시계열 기본 개념

- 시간의 흐름에 따라 기록된 데이터(시계열 데이터)의 패턴을 분석하고, 이를 바탕으로 미래 값을 예측하는 방법론
- 시계열 데이터: 주가, 날씨, 판매량 등 시간의 영향을 받는 대부분의 데이터
  - 행과 행에 시간의 순서(흐름)이 있고, 시간간격이 동일한 데이터 → `Sequential Data`에 포함됨
- 시계열 데이터는 일반적인 데이터와 달리 시간적인 순서와 종속성을 가지므로, 이를 고려한 분석 기법이 필요

시계열 데이터 변환

- `pd.to_datetime()` : object 형태의 날짜 데이터를 datetime 데이터로 바꿔준다.
  - `format = ' '` : 입력하는 날짜의 형태가 어떤 형식인지 알려주는 옵션
  - `%Y-%m-%d` : '몇년-몇월-몇일' 로 작성된지 알려줌
    → 왠만해서는 혼자서 format 형식을 잡을 수 있음

- `strftime()` : 날짜/시간 → 문자열

```
from datetime import datetime

now = datetime.now()
print(now)
# 2025-10-03 14:22:15.123456

# 원하는 형식으로 문자열 변환
print(now.strftime("%Y-%m-%d %H:%M:%S"))
# '2025-10-03 14:22:15'
```

- `strptime()` : 문자열 → 날짜/시간

```
from datetime import datetime

date_str = "2025-10-03 14:22:15"

dt = datetime.strptime(date_str, "%Y-%m-%d %H:%M:%S")
print(dt)
# 2025-10-03 14:22:15
print(type(dt))
# <class 'datetime.datetime'>
```

- `datetime` 포맷 코드 표

Directive	Meaning	Example

Directive	Meaning	Example
%a	Weekday as locale’s abbreviated name	Sun, Mon, …, Sat (en_US); So, Mo, …, Sa (de_DE)
%A	Weekday as locale’s full name	Sunday, Monday, …, Saturday (en_US); Sonntag, Montag, …, Samstag (de_DE)
%w	Weekday as a decimal number (0=Sunday, 6=Saturday)	0, 1, …, 6
%d	Day of the month (zero-padded)	01, 02, …, 31
%b	Month as locale’s abbreviated name	Jan, Feb, …, Dec (en_US); Jan, Feb, …, Dez (de_DE)
%B	Month as locale’s full name	January, February, …, December (en_US); Januar, Februar, …, Dezember (de_DE)
%m	Month as a zero-padded decimal number	01, 02, …, 12
%y	Year without century (zero-padded)	00, 01, …, 99
%Y	Year with century	0001, 0002, …, 2013, …, 9999
%H	Hour (24-hour clock, zero-padded)	00, 01, …, 23
%I	Hour (12-hour clock, zero-padded)	01, 02, …, 12
%p	AM/PM (locale)	AM, PM (en_US); am, pm (de_DE)
%M	Minute (zero-padded)	00, 01, …, 59
%S	Second (zero-padded)	00, 01, …, 59
%f	Microsecond (6 digits, zero-padded)	000000, …, 999999
%z	UTC offset	+0000, -0400, +1030, +063415, -030712.345216
%Z	Time zone name	UTC, GMT, (empty if naive)
%j	Day of the year (zero-padded)	001, 002, …, 366
%U	Week number (Sunday first, zero-padded)	00, 01, …, 53
%W	Week number (Monday first, zero-padded)	00, 01, …, 53
%c	Locale’s date and time	Tue Aug 16 21:30:00 1988 (en_US)
%x	Locale’s date representation	08/16/88 (en_US); 16.08.1988 (de_DE)

Directive	Meaning	Example
%X	Locale's time representation	21:30:00 (en_US)
%%	Literal '%' character	%

- `Series.dt.날짜요소` : 날짜 타입의 변수로부터 날짜 요소를 뽑아낼 수 있다.

  📌 Series.dt 접근자 기본 메서드

메서드	내용
`df['date'].dt.date`	YYYY-MM-DD (문자)
`df['date'].dt.year`	연 (4자리 숫자)
`df['date'].dt.month`	월 (숫자)
`df['date'].dt.month_name()`	월 (문자)
`df['date'].dt.day`	일 (숫자)
`df['date'].dt.time`	HH:MM:SS (문자)
`df['date'].dt.hour`	시 (숫자)
`df['date'].dt.minute`	분 (숫자)
`df['date'].dt.second`	초 (숫자)
`df['date'].dt.quarter`	분기 (숫자)
`df['date'].dt.day_name()`	요일 이름 (문자)
`df['date'].dt.weekday`	요일 숫자 (0=월, 6=일)
`df['date'].dt.dayofyear`	연 기준 몇 일째 (숫자)
`df['date'].dt.days_in_month`	월 일수 (=daysinmonth) (숫자)

  📌 Series.dt vs Series.dt.isocalendar()

구분	`air['Date'].dt`	`air['Date'].dt.isocalendar()`
일	`day`	X
월	`month`	X
연	`year`	`year`
주차	X	`week`
요일	`weekday` : 0~6 (월~일)	`day` : 1~7 (월~일)

시계열의 주요 특성

1. 정상성 (Stationarity)

- 시계열 데이터의 통계적 특성(평균, 분산, 공분산 등)이 **시간의 흐름에 따라 변하지 않고 일정**하게 유지되는 성질
- 많은 시계열 모델(e.g. ARIMA)은 데이터가 정상성을 만족한다고 가정하므로, 시계열 분석에서 매우 중요한 개념
- **정상성의 종류**:
    - **강한 정상성 (Strict Stationarity)**: 시계열의 모든 통계적 특성이 시간에 대해 불변. (현실적으로 충족하기 어려움)
    - **약한 정상성 (Weak Stationarity)**: 시계열의 평균과 분산이 시간에 따라 일정하고, 자기 공분산은 시차(lag)에만 의존. (일반적으로 '정상성'이라고 하면 약한 정상성을 의미)
- **비정상 시계열 (Non-stationary Time Series)**:
    - 시간의 흐름에 따라 평균이나 분산이 변하는 시계열.
    - 대부분의 실제 시계열 데이터는 추세(Trend)나 계절성(Seasonality)을 포함하는 비정상 시계열.
- **정상성으로 변환하는 방법**:
    - **차분 (Differencing)**: 현재 시점의 데이터에서 이전 시점의 데이터를 빼는 방법. 추세를 제거하는 데 효과적입니다.
    - **로그 변환 (Log Transformation)**: 분산이 시간에 따라 증가하는 경우, 로그를 취하여 분산을 안정화시킬 수 있습니다.

2. 자기상관 (Autocorrelation)

- 시계열 데이터에서 현재 시점의 값과 과거 시점의 값 사이의 상관관계를 의미
- '오늘의 주가'가 '어제의 주가'와 얼마나 관련이 있는지를 나타냄
- **측정**: 자기상관 함수(ACF)를 통해 측정하며, 시차(lag)에 따른 상관계수를 계산합니다.