

## 예제 데이터 생성

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# 시계열 데이터 생성 (예시)
np.random.seed(42)
date_rng = pd.date_range(start='2020-01-01', end='2022-12-31', freq='M')
data = np.random.randn(len(date_rng)) + np.linspace(0, 20, len(date_rng)) # 추세
# 추가
ts = pd.Series(data, index=date_rng)
```

## 시계열 분해 (Time Series Decomposition)

- 시계열 데이터를 눈에 보이지 않는 여러 구성 요소로 분해하여 데이터의 구조를 파악하는 방법
- 이를 통해 데이터의 패턴을 더 잘 이해하고 예측 모델링에 활용 가능
- 주요 구성 요소:
  1. **추세 (Trend, T)**: 데이터가 장기적으로 증가하거나 감소하는 경향.
  2. **계절성 (Seasonality, S)**: 특정 시간 단위(e.g., 매년, 매월, 매주)로 주기적으로 나타나는 패턴.
  3. **주기 (Cycle, C)**: 계절성보다 주기가 길고 불규칙한 반복 패턴 (e.g., 경기 순환).
  4. **불규칙 요소 (Irregular/Residual, R)**: 위의 세 가지 요소로 설명되지 않는, 예측 불가능한 임의의 변동.
- 분해 모델:
  - **덧셈 모델 (Additive Model)**:  $Y_t = T_t + S_t + C_t + R_t$ . 각 요소가 서로 독립적이라고 가정.
  - **곱셈 모델 (Multiplicative Model)**:  $Y_t = T_t \times S_t \times C_t \times R_t$ . 시간의 흐름에 따라 계절성의 변동폭이 커지는 경우 사용. (로그 변환을 통해 덧셈 모델로 변환 가능)

```
# 시계열 분해
# model: 'additive' 또는 'multiplicative'
decomposition = seasonal_decompose(ts, model='additive', period=12)

fig = decomposition.plot()
fig.set_size_inches(10, 8)
plt.show()

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.seasonal import seasonal_decompose
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf

# 시계열 데이터 생성 (예시)
np.random.seed(42)
```

```
date_rng = pd.date_range(start='2020-01-01', end='2022-12-31', freq='M')
data = np.random.randn(len(date_rng)) + np.linspace(0, 20, len(date_rng)) # 추세
# 추가
ts = pd.Series(data, index=date_rng)

# 시계열 분해
# model: 'additive' 또는 'multiplicative'
decomposition = seasonal_decompose(ts, model='additive', period=12)

fig = decomposition.plot()
fig.set_size_inches(10, 8)
plt.show()

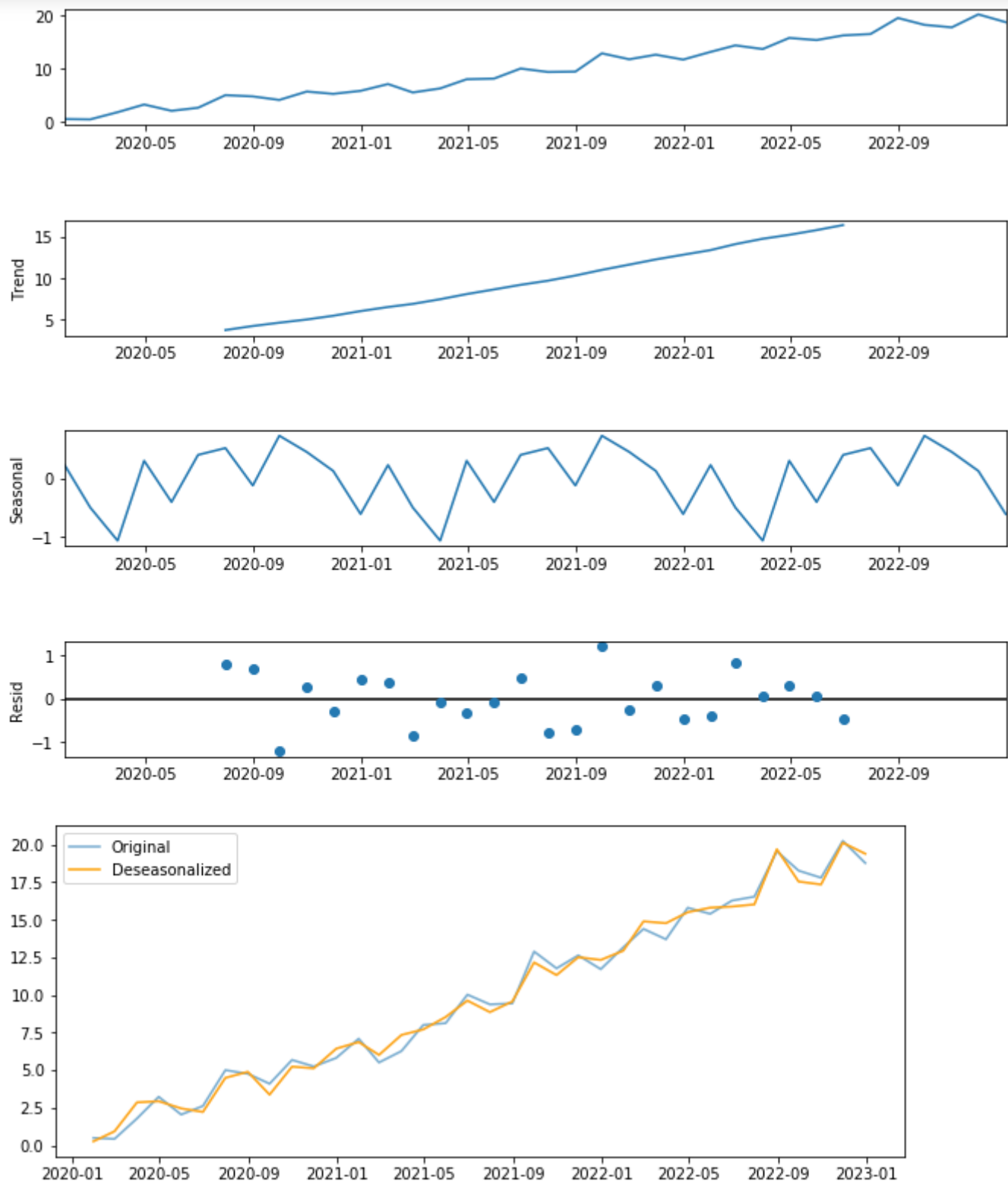
# 각 구성 요소 접근
observed = decomposition.observed # 원본 시계열
trend = decomposition.trend # 추세 (Trend)
seasonal = decomposition.seasonal # 계절성 (Seasonality)
resid = decomposition.resid # 불규칙(잔차, Residual)

# 계절성 제거(비계절화 시계열)
deseasonalized = observed - seasonal

# 추세 제거(중심화)
detrended = observed - trend

# 불규칙 성분만 보기
resid_only = resid.dropna()

# 결과 확인
plt.figure(figsize=(10, 4))
plt.plot(observed, label='Original', alpha=0.6)
plt.plot(deseasonalized, label='Deseasonalized', color='orange')
plt.legend()
plt.show()
```



## 자기상관 함수(ACF)와 편자기상관 함수(PACF)

ACF와 PACF는 시계열 데이터의 자기상관 구조를 파악하고, ARIMA 모델의 차수(p, q)를 결정하는 데 핵심적인 역할을 하는 두 가지 플롯입니다.

### 1. ACF (Autocorrelation Function)

- 개념:** 시차(lag)  $k$ 에 대해, 현재 시점의 데이터( $Y_t$ )와  $k$  시점 전의 데이터( $Y_{t-k}$ ) 사이의 상관계수를 나타냅니다.

- **특징:** 시차  $k$ 까지의 모든 시점( $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k}$ )들이  $Y_t$ 에 미치는 **직접적 및 간접적 영향**을 모두 포함하여 계산됩니다.
- **활용:** **MA(Moving Average)** 모델의 차수( $q$ )를 결정하는 데 사용됩니다. ACF 플롯에서 시차  $q$  이후의 상관계수가 급격히 0에 가까워지면(절단, cut-off), MA( $q$ ) 모델을 고려할 수 있습니다.

## 2. PACF (Partial Autocorrelation Function)

- **개념:** 시차  $k$ 에 대해,  $Y_t$ 와  $Y_{t-k}$  사이의 순수한 상관관계를 나타냅니다. 즉, 그 사이의 다른 시점들( $Y_{t-1}, Y_{t-2}, \dots, Y_{t-k+1}$ )의 영향을 제거하고 계산한 상관계수입니다.
- **특징:**  $Y_t$ 와  $Y_{t-k}$  사이의 **직접적인 영향**만을 보여줍니다.
- **활용:** **AR(Autoregressive)** 모델의 차수( $p$ )를 결정하는 데 사용됩니다. PACF 플롯에서 시차  $p$  이후의 상관계수가 급격히 0에 가까워지면(절단, cut-off), AR( $p$ ) 모델을 고려할 수 있습니다.

## 코드 예시

```
# ACF 및 PACF 플롯
# lags: 확인할 시차의 수
fig, (ax1, ax2) = plt.subplots(2, 1, figsize=(10, 8))
plot_acf(ts, lags=15, ax=ax1)
ax1.set_title('Autocorrelation Function (ACF)')
plot_pacf(ts, lags=15, ax=ax2)
ax2.set_title('Partial Autocorrelation Function (PACF)')
plt.tight_layout()
plt.show()

# 플롯 해석:
# 파란색 음영 영역은 신뢰구간(보통 95%)을 의미합니다.
# 막대그래프가 신뢰구간을 벗어나면 해당 시차의 상관계수가 통계적으로 유의미하다고 판단합니다.
# ACF가 천천히 감소하고 PACF가 특정 시점에서 절단되면 AR 모델을,
# ACF가 특정 시점에서 절단되고 PACF가 천천히 감소하면 MA 모델을 의심할 수 있습니다.
```

