

# |서울특별시 송파구 대기오염도 분석|

빅데이터사례연구

# 목 차

I  
문제 정의

II  
데이터 수집

III  
EDA 및 전처리

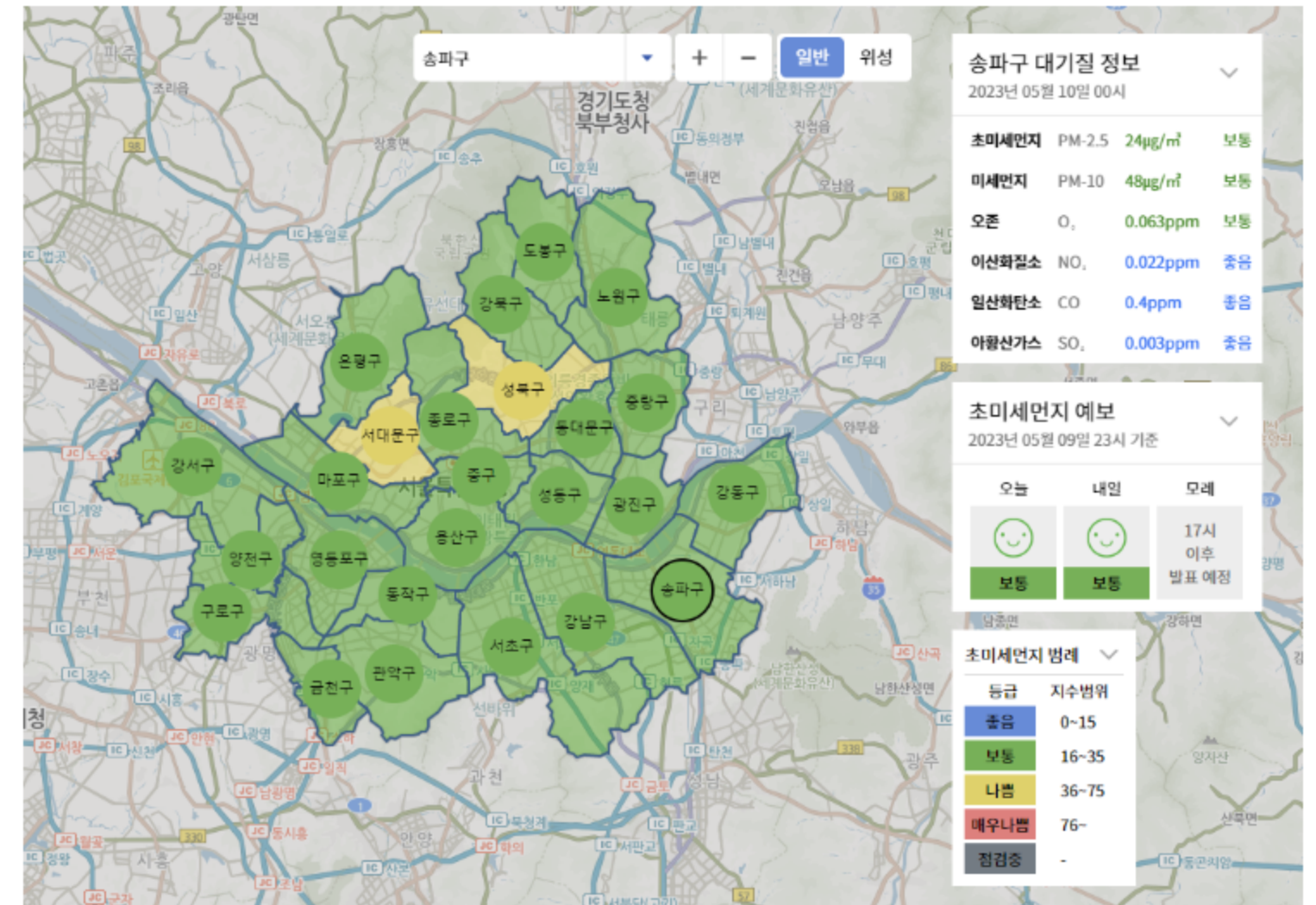
IV  
분석 방법

V  
분석 과정 및 결과

VI  
인사이트

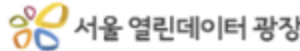
# I 문제 정의

- 서울특별시는 대기오염으로 인한 시민의 건강 피해를 최소화하기 위해 매시간마다 여러 대기오염물질의 농도를 알려줌
- 서울시 일별 평균 대기오염도 데이터 셋에서 측정소명이 송파구인 데이터를 대상으로 상관분석을 통해 대기오염물질 간의 어떤 관계성을 가지고 있고 초미세먼지와 함께 유의해야하는 요인은 무엇인지 알아보고자 함



