In [1]:

```
import pandas as pd
import numpy as np
import datetime as dt
import glob

# ./ 현재 있는 위치
# ../ 현재 있는 위치에서 한 단계 뒤로
air = glob.glob("../data/2020_airkorea/2020y_*")
air.sort()
air
```

Out[1]:

```
['../data/2020_airkorea₩₩2020y_01.xlsx',
 '../data/2020_airkorea₩₩2020y_02.xlsx',
 '../data/2020_airkorea₩₩2020y_03.xlsx',
 '../data/2020_airkorea₩₩2020y_04.xlsx',
 '../data/2020_airkorea₩₩2020y_05.xlsx',
 '../data/2020_airkorea₩₩2020y_06.xlsx',
 '../data/2020_airkorea₩₩2020y_07.xlsx',
 '../data/2020_airkorea₩₩2020y_08.xlsx',
 '../data/2020_airkorea₩₩2020y_09.xlsx',
 '../data/2020_airkorea₩₩2020y_10.xlsx',
 '../data/2020_airkorea₩₩2020y_11.xlsx',
 '../data/2020_airkorea₩₩2020y_12.xlsx']
```

In [3]:

```
# *** 자료 불러오는데 오래 걸리니 건들지 말기 ***

#for i in range(1, 13):
#    globals() ['seoul_air_{}'.format(i)] = pd.read_excel(air[i-1])


seoul_air_1 = pd.read_excel(air[0])
seoul_air_2 = pd.read_excel(air[1])
seoul_air_3 = pd.read_excel(air[2])
seoul_air_4 = pd.read_excel(air[3])
seoul_air_5 = pd.read_excel(air[4])
seoul_air_6 = pd.read_excel(air[5])
seoul_air_7 = pd.read_excel(air[6])
seoul_air_8 = pd.read_excel(air[7])
seoul_air_9 = pd.read_excel(air[8])
seoul_air_10 = pd.read_excel(air[9])
seoul_air_11 = pd.read_excel(air[10])
seoul_air_12 = pd.read_excel(air[11])

# *** 자료 불러오는데 오래 걸리니 건들지 말기 ***
```

In [4]:

```
seoul_air_1
```

Out[4]:

| | 지역 | 망 | 측정소코드 | 측정소명 | 측정일시 | SO2 | CO | O3 | NO2 | PM10 | PM25 | 주소 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 서울중구 | 도시대기 | 111121 | 중구 | 2020010101 | 0.003 | 0.5 | 0.002 | 0.036 | 24.0 | 19.0 | 서울 중구 덕수궁길 15 |
| **1** | 서울중구 | 도시대기 | 111121 | 중구 | 2020010102 | 0.003 | 0.6 | 0.001 | 0.039 | 25.0 | 21.0 | 서울 중구 덕수궁길 15 |
| **2** | 서울중구 | 도시대기 | 111121 | 중구 | 2020010103 | 0.003 | 0.9 | 0.001 | 0.037 | 29.0 | 23.0 | 서울 중구 덕수궁길 15 |
| **3** | 서울중구 | 도시대기 | 111121 | 중구 | 2020010104 | 0.002 | 0.6 | 0.001 | 0.036 | 26.0 | 22.0 | 서울 중구 덕수궁길 15 |
| **4** | 서울중구 | 도시대기 | 111121 | 중구 | 2020010105 | 0.002 | 0.6 | 0.001 | 0.035 | 25.0 | 19.0 | 서울 중구 덕수궁길 15 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **367299** | 인천옹진군 | 도시대기 | 831493 | 영흥 | 2020013120 | 0.003 | 0.4 | 0.052 | 0.004 | 44.0 | 36.0 | 인천광역시 옹진군 영흥면 영흥로251번길 90 |
| **367300** | 인천옹진군 | 도시대기 | 831493 | 영흥 | 2020013121 | 0.003 | 0.4 | 0.052 | 0.004 | 40.0 | 28.0 | 인천광역시 옹진군 영흥면 영흥로251번길 90 |
| **367301** | 인천옹진군 | 도시대기 | 831493 | 영흥 | 2020013122 | 0.003 | 0.4 | 0.051 | 0.004 | 34.0 | 29.0 | 인천광역시 옹진군 영흥면 영흥로251번길 90 |
| **367302** | 인천옹진군 | 도시대기 | 831493 | 영흥 | 2020013123 | 0.003 | 0.4 | 0.049 | 0.005 | 36.0 | 33.0 | 인천광역시 옹진군 영흥면 영흥로251번길 90 |
| **367303** | 인천옹진군 | 도시대기 | 831493 | 영흥 | 2020013124 | 0.003 | 0.4 | 0.049 | 0.004 | 35.0 | 32.0 | 인천광역시 옹진군 영흥면 영흥로251번길 90 |

367304 rows × 12 columns

In [5]:

```
for i in range(1, 13):
    globals()['seoul_air_{}'.format(i)] = globals()['seoul_air_{}'.format(i)][globals()['seoul_air_{
                                                                                ['주소'] == '서울 도봉

seoul_air_1.columns
```

Out[5]:

```
Index(['측정일시', 'SO2', 'CO', 'O3', 'NO2', 'PM10'], dtype='object')
```

In [6]:

```
Total_Seoul_airkorea = pd.DataFrame()

for i in range(1, 13):
    Total_Seoul_airkorea = pd.concat([Total_Seoul_airkorea, globals()['seoul_air_{}'.format(i)]])

Total_Seoul_airkorea = Total_Seoul_airkorea.reset_index(drop = True)

Total_Seoul_airkorea
```

Out[6]:

| | 측정일시 | SO2 | CO | O3 | NO2 | PM10 |
|---|---|---|---|---|---|---|
| **0** | 2020010101 | 0.002 | 0.5 | 0.011 | 0.024 | 19.0 |
| **1** | 2020010102 | 0.002 | 0.6 | 0.005 | 0.030 | 19.0 |
| **2** | 2020010103 | 0.002 | 0.6 | 0.002 | 0.033 | 27.0 |
| **3** | 2020010104 | 0.002 | 0.6 | 0.003 | 0.031 | 20.0 |
| **4** | 2020010105 | 0.002 | 0.7 | 0.003 | 0.031 | 21.0 |
| **...** | ... | ... | ... | ... | ... | ... |
| **8779** | 2020123120 | 0.002 | 0.4 | 0.014 | 0.026 | 29.0 |
| **8780** | 2020123121 | 0.002 | 0.4 | 0.017 | 0.021 | 23.0 |
| **8781** | 2020123122 | 0.002 | 0.4 | 0.025 | 0.013 | 28.0 |
| **8782** | 2020123123 | 0.002 | 0.3 | 0.030 | 0.008 | 24.0 |
| **8783** | 2020123124 | 0.002 | 0.3 | 0.027 | 0.011 | 15.0 |

8784 rows × 6 columns

In [7]:

```
for i in range(len(Total_Seoul_airkorea.columns)):
    if (Total_Seoul_airkorea[Total_Seoul_airkorea.columns[i]].isnull().values.any() == True):
        print("NaN이 하나라도 포함되어 있는  cloumn : ", Total_Seoul_airkorea.columns[i])
```

```
NaN이 하나라도 포함되어 있는  cloumn :  SO2
NaN이 하나라도 포함되어 있는  cloumn :  CO
NaN이 하나라도 포함되어 있는  cloumn :  O3
NaN이 하나라도 포함되어 있는  cloumn :  NO2
NaN이 하나라도 포함되어 있는  cloumn :  PM10
```

In [8]:

```
for i in range(len(Total_Seoul_airkorea.columns)):
    for j in range(len(Total_Seoul_airkorea)):
        if (Total_Seoul_airkorea[Total_Seoul_airkorea.columns[i]][j] == -999):
            print("-999 값이 있는 column : ", Total_Seoul_airkorea.columns[i])
            break
```

In [9]:

```
Total_Seoul_airkorea_mean = round(Total_Seoul_airkorea.mean(), 3)

Total_Seoul_airkorea_mean
```

Out[9]:

```
측정일시    2.020067e+09
SO2     3.000000e-03
CO      4.030000e-01
O3      2.900000e-02
NO2     1.800000e-02
PM10    3.205700e+01
dtype: float64
```

In [10]:

```
for value in Total_Seoul_airkorea_mean:
    print(value)
```

```
2020066724.795
0.003
0.403
0.029
0.018
32.057
```

In [11]:

```
remove_airkorea_columns = []

for i in range(len(Total_Seoul_airkorea_mean)):
    if (np.isnan(Total_Seoul_airkorea_mean[i])):
        print("모든 값이 NaN인 column : ", Total_Seoul_airkorea_columns[i])

        remove_airkorea_columns.append(Total_Seoul_airkorea.columns[i])


for i in range(len(remove_airkorea_columns)):
    print(remove_airkorea_columns[i], '제거')

    del Total_Seoul_airkorea[remove_airkorea_columns[i]]
```

In [12]:

```
Total_Seoul_airkorea = Total_Seoul_airkorea.fillna(Total_Seoul_airkorea_mean)

Total_Seoul_airkorea
```

Out[12]:

| | 측정일시 | SO2 | CO | O3 | NO2 | PM10 |
|---|---|---|---|---|---|---|
| **0** | 2020010101 | 0.002 | 0.5 | 0.011 | 0.024 | 19.0 |
| **1** | 2020010102 | 0.002 | 0.6 | 0.005 | 0.030 | 19.0 |
| **2** | 2020010103 | 0.002 | 0.6 | 0.002 | 0.033 | 27.0 |
| **3** | 2020010104 | 0.002 | 0.6 | 0.003 | 0.031 | 20.0 |
| **4** | 2020010105 | 0.002 | 0.7 | 0.003 | 0.031 | 21.0 |
| **...** | ... | ... | ... | ... | ... | ... |
| **8779** | 2020123120 | 0.002 | 0.4 | 0.014 | 0.026 | 29.0 |
| **8780** | 2020123121 | 0.002 | 0.4 | 0.017 | 0.021 | 23.0 |
| **8781** | 2020123122 | 0.002 | 0.4 | 0.025 | 0.013 | 28.0 |
| **8782** | 2020123123 | 0.002 | 0.3 | 0.030 | 0.008 | 24.0 |
| **8783** | 2020123124 | 0.002 | 0.3 | 0.027 | 0.011 | 15.0 |

8784 rows × 6 columns

In [13]:

```
kma = pd.read_csv("../data/2020_kma/2020_Seoul.csv", encoding = 'cp949')

kma
```

Out[13]:

| | 지점 | 지점명 | 일시 | 기온 (°C) | 기온 QC 플래그 | 강수량 (mm) | 강수량 QC 플래그 | 풍속 (m/s) | 풍속 QC 플래그 | 풍향 (16방위) | ... | 최저운고 (100m) | 시정 (10m) | 지면상태 (지면상태코드) | 현상번호 (국내식) | (°C |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 108 | 서울 | 2020-01-01 00:00 | -6.5 | NaN | 0.0 | NaN | 0.0 | NaN | 0 | ... | NaN | 2000 | NaN | 5.0 | -2. |
| **1** | 108 | 서울 | 2020-01-01 01:00 | -5.9 | NaN | NaN | 9.0 | 1.7 | NaN | 50 | ... | 7.0 | 2000 | NaN | 5.0 | -2. |
| **2** | 108 | 서울 | 2020-01-01 02:00 | -5.7 | NaN | NaN | 9.0 | 0.1 | NaN | 0 | ... | 7.0 | 1988 | NaN | 5.0 | -2. |
| **3** | 108 | 서울 | 2020-01-01 03:00 | -5.6 | NaN | 0.0 | NaN | 0.0 | NaN | 0 | ... | 14.0 | 2000 | NaN | 5.0 | -2. |
| **4** | 108 | 서울 | 2020-01-01 04:00 | -5.4 | NaN | NaN | 9.0 | 0.0 | NaN | 0 | ... | 6.0 | 1908 | NaN | 5.0 | -2. |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | . |
| **8779** | 108 | 서울 | 2020-12-31 19:00 | -7.1 | NaN | NaN | 9.0 | 2.4 | NaN | 250 | ... | NaN | 2000 | NaN | NaN | -4. |
| **8780** | 108 | 서울 | 2020-12-31 20:00 | -7.1 | NaN | NaN | 9.0 | 3.2 | NaN | 250 | ... | NaN | 2000 | NaN | NaN | -5. |
| **8781** | 108 | 서울 | 2020-12-31 21:00 | -7.2 | NaN | NaN | 9.0 | 2.7 | NaN | 250 | ... | NaN | 2000 | NaN | NaN | -5. |
| **8782** | 108 | 서울 | 2020-12-31 22:00 | -7.4 | NaN | NaN | 9.0 | 2.5 | NaN | 270 | ... | NaN | 2000 | NaN | NaN | -6. |
| **8783** | 108 | 서울 | 2020-12-31 23:00 | -7.6 | NaN | NaN | 9.0 | 2.2 | NaN | 290 | ... | NaN | 2000 | NaN | NaN | -6. |

8784 rows × 38 columns

In [14]:

```
seoul_kma = kma[['일시', '기온(° C)', '강수량(mm)', '풍속(m/s)', '습도(%)', '이슬점온도(° C)']]
seoul_kma = seoul_kma.reset_index(drop = True)

seoul_kma
```

Out[14]:

| | 일시 | 기온(°C) | 강수량(mm) | 풍속(m/s) | 습도(%) | 이슬점온도(°C) |
|---|---|---|---|---|---|---|
| **0** | 2020-01-01 00:00 | -6.5 | 0.0 | 0.0 | 38 | -18.5 |
| **1** | 2020-01-01 01:00 | -5.9 | NaN | 1.7 | 40 | -17.3 |
| **2** | 2020-01-01 02:00 | -5.7 | NaN | 0.1 | 42 | -16.5 |
| **3** | 2020-01-01 03:00 | -5.6 | 0.0 | 0.0 | 46 | -15.4 |
| **4** | 2020-01-01 04:00 | -5.4 | NaN | 0.0 | 50 | -14.2 |
| **...** | ... | ... | ... | ... | ... | ... |
| **8779** | 2020-12-31 19:00 | -7.1 | NaN | 2.4 | 58 | -13.9 |
| **8780** | 2020-12-31 20:00 | -7.1 | NaN | 3.2 | 59 | -13.7 |
| **8781** | 2020-12-31 21:00 | -7.2 | NaN | 2.7 | 61 | -13.4 |
| **8782** | 2020-12-31 22:00 | -7.4 | NaN | 2.5 | 66 | -12.6 |
| **8783** | 2020-12-31 23:00 | -7.6 | NaN | 2.2 | 65 | -13.0 |

8784 rows × 6 columns

In [15]:

```
for i in range(len(seoul_kma.columns)):
    if (seoul_kma[seoul_kma.columns[i]].isnull().values.any() == True):
        print("NaN이 하나라도 포함되어 있는  cloumn : ", seoul_kma.columns[i])
```

```
NaN이 하나라도 포함되어 있는  cloumn :  기온(° C)
NaN이 하나라도 포함되어 있는  cloumn :  강수량(mm)
NaN이 하나라도 포함되어 있는  cloumn :  이슬점온도(° C)
```

In [16]:

```
for i in range(len(seoul_kma.columns)):
    for j in range(len(seoul_kma)):
        if (seoul_kma[seoul_kma.columns[i]][j] == -999):
            print("-999 값이 있는 column : ", seoul_kma.columns[i])
            break
```

In [17]:

```
seoul_kma_mean = round(seoul_kma.mean(), 3)

seoul_kma_mean
```

```
C:₩Users₩wogml₩AppData₩Local₩Temp₩ipykernel_276₩3893362722.py:1: FutureWarning: Drop
ping of nuisance columns in DataFrame reductions (with 'numeric_only=None') is depre
cated; in a future version this will raise TypeError.  Select only valid columns bef
ore calling the reduction.
  seoul_kma_mean = round(seoul_kma.mean(), 3)
```

Out[17]:

```
기온(° C)        13.268
강수량(mm)        1.556
풍속(m/s)        2.374
습도(%)         63.259
이슬점온도(° C)     5.711
dtype: float64
```

In [18]:

```
# 모든 값이 NaN인 컬럼을 저장할 removw_kma_columns 라는 리스트를 생성
# mean() 함수를 적용한 값이 NaN이면 그 컬럼의 값은 모두 NaN이라는 뜻
# np.isnan의 파라미터가 nan이면 True를 반환
# 모든 값이 NaN인 컬럼을 출력하고 remove_kma_columns에 칼럼을 append 시킴

remove_kma_columns = []

for i in range(len(seoul_kma_mean)):
    if (np.isnan(seoul_kma_mean[i])):
        print("모든 값이 NaN인 column : ", seoul_kma.columns[i])

        remove_seoul_kma_columns.append(seoul_kma.columns[i])

print(remove_kma_columns)


# del을 이용해 remove_kma_columns에 있는 컬럼 삭제
for i in range(len(remove_kma_columns)):
    print(remove_kma_columns[i], '제거')

    del seoul_kma[remove_kma_columns[i]]
```

```
[]
```

In [19]:

```
Total_Seoul_kma = seoul_kma.fillna(seoul_kma_mean)

Total_Seoul_kma
```

Out[19]:

| | 일시 | 기온(°C) | 강수량(mm) | 풍속(m/s) | 습도(%) | 이슬점온도(°C) |
|---|---|---|---|---|---|---|
| **0** | 2020-01-01 00:00 | -6.5 | 0.000 | 0.0 | 38 | -18.5 |
| **1** | 2020-01-01 01:00 | -5.9 | 1.556 | 1.7 | 40 | -17.3 |
| **2** | 2020-01-01 02:00 | -5.7 | 1.556 | 0.1 | 42 | -16.5 |
| **3** | 2020-01-01 03:00 | -5.6 | 0.000 | 0.0 | 46 | -15.4 |
| **4** | 2020-01-01 04:00 | -5.4 | 1.556 | 0.0 | 50 | -14.2 |
| **...** | ... | ... | ... | ... | ... | ... |
| **8779** | 2020-12-31 19:00 | -7.1 | 1.556 | 2.4 | 58 | -13.9 |
| **8780** | 2020-12-31 20:00 | -7.1 | 1.556 | 3.2 | 59 | -13.7 |
| **8781** | 2020-12-31 21:00 | -7.2 | 1.556 | 2.7 | 61 | -13.4 |
| **8782** | 2020-12-31 22:00 | -7.4 | 1.556 | 2.5 | 66 | -12.6 |
| **8783** | 2020-12-31 23:00 | -7.6 | 1.556 | 2.2 | 65 | -13.0 |

8784 rows × 6 columns

In [20]:

```
Total_Seoul_kma.columns = ['측정일시', 'Seoul_Temp(° C)', 'Seoul_Precipitation(mm)', 'Seoul_Wind_spe

Total_Seoul_kma
```

Out[20]:

| | 측정일시 | Seoul_Temp(°C) | Seoul_Precipitation(mm) | Seoul_Wind_speed(m/s) | Seoul_Humidity(% |
|---|---|---|---|---|---|
| **0** | 2020-01-01 00:00 | -6.5 | 0.000 | 0.0 | 3 |
| **1** | 2020-01-01 01:00 | -5.9 | 1.556 | 1.7 | 4 |
| **2** | 2020-01-01 02:00 | -5.7 | 1.556 | 0.1 | 4 |
| **3** | 2020-01-01 03:00 | -5.6 | 0.000 | 0.0 | 4 |
| **4** | 2020-01-01 04:00 | -5.4 | 1.556 | 0.0 | 5 |
| **...** | ... | ... | ... | ... | . |
| **8779** | 2020-12-31 19:00 | -7.1 | 1.556 | 2.4 | 5 |
| **8780** | 2020-12-31 20:00 | -7.1 | 1.556 | 3.2 | 5 |
| **8781** | 2020-12-31 21:00 | -7.2 | 1.556 | 2.7 | 6 |
| **8782** | 2020-12-31 22:00 | -7.4 | 1.556 | 2.5 | 6 |
| **8783** | 2020-12-31 23:00 | -7.6 | 1.556 | 2.2 | 6 |

8784 rows × 6 columns

In [21]:

```
#Total_Seoul_airkorea = Total_Seoul_airkorea.fillna(Total_Seoul_airkorea_mean)

#Total_Seoul_airkorea
```

In [22]:

```
#Total_Seoul_kma = seoul_kma.fillna(seoul_kma_mean)

#Total_Seoul_kma
```

In [23]:

```
def airkorea_num_to_datetime(date_str):
    if date_str[8:10] != '24':
        return pd.to_datetime(date_str, format = "%Y%m%d%H")

    else:
        date_str = date_str[0:8] + '00' + date_str[10:]

        return pd.to_datetime(date_str, format = '%Y%m%d%H', errors = 'raise') + dt.timedelta(days
```

In [24]:

```
def KMA_num_to_datetime(date_str):
    date_str = date_str[0:4] + date_str[5:7] + date_str[8:10] + date_str[11:13]

    return pd.to_datetime(date_str, format = "%Y%m%d%H")
```

In [25]:

```
for i in range(len(Total_Seoul_airkorea['측정일시'])):
    Total_Seoul_airkorea['측정일시'][i] = pd.Timestamp(airkorea_num_to_datetime((str)(Total_Seoul_ai
    Total_Seoul_kma['측정일시'][i] = pd.Timestamp(KMA_num_to_datetime((str)(Total_Seoul_kma['측정일시
```

```
C:₩Users₩wogml₩AppData₩Local₩Temp₩ipykernel_276₩1882310541.py:2: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  Total_Seoul_airkorea['측정일시'][i] = pd.Timestamp(airkorea_num_to_datetime((str)
(Total_Seoul_airkorea['측정일시'][i])))
C:₩Users₩wogml₩AppData₩Local₩Temp₩ipykernel_276₩1882310541.py:3: SettingWithCopyWarn
ing:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/u
ser_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pa
ndas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  Total_Seoul_kma['측정일시'][i] = pd.Timestamp(KMA_num_to_datetime((str)(Total_Seou
l_kma['측정일시'][i])))
```

In [26]:

```
Total_Seoul_airkorea
```

Out[26]:

| | 측정일시 | SO2 | CO | O3 | NO2 | PM10 |
|---|---|---|---|---|---|---|
| **0** | 2020-01-01 01:00:00 | 0.002 | 0.5 | 0.011 | 0.024 | 19.0 |
| **1** | 2020-01-01 02:00:00 | 0.002 | 0.6 | 0.005 | 0.030 | 19.0 |
| **2** | 2020-01-01 03:00:00 | 0.002 | 0.6 | 0.002 | 0.033 | 27.0 |
| **3** | 2020-01-01 04:00:00 | 0.002 | 0.6 | 0.003 | 0.031 | 20.0 |
| **4** | 2020-01-01 05:00:00 | 0.002 | 0.7 | 0.003 | 0.031 | 21.0 |
| **...** | ... | ... | ... | ... | ... | ... |
| **8779** | 2020-12-31 20:00:00 | 0.002 | 0.4 | 0.014 | 0.026 | 29.0 |
| **8780** | 2020-12-31 21:00:00 | 0.002 | 0.4 | 0.017 | 0.021 | 23.0 |
| **8781** | 2020-12-31 22:00:00 | 0.002 | 0.4 | 0.025 | 0.013 | 28.0 |
| **8782** | 2020-12-31 23:00:00 | 0.002 | 0.3 | 0.030 | 0.008 | 24.0 |
| **8783** | 2021-01-01 00:00:00 | 0.002 | 0.3 | 0.027 | 0.011 | 15.0 |

8784 rows × 6 columns

In [27]:

```
Total_Seoul_kma
```

Out[27]:

| | 측정일시 | Seoul_Temp(°C) | Seoul_Precipitation(mm) | Seoul_Wind_speed(m/s) | Seoul_Humidity |
|---|---|---|---|---|---|
| **0** | 2020-01-01 00:00:00 | -6.5 | 0.000 | 0.0 | |
| **1** | 2020-01-01 01:00:00 | -5.9 | 1.556 | 1.7 | |
| **2** | 2020-01-01 02:00:00 | -5.7 | 1.556 | 0.1 | |
| **3** | 2020-01-01 03:00:00 | -5.6 | 0.000 | 0.0 | |
| **4** | 2020-01-01 04:00:00 | -5.4 | 1.556 | 0.0 | |
| **...** | ... | ... | ... | ... | |
| **8779** | 2020-12-31 19:00:00 | -7.1 | 1.556 | 2.4 | |
| **8780** | 2020-12-31 20:00:00 | -7.1 | 1.556 | 3.2 | |
| **8781** | 2020-12-31 21:00:00 | -7.2 | 1.556 | 2.7 | |
| **8782** | 2020-12-31 22:00:00 | -7.4 | 1.556 | 2.5 | |
| **8783** | 2020-12-31 23:00:00 | -7.6 | 1.556 | 2.2 | |

8784 rows × 6 columns

In [28]:

```
Total_Data_Seoul = pd.merge(Total_Seoul_airkorea, Total_Seoul_kma, on = '측정일시', how = 'left')

Total_Data_Seoul
```

Out[28]:

| | 측정일시 | SO2 | CO | O3 | NO2 | PM10 | Seoul_Temp(°C) | Seoul_Precipitation(mm) | Seoul_ |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2020-01-01 01:00:00 | 0.002 | 0.5 | 0.011 | 0.024 | 19.0 | -5.9 | 1.556 | |
| **1** | 2020-01-01 02:00:00 | 0.002 | 0.6 | 0.005 | 0.030 | 19.0 | -5.7 | 1.556 | |
| **2** | 2020-01-01 03:00:00 | 0.002 | 0.6 | 0.002 | 0.033 | 27.0 | -5.6 | 0.000 | |
| **3** | 2020-01-01 04:00:00 | 0.002 | 0.6 | 0.003 | 0.031 | 20.0 | -5.4 | 1.556 | |
| **4** | 2020-01-01 05:00:00 | 0.002 | 0.7 | 0.003 | 0.031 | 21.0 | -5.2 | 1.556 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8779** | 2020-12-31 20:00:00 | 0.002 | 0.4 | 0.014 | 0.026 | 29.0 | -7.1 | 1.556 | |
| **8780** | 2020-12-31 21:00:00 | 0.002 | 0.4 | 0.017 | 0.021 | 23.0 | -7.2 | 1.556 | |
| **8781** | 2020-12-31 22:00:00 | 0.002 | 0.4 | 0.025 | 0.013 | 28.0 | -7.4 | 1.556 | |
| **8782** | 2020-12-31 23:00:00 | 0.002 | 0.3 | 0.030 | 0.008 | 24.0 | -7.6 | 1.556 | |
| **8783** | 2021-01-01 00:00:00 | 0.002 | 0.3 | 0.027 | 0.011 | 15.0 | NaN | NaN | |

8784 rows × 11 columns

In [29]:

```
Total_Data_Seoul = Total_Data_Seoul.drop(len(Total_Data_Seoul) - 1)

Total_Data_Seoul
```

Out[29]:

| | 측정일시 | SO2 | CO | O3 | NO2 | PM10 | Seoul_Temp(°C) | Seoul_Precipitation(mm) | Seoul_ |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2020-01-01 01:00:00 | 0.002 | 0.5 | 0.011 | 0.024 | 19.0 | -5.9 | 1.556 | |
| **1** | 2020-01-01 02:00:00 | 0.002 | 0.6 | 0.005 | 0.030 | 19.0 | -5.7 | 1.556 | |
| **2** | 2020-01-01 03:00:00 | 0.002 | 0.6 | 0.002 | 0.033 | 27.0 | -5.6 | 0.000 | |
| **3** | 2020-01-01 04:00:00 | 0.002 | 0.6 | 0.003 | 0.031 | 20.0 | -5.4 | 1.556 | |
| **4** | 2020-01-01 05:00:00 | 0.002 | 0.7 | 0.003 | 0.031 | 21.0 | -5.2 | 1.556 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8778** | 2020-12-31 19:00:00 | 0.002 | 0.4 | 0.016 | 0.023 | 26.0 | -7.1 | 1.556 | |
| **8779** | 2020-12-31 20:00:00 | 0.002 | 0.4 | 0.014 | 0.026 | 29.0 | -7.1 | 1.556 | |
| **8780** | 2020-12-31 21:00:00 | 0.002 | 0.4 | 0.017 | 0.021 | 23.0 | -7.2 | 1.556 | |
| **8781** | 2020-12-31 22:00:00 | 0.002 | 0.4 | 0.025 | 0.013 | 28.0 | -7.4 | 1.556 | |
| **8782** | 2020-12-31 23:00:00 | 0.002 | 0.3 | 0.030 | 0.008 | 24.0 | -7.6 | 1.556 | |

8783 rows × 11 columns

In [30]:

```
Total_Data_Seoul.to_csv('./Total_data_Seoul.csv', header = True, index = False)
```

In [37]:

```
# extract_inputoutput 함수 생성
def extract_inputoutput(dataframe, lookback_time = 3, predict_time = 1):
    dfx = pd.DataFrame()
    dfy = pd.DataFrame()

    for i in range(len(dataframe) - (lookback_time - 1) - (predict_time)):
        if i % 1000 == 0:
            print(i)


        # 독립변수에 대한 분류
        rowx = []

        for timestep in range(lookback_time):
            dfRename = dataframe.iloc[[i + timestep]]
            dfRename.index = [i]
            rowx.append(dfRename)

        rowx = pd.concat(rowx, axis = 1)
        dfx = pd.concat([dfx, rowx])
        ##


        # 종속변수에 대한 분류
        rowy = []
        rowy = pd.DataFrame([dataframe['PM10'][i + lookback_time]])
        dfy = pd.concat([dfy, rowy], ignore_index = True)
        ##


    print("x, Y 데이터 분류 완료!")
    return dfx, dfy
##
```

In [38]:

```
X, Y = extract_inputoutput(Total_Data_Seoul)
```

```
0
1000
2000
3000
4000
5000
6000
7000
8000
x, Y 데이터 분류 완료!
```

In [39]:

```
X
```

Out[39]:

| | 측정일시 | SO2 | CO | O3 | NO2 | PM10 | Seoul_Temp(°C) | Seoul_Precipitation(mm) | Seoul_ |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 2020-01-01 01:00:00 | 0.002 | 0.5 | 0.011 | 0.024 | 19.0 | -5.9 | 1.556 | |
| **1** | 2020-01-01 02:00:00 | 0.002 | 0.6 | 0.005 | 0.030 | 19.0 | -5.7 | 1.556 | |
| **2** | 2020-01-01 03:00:00 | 0.002 | 0.6 | 0.002 | 0.033 | 27.0 | -5.6 | 0.000 | |
| **3** | 2020-01-01 04:00:00 | 0.002 | 0.6 | 0.003 | 0.031 | 20.0 | -5.4 | 1.556 | |
| **4** | 2020-01-01 05:00:00 | 0.002 | 0.7 | 0.003 | 0.031 | 21.0 | -5.2 | 1.556 | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | |
| **8775** | 2020-12-31 16:00:00 | 0.002 | 0.3 | 0.031 | 0.008 | 20.0 | -5.5 | 1.556 | |
| **8776** | 2020-12-31 17:00:00 | 0.002 | 0.3 | 0.029 | 0.010 | 24.0 | -6.1 | 1.556 | |
| **8777** | 2020-12-31 18:00:00 | 0.002 | 0.3 | 0.027 | 0.011 | 24.0 | -6.7 | 0.000 | |
| **8778** | 2020-12-31 19:00:00 | 0.002 | 0.4 | 0.016 | 0.023 | 26.0 | -7.1 | 1.556 | |
| **8779** | 2020-12-31 20:00:00 | 0.002 | 0.4 | 0.014 | 0.026 | 29.0 | -7.1 | 1.556 | |

8780 rows × 33 columns

In [40]:

```
Y
```

Out[40]:

| | 0 |
|---|---|
| **0** | 20.0 |
| **1** | 21.0 |
| **2** | 23.0 |
| **3** | 22.0 |
| **4** | 21.0 |
| **...** | ... |
| **8775** | 26.0 |
| **8776** | 29.0 |
| **8777** | 23.0 |
| **8778** | 28.0 |
| **8779** | 24.0 |

8780 rows × 1 columns

In [41]:

```
X.to_csv("./data_result/Total_Data_Seoul_X.csv", header = True, index = False)
```

In [42]:

```
Y.to_csv("./data_result/Total_Data_Seoul_Y.csv", header = True, index = False)
```