Mini project 1: air quality in U.S. cities

In a way, this project is simple: you are given some data on air quality in U.S. metropolitan areas over time together with several questions of interest, and your objective is to answer the questions.

However, unlike the homeworks and labs, there is no explicit instruction provided about *how* to answer the questions or where exactly to begin. Thus, you will need to discern for yourself how to manipulate and summarize the data in order to answer the questions of interest, and you will need to write your own codes from scratch to obtain results. It is recommended that you examine the data, consider the questions, and plan a rough approach before you begin doing any computations.

You have some latitude for creativity: **although there are accurate answers to each question** -- namely, those that are consistent with the data -- **there is no singularly correct answer**. Most students will perform similar operations and obtain similar answers, but there's no specific result that must be considered to answer the questions accurately. As a result, your approaches and answers may differ from those of your classmates. If you choose to discuss your work with others, you may even find that disagreements prove to be fertile learning opportunities.

The questions can be answered using computing skills taught in class so far and basic internet searches for domain background; for this project, you may wish to refer to HW1 and Lab1 for code examples and the EPA website on PM pollution for background. However, you are also encouraged to refer to external resources (package documentation, vignettes, stackexchange, internet searches, etc.) as needed -- this may be an especially good idea if you find yourself thinking, 'it would be really handy to do X, but I haven't seen that in class anywhere'.

The broader goal of these mini projects is to cultivate your problem-solving ability in an unstructured setting. Your work will be evaluated based on the following:

- choice of method(s) used to answer questions;
- clarity of presentation;
- code style and documentation.

Please write up your results separately from your codes; codes should be included at the end of the notebook.

Part I

Merge the city information with the air quality data and tidy the dataset (see notes below). Write a one- to two-paragraph description of the data.

In your description, answer the following questions:

- What is a CBSA (the geographic unit of measurement)?
- How many CBSA's are included in the data?
- In how many states and territories do the CBSA's reside? (*Hint:* `str.split()`)
- In which years were data values recorded?
- How many observations are recorded?
- How many variables are measured?
- Which variables are non-missing most of the time (*i.e.*, in at least 50% of instances)?
- What is PM 2.5 and why is it important?
- What are the basic statistical properties of the variable(s) of interest?

Please write your description in narrative fashion; ***please do not list answers to the questions above one by one***.

Air quality data

ANSWER

This datset contains the information of various aspects of the CBSA, pollutant, number of trend sites, and yearly values from 2000-2019. The CBSA refers to a statistical area of one or more counties with a population of 10,000 or more indiciduals allong with other communites. There are 251 unique CBSAs represented in the data accross 86 states and territories. The dataset is a totoal of 1134 observations spannning accross 27 variables, with most having values. The PM of 2.5 is an air pollutant that can have negative effects on health and visibility which is the hazy air effect.

Part II

Focus on the PM2.5 measurements that are non-missing most of the time. Answer each of the following questions in a brief paragraph or two. Do not describe your analyses step-by-step for your answers; instead, report your findings. Your paragraph(s) should indicate both your answer to the question and a justification for your answer; ***please do not include codes with your answers***.

Has PM 2.5 air pollution improved in the U.S. on the whole since 2000?

ANSWER

Overall, there has been a consistent linear decline in PM 2.5 air pollution in the U.S. since 2000, as evidenced by a scatterplot of Year versus Mean of Weighted Means. To generate this plot, I calculated the mean of each year's weighted means and saved the values in a list, using these means as the y-values and the corresponding years as the x-axis.

Over time, has PM 2.5 pollution become more variable, less variable, or about equally variable from city to city in the U.S.?

***ANSWER**

In the U.S., there has been a reduction in the variability of PM 2.5 pollution levels between cities over time, as demonstrated by a decreasing trend in the scatterplot of Year versus Standard Deviation of Weighted Means. To create this plot, I computed the standard deviation of each year's weighted means and saved the values in a list, using these standard deviations as the y-values and the corresponding years as the x-axis.

Which state has seen the greatest improvement in PM 2.5 pollution over time? Which city has seen the greatest improvement?

ANSWER

Based on the code, the state with the greatest improvement in PM 2.5 is California. I computed the total difference between the "2000" and "2019" columns for each state by grouping the y DataFrame by the "State" column and summing the "Differences" column. Then I found the state with the maximum total difference and printed its name.

In terms of cities, Portsmouth, OH has exhibited the most substantial improvement in PM 2.5 pollution levels. To determine this, I identified the 'Core Based Statistical Area' value corresponding to the highest value in the 'Differences' column.

