0. Introduction

> "데이터 과학의 80%는 데이터 클리닝에 소비되고, 나머지 20%는 데이터 클리닝하는 시간을 불평하는데 쓰인다."
> - Kaggle 창립자

그만큼 데이터 전처리에 들이는 노력이 상당합니다.

EDA 탐색적 데이터 분석

데이터 출처와 주제에 대한 이해

- 서울의 공기 오염도 데이터
- 데이터 출처 : 서울 열린데이터 광장

데이터의 구조 확인

- `head()` 로 데이터의 형태 확인
- `shape` 로 row * column 사이즈 확인
- `isnull()` 으로 결측치 확인
    - NA나 NULL값을 확인하고 제거 혹은 평균값 대입

데이터의 Feature 이해

1. 각 column이 무엇을 나타내고 있는지
    - Measurement date, Station code, Latitude, Longitude, PM2.5, PM10
    - 데이터의 범위 확인 max, min, meaㅜ
2. 속성간의 상관관계 확인
    - PM10 & PM2.5와 SO2, NO2, O3, CO2와의 상관관계
    - 시간별 미세먼지 농도
3. 시각화를 통해 모델에 적합한 데이터 추출

1. Library & Data Load

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
'''라이브러리 불러오기'''
import numpy as np
import pandas as pd
import matplotlib
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

```
'''데이터 불러오기'''
df = pd.read_csv('/content/drive/MyDrive/한국분석/air_pollution_in_seoul/AirPollutionSeoul/Measurement_summary.csv')
df.head()
```

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.056	0.002	1.2	70.0	59.0

2. Data

```
'''데이터의 차원'''
df.shape
```

```
(647511, 11)
```

```
'''Column 데이터 출력'''
df.columns.tolist()
```

```
['Measurement date',
 'Station code',
 'Address',
 'Latitude',
 'Longitude',
 'SO2',
 'NO2',
 'O3',
 'CO',
 'PM10',
 'PM2.5']
```

```
'''데이터 정보'''
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 647511 entries, 0 to 647510
Data columns (total 11 columns):
 #   Column            Non-Null Count   Dtype  
---  ------            --------------   -----  
 0   Measurement date  647511 non-null  object 
 1   Station code      647511 non-null  int64  
 2   Address           647511 non-null  object 
 3   Latitude          647511 non-null  float64
 4   Longitude         647511 non-null  float64
 5   SO2               647511 non-null  float64
 6   NO2               647511 non-null  float64
 7   O3                647511 non-null  float64
 8   CO                647511 non-null  float64
 9   PM10              647511 non-null  float64
 10  PM2.5             647511 non-null  float64
dtypes: float64(8), int64(1), object(2)
memory usage: 54.3+ MB
```

```
'''결측치 확인'''
pd.isnull(df)
df.isnull().sum()
```

	0
Measurement date	0
Station code	0
Address	0
Latitude	0
Longitude	0
SO2	0
NO2	0
O3	0
CO	0
PM10	0
PM2.5	0

dtype: int64

```
'''각 열들의 고유값 정보 출력'''
df.nunique()
```

```
                       0
Measurement date   25906
      Station code    25
           Address    25
          Latitude    25
         Longitude    25
               SO2   186
               NO2   132
                O3   253
                CO   172
              PM10   551
             PM2.5   333

dtype: int64
```

```
'''중복된 데이터 확인'''
df.duplicated()
```

```
             0
      0  False
      1  False
      2  False
      3  False
      4  False
    ...    ...
 647506  False
 647507  False
 647508  False
 647509  False
 647510  False
647511 rows × 1 columns

dtype: bool
```

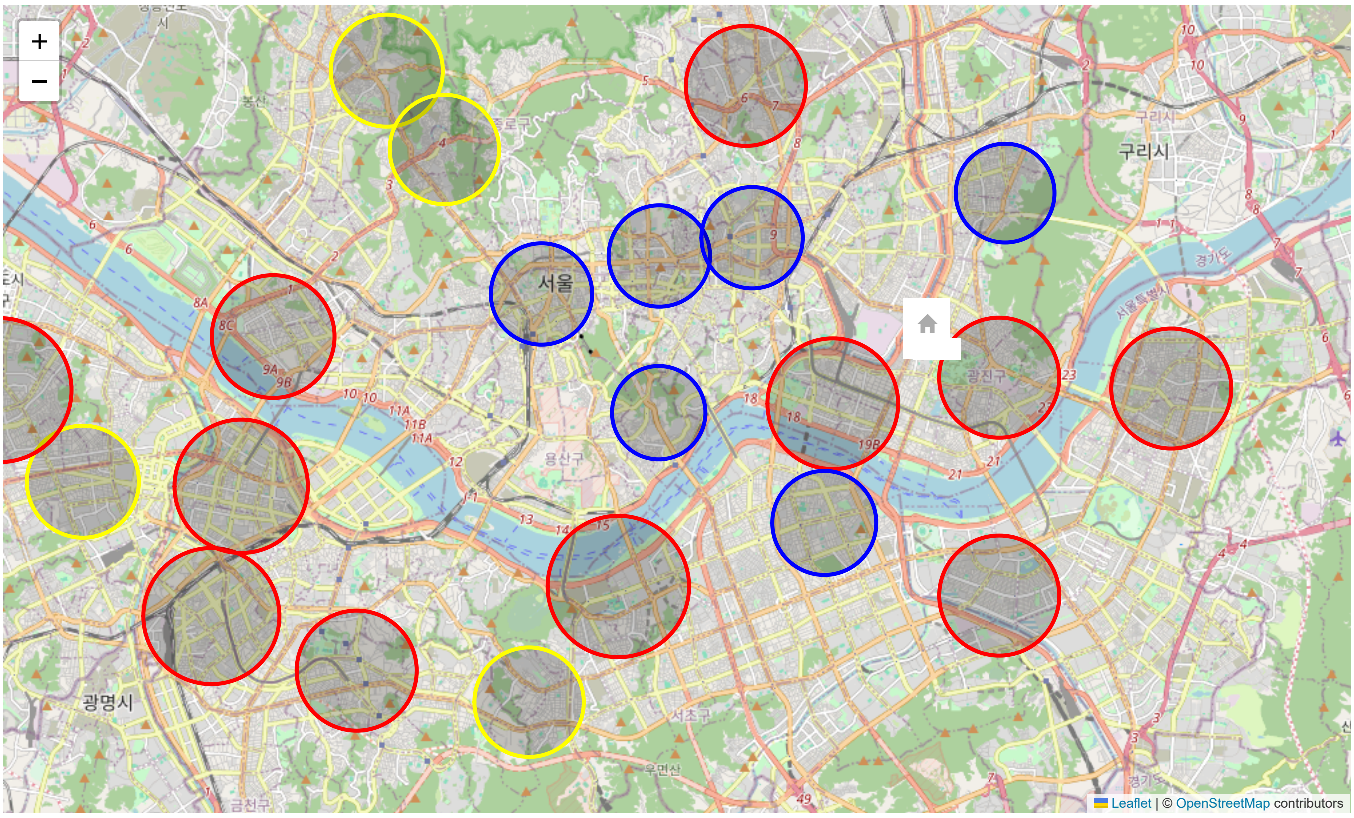
3.1. 위도 & 경도 데이터

```
# 위도 경도 DataFrame
location = df.groupby('Station code')['PM10'].agg([np.mean])
location['Latitude'] = df['Latitude'].unique() # 절대 이렇게 코드짜면 안되요!
location['Longitude'] = df['Longitude'].unique()
location.head()
```

```
<ipython-input-11-72f461a0b840>:2: FutureWarning: The provided callable <function mean at 0x7d646b93fd90> is currently using SeriesGroupBy.mean.
  location = df.groupby('Station code')['PM10'].agg([np.mean])
                   mean   Latitude   Longitude
Station code
         101  37.965605  37.572016  127.005008
         102  37.970469  37.564263  126.974676
         103  35.539183  37.540033  127.004850
         104  42.328468  37.609823  126.934848
         105  41.437737  37.593742  126.949679
```

