In [11]:

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

data = pd.read_csv('SP500_weekly.csv')
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 261 entries, 0 to 260
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    261 non-null    object 
 1   Open    261 non-null    float64
 2   High    261 non-null    float64
 3   Low     261 non-null    float64
 4   Close   261 non-null    float64
 5   Volume  261 non-null    int64  
dtypes: float64(4), int64(1), object(1)
memory usage: 12.4+ KB
```

In [27]:

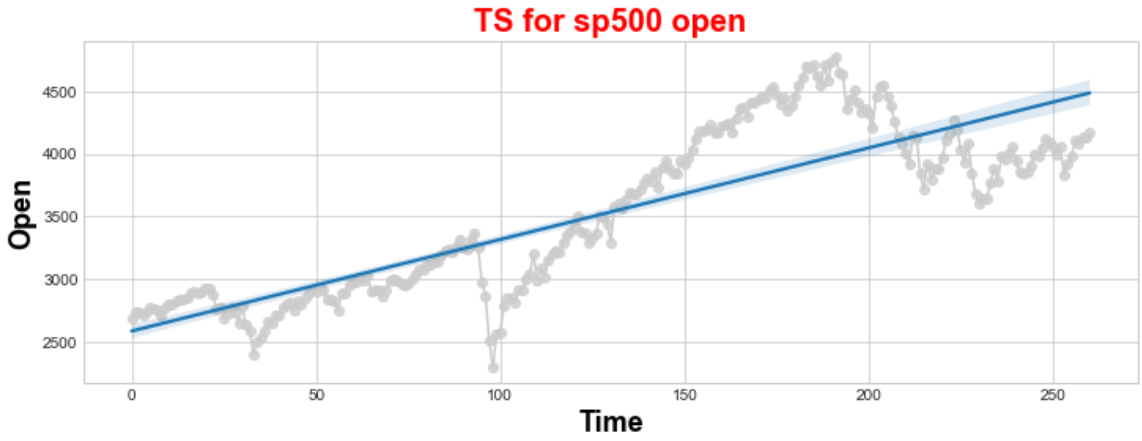
```
data1 = data[['Open',"Date"]]
data1 = data1.set_index('Date')
data1['Time'] = np.arange(len(data1))

plt.style.use('seaborn-whitegrid')  # 繪圖模板
plt.rc('axes',labelweight='bold',labelsize=18,labelcolor='black',titleweight='bold',title
plt.rc('figure',figsize=(12,4),titlesize=25,titleweight='bold')

fig,ax=plt.subplots()
ax.plot('Time','Open',data=data1,color='0.8')
ax = sns.regplot(x='Time',y='Open',data=data1,scatter_kws={'color':'0.8'})
ax.set_title('TS for sp500 open')
```

Out[27]:

```
Text(0.5, 1.0, 'TS for sp500 open')
```



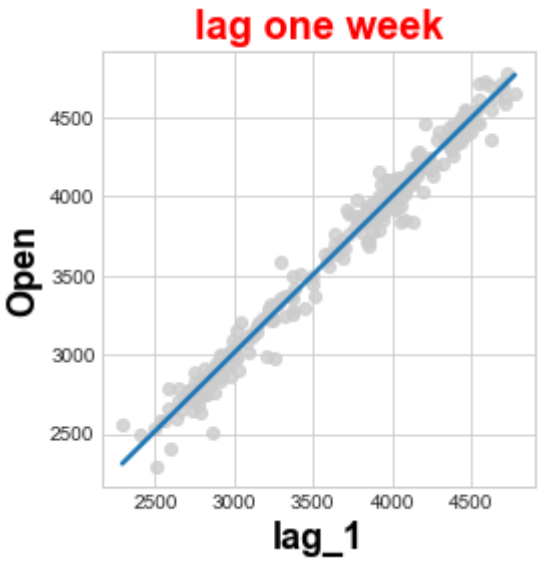
lag one week

In [28]:

```
data1['lag_1'] = data1['Open'].shift(1)
fig,ax=plt.subplots()
ax.set_aspect('equal')  # 設置x,y軸相等
ax = sns.regplot(x='lag_1',y='Open',data=data1,scatter_kws={'color':'0.8'})
ax.set_title('lag one week')
```

Out[28]:

```
Text(0.5, 1.0, 'lag one week')
```



開盤價線性回歸

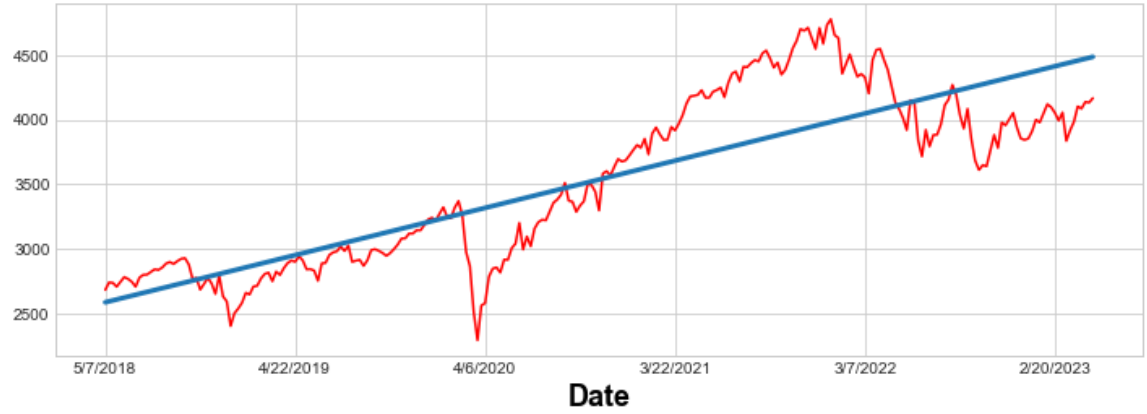
In [29]:

```
from sklearn.linear_model import LinearRegression

plot_params = dict(color='red',style='-',legend=False)

x = data1.loc[:,['Time']]    # dataframe
y = data1.loc[:,'Open']      # series
model = LinearRegression()
model.fit(x,y)
y_pre = pd.Series(model.predict(x),index=x.index)

ax = y.plot(**plot_params)
ax = y_pre.plot(ax=ax,linewidth=3)
```



過去一個月平均趨勢

In [30]:

```
MAV = data1['Open'].rolling(4,center=True).mean()
ax = data1['Open'].plot(style='--',color='0.8')
MAV.plot(ax = ax,linewidth=2,title='last one month moving average')
```

Out[30]:

```
<AxesSubplot:title={'center':'last one month moving average'}, xlabel='Dat
e'>
```

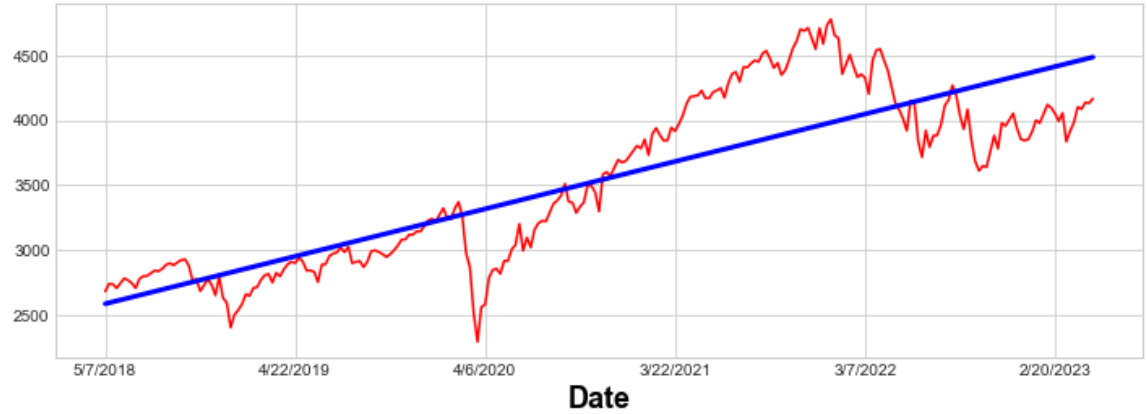
In [32]:

