과제 1

2018111373 최인렬

1번

문제 : $\beta_1 = 2$, $\beta_2 = 2$, $s = 12$일 때, 아래의 주어진 식을 참고해서 다음 문제를 풀이하시오.

$$A\ \sin\left(\frac{2\pi}{s}t+\phi\right) = A\ \sin\left(\frac{2\pi}{s}t\right)\cos(\phi) + A\ \cos\left(\frac{2\pi}{s}t\right)\sin(\phi)$$
$$= \beta_1\ \sin\left(\frac{2\pi}{s}t\right) + \beta_2\ \cos\left(\frac{2\pi}{s}t\right)$$

1-1번. 그래프 그리기

$y = \beta_1\ \sin\left(\frac{2\pi}{s}t\right)$ 와 $y = \beta_2\ \cos\left(\frac{2\pi}{s}t\right)$ 의 그래프 겹쳐 그리기

1. 먼저, 시간대(변수 t)의 범위를 정한다. 여기서는 시작 값을 $0.4 \times \pi$, 끝 값을 200으로 정했다.

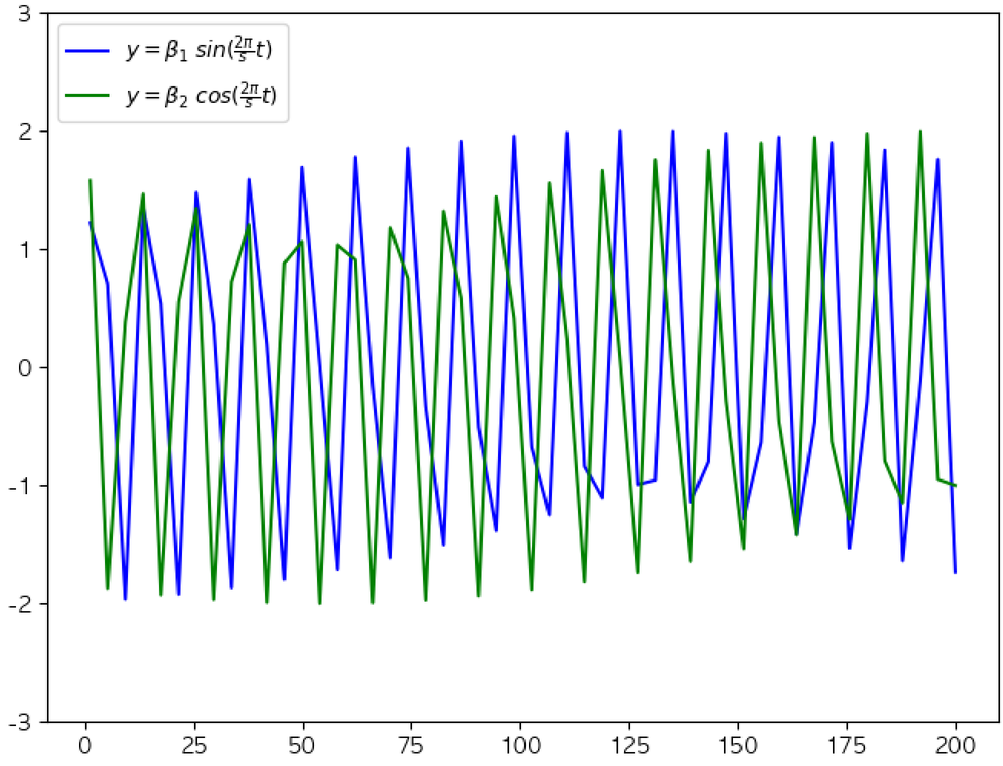
2.
$\beta_1 = 2$, $\beta_2 = 2$ 이므로 이에 맞게 그래프를 그린다.

3. 또한, $\beta_1 = 2$, $\beta_2 = 2$, $s = 12$ 도 변수로 넣는다.

파이썬 코드

```
# 파이썬 사전 설정
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

```
beta1 = 2; beta2 = 2; s = 12

t = np.linspace(0.4 * np.pi, 200)
y1 = beta1 * np.sin((2 * np.pi * t) / s)
y2 = beta2 * np.cos((2 * np.pi * t) / s)

label_y1 = r'$y = \beta_1~sin(\frac{2\pi}{s}t)$'
label_y2 = r'$y = \beta_2~cos(\frac{2\pi}{s}t)$'

plt.figure(figsize=(8, 6))
plt.plot(t, y1, label = label_y1, color='blue')
plt.plot(t, y2, label = label_y2, color='green')
plt.legend(loc='upper left', fontsize = 10)
plt.ylim(-3, 3)
plt.show()
```



1-2번. A와 ϕ 구하기

문제 : 위의 주어진 식을 참고해서 A와 ϕ를 구해보시오

삼각함수의 덧셈정리를 이용해 우리는 A와 ϕ를 구해볼 수 있다.

$$\sin(\alpha + \beta) = \sin(\alpha)\ \cos(\beta)\ +\ \cos(\alpha)\ \sin(\beta)$$
$$\sin(\omega t + \phi) = \sin(\omega t)\ \cos(\phi)\ +\ \cos(\omega t)\ \sin(\phi)$$
$$A\sin(\omega t + \phi) = A\ \sin(\omega t)\ \cos(\phi)\ +\ A\ \cos(\omega t)\ \sin(\phi)$$

이 공식을 다시 이용해보면...

$$A\sin(\omega t + \phi) = A\ \cos(\phi)\ \sin(\omega t)\ +\ A\ sin(\phi)\ \cos(\omega t)$$
$$= \beta_1\ \sin\left(\frac{2\pi}{s}t\right) + \beta_2\ \cos\left(\frac{2\pi}{s}t\right)$$

즉, 우리는 $\beta_1 = A\ \cos(\phi)$ 이고, $\beta_2 = A\ \sin(\phi)$ 임을 알 수 있다. 이때, β_2를 β_1으로 나눠보면 다음과 같다.

$$\frac{A\ \sin(\phi)}{A\ \cos(\phi)} = \tan(\phi) = \frac{\beta_2}{\beta_1}$$

따라서, ϕ를 구해보면 다음과 같다.

$$\phi = tan^{-1}\left(\frac{\beta_2}{\beta_1}\right)$$

파이썬을 이용해서 실제 ϕ를 구해보자.

```
A = np.sqrt(beta1 * 2 + beta2 * 2)
phi = np.arctan(beta2 / beta1)

print(f"A = {A}\nphi = {phi}")
```

결과는 다음과 같다.

```
A = 2.8284271247461903
phi = 0.7853981633974483
```

따라서 $A \approx 2.828$ 이고, $\phi \approx 0.785$ 이다.

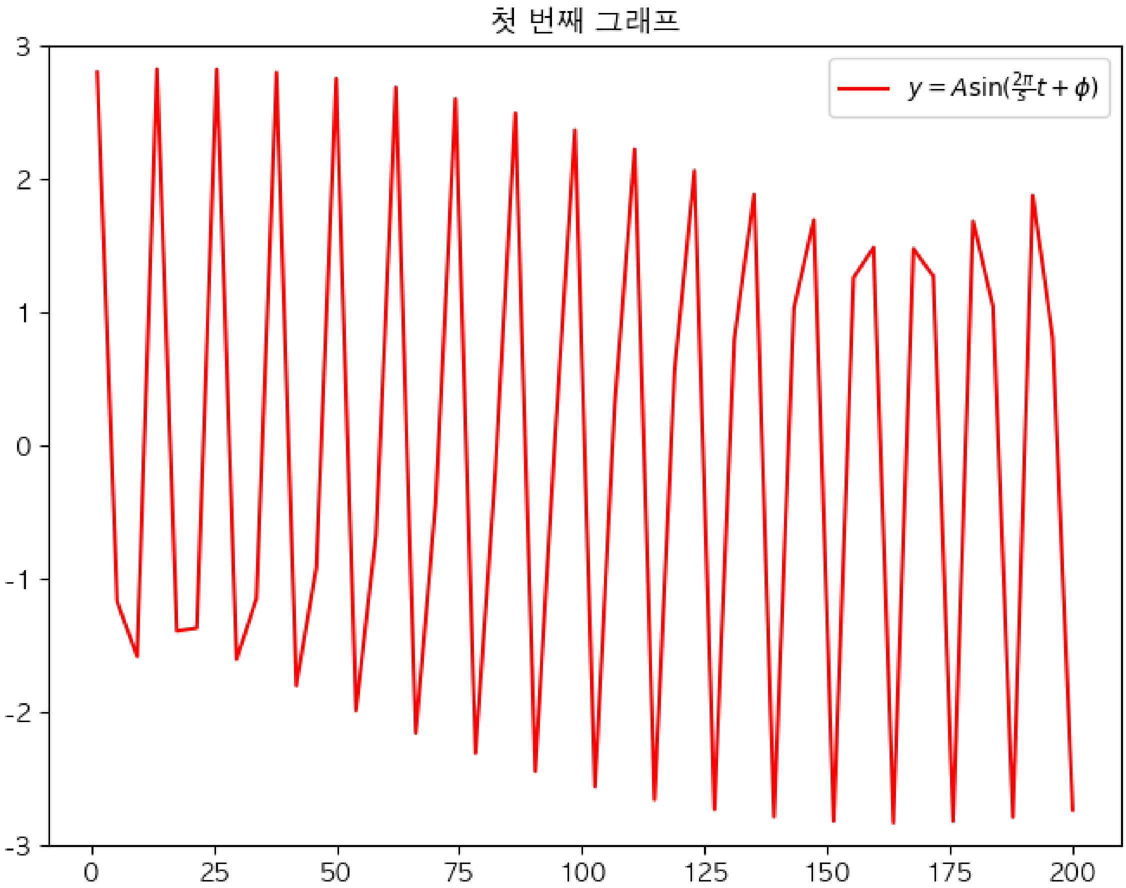
1-3. 그래프 그리고 확인

문제 : $y = A \sin\left(\frac{2\pi}{s}t + \phi\right)$ 그래프를 그린 후,

$y = \beta_1 \sin\left(\frac{2\pi}{s}t\right) + \beta_2 \cos\left(\frac{2\pi}{s}t\right)$ 그래프가 이와 동일한지 확인해보시오.

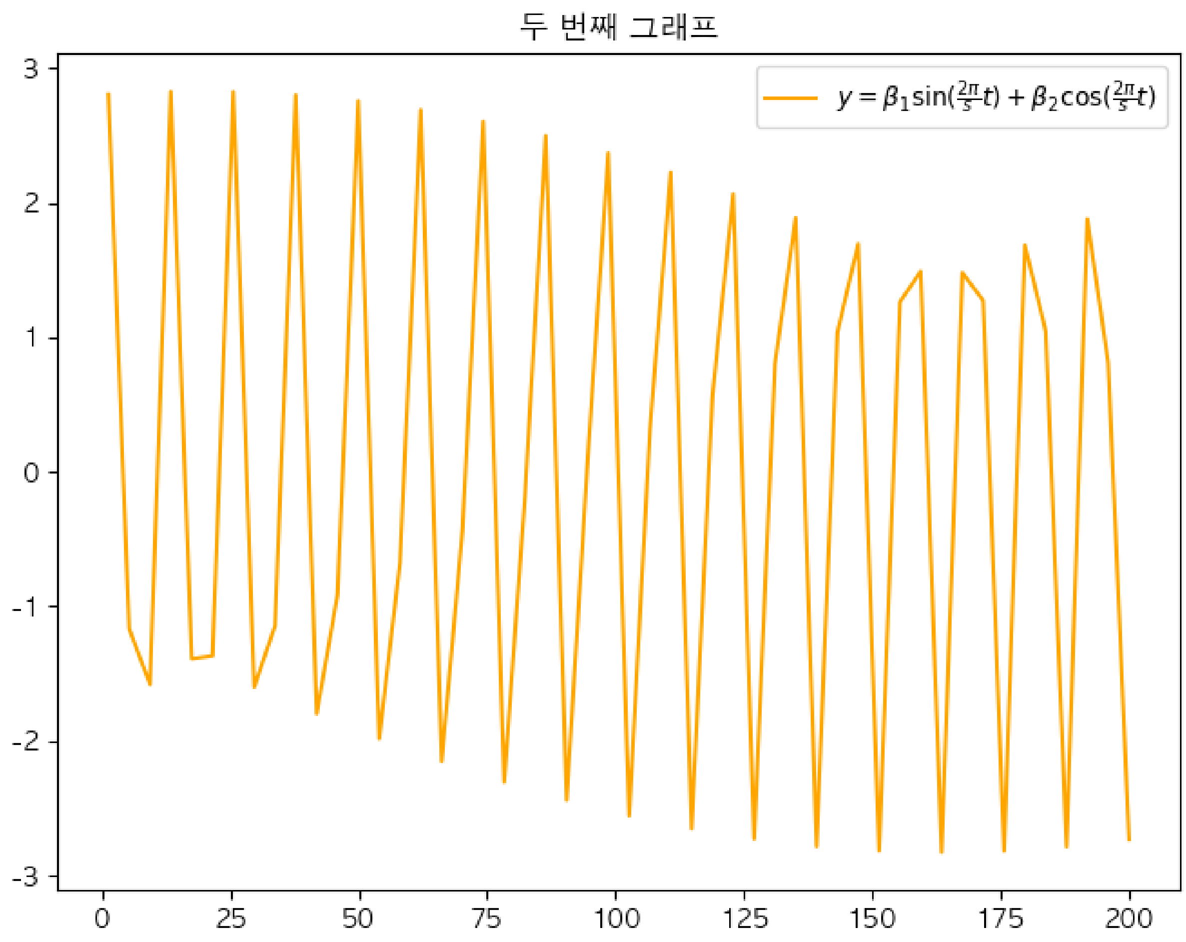
먼저, $y = A \sin\left(\frac{2\pi}{s}t + \phi\right)$ 그래프를 그린다.

```
t = np.linspace(0.4 * np.pi, 200)
y3 = A * np.sin(((2 * np.pi * t) / s) + phi)
label_y3 = r'$y = A \sin (\frac{2\pi}{s}t + \phi )$'

plt.figure(figsize=(8, 6))
plt.title("첫 번째 그래프")
plt.plot(t, y3, label = label_y3, color='red')
plt.legend()
plt.ylim(-3, 3)
plt.show()
```

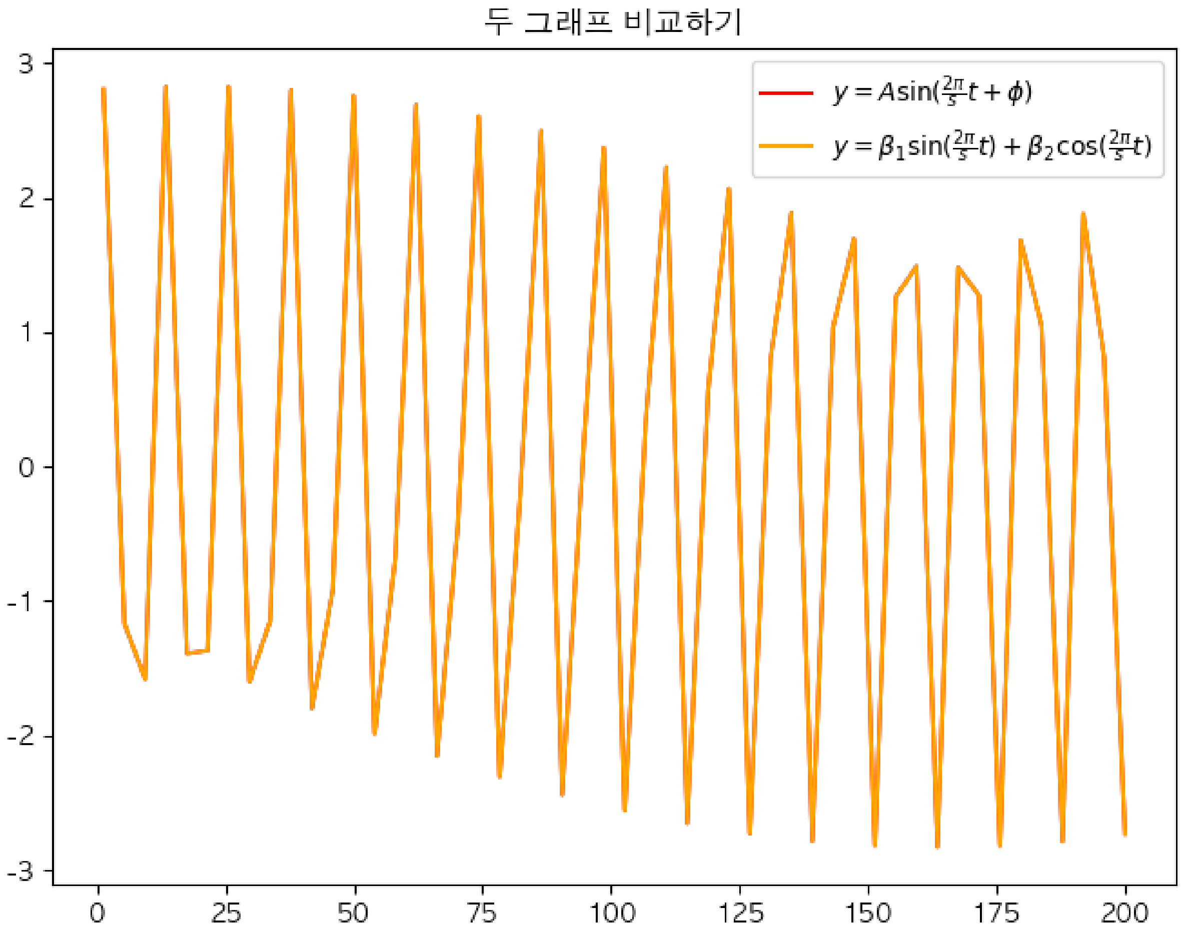


또, $y = \beta_1 \sin\left(\frac{2\pi}{s}t\right) + \beta_2 \cos\left(\frac{2\pi}{s}t\right)$ 그래프를 그려보자

```
y4 = y1 + y2
label_y4 = r'$y = \beta_1 \sin (\frac{2\pi}{s}t ) + \beta_2 \

plt.figure(figsize=(8, 6))
plt.title("두 번째 그래프")
plt.plot(t, y4, label = label_y4, color='orange')
plt.legend()
plt.show()
```



인제 둘을 겹쳐서 그려보자.



빨간색과 오랜지색이 섞여나오는 것을 볼 때, 두 그래프의 차이가 없음을 알 수 있다.

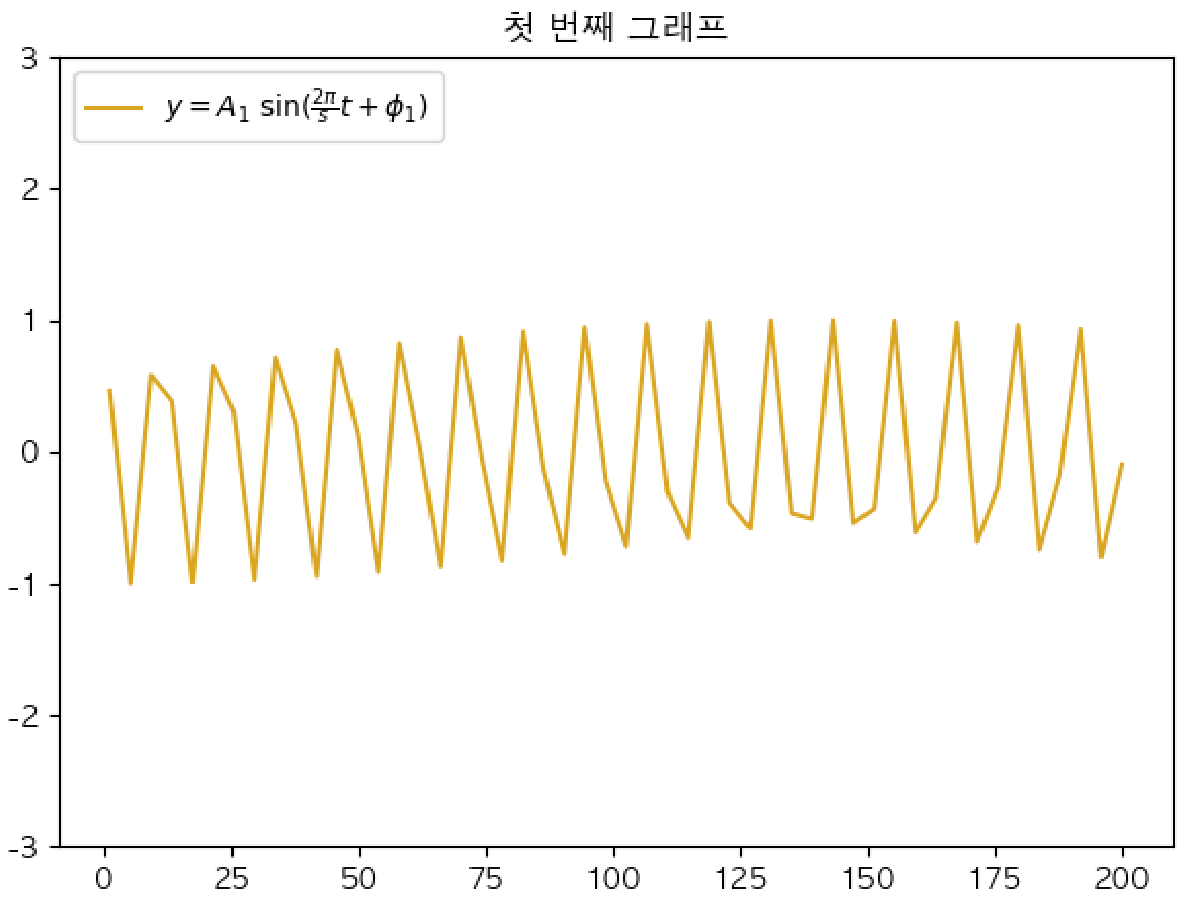
2번

문제 : $A_1 = 1, \phi_1 = 2$ 이고, 주기가 12인 $A_1 \sin\left(\frac{2\pi}{s}t + \phi_1\right)$ 그래프와 $A_2 = 2,$ $\phi_2 = 1$이고 주기가 6인 $A_2 \sin\left(\frac{2\pi}{s}t + \phi_2\right)$$ 그래프를 각각 그린 후, 두 사인 함수를 더한 그래프를 그리시오.

주어진 그래프 그리기

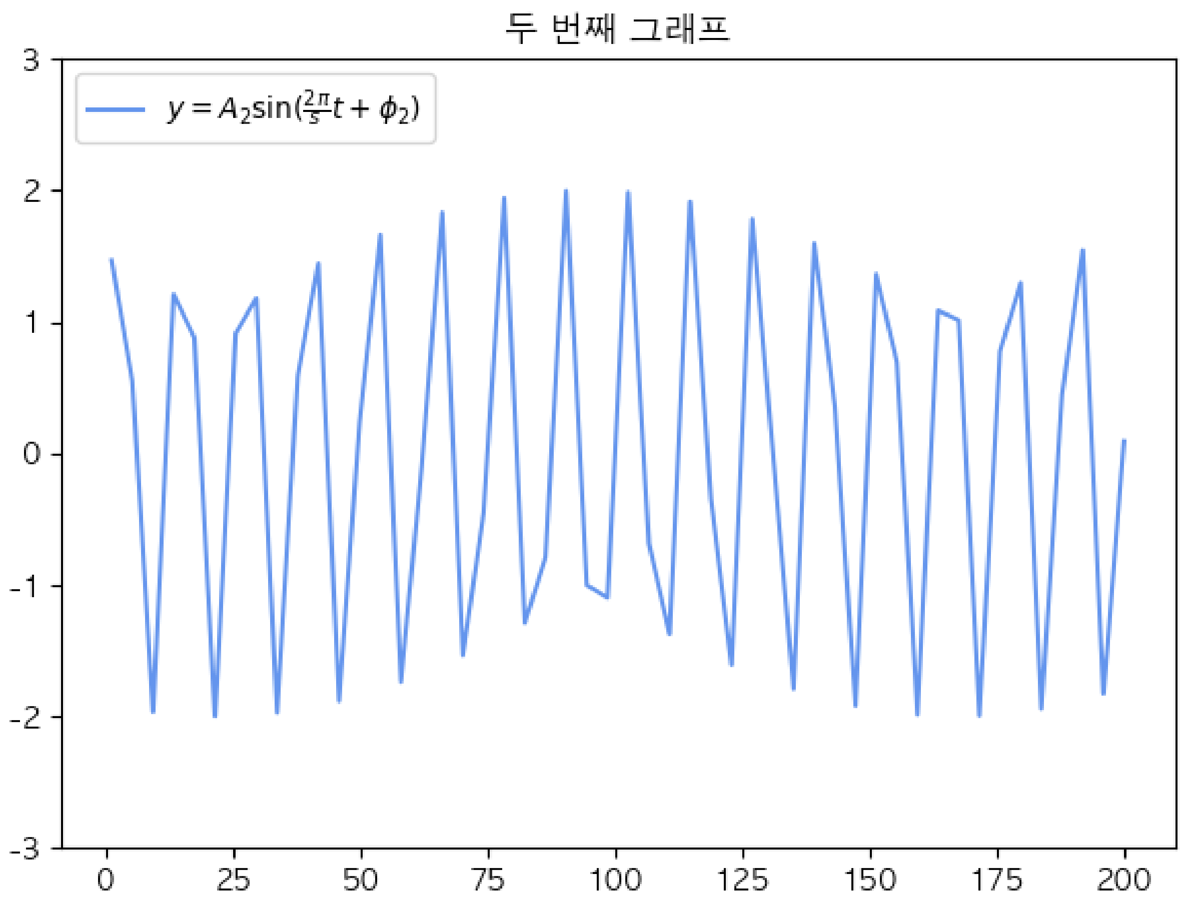
- 시간대 t는 1번에서 쓴 것을 그대로 쓴다.
- $A_1 = 1$, $\phi_1 = 2$ 이고 주기 $s_1 = 12$ 인 $A_1 \sin\left(\frac{2\pi}{s}t + \phi_1\right)$ 그래프를 그린다.
- $A_2 = 2$, $\phi_2 = 1$ 이고 주기 $s_2 = 6$ 인 $A_2 \sin\left(\frac{2\pi}{s}t + \phi_2\right)$ 그래프를 그린다.

```
a_1 = 1; phi_1 = 2; s1 = 12
y_1 = a_1 * np.sin((2 * np.pi * t / s1) + phi_1)
label_y_1 = r'$y = A_1~\sin (\frac{2\pi}{s} t + \phi_1 )$'

plt.figure(figsize=(7, 5))
plt.title("첫 번째 그래프")
plt.plot(t, y_1, label = label_y_1, color='goldenrod')
plt.ylim(-3, 3)
plt.legend(loc='upper left')
plt.show()
```

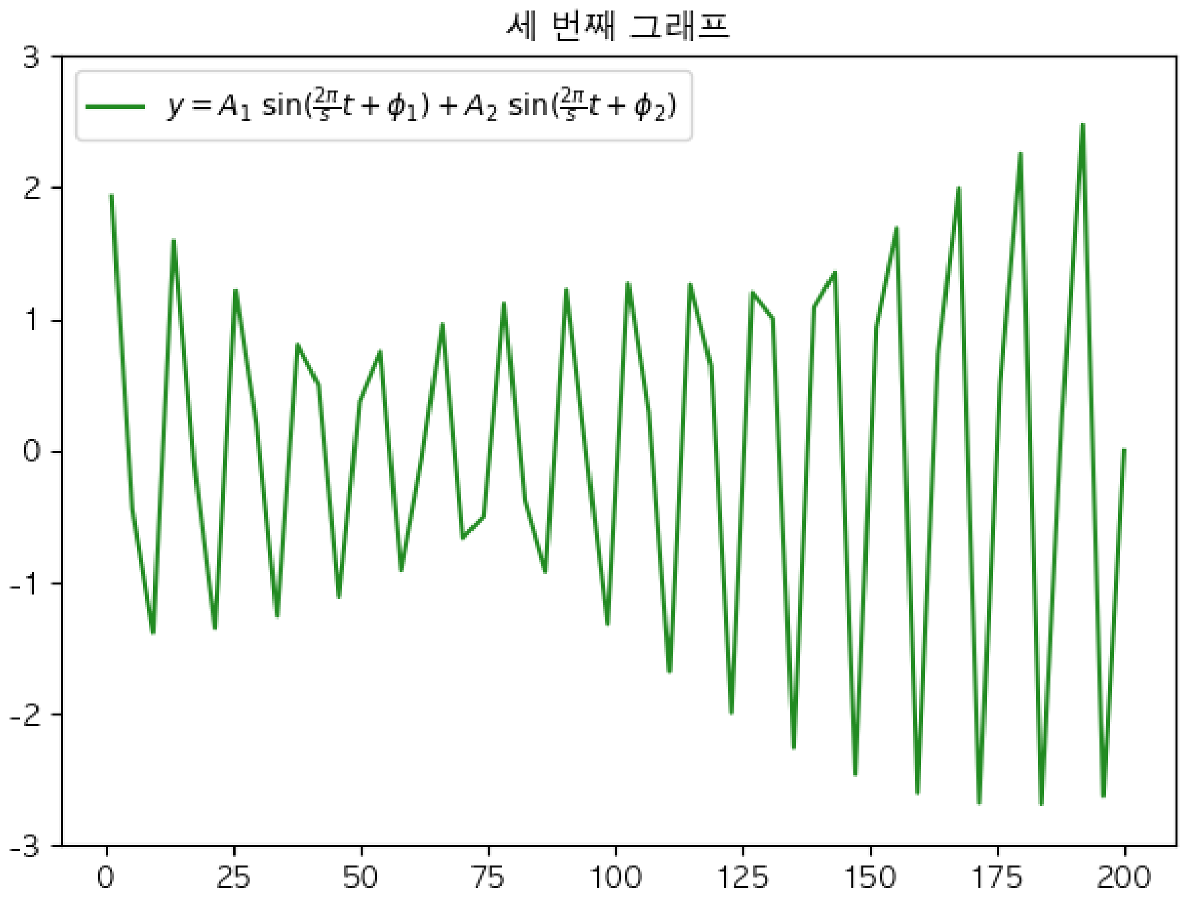



```
a_2 = 2; phi_2 = 1; s2 = 6
y_2 = a_2 * np.sin((2 * np.pi * t / s2) + phi_2)
label_y_2 = r'$y = A_2 \sin (\frac{2\pi}{s} t + \phi_2 )$'

plt.figure(figsize=(7, 5))
```

```
plt.title("두 번째 그래프")
plt.plot(t, y_2, label = label_y_2, color='cornflowerblue')
plt.ylim(-3, 3)
plt.legend(loc='upper left')
plt.show()
```




```
y_3 = y_1 + y_2
label_y_3 = r'$y = A_1~\sin (\frac{2\pi}{s} t + \phi_1 ) + A_

plt.figure(figsize=(7, 5))
plt.title("세 번째 그래프")
plt.plot(t, y_3, label = label_y_3, color='forestgreen')
plt.ylim(-3, 3)
plt.legend(loc='upper left')
plt.show()
```



3번

- 문제 : `Temperature.csv` 자료를 이용하여 서울시의 5년 간 월별 평균 기온을 분석하고자 한다. 주기를 최대 3개까지 사용하여 모형을 적합시켜 본 후 주기를 몇까지 사용하여 분석하는 것이 적절한지 설명하시오.

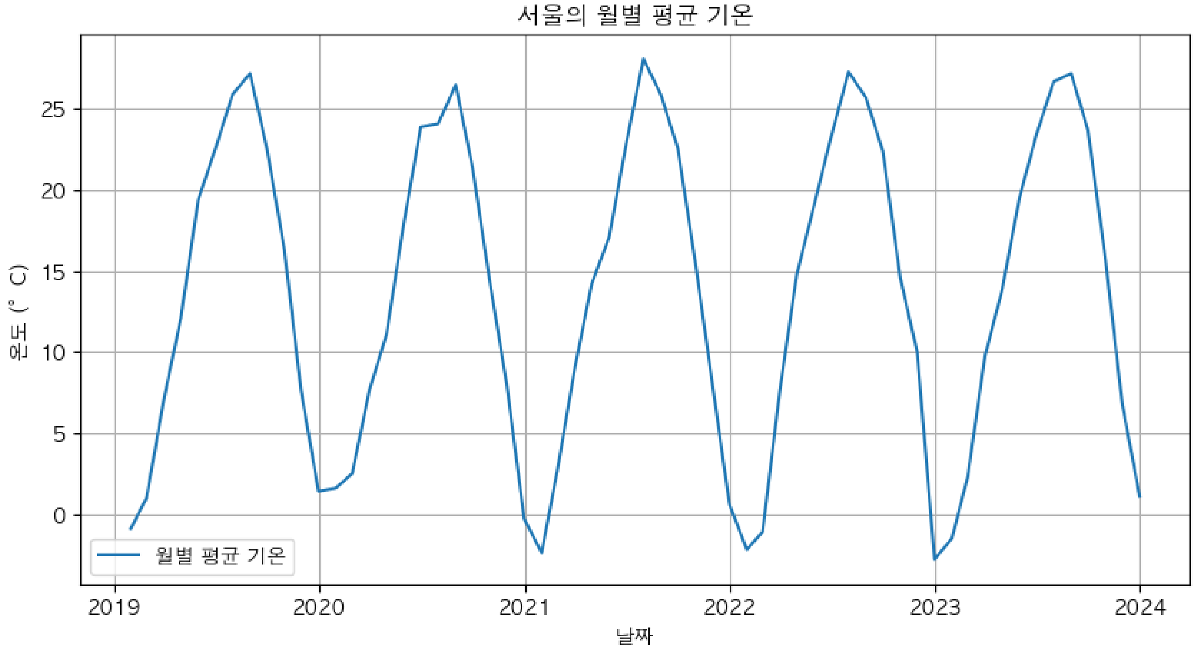
데이터 불러오기와 EDA

```
# 라이브러리 가지고 오기
from statsmodels.tsa.deterministic import DeterministicProces
```

```
# 데이터 불러오기
data = pd.read_csv('./Temperature.csv')

# 데이터 정리: 연도와 월을 합쳐서 datetime 형식의 인덱스를 만들고, 월별로
```

```
data['Date'] = pd.to_datetime(data[['Year', 'Month']].assign(
data.set_index('Date', inplace=True)

# 월별 기온 시계열
monthly_avg_temp = data['Temperature'].resample('ME').mean()
```

```
# 시계열 데이터 시각화
plt.figure(figsize=(10, 5))
plt.plot(monthly_avg_temp, label='월별 평균 기온')
plt.title('서울의 월별 평균 기온')
plt.xlabel('날짜')
plt.ylabel('온도 (°C)')
plt.legend()
plt.grid(True)
plt.show()
```



EDA 결과 및 분석

- sin 항은 2개 정도가 있으면 충분할 것 같음.
- 월별기온이 1년에 따라 반복되고 있음.

적합용 함수 & 그래픽 함수 만들기

```
# 푸리에 급수 기반의 모형 적합 함수 작성
def fit_fourier_series(series, p):
    # 시간 인덱스를 기반으로 푸리에 모형 생성
    dp = DeterministicProcess(
        index=series.index,     # 시간축 (x축)
        constant=True,          # 상수항 여부
        period = p,             # 주기 결정
        fourier = 2             # 사인 항의 갯수
    )

    # 최소제곱법을 사용해 모형 적합
    X = dp.in_sample()  # 훈련 데이터 생성
    y = series.values
    coef = np.linalg.lstsq(X, y, rcond=None)[0]

    # 예측값 계산
    fit = X @ coef
    return fit, coef
```

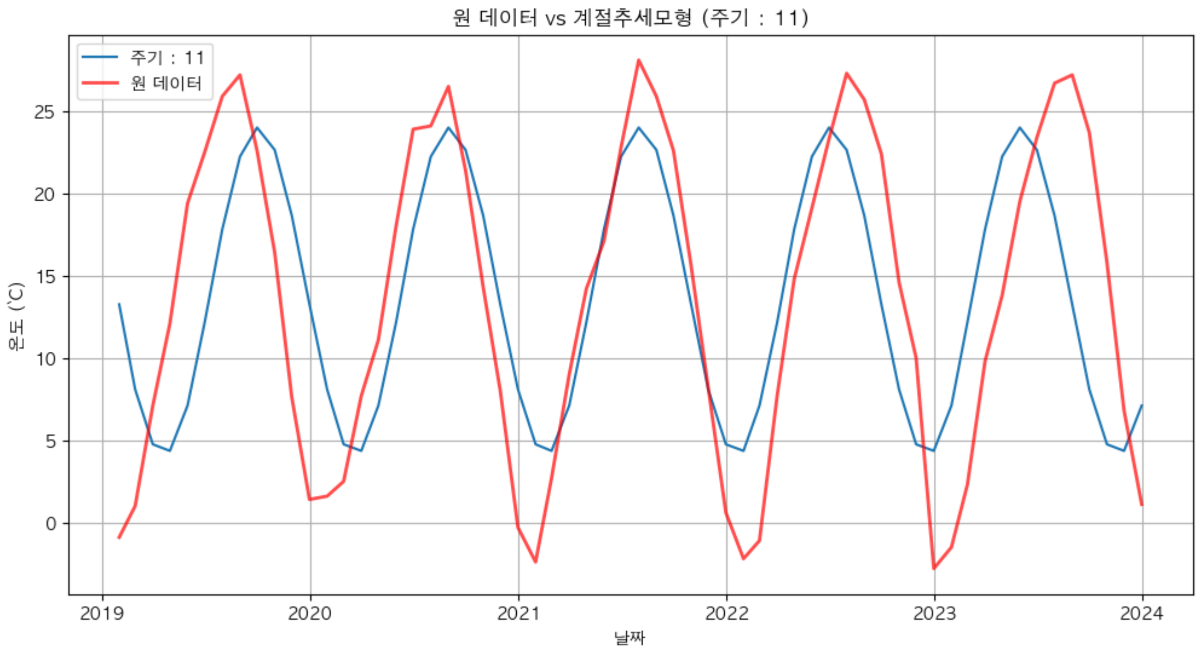
```
def draw_graph(fit, period):
    plt.plot(monthly_avg_temp.index, fit, label=f'주기 : {peri
    plt.plot(monthly_avg_temp, label='원 데이터', color='red',
    plt.title(f'원 데이터 vs 계절추세모형 (주기 : {period})')
    plt.xlabel('날짜')
    plt.ylabel('온도 (`C)')
    plt.legend()
    plt.grid(True)
    plt.show()
```

주기별 그래프 적합 결과

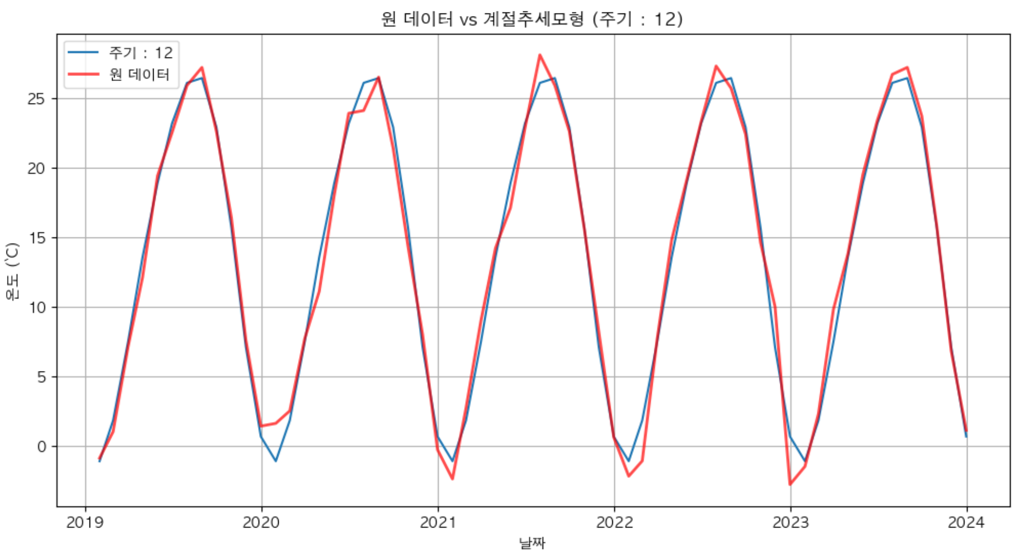
- 주기가 11, 12, 13일때 각각 테스트를 진행.

```
period = 11
plt.figure(figsize=(12, 6))
```

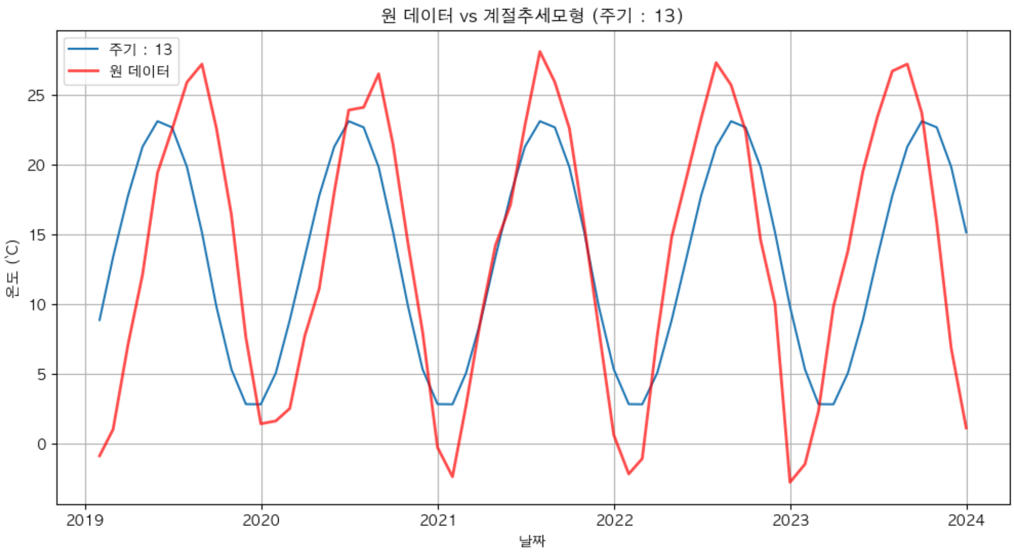
```
fit, coef = fit_fourier_series(monthly_avg_temp, period)
draw_graph(fit, period)
```




```
period = 12
plt.figure(figsize=(12, 6))
fit, coef = fit_fourier_series(monthly_avg_temp, period)
draw_graph(fit, period)
```



```
period = 13
plt.figure(figsize=(12, 6))
fit, coef = fit_fourier_series(monthly_avg_temp, period)
draw_graph(fit, period)
```



가장 적합한 것은 주기가 12일때 임을 알 수 있다.