     3019 non-null   float64       
 4   Close   3019 non-null   float64       
 5   Volume  3019 non-null   int64         
 6   Name    3019 non-null   object        
dtypes: datetime64[ns](1), float64(4), int64(1), object(1)
memory usage: 165.2+ KB
```

In [27]:

```
data.head(10)
```

Out[27]:

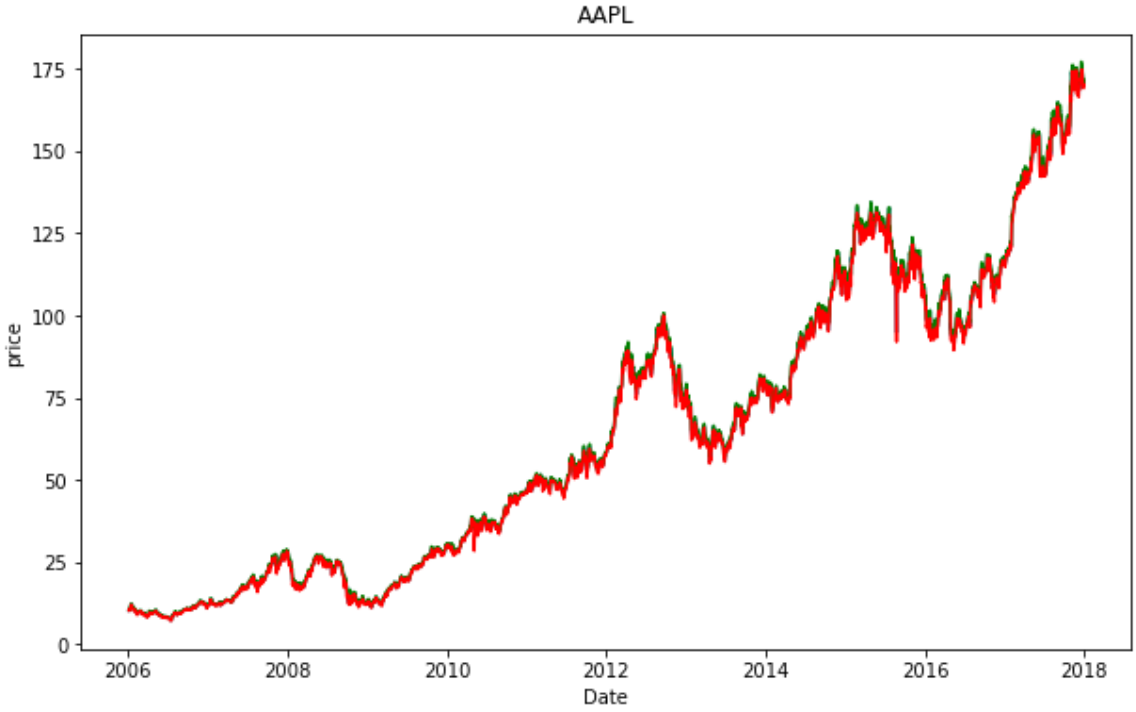
	Date	Open	High	Low	Close	Volume	Name
0	2006-01-03	10.34	10.68	10.32	10.68	201853036	AAPL
1	2006-01-04	10.73	10.85	10.64	10.71	155225609	AAPL
2	2006-01-05	10.69	10.70	10.54	10.63	112396081	AAPL
3	2006-01-06	10.75	10.96	10.65	10.90	176139334	AAPL
4	2006-01-09	10.96	11.03	10.82	10.86	168861224	AAPL
5	2006-01-10	10.89	11.70	10.83	11.55	570088246	AAPL
6	2006-01-11	11.98	12.11	11.80	11.99	373548882	AAPL
7	2006-01-12	12.14	12.34	11.95	12.04	320201966	AAPL
8	2006-01-13	12.14	12.29	12.09	12.23	194153393	AAPL
9	2006-01-17	12.24	12.34	11.98	12.10	209215265	AAPL

In [28]:

```
fig,ax = plt.subplots(figsize = (10,6))
ax.plot('Date','Open',data=data)
ax.plot('Date','High',color='green',data=data)
ax.plot('Date','Low',color='red',data=data)
ax.set_title('AAPL')
plt.xlabel('Date')
plt.ylabel('price')
```

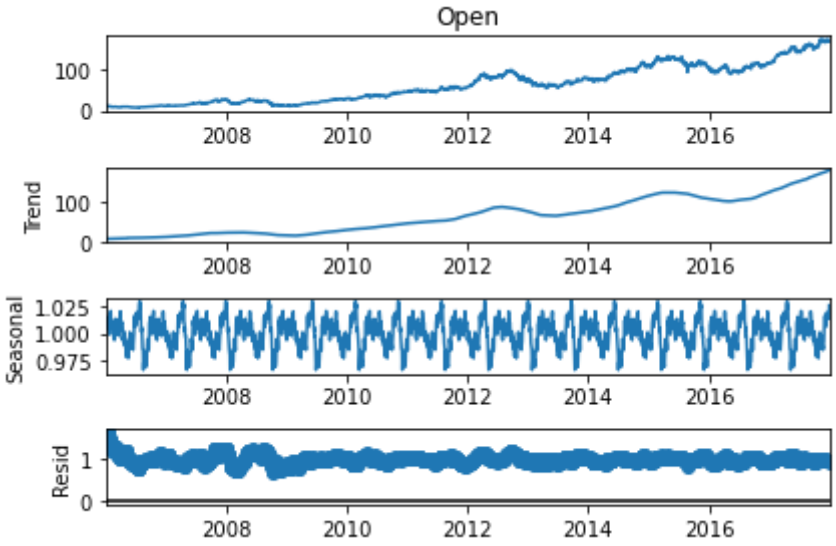
Out[28]:

```
Text(0, 0.5, 'price')
```



In [29]:

```
data1 = data[['Date','Open']].set_index('Date')
resul = sm.tsa.seasonal_decompose(data1['Open'] ,model="multiplicative",extrapolate_trend
fig=resul.plot()
```



平穩性檢定(ADF)

In [30]:

```
from statsmodels.tsa.stattools import adfuller, acf, pacf,arma_order_select_ic

data1 = data[['Date','Open']].set_index('Date')
adftes = adfuller(data1,autolag='AIC')
dfoutput = pd.Series(adftes[0:4], index=['Test Statistic','p-value','#Lags Used','Number
for key,value in adftes[4].items():
    dfoutput[f'Critical Value {key}'] = value

dfoutput
```

Out[30]:

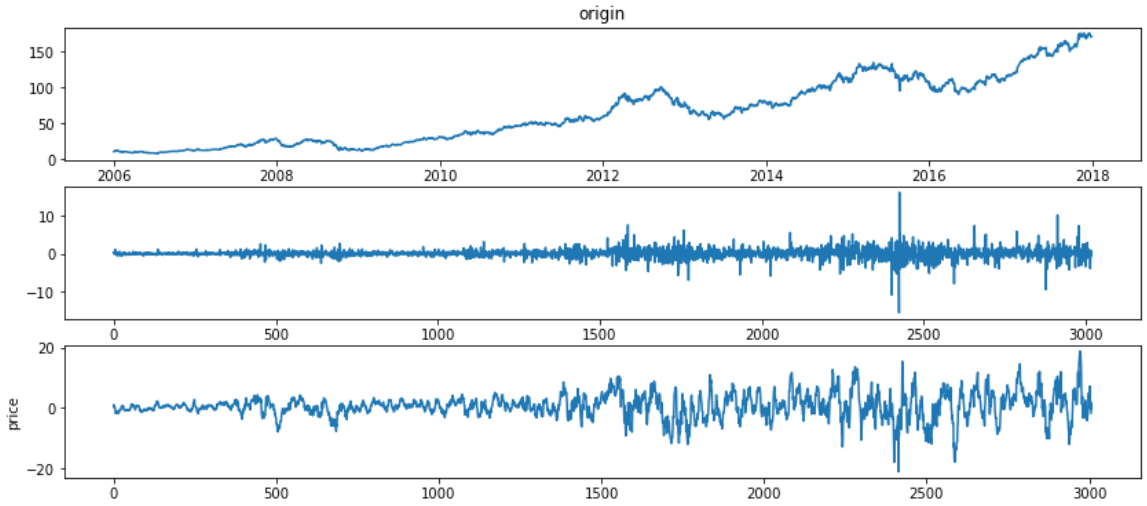
```
Test Statistic                   0.589688
p-value                          0.987359
#Lags Used                      22.000000
Number of Observations Used    2996.000000
Critical Value 1%                -3.432535
Critical Value 5%                -2.862505
Critical Value 10%               -2.567284
dtype: float64
```

In [31]:

```
def differ(dataset, interval=1):
    diff = []
    for i in range(interval, len(dataset)):
        diff.append(dataset.iloc[i,] - dataset.iloc[i - interval,])
    return Series(diff)

data1 = data[['Date','Open']].set_index('Date')
new_ts1=differ(data1)
new_ts12=differ(data1,12)
fig,ax = plt.subplots(3,1,figsize= (14,6))
ax1 = ax[0]
ax2 = ax[1]
ax3 = ax[2]
ax1.plot(data1.index,'Open',data=data1)
ax1.set_title('origin')
ax2.plot(new_ts1)
ax3.plot(new_ts12)
plt.ylabel('price')
```

Out[31]:

```
Text(0, 0.5, 'price')
```



In [32]:

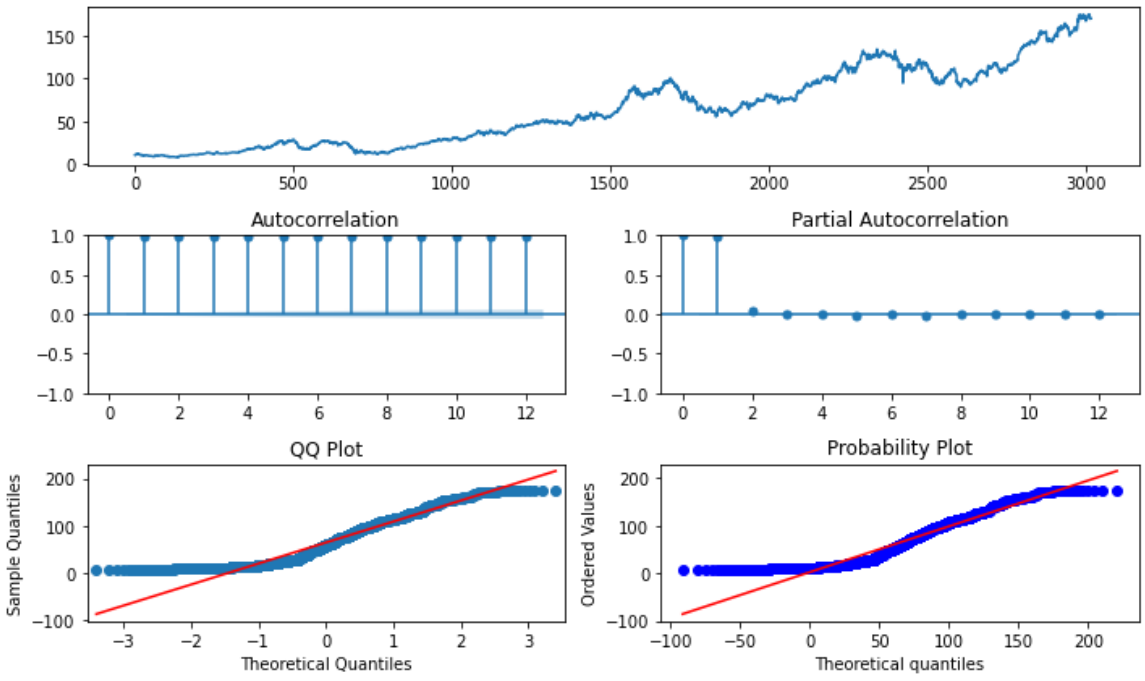
```
adftes = adfuller(new_ts1,autolag='AIC')
dfoutput = pd.Series(adftes[0:4], index=['Test Statistic','p-value','#Lags Used','Number
for key,value in adftes[4].items():
    dfoutput[f'Critical Value {key}'] = value
dfoutput
```

Out[32]:

```
Test Statistic                -1.109840e+01
p-value                        3.920843e-20
#Lags Used                     2.100000e+01
Number of Observations Used    2.996000e+03
Critical Value 1%             -3.432535e+00
Critical Value 5%             -2.862505e+00
Critical Value 10%            -2.567284e+00
dtype: float64
```

In [23]:

```
import statsmodels.tsa.api as smt
import statsmodels.api as sm
import scipy.stats as scs
from statsmodels.tsa.arima_model import ARIMA
import statsmodels.tsa.api as smt


def tsplot(y, lags=None,title=''):
    if not isinstance(y, pd.Series):
        return print('not Series')
    plt.style.context('ggplot')
    fig = plt.figure(figsize=(10, 6))
    #mpl.rcParams['font.family'] = 'Ubuntu Mono'
    layout = (3, 2)
    ts_ax = plt.subplot2grid(layout, (0, 0), colspan=2)
    acf_ax = plt.subplot2grid(layout, (1, 0))
    pacf_ax = plt.subplot2grid(layout, (1, 1))
    qq_ax = plt.subplot2grid(layout, (2, 0))
    pp_ax = plt.subplot2grid(layout, (2, 1))

    y.plot(ax=ts_ax)
    ts_ax.set_title(title)
    smt.graphics.plot_acf(y, lags=lags, ax=acf_ax, alpha=0.5)
    smt.graphics.plot_pacf(y, lags=lags, ax=pacf_ax, alpha=0.5,method='ywm')
    sm.qqplot(y, line='s', ax=qq_ax)
    qq_ax.set_title('QQ Plot')
    scs.probplot(y, sparams=(y.mean(), y.std()), plot=pp_ax)

    plt.tight_layout()

data1 = data[['Date','Open']].set_index('Date')
data1 =pd.Series(data.Open,index=data.index)

tsplot(data1, lags=12,title='')
```



In [9]:

```
data1 = data[['Date','Open']].set_index('Date')
data1 =pd.Series(data.Open,index=data.index)

best_aic = np.inf
best_order = None
best_mdl = None

for i in range(5):
    for j in range(5):
        for k in range(5):
            try:
                tmp_mdl = smt.ARIMA(data1, order=(i, j, k)).fit()
                tmp_aic = tmp_mdl.aic
                if tmp_aic < best_aic:
                    best_aic = tmp_aic
                    best_order = (i, j, k)
                    best_mdl = tmp_mdl
            except: continue
```

```
C:\Users\jerry\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sa
rimax.py:978: UserWarning: Non-invertible starting MA parameters found.
Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
C:\Users\jerry\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sa
rimax.py:978: UserWarning: Non-invertible starting MA parameters found.
Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
C:\Users\jerry\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sa
rimax.py:978: UserWarning: Non-invertible starting MA parameters found.
Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
C:\Users\jerry\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sa
rimax.py:978: UserWarning: Non-invertible starting MA parameters found.
Using zeros as starting parameters.
  warn('Non-invertible starting MA parameters found.'
C:\Users\jerry\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sa
rimax.py:978: UserWarning: Non-invertible starting MA parameters found.
Using zeros as starting parameters.
```

In [10]:

```
best_order
```

Out[10]:

```
(4, 1, 3)
```

In [11]:

```
best_aic
```

Out[11]:

```
10069.656547629183
```