```
import FinanceDataReader as fdr

# 한국거래소 상장종목 전체
df_krx = fdr.StockListing('KRX')
```

```
df_krx.columns
```

```
Index(['Symbol', 'Name', 'Sector', 'Industry'], dtype='object')
```

```
name=input('주식 코드 기업명 ?') #find 종목코드 및 정보
df_krx[df_krx['Name'].str.contains(name)]

import pandas as pd
import datetime
today=datetime.datetime.today()
# KS11 (KOSPI 지수), 2018년~현재
kospi = fdr.DataReader('KS11', '2018')
# 다우지수, 2018년~현재
dow = fdr.DataReader('DJI', '2018')
# 원달러 환율, 2018년~현재
usd_won = fdr.DataReader('USD/KRW', '2018')
# 원중국위안 환율, 2018년~현재
cny_won = fdr.DataReader('CNY/KRW', '2018')
# 비트코인 원화 가격 (빗썸), 2018년~현재
btc_won = fdr.DataReader('BTC/KRW', '2016')
# 비트코인 원화 가격 (빗썸), 2018년~현재
btc_usd = fdr.DataReader('BTC/USD', '2016')
#삼성전자 주가 & 애플 주가 (2018.01.01~현재)
ss=fdr.DataReader('005930', '2018-01-01', today)
apple=fdr.DataReader('AAPL', '2018-01-01', today)

kospi.head(3)

import matplotlib.pyplot as plt
plt.rcParams["font.family"] = 'AppleGothic'
plt.rcParams["figure.figsize"] = (14,8)

plt.subplots_adjust(hspace = 0.5, wspace = 0.3)
plt.subplot(2,1,1)
ax=kospi['Close'].plot()
plt.title('종합주가지수KOSPI . 2018.1.1-today')
plt.subplot(2,1,2)
ax=dow['Close'].plot()
plt.title('미국 Dow 지수 . 2018.1.1-today')
plt.show()

plt.subplots_adjust(hspace = 0.5, wspace = 0.3)

plt.subplot(2,1,1)
ax=usd_won['Close'].plot()
plt.title('원당 달러 환율 . 2018.1.1-today')

plt.subplot(2,1,2)
```

```
ax=cny_won['Close'].plot()
plt.title('원당 위안 환율 . 2018.1.1-today')