Choose a location with some meaning to you (e.g. hometown, family lives there, took a vacation there, etc.). Was that location in compliance with EPA primary standards as of the most recent measurement?

ANSWER

A location that was meaningful to me was Albuquerque NM. This was the last place I visited before the pandemic hit and was the last trip I had taken for a long time. Their measurments are in compliance with the EPA standards since 2019 is less than 12. The way I did this was by filtering the dataframe for Albuquerque in the city collumn.

Imputation

One strategy for filling in missing values ('imputation') is to use non-missing values to predict the missing ones; the success of this strategy depends in part on the strength of relationship between the variable(s) used as predictors of missing values.

Identify one other pollutant that might be a good candidate for imputation based on the PM 2.5 measurements and explain why you selected the variable you did. Can you envision any potential pitfalls to this technique?

ANSWER

PM 10 is another pollutant that could be used to impute the missing values for the PM 2.5 measurement. Since both of them are particulate matter emmited by similar sources and environmental factors. One pitfall can be that these two factors may be unique and not share any relationship.

Codes

In [1]:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
air_quality = pd.read_csv("data/air-quality.csv")
cbsa_info = pd.read_csv("data/cbsa-info.csv")

# Part 1

# Merge the datasets
data = pd.merge(air_quality, cbsa_info, how="left", on="CBSA")

# Split the 'Core Based Statistical Area'
data[["City", "State"]] = data["Core Based Statistical Area"].str.split(",",

# Print of unique CBSAs and states/territories
print("The number of Unique CBSA: :", data["CBSA"].nunique())
print("The number of unique states/territories:", data["State"].nunique())

# Print shape
print("The shape of data:", data.shape)

# Calculate missing values
missing_values = data.isnull().sum()
print("number of missing vlaues: ", missing_values)


# Display the first 50 rows
data.head(50)
```

```
The number of Unique CBSA: : 351
The number of unique states/territories: 86
The shape of data: (1134, 27)
number of missing vlaues:  CBSA                          0
Pollutant                       0
Trend Statistic                 0
Number of Trends Sites          0
2000                            0
2001                            0
2002                            0
2003                            0
2004                            0
2005                            0
2006                            0
2007                            0
2008                            0
2009                            0
2010                            0
2011                            0
2012                            0
2013                            0
2014                            0
2015                            0
2016                            0
2017                            0
2018                            0
2019                            0
Core Based Statistical Area     0
City                            0
State                           0
dtype: int64
```

Out[1]:

	CBSA	Pollutant	Trend Statistic	Number of Trends Sites	2000	2001	2002	2003	2004	20
0	10100	PM10	2nd Max	1	50.000	58.000	59.000	66.000	39.000	48.0
1	10100	PM2.5	Weighted Annual Mean	1	8.600	8.600	7.900	8.400	8.100	9.0
2	10100	PM2.5	98th Percentile	1	23.000	23.000	20.000	21.000	23.000	23.0
3	10300	O3	4th Max	1	0.082	0.086	0.089	0.088	0.074	0.0
4	10420	CO	2nd Max	1	2.400	2.700	1.800	1.900	2.100	1.6
5	10420	O3	4th Max	2	0.085	0.096	0.100	0.090	0.079	0.0
6	10420	PM2.5	Weighted Annual Mean	3	16.200	16.200	16.000	14.100	13.800	15.7
7	10420	PM2.5	98th Percentile	3	37.000	44.000	40.000	34.000	35.000	43.0

8	10420	SO2	99th Percentile	1	163.000	132.000	145.000	167.000	172.000	140.0
9	10500	PM2.5	Weighted Annual Mean	1	16.600	14.600	13.800	13.400	14.100	14.6
10	10500	PM2.5	98th Percentile	1	38.000	36.000	31.000	27.000	36.000	35.0
11	10580	CO	2nd Max	1	1.100	1.200	1.500	1.500	1.100	1.4
12	10580	O3	4th Max	2	0.070	0.089	0.091	0.081	0.075	0.0
13	10580	PM2.5	Weighted Annual Mean	1	12.400	12.300	12.100	11.900	11.000	12.6
14	10580	PM2.5	98th Percentile	1	30.000	31.000	33.000	34.000	32.000	36.0
15	10580	SO2	99th Percentile	1	56.000	67.000	34.000	42.000	39.000	25.0
16	10740	CO	2nd Max	1	3.300	2.900	2.800	2.100	2.200	2.0
17	10740	NO2	Annual Mean	1	17.000	17.000	19.000	18.000	17.000	16.0
18	10740	NO2	98th Percentile	1	54.000	54.000	57.000	55.000	52.000	51.0
19	10740	O3	4th Max	3	0.072	0.071	0.074	0.076	0.071	0.0
20	10740	PM10	2nd Max	3	99.300	91.000	110.700	152.700	111.000	124.7
21	10740	PM2.5	Weighted Annual Mean	1	6.600	6.400	6.300	6.900	6.800	7.0
22	10740	PM2.5	98th Percentile	1	20.000	19.000	18.000	17.000	19.000	19.0
23	10900	NO2	Annual Mean	1	17.000	16.000	13.000	13.000	14.000	15.0
24	10900	O3	4th Max	3	0.089	0.093	0.094	0.086	0.085	0.0

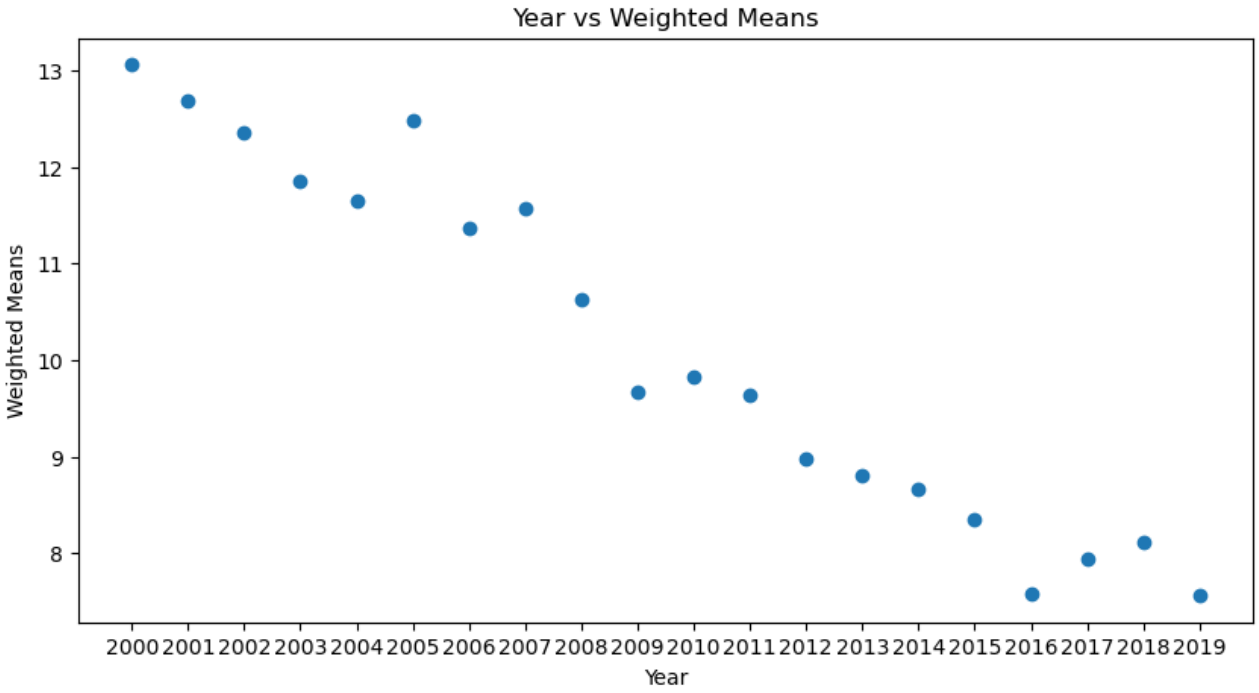
25	10900	PM10	2nd Max	1	78.000	78.000	90.000	49.000	45.000	54.0
26	10900	PM2.5	Weighted Annual Mean	1	13.700	15.300	14.700	14.300	13.700	14.2
27	10900	PM2.5	98th Percentile	1	37.000	43.000	46.000	37.000	35.000	36.0
28	10900	SO2	99th Percentile	1	67.000	67.000	64.000	63.000	92.000	90.0
29	11020	O3	4th Max	1	0.080	0.083	0.089	0.083	0.073	0.0
30	11020	PM10	2nd Max	1	50.000	76.000	67.000	95.000	63.000	74.0
31	11020	SO2	99th Percentile	1	62.000	68.000	51.000	62.000	61.000	72.0
32	11140	O3	4th Max	1	0.092	0.082	0.070	0.072	0.070	0.0
33	11260	CO	2nd Max	1	5.400	5.700	4.700	5.700	6.400	4.8
34	11260	PM2.5	Weighted Annual Mean	2	5.800	6.200	5.800	6.600	6.900	6.7
35	11260	PM2.5	98th Percentile	2	20.000	25.000	25.000	23.000	30.000	22.0
36	11460	O3	4th Max	1	0.078	0.092	0.091	0.091	0.071	0.0
37	11460	PM2.5	Weighted Annual Mean	1	14.300	14.400	14.900	14.600	12.700	15.6
38	11460	PM2.5	98th Percentile	1	30.000	40.000	31.000	39.000	32.000	52.0
39	11540	O3	4th Max	1	0.066	0.085	0.075	0.075	0.071	0.0
40	11540	PM2.5	Weighted Annual Mean	1	11.500	11.100	10.800	10.400	9.700	11.5
41	11540	PM2.5	98th Percentile	1	32.000	33.000	29.000	26.000	29.000	37.0
42	11700	O3	4th Max	3	0.086	0.080	0.090	0.077	0.072	0.0
43	11700	PM2.5	Weighted Annual Mean	1	15.400	13.500	13.800	12.600	12.300	13.1
			98th							

44	11700	PM2.5	Percentile	1	34.000	31.000	31.000	31.000	27.000	32.0
45	11780	O3	4th Max	1	0.082	0.097	0.103	0.099	0.081	0.0
46	11780	SO2	99th Percentile	1	50.000	56.000	55.000	44.000	36.000	39.0
47	11900	PM2.5	Weighted Annual Mean	1	12.400	12.400	12.700	12.300	11.400	13.3
48	11900	PM2.5	98th Percentile	1	32.000	32.000	33.000	29.000	33.000	33.0
49	12020	O3	4th Max	1	0.084	0.084	0.084	0.072	0.078	0.0

50 rows × 27 columns

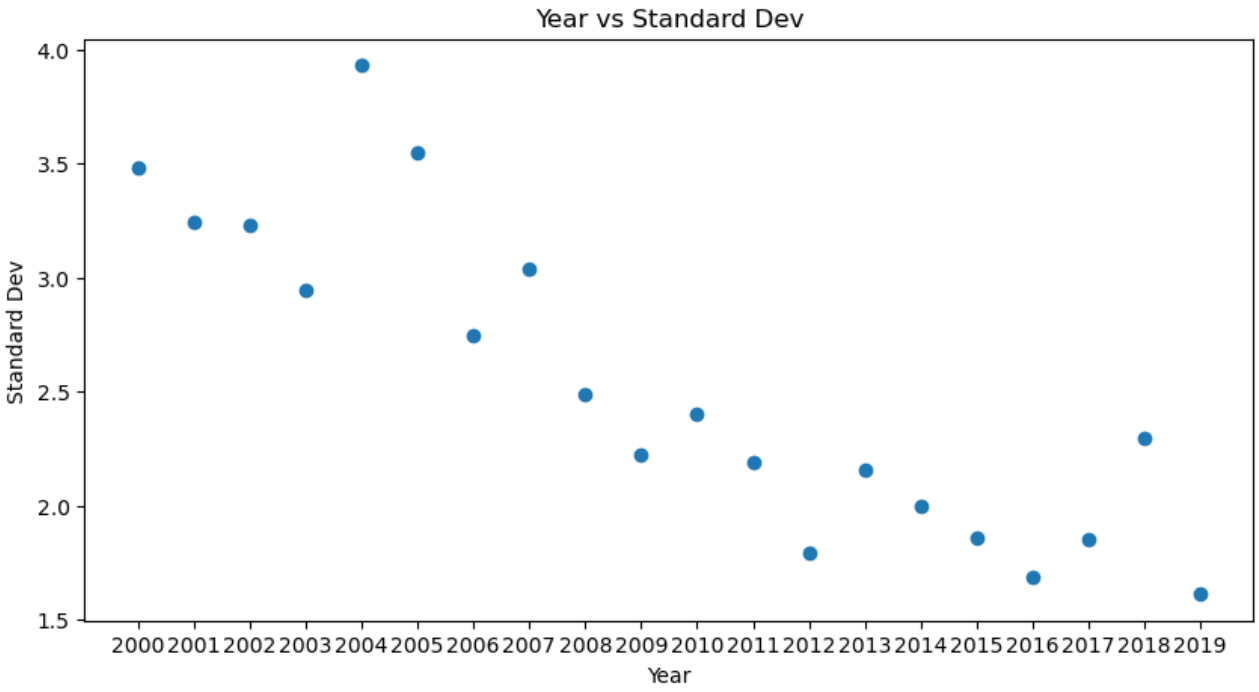
In [2]:
```
# Filter the data
filtered_data = data[(data["Pollutant"] == "PM2.5") & (data["Trend Statistic

# Drop columns
filtered_data = filtered_data.drop(columns=["Pollutant", "Trend Statistic",

# sort the data
filtered_data = filtered_data.sort_values(by=["CBSA"])

# Calculate the mean and standard deviation
means = filtered_data.loc[:, "2000":"2019"].mean()
stddevs = filtered_data.loc[:, "2000":"2019"].std()

# Question 1
fig, ax = plt.subplots(figsize=(10, 5))
ax.scatter(means.index, means.values)
ax.set_title("Year vs Weighted Means")
ax.set_xlabel("Year")
ax.set_ylabel("Weighted Means")
```

Out[2]:
```
Text(0, 0.5, 'Weighted Means')
```



In [3]:
```
# Question 2
fig, ax = plt.subplots(figsize=(10, 5))
ax.scatter(stddevs.index, stddevs.values)
ax.set_title("Year vs Standard Dev")
ax.set_xlabel("Year")
ax.set_ylabel("Standard Dev")
```

Out[3]: Text(0, 0.5, 'Standard Dev')



In [4]:
```
# Question 3
# Copy the DataFrame
difference_data = filtered_data.copy()
difference_data["State"] = filtered_data["State"].apply(lambda s: str(s))

# Filter to 3-letter state codes
difference_data = difference_data.loc[difference_data["State"].apply(lambda

# Compute the differences
difference_data["Differences"] = difference_data["2000"] - difference_data["

# Group state and sum the differences
state_differences = difference_data.groupby("State")["Differences"].sum().re

# Find the max state
max_state = state_differences.loc[state_differences["Differences"].idxmax(),
print("State with the maximum total difference:", max_state)

# Find the max city
max_diff_CBSA = (difference_data["2000"] - difference_data["2019"]).idxmax()
city_max = difference_data.loc[max_diff_CBSA, "Core Based Statistical Area"]
print("City with the maximum difference: ", city_max)
```

```
State with the maximum total difference:  CA
City with the maximum difference:  Portsmouth, OH
```

In [5]:
```
# Question 4
albuquerque_data = filtered_data.loc[filtered_data["City"] == "Albuquerque"]
print(albuquerque_data)
```

```
     CBSA  2000  2001  2002  2003  2004  2005  2006  2007  2008  ...  2013
\
21  10740   6.6   6.4   6.3   6.9   6.8   7.0   7.6   6.7   5.9  ...   5.7

    2014  2015  2016  2017  2018  2019  Core Based Statistical Area  \
21   6.4   6.6   5.4   5.6   5.1   6.0              Albuquerque, NM

           City  State
21  Albuquerque     NM

[1 rows x 24 columns]
```

Notes on merging (keep at bottom of notebook)

To combine datasets based on shared information, you can use the `pd.merge(A, B, how = ..., on = SHARED_COLS)` function, which will match the rows of `A` and `B` based on the shared columns `SHARED_COLS`. If `how = 'left'`, then only rows in `A` will be retained in the output (so `B` will be merged *to* `A`); conversely, if `how = 'right'`, then only rows in `B` will be retained in the output (so `A` will be merged *to* `B`).

A simple example of the use of `pd.merge` is illustrated below:

In [6]:
```
# toy data frames
A = pd.DataFrame(
    {'shared_col': ['a', 'b', 'c'],
    'x1': [1, 2, 3],
    'x2': [4, 5, 6]}
)

B = pd.DataFrame(
    {'shared_col': ['a', 'b'],
    'y1': [7, 8]}
)
```

In [7]:
```
A
```

Out[7]:

	shared_col	x1	x2
0	a	1	4
1	b	2	5
2	c	3	6

In [8]:
```
B
```

Out[8]:

	shared_col	y1
0	a	7
1	b	8

Below, if `A` and `B` are merged retaining the rows in `A`, notice that a missing value is input because `B` has no row where the shared column (on which the merging is done) has value `c`. In other words, the third row of `A` has no match in `B`.

In [9]:
```
# left join
pd.merge(A, B, how = 'left', on = 'shared_col')
```

Out[9]:

	shared_col	x1	x2	y1
0	a	1	4	7.0
1	b	2	5	8.0
2	c	3	6	NaN

If the direction of merging is reversed, and the row structure of `B` is dominant, then the third row of `A` is dropped altogether because it has no match in `B`.

In [10]:
```
# right join
pd.merge(A, B, how = 'right', on = 'shared_col')
```

Out[10]:

	shared_col	x1	x2	y1
0	a	1	4	7
1	b	2	5	8