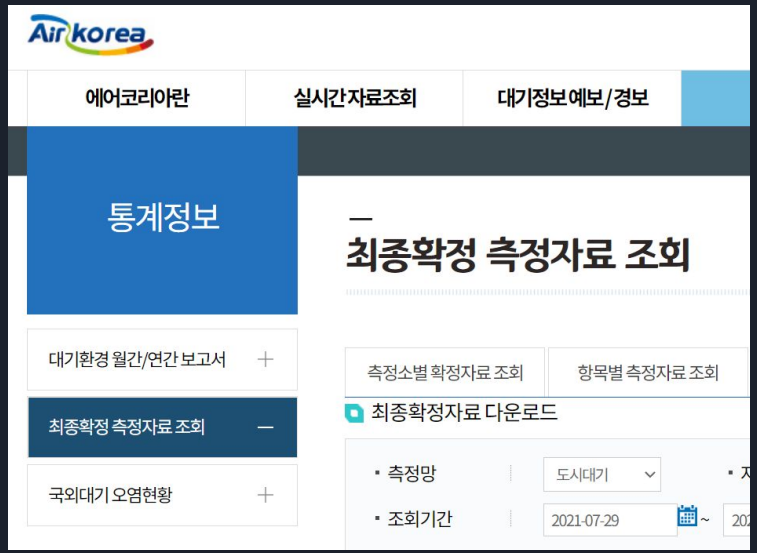
미세먼지 데이터 분석

2팀 강호연 김용현 유혁재

데이터 획득 및 정리

1) 데이터 불러오기
2) 데이터 합치기
3) 확인하기

1) 데이터 불러오기



→

```
In [19]: #2017년 데이터를 불러오자
csv171 = pd.read_csv("2017년 1월.csv")
csv172 = pd.read_csv("2017년 2월.csv")
csv173 = pd.read_csv("2017년 3월.csv")
csv174 = pd.read_csv("2017년 4월.csv")
csv175 = pd.read_csv("2017년 5월.csv")
csv176 = pd.read_csv("2017년 6월.csv")
csv177 = pd.read_csv("2017년 7월.csv")
csv178 = pd.read_csv("2017년 8월.csv")
csv179 = pd.read_csv("2017년 9월.csv")
csv1710 = pd.read_csv("2017년 10월.csv")
csv1711 = pd.read_csv("2017년 11월.csv")
csv1712 = pd.read_csv("2017년 12월.csv")

#2018년 데이터를 불러오자
csv18_01 = pd.read_csv("2018년 1분기.csv")
csv18_02 = pd.read_csv("2018년 2분기.csv")
csv18_03 = pd.read_csv("2018년 3분기.csv")
csv18_04 = pd.read_csv("2018년 4분기.csv")
```

데이터 획득

데이터 불러오기

2) 데이터 합치기

월별 데이터를 연간 데이터로 결합해준다)

```
#17년 데이터 합치기
csv17 = pd.concat([csv171, csv172, csv173, csv174, csv175, csv176, csv177, csv178, csv179, csv1710, csv1711, csv1712], ignore_index
csv17.drop(["Unnamed: 0"], axis=1, inplace =True)
print(csv17.head())

#18년 데이터 합치기
csv18 = pd.concat([csv18_01, csv18_02, csv18_03, csv18_04], ignore_index = True)
csv18.drop(["Unnamed: 0"], axis=1, inplace =True)
print(csv18.head())

#19년 데이터 합치기
csv19 = pd.concat([csv191, csv192, csv193, csv194, csv195, csv196, csv197, csv198, csv199, csv1910, csv1911, csv1912], ignore_index
csv19.drop(["Unnamed: 0"], axis=1, inplace =True)
print(csv19.head())

#20년 데이터 합치기
csv20 = pd.concat([csv201, csv202, csv203, csv204, csv205, csv206, csv207, csv208, csv209, csv2010, csv2011, csv2012], ignore_index
csv20.drop(["Unnamed: 0"], axis=1, inplace =True)
print(csv20.head())
```

예시로 결합한 17년도 상위 5개의 데이터 확인

```
     지역     망   측정소코드 측정소명          측정일시   SO2   CO     O3     NO2   PM10  PM25  ₩
0  서울 중구  도시대기  111121   중구  2017010101  0.006  1.3  0.002  0.068  77.0  63.0
1  서울 중구  도시대기  111121   중구  2017010102  0.006  1.4  0.002  0.066  76.0  63.0
2  서울 중구  도시대기  111121   중구  2017010103  0.005  1.2  0.002  0.063  73.0  57.0
3  서울 중구  도시대기  111121   중구  2017010104  0.005  1.1  0.002  0.053  67.0  55.0
4  서울 중구  도시대기  111121   중구  2017010105  0.004  1.1  0.002  0.051  66.0  54.0

            주소
0  서울 중구 덕수궁길 15
1  서울 중구 덕수궁길 15
2  서울 중구 덕수궁길 15
3  서울 중구 덕수궁길 15
4  서울 중구 덕수궁길 15
```

3) 확인하기

원하는 지역을 키워드를 이용하여 불러온다)

```
csv17 = csv17[csv17['지역'].str.contains('서울')]
csv18 = csv18[csv18['지역'].str.contains('서울')]
csv19 = csv19[csv19['지역'].str.contains('서울')]
csv20 = csv20[csv20['지역'].str.contains('서울')]
```

```
csv17.reset_index(drop = True, inplace=True)
csv18.reset_index(drop = True, inplace=True)
csv19.reset_index(drop = True, inplace=True)
csv20.reset_index(drop = True, inplace=True)
```

```
csv17_PM10 = csv17[["지역", "측정일시", "PM10"]]
csv18_PM10 = csv18[["지역", "측정일시", "PM10"]]
csv19_PM10 = csv19[["지역", "측정일시", "PM10"]]
csv20_PM10 = csv20[["지역", "측정일시", "PM10"]]

csv17_PM25 = csv17[["지역", "측정일시", "PM25"]]
csv18_PM25 = csv18[["지역", "측정일시", "PM25"]]
csv19_PM25 = csv19[["지역", "측정일시", "PM25"]]
csv20_PM25 = csv20[["지역", "측정일시", "PM25"]]

csv17_PM10.head()
```

	지역	측정일시	PM10
0	서울 중구	2017010101	77.0
1	서울 중구	2017010102	76.0
2	서울 중구	2017010103	73.0
3	서울 중구	2017010104	67.0
4	서울 중구	2017010105	66.0

3) 확인하기

서울의 구가 모두 있는지 확인한다)

```
csv_17_seoul_PM10 = csv17_PM10["지역"].unique()
csv_18_seoul_PM10 = csv18_PM10["지역"].unique()
csv_19_seoul_PM10 = csv19_PM10["지역"].unique()
csv_20_seoul_PM10 = csv20_PM10["지역"].unique()
csv_21_seoul_PM10 = csv21_PM10["지역"].unique()

csv_17_seoul_PM25 = csv17_PM25["지역"].unique()
csv_18_seoul_PM25 = csv18_PM25["지역"].unique()
csv_19_seoul_PM25 = csv19_PM25["지역"].unique()
csv_20_seoul_PM25 = csv20_PM25["지역"].unique()
csv_21_seoul_PM25 = csv21_PM25["지역"].unique()

print(len(csv_17_seoul_PM10))
print(len(csv_18_seoul_PM10))
print(len(csv_19_seoul_PM10))
print(len(csv_20_seoul_PM10))
print(len(csv_21_seoul_PM10))

print(len(csv_17_seoul_PM25))
print(len(csv_18_seoul_PM25))
print(len(csv_19_seoul_PM25))
print(len(csv_20_seoul_PM25))
print(len(csv_21_seoul_PM25))
```

```
25
48
25
25
25
25
48
25
25
25
```

← 18년도 구 개수가 48

3) 확인하기

각 연도별 데이터 결측값을 확인한다)

```
print("미세먼지데이터")
print("2017년 서울데이터")
print(csv17_PM10.isnull().sum())
print("")
print("2018년 서울산데이터")
print(csv18_PM10.isnull().sum())
print("")
print("2019년 서울데이터")
print(csv19_PM10.isnull().sum())
print("")
print("2020년 서울데이터")
print(csv20_PM10.isnull().sum())
print("")
print("")
```

```
print("초미세먼지데이터")
print("2017년 서울데이터")
print(csv17_PM25.isnull().sum())
print("")
print("2018년 서울데이터")
print(csv18_PM25.isnull().sum())
print("")
print("2019년 서울데이터")
print(csv19_PM25.isnull().sum())
print("")
print("2020년 서울데이터")
print(csv20_PM25.isnull().sum())
```

```
csv17_PM10.dropna(axis=0, inplace = True)
csv18_PM10.dropna(axis=0, inplace = True)
csv19_PM10.dropna(axis=0, inplace = True)
csv20_PM10.dropna(axis=0, inplace = True)

csv17_PM25.dropna(axis=0, inplace = True)
csv18_PM25.dropna(axis=0, inplace = True)
csv19_PM25.dropna(axis=0, inplace = True)
csv20_PM25.dropna(axis=0, inplace = True)
```

ex)

```
지역          0
측정일시        0
PM10    24252
dtype: int64
```

→

```
지역      0
측정일시    0
PM10    0
dtype: int64
```

3) 확인하기

연도별 데이터내에 측정일시를 월로 바꾼다)

```
csv17_PM10["월별"] = csv17_PM10["측정일시"].astype(str)
csv17_PM10["월별"] = csv17_PM10["월별"].str[4:6]
csv17_PM10 = csv17_PM10[["지역","월별","PM10"]]

csv18_PM10["월별"] = csv18_PM10["측정일시"].astype(str)
csv18_PM10["월별"] = csv18_PM10["월별"].str[4:6]
csv18_PM10 = csv18_PM10[["지역","월별","PM10"]]

csv19_PM10["월별"] = csv19_PM10["측정일시"].astype(str)
csv19_PM10["월별"] = csv19_PM10["월별"].str[4:6]
csv19_PM10 = csv19_PM10[["지역","월별","PM10"]]

csv20_PM10["월별"] = csv20_PM10["측정일시"].astype(str)
csv20_PM10["월별"] = csv20_PM10["월별"].str[4:6]
csv20_PM10 = csv20_PM10[["지역","월별","PM10"]]
```

미세먼지 데이터

```
csv17_PM25["월별"] = csv17_PM25["측정일시"].astype(str)
csv17_PM25["월별"] = csv17_PM25["월별"].str[4:6]
csv17_PM25 = csv17_PM25[["지역","월별","PM25"]]

csv18_PM25["월별"] = csv18_PM25["측정일시"].astype(str)
csv18_PM25["월별"] = csv18_PM25["월별"].str[4:6]
csv18_PM25 = csv18_PM25[["지역","월별","PM25"]]

csv19_PM25["월별"] = csv19_PM25["측정일시"].astype(str)
csv19_PM25["월별"] = csv19_PM25["월별"].str[4:6]
csv19_PM25 = csv19_PM25[["지역","월별","PM25"]]

csv20_PM25["월별"] = csv20_PM25["측정일시"].astype(str)
csv20_PM25["월별"] = csv20_PM25["월별"].str[4:6]
csv20_PM25 = csv20_PM25[["지역","월별","PM25"]]
```

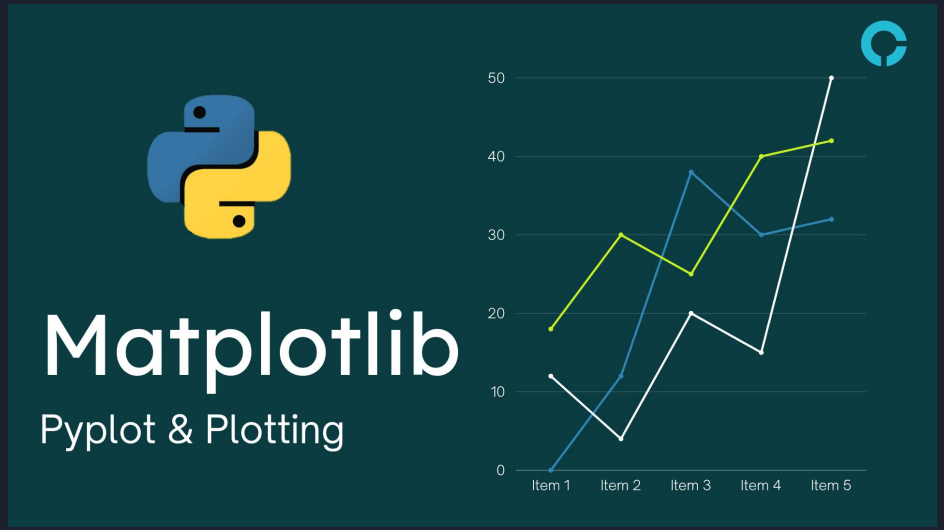
초미세먼지 데이터

시각화

1) pyplot이란
2) 미세먼지가 가장 많은 달과 적은 달은?
3) 서울에서 가장 미세먼지가 많은 곳은?

1) pyplot이란

- Matplotlib에서 지원하는 모듈 중 하나이다.
- pyplot은 사용환경 인터페이스를 제공한다.
- pyplot의 인터페이스는 겉으로는 드러나지 않으면서 자동으로 figure와 axes를 생성하며, 정의된 플롯을 얻을 수 있도록 만들어 준다



2) 미세먼지가 가장 많은 달과 적은 달은?

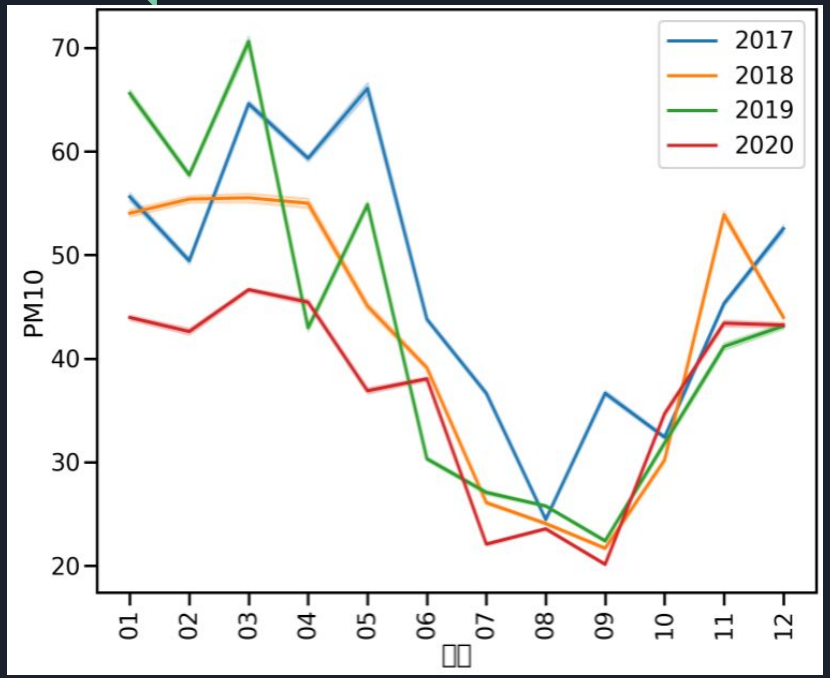
1) 한글을 사용하기 위해 폰트 불러오기
2) pyplot 모듈을 활용하여 년도별로 월별 미세먼지 데이터를 그래프로 나타내기

```
[31] import matplotlib.pyplot as plt
     plt.rc('font',family='Malgun Gothic')
     from IPython.display import set_matplotlib_formats
     set_matplotlib_formats('retina')
```

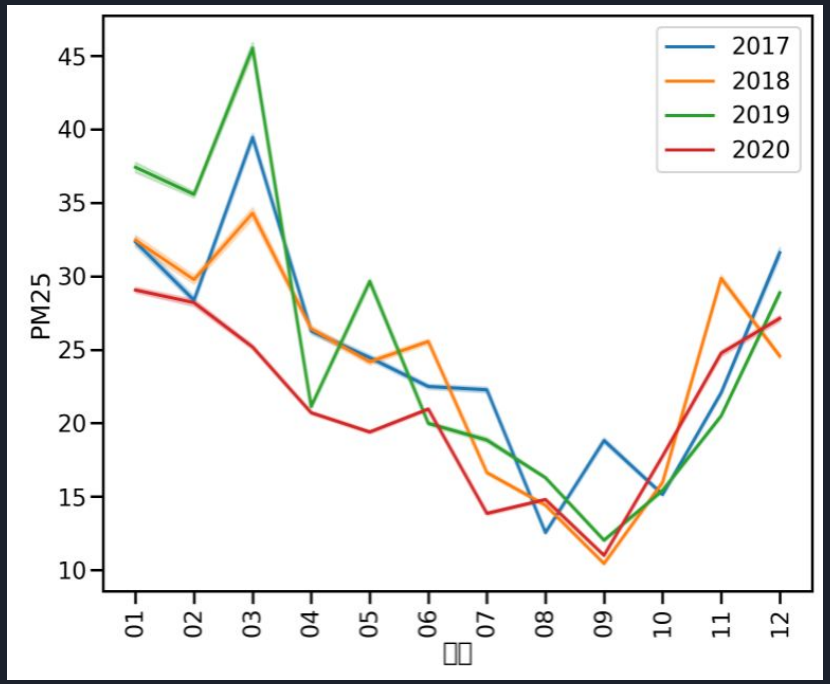
```
plt.figure(figsize = (12,10))

sns.lineplot(x='월별',y='PM10',data = csv17_PM10, label = "2017")
sns.lineplot(x='월별',y='PM10',data = csv18_PM10, label = "2018")
sns.lineplot(x='월별',y='PM10',data = csv19_PM10, label = "2019")
sns.lineplot(x='월별',y='PM10',data = csv20_PM10, label = "2020")

plt.legend()
sns.set_context('poster', font_scale = 1) #seaborn의 크기를 조절하는데 paper, talk, poster순으로 크다.

plt.xticks(rotation=90)
```

2) 결과 및 결론

3-4월에 가장 (초)미세먼지가 많고, 8-9월에 가장 적은 것을 눈으로 확인할 수 있었다.



미세먼지



초미세먼지

3) 서울에서 가장 미세먼지가 많은 곳은?

```
[ ]  csv17_PM25
```

	지역	월별	PM25
0	서울 중구	01	63.0
1	서울 중구	01	63.0
2	서울 중구	01	57.0
3	서울 중구	01	55.0
4	서울 중구	01	54.0
...	...	...	...
340891	서울 노원구	12	12.0
340892	서울 노원구	12	15.0
340893	서울 노원구	12	15.0
340894	서울 노원구	12	15.0
340895	서울 노원구	12	14.0

212815 rows × 3 columns

<- 17년도 서울 모든

지역의 초미세먼지 데이터

구 별로 정리하자!

```
csv17_mapo_PM10 = csv17_PM10[csv17['지역'].str.contains('마포')]
csv17_mapo_PM10
```

	지역	월별	PM10
12648	서울 마포구	01	71.0
12649	서울 마포구	01	70.0
12650	서울 마포구	01	72.0
12651	서울 마포구	01	66.0
12652	서울 마포구	01	66.0
...	...	...	...
326755	서울 마포구	12	65.0
326756	서울 마포구	12	58.0
326757	서울 마포구	12	80.0
326758	서울 마포구	12	63.0
326759	서울 마포구	12	46.0

17171 rows × 3 columns

3) 시각화 처리

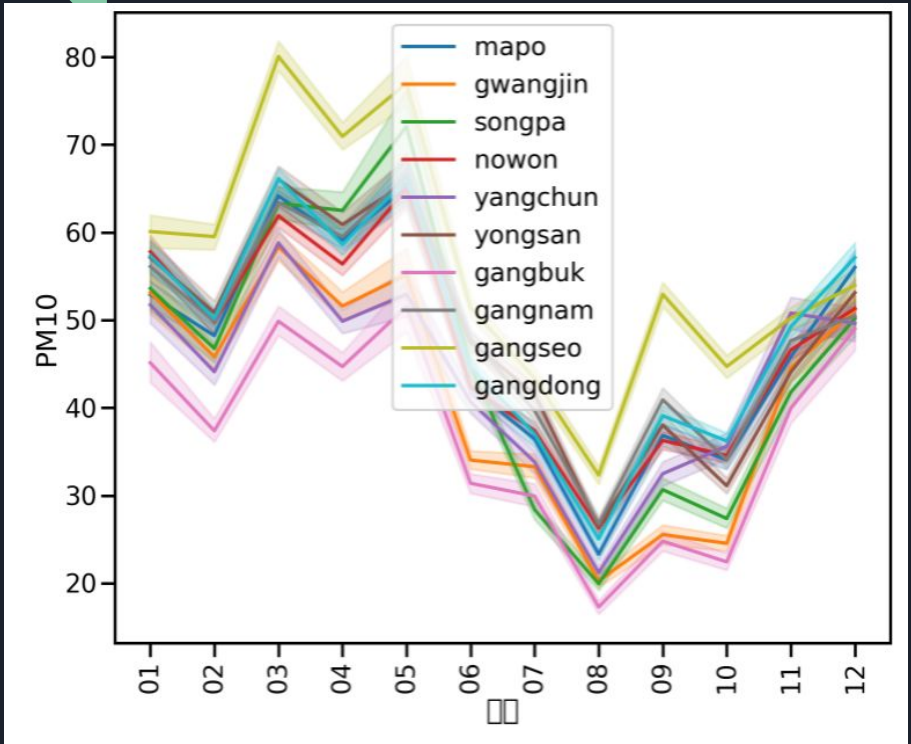
```
csv17_gwangjin_PM10 = csv17_PM10[csv17['지역'].str.contains('광진')]
csv17_songpa_PM10 = csv17_PM10[csv17['지역'].str.contains('송파')]
csv17_nowon_PM10 = csv17_PM10[csv17['지역'].str.contains('노원')]
csv17_yangchun_PM10 = csv17_PM10[csv17['지역'].str.contains('양천')]
csv17_yongsan_PM10 = csv17_PM10[csv17['지역'].str.contains('용산')]
csv17_gangbuk_PM10 = csv17_PM10[csv17['지역'].str.contains('강북')]
csv17_gangnam_PM10 = csv17_PM10[csv17['지역'].str.contains('강남')]
csv17_gangseo_PM10 = csv17_PM10[csv17['지역'].str.contains('강서')]
csv17_gangdong_PM10 = csv17_PM10[csv17['지역'].str.contains('강동')]
```

전부 다 하기엔 그래프가 잘 보이지 않을 것 같아, 강북, 강동, 강서, 강남 기준으로 2, 3개만 선정

앞서 그렸던 방식과 그대로 그리기

```
plt.figure(figsize = (12,10))

sns.lineplot(x='월별',y='PM10',data = csv17_mapo_PM10, label = "mapo")
sns.lineplot(x='월별',y='PM10',data = csv17_gwangjin_PM10, label = "gwangjin")
sns.lineplot(x='월별',y='PM10',data = csv17_songpa_PM10, label = "songpa")
sns.lineplot(x='월별',y='PM10',data = csv17_nowon_PM10, label = "nowon")
sns.lineplot(x='월별',y='PM10',data = csv17_yangchun_PM10, label = "yangchun")
sns.lineplot(x='월별',y='PM10',data = csv17_yongsan_PM10, label = "yongsan")
sns.lineplot(x='월별',y='PM10',data = csv17_gangbuk_PM10, label = "gangbuk")
sns.lineplot(x='월별',y='PM10',data = csv17_gangnam_PM10, label = "gangnam")
sns.lineplot(x='월별',y='PM10',data = csv17_gangseo_PM10, label = "gangseo")
sns.lineplot(x='월별',y='PM10',data = csv17_gangdong_PM10, label = "gangdong")


plt.legend()
sns.set_context('poster', font_scale = 1)

plt.xticks(rotation=90)
```

3) 결과 및 결론



가장 미세먼지가 심한 곳은 강서구

가장 미세먼지가 덜한 곳은 강북구

마무리- 느낀점

김용현

엄청난 양의 데이터들을 코드 몇줄로 처리하고, 시각화까지 할 수 있는 것을 내 손으로 느끼면서 정말 유용하다고 느꼈다. 좋은 경험이었고, 많은 것들을 배울 수 있는 시간이었다.

유혁재

자유 주제로 자율성이 더 높아지면서 분석을 할 때 혼자 고민하고 생각을 많이 하게 되었다. 그 덕분에 데이터 분석에 대한 이해도가 더 깊어졌고 스스로 문제를 처리할 수 있는 능력이 향상되었다.

감사합니다.