## II 데이터 수집

- 이 데이터는 서울 열린데이터 광장에서 제공하는 서울시 일별 평균 대기오염도 정보 데이터임
- 연도별로 제공된 데이터 파일들 중 2020년부터 2022년 총 3개의 데이터 파일을 수집함

서울 열린데이터 광장


공공데이터통계서울빅데이터소식&참여이용안내

데이터셋

Home > 공공데이터 > 공공데이터

☐결과 내 재검색

공공데이터



환경






활용사례(갤러리) 등록URL 복사목록 이동

서울시 일별 평균 대기오염도 정보

대기 환경지수, 미세먼지, 오존, 이산화질소, 일산화탄소, 아황산가스 등의 평균 대기오염도 일별 정보를 제공합니다.  
※ Sheet 서비스는 최근 1년 이내의 데이터만 출력합니다.

파일내려받기

\* 파일이 이상이 있는 경우 '오류신고'를 통해 운영자에게 알려주세요.

NO	항목	파일명	용량 (MB)	수정일	내려받기
1	데이터	일별평균대기오염도_2022.csv	0.8	2023.01.12.	
2	데이터	일별평균대기오염도_2021.csv	0.8	2022.10.27.	
3	데이터	일별평균대기오염도_2020.csv	0.8	2022.10.27.	
4	데이터	일별평균대기오염도_2019.csv	0.77	2022.10.27.	
5	데이터	일별평균대기오염도_2018.csv	0.64	2022.10.27.	

[전체 파일보기](#)

서울특별시 송파구 대기오염도 분석

# III EDA 및 전처리

## • 데이터 정보 확인

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
data = pd.read_csv('/content/drive/MyDrive/서울시_3년_일별평균대기오염도.csv')
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 54800 entries, 0 to 54799
Data columns (total 8 columns):
 #   Column                Non-Null Count  Dtype  
---  -
 0   측정일시              54800 non-null  int64  
 1   측정소명              54800 non-null  object  
 2   이산화질소농도(ppm)  54045 non-null  float64 
 3   오존농도(ppm)         54211 non-null  float64 
 4   일산화탄소농도(ppm)  54013 non-null  float64 
 5   아황산가스농도(ppm)  54142 non-null  float64 
 6   미세먼지농도(μg/m³)  53899 non-null  float64 
 7   초미세먼지농도(μg/m³) 53920 non-null  float64 
dtypes: float64(6), int64(1), object(1)
memory usage: 3.3+ MB
```

## • 측정소명이 송파구인 행 추출

```
data_s = data[data['측정소명'] == '송파구']
data_s.head()
```

	측정일시	측정소명	이산화질소농도(ppm)	오존농도(ppm)	일산화탄소농도(ppm)	아황산가스농도(ppm)	미세먼지농도(μg/m³)	초미세먼지농도(μg/m³)
10220	20220101	송파구	0.034	0.013	0.6	0.004	28.0	14.0
10221	20220102	송파구	0.036	0.008	0.5	0.003	32.0	22.0
10222	20220103	송파구	0.043	0.007	0.6	0.004	27.0	15.0
10223	20220104	송파구	0.032	0.013	0.6	0.004	35.0	20.0
10224	20220105	송파구	0.044	0.005	0.8	0.004	46.0	30.0

### III EDA 및 전처리

- 측정소명, 측정일시, 미세먼지농도 컬럼 삭제

```
data_s.drop('측정소명', axis = 1, inplace = True)
data_s.drop('측정일시', axis = 1, inplace = True)
data_s.drop('미세먼지농도(μg/m³)', axis = 1, inplace = True)
```

- 행과 열의 개수 확인

