

시각화라이브러리 : matplotlib

- 목표 : matplotlib의 기본 차트 그리기를 익힌다.
 - 코드의 기본 구조
 - 데이터 요건
 - 차트 꾸미기(이 부분은 가져다 사용할 수 있으면 됩니다.)

1.환경준비

(1) 라이브러리

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

# 시각화 라이브러리
import matplotlib.pyplot as plt
import seaborn as sns
```

(2) 데이터 불러오기

[airquality 데이터 셋 정보]

- Ozone: 오존 농도
- Solar.R: 태양복사량
- Wind: 풍속
- Temp: 기온
- Date : 연,월,일

```
In [2]: path = 'https://raw.githubusercontent.com/DA4BAM/dataset/master/airquality_simple2.csv'
data = pd.read_csv(path)
data['Date'] = pd.to_datetime(data['Date'])
data.dropna(axis = 0, inplace = True)
data.head()
```

```
Out[2]:
```

	Ozone	Solar.R	Wind	Temp	Date
0	34.0	286.0	8.6	78.0	1973-06-01
1	29.0	287.0	9.7	74.0	1973-06-02
2	18.0	242.0	16.1	67.0	1973-06-03
3	48.0	186.0	9.2	84.0	1973-06-04
4	49.0	220.0	8.6	85.0	1973-06-05

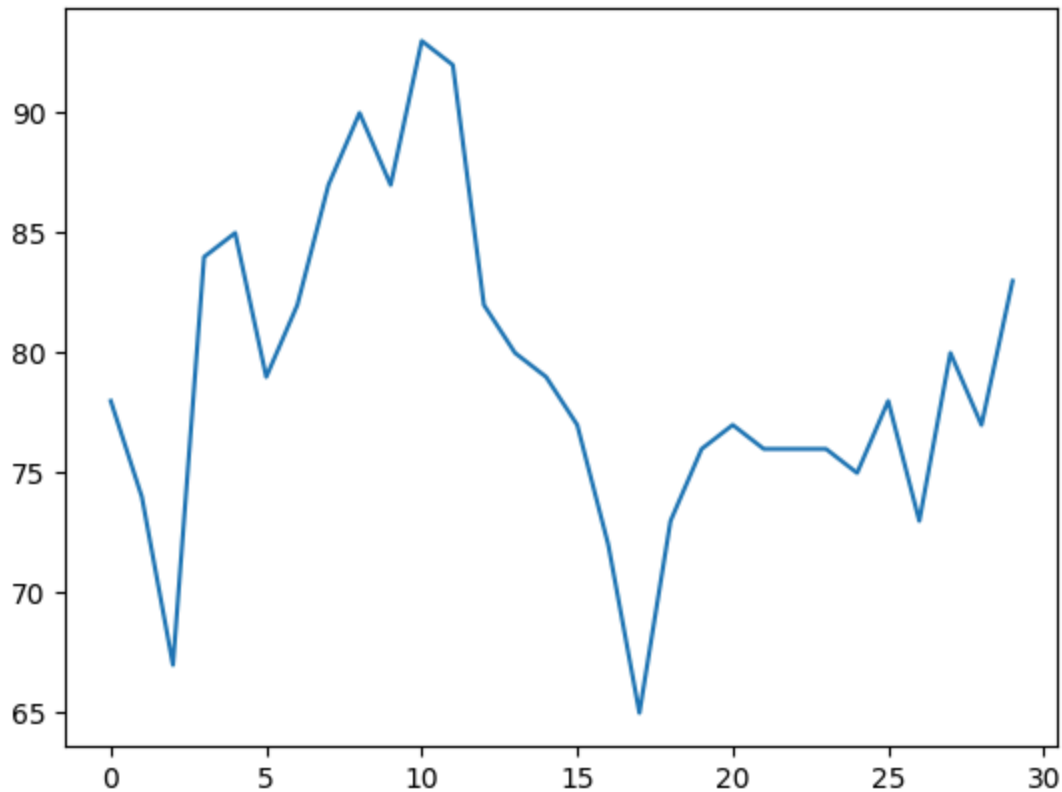
2.기본차트 그리기

(1) 차트 그리기

- plt.plot(1차원 값)
 - x축 : 인덱스
 - y축 : 1차원 값
 - 라인차트를 그려줍니다.

```
In [3]: # 차트 그리기
plt.plot(data['Temp'])

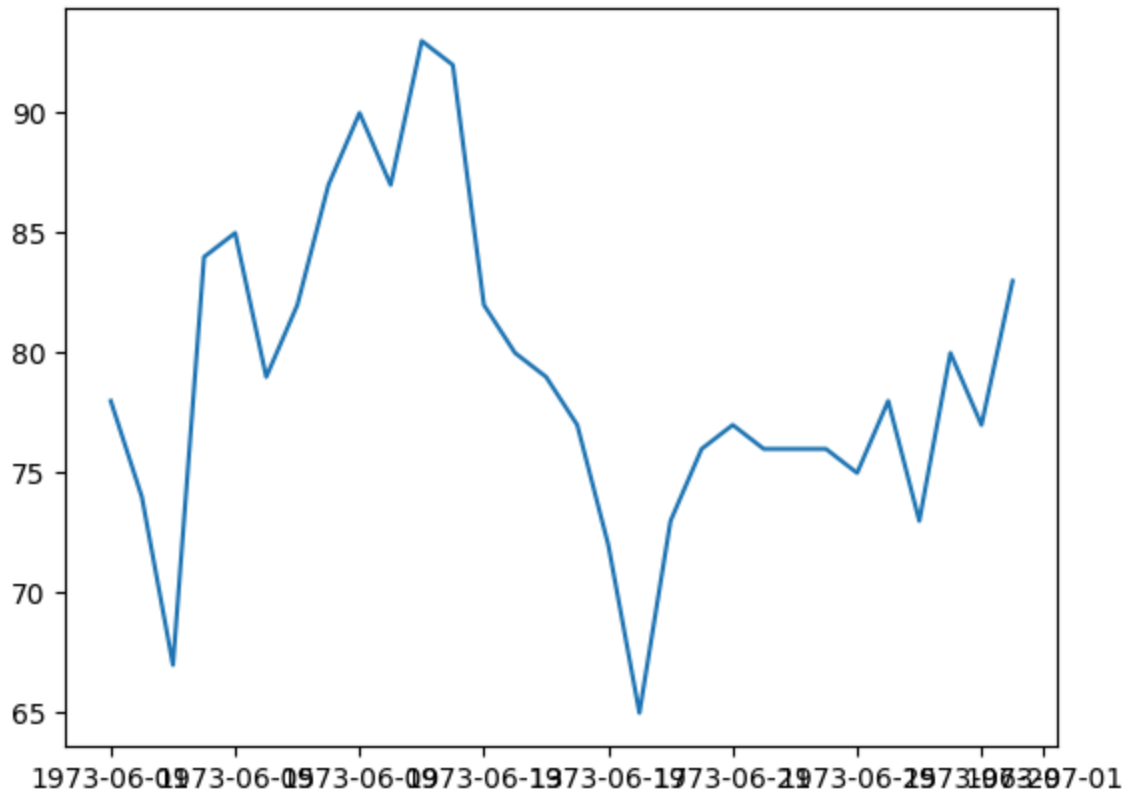
# 화면에 보여주기
plt.show()
```



(2) x축과 y축의 값 지정하기

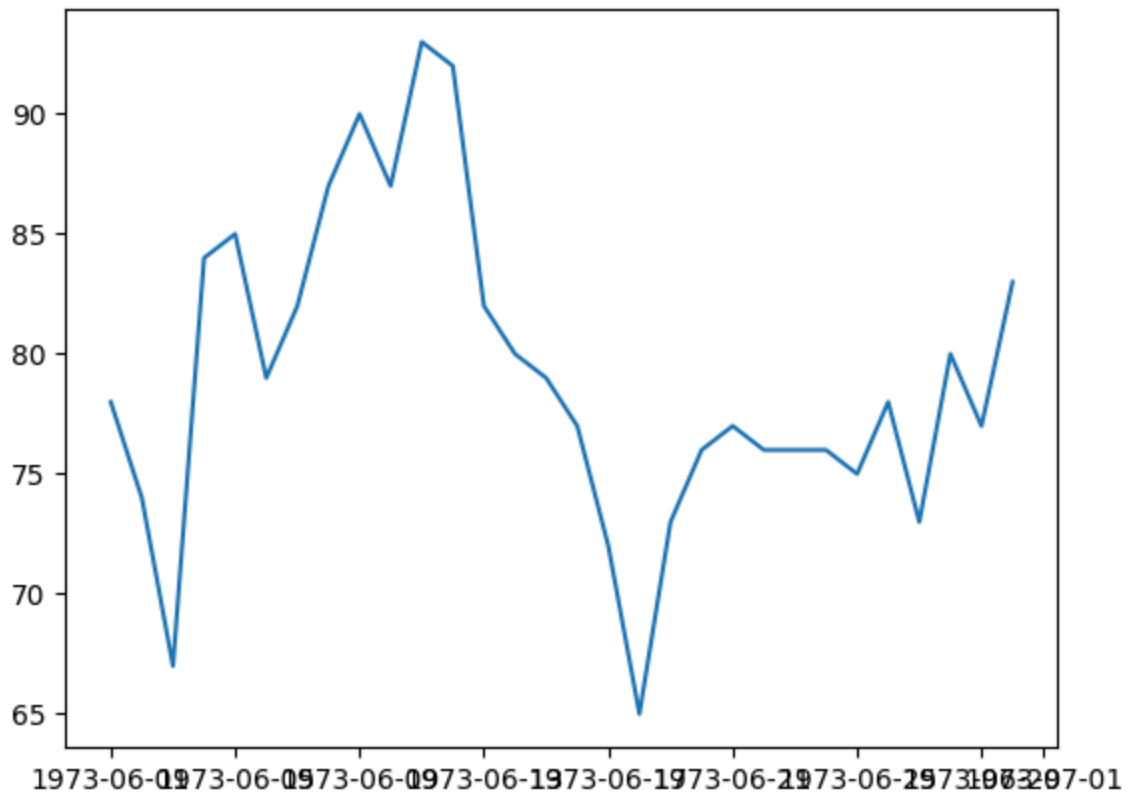
- plt.plot(x축, y축)

```
In [4]: # 타입1
plt.plot(data['Date'], data['Temp'])
plt.show()
```



- `plt.plot(x축, y축, data =)`

```
In [5]: # 타입2
plt.plot('Date', 'Temp', data = data)
plt.show()
```



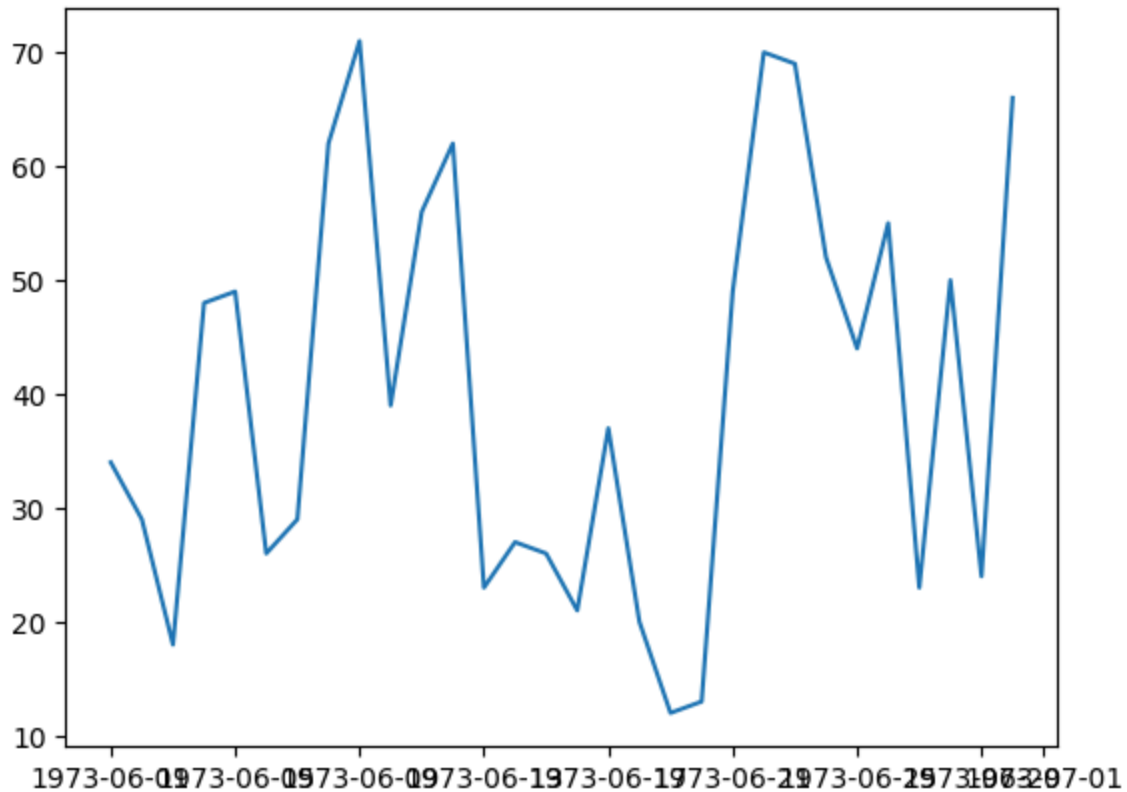
-연습문제-

아래 값으로 차트를 그리시오.

- x 축 : Date
- y 축 : Ozone

```
In [7]: # 날짜별 오존변화
plt.plot('Date', 'Ozone', data=data)

# 화면에 보여주기
plt.show()
```



3.차트 꾸미기

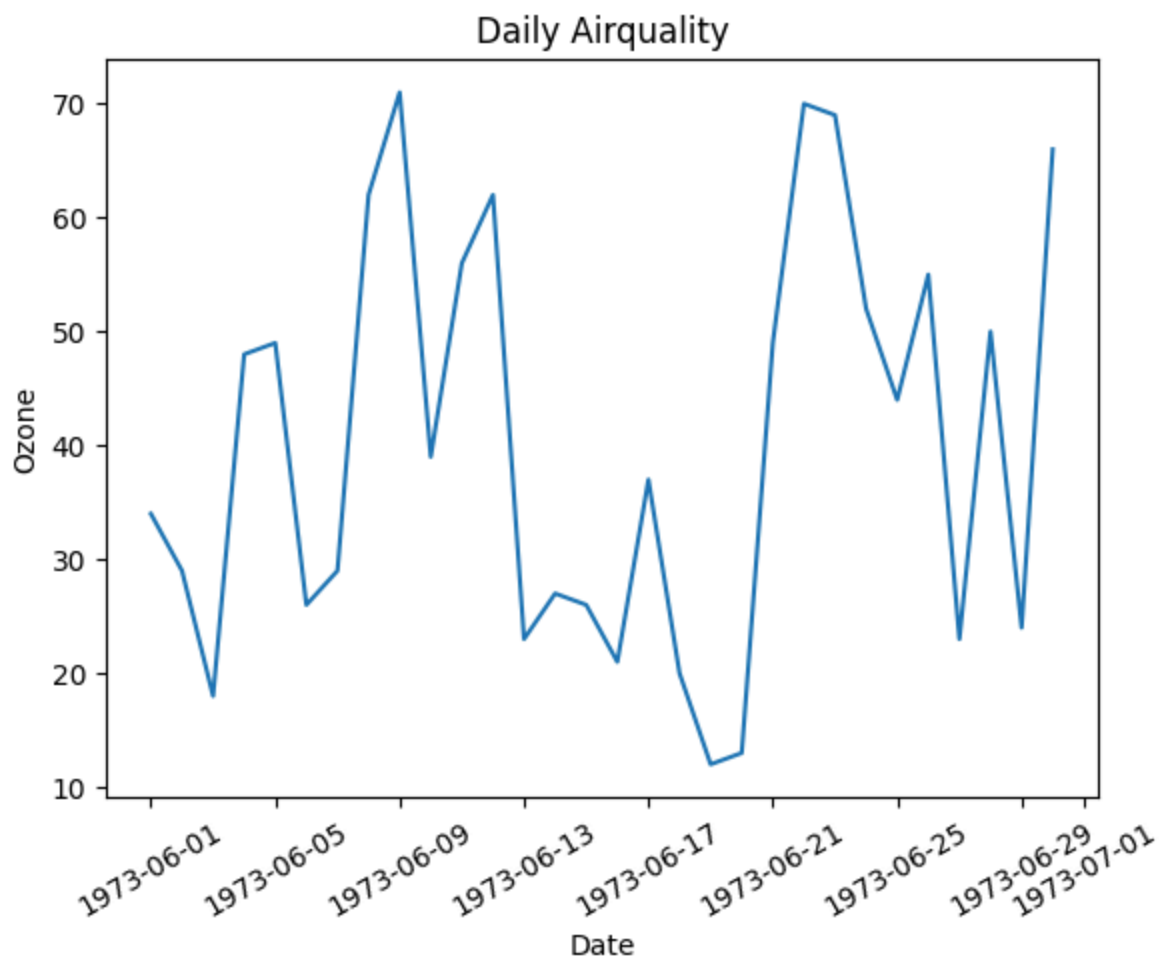
https://matplotlib.org/2.1.2/api/_as_gen/matplotlib.pyplot.plot.html

(1) x축, y축 이름, 타이틀 붙이기

```
In [8]: plt.plot(data['Date'], data['Ozone'])

plt.xticks(rotation = 30)      # x축 값 꾸미기 : 방향을 30도 틀어서
plt.xlabel('Date')             # x축 이름 지정
plt.ylabel('Ozone')            # y축 이름 지정
plt.title('Daily Airquality')   # 타이틀

plt.show()
```



(2) 라인 스타일 조정하기

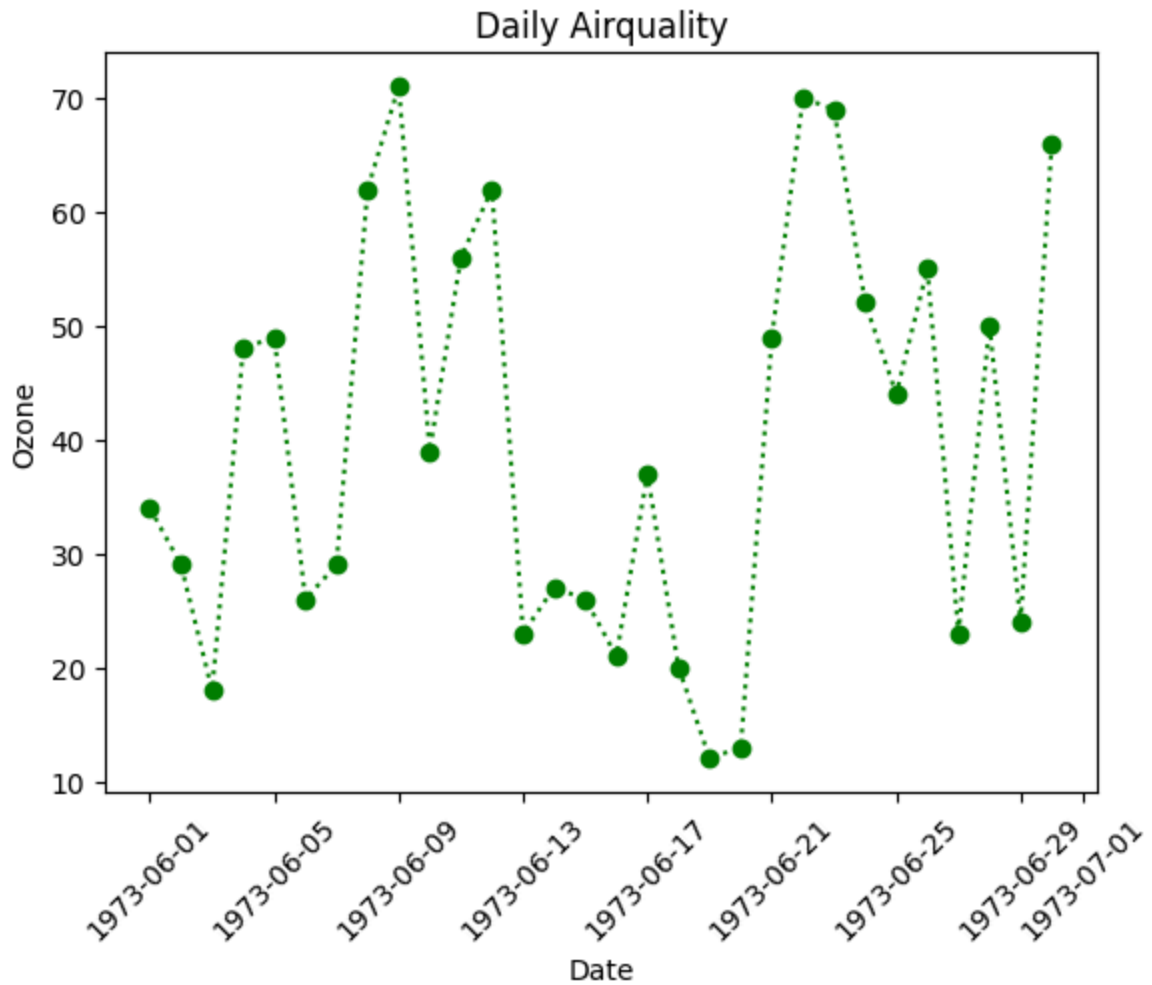
- color=
 - 'red','green','blue' ...
 - 혹은 'r', 'g', 'b', ...
 - https://matplotlib.org/stable/gallery/color/named_colors.html
- linestyle=
 - 'solid', 'dashed', 'dashdot', 'dotted'
 - 혹은 '-', '--', '-.', ':'
- marker=

marker	description
."	point
","	pixel
"o"	circle
"v"	triangle_down
"^"	triangle_up
"<"	triangle_left
">"	triangle_right

