시계열 데이터는 일정 기간 동안 일관된 간격으로 수집된 일련의 데이터 입니다. 특정 시점의 특정 변수 값을 기록합니다. 시계열 데이터의 예로는 주가, 날씨 데이터, 판매 데이터 및 경제 데이터가 있습니다. 시계열 데이터는 일반적으로 재무, 경제 및 엔지니어링과 같은 분야에서 패턴을 식별하고 미래 추세를 예측하며 과거 데이터를 기반으로 의사 결정을 내리는 데 사용됩니다. 시계열 데이터는 시계열 분석, ARIMA 모델, 딥 러닝 모델과 같은 통계 및 머신 러닝 방법을 사용하여 예측하고 인사이트를 얻습니다.

시계열 데이터를 이용한 시계열 예측(time series forecasting)은 과거 데이터 추세 및 패턴을 기반으로 미래 값을 예측하는 통계적 방법입니다. 시간 종속 데이터를 분석하고 판매, 주가 및 날씨 패턴과 같은 미래 이벤트에 대한 예측을 수행하는 데 사용됩니다. 여기에는 ARIMA 또는 LSTM과 같은 모델을 선택하여 데이터를 맞추고 예측하는 작업이 포함됩니다. 예측의 정확도는 데이터 품질, 모델 선택 및 데이터의 기본 패턴 존재 여부에 따라 달라집니다.

시계열 예측에는 다음과 같은 사례가 있습니다.

- 판매 예측: 회사 또는 제품의 향후 판매를 예측합니다.
- 재무 예측: 주가, 환율 또는 이자율을 예측합니다.
- 기후 예측: 기온, 강수량, 풍속과 같은 날씨 패턴을 예측합니다.
- 에너지 예측: 발전소 및 유틸리티의 에너지 수요, 생산 및 가격을 예측합니다.
- 의료 예측: 질병 확산, 입원율, 의료 자원 활용도를 예측합니다.
- 교통 예측: 교통 계획을 위한 교통 패턴, 혼잡 및 이동 시간을 예측합니다.
- 공급망 예측: 원자재, 재고 수준 및 생산 일정에 대한 수요를 예측합니다.
- 마케팅 예측: 고객 행동, 시장 동향 및 광고 효과를 예측합니다.
- 농업 예측: 작물 수확량, 날씨 패턴 및 시장 가격과 같은 농업 데이터의 미래 추세 및 패턴 예측이 포함됩니다.
- 교육 예측: 학생 등록, 졸업율 및 교사 인력과 같은 교육 데이터의 미래 추세 및 패턴을 예측합니다.
- 여행 예측: 승객 수, 항공편 예약 및 호텔 점유율과 같은 여행 관련 데이터의 미래 추세 및 패턴 예측이 포함됩니다.

표준적인 회귀(regression) 접근법은 다음과 같은 이유로 시계열 데이터에서는 잘 동작하지 않습니다.

- 시간적 종속성(temporal dependence): 시계열 데이터는 시간적 종속성이 강합니다. 데이터가 시간에 대해 상관성(correlation)을 가지고 있으므로 특정 시점의 값이 이전 값에 따라 달라집니다.
- 비정상성(non-stationarity): 시계열 데이터는 시간이 지남에 따라 변경될 수 있으므로 모든 기간에 사용할 수 있는 단일 모델을 개발하기 어렵습니다. 그 이유는 대부분의 머신 러닝 모델들이 정상성(stationary) 데이터를 가정하고 있기 때문 입니다.
- 독립적인 관찰의 부족(Lack of independent observations): 전통적인 지도 학습에서는 훈련 및 검증 세트가 서로 독립적이지만 시계열 데이터의 경우에는 그렇지 않습니다.
- 별도의 레이블(target)이 없음: 표준적인 지도 학습에서는 데이터의 특성(feature)과 레이블(target)이 별도로 분리되어 있지만 시계열 데이터의 경우 특성과 레이블이 동일 합니다

따라서 시계열 데이터는 데이터를 효과적으로 모델링하기 위해 추세 및 계절성 분석, 정상성 테스트 및 시간별 훈련/검증 세트 분할과 같은 특수 기술이 필요합니다. 또한, 수년 혹은 수십년에 걸친 패턴을 파악하려면 장기간에 걸친 데이터가 필요 합니다.

시계열 데이터의 종류

시계열 데이터는 여러가지 유형으로 나눌 수 있습니다. 각 유형은 복합적으로 하나의 시계열을 구성할 수 있습니다.

정량적(quantitative) 시계열: 이 유형의 시계열 데이터는 숫자 데이터이며 연속적(continuous)이거나 이산적(discrete)일 수 있습니다. 예를 들면 주가, 매출 데이터 및 온도 등이 있습니다.

정성적(qualitative) 시계열: 이 유형의 시계열 데이터는 범주형(categorical)이며 이진 또는 다중 클래스일 수 있습니다. 예를 들면 요일, 월, 기상 조건 등이 있습니다.

또한 시계열 데이터는 다음과 같은 동작에 따라 추가로 분류할 수도 있습니다.

추세(trend) 시계열: 시간 경과에 따른 시계열 값의 점진적인 변화입니다.

계절성(seasonality) 시계열: 특정 시간 간격(예: 1년 또는 분기) 동안 발생하는 시계열의 반복 패턴입니다.

주기적(periodic) 시계열: 몇 년과 같이 계절성보다 긴 시간 간격에 걸쳐 발생하는 시계열의 반복 패턴입니다.

불규칙(Irregular) 시계열: 명확한 추세나 반복되는 패턴을 나타내지 않는 시계열.

변수의 개수에 따라 다음과 같이 구분할 수도 있습니다.

일변량(univariate) 시계열: 일정한 시간 간격으로 기록된 일련의 데이터 포인트이며 여기서 하나의 변수만 측정되고 기록됩니다. 시간 경과에 따른 단일 변수의 동작에 대한 정보를 제공합니다.

다변량(multivariate) 시계열: 하나 이상의 변수를 포함하며 각 변수는 동일한 시간 간격으로 기록되고 분석됩니다. 시간 경과에 따른 여러 변수 간의 관계에 대한 정보를 제공합니다. 다변량 시계열 분석의 목표는 서로 다른 변수가 서로 상호 작용하는 방식과 시간이 지남에 따라 서로의 행동에 미치는 영향을 이해하는 것입니다.

통계적 특성에 따라 정상(stationary)과 비정상(non-stationary) 시계열로 나눌 수도 있습니다.

정상(stationary) 시계열 : 평균, 분산, 자기 상관성과 같은 통계적 속성이 시간이 지남에 따라 변경되지 않는 경우 시계열은 정상성(stationarity)을 가진 것으로 간주됩니다.

비정상(non-stationary) 시계열: 시간이 지남에 따라 변하는 평균, 분산 또는 자기 상관성을 가진 시계열 데이터 입니다. 이러한 시계열은 데이터의 패턴과 추세를 식별하기 더 어렵게 만들 수 있기 때문에 정상 시계열보다 모델링하고 예측하기가 더 어려운 경우가 많습니다.

정상 시계열은 ARIMA 및 GARCH 모델을 비롯한 많은 통계 모델링 기법에서 가정하는 경우가 많습니다. 미래 가치를 더 쉽게 모델링하고 예측할 수 있기 때문입니다. 시계열이 비정상적이면 연속 관측치 간의 차이를 사용하여 추세를 제거하고 시간이 지남에 따라 평균과 분산을 일정하게 만드는 차분(differencing)이라는 프로세스를 통해 정상 상태로 만들 수 있습니다.

정상, 비정상 시계열(stationary, non-stationary time series) 데이터 이해

파이썬 프로그램을 이용하여 정상, 비정상 시계열의 특징을 이해해 봅니다.
우선 필요한 라이브러리를 임포트 합니다.

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

정상성(stationarity) 이해

1. 일정한 평균 (constant mean)
2. 일정한 분산 (constant variance)
3. 일정한 자기상관 구조 (constant autocorrelation structure)
4. 주기적 성분 (periodic component) 없음

이상의 4 가지 특성을 가지고 있는 시계열의 경우 정상(stationary) 시계열이라고 합니다.
위와 같은 특성을 가진 가상의 정상 시계열을 인공적으로 만들어 시각화 해 보겠습니다.

```
# 난수 발생 초기화
np.random.seed(101)
# 시간 스텝 지정
time_steps = np.arange(500)
# white noise 생성
stationary_noise = np.random.normal(loc=0, scale=1.0, size=len(time_steps))
```

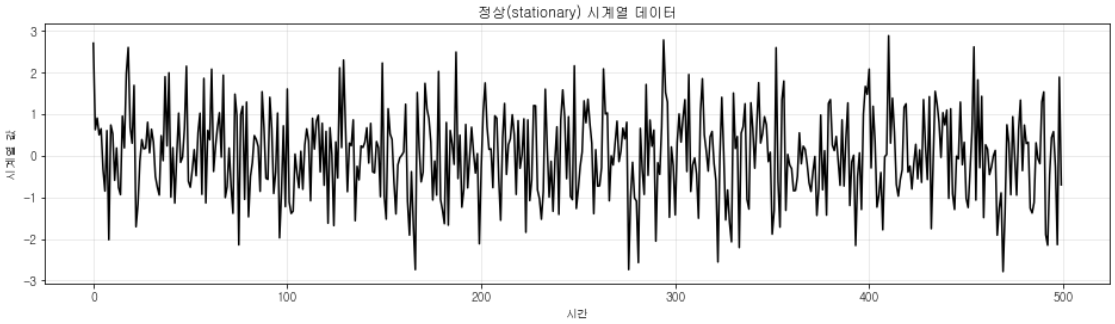
- 랜덤 숫자를 백색 잡음(white noise)으로 시계열에 추가 합니다. 백색 잡음은 일정한 평균과 일정한 분산을 갖는 값의 임의의 시퀀스 데이터이며, 어느 시점의 값이 다른 시점의 값과 상관 관계가 없다는 특성을 가지고 있습니다. 위에서는 평균이 0 이고 표준 편차가 1 인 표준 정규 분포의 잡음을 인위적으로 만들었습니다.

```
def plot_time_series(x, y, title):
    plt.figure(figsize=(16, 4))
    plt.plot(x, y, 'k-')
    plt.title(title)
    plt.xlabel("시간")
```

```
    plt.ylabel("시계열 값")
    plt.grid(alpha=0.3)
    plt.show()
```

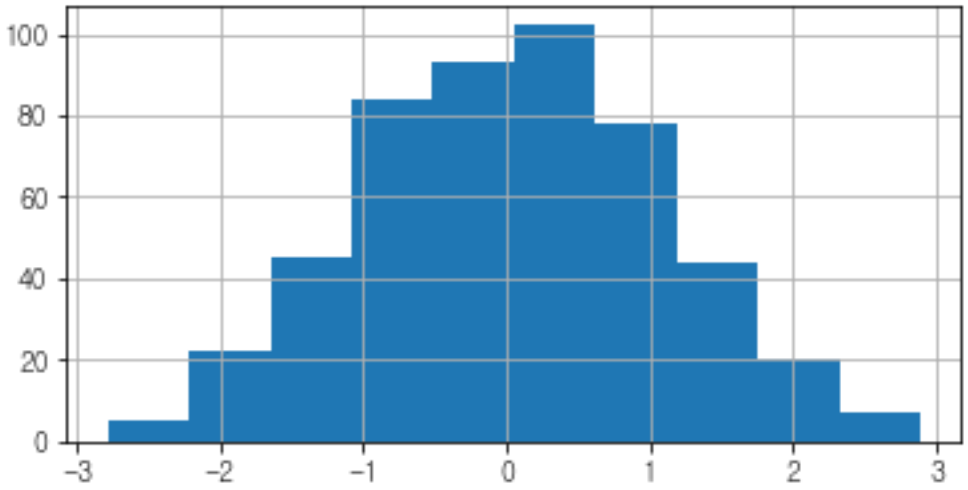
- 반복되는 플로팅을 위해 시계열 데이터를 시각화하는 함수를 정의하고 위에서 생성한 백색 잡음 시계열을 시각화 해 봅니다.

```
plot_time_series(time_steps, stationary_noise, title="정상(stationary) 시계열 데이터")
```



- 위의 플롯에는 뚜렷한 추세나 계절성이 없습니다. 이 시계열 데이터는 4 가지 정상성 조건인 일정한 평균, 일정한 분산, 일정한 자기상관, 주기 성분 없음을 충족합니다.

```
pd.Series(stationary_noise).hist(figsize=(6, 3));
```



- 시계열의 히스토그램을 시각화하면 평균과 분산이 일정한 정규 분포를 이루고 있는 것을 확인할 수 있습니다.

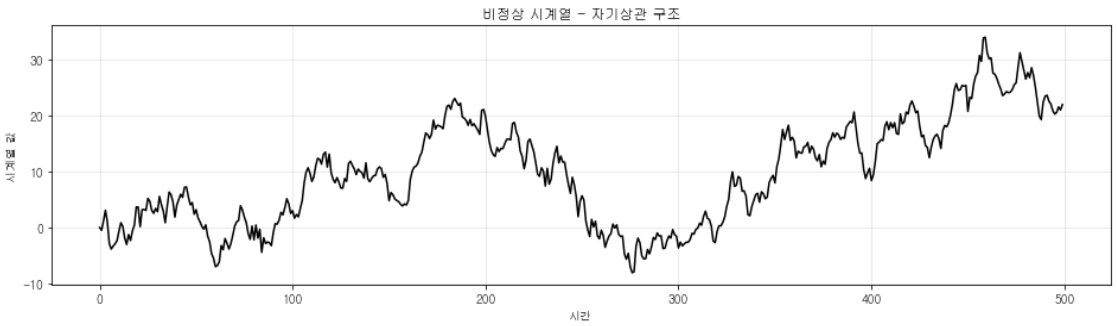
자기상관 구조(autocorrelation structure) 이해

자기 상관 구조는 서로 다른 시차를 가진 시계열 값 간의 상관 패턴을 나타냅니다. 자기 상관 구조는 시계열의 패턴이나 추세를 식별하는 데 사용할 수 있습니니다. 필요한 경우, 자기상관성은 차분(difference)에 의해 정상성 시계열로 바꿀 수 있습니다.

```
# 시계열 초기값
seed_value = 0

# 자기 상관 데이터 생성
auto_correlated_series = np.empty_like(time_steps, dtype='float')
for t in time_steps:
    auto_correlated_series[t] = seed_value + np.random.normal(loc=0, scale=1.5,
size=1)
    seed_value = auto_correlated_series[t]   #자기 상관성 부여
```

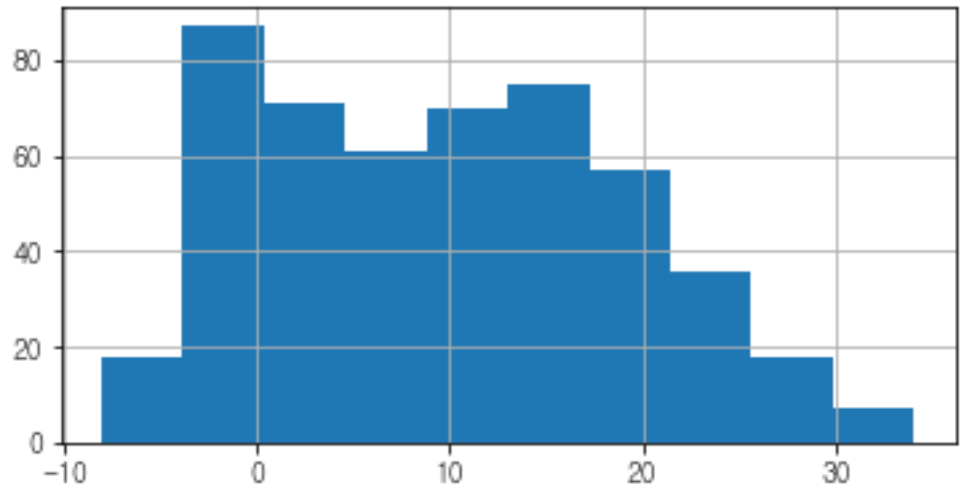
- 시간 단계의 값이 다음 시간 단계의 값과 상관 관계를 가지도록 seed_value 변수 값을 바꾸어 주며 시계열 데이터를 생성 하였습니다.
- 생성한 데이터를 시각화 해 봅니다.

```
plot_time_series(time_steps, auto_correlated_series, title="비정상 시계열 - 자기상관
구조")
```



- 위에서 생성한 시계열이 균일분포 등 비정규분포를 보이면 시계열 데이터가 비정상(non-stationay)이라고 판단 가능합니다.

```
pd.Series(auto_correlated_series).hist(figsize=(6, 3));
```



- 시계열 데이터가 정규 분포 형태를 보이지 않으므로 비정상(non_stationary)이라고 판단됩니다.

비정상(non-stationary) 데이터의 특징

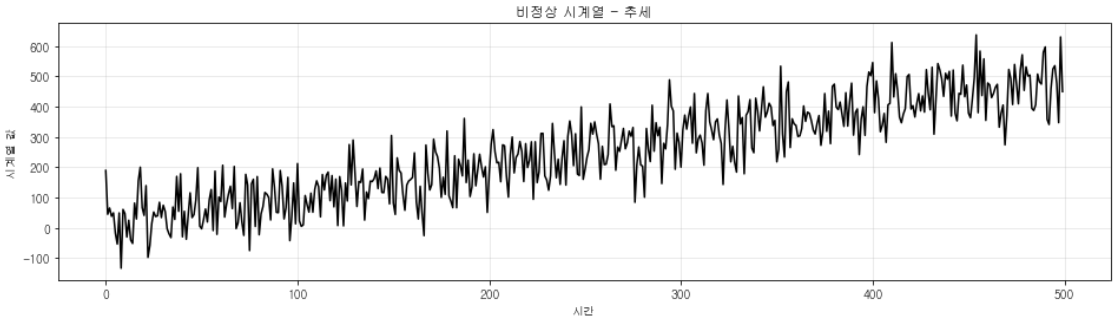
5. 추세(Trend)가 있는 시계열 데이터 - 평균의 변화가 존재
6. 이분산성 - 분산의 변동이 존재
7. 계절성(Seasonality) - 주기적 구성요소가 존재
8. 추세 + 계절성 (Trend + Seasonality)

위의 특징 중 한 가지가 존재하면 비정상(Non-Stationary) 데이터라고 부릅니다.

추세(trend)

추세는 시간이 지남에 따라 증가하거나 감소하는 시계열 값의 체계적인 변화 패턴을 나타냅니다. 추세가 있는 시계열 데이터는 평균이 시간이 지남에 따라 변하므로 비정상 데이터입니다.

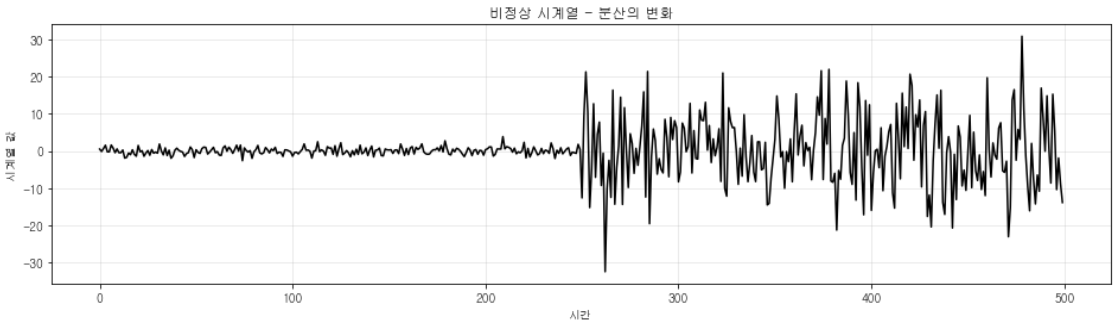
```
trend_time_series = time_steps + stationary_noise * 70

plot_time_series(time_steps, trend_time_series, title="비정상 시계열 - 추세")
```



이분산성(heteroscedasticity)

시계열 데이터의 이분산성은 시간이 지남에 따라 시계열 값이 일정하지 않게 변하는 특성을 나타냅니다. 이는 시간이 지남에 따라 값의 변동성(분산)이 변하여 데이터 모델링 및 정확한 예측에 어려움을 초래할 수 있음을 의미합니다. 이분산성은 데이터를 생성하는 기본 프로세스의 변경 또는 변동성에 영향을 미치는 외부 이벤트와 같은 다양한 요인으로 인해 발생할 수 있습니다.

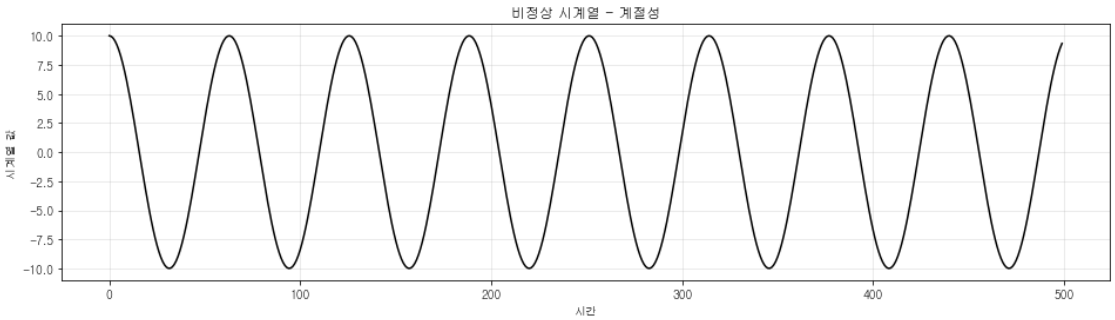
```
# 난수 발생 초기화
np.random.seed(42)

# 분산이 다른 시계열 생성
series_1 = np.random.normal(loc=0, scale=1.0, size = 250)
series_2 = np.random.normal(loc=0, scale=10.0, size = 250)
hetero_time_series = np.append(series_1, series_2)

plot_time_series(time_steps, hetero_time_series, title="비정상 시계열 - 분산의 변화
")
```



계절성(seasonality)

시계열에 주기적인 요소가 포함되어 있으므로 비정상(non-stationary) 시계열로 분류 됩니다.

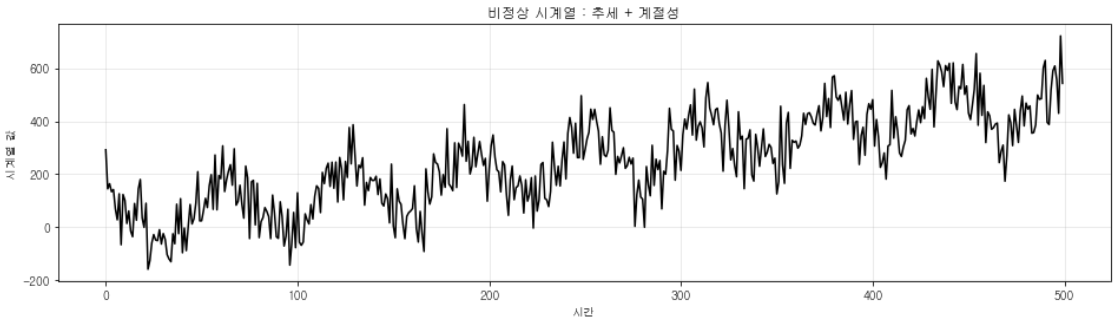
```
seasonality = np.cos(time_steps/10) * 10
plot_time_series(time_steps, seasonality, title="비정상 시계열 - 계절성")
```



추세 + 계절성

평균이 시간 경과에 따라 변화하고 주기적인 구성 요소가 있는 인공 시계열을 만들어 시각화 해 봅니다.

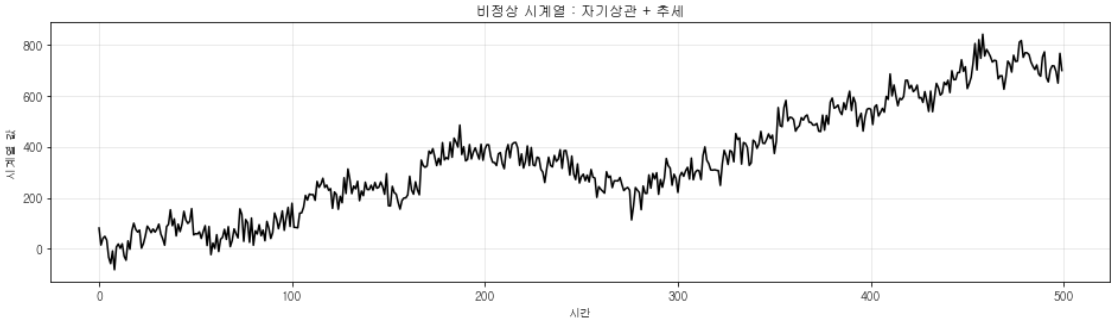
```
trend_and_seasonality = trend_time_series + seasonality * 10 + stationary_noise

plot_time_series(time_steps, trend_and_seasonality, title="비정상 시계열 : 추세 + 계절성")
```



자기상관 + 추세 (autocorrelation + trend)

자기상관 구조와 추세를 동시에 가지고 있는 인공 시계열 데이터를 생성하여 시각화 해 봅니다.

```
auto_correlated_and_trend = auto_correlated_series * 10 + trend

plot_time_series(time_steps, auto_correlated_and_trend, title="비정상 시계열 : 자기상관 + 추세")
```



시계열 데이터 처리에서의 정상성 (stationarity)의 중요성

정상성은 전통적인 시계열 분석에서 중요한 개념이며 시계열 모델링을 위한 머신 러닝 모델의 성능에 영향을 미칠 수 있습니다.

통계학적 기법들은 모두 시계열 데이터의 정상성을 전제로 하고 있습니다. 평균 및 분산과 같은 통계학적 속성이 시간 경과에 무관하게 일정하므로 패턴을 모델링하고 미래 값을 예측하기가 더 쉽습니다. 반면 비정상 시계열은 시간이 지남에 따라 통계적 특성이 변하므로 모델링 및 예측이 더 어려울 수 있습니다.

심층 신경망의 경우에도 정상성은 모델의 안정성과 예측 정확도에 영향을 줄 수 있으므로 여전히 중요한 고려 사항입니다. 그러나 복잡한 표현을 학습하는 심층 신경망의 능력은 종종 비정상성으로 인한 문제를 어느 정도 극복할 수 있습니다.

따라서 심층 신경망을 사용하기 전에 시계열 데이터의 정상성을 확인하고 그에 따라 사전 처리하는 것이 좋습니다. 시계열을 정상 상태로 만드는 가장 일반적인 방법은 추세와 계절성을 제거하는 것입니다. 제거 방법은 뒤에서 자세히 설명 하겠습니다.

시계열 데이터에 대한 일반적인 접근법은 다음과 같습니다.

1 단계 : 시계열 데이터의 정상성을 체크하고 비정상(non-stationary) 데이터의 경우 2 단계 이후를 실행합니다.

2 단계 : 시계열 데이터에 다음 사항 적용

- 추세(trend)를 제거하여 평균값을 일정하게 만듭니다.
- 이분산성(heteroscedasticity)을 제거하여 분산을 일정하게 만듭니다.
- 자기상관성(autocorrelation)을 제거 합니다.
- 계절성(seasonality)을 제거 합니다.

3 단계 : 위에서 얻어진 정상 시계열을 이용하여 머신 러닝 모델을 구축 합니다.

그러면 구체적으로 정상성을 체크하는 방법에 대해 알아보겠습니다.

정상성 체크 방법

시계열 데이터의 정상성을 체크하는 대표적인 방법은 Augmented Dickey-Fuller(ADF) Test 입니다.

ADF Test는 시계열의 정상성을 확인하는 데 사용되는 통계 테스트입니다. 정상 시계열은 시간이 지남에 따라 일정한 평균과 분산을 가지며 많은 시계열 모델에서 중요한 가정입니다.

ADF 검정은 검정 통계량과 p-값을 출력합니다. 검정 통계량은 비정상성이라는 귀무가설(H0)에 대한 증거의 강도를 측정하고 p-값은 귀무가설이 참인 경우(즉, 시계열이 비정상성임) 검정 통계량 값 보다 더 극단적일 것으로 관측될 확률을 나타냅니다. 작은 p-값(일반적으로 0.05 미만)은 귀무가설에 반대(즉, 시계열이 정상성임)되는 대립가설(H1)의 강력한 증거를 나타내므로 시계열이 정상성이라는 결론을 뒷받침합니다.

Augmented Dickey-Fuller 테스트는 계량 경제학 및 금융 분야에서 널리 사용되며 시계열 데이터의 정상성에 대한 표준 테스트 중 하나로 간주됩니다. 그러나 ADF 테스트는 정상성을 보장하지 않으며 다른 테스트와 방법을 사용하여 시계열의 정상성을 확인해야 한다는 점에 유의해야 합니다.

Augmented Dickey-Fuller(ADF) Test를 요약 정리하면 다음과 같습니다.

- 귀무가설(H0) : 시계열이 비정상성(non-stationary)이다.
- 대립가설(H1) : 시계열이 정상성(stationary)이다.
- p-값이 0.05 보다 작으면 H0를 기각 합니다. 따라서, 정상 시계열.
- p-값이 0.05 보다 크면 H0를 기각할 수 없습니다. 따라서, 비정상 시계열.

귀무가설(null hypothesis)과 대립가설(alternative hypothesis)은 통계적 검정에서 중요한 개념입니다.
귀무가설은 우리가 검정하려는 가설이 참이라는 가정입니다. 예를 들어, 특정 처방약이 효과가 없다는 가설이 귀무가설이 될 수 있습니다. 영가설이라고도 불리는 귀무가설은 H0로 표시하며 처음부터 기각(reject)될 것을 예상하는 가설입니다.

대립가설은 귀무가설이 거짓이라는 가정입니다. 위의 예에서, 특정 처방약이 효과가 있다는 것이 대립가설입니다.

귀무가설은 검정이 시작되기 전에 설정되어야 하며, 검정 결과에 따라 귀무가설을 기각하거나 채택할 수 있습니다.

파이썬 코드를 이용하여 ADF Test의 통계학적 배경이 되는 가설검정을 이해해 보겠습니다.

검정과 유의 확률

가설(hypothesis)은 확률 분포에 대한 어떤 주장을 말합니다.

검정(testing)은 통계적 가설 검정(statistical hypothesis testing)의 줄임말이며 데이터 뒤에 숨어있는 확률변수의 분포에 대한 가설이 맞는지 틀리는지 정량적으로 증명하는 작업입니다. 예를 들어 다음과 같은 문제가 가설 검정의 대상 입니다.

- 어떤 동전을 15 번 던졌는데 앞면이 12 번 나왔다면 이 동장은 공정한 동전(fair coin)이라고 말 할 수 있습니까 ?
- Apple 주식의 5 일간 수익률이 다음과 같다면 Apple 주식은 장기적으로 손실이 나는 주식입니까 ?
  -2.5%, -5%, 4.3%, -3.7%, -5.6%

귀무가설(null hypothesis, 영가설)

검정 작업을 하기 위해 두 가지의 가정을 합니다.

1. 데이터가 어떤 확률 분포를 따른다고 가정
2. 그 확률 분포의 모수(parameter) θ 값이 특정한 값 θ_0 라고 가정 → 즉, 모집단의 모수는 θ_0와 같다고 가정

위에서 확률 분포의 모수에 대한 가정을 귀무가설 또는 영가설이라고 하며 H_0로 표시합니다. 귀무가설은 부등호를 사용하면 복잡해 지므로 편의상 다음과 같은 등식으로 표현하며, 여기서 θ_0는 우리가 증명하고자 하는 가설에 대한 기준값이 되는 상수입니다.

$$H_0: \theta = \theta_0$$

예를 들어 다음과 같습니다.

동전 던지기 문제

확률 분포 가정 : 베르누이 분포
모수 가정 : 공정한 동전이라면 앞면이 나올 확률이 0.5 이므로 $\mu = 0.5$로 가정.

따라서, 동전 던지기 문제의 귀무가설은 다음과 같이 정의합니다.

$$H_0: \mu = 0.5$$

Apple 주식 문제

확률 분포 가정 : 정규 분포

모수 가정 : 장기적으로 손실을 보는 경우는 기대값이 음수, 이익을 보는 경우는 기대값이 양수이므로 두 가지 경우를 나누는 기준값 0 을 $\mu = 0$로 가정.

따라서, Apple 주식 문제의 귀무가설은 다음과 같이 정의합니다.

$$H_0: \mu = 0$$

대립가설(alternative hypothesis, 대안가설)

귀무가설이 거짓일 경우 대안적으로 참이 되는 가설을 대립가설이라고 합니다. 즉, "모집단의 모수는 θ_0와 다르다" 또는 "모집단의 모수는 θ_0와 차이가 있다"라고 가정합니다. 대립가설은 H_1으로 표시하고 일반적으로 다음의 3 가지 경우로 나눌 수 있습니다.

1) 모수 θ가 θ_0가 아니라는 것을 증명하고 싶은 경우

$$H_0: \theta = \theta_0, \quad H_1: \theta \neq \theta_0$$

2) 모수 θ가 θ_0보다 크다는 것을 증명하고 싶은 경우

$$H_0: \theta = \theta_0, \quad H_1: \theta > \theta_0$$

3) 모수 θ가 θ_0보다 작는 것을 증명하고 싶은 경우

$$H_0: \theta = \theta_0, \quad H_1: \theta < \theta_0$$

위에서 예를 든 문제를 귀무가설과 대립가설로 다시 정리하면 다음과 같습니다.

동전 던지기 문제

동전 던지기 문제의 경우 실제 데이터에서 얻은 검정 통계량이 $\frac{12}{15} = 0.8$이므로 우리가 증명하고 싶은 것은 동전이 공정하지 않다는 것 입니다. 이 경우 귀무가설과 대립가설은은 다음과 같이 정의합니다.

$$H_0: \mu = 0.5, \quad H_1: \mu \neq 0.5$$

이 가정을 증명하려면 귀무가설이 틀렸다는 것을 증명하면 됩니다.

Apple 주식 문제

Apple 주식 문제의 경우 실제 데이터에서 얻은 검정 통계량, 즉 5 거래일간 평균 수익률이 ((-2.5) + (-5) + 4.3 + (-3.7) + (-5.6)) / 5 = -2.5 이고 이 주식이 장기적으로 손실을 내는가가 질문이므로 우리가 증명하고 싶은 것은 Apple 주식의 평균 수익률이 0 보다 작다는 것 입니다. 이 경우 귀무가설과 대립가설은 다음과 같이 정의합니다.

$$H_0: \mu = 0, \quad H_1: \mu < 0$$

이 가정의 증명은 귀무가설이 틀렸다는 것을 증명하되, 대립가설이 맞는 방향으로 귀무가설이 틀렸다는 것을 증명하면 됩니다.