```
from statsmodels.tsa.deterministic import DeterministicProcess
from sklearn.linear_model import LinearRegression

dp = DeterministicProcess(index=data1.index,constant=True,order=1,drop=True)
x1 = dp.in_sample()
y = data1.loc[:,'Open']

model = LinearRegression()
model.fit(x1,y)
y_pre = pd.Series(model.predict(x1),index=x1.index)

plot_params = dict(color='red',style='-',legend=False)
ax = y.plot(**plot_params)
ax = y_pre.plot(ax=ax,linewidth=3,color='blue', label="Trend")
```



季節性(一年48週)

In [33]:

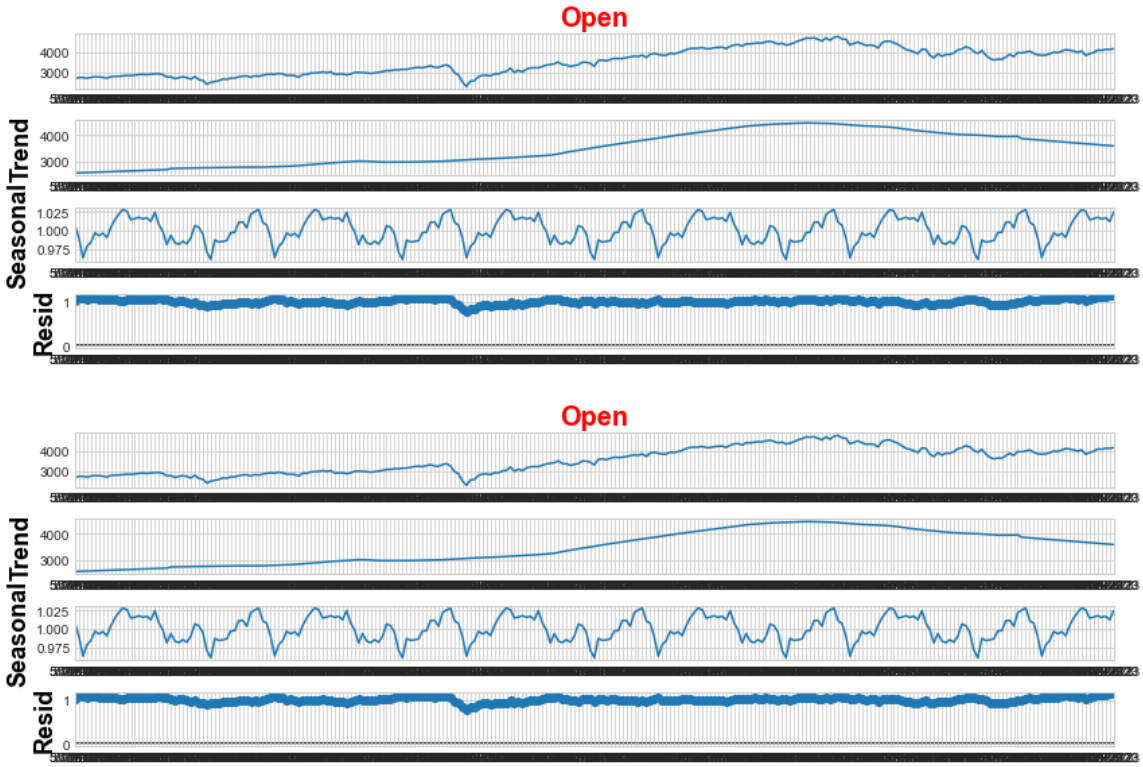
```
from statsmodels.tsa.seasonal import seasonal_decompose
```

乘法

In [34]:

```
resul_mul =seasonal_decompose(data1['Open'],model='multiplicative',extrapolate_trend='fre
resul_mul.plot()
```

Out[34]:



加法

In [35]:

```
resul_mul =seasonal_decompose(data1['Open'],model='additive',extrapolate_trend='freq',per
resul_mul.plot()
```

Out[35]: