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.

In [1]:
```
# Mini Project #1 : Jennifer Rink
```

Part I: Dataset

Merge the city information with the air quality data and tidy the dataset (see notes below). Write a brief 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 (*i.e.*, in at least 50% of instances)?
- What is PM 2.5 and why is it important?

Please write your description in narrative fashion; ***please do not list answers to the questions above one by one***. A few brief paragraphs should suffice; please limit your data description to three paragraphs or less.

Air quality data

This dataset of air quality in U.S metropolitan areas over time includes CBSA's and 9 estimated Pollutant concentration measurements for the years 2000 through 2019. A CBSA, or Core Based Statistical Area, is U.S geographically defined area that consists of one or more counties anchored by an urban center that atleast 10,000 people inhabit. There are a total of 351 CBSA's included in this data set, and they reside in 50 states and 36 territories.

This dataset has 7,020 observations of measured Pollutant concentration variables from 2000 through 2019. There are many instances of missing data but the variables O3, PM2.5-98th Percentile, and PM2.5-Weighted Annual Mean are non-missing in atleast 50% of instances.

Analyzing pollution levels is important for the health of the general public and this data set includes air pollutants like PM2.5 measurements, which are fine inhalable particle pollutions, and PM10, which are very small hazardous particles found in dust and smoke. Particulate matter like PM2.5 contains microscopic solid or liquid droplets so small they can be inhaled deep in your lungs, or even absorbed in your bloodstream, and can cause serious health problems. It is important to monitor these pollution levels so that we can keep our cities and our country safe.

Part II: Descriptive analysis

Focus on the PM2.5 measurements that are non-missing. Answer each of the following questions in a brief paragraph or two. Your paragraph(s) should indicate both your answer and a description of how you obtained it; ***please do not include codes with your answers***.

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

PM2.5 air pollution have improved in the U.S. on the whole since 2000, dropping from about a 13 µg/m3 average in 2000 to about a 5 µg/m3 average in 2019.

To find these values, I created a graph of over-all average PM values over the 20 year period. By grouping the data just by Year, I was able to find averages over years instead of by states or cities, which I will explore in the next questions. Using the average of PM2.5-Weighted Annual Means for each year on the y axis and Years on the x axis, the line graph shows a steady decline in pollution levels. Based on EPA standards, the U.S. on the whole meets the primary standard PM2.5 levels of 12.0 µg/m3 and under in 2019.

Additionally, I added a line indicating the change in averages of PM2.5-98th Percentile values over the same time period and observed an even more dramatic decrease in pollution levels. The 98th Percentile concentrations dropped from about a 34 µg/m3 average to about a 20 µg/m3 average.

Therefore, I came to the conclusion that PM2.5 air pollution has improved in the U.S. on the whole since 2000.

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

PM2.5 pollution has become less variable from city to city in the U.S. over time, dropping from about a 3.5 µg/m3 standard deviation in 2000 to about a 1.5 µg/m3 standard deviation in 2019.

To find these values, I created a graph of over-all standard deviation PM2.5 values over the 20 year period. I grouped by Year, similarly to the problem above, and found the standard deviations of PM2.5 pollution concentrations for each year and plotted those values over time. The graph produced a line that started at about a 3.5 µg/m3 standard deviation in 2000, spiked up to almost 4 µg/m3 in 2004, and dropped down to 1.5 µg/m3 in 2019.

Therefore, I came to the conclusion that PM2.5 air pollution had become less variable because the size of the standard deviations significantly decreased since the year 2000.

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

The Treasure State, **Montana**, saw the greatest improvement in PM2.5 pollution over time and the city **Portsmouth, Ohio** saw the greatest improvement in PM2.5 pollution over time.

I defined 'best improvement' as the difference between the PM2.5-Weighted Annual Means of 2019 and 2000, divided by the PM2.5-Weighted Annual Mean of 2000.