plt.show()

plt.subplots_adjust(hspace = 0.5, wspace = 0.3)

plt.subplot(2,1,1)
ax=btc_won['Close'].plot()
plt.title('원당 비트코인 가격 . 2018.1.1-today')

plt.subplot(2,1,2)
ax=btc_usd['Close'].plot()
plt.title('달러당 비트코인 가격 . 2018.1.1-today')
plt.show()

plt.subplots_adjust(hspace = 0.5, wspace = 0.3)

plt.subplot(2,1,1)
ax=ss['Close'].plot()
plt.title('삼성전자. 2018.1.1-today')

plt.subplot(2,1,2)
ax=apple['Close'].plot()
plt.title('애플. 2018.1.1-today')
plt.show()

fin=pd.concat([kospi['Close'],ss['Close'],apple['Close']], axis=1)
fin.head(3)

fin.set_axis(['코스피','삼성','애플'],axis=1,inplace=True)
fin.head(3)

fin_L=fin.shift(1); fin_L.head(3)

fin0=pd.concat([fin,fin_L],axis=1)
fin0.head(3)
```

주식 코드 기업명 ?삼성

	Symbol	Name	Sector	
32	006400	삼성SDI	일차전지 및 축전지 제조업	칼라브라운관,PDP,평판표시
33	207940	삼성바이오로직스	기초 의약물질 및 생물학적 제제 제조업	
34	068290	삼성출판사	서적, 잡지 및 기타 인쇄물 출판업	
35	029780	삼성카드	기타 금융업	신용카드업,상품신
36	000810	삼성화재해상보험	보험업	
624	018260	삼성에스디에스	컴퓨터 프로그래밍, 시스템 통합 및 관리업	
625	010140	삼성중공업	선박 및 보트 건조업	선박(벌크·
626	016360	삼성증권	금융 지원 서비스업	
911	028260	삼성물산	기타 전문 도매업	
912	005930	삼성전자	통신 및 방송 장비 제조업	IMT2000 서비스용 동기식
913	001360	삼성제약	의약품 제조업	의약품(
1166	006660	삼성공조	자동차 신품 부품 제조업	
1167	032830	삼성생명	보험업	
1168	009150	삼성전기	전자부품 제조업	영상,음향,통신장비,모듈,다층인
1445	028050	삼성엔지니어링	건축기술, 엔지니어링 및 관련 기술 서비스업	
1557	309930	삼성머스트스팩3호	금융 지원 서비스업	
1849	291230	삼성스팩2호	금융 지원 서비스업	

```
fin0.reset_index(inplace=True)
fin0.columns=['Date','코스피','삼성','애플','코스피시차','삼성시차','애플시차']
fin0['연도']=fin0['Date'].dt.year
fin0['월']=fin0['Date'].dt.month
fin_corr=fin0.groupby(['연도','월']).corr()
fin_corr.reset_index(inplace=True)

fin_corr[fin_corr['연도']==2018].head(2)
fin_corr[(fin_corr['연도']==2018)&(fin_corr['level_2']=='삼성')].head(2)
fin_corr.set_index(['연도','월','level_2'],inplace=True)

fin_corr[abs(fin_corr['코스피'])>0.7]['코스피']

import FinanceDataReader as fdr
import pandas as pd
import datetime
today=datetime.datetime.today()
# KS11 (KOSPI 지수), 2019년~현재
kospi = fdr.DataReader('KS11', '2019')
#삼성전자 주가 (2019.01.01~현재)
samsung=fdr.DataReader('005930', '2019-01-01', today)
hynix=fdr.DataReader('000660', '2019-01-01')
today

```

```
24
25 df_stock=pd.concat([kospi['Close'],samsung['Close'],hynix['Close']],axis=1)
26 df_stock.set_axis(['코스피','삼성전자','하이닉스'],axis=1,inplace=True)
```

```
FrozenList(['Date'])
```

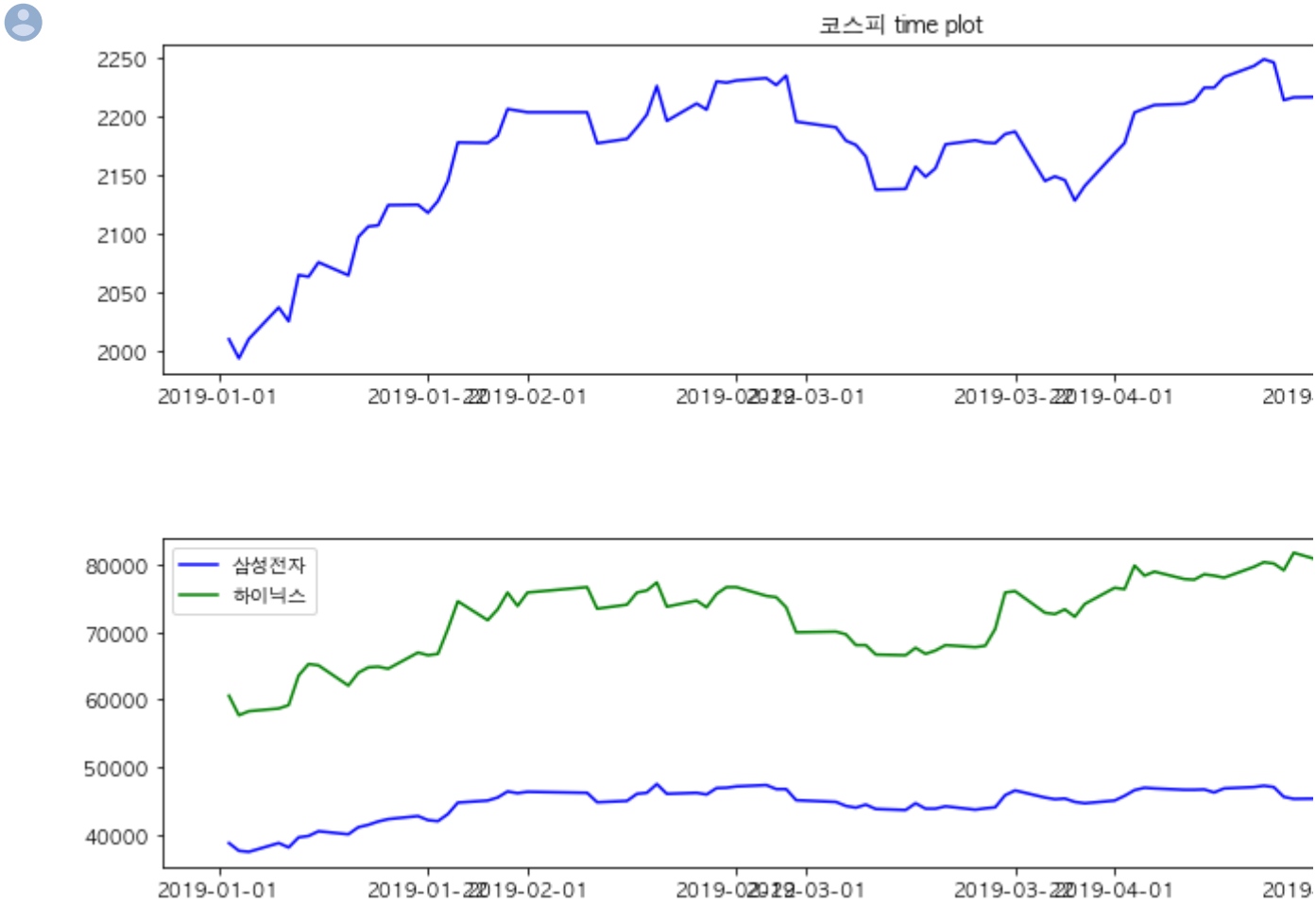
```
1 df_stock.head(3)
```

Date	코스피	삼성전자	하이닉스
2019-01-02	2010.00	38750	60600
2019-01-03	1993.70	37600	57700
2019-01-04	2010.25	37450	58300

```
1 import matplotlib.pylab as plt
2 plt.subplots_adjust(hspace = 0.5, wspace = 0.3)
3 plt.subplot(2,1,1)
4 plt.plot(df_stock['코스피'], color='blue',label='코스피')
5 plt.title('코스피 time plot')
6 plt.subplot(2,1,2)
7 plt.plot(df_stock['삼성전자'], color='blue',label='삼성전자')
8 plt.plot(df_stock['하이닉스'], color='green',label='하이닉스')
9 plt.legend(loc='best')
10 plt.show()
11
12 #Perform Dickey-Fuller test
13 from statsmodels.tsa.stattools import adfuller, kpss
14 adfuller(df_stock['코스피'], autolag='AIC')
15
16 from statsmodels.tsa.stattools import kpss
17 kpss(df_stock['삼성전자'], regression='c')
18
19 adfuller(df_stock['삼성전자'], autolag='AIC')[0:2]
20
21 adfuller(df_stock['하이닉스'], autolag='AIC')[0:2]
22
23 #삼성전자 주가 (2019.01.01~현재)
24 samsung=fdr.DataReader('005930', '2018-01-01', today)
25 ss_m5=samsung['Close'].rolling(window=5).mean()
26 ss_m20=samsung['Close'].rolling(window=20).mean()
27 ss_m60=samsung['Close'].rolling(window=60).mean()
28
29 plt.title('삼성전자 2018년~현재 이동평균법')
30 plt.plot(samsung['Close'], color='black',label='종가')
31 plt.plot(ss_m5, color='blue',ls='--',label='5일')
32 plt.plot(ss_m20, color='green',label='20일')
33 plt.plot(ss_m60, color='red',label='60일')
34 plt.legend(loc='best')
35 plt.show()
36
37 # Simple Exponential Smoothing
38 import FinanceDataReader as fdr
```

```
import pandas as pd
import datetime
today=datetime.datetime.today()
samsung=fdr.DataReader('005930', '2018-01-01', today)
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
fit1 = SimpleExpSmoothing(samsung['Close'],h=5).fit(smoothing_level=0.1,optimize
fit2 = SimpleExpSmoothing(samsung['Close'],h=5).fit() #recommend

print('alpha=0.1 *** \n', fit1.forecast(3),'\n optimal alpha= ***\n', fit2.forec

import matplotlib.pyplot as plt
plt.rcParams["font.family"] = 'AppleGothic'
plt.rcParams["figure.figsize"] = (14,8)
plt.title('삼성전자 2018년~현재 단순지수평활법')
plt.plot(samsung['Close'], color='black',label='종가')