다음 단계: location변수로 코드 생성 | 추천 차트 보기 | New interactive sheet

```
import folium
from folium.plugins import MarkerCluster

# PM10에 따른 color 변화
def color_select(x):
    if x >= 45:
        return 'red'
    elif x >= 40:
        return 'yellow'
    else:
        return 'blue'

# Map
seoul = folium.Map(location=[37.55138077230307, 126.98712254969668], zoom_start=12)

# Circle
for i in range(len(location)):
    # 관측소
    folium.Circle(location=[location.iloc[i,1], location.iloc[i,2]], radius = location.iloc[i, 0]*30, color=color_select(location.iloc[i,0]),fill_col

# Marker / Sejong Univ.
folium.Marker([37.55195608145124, 127.07362532752212], icon=folium.Icon(popup='Sejoing Univ.', color='red', icon='glyphicon glyphicon-home')).add_to(
seoul
```



3.2. 시간별 미세먼지 농도

```
from datetime import datetime

df['Measurement date'] = df['Measurement date'].astype('datetime64[ns]')
df['hour'] = df.loc[:, "Measurement date"].dt.hour
df = df.drop('Measurement date', axis=1)
```

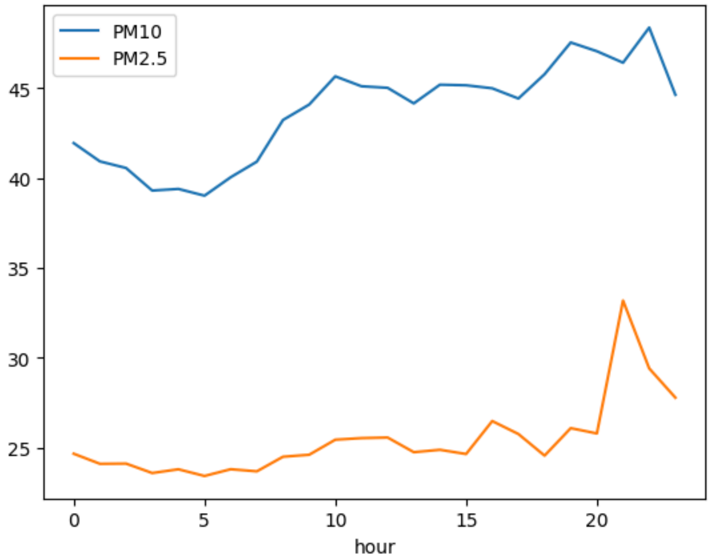
```
data = df.groupby('hour', as_index=False).agg({'SO2':'mean', 'NO2':'mean', 'O3':'mean', 'CO':'mean', 'PM10':'mean', 'PM2.5':'mean'})
data
```

	hour	SO2	NO2	O3	CO	PM10	PM2.5
0	0	-0.001909	0.024584	0.011626	0.529891	41.944368	24.651817
1	1	-0.002126	0.021533	0.012065	0.524424	40.927181	24.091507
2	2	-0.002047	0.019609	0.012448	0.516841	40.558962	24.103003
3	3	-0.002203	0.018209	0.012198	0.509860	39.303351	23.577686
4	4	-0.002267	0.018247	0.011033	0.507101	39.394380	23.785717
5	5	-0.002303	0.020649	0.008226	0.515338	39.018339	23.413360
6	6	-0.001988	0.025026	0.005753	0.536940	40.044229	23.788876
7	7	-0.001966	0.027204	0.005840	0.562405	40.911421	23.670649
8	8	-0.001507	0.028721	0.008698	0.576376	43.231577	24.484415
9	9	-0.001398	0.027778	0.014076	0.559335	44.089890	24.593978
10	10	-0.001183	0.024440	0.019282	0.529488	45.661971	25.433010
11	11	-0.001360	0.020498	0.022907	0.500881	45.108788	25.520666
12	12	-0.001286	0.018103	0.028270	0.482626	45.022000	25.553704
13	13	-0.001212	0.016675	0.031950	0.470215	44.155556	24.737778
14	14	-0.001377	0.016059	0.034195	0.459807	45.197106	24.870575
15	15	-0.001595	0.016763	0.034715	0.438539	45.164152	24.636552
16	16	-0.001363	0.018260	0.032921	0.446770	44.994926	26.470481
17	17	-0.001417	0.021212	0.029132	0.455848	44.426963	25.748778
18	18	-0.001553	0.024476	0.023835	0.481205	45.778767	24.546031
19	19	-0.001558	0.026721	0.019516	0.507660	47.546279	26.075595
20	20	-0.001681	0.027166	0.016499	0.522186	47.059484	25.781436
21	21	-0.001718	0.027119	0.014502	0.528154	46.420701	33.179113
22	22	-0.001829	0.027196	0.012546	0.531633	48.375496	29.401208
23	23	-0.004239	0.024149	0.009250	0.526959	44.636799	27.775472

다음 단계: data변수로 코드 생성 | 추천 차트 보기 | New interactive sheet

```
# 미세먼지 농도변화 Hour
data.plot(x='hour', y=['PM10', 'PM2.5'])
```

```
<Axes: xlabel='hour'>
```



3.2.1. 세종대 주변 미세먼지 농도

```
df_sj = pd.read_csv('/content/drive/MyDrive/한국분석/air_pollution_in_seoul/AirPollutionSeoul/Measurement_summary.csv')
## 데이터 다시 가져오기
df_sj.head()
```

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.056	0.002	1.2	70.0	59.0

```
df_sj_date = df_sj['Measurement date'].str.split(" ",n=1,expand=True)
## 시간 날짜를 분리
df_sj_date.head()
```

