In [6]:
```
import pandas as pd
from pmdarima import auto_arima
from statsmodels.tsa import seasonal
import statsmodels.api as st
import numpy as np
import matplotlib.pyplot as plt

path = './AdventureWorks_Database.xlsx'

data = pd.read_excel(path,sheet_name='Sales')
print(data.shape)
print(data.info())
data.head()
```

```
(58189, 13)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58189 entries, 0 to 58188
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ProductKey            58189 non-null  int64
 1   OrderDate             58189 non-null  datetime64[ns]
 2   ShipDate              58189 non-null  datetime64[ns]
 3   CustomerKey           58189 non-null  int64
 4   PromotionKey          58189 non-null  int64
 5   SalesTerritoryKey     58189 non-null  int64
 6   SalesOrderNumber      58189 non-null  object
 7   SalesOrderLineNumber  58189 non-null  int64
 8   OrderQuantity         58189 non-null  int64
 9   UnitPrice             58189 non-null  float64
 10  TotalProductCost      58189 non-null  float64
 11  SalesAmount           58189 non-null  float64
 12  TaxAmt                58189 non-null  float64
dtypes: datetime64[ns](2), float64(4), int64(6), object(1)
memory usage: 5.8+ MB
None
```

Out[6]:

	ProductKey	OrderDate	ShipDate	CustomerKey	PromotionKey	SalesTerritoryKey	SalesOrderNumb
0	310	2014-01-01	2014-01-08	21768	1	6	SO436
1	346	2014-01-01	2014-01-08	28389	1	7	SO436
2	346	2014-01-01	2014-01-08	25863	1	1	SO436
3	336	2014-01-01	2014-01-08	14501	1	4	SO437
4	346	2014-01-01	2014-01-08	11003	1	9	SO437

In [7]:
```
data['Year'] = pd.DatetimeIndex(data['OrderDate']).year
```

In [8]:
```
data['Month'] = pd.DatetimeIndex(data['OrderDate']).month
```

In [22]:
```
data.head()
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58189 entries, 0 to 58188
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype         
---  ------                --------------  -----         
 0   ProductKey            58189 non-null  int64         
 1   OrderDate             58189 non-null  datetime64[ns]
 2   ShipDate              58189 non-null  datetime64[ns]
 3   CustomerKey           58189 non-null  int64         
 4   PromotionKey          58189 non-null  int64         
 5   SalesTerritoryKey     58189 non-null  int64         
 6   SalesOrderNumber      58189 non-null  object        
 7   SalesOrderLineNumber  58189 non-null  int64         
 8   OrderQuantity         58189 non-null  int64         
 9   UnitPrice             58189 non-null  float64       
 10  TotalProductCost      58189 non-null  float64       
 11  SalesAmount           58189 non-null  float64       
 12  TaxAmt                58189 non-null  float64       
 13  Year                  58189 non-null  int64         
 14  Month                 58189 non-null  int64         
 15  new_date              58189 non-null  datetime64[ns]
dtypes: datetime64[ns](3), float64(4), int64(8), object(1)
memory usage: 7.1+ MB
```

In [23]:
```
newData = data.groupby(['Year','Month']).sum()
newData[['SalesAmount']]
newData.info()
```

```
<class 'pandas.core.frame.DataFrame'>
MultiIndex: 36 entries, (2014, 1) to (2016, 12)
Data columns (total 10 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   ProductKey            36 non-null     int64  
 1   CustomerKey           36 non-null     int64  
 2   PromotionKey          36 non-null     int64  
 3   SalesTerritoryKey     36 non-null     int64  
 4   SalesOrderLineNumber  36 non-null     int64  
 5   OrderQuantity         36 non-null     int64  
 6   UnitPrice             36 non-null     float64
 7   TotalProductCost      36 non-null     float64
 8   SalesAmount           36 non-null     float64
 9   TaxAmt                36 non-null     float64
dtypes: float64(4), int64(6)
memory usage: 3.1 KB
```

In [25]:
```
data['OrderDate'] = pd.to_datetime(data['OrderDate'])
```

In [19]:
```
data['new_date'] = pd.to_datetime(dict(year=data['OrderDate'].dt.year,month=data['Orde
```

In [20]:
```
data
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 58189 entries, 0 to 58188
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ProductKey            58189 non-null  int64
 1   OrderDate             58189 non-null  datetime64[ns]
 2   ShipDate              58189 non-null  datetime64[ns]
 3   CustomerKey           58189 non-null  int64
 4   PromotionKey          58189 non-null  int64
 5   SalesTerritoryKey     58189 non-null  int64
 6   SalesOrderNumber      58189 non-null  object
 7   SalesOrderLineNumber  58189 non-null  int64
 8   OrderQuantity         58189 non-null  int64
 9   UnitPrice             58189 non-null  float64
 10  TotalProductCost      58189 non-null  float64
 11  SalesAmount           58189 non-null  float64
 12  TaxAmt                58189 non-null  float64
 13  Year                  58189 non-null  int64
 14  Month                 58189 non-null  int64
 15  new_date              58189 non-null  datetime64[ns]
dtypes: datetime64[ns](3), float64(4), int64(8), object(1)
memory usage: 7.1+ MB
```

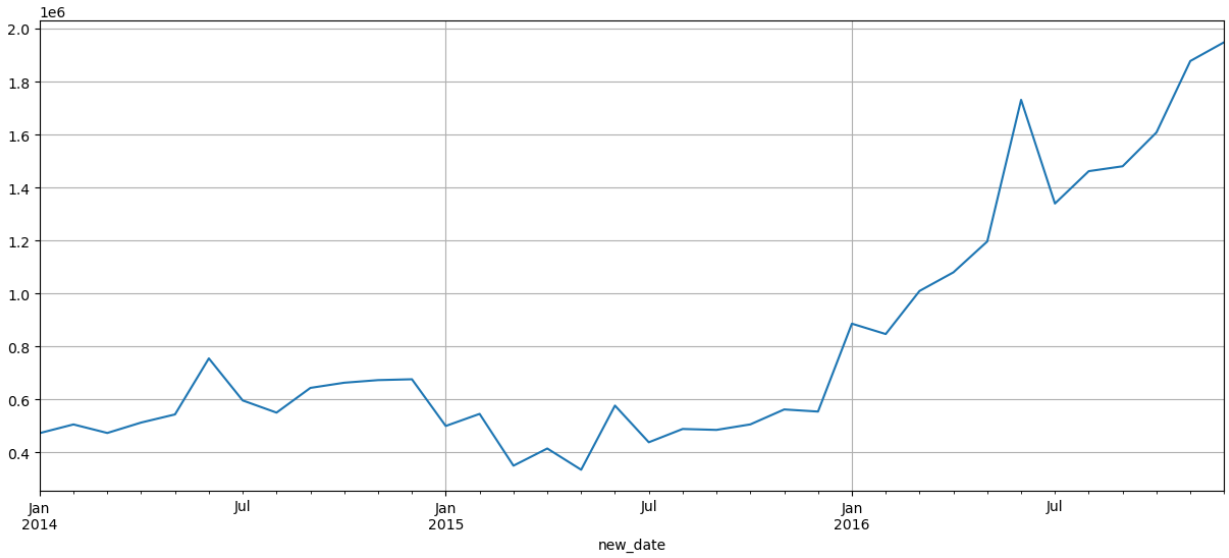
In [21]:
```
newData = data.groupby('new_date').sum()
newData.head()
```

Out[21]:

new_date	ProductKey	CustomerKey	PromotionKey	SalesTerritoryKey	SalesOrderLineNumber	OrderQ
2014-01-12	46934	2700045	146	1008	146	
2014-02-12	50121	2959979	156	1112	156	
2014-03-12	46300	2899467	146	999	146	
2014-04-12	51505	3220148	161	1091	161	
2014-05-12	54281	3539694	169	1088	169	

