In [4]:
```
# Import libraries
import pandas as pd
import numpy as np
```

In [52]:
```
# Read the Data and Showed 10 rows
df = pd.read_csv(r'C:\Users\HP\Desktop\Advance Data Analyst\2. Getting started with Python\4. Module 4\3. Arrays and Vectors\Files\o2_epa_air_quality.csv')
df.head(10)
```

Out[52]:

	state_code	state_name	county_code	county_name	aqi	state_code_int	county_code_int
0	4	Arizona	13	Maricopa	18.0	4	13
1	4	Arizona	13	Maricopa	9.0	4	13
2	4	Arizona	19	Pima	20.0	4	19
3	6	California	1	Alameda	11.0	6	1
4	6	California	7	Butte	6.0	6	7
5	6	California	19	Fresno	11.0	6	19
6	6	California	29	Kern	7.0	6	29
7	6	California	29	Kern	3.0	6	29
8	6	California	29	Kern	7.0	6	29
9	6	California	37	Los Angeles	13.0	6	37

In [54]:
```
# Summary data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1725 entries, 0 to 1724
Data columns (total 7 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   state_code      1725 non-null   int64 
 1   state_name      1725 non-null   object
 2   county_code     1725 non-null   int64 
 3   county_name     1725 non-null   object
 4   aqi             1725 non-null   float64
 5   state_code_int  1725 non-null   int64 
 6   county_code_int 1725 non-null   int64 
dtypes: float64(1), int64(4), object(2)
memory usage: 94.5+ KB
```

In [56]:
```
# Summary stats
df.describe()
```

Out[56]:

	state_code	county_code	aqi	state_code_int	county_code_int
count	1725.000000	1725.000000	1725.000000	1725.000000	1725.000000
mean	26.595942	83.939130	11.034783	26.595942	83.939130
std	18.702416	118.027324	10.385993	18.702416	118.027324
min	1.000000	1.000000	0.000000	1.000000	1.000000
25%	6.000000	20.000000	5.000000	6.000000	20.000000
50%	26.000000	55.000000	8.000000	26.000000	55.000000
75%	42.000000	101.000000	15.000000	42.000000	101.000000
max	80.000000	810.000000	93.000000	80.000000	810.000000

In [64]:
```
# Rows per State
df['state_name'].value_counts()
```

Out[64]:

```
state_name
California              342
Texas                   104
Pennsylvania            100
Florida                  81
Arizona                  72
Colorado                 66
Nevada                   65
Ohio                     63
Virginia                 53
New York                 51
New Jersey               45
Illinois                 37
Washington               36
North Carolina           34
Missouri                 33
Massachusetts            33
Michigan                 31
New Mexico               30
Minnesota                29
Country Of Mexico        28
Tennessee                27
Indiana                  25
Utah                     24
Kentucky                 24
Oklahoma                 22
Alabama                  22
Connecticut              21
Wisconsin                20
Montana                  20
Puerto Rico              19
Oregon                   17
Hawaii                   16
West Virginia            15
Kansas                   15
Maryland                 15
Arkansas                 14
Alaska                   14
Nebraska                 13
Iowa                     12
District Of Columbia     12
Louisiana                12
Vermont                  11
Name: count, dtype: int64
```

In [66]:
```
# Sort by Air Quality Index(AQI)
df_sorted = df.sort_values(by='aqi', ascending=False)
df_sorted.head(10)
```

Out[66]:

	state_code	state_name	county_code	county_name	aqi	state_code_int	county_code_int
253	6	California	37	Los Angeles	93.0	6	37
1324	80	Country Of Mexico	2	BAJA CALIFORNIA NORTE	79.0	80	2
116	53	Washington	61	Snohomish	76.0	53	61
107	47	Tennessee	157	Shelby	74.0	47	157
123	4	Arizona	13	Maricopa	66.0	4	13
607	4	Arizona	13	Maricopa	66.0	4	13
787	9	Connecticut	3	Hartford	61.0	9	3
980	80	Country Of Mexico	2	BAJA CALIFORNIA NORTE	60.0	80	2
125	4	Arizona	13	Maricopa	60.0	4	13
472	6	California	37	Los Angeles	59.0	6	37

In [70]:
```
# Use iloc to select rows
df_sorted.iloc[10:12]
```

Out[70]:

	state_code	state_name	county_code	county_name	aqi	state_code_int	county_code_int
173	53	Washington	77	Yakima	58.0	53	77
174	53	Washington	77	Yakima	57.0	53	77

In [72]:
```
# Basic Boolean masking to examine california data
mask = df_sorted['state_name'] == 'California'
ca_df = df_sorted[mask]
ca_df.head()
```

Out[72]:

	state_code	state_name	county_code	county_name	aqi	state_code_int	county_code_int
253	6	California	37	Los Angeles	93.0	6	37
472	6	California	37	Los Angeles	59.0	6	37
615	6	California	59	Orange	47.0	6	59
135	6	California	83	Santa Barbara	47.0	6	83
403	6	California	59	Orange	47.0	6	59

In [76]:
```
# Validate CA data
ca_df.shape
```

Out[76]:

```
(342, 7)
```

In [78]:
```
# Rows per CA county
ca_df['county_name'].value_counts()
```

Out[78]:

```
county_name
Los Angeles          55
Santa Barbara        26
San Bernardino       21
San Diego            19
Orange               19
Sacramento           17
Alameda              17
Fresno               14
Riverside            14
Contra Costa         13
Ventura              10
San Francisco         9
Monterey              8
Humboldt              8
El Dorado             7
Santa Clara           7
Placer                6
Butte                 6
Mendocino             6
Kern                  6
Tulare                5
Ventura               5
San Joaquin           5
Solano                4
Colusa                4
San Mateo             4
Marin                 3
Stanislaus            3
Sonoma                3
Napa                  2
Santa Cruz            2
San Luis Obispo       2
Calaveras             2
Shasta                1
Inyo                  1
Yolo                  1
Tuolumne              1
Mono                  1
Name: count, dtype: int64
```

In [80]:
```
# Calculate mean AQI for Los Angeles county
mask = ca_df['county_name'] == 'Los Angeles'
ca_df[mask]['aqi'].mean()
```

Out[80]:

```
13.4
```

In [86]:
```
# Groupby
# Filter the DataFrame to include only numeric columns before performing the groupby operation
numeric_cols = df.select_dtypes(include='number').columns
df.groupby('state_name')[numeric_cols].mean()[['aqi']]
```

Out[86]:

	aqi
state_name	
Alabama	7.500000
Alaska	15.714286
Arizona	16.597222
California	9.412281
Colorado	12.136364
Connecticut	12.619048
Country Of Mexico	19.071429
District Of Columbia	15.916667
Florida	11.654321
Georgia	7.071429
Hawaii	7.687500
Illinois	11.864865
Indiana	11.148148
Iowa	8.000000
Kansas	6.400000
Kentucky	8.625000
Louisiana	14.833333
Maryland	9.400000
Massachusetts	9.454545
Michigan	7.322581
Minnesota	8.896552
Missouri	7.060606
Montana	10.600000
Nebraska	15.153846
Nevada	10.323077
New Jersey	14.222222
New Mexico	12.833333
New York	9.235294
North Carolina	13.470588
Ohio	9.682540
Oklahoma	9.681818
Oregon	22.411765
Pennsylvania	6.690000
Puerto Rico	15.947368
Tennessee	15.000000
Texas	8.375000
Utah	18.192308
Vermont	11.818182
Virginia	8.588235
Washington	24.972222
West Virginia	6.600000
Wisconsin	8.100000

In [88]:
```
# Read in the second file
other_states = pd.read_csv(r'C:\Users\HP\Desktop\Advance Data Analyst\2. Getting started with Python\4. Module 4\3. Arrays and Vectors\Files\epa_others.csv')
other_states.head(10)
```

Out[88]:

	state_code	state_name	county_code	county_name	aqi
0	4	Arizona	13	Maricopa	18.0
1	4	Arizona	13	Maricopa	9.0
2	4	Arizona	19	Pima	20.0
3	8	Colorado	41	El Paso	9.0
4	12	Florida	31	Duval	15.0
5	12	Florida	31	Duval	13.0
6	12	Florida	57	Hillsborough	19.0
7	15	Hawaii	3	Honolulu	10.0
8	17	Illinois	167	Sangamon	20.0
9	18	Indiana	97	Marion	32.0

In [90]:
```
# Concatenate the data
combined_df = pd.concat([df, other_states], axis=0)
len(combined_df) == len(df) + len(other_states)
```

Out[90]:

```
True
```

In [92]:
```
# Complex Boolean masking
mask = (combined_df['state_name'] == 'Washington') & (combined_df['aqi'] == 51)
combined_df[mask]
```

Out[92]:

	state_code	state_name	county_code	county_name	aqi	state_code_int	county_code_int
57	53	Washington	33	King	55.0	53.0	33.0
116	53	Washington	61	Snohomish	76.0	53.0	61.0
173	53	Washington	77	Yakima	58.0	53.0	77.0
174	53	Washington	77	Yakima	57.0	53.0	77.0
40	53	Washington	33	King	55.0	NaN	NaN
82	53	Washington	61	Snohomish	76.0	NaN	NaN
121	53	Washington	77	Yakima	58.0	NaN	NaN
122	53	Washington	77	Yakima	57.0	NaN	NaN

In [ ]:
```
# Conclusion
# It is comes with many built-in functions and tools specifically designed for use with tabular data to simplify common tasks such as:

# Reading and writing data to/from files
# Quickly computing summary statistics about your data
# Manipulating, selecting, and filtering data
# Grouping and aggregating data
```

```
# Adding new data to existing data
# It's powered by NumPy, which uses the power of array operations to enhance performance.

# Its interface makes working with tabular data easy because it allows you to visualize your data in rows and columns.
```