	0	1
0	2017-01-01	00:00
1	2017-01-01	01:00
2	2017-01-01	02:00
3	2017-01-01	03:00
4	2017-01-01	04:00

```
df_sj['date'] = df_sj_date[0]
df_sj['time'] = df_sj_date[1]
df_sj = df_sj.drop(['Measurement date'],axis = 1)
df_sj.head()
## 시간 날짜 붙여놓고, 쓸모없는 데이터 버리기
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	date	time
0	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.059	0.002	1.2	73.0	57.0	2017-01-01	00:00
1	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.058	0.002	1.2	71.0	59.0	2017-01-01	01:00
2	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.056	0.002	1.2	70.0	59.0	2017-01-01	02:00

```
condition = (df_sj.date == '2019-04-03')
df_birth = df_sj[condition]
df_birth.head()
## 특정 시간 기준으로 추출
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	date	time
19505	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.003	0.026	0.035	0.4	30.0	18.0	2019-04-03	00:00
19506	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.003	0.026	0.033	0.5	29.0	17.0	2019-04-03	01:00
19507	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.003	0.025	0.035	0.5	31.0	21.0	2019-04-03	02:00

```
cheak = df_birth['Address'].unique()
cheak
## 세종대 주변 위치 찾기
```

```
array(['19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republic of Korea',
       '15, Deoksugung-gil, Jung-gu, Seoul, Republic of Korea',
       '136, Hannam-daero, Yongsan-gu, Seoul, Republic of Korea',
       '215, Jinheung-ro, Eunpyeong-gu, Seoul, Republic of Korea',
       '32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul, Republic of Korea',
       '10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic of Korea',
       '18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Republic of Korea',
       '571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republic of Korea',
       '43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul, Republic of Korea',
       '369, Yongmasan-ro, Jungnang-gu, Seoul, Republic of Korea',
       '70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Republic of Korea',
       '49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Republic of Korea',
       '34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Republic of Korea',
       '17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republic of Korea',
       '56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Republic of Korea',
       '71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Republic of Korea',
       '45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republic of Korea',
       '20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Republic of Korea',
       '11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul, Republic of Korea',
       '6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Republic of Korea',
       '14, Sillimdong-gil, Gwanak-gu, Seoul, Republic of Korea',
       '16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Republic of Korea',
       '426, Hakdong-ro, Gangnam-gu, Seoul, Republic of Korea',
       '236, Baekjegobun-ro, Songpa-gu, Seoul, Republic of Korea',
       '59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul, Republic of Korea'],
      dtype=object)
```

```
condition = (df_birth.Address == '571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republic of Korea')
df_add = df_birth[condition]
df_add.head()
## 광진구 기준으로 데이터 추출
```

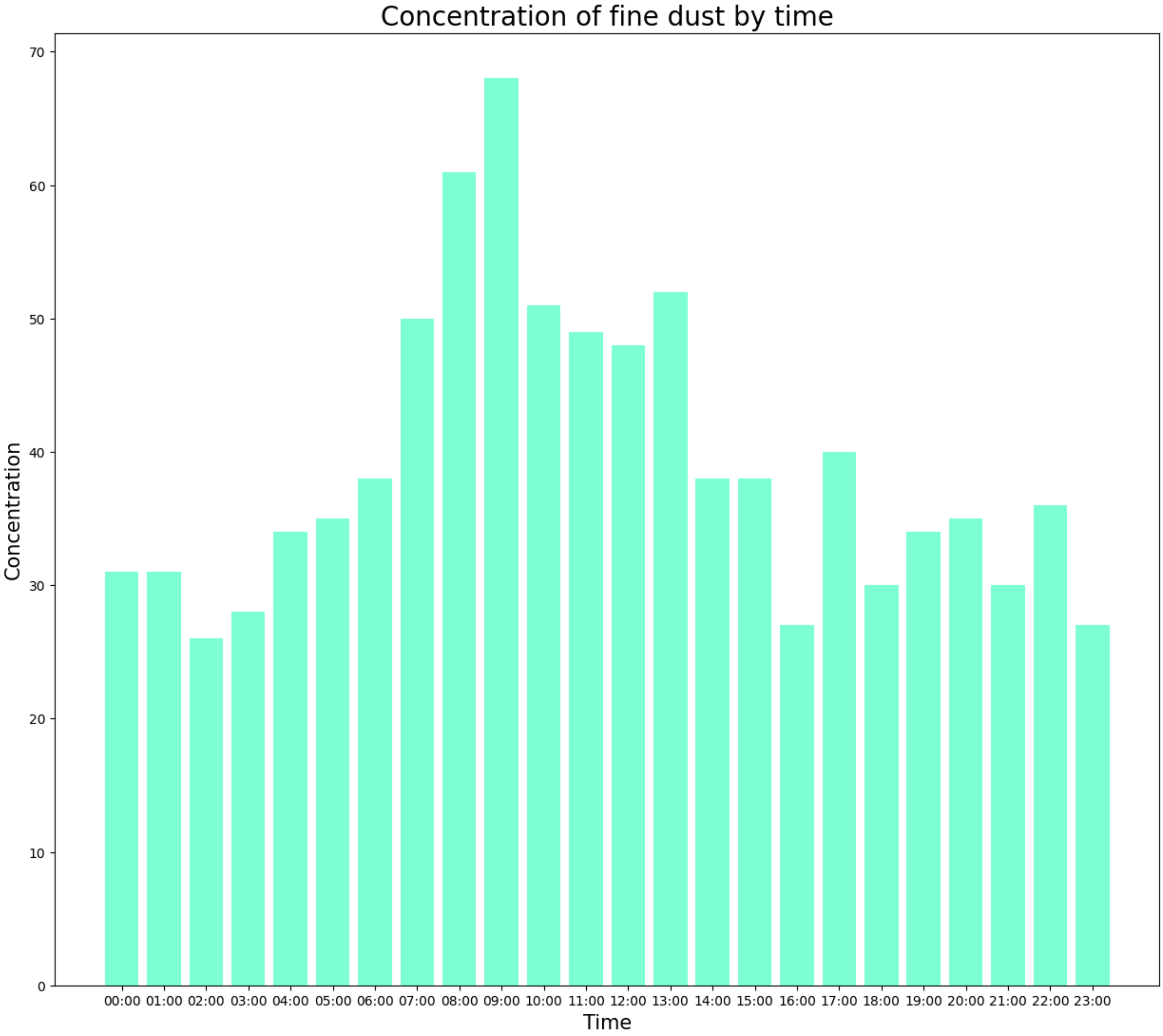
	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	date	time
200801	108	571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...	37.54718	127.092493	0.004	0.029	0.027	0.6	31.0	24.0	2019-04-03	00:00
200802	108	571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...	37.54718	127.092493	0.004	0.026	0.029	0.6	31.0	18.0	2019-04-03	01:00
200803	108	571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...	37.54718	127.092493	0.004	0.021	0.035	0.6	26.0	18.0	2019-04-03	02:00

```
df_add = df_add.loc[:,['SO2','NO2','O3','CO','PM10','PM2.5','time']]
df_add.head()
## 원하는 컬럼들로만 데이터프레임 다시 만들기
```

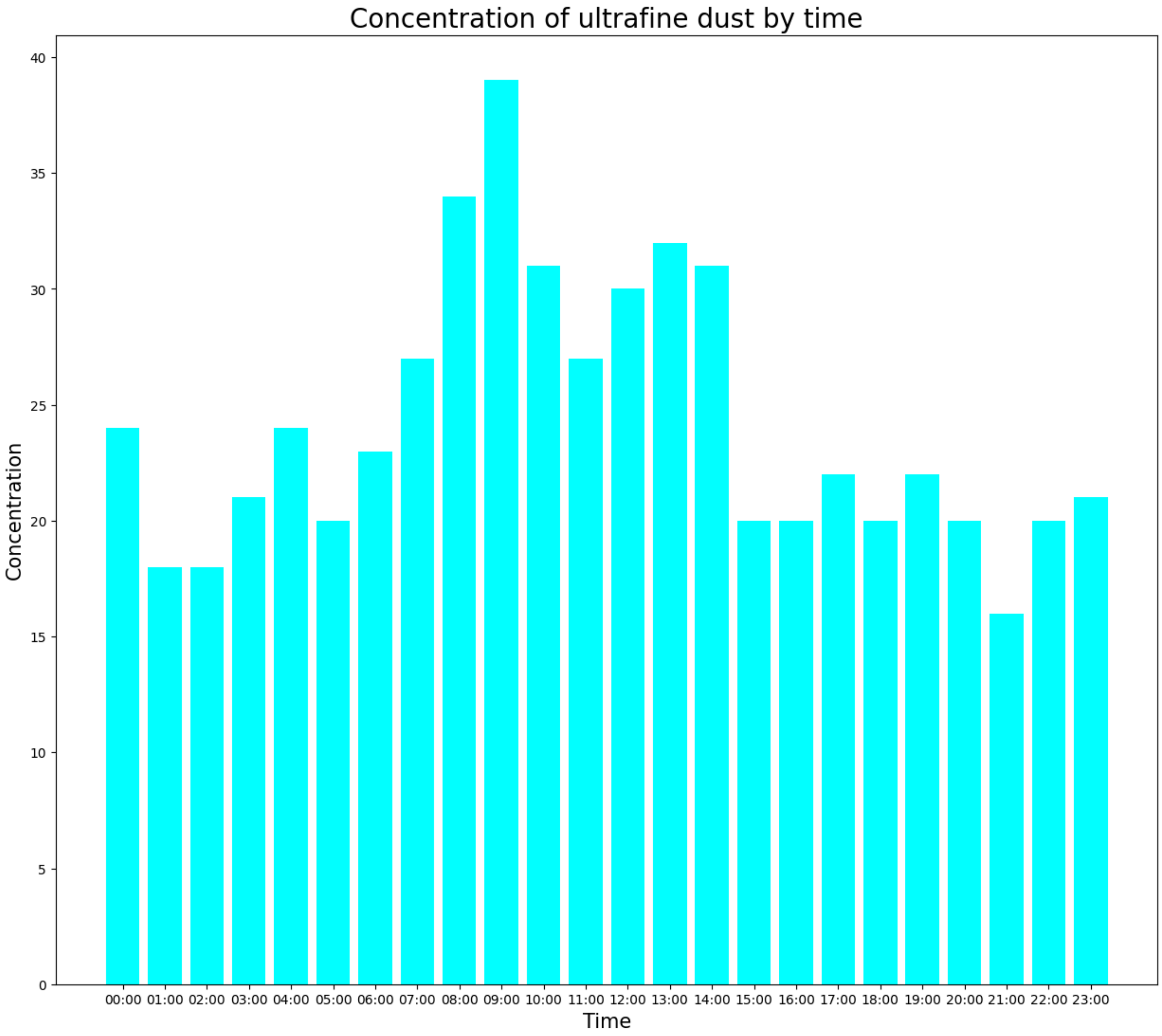
	SO2	NO2	O3	CO	PM10	PM2.5	time
200801	0.004	0.029	0.027	0.6	31.0	24.0	00:00
200802	0.004	0.026	0.029	0.6	31.0	18.0	01:00
200803	0.004	0.021	0.035	0.6	26.0	18.0	02:00
200804	0.004	0.025	0.028	0.6	28.0	21.0	03:00
200805	0.004	0.043	0.004	0.7	34.0	24.0	04:00

```
X_sj = df_add['time']
Y_sj = df_add['PM10']
Y2_sj = df_add['PM2.5']
## 그래프에 나타낼 데이터 추출하기
```

```
plt.figure(figsize = (15,13))
plt.bar(X_sj,Y_sj,color = 'aquamarine')
plt.title('Concentration of fine dust by time',fontsize = 20)
plt.xlabel('Time',fontsize=15)
plt.ylabel('Concentration',fontsize = 15)
plt.show()

## 세종대 주변 시간별 미세먼지 그래프
```



```
plt.figure(figsize = (15,13))
plt.bar(X_sj,Y2_sj,color = 'cyan')
plt.title('Concentration of ultrafine dust by time',fontsize = 20)
plt.xlabel('Time',fontsize=15)
plt.ylabel('Concentration',fontsize = 15)
plt.show()

## 세종대 주변 시간별 초미세먼지 시간별 그래프
```



3.3. 지역별 미세먼지 농도

SO2

```
'''SO2 비율이 높은 정보 10개 출력'''
SO2 = df.sort_values(by = ['SO2'], ascending=False)
SO2.head(10)
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	hour
424709	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	3.736	38.445	12.455	0.4	35.0	17.0	9
424686	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	2.700	20.100	33.600	0.3	8.0	1.0	10
424685	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	2.700	30.700	23.400	0.4	5.0	6.0	9
424710	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	1.330	12.805	6.320	0.5	34.0	15.0	10
15589	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.406	0.044	0.003	40.0	22.0	12.0	13
644605	125	59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...	37.544962	127.136792	0.378	-1.000	0.002	36.7	14.0	1.0	14

```
SO2_Address = df.groupby('Address').agg({'SO2' : 'median'}).sort_values('SO2',ascending=False).reset_index()
# Address를 기준으로 그룹화하여 SO2 집단별 평균으로 내림차순으로 정렬
# reset_index -> 인덱스 리셋(단순한 정수 인덱스로 세팅)
print(SO2_Address)
```

```
                                              Address    SO2
0   369, Yongmasan-ro, Jungnang-gu, Seoul, Republi...  0.006
1   71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...  0.005
2   45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...  0.005
3   43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul...  0.005
4   426, Hakdong-ro, Gangnam-gu, Seoul, Republic o...  0.005
5   236, Baekjegobun-ro, Songpa-gu, Seoul, Republi...  0.004
6   59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...  0.004
7   571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...  0.004
8   56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Re...  0.004
9   34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...  0.004
10  11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...  0.004
11  10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...  0.004
12  215, Jinheung-ro, Eunpyeong-gu, Seoul, Republi...  0.004
13  20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Rep...  0.004
14  19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...  0.004
15  18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Re...  0.004
16  17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republ...  0.004
17  16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Repu...  0.004
18  14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...  0.004
19  32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul,...  0.004
20  49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Rep...  0.003
21  15, Deoksugung-gil, Jung-gu, Seoul, Republic o...  0.003
22  136, Hannam-daero, Yongsan-gu, Seoul, Republic...  0.003
23  6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Repub...  0.003
24  70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...  0.003
```

중랑구 용마산로의 **SO2 비율**이 가장 높은 것을 볼 수 있습니다.

```
# 상위 10개 데이터만 저장
SO2 = SO2_Address.sort_values('SO2',ascending=False).head(10)
```

NO2

```
'''NO2 비율이 높은 정보 10개 출력'''
NO2 = df.sort_values(by = ['NO2'], ascending=False)
NO2.head(10)
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	hour
424709	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	3.736	38.445	12.455	0.4	35.0	17.0	9
424684	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	0.000	37.500	0.000	0.0	0.0	0.0	8
424685	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	2.700	30.700	23.400	0.4	5.0	6.0	9
424686	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	2.700	20.100	33.600	0.3	8.0	1.0	10
424710	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	1.330	12.805	6.320	0.5	34.0	15.0	10
485104	119	11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...	37.525007	126.897370	0.010	0.310	0.035	1.1	74.0	51.0	17
		15, Deoksugung-gil, Jung-gu, Seoul,									

```
NO2_Address = df.groupby('Address').agg({'NO2' : 'median'}).sort_values('NO2',ascending=False).reset_index()
# Address를 기준으로 그룹화하여 NO2 집단별 평균으로 내림차순으로 정렬
# reset_index -> 인덱스 리셋(단순한 정수 인덱스로 세팅)
print(NO2_Address)
```

```
                                              Address    NO2
0    15, Deoksugung-gil, Jung-gu, Seoul, Republic o...  0.030
1    19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...  0.028
2    136, Hannam-daero, Yongsan-gu, Seoul, Republic...  0.028
3    70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...  0.028
4    56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Re...  0.028
5    236, Baekjegobun-ro, Songpa-gu, Seoul, Republi...  0.027
6    11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...  0.027
7    20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Rep...  0.027
8    426, Hakdong-ro, Gangnam-gu, Seoul, Republic o...  0.026
9    59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...  0.026
10   71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...  0.026
11   6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Repub...  0.026
12   18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Re...  0.026
13   16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Repu...  0.026
14   14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...  0.026
15   43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul...  0.025
16   571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...  0.024
17   10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...  0.023
18   369, Yongmasan-ro, Jungnang-gu, Seoul, Republi...  0.023
19   17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republ...  0.023
20   32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul,...  0.022
21   45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...  0.021
22   215, Jinheung-ro, Eunpyeong-gu, Seoul, Republi...  0.021
23   34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...  0.018
24   49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Rep...  0.017
```

중구 덕수궁길의 **NO2 비율**이 가장 높은 것을 볼 수 있습니다.

```
# 상위 10개 데이터만 저장
NO2 = NO2_Address.sort_values('NO2',ascending=False).head(10)
```

O3

```
'''O3 비율이 높은 정보 10개 출력'''
O3 = df.sort_values(by = ['O3'], ascending=False)
O3.head(10)
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	hour
424686	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	2.700	20.100	33.600	0.3	8.0	1.0	10
424685	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	2.700	30.700	23.400	0.4	5.0	6.0	9
424709	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	3.736	38.445	12.455	0.4	35.0	17.0	9
424710	117	45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...	37.498498	126.889692	1.330	12.805	6.320	0.5	34.0	15.0	10
263389	111	70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...	37.606719	127.027279	0.002	0.021	5.297	0.4	14.0	8.0	21
263386	111	70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...	37.606719	127.027279	0.003	0.026	1.901	0.5	36.0	29.0	18
		70, Samyang-ro 2-gil, Seongbuk-gu, Seoul,									

```
O3_Address = df.groupby('Address').agg({'O3' : 'median'}).sort_values('O3',ascending=False).reset_index()
# Address를 기준으로 그룹화하여 O3 집단별 평균으로 내림차순으로 정렬
# reset_index -> 인덱스 리셋(단순한 정수 인덱스로 세팅)
print(O3_Address)
```

```
                                          Address     O3
0    49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Rep...  0.027
1    215, Jinheung-ro, Eunpyeong-gu, Seoul, Republi...  0.025
2    34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...  0.025
3    32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul,...  0.023
4    71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...  0.023
5    16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Repu...  0.022
6    6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Repub...  0.022
7    19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...  0.022
8    45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...  0.022
9    15, Deoksugung-gil, Jung-gu, Seoul, Republic o...  0.022
10   17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republ...  0.021
11   10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...  0.021
12   20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Rep...  0.021
13   369, Yongmasan-ro, Jungnang-gu, Seoul, Republi...  0.020
14   571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...  0.020
15   70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...  0.020
16   11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...  0.019
17   14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...  0.019
18   236, Baekjegobun-ro, Songpa-gu, Seoul, Republi...  0.019
19   43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul...  0.019
20   56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Re...  0.019
21   59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...  0.019
22   136, Hannam-daero, Yongsan-gu, Seoul, Republic...  0.018
23   18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Re...  0.018
24   426, Hakdong-ro, Gangnam-gu, Seoul, Republic o...  0.017
```

강북구 삼양로 139길 지역에서 **O3 비율** 이 높은 것을 알 수 있다.

```
# 상위 10개 데이터만 저장
O3 = O3_Address.sort_values('O3',ascending=False).head(10)
```

CO

```
'''CO 비율이 높은 정보 10개 출력'''
CO = df.sort_values(by = ['CO'], ascending=False)
CO.head(10)
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	hour
311153	113	34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...	37.654192	127.029088	0.003	0.030	0.011	71.7	38.0	9.0	20
311154	113	34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...	37.654192	127.029088	0.003	0.019	0.019	69.1	31.0	16.0	21
311155	113	34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...	37.654192	127.029088	0.004	0.038	0.009	59.3	41.0	20.0	22
311152	113	34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...	37.654192	127.029088	0.003	0.039	0.008	47.2	30.0	21.0	19
15589	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.406	0.044	0.003	40.0	22.0	12.0	13
15565	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.372	0.030	0.030	38.4	15.0	4.0	13
		571, Gwangnaru-ro, Gwangjin-gu, Seoul,									

```
CO_Address = df.groupby('Address').agg({'CO' : 'median'}).sort_values('CO',ascending=False).reset_index()
# Address를 기준으로 그룹화하여 CO 집단별 평균으로 내림차순으로 정렬
# reset_index -> 인덱스 리셋(단순한 정수 인덱스로 세팅)
print(CO_Address)
```

```
                                             Address   CO
0   70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...  0.6
1   571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...  0.6
2   10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...  0.5
3   215, Jinheung-ro, Eunpyeong-gu, Seoul, Republi...  0.5
4   59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...  0.5
5   56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Re...  0.5
6   43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul...  0.5
7   34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...  0.5
8   11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...  0.5
9   236, Baekjegobun-ro, Songpa-gu, Seoul, Republi...  0.5
10  32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul,...  0.5
11  19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...  0.5
12  17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republ...  0.5
13  15, Deoksugung-gil, Jung-gu, Seoul, Republic o...  0.5
14  20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Rep...  0.4
15  18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Re...  0.4
16  369, Yongmasan-ro, Jungnang-gu, Seoul, Republi...  0.4
17  426, Hakdong-ro, Gangnam-gu, Seoul, Republic o...  0.4
18  45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...  0.4
19  49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Rep...  0.4
20  16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Repu...  0.4
21  14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...  0.4
22  6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Repub...  0.4
23  136, Hannam-daero, Yongsan-gu, Seoul, Republic...  0.4
24  71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...  0.4
```

성북구 삼양로 2길 지역에서 **CO 비율** 이 높은 것을 알 수 있다.

```
# 상위 10개 데이터만 저장
CO = CO_Address.sort_values('CO',ascending=False).head(10)
```

PM10

더블클릭 또는 Enter 키를 눌러 수정

```
'''PM10 비율이 높은 정보 10개 출력'''
PM10 = df.sort_values(by = ['PM10'], ascending=False)
PM10.head(10)
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	hour
397291	116	71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...	37.54464	126.835151	0.007	0.058	0.003	1.1	3586.0	23.0	9
397290	116	71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...	37.54464	126.835151	0.005	0.059	0.002	1.3	3577.0	28.0	8
397289	116	71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...	37.54464	126.835151	0.005	0.057	0.002	1.3	3568.0	23.0	7
397288	116	71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...	37.54464	126.835151	0.005	0.055	0.002	1.2	3561.0	19.0	6
397287	116	71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...	37.54464	126.835151	0.004	0.048	0.002	1.1	3556.0	16.0	5
397286	116	71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...	37.54464	126.835151	0.004	0.052	0.003	1.3	3552.0	15.0	4
		71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul,									

```
PM10_Address = df.groupby('Address').agg({'PM10' : 'median'}).sort_values('PM10',ascending=False).reset_index()
# Address를 기준으로 그룹화하여 PM3 집단별 평균으로 내림차순으로 정렬
# reset_index -> 인덱스 리셋(단순한 정수 인덱스로 세팅)
print(PM10_Address)
```

```
                                              Address  PM10
0    11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...  41.0
1    59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...  39.0
2    70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...  38.0
3    18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Re...  38.0
4    71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...  37.0
5    14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...  37.0
6    16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Repu...  37.0
7    6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Repub...  36.0
8    56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Re...  36.0
9    45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...  36.0
10   10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...  36.0
11   236, Baekjegobun-ro, Songpa-gu, Seoul, Republi...  35.0
12   215, Jinheung-ro, Eunpyeong-gu, Seoul, Republi...  35.0
13   571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...  35.0
14   17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republ...  35.0
15   34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...  34.0
16   20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Rep...  34.0
17   32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul,...  34.0
18   426, Hakdong-ro, Gangnam-gu, Seoul, Republic o...  33.0
19   369, Yongmasan-ro, Jungnang-gu, Seoul, Republi...  32.0
20   43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul...  32.0
21   49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Rep...  32.0
22   19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...  32.0
23   15, Deoksugung-gil, Jung-gu, Seoul, Republic o...  32.0
24   136, Hannam-daero, Yongsan-gu, Seoul, Republic...  30.0
```

영동포구 양산로 23길 지역에서 **PM10 비율** 이 높은 것을 알 수 있다.

```
# 상위 10개 데이터만 저장
PM10 = PM10_Address.sort_values('PM10',ascending=False).head(10)
```

PM2.5

```
'''PM2.5 비율이 높은 정보 10개 출력'''
PM2_5 = df.sort_values(by = ['PM2.5'], ascending=False)
PM2_5.head(10)
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	hour
62770	103	136, Hannam-daero, Yongsan-gu, Seoul, Republic...	37.540033	127.004850	0.004	0.018	0.048	0.3	68.0	6256.0	16
281354	111	70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...	37.606719	127.027279	0.003	0.028	0.033	0.6	33.0	995.0	13
281426	111	70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...	37.606719	127.027279	0.003	0.025	0.048	0.5	985.0	995.0	13
140880	106	10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...	37.555580	126.905597	0.005	0.054	0.001	0.7	77.0	985.0	4
535694	121	14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...	37.487355	126.927102	0.004	0.020	0.022	0.3	77.0	985.0	19
480333	119	11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...	37.525007	126.897370	0.004	0.017	0.059	0.5	29.0	985.0	21
		16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul,									

```
PM2_5_Address = df.groupby('Address').agg({'PM2.5' : 'median'}).sort_values('PM2.5',ascending=False).reset_index()
# Address를 기준으로 그룹화하여 PM2.5 집단별 평균으로 내림차순으로 정렬
# reset_index -> 인덱스 리셋(단순한 정수 인덱스로 세팅)
print(PM2_5_Address)
```

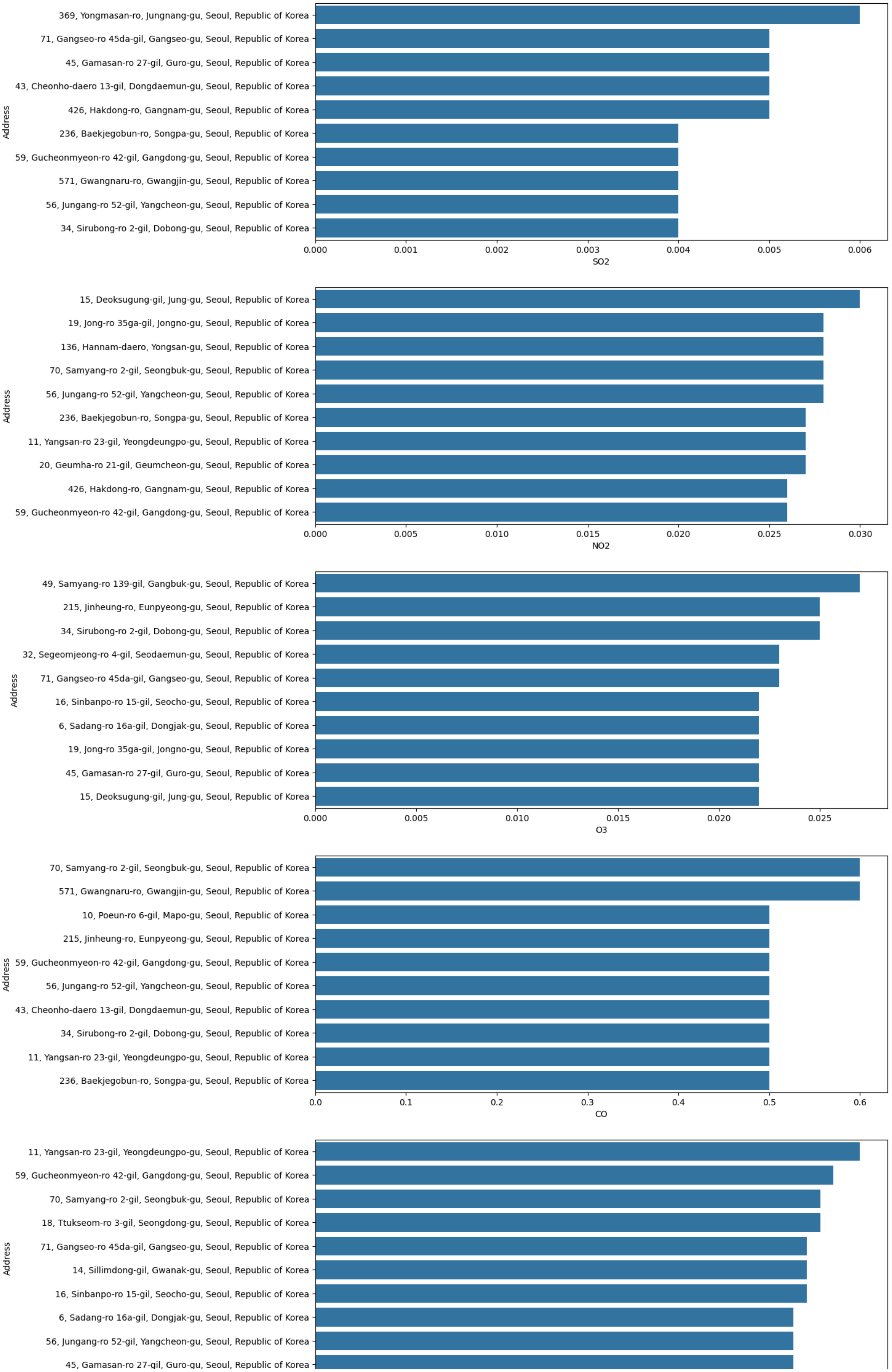
```
                                              Address  PM2.5
0    11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul,...   22.0
1    10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic o...   21.0
2    14, Sillimdong-gil, Gwanak-gu, Seoul, Republic...   21.0
3    56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Re...   21.0
4    6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Repub...   20.0
5    59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul...   20.0
6    17, Sanggye-ro 23-gil, Nowon-gu, Seoul, Republ...   20.0
7    18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Re...   20.0
8    20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Rep...   20.0
9    215, Jinheung-ro, Eunpyeong-gu, Seoul, Republi...   20.0
10   426, Hakdong-ro, Gangnam-gu, Seoul, Republic o...   19.0
11   70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Repu...   19.0
12   571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republi...   19.0
13   45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republi...   19.0
14   71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Re...   19.0
15   16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Repu...   19.0
16   136, Hannam-daero, Yongsan-gu, Seoul, Republic...   19.0
17   369, Yongmasan-ro, Jungnang-gu, Seoul, Republi...   18.0
18   43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul...   18.0
19   34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Repub...   18.0
20   236, Baekjegobun-ro, Songpa-gu, Seoul, Republi...   18.0
21   19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...   18.0
22   15, Deoksugung-gil, Jung-gu, Seoul, Republic o...   18.0
23   32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul,...   18.0
24   49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Rep...   17.0
```

영동포구 양산로 23길 지역에서 **PM2.5 비율**이 높은 것을 알 수 있다.

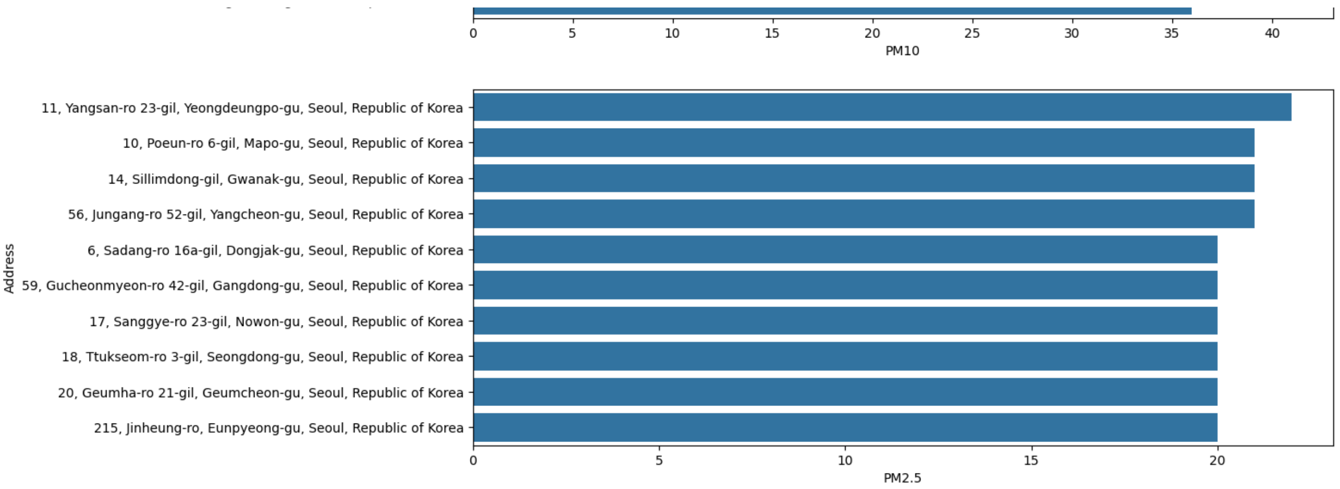
```
# 상위 10개 데이터만 저장
PM2_5 = PM2_5_Address.sort_values('PM2.5',ascending=False).head(10)
```

```
plt.figure(figsize=(12,35))

plt.subplot(6,1,1)
sns.barplot(y="Address", x="SO2", data = SO2_Address.head(10))

plt.subplot(6,1,2)
sns.barplot(y="Address", x="NO2", data = NO2_Address.head(10))

plt.subplot(6,1,3)
sns.barplot(y="Address", x="O3", data = O3_Address.head(10))

plt.subplot(6,1,4)
sns.barplot(y="Address", x="CO", data = CO_Address.head(10))

plt.subplot(6,1,5)
sns.barplot(y="Address", x="PM10", data = PM10_Address.head(10))

plt.subplot(6,1,6)
sns.barplot(y="Address", x="PM2.5", data = PM2_5_Address.head(10))
```

```
<Axes: xlabel='PM2.5', ylabel='Address'>
```



Address
369, Yongmasan-ro, Jungnang-gu, Seoul, Republic of Korea
71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Republic of Korea
45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republic of Korea
43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul, Republic of Korea
426, Hakdong-ro, Gangnam-gu, Seoul, Republic of Korea
236, Baekjegobun-ro, Songpa-gu, Seoul, Republic of Korea
59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul, Republic of Korea
571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republic of Korea
56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Republic of Korea
34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Republic of Korea
0.000
0.001
0.002
0.003
0.004
0.005
0.006
SO2
Address
15, Deoksugung-gil, Jung-gu, Seoul, Republic of Korea
19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republic of Korea
136, Hannam-daero, Yongsan-gu, Seoul, Republic of Korea
70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Republic of Korea
56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Republic of Korea
236, Baekjegobun-ro, Songpa-gu, Seoul, Republic of Korea
11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul, Republic of Korea
20, Geumha-ro 21-gil, Geumcheon-gu, Seoul, Republic of Korea
426, Hakdong-ro, Gangnam-gu, Seoul, Republic of Korea
59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul, Republic of Korea
0.000
0.005
0.010
0.015
0.020
0.025
0.030
NO2
Address
49, Samyang-ro 139-gil, Gangbuk-gu, Seoul, Republic of Korea
215, Jinheung-ro, Eunpyeong-gu, Seoul, Republic of Korea
34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Republic of Korea
32, Segeomjeong-ro 4-gil, Seodaemun-gu, Seoul, Republic of Korea
71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Republic of Korea
16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Republic of Korea
6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Republic of Korea
19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republic of Korea
45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republic of Korea
15, Deoksugung-gil, Jung-gu, Seoul, Republic of Korea
0.000
0.005
0.010
0.015
0.020
0.025
O3
Address
70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Republic of Korea
571, Gwangnaru-ro, Gwangjin-gu, Seoul, Republic of Korea
10, Poeun-ro 6-gil, Mapo-gu, Seoul, Republic of Korea
215, Jinheung-ro, Eunpyeong-gu, Seoul, Republic of Korea
59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul, Republic of Korea
56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Republic of Korea
43, Cheonho-daero 13-gil, Dongdaemun-gu, Seoul, Republic of Korea
34, Sirubong-ro 2-gil, Dobong-gu, Seoul, Republic of Korea
11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul, Republic of Korea
236, Baekjegobun-ro, Songpa-gu, Seoul, Republic of Korea
0.0
0.1
0.2
0.3
0.4
0.5
0.6
CO
Address
11, Yangsan-ro 23-gil, Yeongdeungpo-gu, Seoul, Republic of Korea
59, Gucheonmyeon-ro 42-gil, Gangdong-gu, Seoul, Republic of Korea
70, Samyang-ro 2-gil, Seongbuk-gu, Seoul, Republic of Korea
18, Ttukseom-ro 3-gil, Seongdong-gu, Seoul, Republic of Korea
71, Gangseo-ro 45da-gil, Gangseo-gu, Seoul, Republic of Korea
14, Sillimdong-gil, Gwanak-gu, Seoul, Republic of Korea
16, Sinbanpo-ro 15-gil, Seocho-gu, Seoul, Republic of Korea
6, Sadang-ro 16a-gil, Dongjak-gu, Seoul, Republic of Korea
56, Jungang-ro 52-gil, Yangcheon-gu, Seoul, Republic of Korea
45, Gamasan-ro 27-gil, Guro-gu, Seoul, Republic of Korea



3.4. 미세먼지 상관관계

데이터를 새로 가져와줍니다.

```
df_summary = pd.read_csv('/content/drive/MyDrive/한국분석/air_pollution_in_seoul/AirPollutionSeoul/Measurement_summary.csv')
df_summary.head()
```

	Measurement date	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01 00:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.059	0.002	1.2	73.0	57.0
1	2017-01-01 01:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.058	0.002	1.2	71.0	59.0
2	2017-01-01 02:00	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.056	0.002	1.2	70.0	59.0

```
df_date = df_summary['Measurement date'].str.split(" ", n=1, expand=True)
df_date.head()
```

	0	1
0	2017-01-01	00:00
1	2017-01-01	01:00
2	2017-01-01	02:00
3	2017-01-01	03:00
4	2017-01-01	04:00

```
df_summary['date'] = df_date[0]
df_summary['time'] = df_date[1]
df_summary = df_summary.drop(['Measurement date'], axis=1)
df_summary.head()
```

	Station code	Address	Latitude	Longitude	SO2	NO2	O3	CO	PM10	PM2.5	date	time
0	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.059	0.002	1.2	73.0	57.0	2017-01-01	00:00
1	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.058	0.002	1.2	71.0	59.0	2017-01-01	01:00
2	101	19, Jong-ro 35ga-gil, Jongno-gu, Seoul, Republ...	37.572016	127.005008	0.004	0.056	0.002	1.2	70.0	59.0	2017-01-01	02:00

```
df_0 = df_summary.groupby(['date'], as_index=False).agg({'SO2':'mean', 'NO2':'mean', 'O3':'mean', 'CO':'mean', 'PM10':'mean', 'PM2.5':'mean'})
df_0.head()
```

	date	SO2	NO2	O3	CO	PM10	PM2.5
0	2017-01-01	0.003627	0.044765	0.002478	0.981833	77.201667	56.773333
1	2017-01-02	0.002707	0.035960	0.013127	0.891333	109.243333	77.838333
2	2017-01-03	0.000602	0.037017	0.008223	0.753833	78.546667	51.533333
3	2017-01-04	0.004122	0.048813	0.006918	0.878500	54.966667	34.533333
4	2017-01-05	0.003122	0.033892	0.009725	0.656333	36.246667	22.168333

다음 단계: df_0변수로 코드 생성 | 추천 차트 보기 | New interactive sheet

corr()을 이용해서 상관계수 계산하기

```
# Convert 'date' column to datetime objects
df_0['date'] = pd.to_datetime(df_0['date'])

# Extract numerical features from the date
df_0['dayofweek'] = df_0['date'].dt.dayofweek  # Day of the week (0 = Monday, 6 = Sunday)
# ... add other relevant features like month, year, etc. if needed ...

# Now drop the original 'date' column as it's no longer needed for correlation
df_0 = df_0.drop('date', axis=1)

# Calculate the correlation matrix
df_air = df_0.corr()
df_air
```