For State, I pivoted my dataframe just on State to find the averages by each state and territory. Then I created a column of average changes and sorted the dataframe in ascending order on this column and returned the first row; which gave me Montana. For City, I pivoted my dataframe on both State and City to find the averages by City. Then I followed the same steps as described above and returned the first row; which gave me Portsmouth, Ohio.

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?

The city I chose to analyze is Santa Rosa, CA; my hometown. I was curious to see how the pollution measurements changed over time because we have been extremely affected by fires for the past 5 years. I filtered and sliced the rows from the dataset and it returned two measured pollutants for Santa Rosa; O3-4th Max and PM10-2nd Max.

The EPA primary standards for O3 and PM10 are 0.070 parts per million (ppm) and 150 µg/m3 respectively. Based on the most recent measurements in 2019 for Santa Rosa, the recorded O3 was 0.056 ppm and the recorded PM10 was 45.300 µg/m3. Therefore, Santa Rosa was in compliance with EPA primary standards as of the most recent measurement.

Additionally, it is interesting to note in 2018, just one year before, the recorded PM10 was 189.700 µg/m3, which fails to meet the EPA standards. PM10 is defined as "including dust from construction sites, landfills and agriculture, wildfires and brush/waste burning, industrial sources, wind-blown dust from open lands, pollen and fragments of bacteria", which corresponds with our data because Santa Rosa was heavily affected by the Mendocino Complex Fire and the Camp Fire.

Extra credit: 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:
A pollutant that would be a good candidate for imputation based on the PM2.5 measurements would be the O3 pollutant. I chose this pollutant because its correlation coefficient (0.680921) with PM2.5 Weighted Annual Mean was the strongest of all the correlation coefficients for PM2.5, as I concluded from the correlation matrix I created below. A potential pitfall to this technique would be that it does not preserve relationships between variables (correlations) or that it would reduce the variance of the imputed/estimated variables.

Codes

In [2]:
```
# packages
import numpy as np
import pandas as pd
import altair as alt
# Suppressing the FutureWarning from latest Python update
import warnings
warnings.simplefilter(action='ignore', category=FutureWarning)
alt.data_transformers.disable_max_rows()

# raw data
air_raw = pd.read_csv('air-quality.csv')
cbsa_info = pd.read_csv('cbsa-info.csv')

## PART I
##########
air_cbsa_merged = pd.merge(air_raw, cbsa_info, how = 'left', on = 'CBSA')

#air_cbsa_merged.sort_values(by=["Pollutant"])

air_cbsa_merged['Pollutant Statistic'] = air_cbsa_merged[['Pollutant','Trend Statistic']].agg('-'.join, axis=1)


air_cbsa_merged=air_cbsa_merged.drop(columns=["Pollutant", "Trend Statistic", "Number of Trends Sites"], axis=1)

air_cbsa_merged = air_cbsa_merged.loc[:,['CBSA','Core Based Statistical Area','Pollutant Statistic',
                                         '2000', '2001', '2002', '2003', '2004', '2005', '2006',
                                         '2007', '2008', '2009', '2010', '2011',
                                         '2012', '2013', '2014', '2015', '2016', '2017', '2018', '2019']]

air_cbsa_merged.head()
```

Out[2]:

	CBSA	Core Based Statistical Area	Pollutant Statistic	2000	2001	2002	2003	2004	2005	2006	...	2010	2011	2012	2013	2014	201
0	10100	Aberdeen, SD	PM10-2nd Max	50.000	58.000	59.000	66.000	39.000	48.000	51.000	...	46.000	29.000	62.000	66.000	36.000	43.00
1	10100	Aberdeen, SD	PM2.5-Weighted Annual Mean	8.600	8.600	7.900	8.400	8.100	9.000	8.200	...	8.700	7.100	7.500	7.300	6.200	6.20
2	10100	Aberdeen, SD	PM2.5-98th Percentile	23.000	23.000	20.000	21.000	23.000	23.000	21.000	...	27.000	18.000	23.000	22.000	17.000	14.00
3	10300	Adrian, MI	O3-4th Max	0.082	0.086	0.089	0.088	0.074	0.082	0.074	...	0.066	0.076	0.087	0.064	0.068	0.06
4	10420	Akron, OH	CO-2nd Max	2.400	2.700	1.800	1.900	2.100	1.600	1.400	...	1.400	1.000	1.100	0.800	0.800	1.00

5 rows × 23 columns

In [3]:
```
# How many CBSA's are included in the dataset
air_cbsa_merged.CBSA.nunique()

# 351 unique CBSA's.
```

Out[3]: 351

In [4]:
```
# Splitting Core Based Statistical Area into State/Territory and City
merged_copy= air_cbsa_merged.copy()

merged_copy[['City', 'State']] = merged_copy["Core Based Statistical Area"].str.split(', ', expand=True)
```

```
merged_copy=merged_copy.assign(State=merged_copy['State'].str.split('-')).explode('State')

merged_copy.head()
```

Out[4]:

	CBSA	Core Based Statistical Area	Pollutant Statistic	2000	2001	2002	2003	2004	2005	2006	...	2012	2013	2014	2015	2016	201
0	10100	Aberdeen, SD	PM10-2nd Max	50.000	58.000	59.000	66.000	39.000	48.000	51.000	...	62.000	66.000	36.000	43.000	65.000	40.00
1	10100	Aberdeen, SD	PM2.5-Weighted Annual Mean	8.600	8.600	7.900	8.400	8.100	9.000	8.200	...	7.500	7.300	6.200	6.200	5.400	5.80
2	10100	Aberdeen, SD	PM2.5-98th Percentile	23.000	23.000	20.000	21.000	23.000	23.000	21.000	...	23.000	22.000	17.000	14.000	14.000	13.00
3	10300	Adrian, MI	O3-4th Max	0.082	0.086	0.089	0.088	0.074	0.082	0.074	...	0.087	0.064	0.068	0.065	0.069	0.06
4	10420	Akron, OH	CO-2nd Max	2.400	2.700	1.800	1.900	2.100	1.600	1.400	...	1.100	0.800	0.800	1.000	1.100	0.90

5 rows × 25 columns

In [5]:
```
melt_data=air_cbsa_merged.copy()

# Melt Values
melt_data = melt_data.melt(
    id_vars=['CBSA', 'Pollutant Statistic'],
    value_vars=['2000', '2001', '2002', '2003', '2004', '2005', '2006',
                '2007', '2008', '2009', '2010', '2011','2012', '2013',
                '2014', '2015', '2016', '2017', '2018', '2019'],
    var_name='Year',
    value_name='Concentration ug/m3').pivot_table(index=['CBSA', 'Year'],
                                                  columns='Pollutant Statistic',
                                                  values='Concentration ug/m3').reset_index()

# In how many states and territories do the CBSA's reside?
melt_data.shape
```

Out[5]:
```
(7020, 11)
```

In [6]:
```
# How many observations are recorded?
rows=melt_data.shape[0]
print(rows)

# 7020 observations are recorded.
```

```
7020
```

In [7]:
```
# Missingness
melt_data.isna().mean()
```

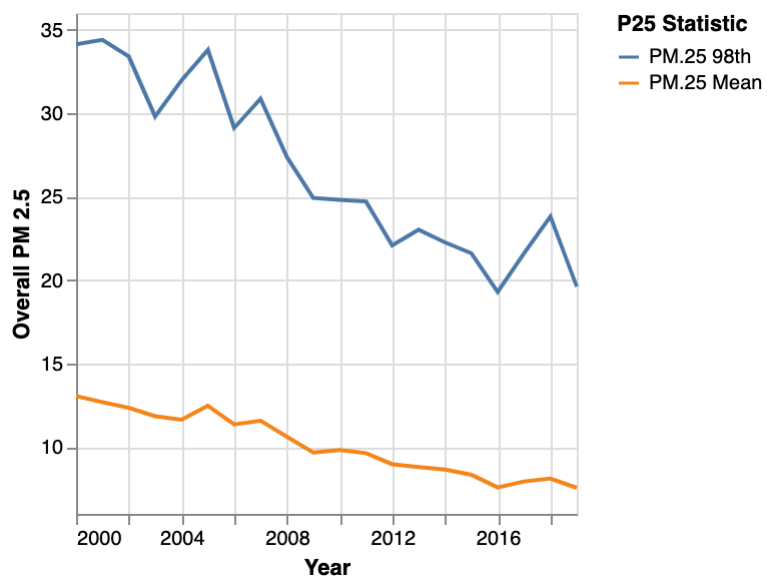
Out[7]:
```
Pollutant Statistic
CBSA                          0.000000
Year                          0.000000
CO-2nd Max                    0.831909
NO2-98th Percentile           0.809117
NO2-Annual Mean               0.746439
O3-4th Max                    0.190883
PM10-2nd Max                  0.706553
PM2.5-98th Percentile         0.390313
PM2.5-Weighted Annual Mean    0.390313
Pb-Max 3-Month Average        0.957265
SO2-99th Percentile           0.746439
dtype: float64
```

In [8]:
```
# PROB 1
# Professor Baracaldo helped me with this code in Office Hours, I altered what she gave me slightly.

MeanP25=melt_data.loc[:,['Year', 'PM2.5-Weighted Annual Mean','PM2.5-98th Percentile' ]].groupby('Year').mean().res
MeanP25=MeanP25.melt(id_vars = 'Year',
    var_name = 'P25 Statistic',
    value_name = 'P25 Value').reset_index()

alt.Chart(MeanP25).mark_line().encode(
    x = alt.X('Year:T', scale = alt.Scale(zero = False)),
    y = alt.Y('P25 Value', title = 'Overall PM 2.5', scale = alt.Scale(zero = False)),
    color = alt.Size('P25 Statistic') # change here
).properties(
    width = 250,
    height = 250
)
```

Out[8]:



In [9]:

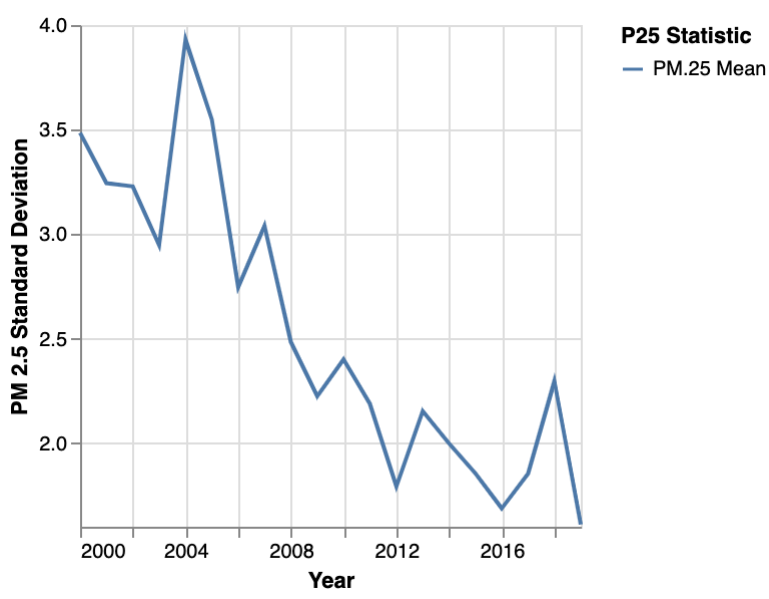
```
# PROB 2

SDP25=melt_data.loc[:,['Year', 'PM2.5-Weighted Annual Mean']].groupby(
    'Year').std().reset_index().rename(
    columns= {'PM2.5-Weighted Annual Mean': 'PM.25 Mean'})
SDP25=SDP25.melt(id_vars = 'Year',
    var_name = 'P25 Statistic',
    value_name = 'P25 Value').reset_index()

alt.Chart(SDP25).mark_line().encode(
    x = alt.X('Year:T', scale = alt.Scale(zero = False)),
    y = alt.Y('P25 Value', title = 'PM 2.5 Standard Deviation', scale = alt.Scale(zero = False)),
    color = alt.Size('P25 Statistic') # change here
).properties(
    width = 250,
    height = 250
)

# Pollution has gotten less variable.
```

Out[9]:



In [10]:

```
# PROB 3 (Most Improved State)

data_prob3=merged_copy.copy()
data_prob3=data_prob3.melt(
    id_vars=['CBSA', 'Pollutant Statistic', 'City', 'State'],
    value_vars=['2000', '2001', '2002', '2003', '2004', '2005', '2006',
                '2007', '2008', '2009', '2010', '2011','2012', '2013',
                '2014', '2015', '2016', '2017', '2018', '2019'],
    var_name='Year',
    value_name='Concentration ug/m3').pivot_table(index=['CBSA', 'Year','City', 'State'],
                                                  columns='Pollutant Statistic',
                                                  values='Concentration ug/m3').reset_index()


data_prob3=data_prob3.drop(columns=['CO-2nd Max', 'NO2-98th Percentile', 'NO2-Annual Mean',
                        'O3-4th Max', 'PM10-2nd Max', 'PM2.5-98th Percentile',
                        'Pb-Max 3-Month Average', 'SO2-99th Percentile'])

data_prob3_pt2=data_prob3.copy()

data_prob3_pt2=data_prob3_pt2.groupby(['Year', 'State']).mean()

# Pivoting Table!
data_prob3_pt2=data_prob3_pt2.reset_index().pivot(
    index="State",
    columns="Year",
    values="PM2.5-Weighted Annual Mean"
)
```

```
data_prob3_pt2['Change in PM2.5']= ((data_prob3_pt2['2019']-data_prob3_pt2['2000']) / data_prob3_pt2['2000'])


data_prob3_pt2.head()

greatest_improvement_state=data_prob3_pt2.sort_values(by='Change in PM2.5')
greatest_improvement_state.head(1)

#Answer: Alaska
```

Out[10]:

Year	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	...	2011	2012	2013	2014	2015	2016	2017	2018	2019	ii
State																					
MT	13.5	7.0	6.8	9.7	8.5	10.3	10.8	12.8	10.1	9.8	...	9.9	11.2	9.6	9.1	10.5	7.3	13.3	5.6	5.3	-0

1 rows × 21 columns

In [11]:

```
# PROB 3 (Most Improved City)

data_prob3_pt3=data_prob3.copy()

data_prob3_pt3=data_prob3_pt3.groupby(['Year', 'State', 'City']).mean()

# Pivoting Table!
data_prob3_pt3=data_prob3_pt3.reset_index().pivot(
    index=["State", "City"],
    columns="Year",
    values="PM2.5-Weighted Annual Mean"
)

data_prob3_pt3['Change in PM2.5']= ((data_prob3_pt3['2019']-data_prob3_pt3['2000']) / data_prob3_pt3['2000'])


greatest_improvement_state=data_prob3_pt3.sort_values(by='Change in PM2.5')
greatest_improvement_state.head(1)

#Answer: Portsmith, OH
```

Out[11]:

Year		2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	...	2011	2012	2013	2014	2015	2016	2017	20
State	**City**																			
OH	**Portsmouth**	21.1	20.3	16.7	14.7	12.9	16.2	14.3	14.0	12.1	10.9	...	10.1	9.8	9.0	8.2	8.5	8.3	6.9	

1 rows × 21 columns

In [12]:

```
# PROB 4: Choose City and analyze if it was compliant with EPA primary standards as of the most recent measurement?

# Currently, EPA has primary standards for PM2.5 levels of 12.0 µg/m3 and
# 24-hour standards with 98th percentile forms and levels of 35 µg/m3
# The existing primary standards for ozone (O3) are 0.070 parts per million (ppm)
# PM10 standard: 150 µg/m3

data_prob4=merged_copy.copy()

data_prob4=data_prob4.drop(columns=['Core Based Statistical Area'])


# Santa Rosa, California
data_prob4.loc[data_prob4['City'] == 'Santa Rosa']
```

Out[12]:

	CBSA	Pollutant Statistic	2000	2001	2002	2003	2004	2005	2006	2007	...	2012	2013	2014	2015	2016	2017	
921	42220	O3-4th Max	0.061	0.065	0.065	0.058	0.06	0.05	0.056	0.06	...	0.058	0.055	0.062	0.059	0.055	0.062	(
922	42220	PM10-2nd Max	32.300	35.700	29.300	25.000	25.30	27.30	27.700	26.70	...	26.000	26.700	35.000	47.700	39.000	122.000	18

2 rows × 24 columns

In [13]:

```
# IMPUTATION EXTRA CREDIT
melt_data.corr(method = "pearson")
```

Out[13]:

Pollutant Statistic	CBSA	CO-2nd Max	NO2-98th Percentile	NO2-Annual Mean	O3-4th Max	PM10-2nd Max	PM2.5-98th Percentile	PM2.5-Weighted Annual Mean	Pb-Max 3-Month Average	SO2-99th Percentile
Pollutant Statistic										
CBSA	1.000000	-0.101244	-0.201798	-0.076073	-0.017835	-0.118024	0.014909	-0.031113	0.297165	0.042302
CO-2nd Max	-0.101244	1.000000	0.566425	0.609102	0.329461	0.344990	0.282994	0.217951	0.029316	0.314266
NO2-98th Percentile	-0.201798	0.566425	1.000000	0.871111	0.572854	0.391540	0.461943	0.562625	-0.143588	0.274508
NO2-Annual Mean	-0.076073	0.609102	0.871111	1.000000	0.520512	0.415780	0.423815	0.431196	-0.172479	0.278752
O3-4th Max	-0.017835	0.329461	0.572854	0.520512	1.000000	0.099132	0.558023	0.680921	0.040126	0.348519
PM10-2nd Max	-0.118024	0.344990	0.391540	0.415780	0.099132	1.000000	0.079598	-0.045930	-0.251197	0.020013
PM2.5-98th Percentile	0.014909	0.282994	0.461943	0.423815	0.558023	0.079598	1.000000	0.738820	-0.066222	0.473462
PM2.5-Weighted Annual Mean	-0.031113	0.217951	0.562625	0.431196	0.680921	-0.045930	0.738820	1.000000	0.027223	0.637939
Pb-Max 3-Month Average	0.297165	0.029316	-0.143588	-0.172479	0.040126	-0.251197	-0.066222	0.027223	1.000000	0.209959
SO2-99th Percentile	0.042302	0.314266	0.274508	0.278752	0.348519	0.020013	0.473462	0.637939	0.209959	1.000000

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 [14]:
```
# 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 [15]:
```
A
```

Out[15]:

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

In [16]:
```
B
```

Out[16]:

	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 [17]:
```
# left join
pd.merge(A, B, how = 'left', on = 'shared_col')
```

Out[17]:

	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 [18]:
```
# right join
pd.merge(A, B, how = 'right', on = 'shared_col')
```

Out[18]:

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

Submission Checklist

1. Save file to confirm all changes are on disk
2. Run *Kernel > Restart & Run All* to execute all code from top to bottom
3. Save file again to write any new output to disk
4. Select *File > Download as > HTML*.
5. Open in Google Chrome and print to PDF.
6. Submit to Gradescope