plt.plot(fit1.fittedvalues, color='blue',ls='--',label='alpha=0.1')
plt.plot(fit2.fittedvalues, color='red',ls='--',label='optimal')
plt.legend(loc='best')
plt.show()

fit3=Holt(samsung['Close']).fit(smoothing_level=0.8, smoothing_slope=0.2,optimiz
fit4=Holt(samsung['Close'],exponential=True).fit(smoothing_level=0.8, smoothing_
fit5=Holt(samsung['Close'],damped=True).fit(smoothing_level=0.8, smoothing_slope

plt.title('삼성전자 2018년~현재 Holts 지수평활법')
plt.plot(samsung['Close'], color='black',label='종가')
plt.plot(fit3.fittedvalues, color='blue',ls='--',label='alpha=0.8, beta=0.2')
plt.plot(fit4.fittedvalues, color='red',ls='--',label='exponential')
plt.plot(fit5.fittedvalues, color='green',ls='--',label='damped')
plt.legend(loc='best')
plt.show()

plt.title('삼성전자 2018년~현재 Holts지수평활법 예측 향후 20기')
plt.plot(fit3.forecast(20), color='blue',ls='--',label='alpha=0.8, beta=0.2')
plt.plot(fit4.forecast(20), color='red',ls='--',label='exponential')
plt.plot(fit5.forecast(20), color='red',ls='--',label='damped')
plt.legend(loc='best')
plt.show()

fit4.forecast(3)

from statsmodels.tsa.api import ExponentialSmoothing
fit1 = ExponentialSmoothing(samsung['Close'], seasonal_periods=5, trend='add', s
fit2 = ExponentialSmoothing(samsung['Close'], seasonal_periods=5, trend='add', s
fit3 = ExponentialSmoothing(samsung['Close'], seasonal_periods=5, trend='add', s
fit4 = ExponentialSmoothing(samsung['Close'], seasonal_periods=5, trend='add', s

samsung['Close'].plot(style='-', color='black',label='삼성전자_종가')
fit1.fittedvalues.plot(style='--', color='red',label='가법경향,가법계절')
fit2.fittedvalues.plot(style='--', color='green',label='가법경향,승법계절')
fit3.fittedvalues.plot(style='--', color='blue',label='Damped가법경향,가법계절')
fit4.fittedvalues.plot(style='--', color='yellow',label='Damped가법경향,승법계절')
plt.legend(loc='best')
plt.show()

```

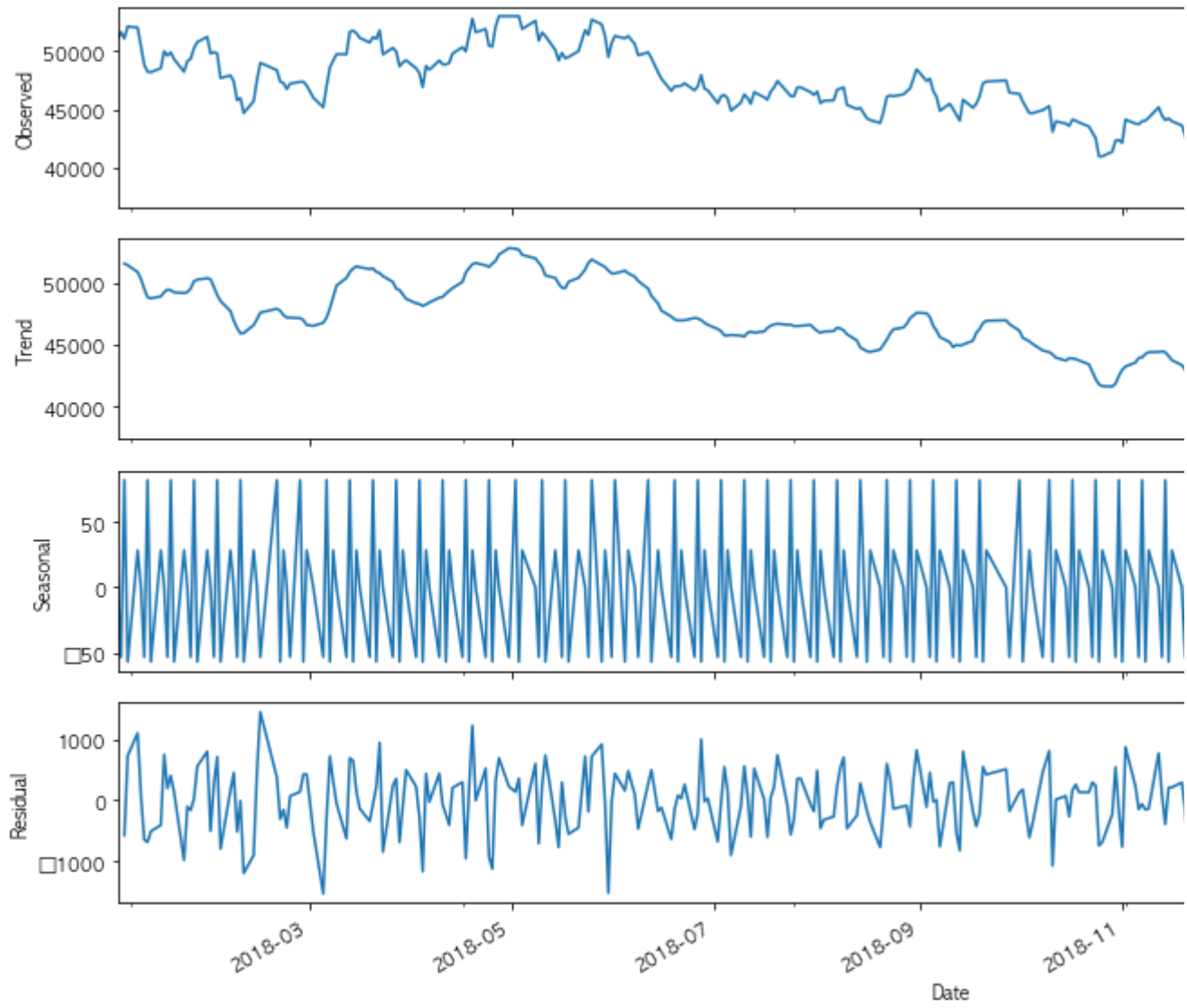
```
fit1.forecast(20).plot(style='--', marker='o', color='red', label='가법경향,가법계절
fit2.forecast(20).plot(style='--', marker='o', color='green',label='가법경향,승법계절
fit3.forecast(20).plot(style='--', marker='o', color='blue', label='Damped가법경향
fit4.forecast(20).plot(style='--', marker='o', color='yellow', label='Damped가법경
plt.legend(loc='best')
plt.show()
```



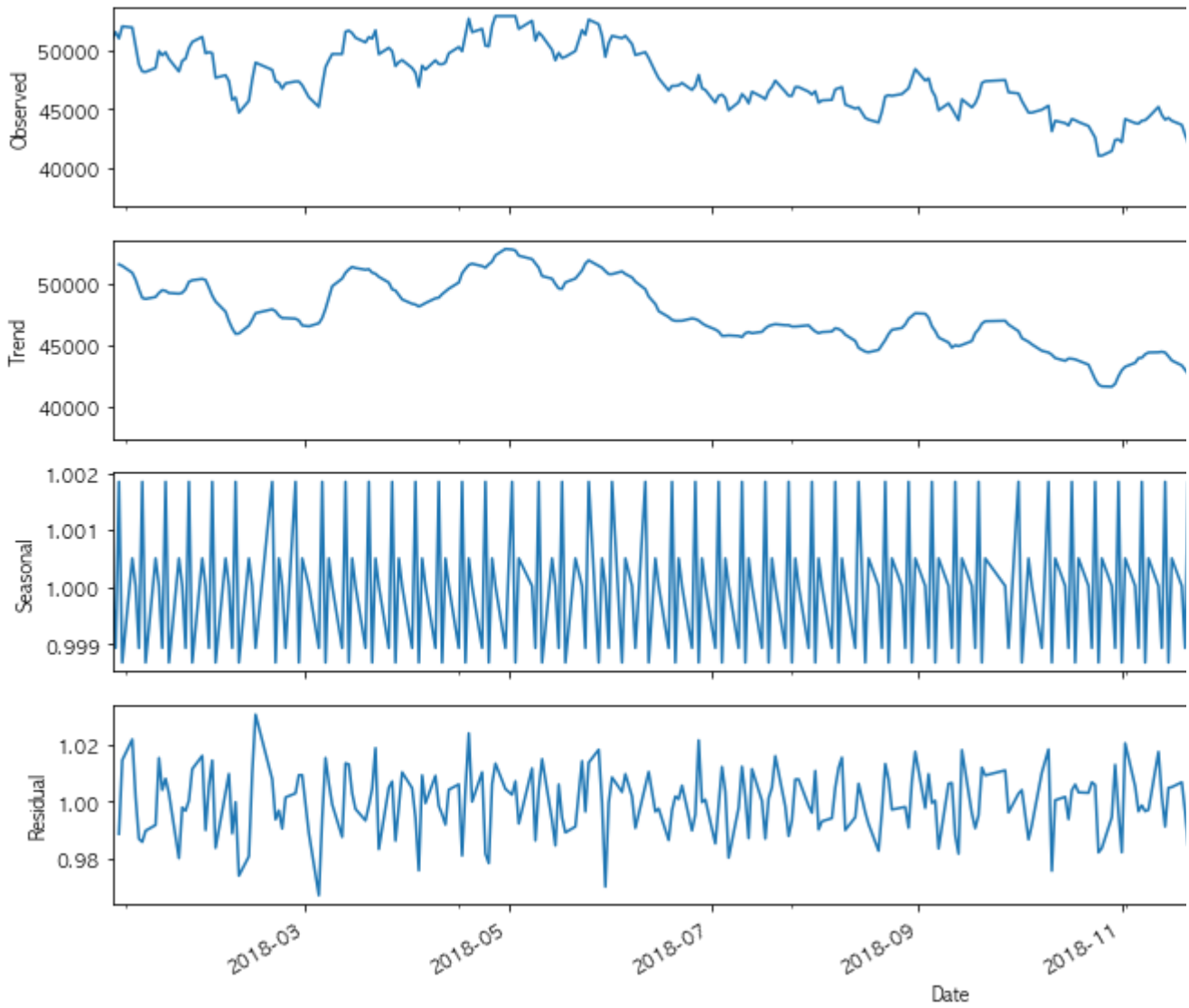

```
from statsmodels.tsa.seasonal import seasonal_decompose
import FinanceDataReader as fdr
samsung=fdr.DataReader('005930', '2018-01-01')
```

```
series=pd.Series(samsung['Close'])
```

```
result=seasonal_decompose(samsung['Close'], model='additive',freq=5)
import matplotlib.pyplot as plt
result.plot()
plt.show()
```




```
result2=seasonal_decompose(samsung['Close'], model='multiplicative',freq=5)
import matplotlib.pyplot as plt
result2.plot()
plt.show()
```


Observed
50000
45000
40000
Trend
50000
45000
40000
Seasonal
1.002
1.001
1.000
0.999
Residual
1.02
1.00
0.98
2018-03
2018-05
2018-07
2018-09
2018-11
Date