In [17]:
```
newData['SalesAmount'].plot(figsize=(15,6),grid='on')
```

Out[17]:
```
<AxesSubplot:xlabel='new_date'>
```

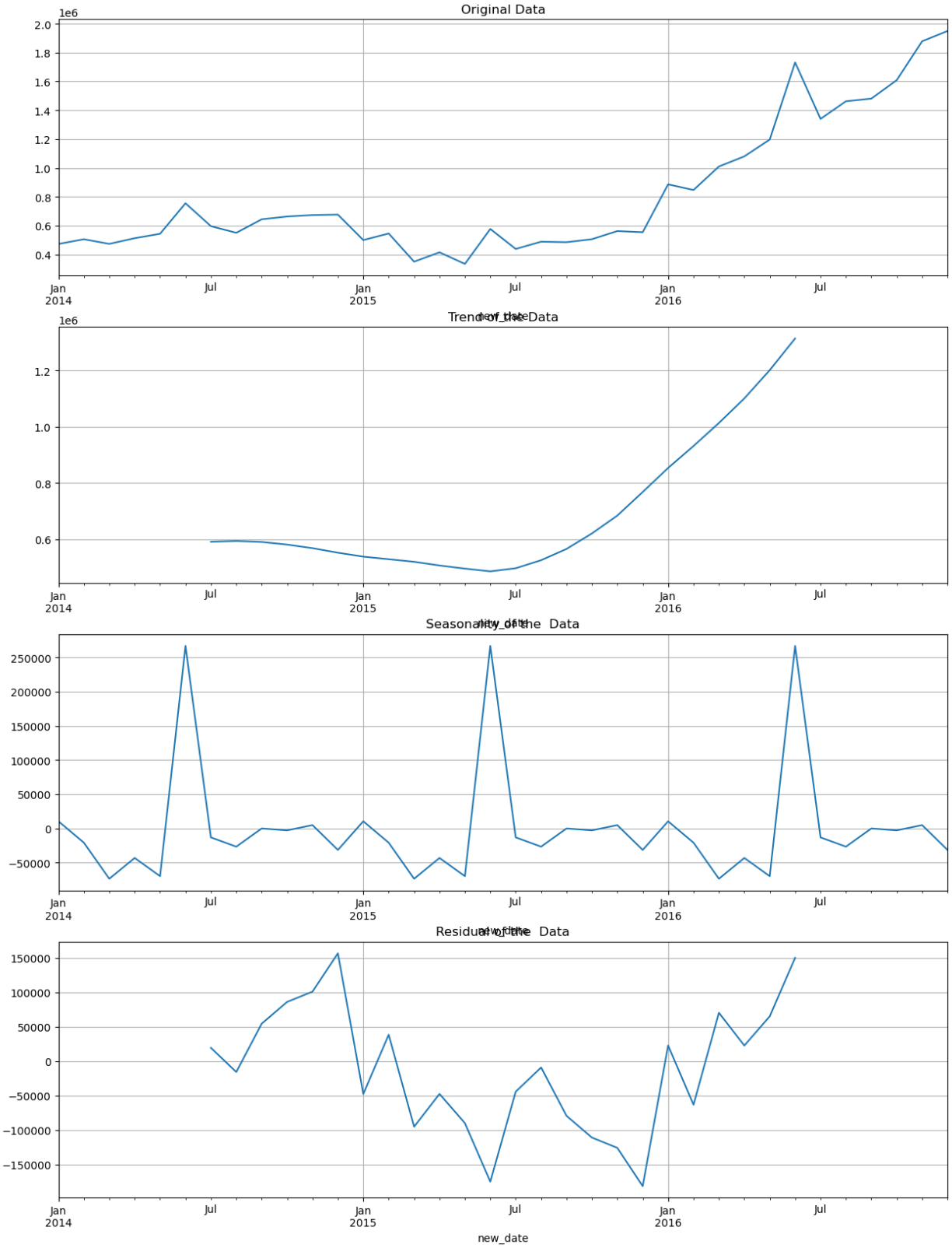


In [22]:
```
X = newData['SalesAmount']
```

In [23]:
```
var = seasonal.seasonal_decompose(X)
plt.figure(1,(15,20))
plt.subplot(4,1,1)
var.observed.plot(title='Original Data',grid='on')

plt.subplot(4,1,2)
var.trend.plot(title='Trend of the Data',grid='on')


plt.subplot(4,1,3)
var.seasonal.plot(title='Seasonality of the  Data',grid='on')


plt.subplot(4,1,4)
var.resid.plot(title='Residual of the  Data',grid='on')
from pmdarima import auto_arima

temp = X.copy()

auto_arima(temp,seasonal=False,trace=True)
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(0,0,0)[0] intercept   : AIC=946.226, Time=0.32 sec
 ARIMA(0,1,0)(0,0,0)[0] intercept   : AIC=942.097, Time=0.03 sec
 ARIMA(1,1,0)(0,0,0)[0] intercept   : AIC=940.140, Time=0.04 sec
 ARIMA(0,1,1)(0,0,0)[0] intercept   : AIC=941.564, Time=0.05 sec
 ARIMA(0,1,0)(0,0,0)[0]             : AIC=942.464, Time=0.03 sec
 ARIMA(2,1,0)(0,0,0)[0] intercept   : AIC=942.055, Time=0.04 sec
 ARIMA(1,1,1)(0,0,0)[0] intercept   : AIC=942.220, Time=0.06 sec
 ARIMA(2,1,1)(0,0,0)[0] intercept   : AIC=944.134, Time=0.07 sec
 ARIMA(1,1,0)(0,0,0)[0]             : AIC=942.586, Time=0.03 sec

Best model:  ARIMA(1,1,0)(0,0,0)[0] intercept
Total fit time: 0.719 seconds
```

Out[23]:
```
ARIMA(order=(1, 1, 0), scoring_args={}, suppress_warnings=True)
```



In [24]:

```
from statsmodels.tsa import arima
temp = X.copy()
samples = 12
for i in range(samples):
    Arima_Model = arima.api.ARIMA(temp,order=(2,1,1))

    Arima_Model = Arima_Model.fit()

    op = Arima_Model.forecast()
```

```
    print(op)

    temp = pd.concat([temp,op])


plt.figure(1)
temp.plot(kind='line')
X.plot(kind='line')
```

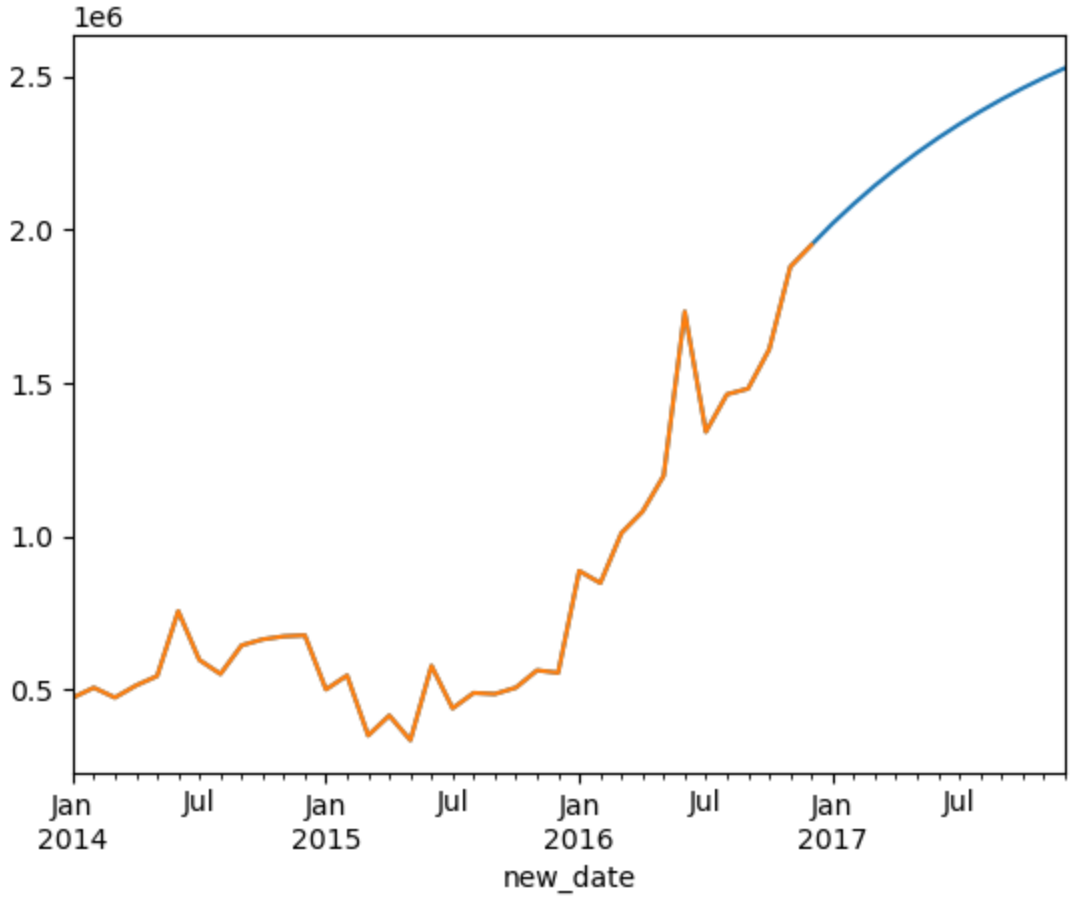
```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarimax.py:966: UserWarning: Non-stationary starting autoregressive parameters found. Using zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
2017-01-01    2.019267e+06
Freq: MS, dtype: float64
2017-02-01    2.082887e+06
Freq: MS, dtype: float64
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
2017-03-01    2.142619e+06
Freq: MS, dtype: float64
2017-04-01    2.198046e+06
Freq: MS, dtype: float64
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
2017-05-01    2.249743e+06
Freq: MS, dtype: float64
2017-06-01    2.297892e+06
Freq: MS, dtype: float64
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
2017-07-01    2.342779e+06
Freq: MS, dtype: float64
2017-08-01    2.384615e+06
Freq: MS, dtype: float64
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
2017-09-01    2.423642e+06
Freq: MS, dtype: float64
2017-10-01    2.460041e+06
Freq: MS, dtype: float64
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
2017-11-01    2.493999e+06
Freq: MS, dtype: float64
2017-12-01    2.525681e+06
Freq: MS, dtype: float64
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
```

Out[24]: <AxesSubplot:xlabel='new_date'>



In [25]:
```
# WITH SEASONAL PARAMETER
temp = X.copy()

auto_arima(temp,seasonal=True,trace=True,m=12)
```

```
Performing stepwise search to minimize aic
 ARIMA(2,1,2)(1,1,1)[12]             : AIC=622.988, Time=0.32 sec
 ARIMA(0,1,0)(0,1,0)[12]             : AIC=625.593, Time=0.03 sec
 ARIMA(1,1,0)(1,1,0)[12]             : AIC=617.132, Time=0.11 sec
 ARIMA(0,1,1)(0,1,1)[12]             : AIC=622.109, Time=0.13 sec
 ARIMA(1,1,0)(0,1,0)[12]             : AIC=624.653, Time=0.05 sec
 ARIMA(1,1,0)(2,1,0)[12]             : AIC=618.733, Time=0.39 sec
 ARIMA(1,1,0)(1,1,1)[12]             : AIC=618.755, Time=0.19 sec
 ARIMA(1,1,0)(0,1,1)[12]             : AIC=621.285, Time=0.13 sec
 ARIMA(1,1,0)(2,1,1)[12]             : AIC=620.733, Time=0.41 sec
 ARIMA(0,1,0)(1,1,0)[12]             : AIC=615.490, Time=0.07 sec
 ARIMA(0,1,0)(2,1,0)[12]             : AIC=616.134, Time=0.20 sec
 ARIMA(0,1,0)(1,1,1)[12]             : AIC=615.838, Time=0.15 sec
 ARIMA(0,1,0)(0,1,1)[12]             : AIC=618.968, Time=0.10 sec
 ARIMA(0,1,0)(2,1,1)[12]             : AIC=617.779, Time=0.40 sec
 ARIMA(0,1,1)(1,1,0)[12]             : AIC=618.382, Time=0.12 sec
 ARIMA(1,1,1)(1,1,0)[12]             : AIC=619.447, Time=0.25 sec
 ARIMA(0,1,0)(1,1,0)[12] intercept   : AIC=615.617, Time=0.17 sec

Best model:  ARIMA(0,1,0)(1,1,0)[12]
Total fit time: 3.344 seconds
```

Out[25]:
```
ARIMA(order=(0, 1, 0), scoring_args={}, seasonal_order=(1, 1, 0, 12),
      suppress_warnings=True, with_intercept=False)
```

In [26]:
```
temp = X.copy()
samples = 12
for i in range(samples):
    sArima_Model = st.tsa.statespace.SARIMAX(temp,order=(0,1,0),seasonal_order=(1,1,0,

    sArima_Model = sArima_Model.fit()

    op = sArima_Model.forecast()
    print(op)

    temp = pd.concat([temp,op])


plt.figure(1)
temp.plot(kind='line')
X.plot(kind='line')
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\statespace\sarimax.py:997:
UserWarning: Non-stationary starting seasonal autoregressive Using zeros as starting
parameters.
  warn('Non-stationary starting seasonal autoregressive'
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
```

```
2017-01-01    1.924440e+06
Freq: MS, dtype: float64
2017-02-01    1.944777e+06
Freq: MS, dtype: float64
2017-03-01    1.856046e+06
Freq: MS, dtype: float64
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
```

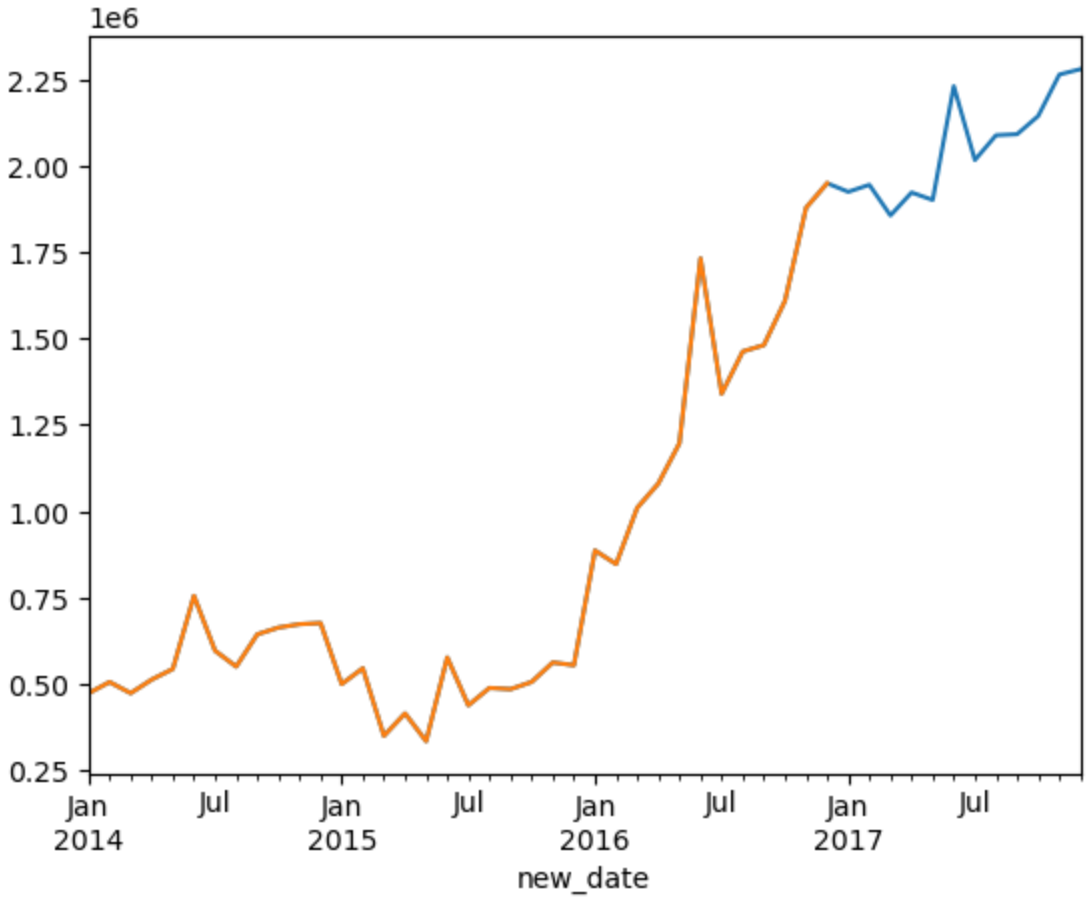
```
2017-04-01    1.922539e+06
Freq: MS, dtype: float64
2017-05-01    1.900903e+06
Freq: MS, dtype: float64
2017-06-01    2.230320e+06
Freq: MS, dtype: float64
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
```

```
2017-07-01    2.016446e+06
Freq: MS, dtype: float64
2017-08-01    2.088131e+06
Freq: MS, dtype: float64
2017-09-01    2.091154e+06
Freq: MS, dtype: float64
```

```
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
C:\Users\ameen\anaconda3\lib\site-packages\statsmodels\tsa\base\tsa_model.py:471: Val
ueWarning: No frequency information was provided, so inferred frequency MS will be us
ed.
  self._init_dates(dates, freq)
2017-10-01    2.143873e+06
Freq: MS, dtype: float64
2017-11-01    2.263780e+06
Freq: MS, dtype: float64
2017-12-01    2.279355e+06
Freq: MS, dtype: float64
```

```
Out[26]: <AxesSubplot:xlabel='new_date'>
```


1e6
2.25
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
Jan
2014
Jul
Jan
2015
Jul
Jan
2016
Jul
Jan
2017
Jul
new_date

In [ ]: