시계열 분석

시계열 분석은 데이터가 일련의 시간 순서에 따라 수집되었을 때 발생하는 패턴과 구조를 이해하고 예측하기 위한 기법입니다. 시계열 데이터는 일정한 간격으로 측정된 관찰값을 포함하며, 경향성, 계절성, 주기성 등의 패턴을 가지고 있을 수 있습니다. 시계열 분석은 시간 경과에 따른 변동성을 이해하고 예측하기 위해 사용됩니다.

Statsmodels

Statsmodels는 통계 모델링, 가설 검정, 데이터 탐색 분석, 시계열 분석 등을 수행하기 위한 다양한 통계 모델과 함수를 제공하는 오픈 소스 라이브러리입니다. R과 유사한 기능을 Python에서 구현하며, pandas, numpy, scipy 등의 라이브러리와 잘 연동되어 데이터 분석을 더욱 쉽게 수행할 수 있습니다.

In [3]:
```
# 설치방법
!pip install statsmodels
```

```
Requirement already satisfied: statsmodels in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (0.13.5)
Requirement already satisfied: scipy>=1.3 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from statsmodels) (1.10.0)
Requirement already satisfied: packaging>=21.3 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from statsmodels) (22.0)
Requirement already satisfied: pandas>=0.25 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from statsmodels) (2.1.4)
Requirement already satisfied: patsy>=0.5.2 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from statsmodels) (0.5.3)
Requirement already satisfied: numpy>=1.17 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from statsmodels) (1.23.5)
Requirement already satisfied: pytz>=2020.1 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas>=0.25->statsmodels) (2022.7)
Requirement already satisfied: tzdata>=2022.1 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas>=0.25->statsmodels) (2023.3)
Requirement already satisfied: python-dateutil>=2.8.2 in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from pandas>=0.25->statsmodels) (2.8.2)
Requirement already satisfied: six in /Users/youngjinseo/anaconda3/lib/python3.10/site-packages (from patsy>=0.5.2->statsmodels) (1.16.0)
```

라이브러리 불러오기

In [1]:
```
import pandas as pd

from statsmodels.tsa.arima.model import ARIMA # 시계열 분석

import numpy as np

import matplotlib.pyplot as plt
```

In [ ]:
```
```

ARIMA

ARIMA는 AutoRegressive Integrated Moving Average의 약자로, 시계열 데이터의 예측 및 분석을 위해 널리 사용되는 통계적 모델입니다. 이 모델은 세 가지 구성 요소로 이루어져 있으며, 각 구성 요소의 역할은 다음과 같습니다.

데이터셋 설명

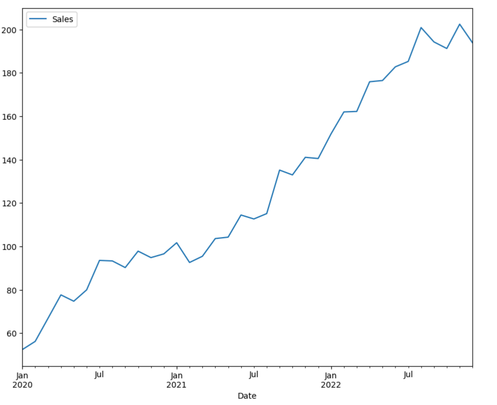
- 이 데이터셋은 월별 판매량의 시계열 데이터를 생성합니다.
- 계절성, 추세, 그리고 랜덤 요소가 포함되어 있습니다.

In [2]:
```
import pandas as pd
import numpy as np

# 랜덤 시드 설정
np.random.seed(42)

# 날짜 생성
date_range = pd.date_range(start='2020-01-01', periods=36, freq='M')

# 추세, 계절성, 랜덤 요소를 합쳐 시계열 데이터 생성
trend = np.linspace(50, 200, 36)
seasonality = 10 * np.sin(np.linspace(0, 3 * np.pi, 36))
random_noise = np.random.normal(0, 5, 36)

sales_ts = trend + seasonality + random_noise

# 데이터프레임 생성
timeseries_data = pd.DataFrame({
    'Date': date_range,
    'Sales': sales_ts
}).set_index('Date')

# 데이터 확인
print(timeseries_data.head())
```

```
                Sales
Date
2020-01-31  52.483571
2020-02-29  56.254761
2020-03-31  66.938864
2020-04-30  77.700241
2020-05-31  74.778046
```

In [8]:
```
# 선 그래프
timeseries_data.plot(figsize = (10,8))
```

Out[8]:
```
<Axes: xlabel='Date'>
```



In [3]:
```
# ARIMA 모델 (p, d, q)
p, d, q = 1, 1, 1

# ARIMA 모델 적합
model = ARIMA(timeseries_data['Sales'], order=(p, d, q))
model_fit = model.fit()

# 모델 요약
print(model_fit.summary())
```

```
                               SARIMAX Results
==============================================================================
Dep. Variable:                  Sales   No. Observations:                   36
Model:                 ARIMA(1, 1, 1)   Log Likelihood                -119.396
Date:                Thu, 15 Aug 2024   AIC                            244.792
Time:                        23:23:00   BIC                            249.458
Sample:                    01-31-2020   HQIC                           246.403
                         - 12-31-2022
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          1.0000      0.002    561.223      0.000       0.997       1.003
ma.L1         -0.9982      0.302     -3.306      0.001      -1.590      -0.406
sigma2        50.2264      0.006   8291.748      0.000      50.214      50.238
===================================================================================
Ljung-Box (L1) (Q):                   3.89   Jarque-Bera (JB):                 0.74
Prob(Q):                              0.05   Prob(JB):                         0.69
Heteroskedasticity (H):               1.78   Skew:                             0.08
Prob(H) (two-sided):                  0.33   Kurtosis:                         2.30
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
/Users/youngjinseo/anaconda3/lib/python3.10/site-packages/statsmodels/tsa/base/tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency
M will be used.
  self._init_dates(dates, freq)
/Users/youngjinseo/anaconda3/lib/python3.10/site-packages/statsmodels/tsa/base/tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency
M will be used.
  self._init_dates(dates, freq)
/Users/youngjinseo/anaconda3/lib/python3.10/site-packages/statsmodels/tsa/base/tsa_model.py:471: ValueWarning: No frequency information was provided, so inferred frequency
M will be used.
  self._init_dates(dates, freq)
/Users/youngjinseo/anaconda3/lib/python3.10/site-packages/statsmodels/tsa/statespace/sarimax.py:966: UserWarning: Non-stationary starting autoregressive parameters found. U
sing zeros as starting parameters.
  warn('Non-stationary starting autoregressive parameters'
/Users/youngjinseo/anaconda3/lib/python3.10/site-packages/statsmodels/tsa/statespace/sarimax.py:978: UserWarning: Non-invertible starting MA parameters found. Using zeros a
s starting parameters.
  warn('Non-invertible starting MA parameters found.'
/Users/youngjinseo/anaconda3/lib/python3.10/site-packages/statsmodels/base/model.py:604: ConvergenceWarning: Maximum Likelihood optimization failed to converge. Check mle_r
etvals
  warnings.warn("Maximum Likelihood optimization failed to "
```

AR (AutoRegressive) - p

개념: 자기회귀 모델은 과거의 값들이 현재의 값에 영향을 미친다는 가정을 기반으로 합니다. p는 과거의 몇 시점까지의 데이터를 사용하여 현재의 값을 예측할지를 결정합니다.

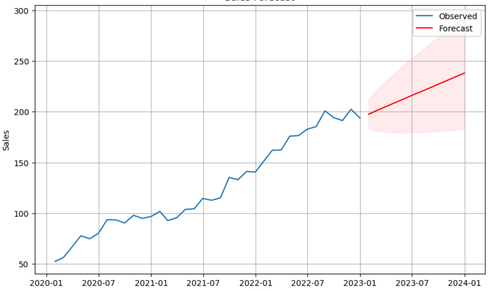
I (Integrated) - d

개념: 비정상(non-stationary) 시계열 데이터를 차분(differencing)하여 정상(stationary) 시계열로 변환합니다. d는 차분을 몇 번 수행할지를 나타냅니다.

MA (Moving Average) - q

개념: 이동평균 모델은 과거의 오차(term)들이 현재의 값에 영향을 미친다는 가정을 기반으로 합니다. q는 과거의 몇 시점까지의 오차를 사용하여 현재의 값을 예측할지를 결정합니다.

In [5]:
```
# 미래 예측
forecast_steps = 12
forecast = model_fit.get_forecast(steps=forecast_steps)
forecast_ci = forecast.conf_int()

# 예측 결과 시각화
plt.figure(figsize=(10, 6))
plt.plot(timeseries_data['Sales'], label='Observed')
plt.plot(forecast.predicted_mean, label='Forecast', color='red')
plt.fill_between(forecast_ci.index,
                 forecast_ci.iloc[:, 0],
                 forecast_ci.iloc[:, 1], color='pink', alpha=0.3)
plt.title('Sales Forecast')
plt.xlabel('Date')
plt.ylabel('Sales')
plt.legend()
plt.grid(True)
plt.show()
```



결론

ARIMA 모델은 시계열 데이터 분석에서 널리 사용되는 강력한 도구입니다. (p, d, q) 파라미터의 적절한 선택은 모델의 성능에 큰 영향을 미치며, 데이터의 패턴 및 특성에 따라 조정되어야 합니다. ARIMA 모델을 통해 시계열 데이터의 복잡한 패턴을 이해하고 미래의 변화를 예측할 수 있습니다.