```
data_s.shape

(1096, 5)
```

- 컬럼명 변경

```
data_s.columns = ['N02', 'O3', 'CO', 'SO2', 'PM2.5']
data_s.columns

Index(['N02', 'O3', 'CO', 'SO2', 'PM2.5'], dtype='object')
```

- 요약통계값 확인

```
data_s.describe()
```

	이산화질소	오존	일산화탄소	아황산가스	초미세먼지
	N02	O3	CO	SO2	PM2.5
count	1094.000000	1095.000000	1095.000000	1091.000000	1093.000000
mean	0.026175	0.025916	0.489498	0.002998	19.366880
std	0.012540	0.013351	0.159493	0.000760	11.951592
min	0.003000	0.002000	0.200000	0.001000	1.000000
25%	0.016000	0.016000	0.400000	0.002000	11.000000
50%	0.023000	0.025000	0.500000	0.003000	17.000000
75%	0.035000	0.036000	0.600000	0.003000	25.000000
max	0.068000	0.078000	1.100000	0.005000	90.000000



### III EDA 및 전처리

- 데이터 분포 확인 [박스플롯]

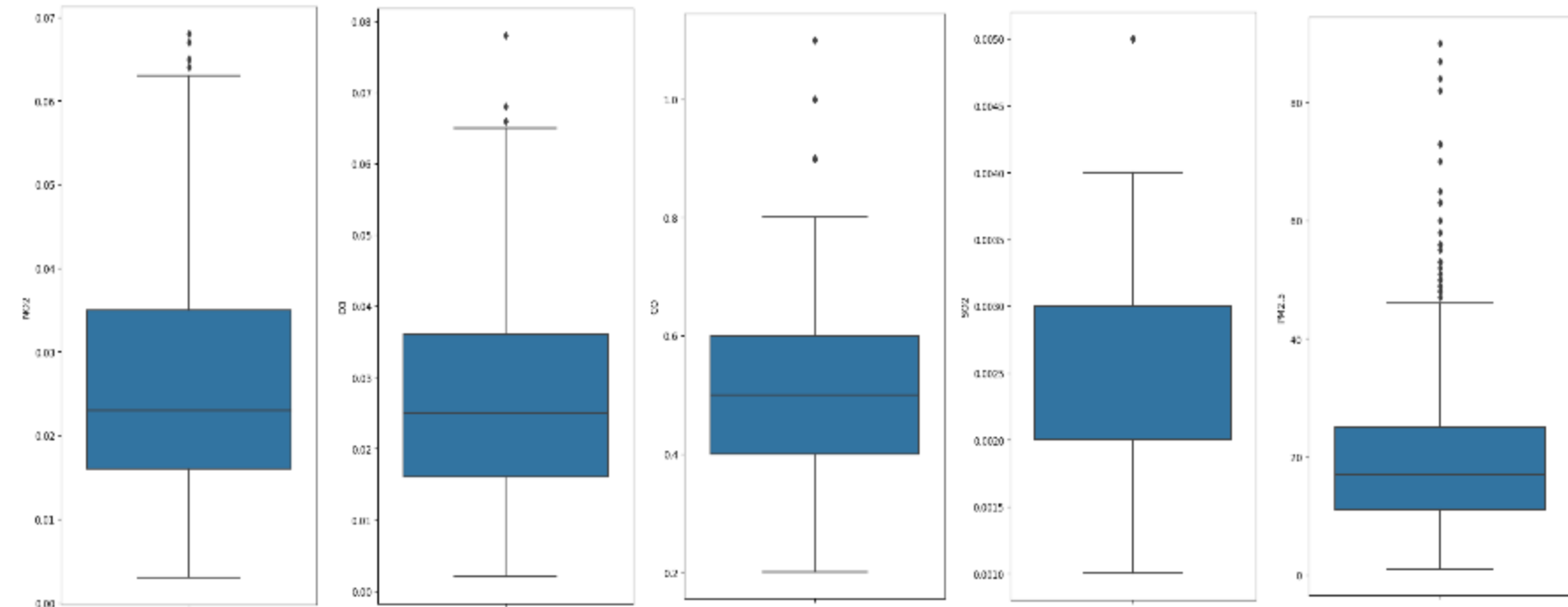
```
plt.figure(figsize = (5,10))  
sns.boxplot(y = data_s['N02'])
```

```
plt.figure(figsize = (5,10))  
sns.boxplot(y = data_s['O3'])
```

```
plt.figure(figsize = (5,10))  
sns.boxplot(y = data_s['CO'])
```

```
plt.figure(figsize = (5,10))  
sns.boxplot(y = data_s['SO2'])
```

```
plt.figure(figsize = (5,10))  
sns.boxplot(y = data_s['PM2.5'])
```



### III EDA 및 전처리

- 결측치 확인

```
data_s.isnull().sum()
```

NO2	2	이산화질소
O3	1	오존
CO	1	일산화탄소
SO2	5	아황산가스
PM2.5	3	초미세먼지

dtype: int64

- 결측치 대체

```
data_s['NO2'] = data_s['NO2'].fillna(data_s['NO2'].median())
data_s['O3'] = data_s['O3'].fillna(data_s['O3'].median())
data_s['CO'] = data_s['CO'].fillna(data_s['CO'].median())
data_s['SO2'] = data_s['SO2'].fillna(data_s['SO2'].median())
data_s['PM2.5'] = data_s['PM2.5'].fillna(data_s['PM2.5'].median())
```

```
data_s.isnull().sum()
```

NO2	0	이산화질소
O3	0	오존
CO	0	일산화탄소
SO2	0	아황산가스
PM2.5	0	초미세먼지

dtype: int64



## IV 분석 방법

### 상관분석

- 상관분석은 두 변수 간의 관계의 정도를 알아보기 위한 분석방법으로 상관계수를 이용해 확인하여 1에 가까울수록 강한 양의 상관관계가 있고, -1에 가까울수록 강한 음의 상관관계가 있음

상관	상관계수
양의 상관	+0.7 ~ +1.0 이면 강한 양의 상관관계 +0.3 ~ +0.7 이면 뚜렷한 양의 상관관계 +0.1 ~ +0.3 이면 약한 양의 상관관계
무상관	-0.1 ~ +0.1 이면 관계가 없음
음의 상관	-1.0 ~ -0.7 이면 강한 음의 상관관계 -0.7 ~ -0.3 이면 뚜렷한 음의 상관관계 -0.3 ~ -0.1 이면 약한 음의 상관관계

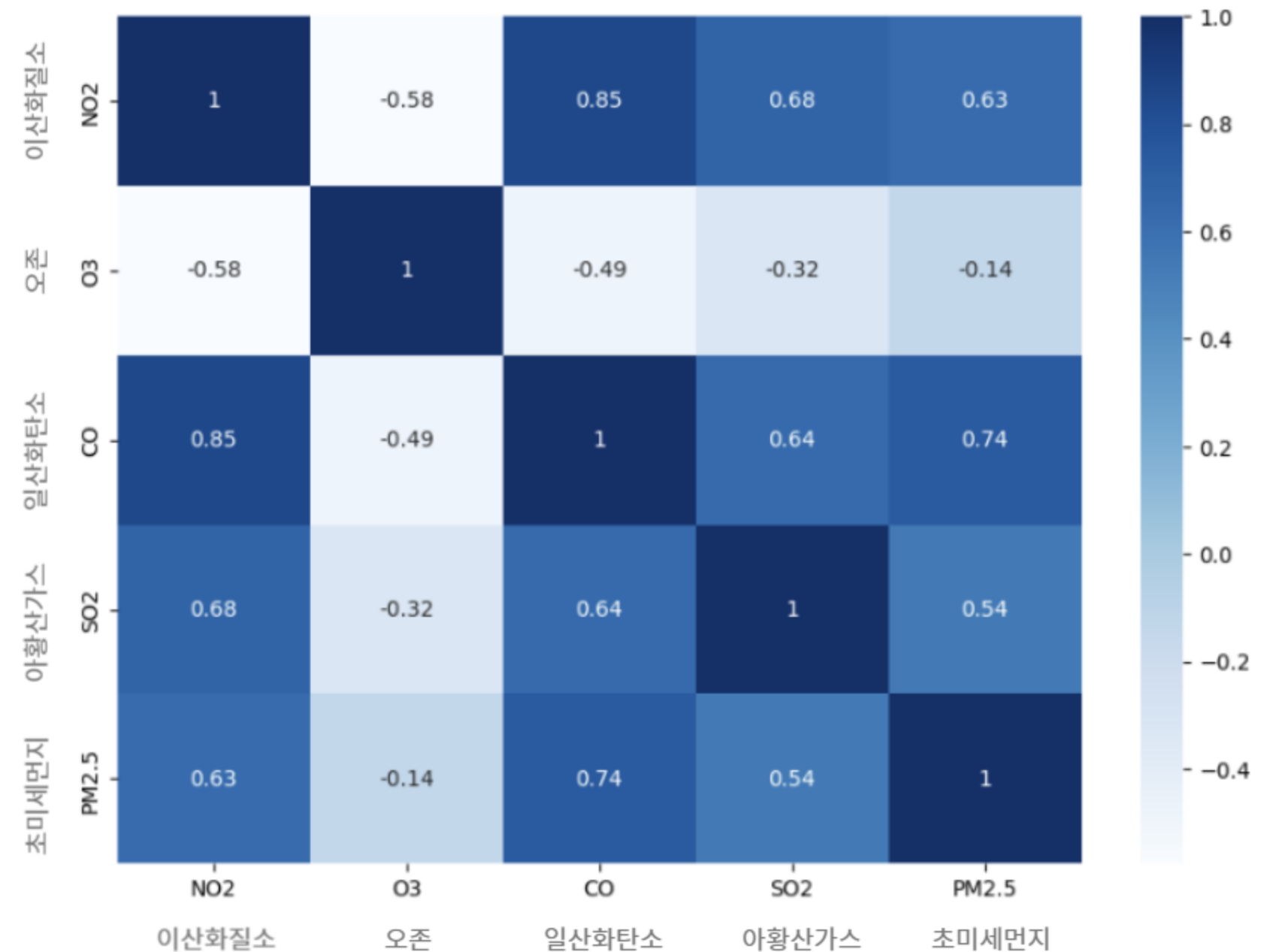
## V 분석 과정 및 결과

### 분석 과정

- 상관 분석 실시

```
corr = data_s.corr()

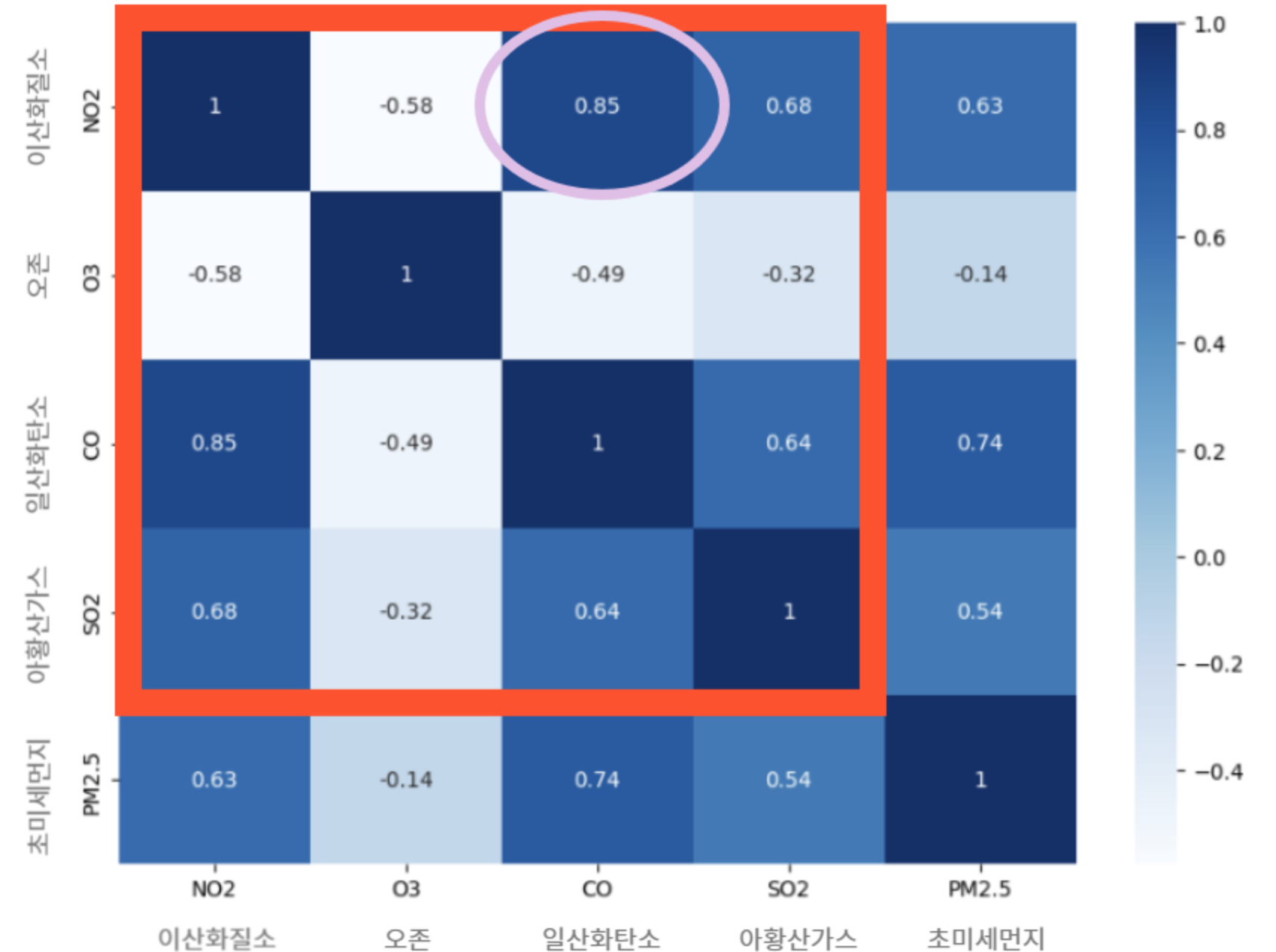
plt.figure(figsize = (10,7))
sns.heatmap(corr, annot = True, cmap = 'Blues')
```



## V 분석 과정 및 결과

### 분석 결과

- 먼저 PM2.5를 제외한 나머지 변수들 간의 상관성을 보면 4개의 변수 모두 서로 어느 정도의 상관성을 가지고 있음
- 특히 NO2와 CO의 상관계수는 0.85로 강한 양의 상관관계를 가지고 있다는 것을 알 수 있음



## V 분석 과정 및 결과

### 분석 결과

- PM2.5와 CO의 상관계수가 0.74로 두 변수는 강한 양의 상관관계를 가지고 있음
- 그 다음으로 NO2와 0.62, SO2와 0.53 순으로 뚜렷한 양의 상관관계를 가지고 있음
- 마지막으로 O3와는 -0.14로 약한 음의 상관관계를 가지고 있음

```
corr['PM2.5'].sort_values(ascending=False)
```

PM2.5	1.000000	초미세먼지
CO	0.740859	일산화탄소
NO2	0.629249	이산화질소
SO2	0.538628	아황산가스
O3	-0.140855	오존

## VI 인사이트

### 인사이트 - 1

NO2와 CO는 모든 변수들 중 가장 강한 양의 상관관계를 가지고 있어 두 대기오염 물질을 함께 유의하여 대책을 세워야 함

### 인사이트 - 2

PM2.5와 CO는 강한 양의 상관관계를 가지고 있으므로 PM2.5가 높은 날에는 CO도 함께 높으므로 함께 유의하여 예방해야 함

### 인사이트 - 3

PM2.5와 NO2, SO2도 CO 만큼은 아니지만 뚜렷한 양의 상관관계를 가지고 있으므로 PM2.5가 높은 날에는 NO2와 SO2도 조심해야 함

# 감사합니다!