```
In [24]: plt.plot(data['Date'], data['Ozone']
              , color='green'          # 컬러
              , linestyle='dotted'     # 라인스타일
              , marker='o')           # 값 마커(모양)

plt.xlabel('Date')
plt.ylabel('Ozone')
plt.title('Daily Airquality')
plt.xticks(rotation=45)

plt.show()
```



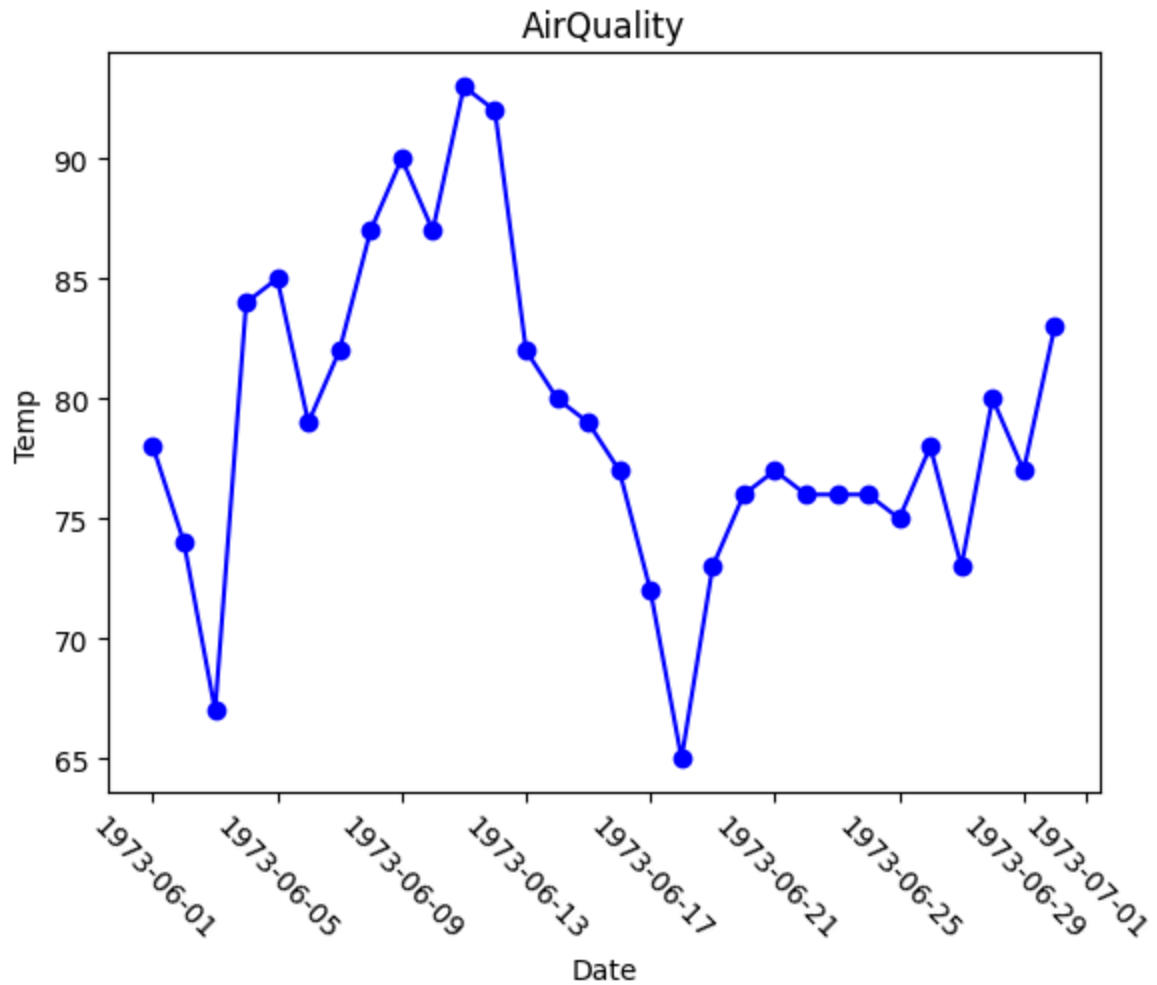
-연습문제-

[문1] 아래 조건에 맞게 그래프를 그려 봅시다.

- x 축 : Date
- y 축 : Temp
- x축 이름 : Date
- x축 label : 45도 돌려서.
- title : Airquality
- color : 여러분이 원하는대로!
- 선 스타일 : 여러분이 원하는대로!
- marker : 여러분이 원하는대로!

```
In [25]: plt.plot(data['Date'], data['Temp'], color='b', linestyle='solid', marker='o')
plt.xlabel('Date')
plt.ylabel('Temp')
plt.xticks(rotation=-45)
plt.title('AirQuality')

plt.show()
```

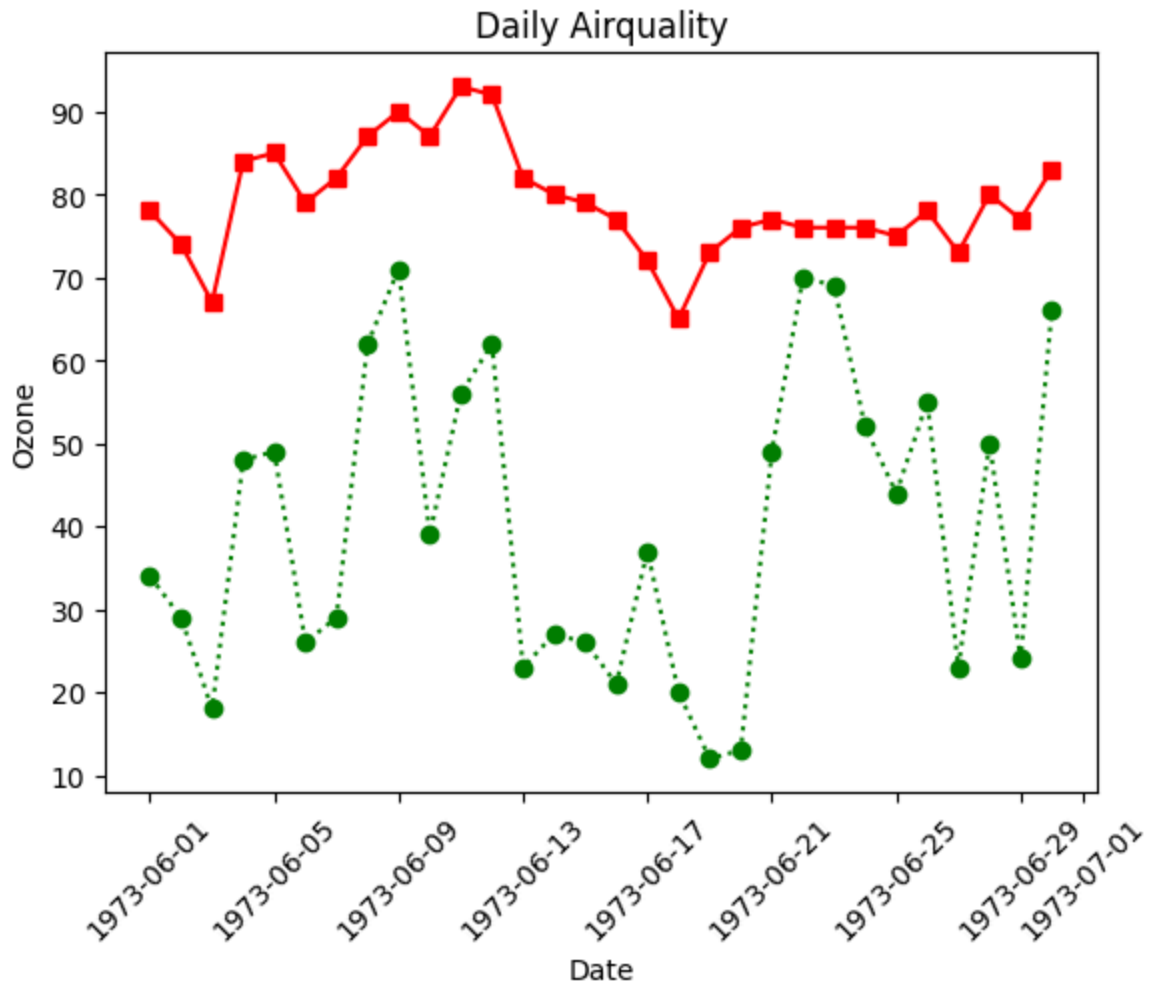


(3) 여러 그래프 겹쳐서 그리기

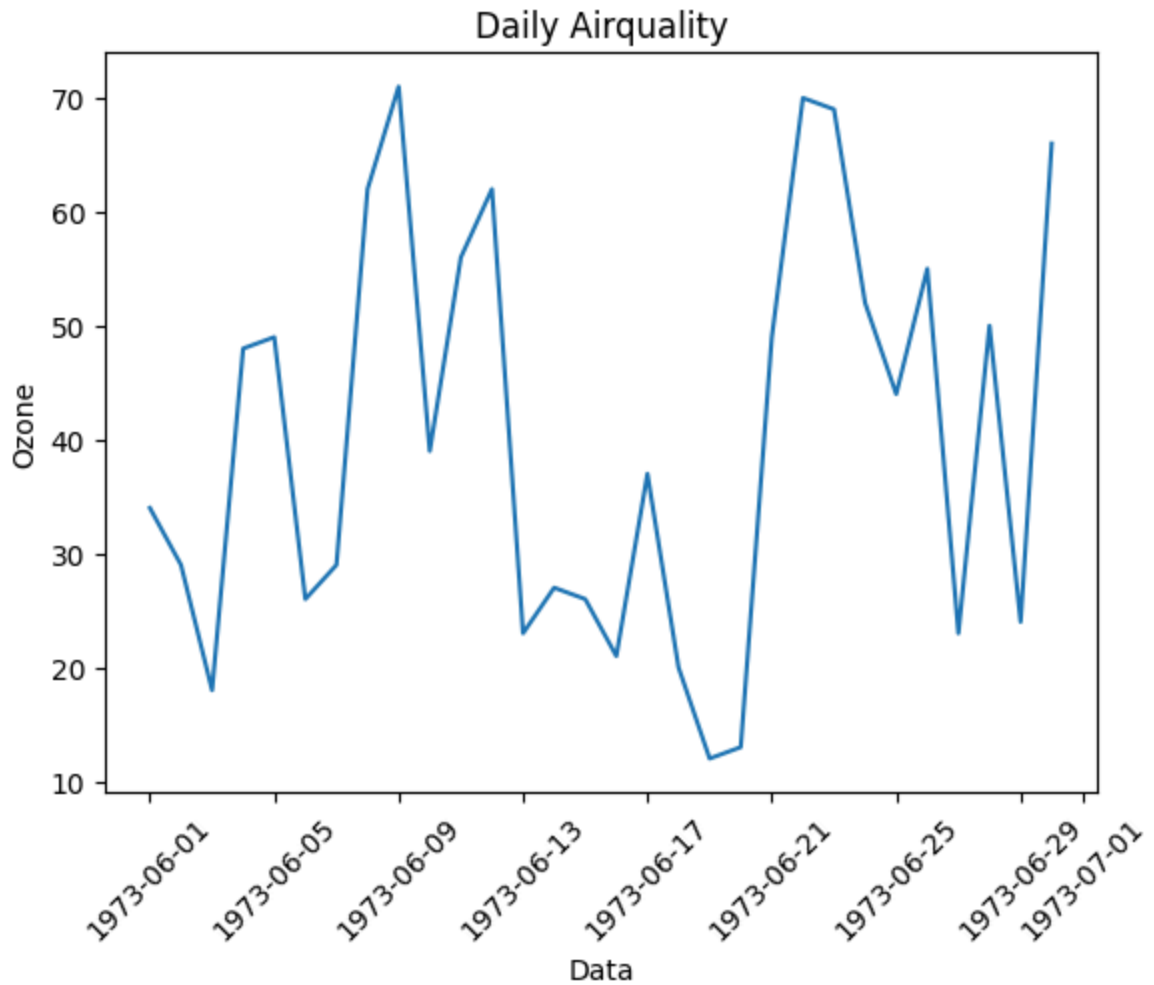
```
In [26]: # 첫번째 그래프
plt.plot(data['Date'], data['Ozone'], color='green', linestyle='dotted', marker='o')
# 두번째 그래프
plt.plot(data['Date'], data['Temp'], color='r', linestyle='-', marker='s')

plt.xlabel('Date')
plt.ylabel('Ozone')
plt.title('Daily Airquality')
plt.xticks(rotation=45)

# 위 그래프와 설정 한꺼번에 보여주기
plt.show()
```

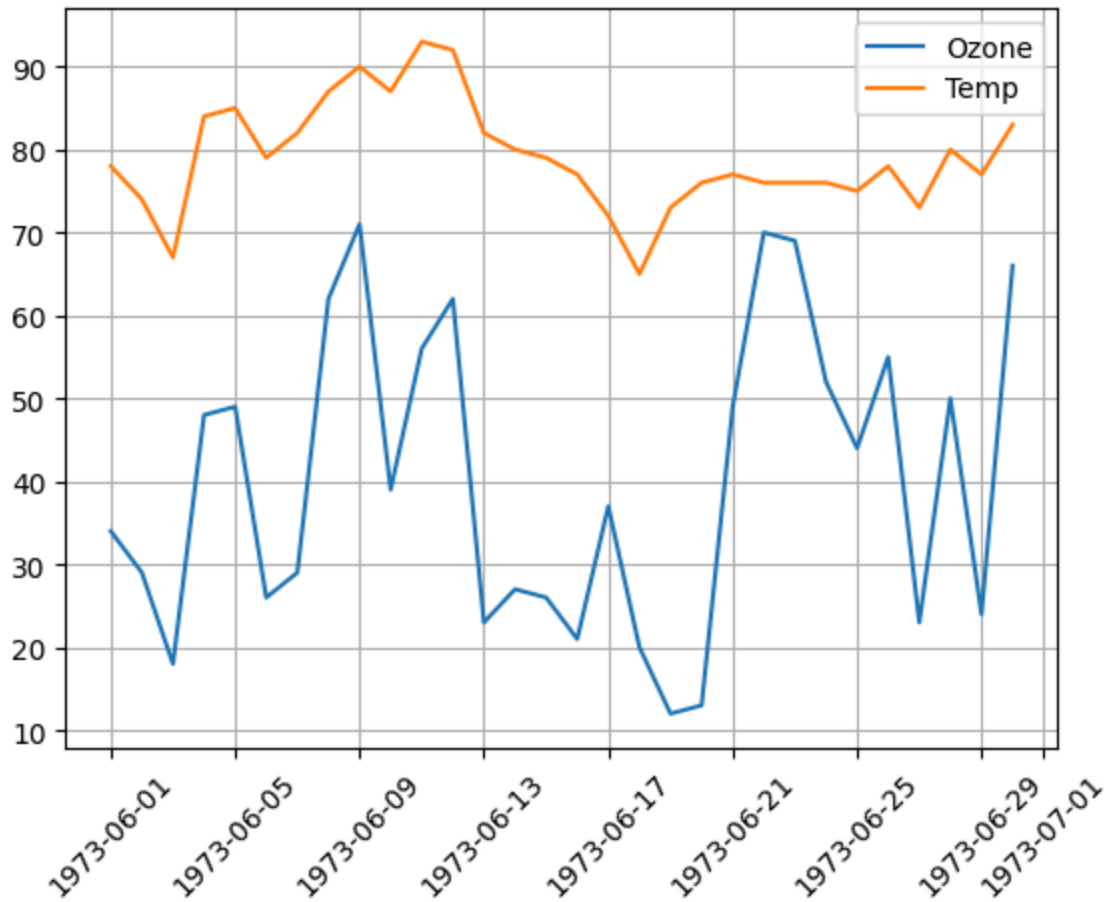
```
In [27]: plt.plot(data['Date'], data['Ozone'])
plt.xticks(rotation = 45)      # x축 값 꾸미기 : 방향을 45도 틀어서
plt.xlabel('Date')            # x축 이름 지정
plt.ylabel('Ozone')           # y축 이름 지정
plt.title('Daily Airquality')  # 타이틀
plt.show()
```



(4) 범례, 그리드 추가

```
In [31]: plt.plot(data['Date'], data['Ozone'], label = 'Ozone') # label = : 범례추가를 위한 레이블값
plt.plot(data['Date'], data['Temp'], label = 'Temp')

plt.legend(loc = 'upper right') # 레이블 표시하기. loc = : 위치
plt.grid()
plt.xticks(rotation=45)
plt.show()
```



- 연습문제 -

아래 조건에 맞게 그래프를 그려 봅시다.

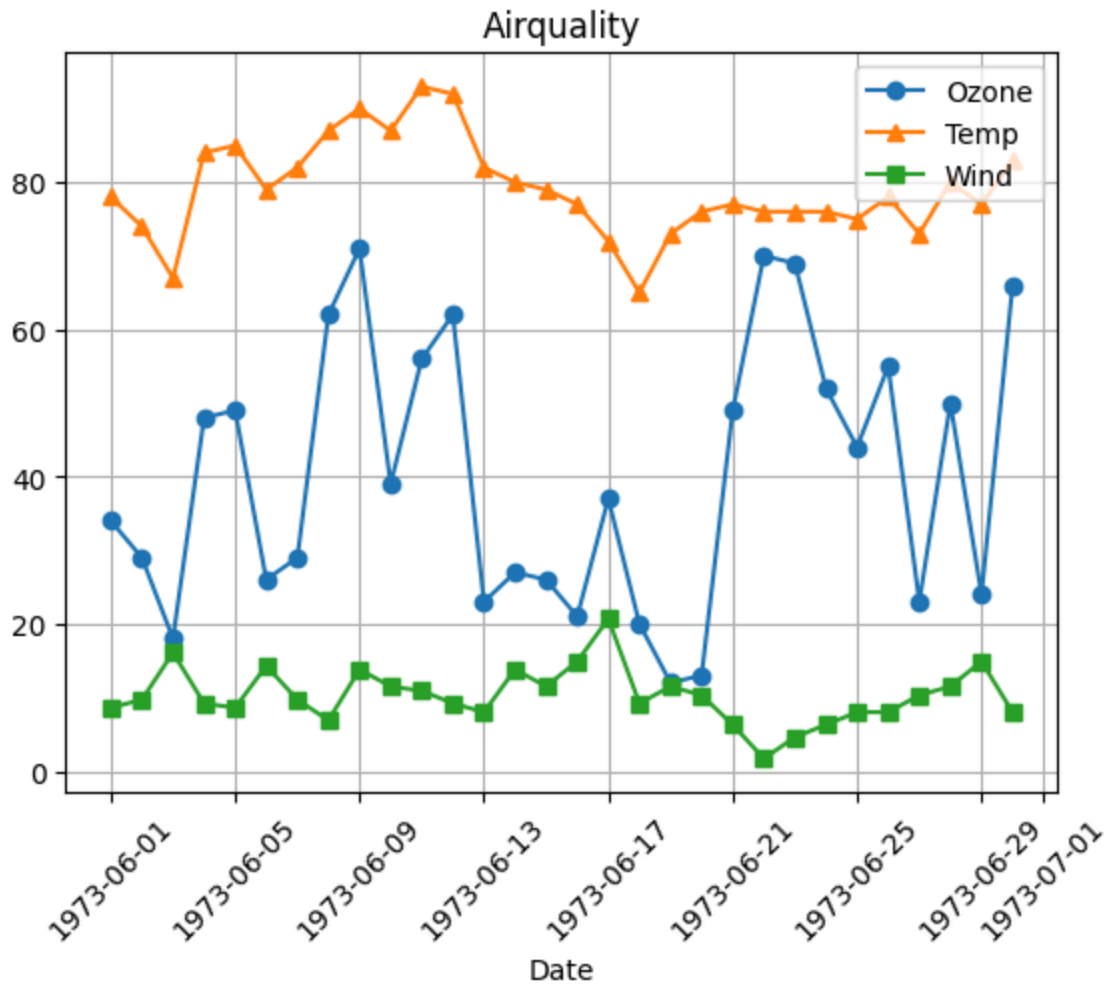
- 그래프1 : Date, Ozone
- 그래프2 : Date, Temp
- 그래프3 : Date, Wind
- x축 이름 : Date
- title : Airquality
- legend 추가
- grid 추가
- 그래프 크기 : 12,8 (이 부분은 배우지는 않았습니다. 아래 4.추가기능 - (3) 그래프 크기 조정 부분을 참조하여 적용해 봅시다.)

```
In [40]: # 1번 그래프
plt.plot(data['Date'], data['Ozone'], label='Ozone', marker='o')
# 2번 그래프
plt.plot(data['Date'], data['Temp'], label='Temp', marker='^')
# 3번 그래프
plt.plot(data['Date'], data['Wind'], label='Wind', marker='s')

# x축 이름
plt.xlabel('Date')
# x축 회전
```

```
plt.xticks(rotation=45)
# 제목
plt.title('Airquality')
# 범례
plt.legend(loc = 'upper right')
# 그리드 추가
plt.grid()

# 화면에 보여주기
plt.show()
```



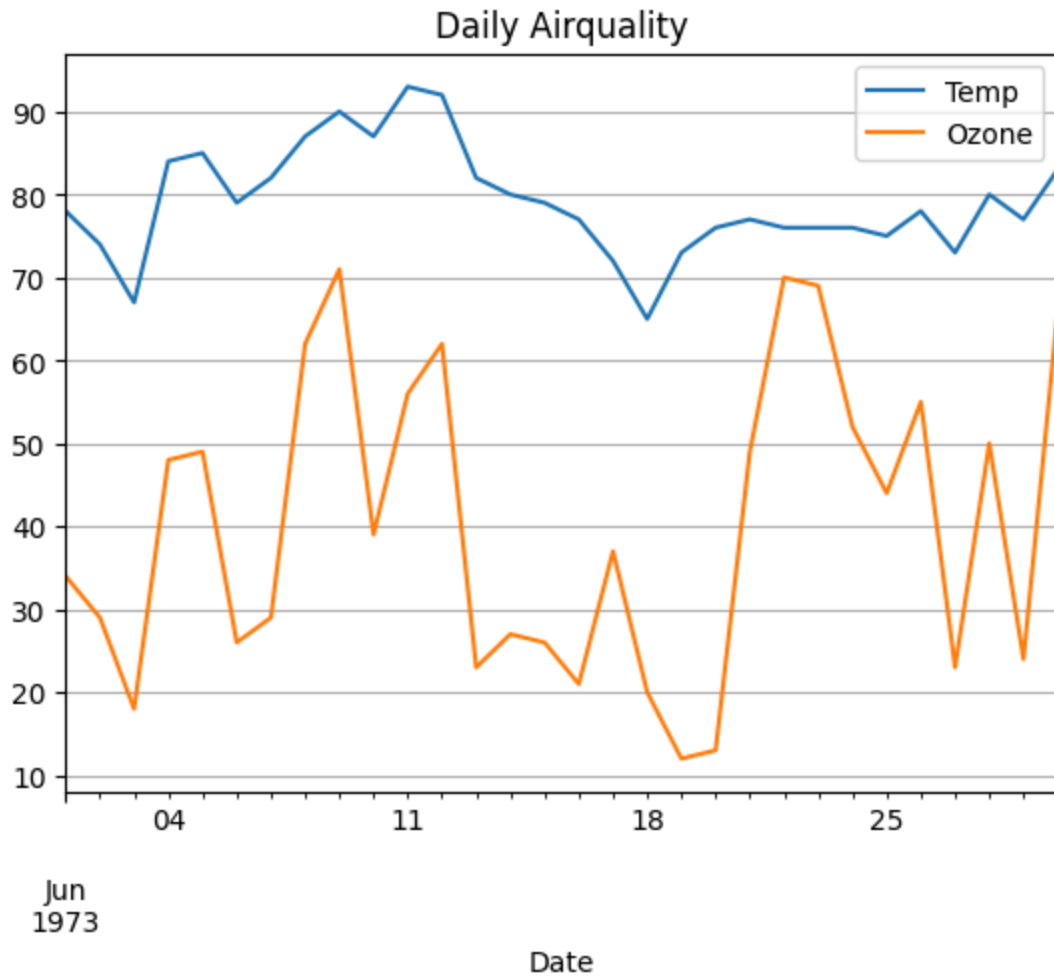
교육과정은 여기까지 다룹니다. 아래 추가 기능은 필요시 살펴보시기 바랍니다.

4.추가기능①

(1) 데이터프레임.plot()

- <https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.plot.html>

```
In [41]: data.plot(x = 'Date', y = ['Temp', 'Ozone']
           , title = 'Daily Airquality')
plt.grid()
plt.show()
```



-연습문제-

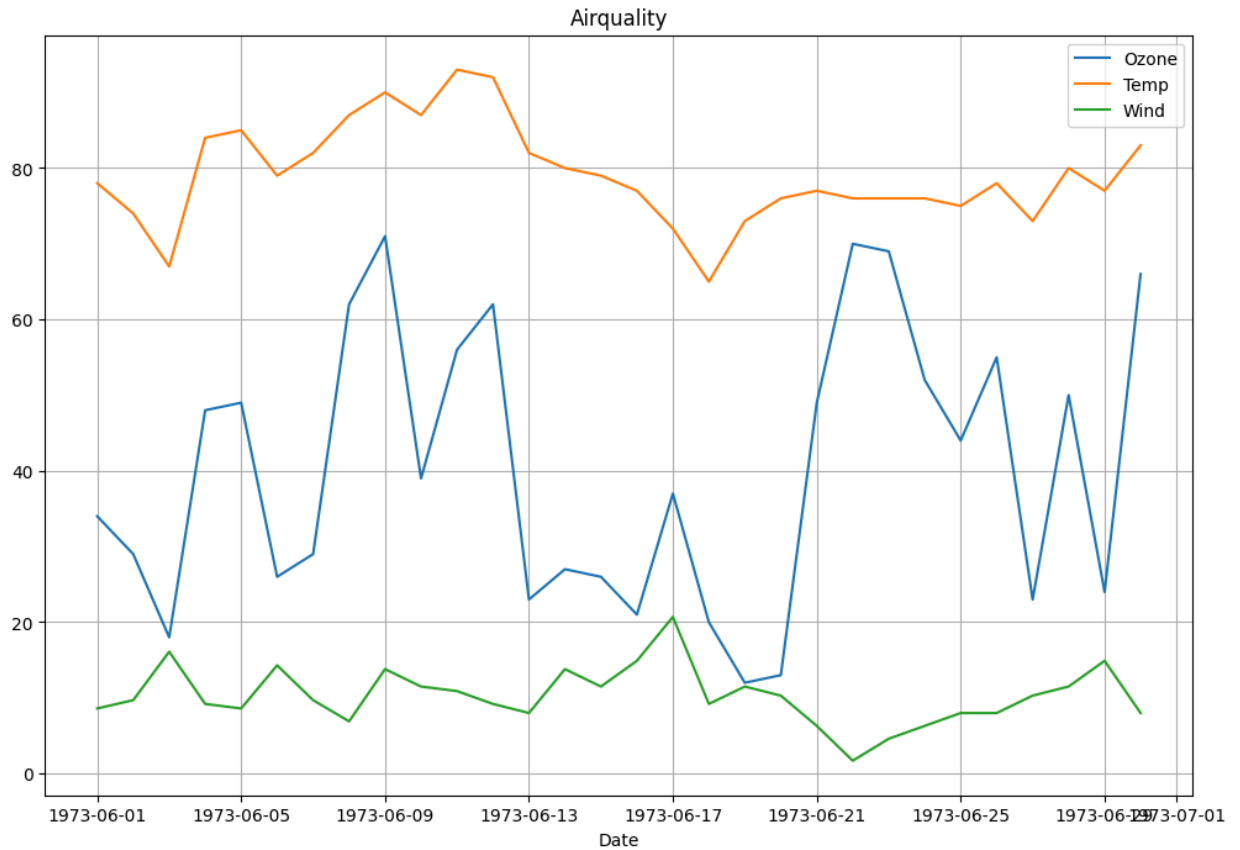
아래 조건에 맞게 그래프를 그려 봅시다.(이번에는 데이터프레임, plot 으로 그려 봅시다.)

- 값
 - x축 : Date
 - y축 : Ozone, Temp, Wind
- x축 이름 : Date
- title : Airquality
- legend 추가
- grid 추가
- 그래프 크기 : 12,8
 - .plot(, figsize=(12, 8))

```
In [55]: # 그래프 크기
plt.figure(figsize=(12, 8))
# 그래프
plt.plot(data['Date'], data[['Ozone', 'Temp', 'Wind']])
# x축 이름
plt.xlabel('Date')
# 제목
plt.title('Airquality')
# 범례
```

```
plt.legend(['Ozone', 'Temp', 'Wind'], loc = 'upper right')
# 그리드
plt.grid()

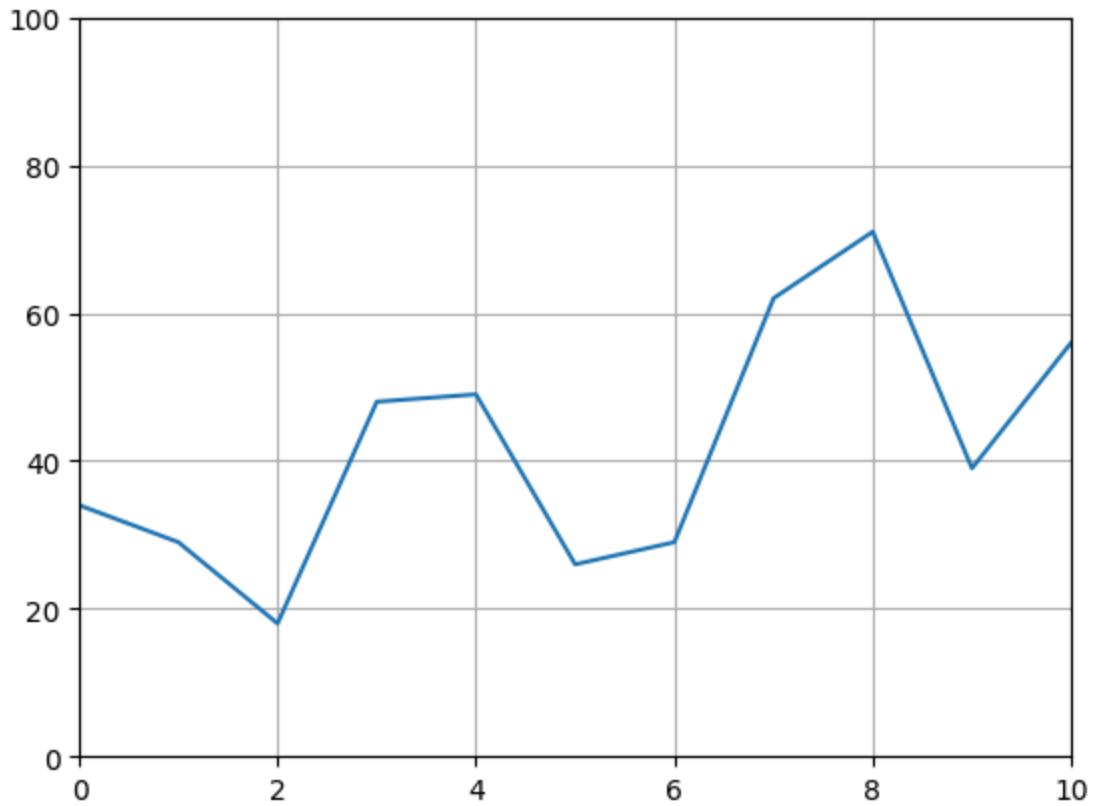
# 보여주기
plt.show()
```



(2) 축 범위 조정하기

```
In [57]: plt.plot(data['Ozone'])

# y축의 범위 지정
plt.ylim(0, 100)
# x축의 범위 지정
plt.xlim(0, 10)
plt.grid()
plt.show()
```



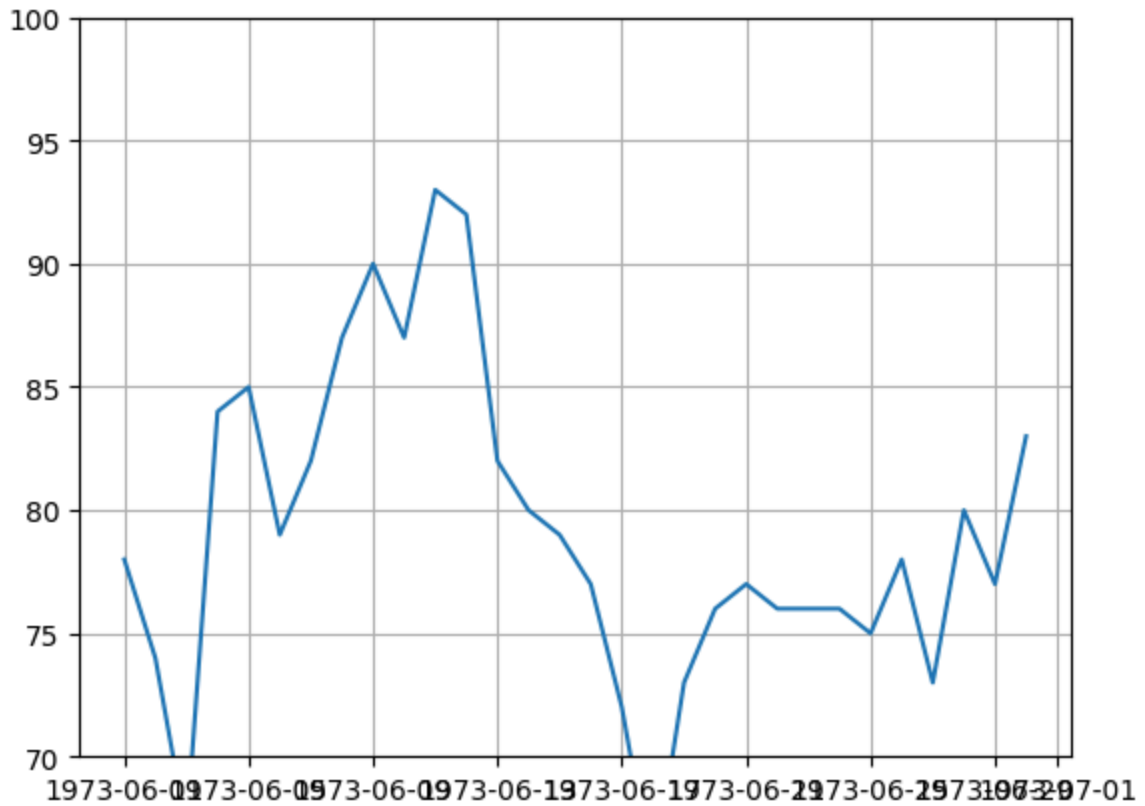
- 연습문제 -

아래 조건에 맞게 그래프를 그려 봅시다.

- 값
 - x축 : Date
 - y축 : Temp
- y축 범위 : 70 ~ 100

```
In [58]: plt.plot(data['Date'], data['Temp'])
```

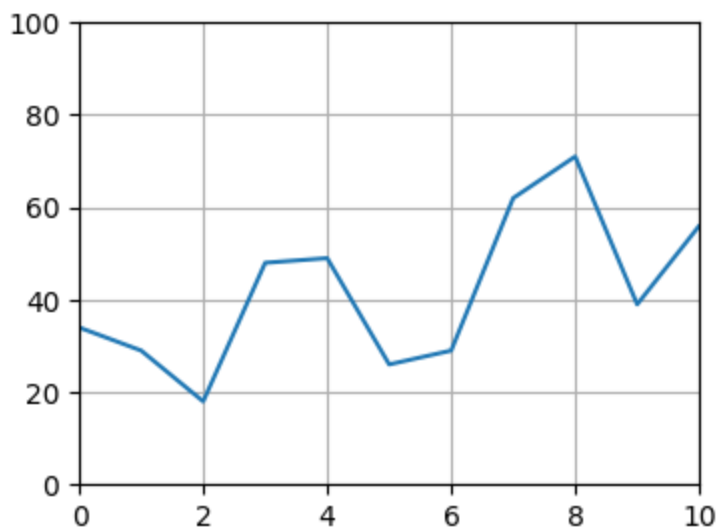
```
# y축의 범위 지정  
plt.ylim(70, 100)  
# x축의 범위 지정  
#plt.xlim(0, 10)  
plt.grid()  
plt.show()
```



(3) 그래프 크기 조정

* default size는 6.4, 4.4

```
In [59]: plt.figure(figsize = (4, 3))
plt.plot(data['Ozone'])
plt.ylim(0, 100)
plt.xlim(0,10)
plt.grid()
plt.show()
```

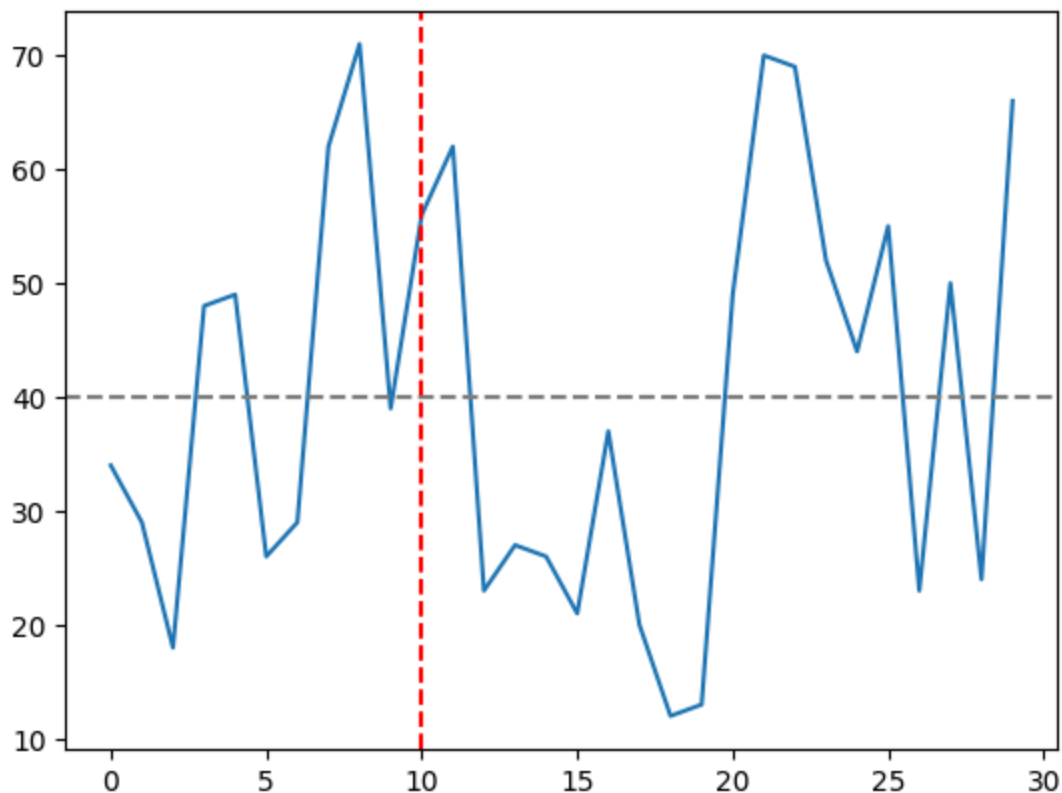


- 그래프 코드 순서
 - 1. 크기 조정: plt.figure(figsize = (,))
 - 1. 그래프 그리기: plt.plot()
 - 1. 꾸미기:
 - 1. 이제는 보여줘: plt.show()

(4) 수평선 수직선 추가

```
In [60]: plt.plot(data['Ozone'])

plt.axhline(40, color = 'grey', linestyle = '--')
plt.axvline(10, color = 'red', linestyle = '--')
plt.show()
```



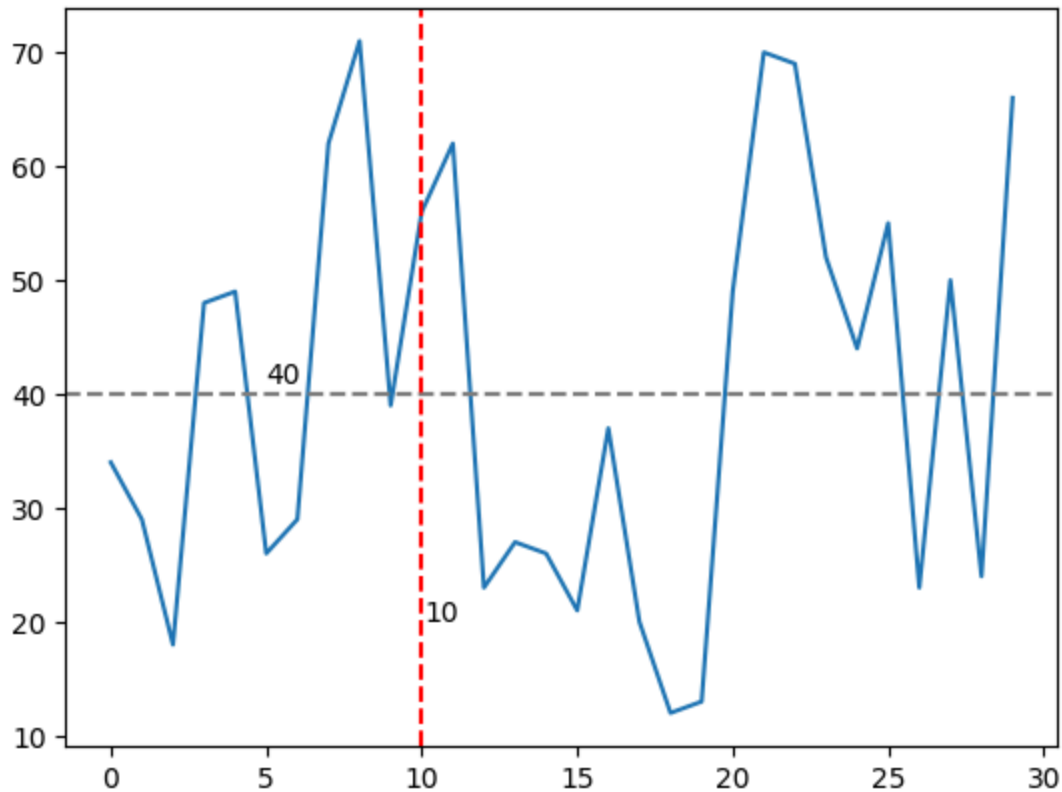
(5) 그래프에 텍스트 추가

```
In [62]: plt.plot(data['Ozone'])

plt.axhline(40, color = 'grey', linestyle = '--')
plt.axvline(10, color = 'red', linestyle = '--')

# 텍스트 표현 # x, y, 표현문자
plt.text(5, 41, '40')
plt.text(10.1, 20, '10')

plt.show()
```



-연습문제-

아래 정보로 그래프를 그려 봅시다.

- 오존 농도를 그래프로 그리고,
- 오존 농도의 평균선(수평선)과 평균값(text)을 추가하시오.

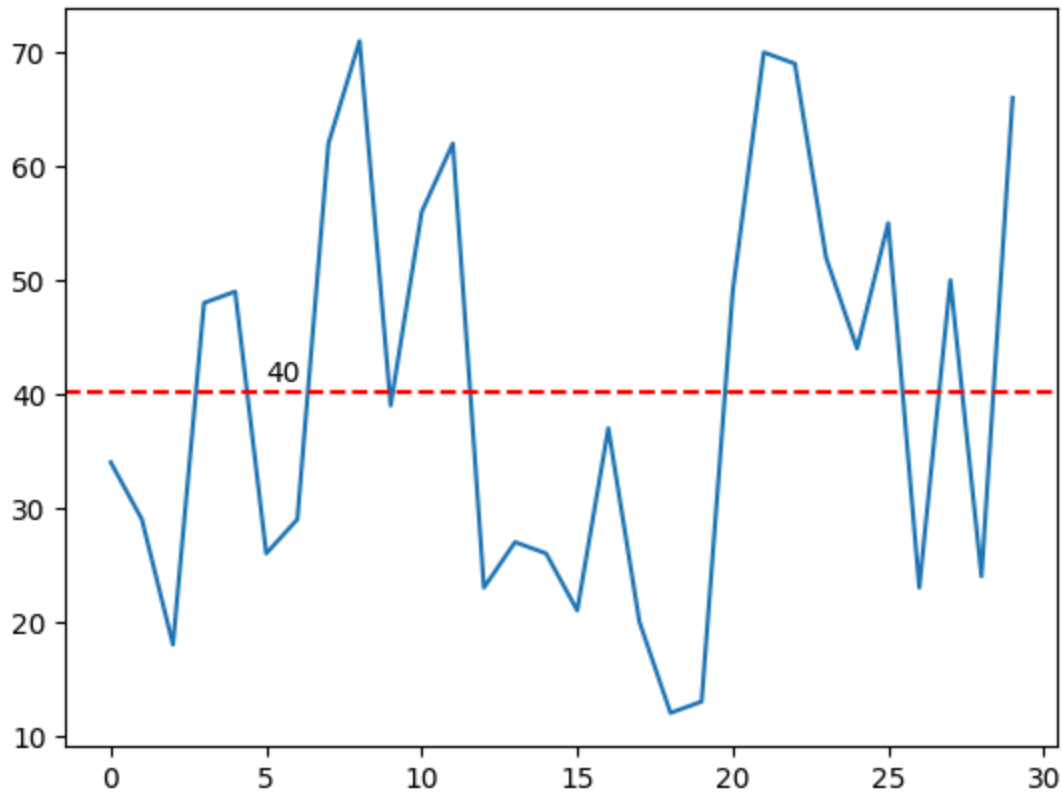
```
In [78]: plt.plot(data['Ozone'])

# 오존 평균값
ozone_mean = data['Ozone'].mean()

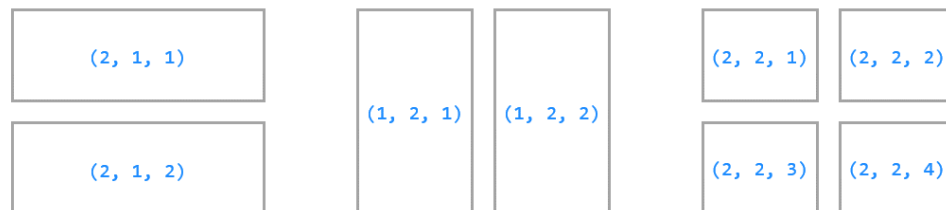
plt.axhline(ozone_mean, color = 'red', linestyle = '--')

# 텍스트 표현 # x, y, 표현문자
plt.text(5, ozone_mean+1, '40')

plt.show()
```



5.추가기능② : 여러 그래프 나눠서 그리기



`plt.subplot(row, column, index)`

- row : 고정된 행 수
- column : 고정된 열 수
- index : 순서

(1) 위, 아래로(3행 1열) 그래프 그리기.

```
In [63]: data.head()
```

Out[63]:

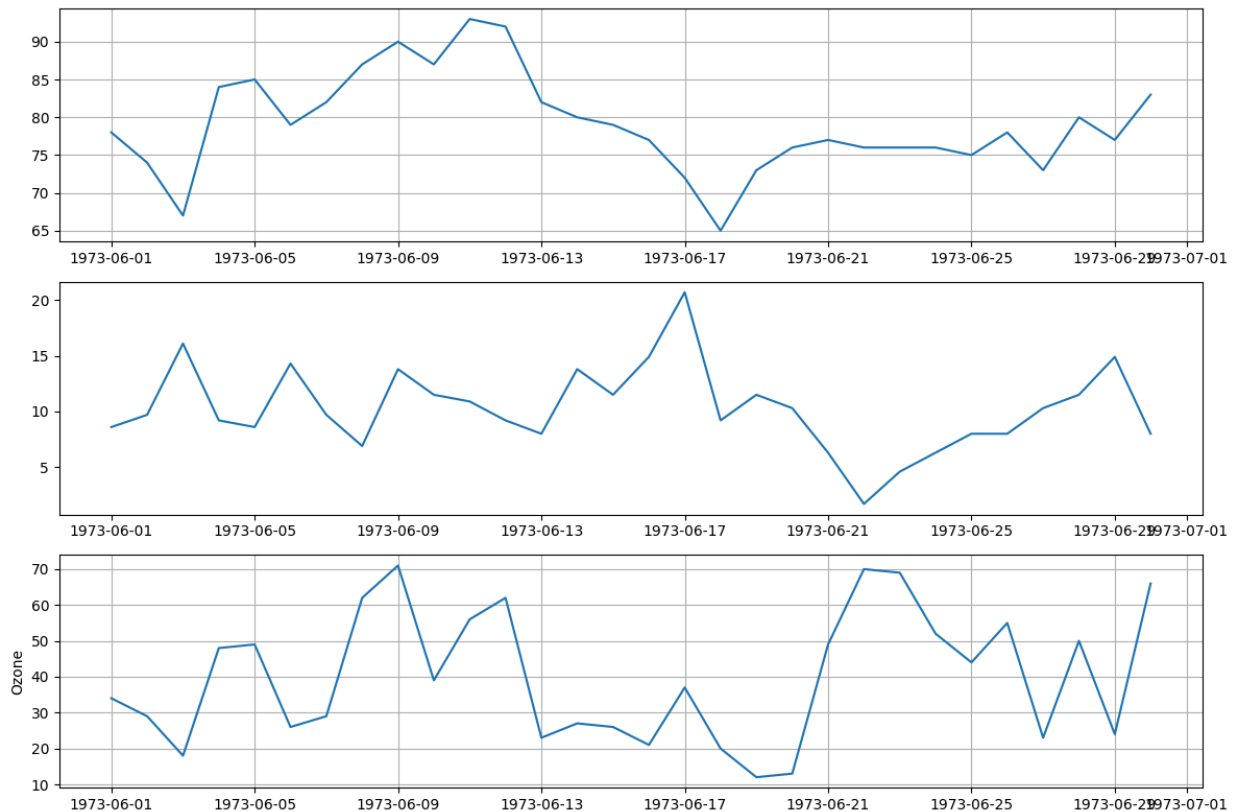
	Ozone	Solar.R	Wind	Temp	Date
0	34.0	286.0	8.6	78.0	1973-06-01
1	29.0	287.0	9.7	74.0	1973-06-02
2	18.0	242.0	16.1	67.0	1973-06-03
3	48.0	186.0	9.2	84.0	1973-06-04
4	49.0	220.0	8.6	85.0	1973-06-05

```
In [68]: plt.figure(figsize = (12,8))
plt.subplot(3,1,1)
plt.plot('Date', 'Temp', data = data)
plt.grid()

plt.subplot(3,1,2)
plt.plot('Date', 'Wind', data = data)

plt.subplot(3,1,3)
plt.plot('Date', 'Ozone', data = data)
plt.grid()
plt.ylabel('Ozone')

plt.tight_layout() # 그래프간 간격을 적절히 맞추기
plt.show()
```



(2) 옆으로 (1행 3열) 그래프 그리기.

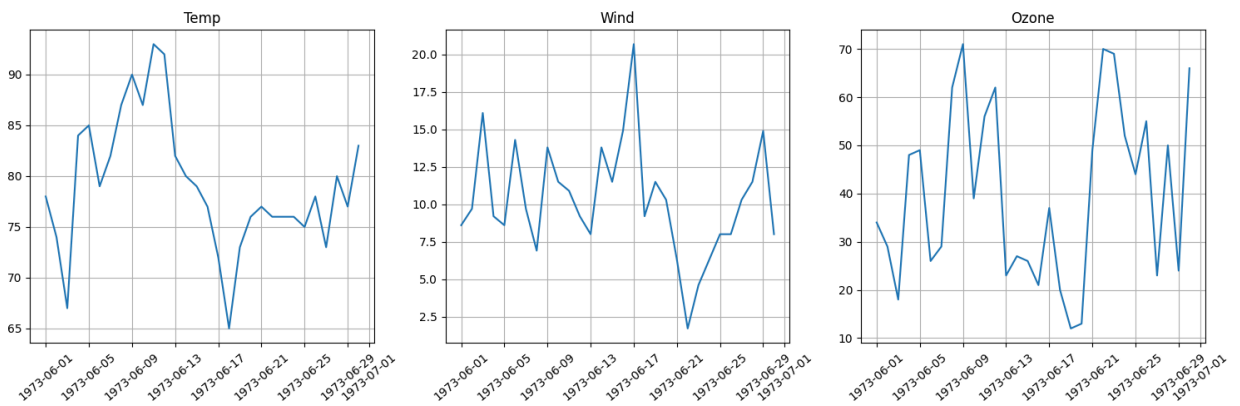
```
In [69]: plt.figure(figsize = (15,5))
plt.subplot(1,3,1)
```

```
plt.plot('Date', 'Temp', data = data)
plt.title('Temp')
plt.xticks(rotation = 40)
plt.grid()

plt.subplot(1,3,2)
plt.plot('Date', 'Wind', data = data)
plt.title('Wind')
plt.xticks(rotation = 40)
plt.grid()

plt.subplot(1,3,3)
plt.plot('Date', 'Ozone', data = data)
plt.title('Ozone')
plt.xticks(rotation = 40)
plt.grid()

plt.tight_layout() # 그래프간 간격을 적절히 맞추기
plt.show()
```



-연습문제-

주어진 데이터로 그래프를 그려봅시다.

- 그래프1 : x, y1
- 그래프2 : x, y2
- 1행, 2열로 그래프를 그리시오.

```
In [79]: dict1 = {'x':[1,2,3,4,5,6,7,8,9,10,11,12], 'y1':[21,56,32,18,27,54,35,49,92,87,74,76], 'y2':[3
df = pd.DataFrame(dict1)
df
```

Out[79]:

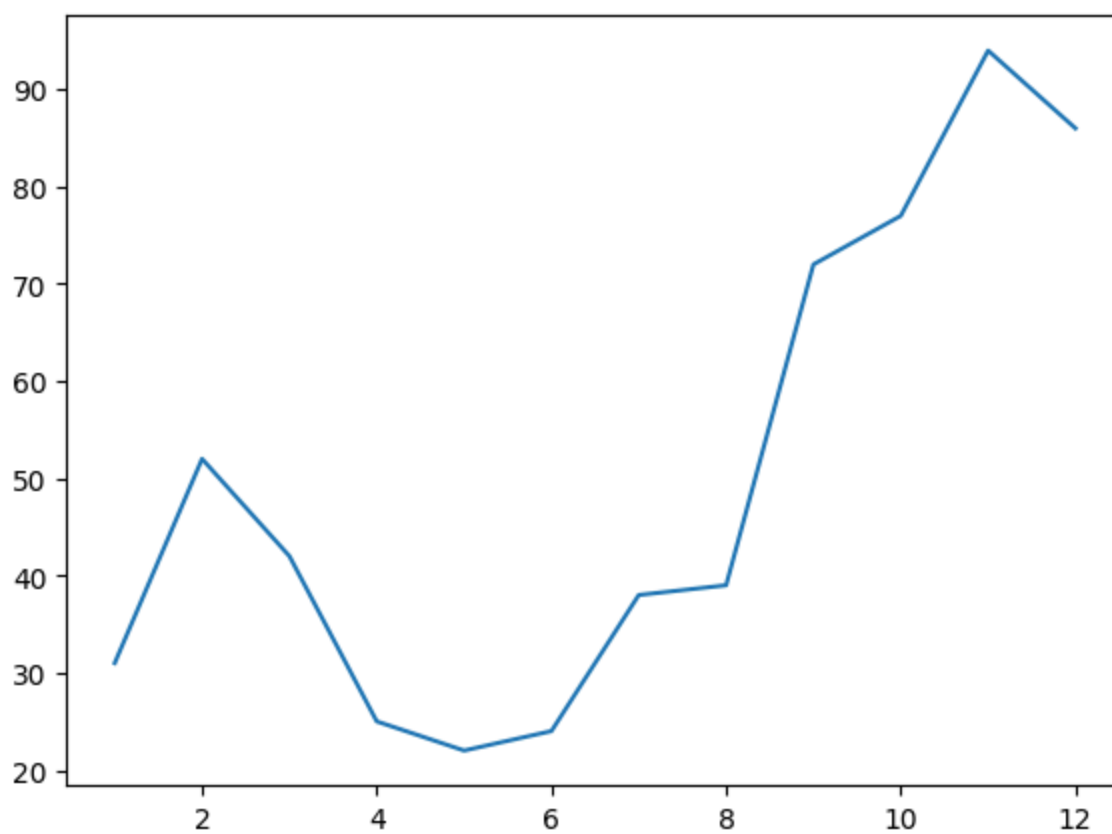
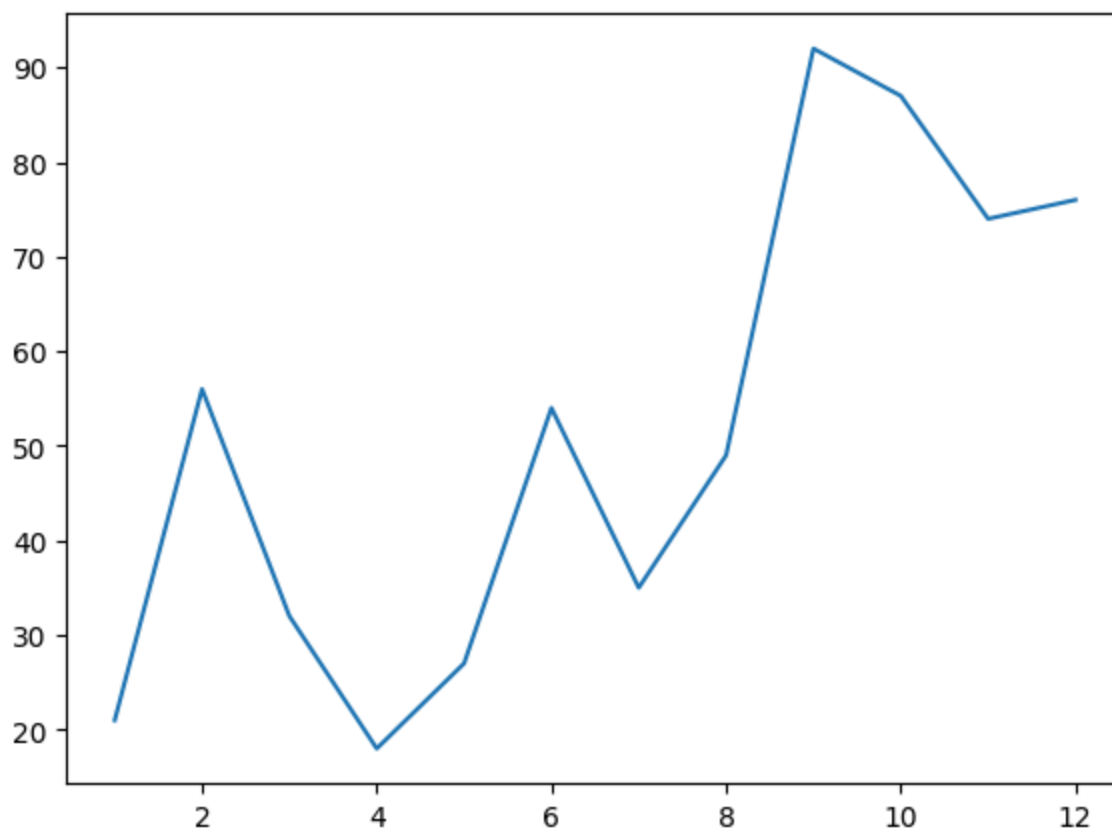
	x	y1	y2
0	1	21	31
1	2	56	52
2	3	32	42
3	4	18	25
4	5	27	22
5	6	54	24
6	7	35	38
7	8	49	39
8	9	92	72
9	10	87	77
10	11	74	94
11	12	76	86

In [84]:

```
plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 1)
plt.plot(df['x'], df['y1'])

plt.figure(figsize=(15, 5))
plt.subplot(1, 2, 2)
plt.plot(df['x'], df['y2'])

plt.show()
```



In []: