

```
In [28]: #한글깨짐
import matplotlib.pyplot as plt
from matplotlib import rc
%matplotlib inline
from matplotlib import font_manager
f_path = "C:/windows/Fonts/malgun.ttf"
font_manager.FontProperties(fname=f_path).get_name()
rc('font', family='Malgun Gothic')
```

```
In [32]: import pandas as pd

# CSV 파일 불러오기
data = pd.read_csv('total_data.csv') # 'your_data.csv'는 실제 파일 경로에 맞게

# 'be_date' 열을 날짜 형식으로 변환
data['be_date'] = pd.to_datetime(data['be_date'])

# 'be_date' 열을 인덱스로 설정
data.set_index('be_date', inplace=True)

# 'be_total_energy'가 같은 시간대 데이터를 합산하고 10분 간격으로 리샘플링
resampled_data = data.groupby(pd.Grouper(freq='10T')).agg({
    'be_ac_energy': 'sum',
    'be_light_energy': 'sum',
    'be_plug_energy': 'sum',
    'be_total_energy': 'sum',
    'be_floor': 'first' # 층 정보는 그대로 유지
})

# 리샘플링된 데이터 확인
print(resampled_data.head())
```

	be_ac_energy	be_light_energy	be_plug_energy	W
be_date				
2018-07-01 00:00:00	613.13	742.47	397.63	
2018-07-01 00:10:00	614.04	743.90	397.52	
2018-07-01 00:20:00	615.39	744.14	395.86	
2018-07-01 00:30:00	614.62	747.23	402.12	
2018-07-01 00:40:00	616.77	749.85	406.00	

	be_total_energy	be_floor
be_date		
2018-07-01 00:00:00	1753.23	1
2018-07-01 00:10:00	1755.46	1
2018-07-01 00:20:00	1755.39	1
2018-07-01 00:30:00	1763.97	1
2018-07-01 00:40:00	1772.62	1

```

In [34]: import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose

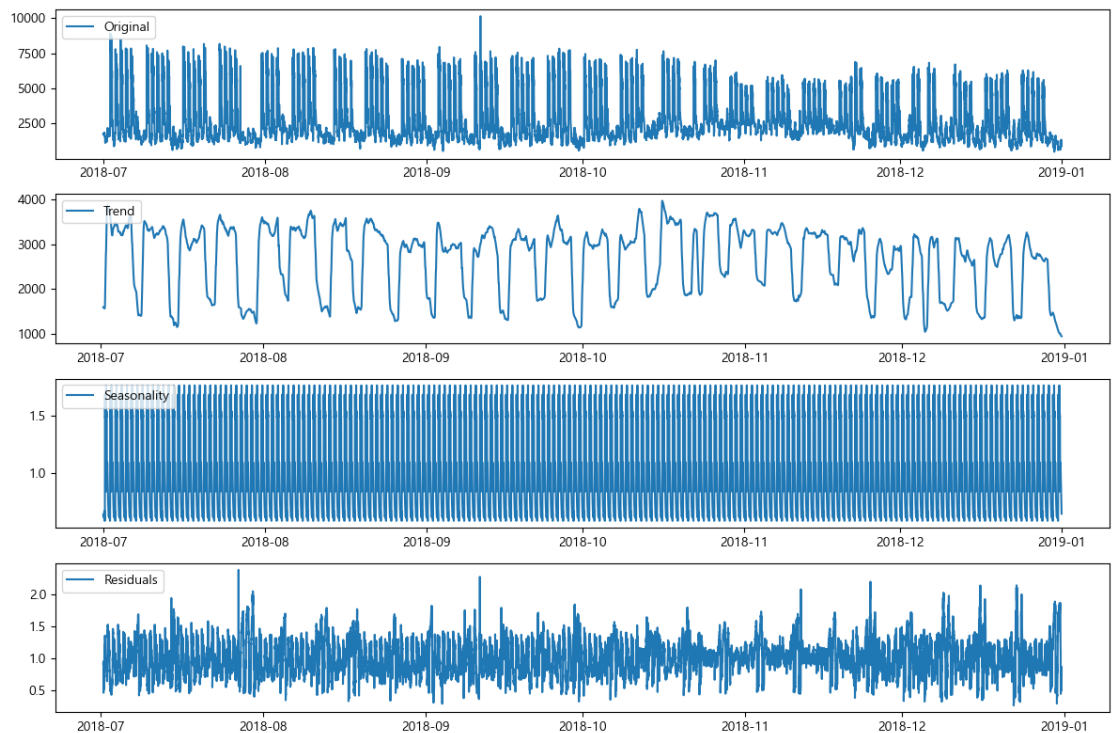
# CSV 파일 불러오기 (이전에 리샘플링된 데이터를 사용한다고 가정합니다)
data = resampled_data

# 계절성 성분 분해
result = seasonal_decompose(data['be_total_energy'], model='multiplicative', per

# 분해된 성분 확인
trend = result.trend.dropna()
seasonal = result.seasonal.dropna()
residual = result.resid.dropna()

# 성분 시각화
plt.figure(figsize=(12, 8))
plt.subplot(411)
plt.plot(data['be_total_energy'], label='Original')
plt.legend(loc='upper left')
plt.subplot(412)
plt.plot(trend, label='Trend')
plt.legend(loc='upper left')
plt.subplot(413)
plt.plot(seasonal, label='Seasonality')
plt.legend(loc='upper left')
plt.subplot(414)
plt.plot(residual, label='Residuals')
plt.legend(loc='upper left')
plt.tight_layout()
plt.show()

```



```
In [37]: from statsmodels.tsa.stattools import adfuller

# ADF 테스트를 통한 정상성 검정 토탈
result = adfuller(data['be_total_energy'])

print('ADF 통계량 (be_total_energy):', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])

if result[1] <= 0.05:
    print('결과: be_total_energy 데이터는 정상성을 가집니다.')
else:
    print('결과: be_total_energy 데이터는 정상성을 가지지 않습니다.')

# ADF 테스트를 통한 정상성 검정 AC
result = adfuller(data['be_ac_energy'])

print('ADF 통계량 (be_ac_energy):', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])

if result[1] <= 0.05:
    print('결과: be_ac_energy 데이터는 정상성을 가집니다.')
else:
    print('결과: be_ac_energy 데이터는 정상성을 가지지 않습니다.')

# ADF 테스트를 통한 정상성 검정 전등
result = adfuller(data['be_light_energy'])

print('ADF 통계량 (be_light_energy):', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])

if result[1] <= 0.05:
    print('결과: be_light_energy 데이터는 정상성을 가집니다.')
else:
    print('결과: be_light_energy 데이터는 정상성을 가지지 않습니다.')

# ADF 테스트를 통한 정상성 검정 전등
result = adfuller(data['be_plug_energy'])

print('ADF 통계량 (be_plug_energy):', result[0])
print('p-value:', result[1])
print('Critical Values:', result[4])

if result[1] <= 0.05:
    print('결과: be_plug_energy 데이터는 정상성을 가집니다.')
else:
    print('결과: be_plug_energy 데이터는 정상성을 가지지 않습니다.')
```

ADF 통계량 (be\_total\_energy): -23.245275126660538  
p-value: 0.0  
Critical Values: {'1%': -3.430597293913671, '5%': -2.861649296685967, '10%': -2.5668281753844227}  
결과: be\_total\_energy 데이터는 정상성을 가집니다.  
ADF 통계량 (be\_ac\_energy): -23.755650656971607  
p-value: 0.0  
Critical Values: {'1%': -3.430597293913671, '5%': -2.861649296685967, '10%': -2.5668281753844227}  
결과: be\_ac\_energy 데이터는 정상성을 가집니다.  
ADF 통계량 (be\_light\_energy): -17.549647290448625  
p-value: 4.148786700782625e-30  
Critical Values: {'1%': -3.4305971163568896, '5%': -2.861649218214323, '10%': -2.5668281336157257}  
결과: be\_light\_energy 데이터는 정상성을 가집니다.  
ADF 통계량 (be\_plug\_energy): -17.159808952294785  
p-value: 6.853825214049039e-30  
Critical Values: {'1%': -3.4305972565120952, '5%': -2.8616492801562514, '10%': -2.5668281665860255}  
결과: be\_plug\_energy 데이터는 정상성을 가집니다.