AQI Data (Year- 2019, Year- 2020, Year-2021)

In [1]:
```
#importing libraries
import pandas as pd
import numpy as np
```

In [2]:
```
#Reading the AQI datasets of Year- 2019, Year- 2020, Year-2021
AQI_df1 = pd.read_csv('/Users/vishesh/Downloads/DA/daily_aqi_by_county_2019.csv
AQI_df2 = pd.read_csv('/Users/vishesh/Downloads/DA/daily_aqi_by_county_2020.csv
AQI_df3 = pd.read_csv('/Users/vishesh/Downloads/DA/daily_aqi_by_county_2021.csv
```

Checking the Dataframe

In [3]:
```
# Top 5 Rows of AQI_Year-2019
AQI_df1.head()
```

Out[3]:

	State Name	county Name	State Code	County Code	Date	AQI	Category	Defining Parameter	Defining Site	Number of Sites Reporting
0	Alabama	Baldwin	1	3	2019-01-03	18	Good	PM2.5	01-003-0010	1
1	Alabama	Baldwin	1	3	2019-01-06	35	Good	PM2.5	01-003-0010	1
2	Alabama	Baldwin	1	3	2019-01-09	14	Good	PM2.5	01-003-0010	1
3	Alabama	Baldwin	1	3	2019-01-12	36	Good	PM2.5	01-003-0010	1
4	Alabama	Baldwin	1	3	2019-01-15	38	Good	PM2.5	01-003-0010	1

In [4]:
```
# Top 5 Rows of AQI_Year-2020
AQI_df2.head()
```

Out[4]:

	State Name	county Name	State Code	County Code	Date	AQI	Category	Defining Parameter	Defining Site	Number of Sites Reporting
0	Alabama	Baldwin	1	3	2020-01-01	48	Good	PM2.5	01-003-0010	1
1	Alabama	Baldwin	1	3	2020-01-04	13	Good	PM2.5	01-003-0010	1
2	Alabama	Baldwin	1	3	2020-01-07	14	Good	PM2.5	01-003-0010	1
3	Alabama	Baldwin	1	3	2020-01-10	39	Good	PM2.5	01-003-0010	1
4	Alabama	Baldwin	1	3	2020-01-13	29	Good	PM2.5	01-003-0010	1

In [5]:
```
# Top 5 Rows of AQI_Year-2021
AQI_df3.head()
```

Out[5]:

	State Name	county Name	State Code	County Code	Date	AQI	Category	Defining Parameter	Defining Site	Number of Sites Reporting
0	Alabama	Baldwin	1	3	2021-01-01	27	Good	PM2.5	01-003-0010	1
1	Alabama	Baldwin	1	3	2021-01-04	47	Good	PM2.5	01-003-0010	1
2	Alabama	Baldwin	1	3	2021-01-07	24	Good	PM2.5	01-003-0010	1
3	Alabama	Baldwin	1	3	2021-01-10	39	Good	PM2.5	01-003-0010	1
4	Alabama	Baldwin	1	3	2021-01-13	46	Good	PM2.5	01-003-0010	1

Data Wrangling and Cleaning

In [6]:
```
# Summary of the AQI_df1 DataFrame, including the number of non-null values in
AQI_df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 325331 entries, 0 to 325330
Data columns (total 10 columns):
 #   Column                    Non-Null Count   Dtype 
---  ------                    --------------   ----- 
 0   State Name                325331 non-null  object
 1   county Name               325331 non-null  object
 2   State Code                325331 non-null  int64 
 3   County Code               325331 non-null  int64 
 4   Date                      325331 non-null  object
 5   AQI                       325331 non-null  int64 
 6   Category                  325331 non-null  object
 7   Defining Parameter        325331 non-null  object
 8   Defining Site             325331 non-null  object
 9   Number of Sites Reporting 325331 non-null  int64 
dtypes: int64(4), object(6)
memory usage: 24.8+ MB
```

In [7]:
```
# Summary of the AQI_df2 DataFrame, including the number of non-null values in
AQI_df2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 324338 entries, 0 to 324337
Data columns (total 10 columns):
 #   Column                     Non-Null Count   Dtype 
---  ------                     --------------   ----- 
 0   State Name                 324338 non-null  object
 1   county Name                324338 non-null  object
 2   State Code                 324338 non-null  int64 
 3   County Code                324338 non-null  int64 
 4   Date                       324338 non-null  object
 5   AQI                        324338 non-null  int64 
 6   Category                   324338 non-null  object
 7   Defining Parameter         324338 non-null  object
 8   Defining Site              324338 non-null  object
 9   Number of Sites Reporting  324338 non-null  int64 
dtypes: int64(4), object(6)
memory usage: 24.7+ MB
```

In [8]:
```
# Summary of the AQI_df3 DataFrame, including the number of non-null values in
AQI_df3.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 218196 entries, 0 to 218195
Data columns (total 10 columns):
 #   Column                     Non-Null Count   Dtype 
---  ------                     --------------   ----- 
 0   State Name                 218196 non-null  object
 1   county Name                218196 non-null  object
 2   State Code                 218196 non-null  int64 
 3   County Code                218196 non-null  int64 
 4   Date                       218196 non-null  object
 5   AQI                        218196 non-null  int64 
 6   Category                   218196 non-null  object
 7   Defining Parameter         218196 non-null  object
 8   Defining Site              218196 non-null  object
 9   Number of Sites Reporting  218196 non-null  int64 
dtypes: int64(4), object(6)
memory usage: 16.6+ MB
```

In [9]:
```
# checking null values in AQI_df1 DataFrame
AQI_df1.isna().sum()
```

Out[9]:
```
State Name                   0
county Name                  0
State Code                   0
County Code                  0
Date                         0
AQI                          0
Category                     0
Defining Parameter           0
Defining Site                0
Number of Sites Reporting    0
dtype: int64
```

In [10]:
```
# checking null values in AQI_df2 DataFrame
AQI_df2.isna().sum()
```

Out[10]:
```
State Name                    0
county Name                   0
State Code                    0
County Code                   0
Date                          0
AQI                           0
Category                      0
Defining Parameter            0
Defining Site                 0
Number of Sites Reporting     0
dtype: int64
```

In [11]:
```
# checking null values in AQI_df3 DataFrame
AQI_df3.isna().sum()
```

Out[11]:
```
State Name                    0
county Name                   0
State Code                    0
County Code                   0
Date                          0
AQI                           0
Category                      0
Defining Parameter            0
Defining Site                 0
Number of Sites Reporting     0
dtype: int64
```

According to the 3 Dataframes it means that all the three datasets have the same number of columns and the column names are also the same in all the three datasets. So, we can concatenate the three datasets together into one and the resulting DataFrame will have the same structure as the individual DataFrames. It will be easier to analyze and work with one large dataset rather than working with three different datasets.

In [12]:
```
# Concateninig the three AQI Dataframes
AQI_df = pd.concat([AQI_df1, AQI_df2, AQI_df3], ignore_index=True)
```

In [13]:
```
# Checking the Concatenated dataframe first few rows
AQI_df.head()
```

Out[13]:

	State Name	county Name	State Code	County Code	Date	AQI	Category	Defining Parameter	Defining Site	Number of Sites Reporting
0	Alabama	Baldwin	1	3	2019-01-03	18	Good	PM2.5	01-003-0010	1
1	Alabama	Baldwin	1	3	2019-01-06	35	Good	PM2.5	01-003-0010	1
2	Alabama	Baldwin	1	3	2019-01-09	14	Good	PM2.5	01-003-0010	1
3	Alabama	Baldwin	1	3	2019-01-12	36	Good	PM2.5	01-003-0010	1
4	Alabama	Baldwin	1	3	2019-01-15	38	Good	PM2.5	01-003-0010	1

In [14]:
```
# Checking the Concatenated dataframe last few rows
AQI_df.tail()
```

Out[14]:

	State Name	county Name	State Code	County Code	Date	AQI	Category	Defining Parameter	Defining Site	Numbe of Site Reportin
867860	Wyoming	Weston	56	45	2021-06-26	41	Good	Ozone	56-045-0003	
867861	Wyoming	Weston	56	45	2021-06-27	40	Good	Ozone	56-045-0003	
867862	Wyoming	Weston	56	45	2021-06-28	41	Good	Ozone	56-045-0003	
867863	Wyoming	Weston	56	45	2021-06-29	48	Good	Ozone	56-045-0003	
867864	Wyoming	Weston	56	45	2021-06-30	54	Moderate	Ozone	56-045-0003	

In [15]:
```
# Checking the number of columns and rowa
AQI_df.shape
```

Out[15]:
```
(867865, 10)
```

In [16]:
```
# Droping unnecessary columns from the dataframe
AQI_df = AQI_df.drop(columns = ['Defining Parameter', 'Defining Site', 'Number
```

In [17]:
```
# Checking the random rows
AQI_df.sample(5)
```

Out[17]:

	State Name	State Code	Date	AQI	Category
808471	Oregon	41	2021-04-14	16	Good
577331	South Carolina	45	2020-04-30	11	Good
265903	Tennessee	47	2019-09-14	49	Good
313491	Wisconsin	55	2019-10-12	32	Good
230385	Oregon	41	2019-05-13	19	Good

In [18]:
```
# Checking the final information again in order to process further
AQI_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 867865 entries, 0 to 867864
Data columns (total 5 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   State Name   867865 non-null  object
 1   State Code   867865 non-null  int64 
 2   Date         867865 non-null  object
 3   AQI          867865 non-null  int64 
 4   Category     867865 non-null  object
dtypes: int64(2), object(3)
memory usage: 33.1+ MB
```

It's important to make sure that the data in the column is in the format that can be converted to the desired datatype, otherwise it will throw an error.

In [19]:
```
# converting the 'Date' column of AQI_df DataFrame into datetime datatype.
AQI_df['Date'] = pd.to_datetime(AQI_df['Date'])

# converting the 'State Name' column of AQI_df DataFrame into a string datatype
AQI_df['State_Name'] = AQI_df['State Name'].astype(str)
```

In [20]:
```
# Setting the index
AQI_df.set_index("Date", inplace=True)
```

In [21]:
```
# Changing the format of date to Y-M
AQI_df['year_month'] = AQI_df.index.strftime('%Y-%m')


# Grouping the AQI_df DataFrame by 'State_Name' and 'year_month' columns and th
df_monthly = AQI_df.groupby(['State_Name', 'year_month'], as_index=False).mean(
```

It is a useful technique when you want to analyze or visualize data based on a specific time period (in this case, month) or when you want to aggregate the data based on a specific column.

In [22]:
```
df_monthly.head()
```

Out[22]:

	State_Name	year_month	State Code	AQI
0	Alabama	2019-01	1.0	32.812925
1	Alabama	2019-02	1.0	31.391608
2	Alabama	2019-03	1.0	43.827214
3	Alabama	2019-04	1.0	44.484979
4	Alabama	2019-05	1.0	42.683992

In [23]:
```
df_monthly.shape
```

Out[23]:
```
(1796, 4)
```

In [24]:
```
df_monthly.dtypes
```

Out[24]:
```
State_Name     object
year_month     object
State Code    float64
AQI           float64
dtype: object
```

In [25]:
```
# changing the format of statecode type from float to INT
df_monthly['State Code'] = df_monthly['State Code'].astype(int)
```

In [26]:
```
import warnings
warnings.filterwarnings('ignore')
df_monthly['year_month'] = pd.to_datetime(df_monthly['year_month'], format='%Y-
df_monthly['Month'] = df_monthly['year_month'].dt.strftime('%m')
df_monthly['Year'] = df_monthly['year_month'].dt.strftime('%y')
df_monthly['Month'] = df_monthly['Month'].astype(int)
df_monthly['Year'] = df_monthly['Year'].astype(str)
```

In [27]:
```
df_monthly.head()
```

Out[27]:

	State_Name	year_month	State Code	AQI	Month	Year
0	Alabama	2019-01-01	1	32.812925	1	19
1	Alabama	2019-02-01	1	31.391608	2	19
2	Alabama	2019-03-01	1	43.827214	3	19
3	Alabama	2019-04-01	1	44.484979	4	19
4	Alabama	2019-05-01	1	42.683992	5	19

In [28]:
```
df_monthly['State_Name'].unique()
```

Out[28]:
```
array(['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Country Of Mexico', 'Delaware',
       'District Of Columbia', 'Florida', 'Georgia', 'Hawaii', 'Idaho',
       'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana',
       'Maine', 'Maryland', 'Massachusetts', 'Michigan', 'Minnesota',
       'Mississippi', 'Missouri', 'Montana', 'Nebraska', 'Nevada',
       'New Hampshire', 'New Jersey', 'New Mexico', 'New York',
       'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'Puerto Rico', 'Rhode Island', 'South Carolina',
       'South Dakota', 'Tennessee', 'Texas', 'Utah', 'Vermont',
       'Virgin Islands', 'Virginia', 'Washington', 'West Virginia',
       'Wisconsin', 'Wyoming'], dtype=object)
```

In [29]:
```
country_list=['Alabama', 'Alaska', 'Arizona', 'Arkansas', 'California',
       'Colorado', 'Connecticut', 'Country Of Mexico', 'Delaware',
       'District Of Columbia', 'Florida', 'Georgia', 'Hawaii', 'Idaho',
       'Illinois', 'Indiana', 'Iowa', 'Kansas', 'Kentucky', 'Louisiana',
       'Maine', 'Maryland', 'Massachusetts', 'Michigan', 'Minnesota',
       'Mississippi', 'Missouri', 'Montana', 'Nebraska', 'Nevada',
       'New Hampshire', 'New Jersey', 'New Mexico', 'New York',
       'North Carolina', 'North Dakota', 'Ohio', 'Oklahoma', 'Oregon',
       'Pennsylvania', 'Puerto Rico', 'Rhode Island', 'South Carolina',
       'South Dakota', 'Tennessee', 'Texas', 'Utah', 'Vermont',
       'Virgin Islands', 'Virginia', 'Washington', 'West Virginia',
       'Wisconsin', 'Wyoming']
```

{ Data Visualization }

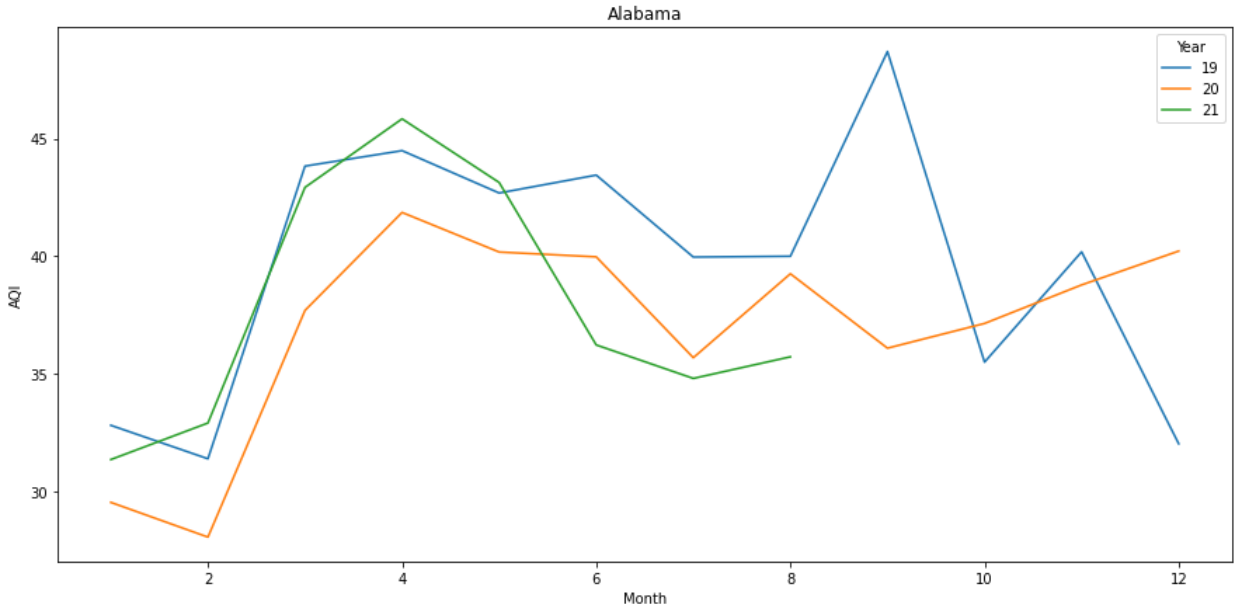
In [30]:
```
# importing these libraries for data visualization and creating different types

# Seaborn library
import seaborn as sns

#matplotlib library
import matplotlib
import matplotlib.pyplot as plt
```

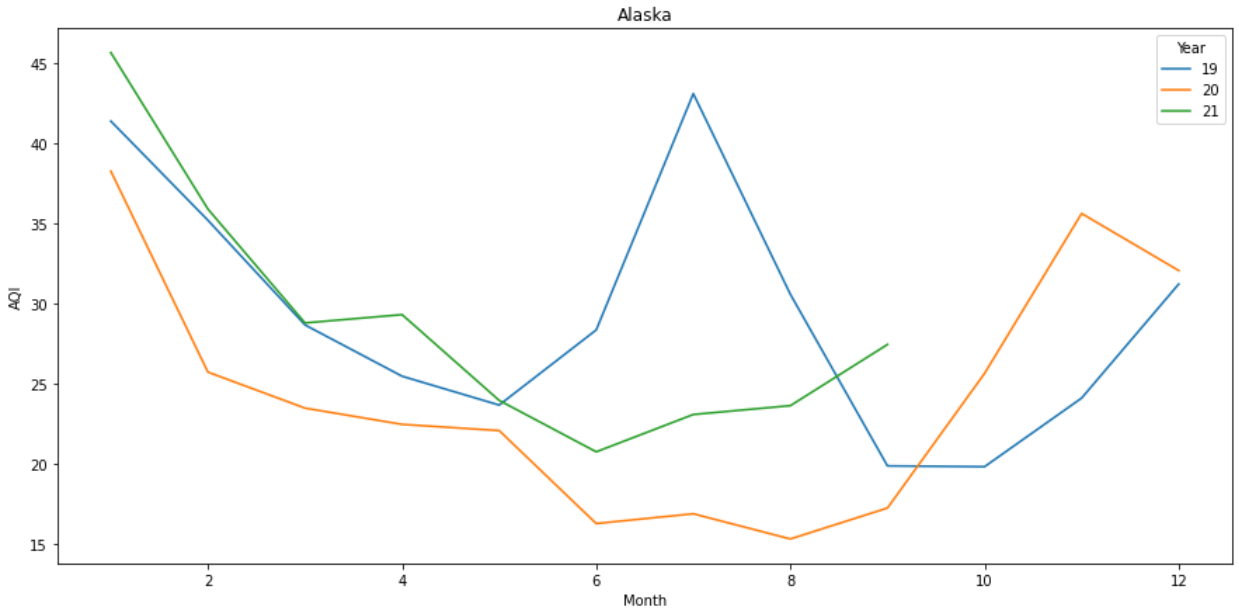
The below cell creates line plots of AQI data for each country over a 3-year period (2019-2020-2021), with the blue line representing the data for the year 2019, the orange line representing the data for the year 2020, and the green line representing the data for the year 2021.

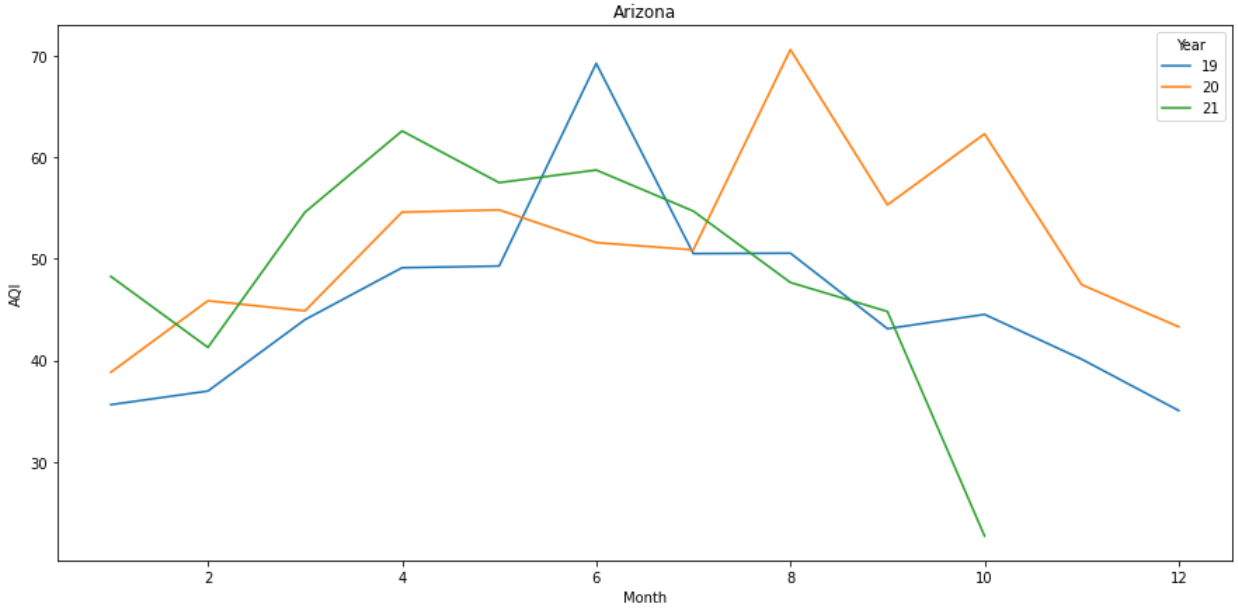
This allows for easy visualization of AQI trends for each country over the 3-year period and makes it easy to see how AQI levels changed during the lockdown period from month February to April .

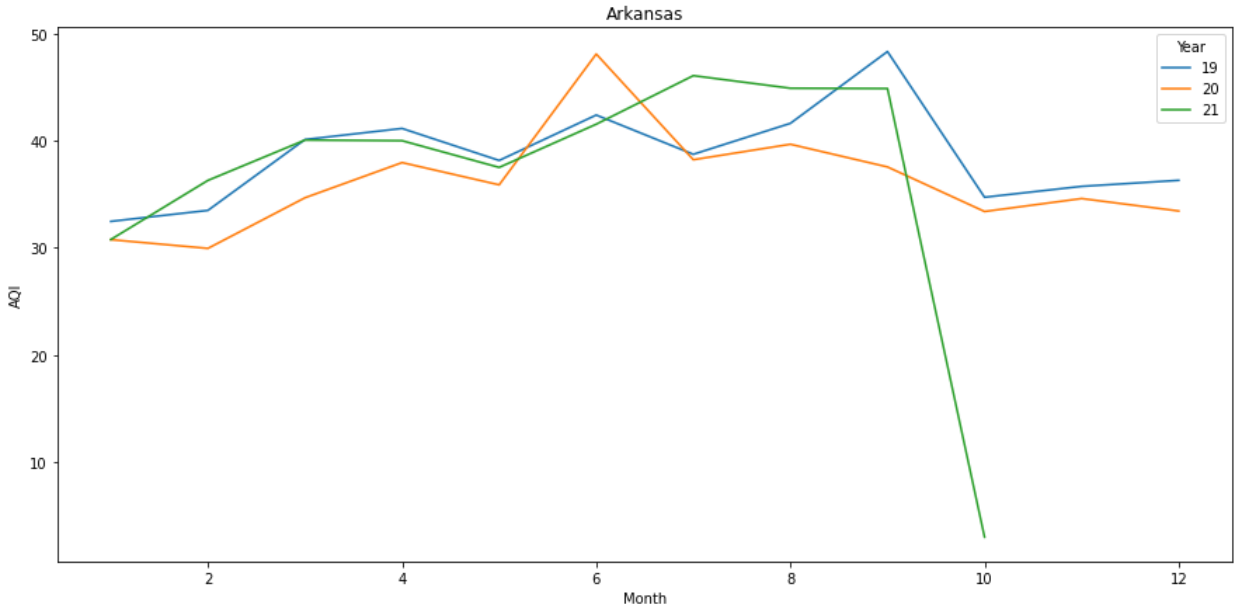
It is easy to understand if the AQI was high or low during the lockdown period for each country, and it also allows to compare AQI levels between different years. Overall, the visualizations of all the countries provides a useful charts for analyzing AQI data and understanding how it has changed over time.

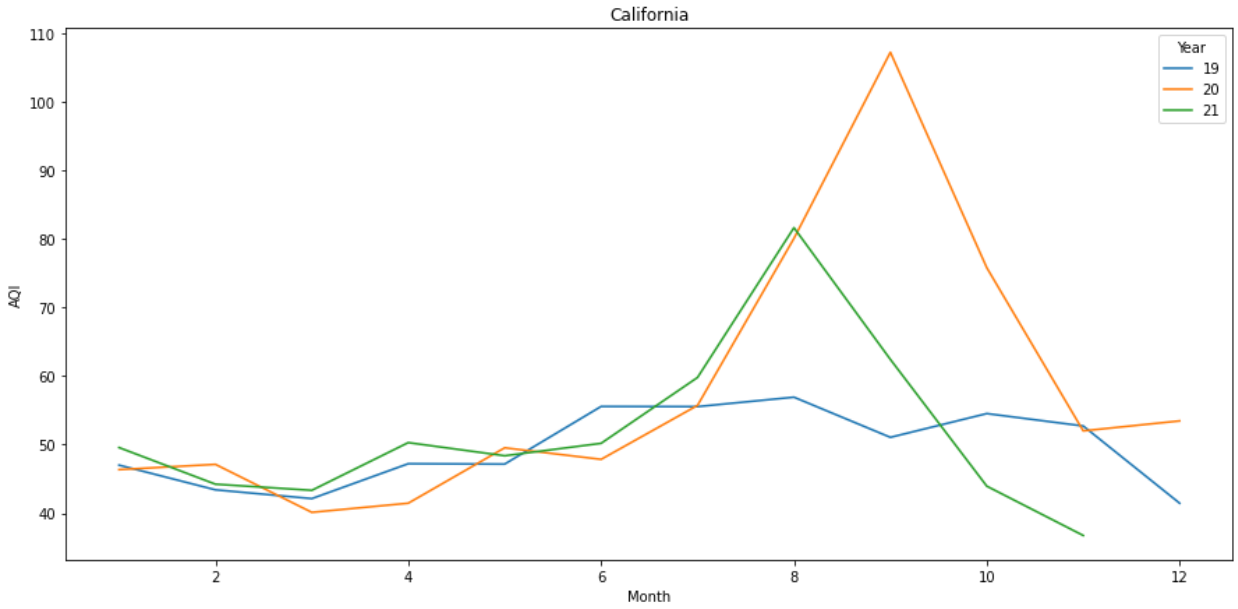
In [31]:
```
for country in country_list:
  cn=[]
  cn.append(country)
  df_plotter = pd.DataFrame()
  df_plotter=df_monthly[df_monthly['State_Name'].isin(cn)]

  plt.figure(figsize=(15,7))
  country= sns.lineplot(x='Month',y='AQI',data=df_plotter, hue='Year').set(titl
```

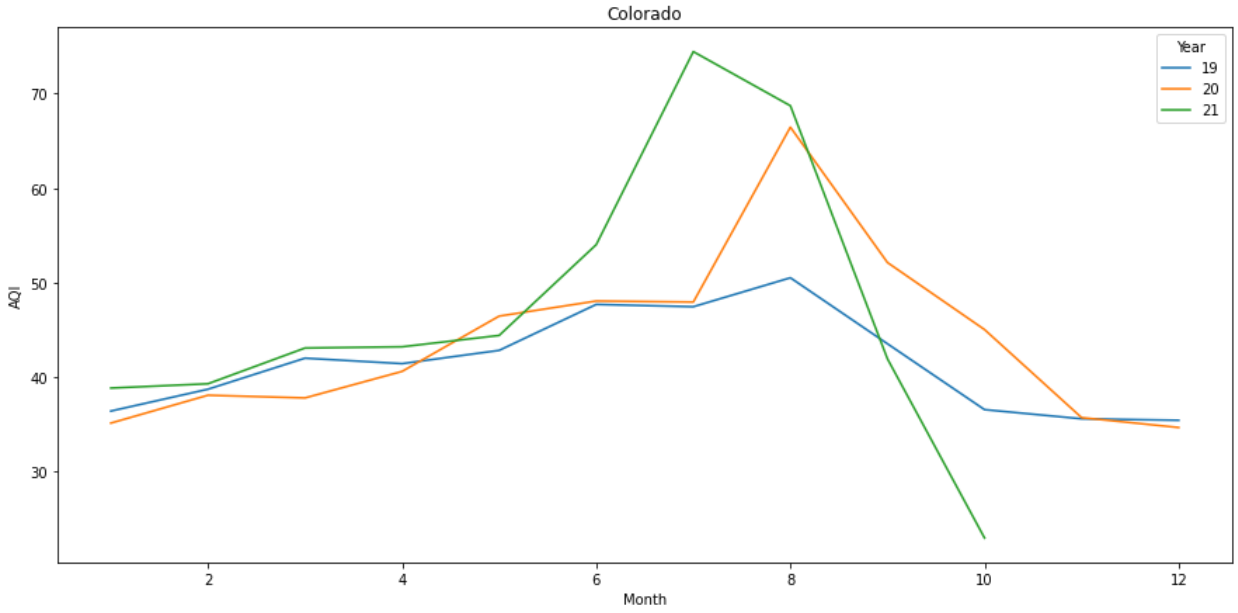


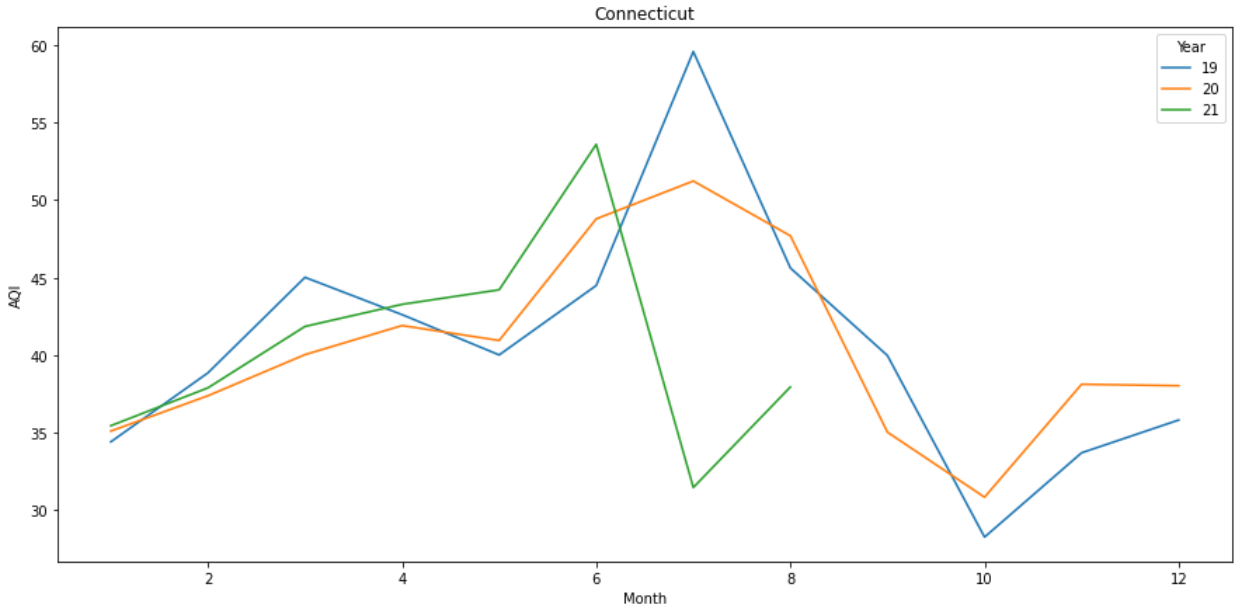


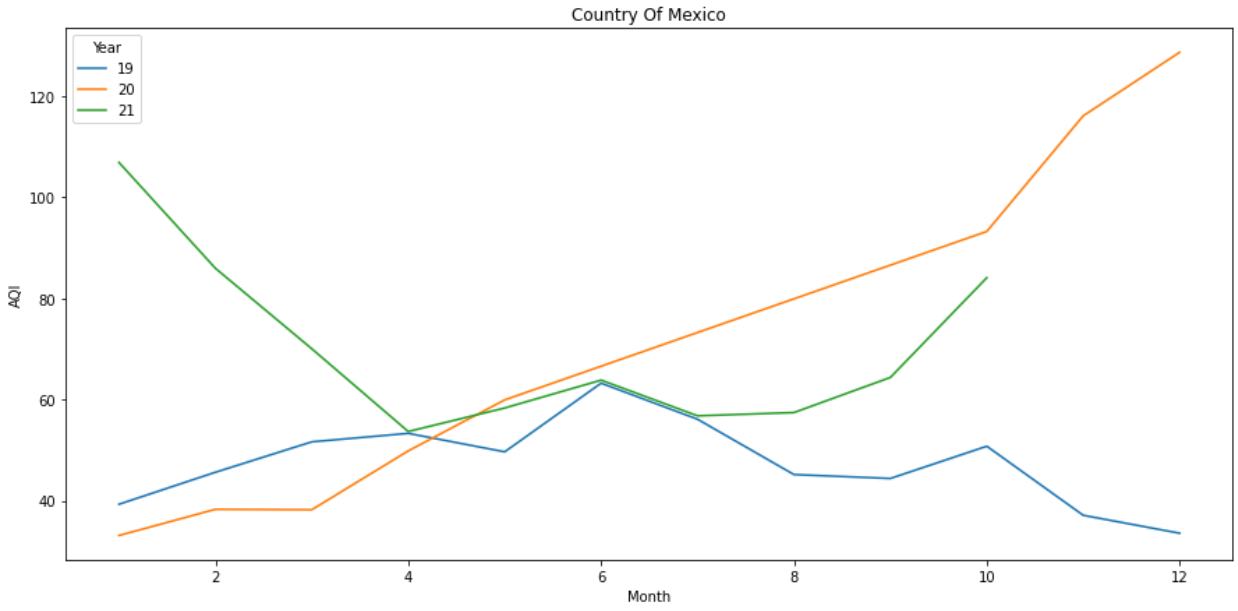

Arizona
AQI
Month
Year
19
20
21

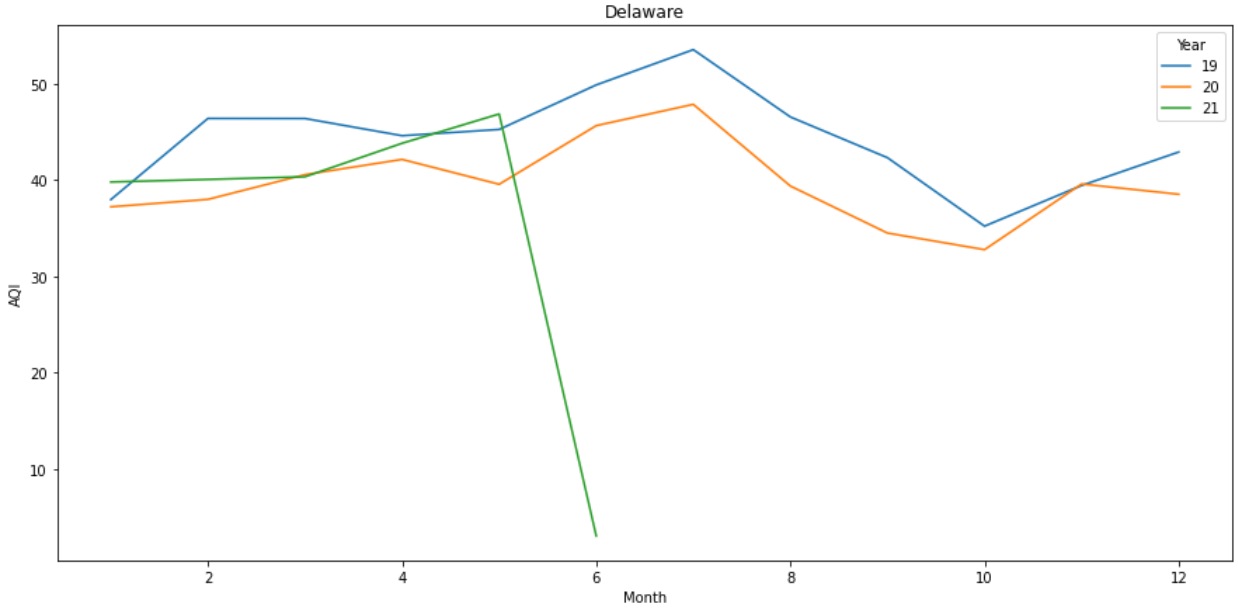

Arkansas
AQI
Month
Year
19
20
21

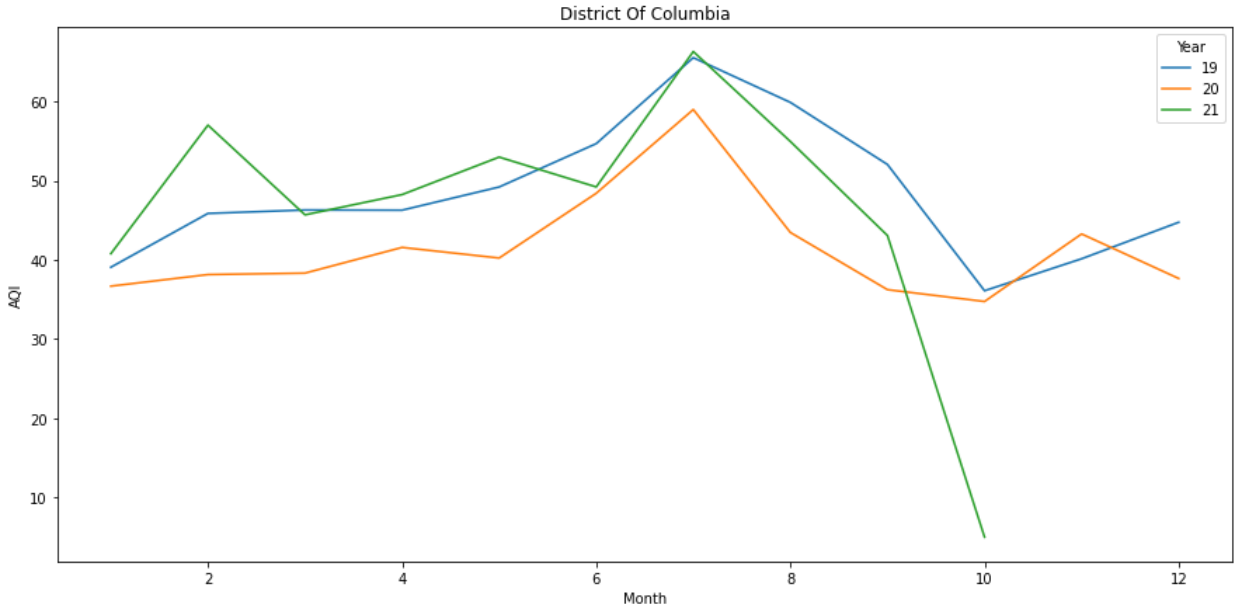

California
AQI
Month
Year
19
20
21

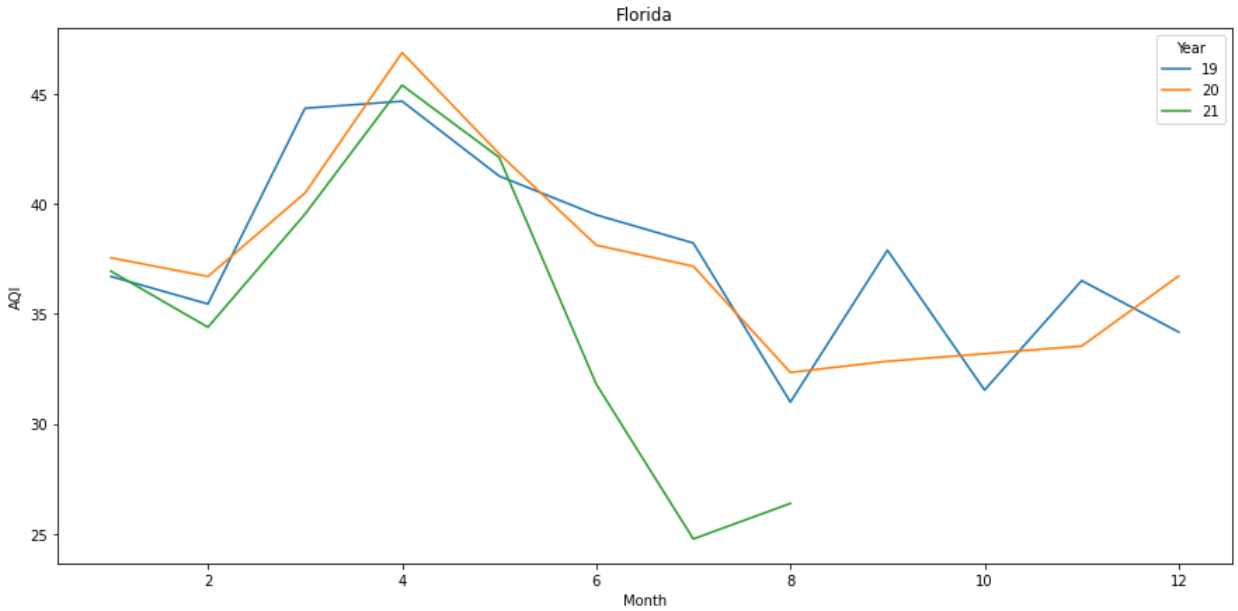

Colorado
Year
19
20
21
AQI
Month
70
60
50
40
30
2
4
6
8
10
12

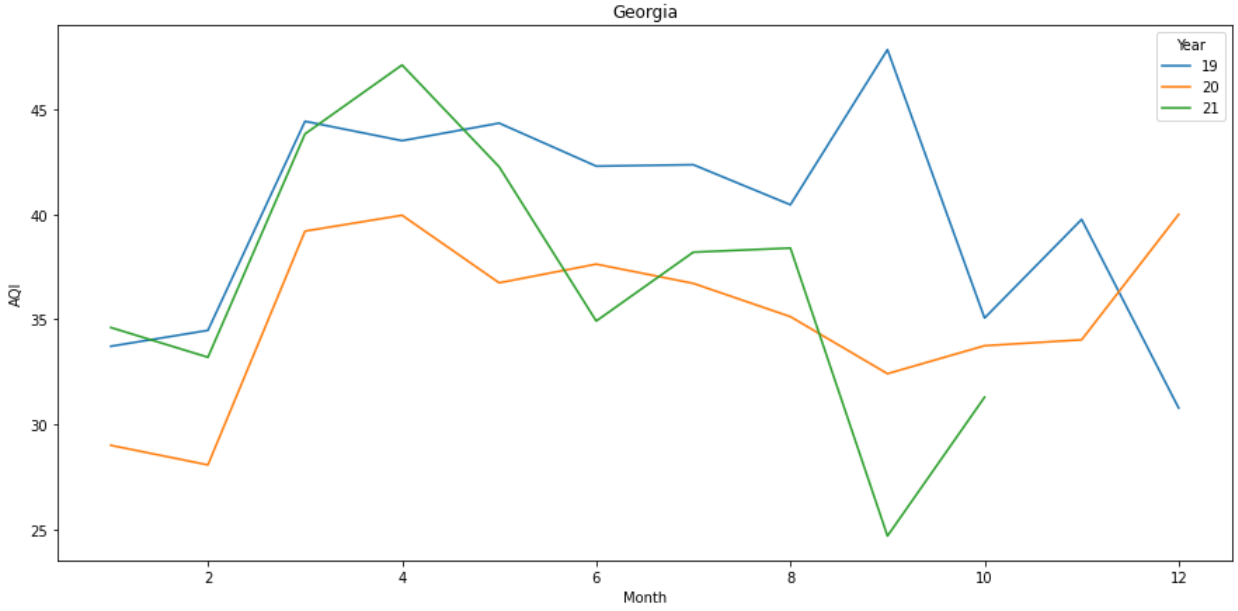

Connecticut
Year
19
20
21
AQI
Month
60
55
50
45
40
35
30
2
4
6
8
10
12

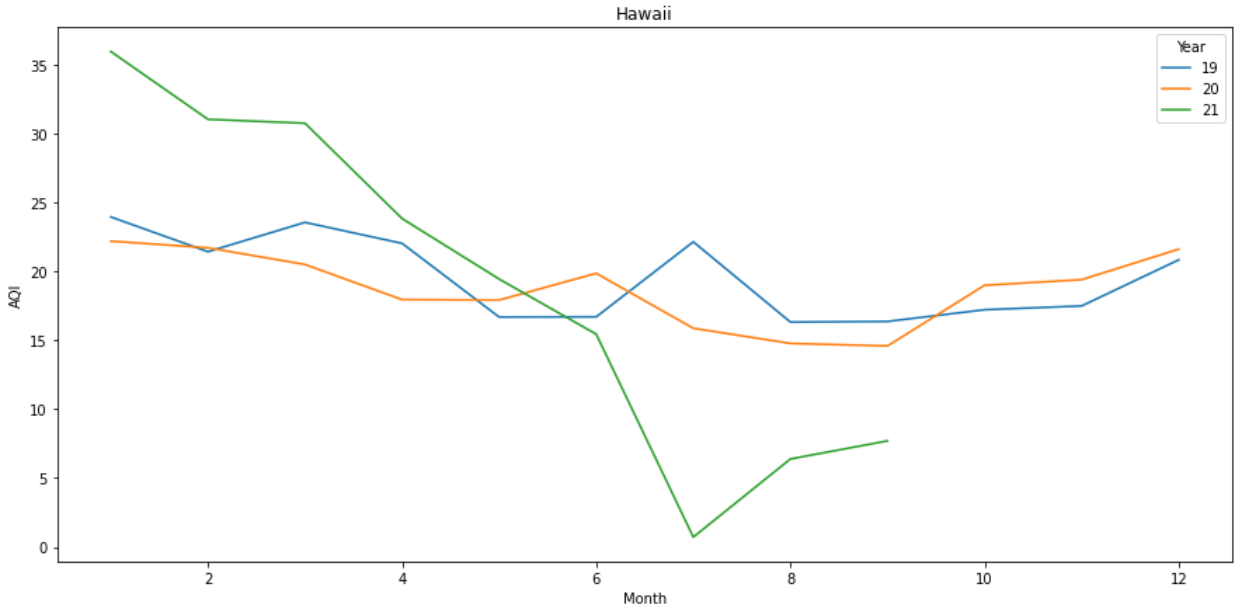

Country Of Mexico
Year
19
20
21
AQI
Month
120
100
80
60
40
2
4
6
8
10
12

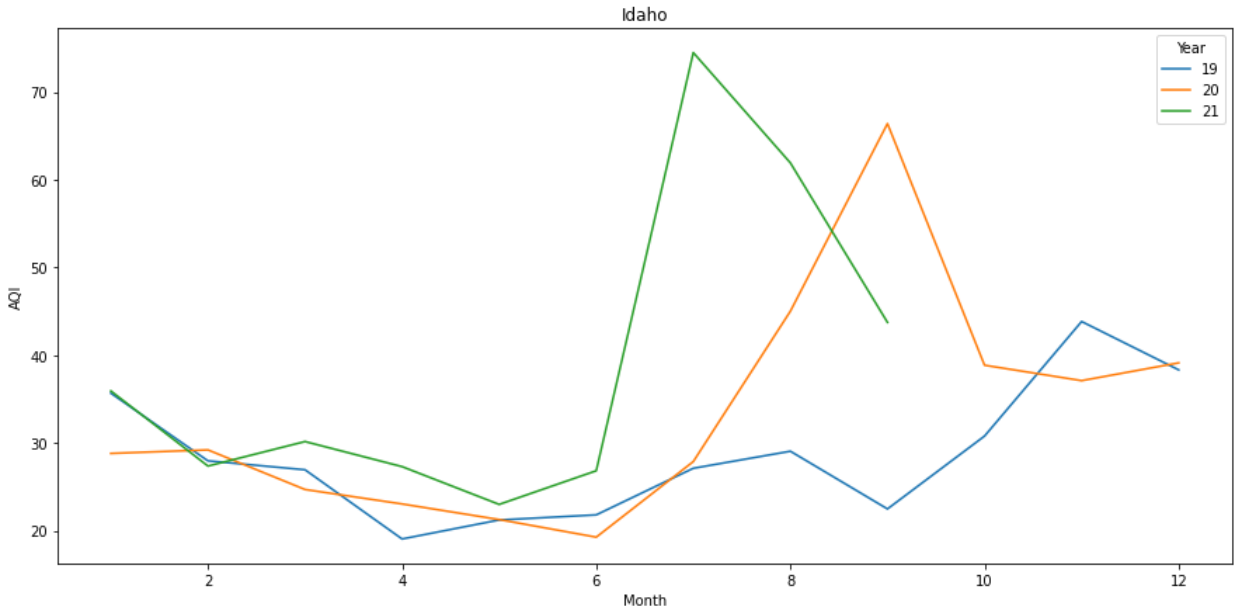

Delaware
Year
19
20
21
AQI
Month
10
20
30
40
50
2
4
6
8
10
12

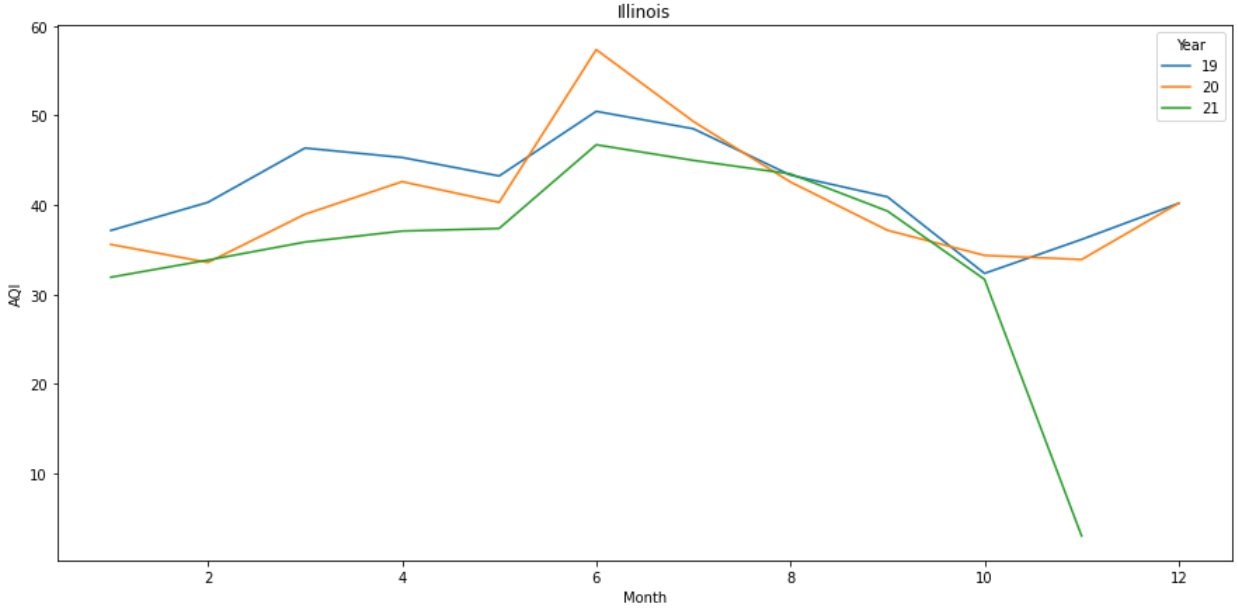

District Of Columbia
Year
19
20
21
AQI
Month
10
20
30
40
50
60
2
4
6
8
10
12

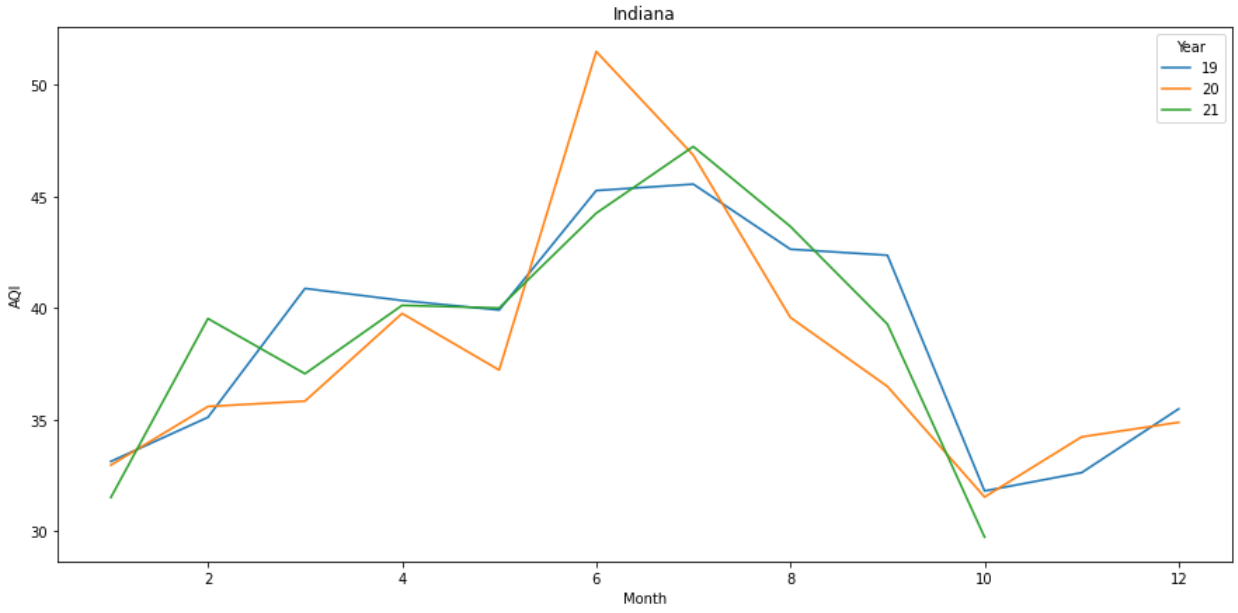

Florida
Year
19
20
21
AQI
Month
25
30
35
40
45
2
4
6
8
10
12

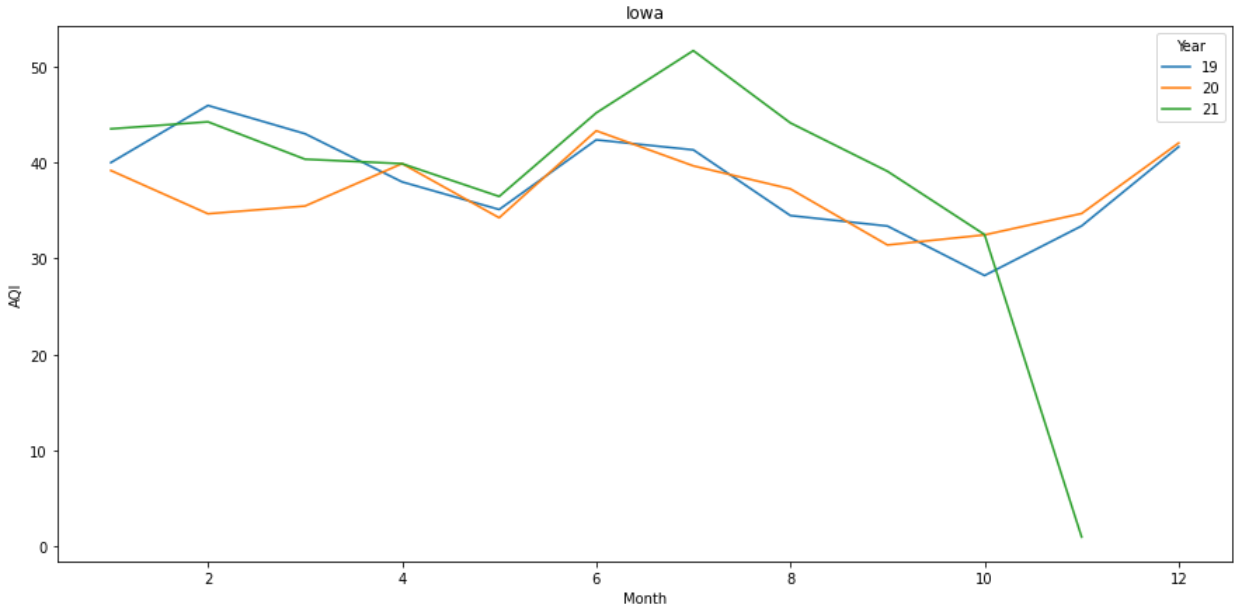

Georgia
Year
19
20
21
AQI
Month
45
40
35
30
25
2
4
6
8
10
12

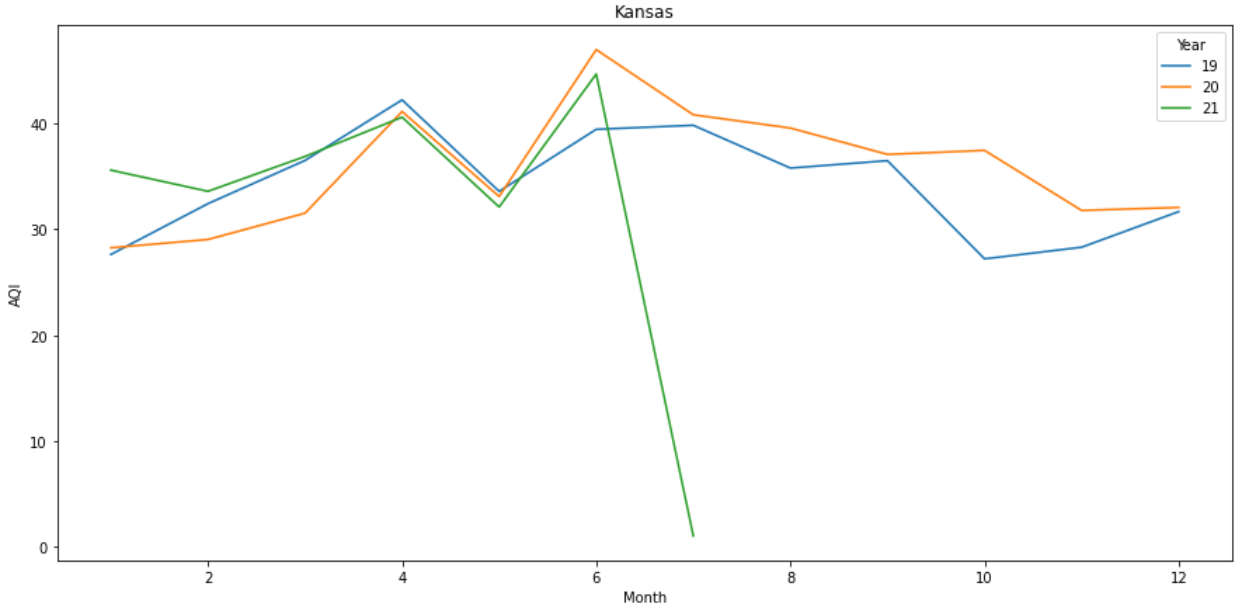

Hawaii
Year
19
20
21
AQI
Month
35
30
25
20
15
10
5
0
2
4
6
8
10
12

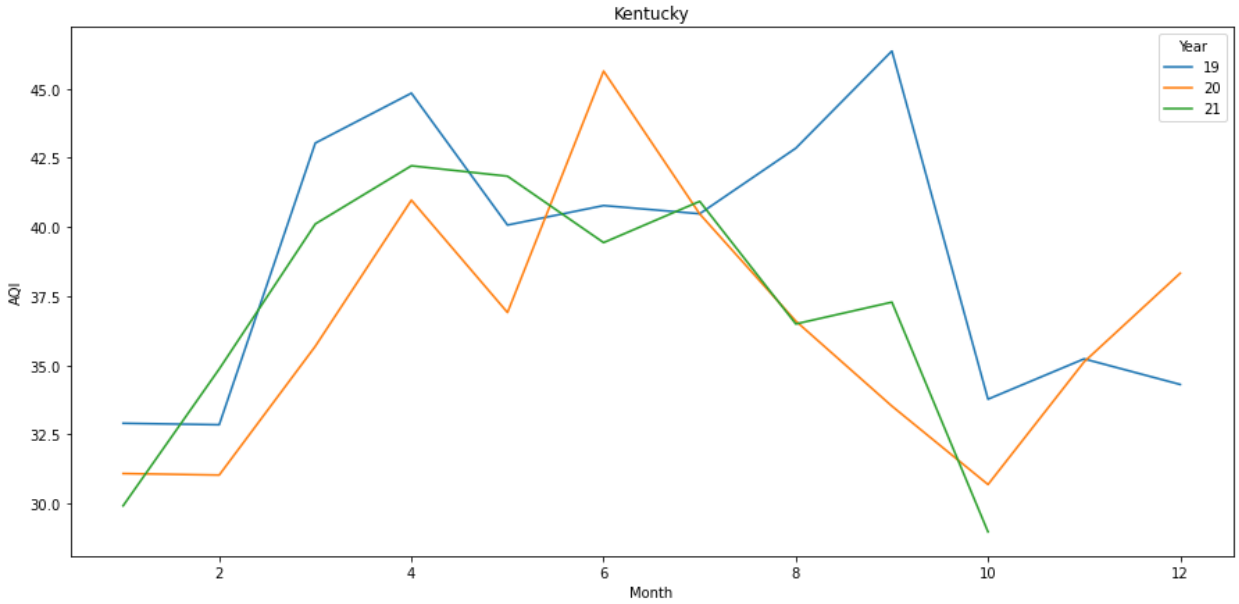

Idaho
Year
19
20
21
AQI
Month
70
60
50
40
30
20
2
4
6
8
10
12

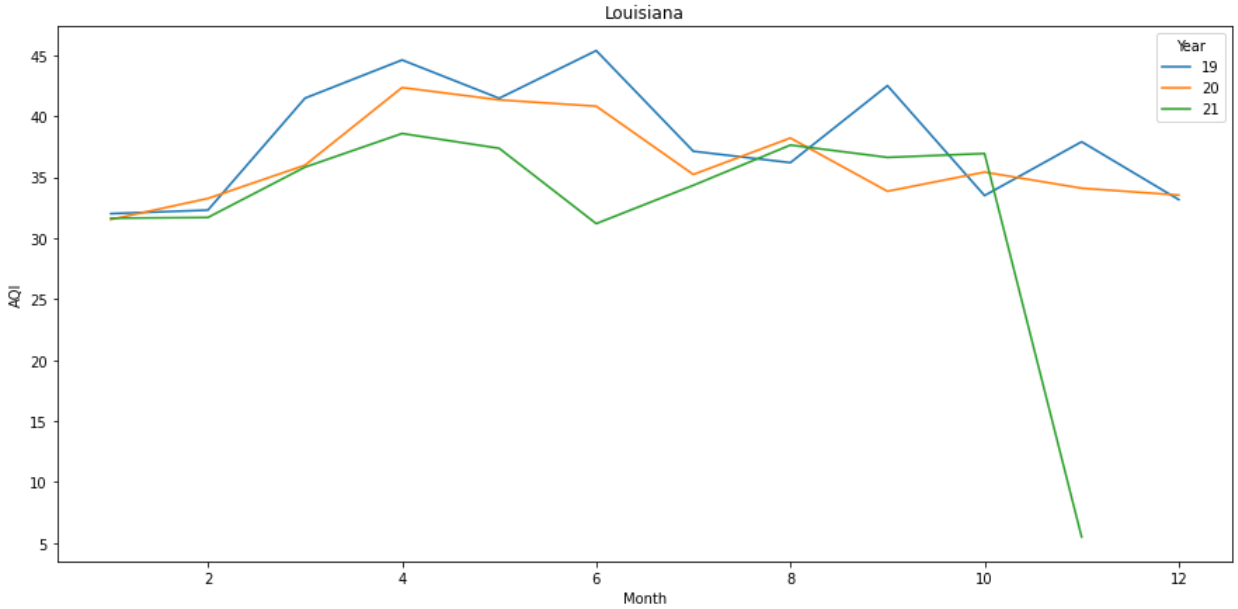

Illinois
AQI
Month
Year
19
20
21

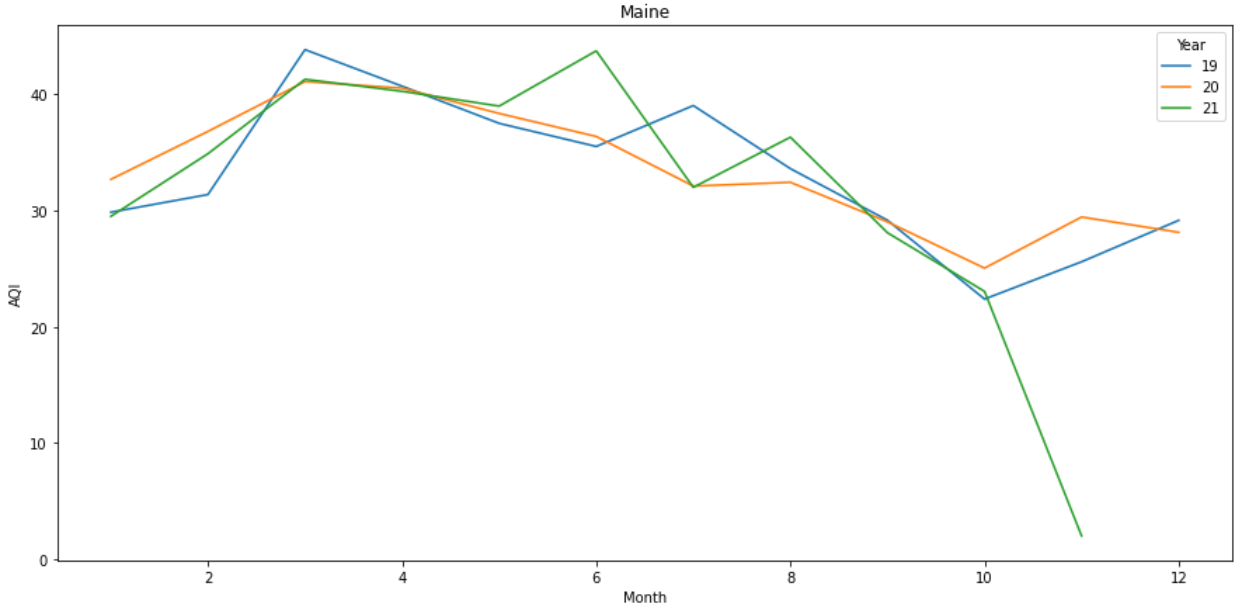

Indiana
AQI
Month
Year
19
20
21

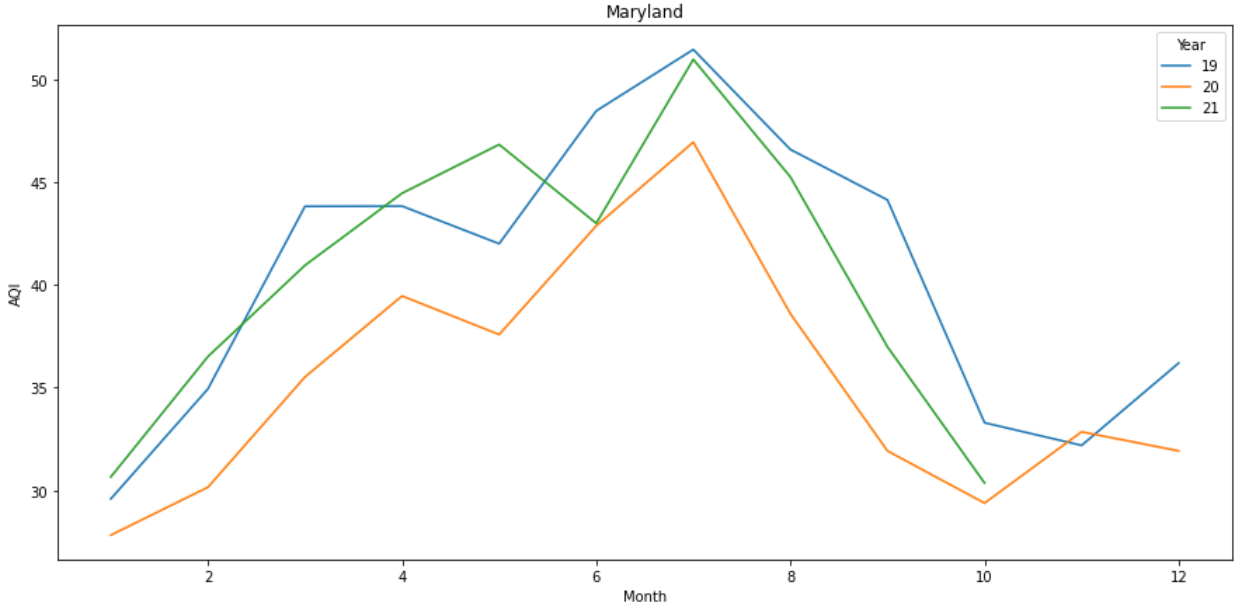

Iowa
AQI
Month
Year
19
20
21

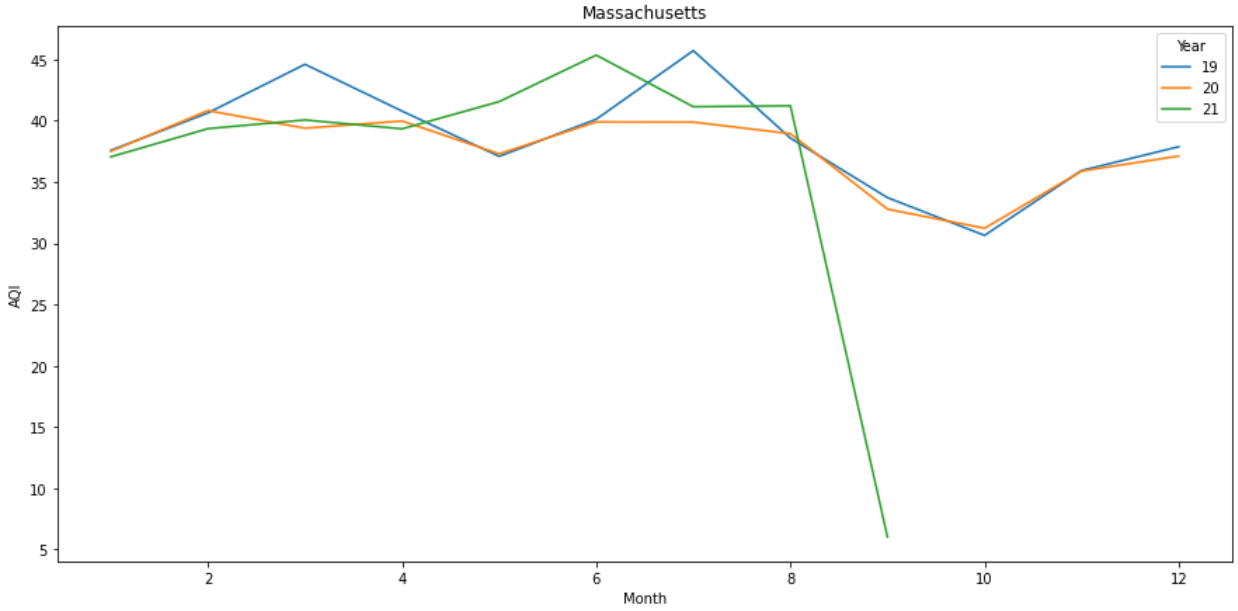

Kansas
Year
19
20
21
AQI
Month
0
10
20
30
40
2
4
6
8
10
12

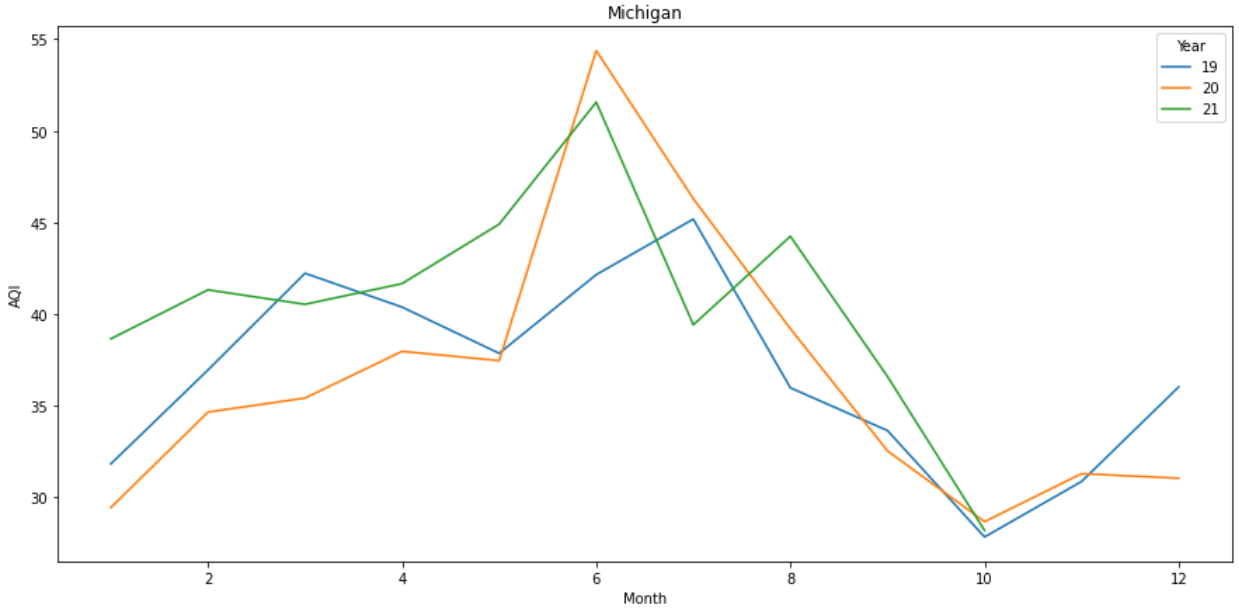

Kentucky
Year
19
20
21
AQI
Month
30.0
32.5
35.0
37.5
40.0
42.5
45.0
2
4
6
8
10
12

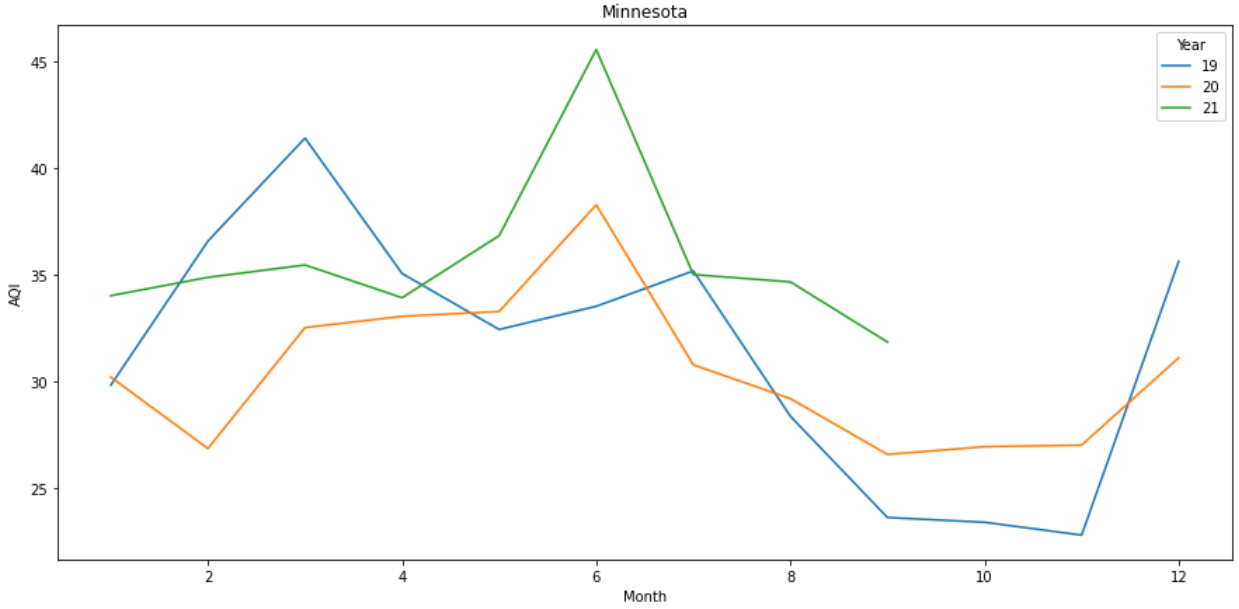

Louisiana
Year
19
20
21
AQI
Month
5
10
15
20
25
30
35
40
45
2
4
6
8
10
12

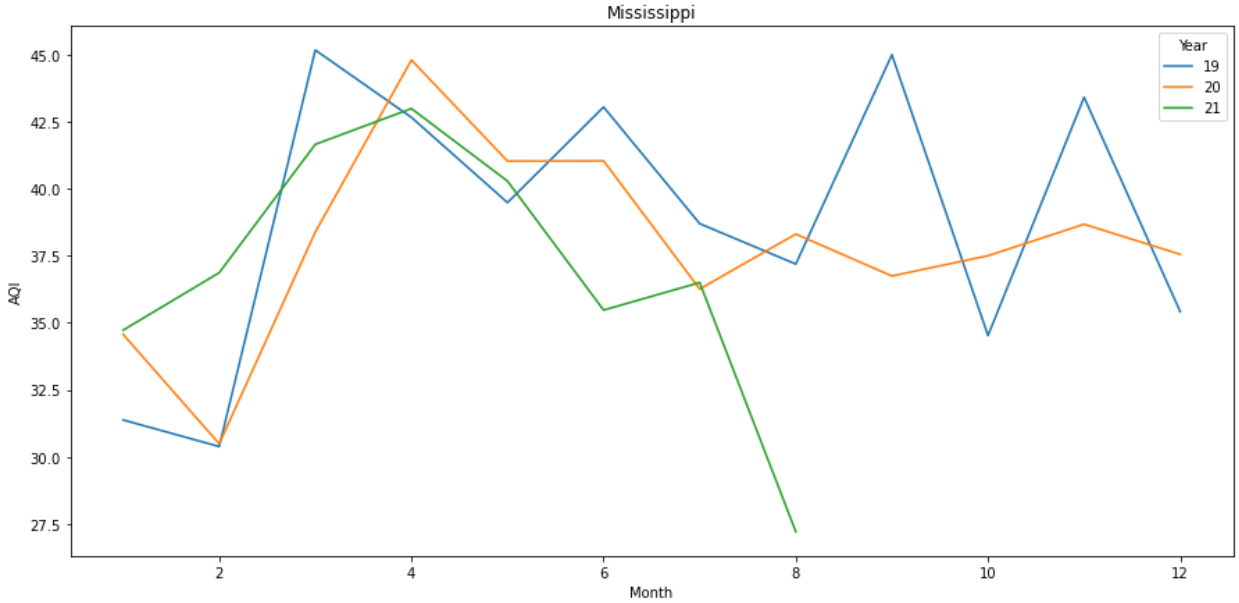

Maine
Year
19
20
21
AQI
Month
0
10
20
30
40
2
4
6
8
10
12

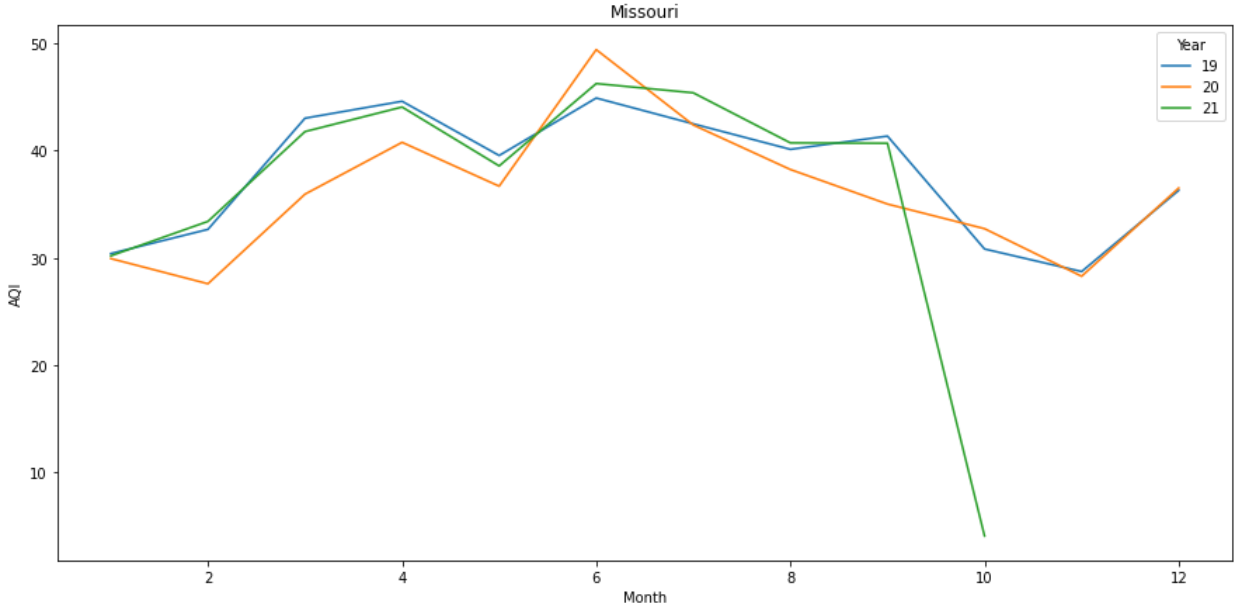

Maryland
Year
19
20
21
AQI
Month
30
35
40
45
50
2
4
6
8
10
12

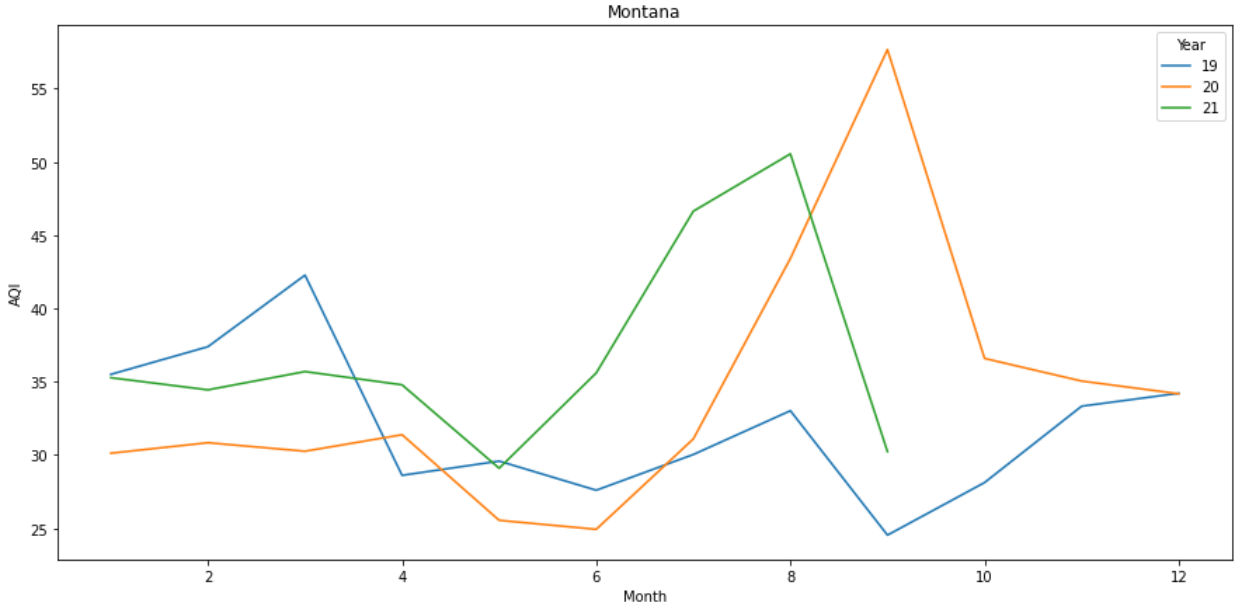

Massachusetts
Year
19
20
21
AQI
Month
5
10
15
20
25
30
35
40
45
2
4
6
8
10
12

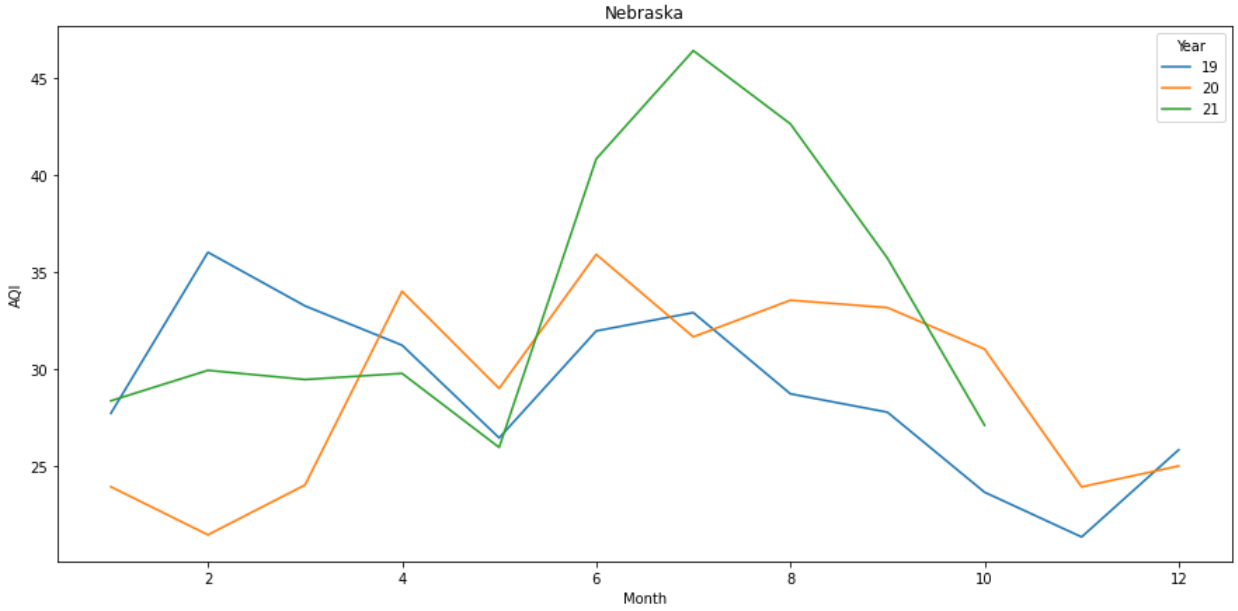

Michigan
Year
19
20
21
AQI
Month
55
50
45
40
35
30
2
4
6
8
10
12

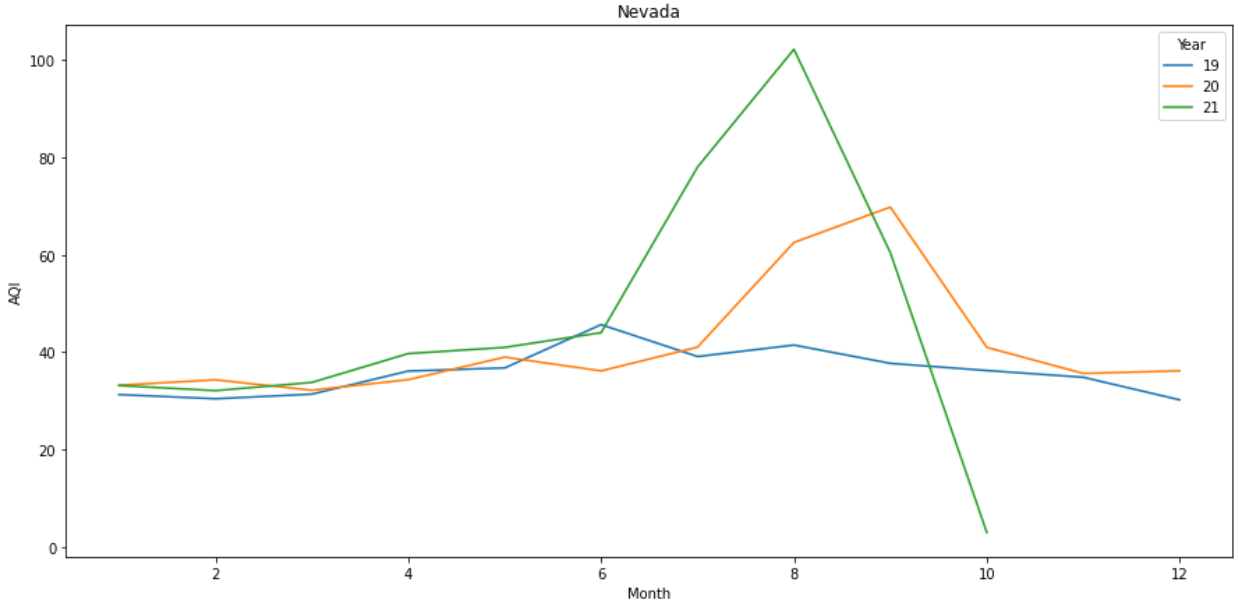

Minnesota
Year
19
20
21
AQI
Month
45
40
35
30
25
2
4
6
8
10
12

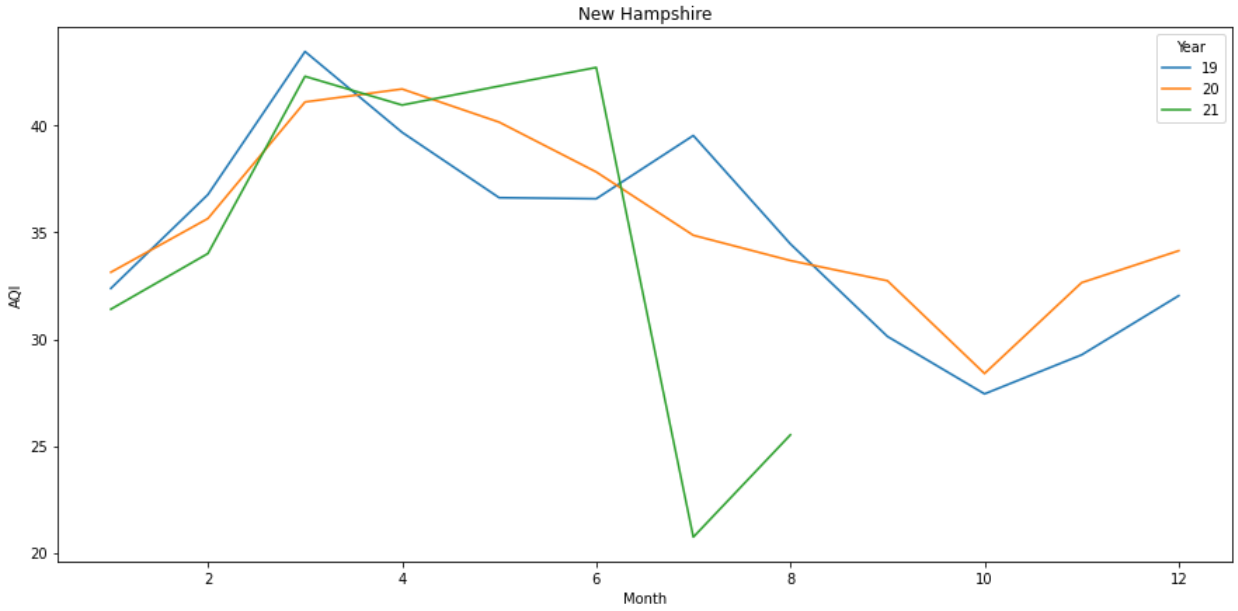

Mississippi
Year
19
20
21
AQI
Month
45.0
42.5
40.0
37.5
35.0
32.5
30.0
27.5
2
4
6
8
10
12

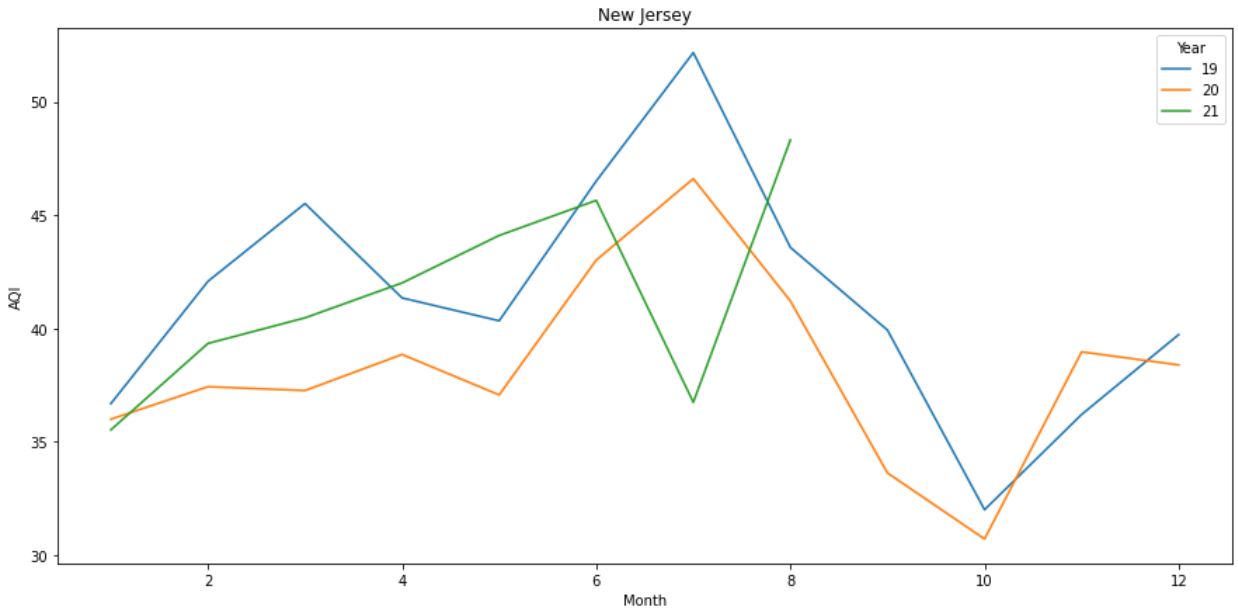

Missouri
Year
19
20
21
AQI
Month
50
40
30
20
10
2
4
6
8
10
12

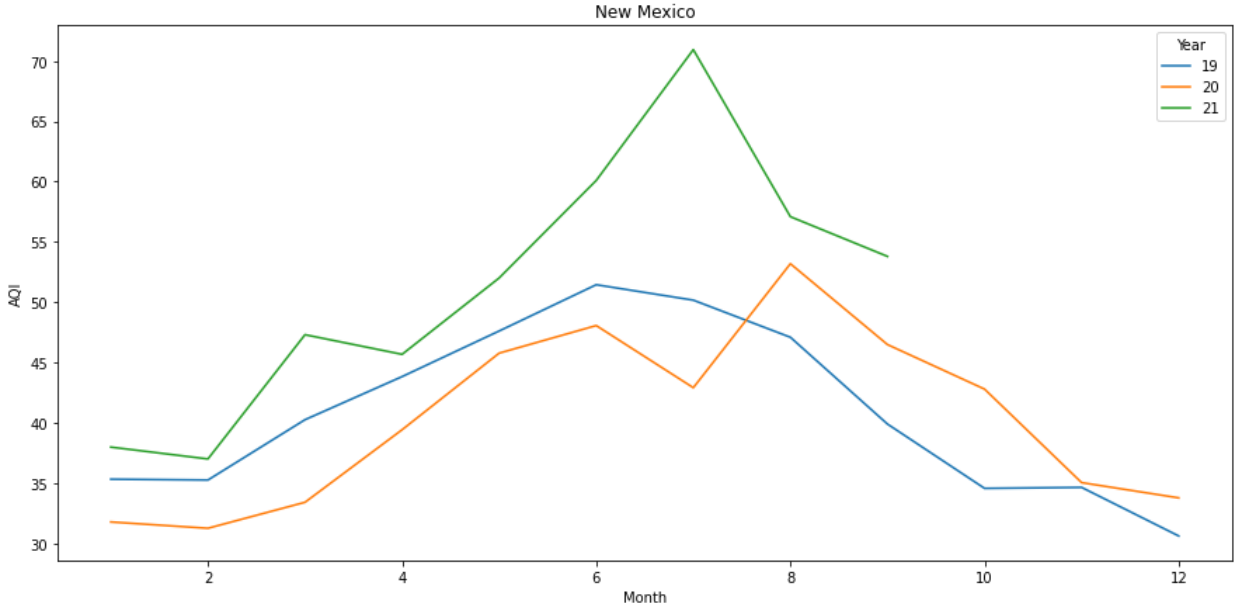

Montana
Year
19
20
21
AQI
Month
55
50
45
40
35
30
25
2
4
6
8
10
12

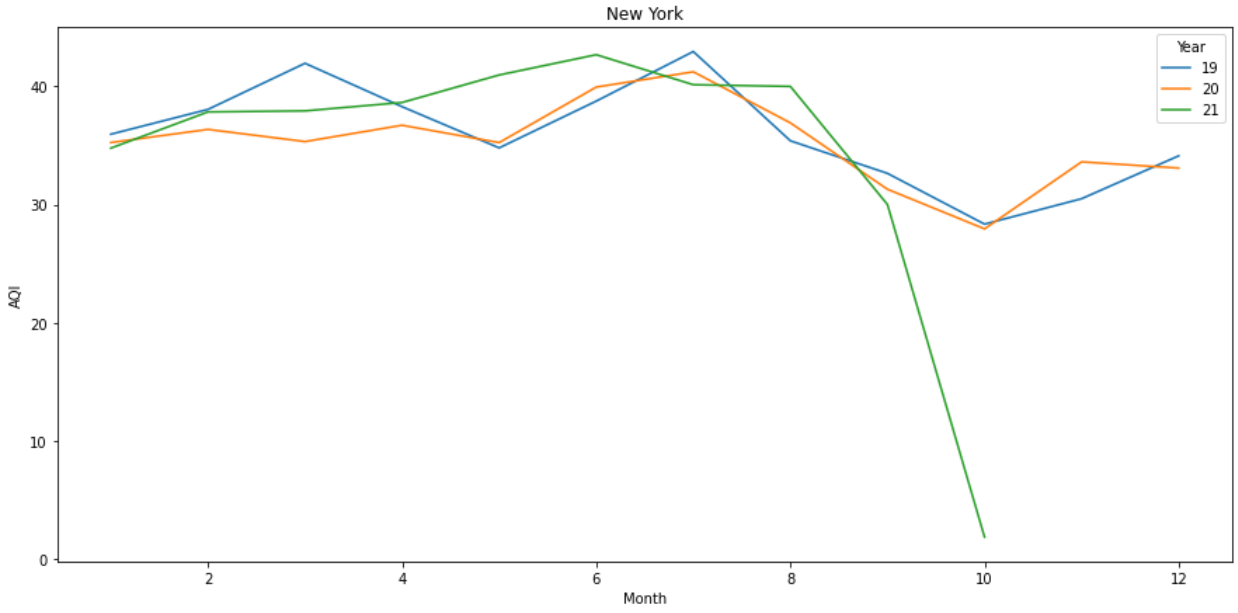

Nebraska
Year
19
20
21
AQI
Month
45
40
35
30
25
2
4
6
8
10
12

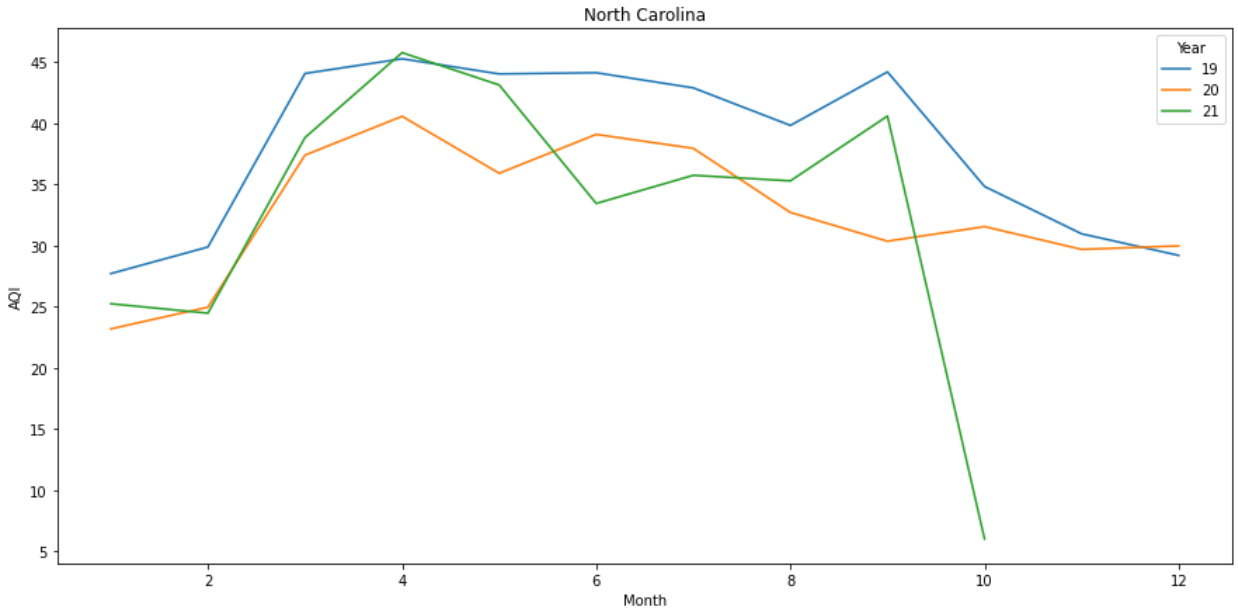

Nevada
Year
19
20
21
AQI
Month
0
20
40
60
80
100
2
4
6
8
10
12

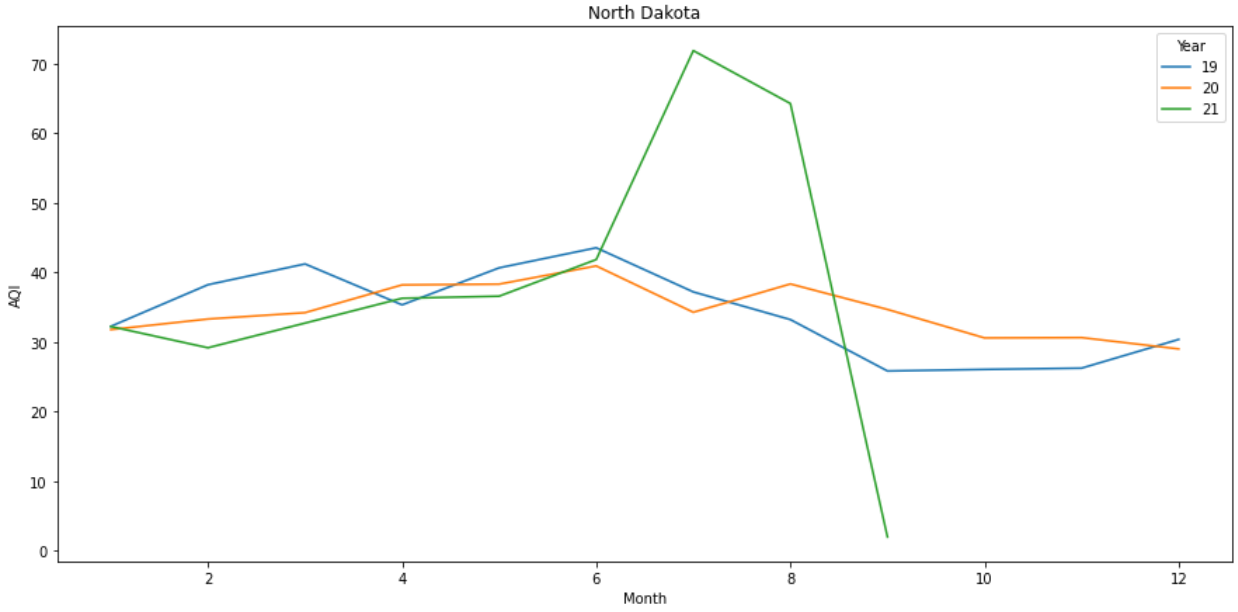

New Hampshire
Year
19
20
21
AQI
Month
20
25
30
35
40
2
4
6
8
10
12

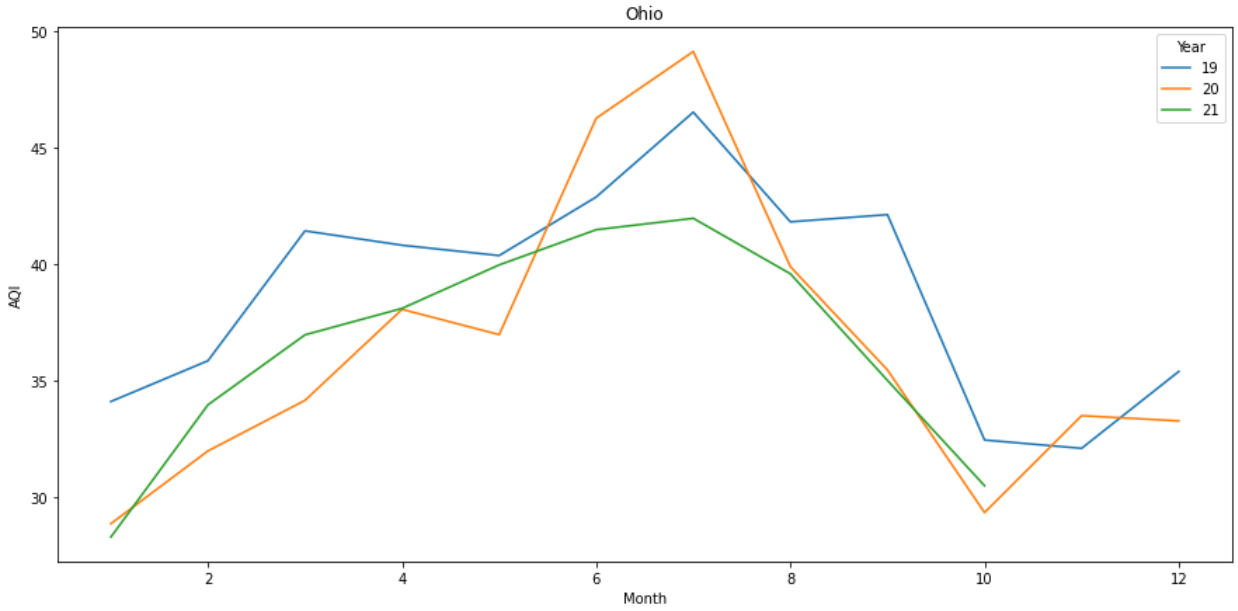

New Jersey
Year
19
20
21
AQI
Month
30
35
40
45
50
2
4
6
8
10
12

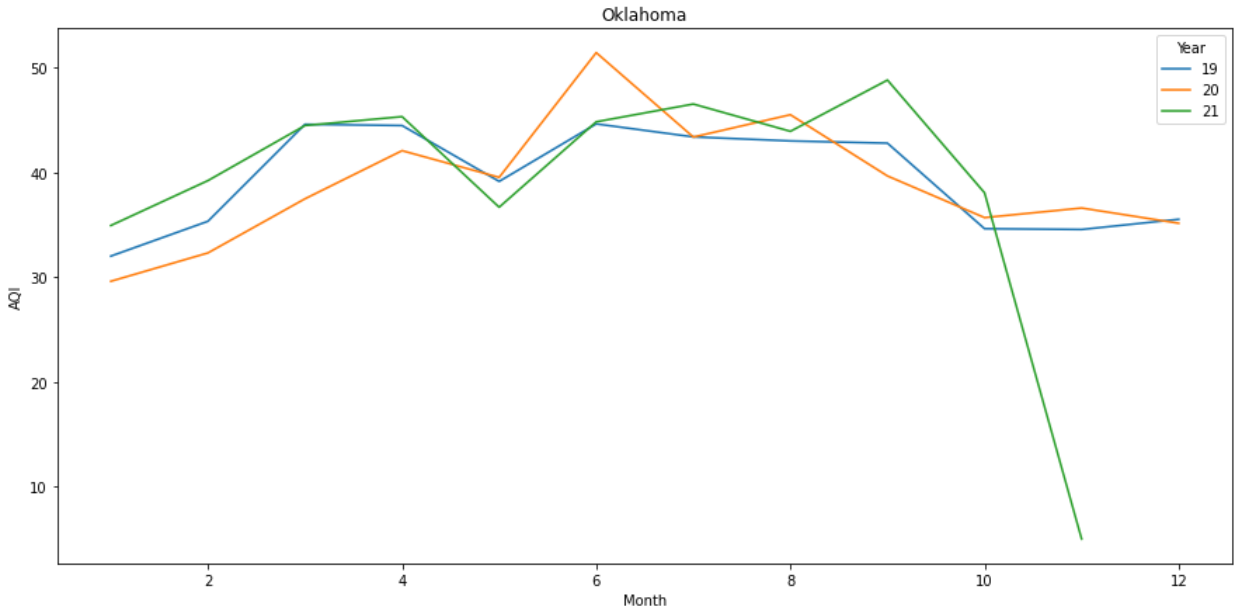

New Mexico
Year
19
20
21
AQI
Month
70
65
60
55
50
45
40
35
30
2
4
6
8
10
12

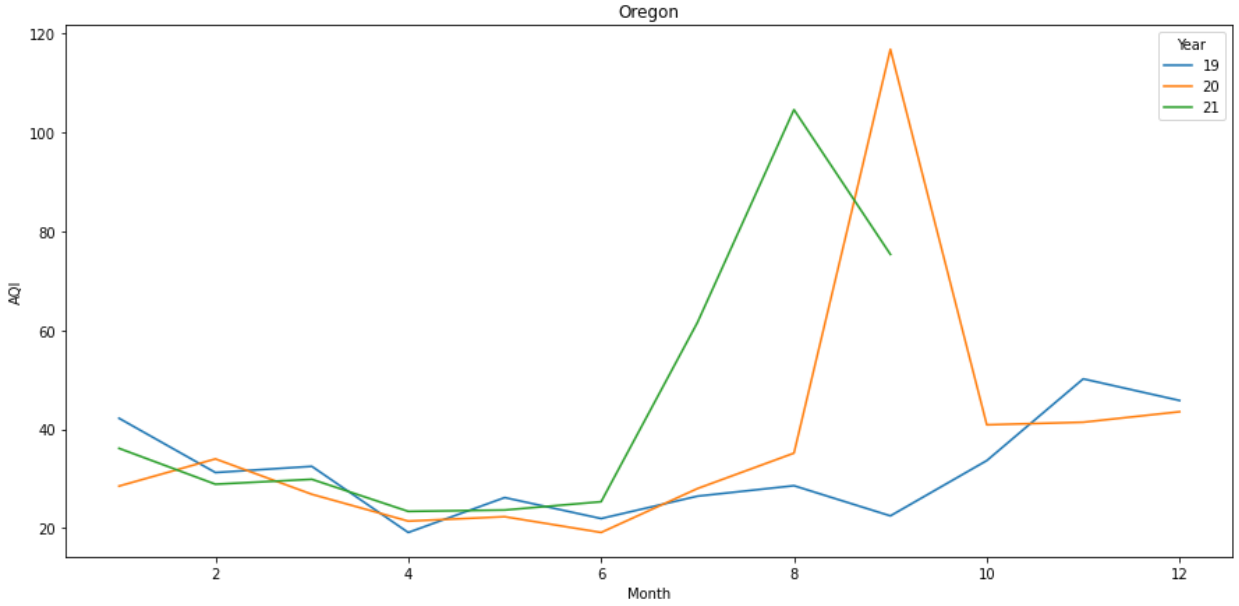

New York
Year
19
20
21
AQI
Month
40
30
20
10
0
2
4
6
8
10
12

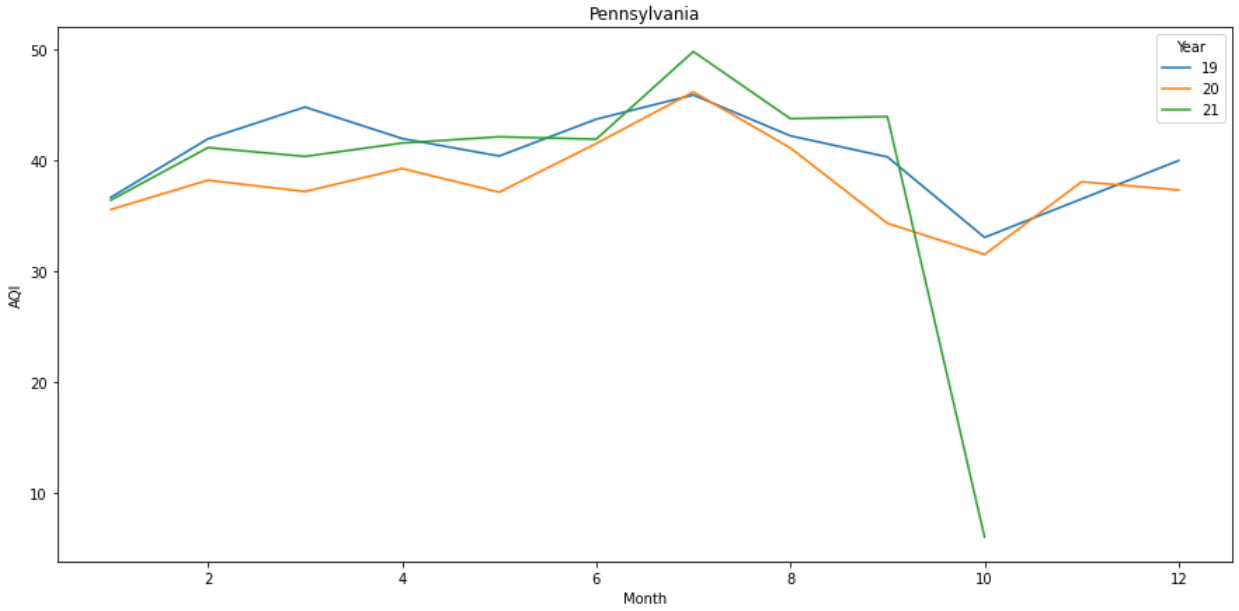

North Carolina
Year
19
20
21
AQI
Month
45
40
35
30
25
20
15
10
5
2
4
6
8
10
12

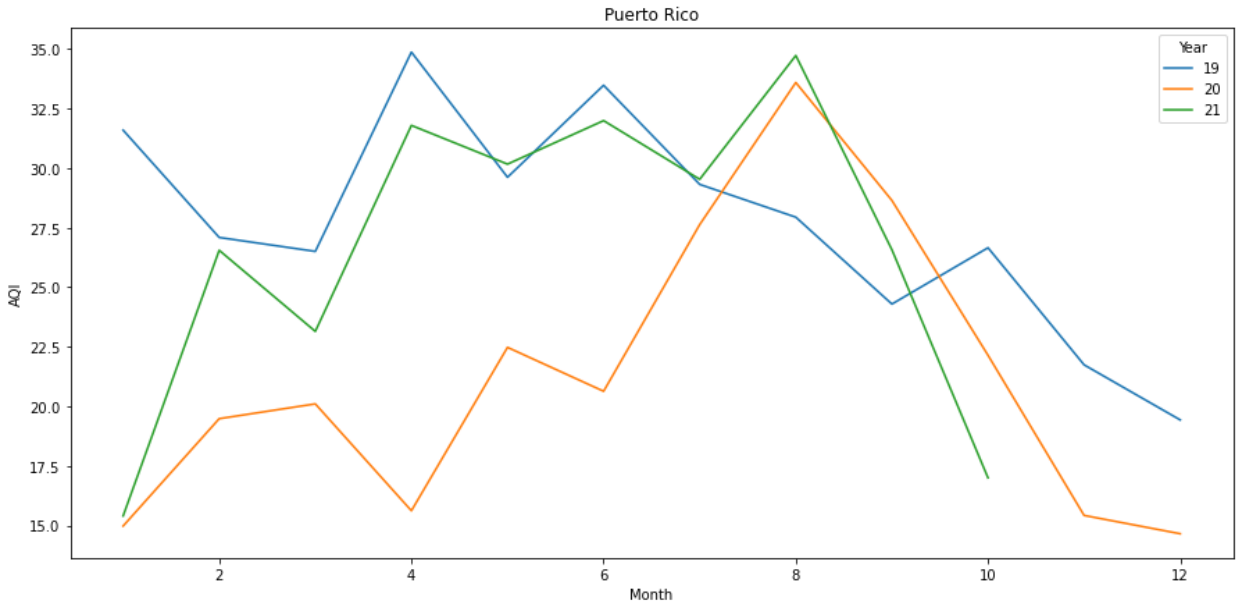

North Dakota
Year
19
20
21
AQI
Month
0
10
20
30
40
50
60
70
2
4
6
8
10
12

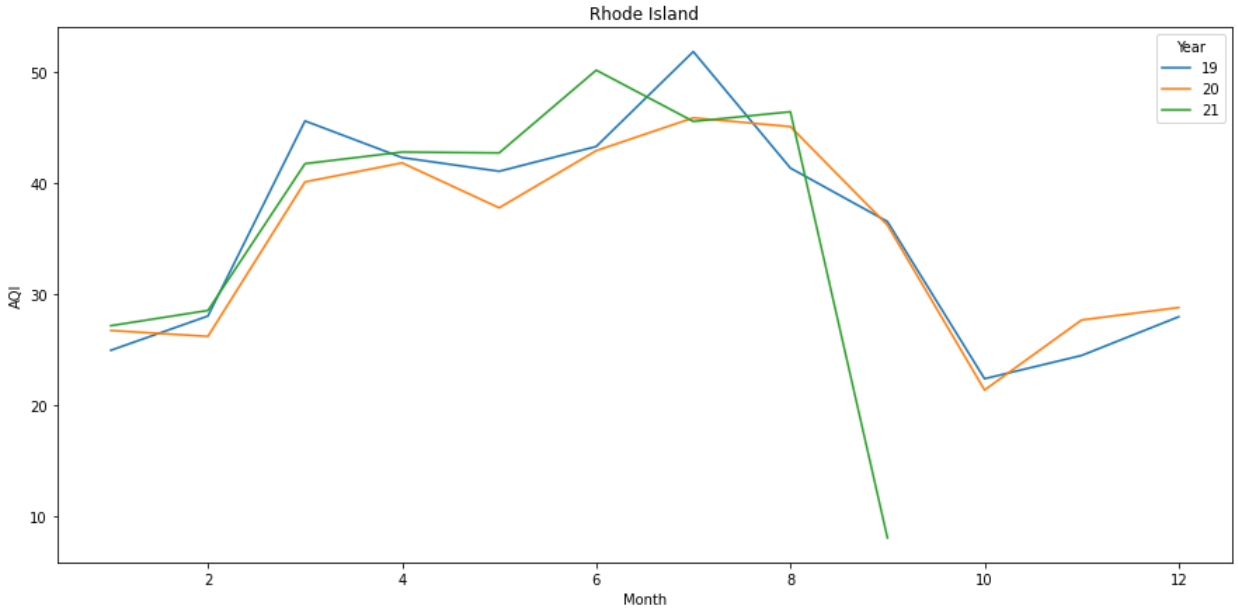

Ohio
Year
19
20
21
AQI
Month
30
35
40
45
50
2
4
6
8
10
12

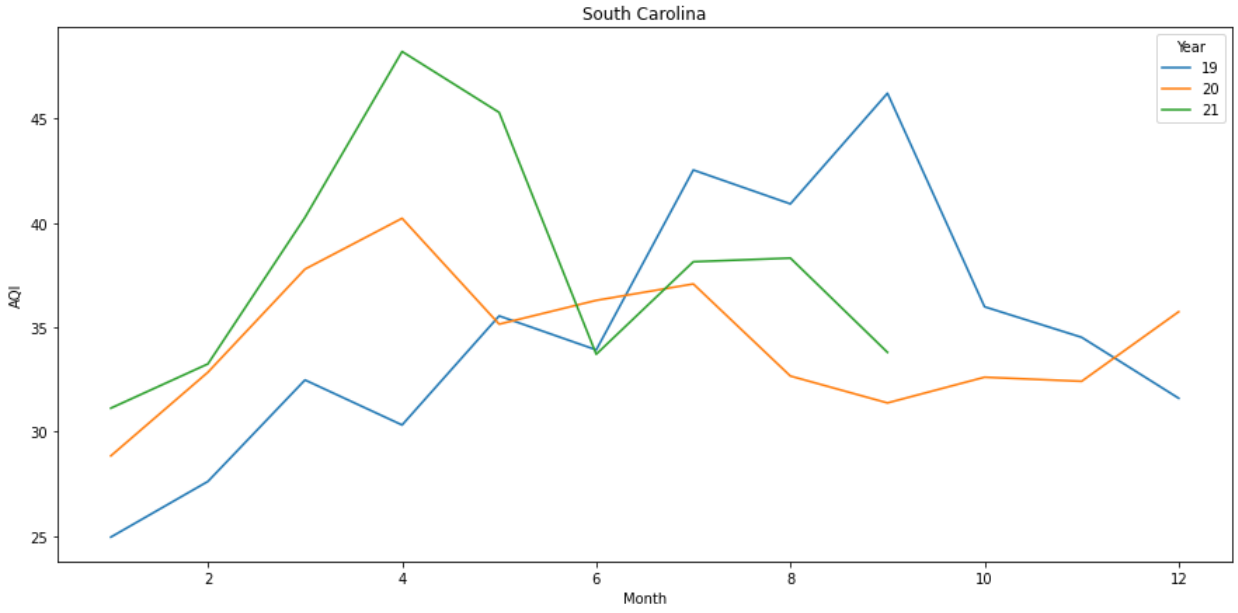

Oklahoma
Year
19
20
21
AQI
Month
10
20
30
40
50
2
4
6
8
10
12

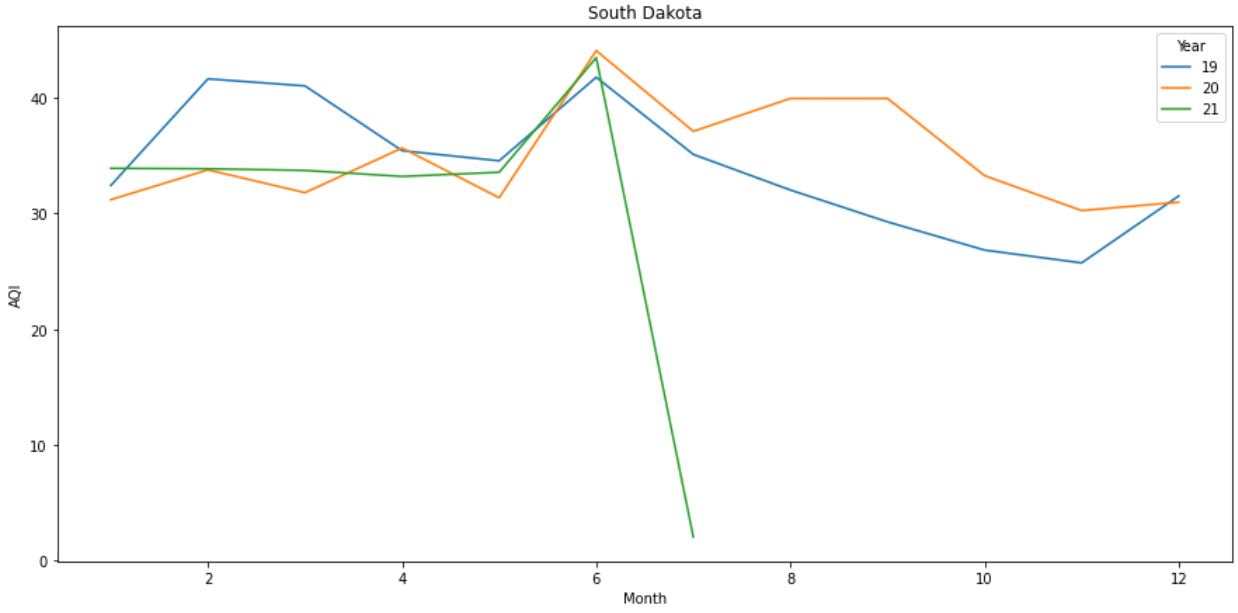

Oregon
AQI
Month
Year
19
20
21

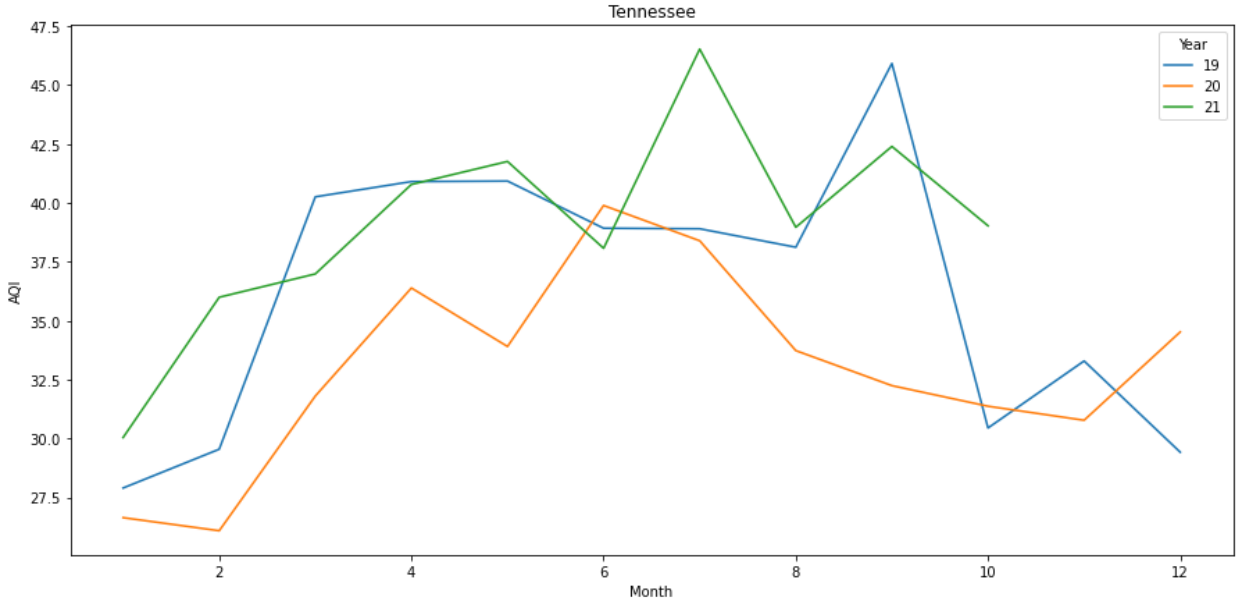

Pennsylvania
AQI
Month
Year
19
20
21

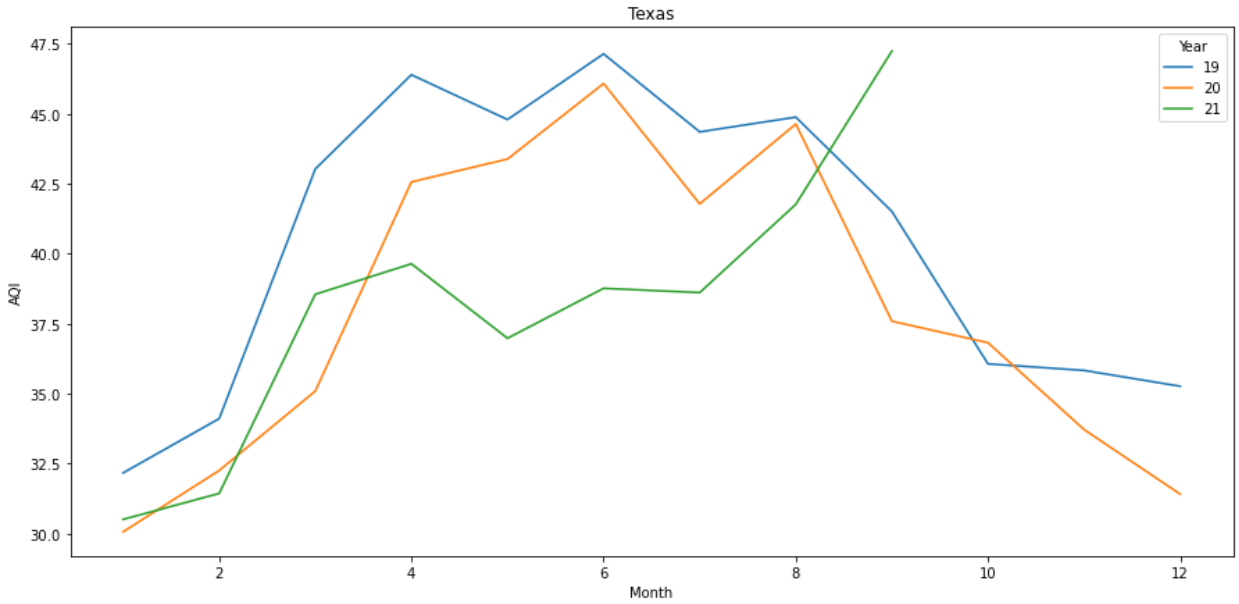

Puerto Rico
AQI
Month
Year
19
20
21

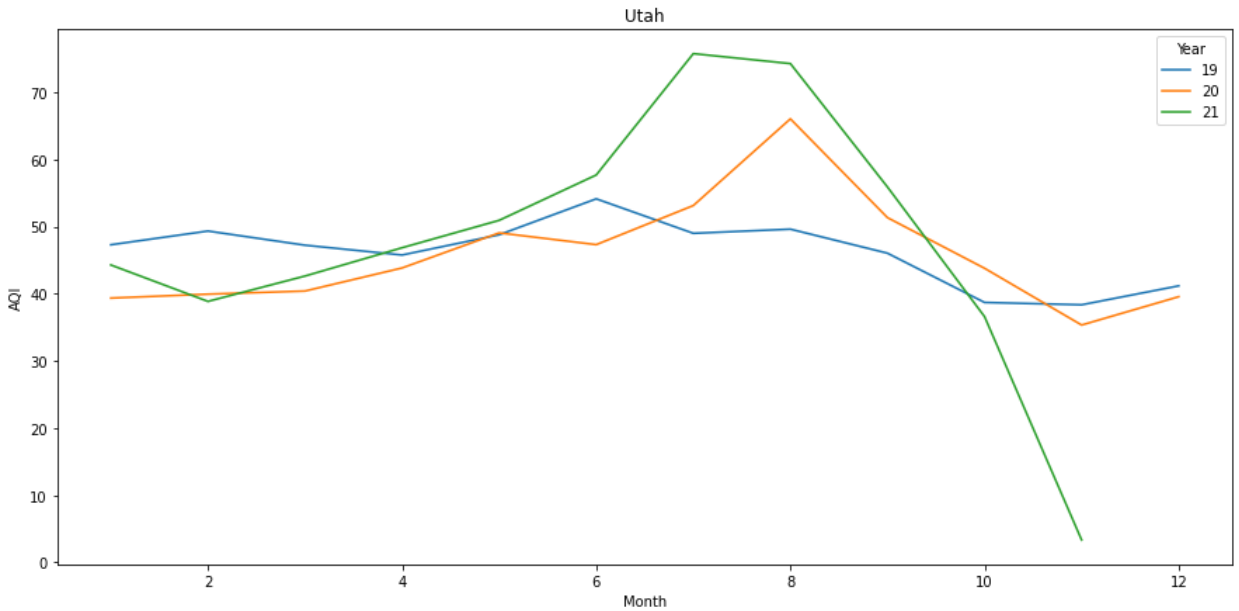

Rhode Island
Year
19
20
21
AQI
Month
10
20
30
40
50
2
4
6
8
10
12

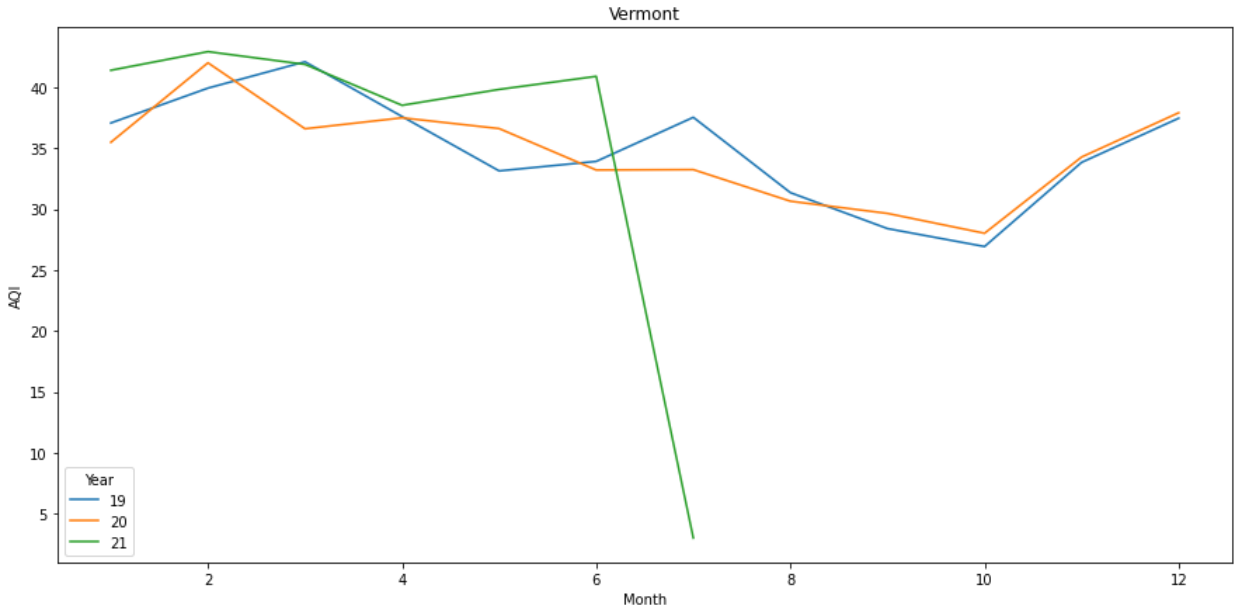

South Carolina
Year
19
20
21
AQI
Month
25
30
35
40
45
2
4
6
8
10
12

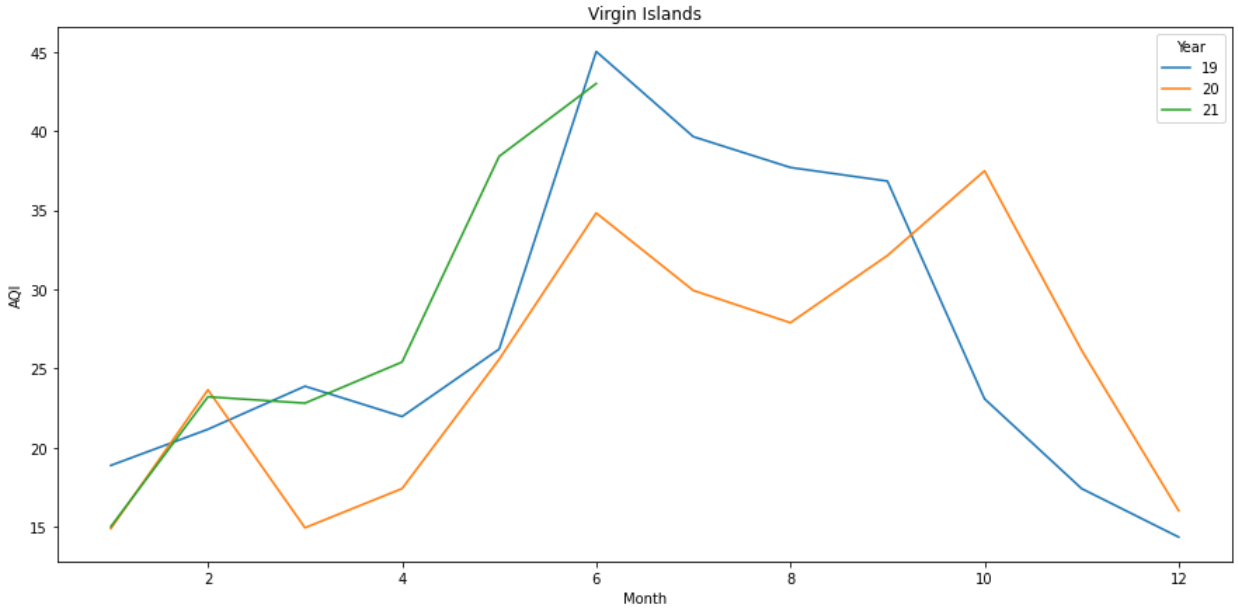

South Dakota
Year
19
20
21
AQI
Month
0
10
20
30
40
2
4
6
8
10
12

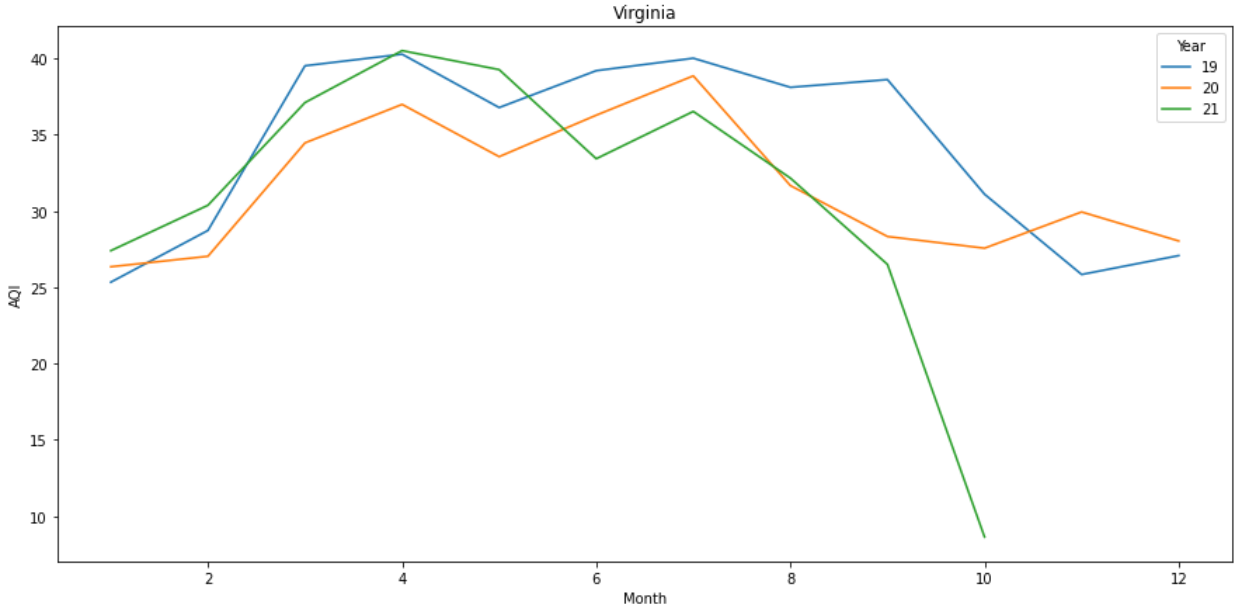

Tennessee
AQI
Month
Year
19
20
21

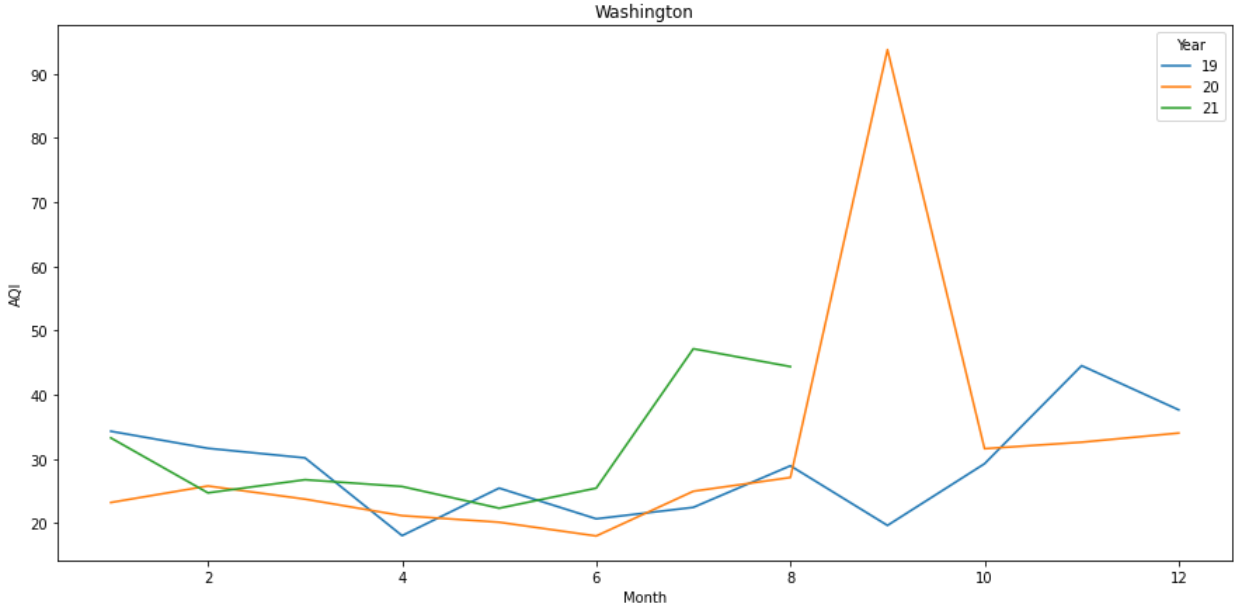

Texas
AQI
Month
Year
19
20
21

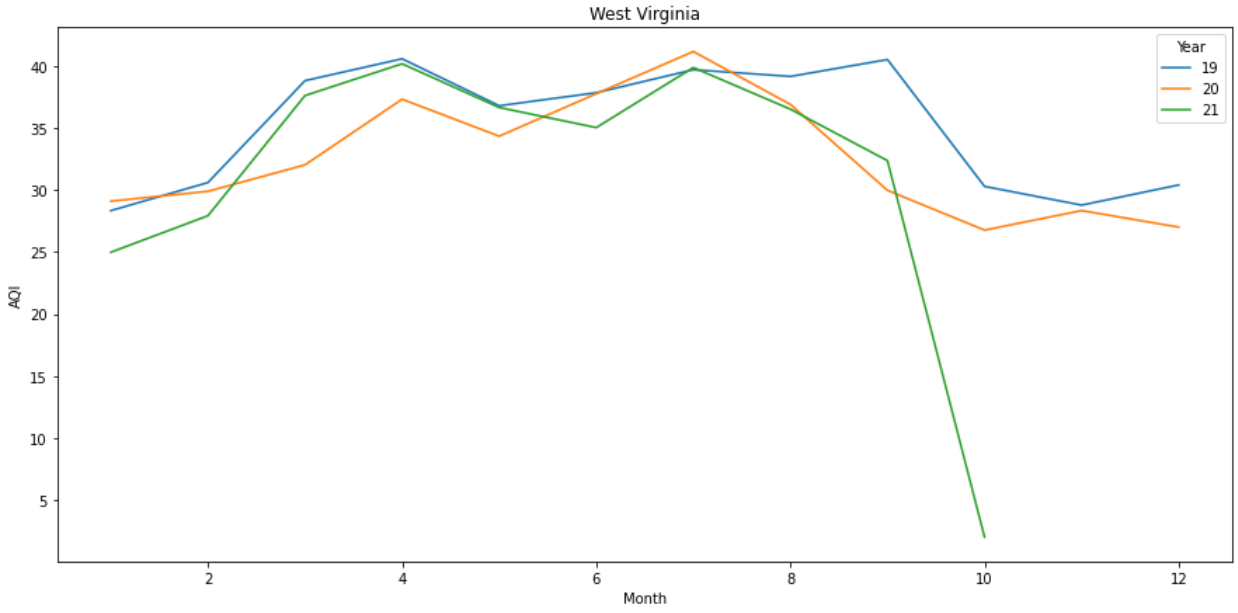

Utah
AQI
Month
Year
19
20
21

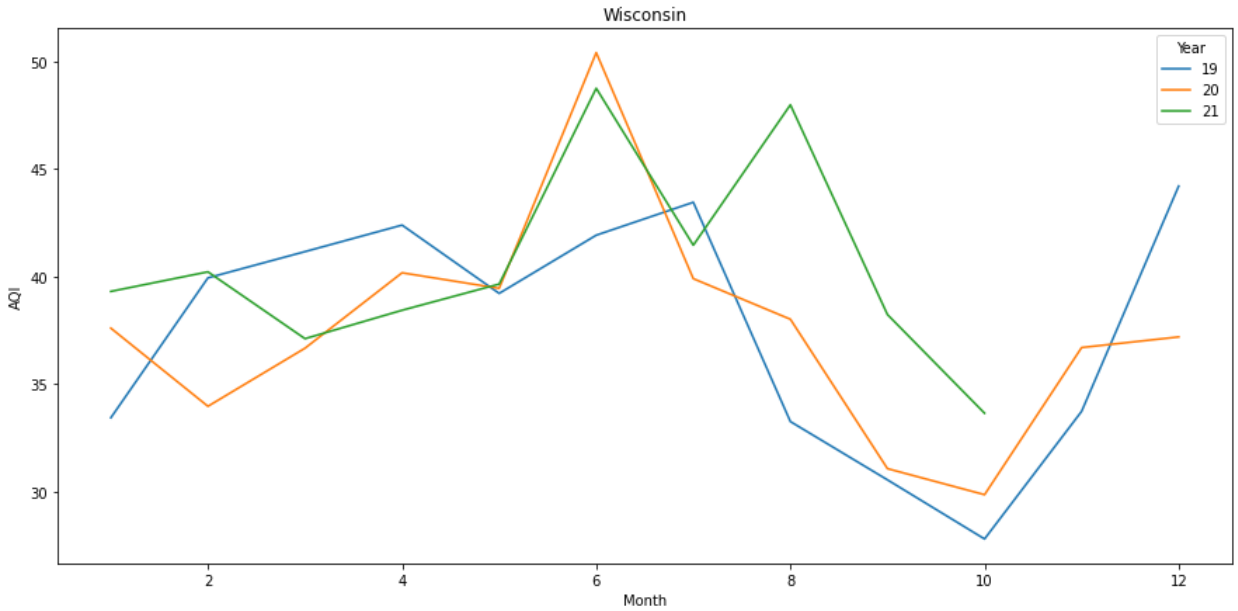

Vermont
AQI
Month
Year
19
20
21

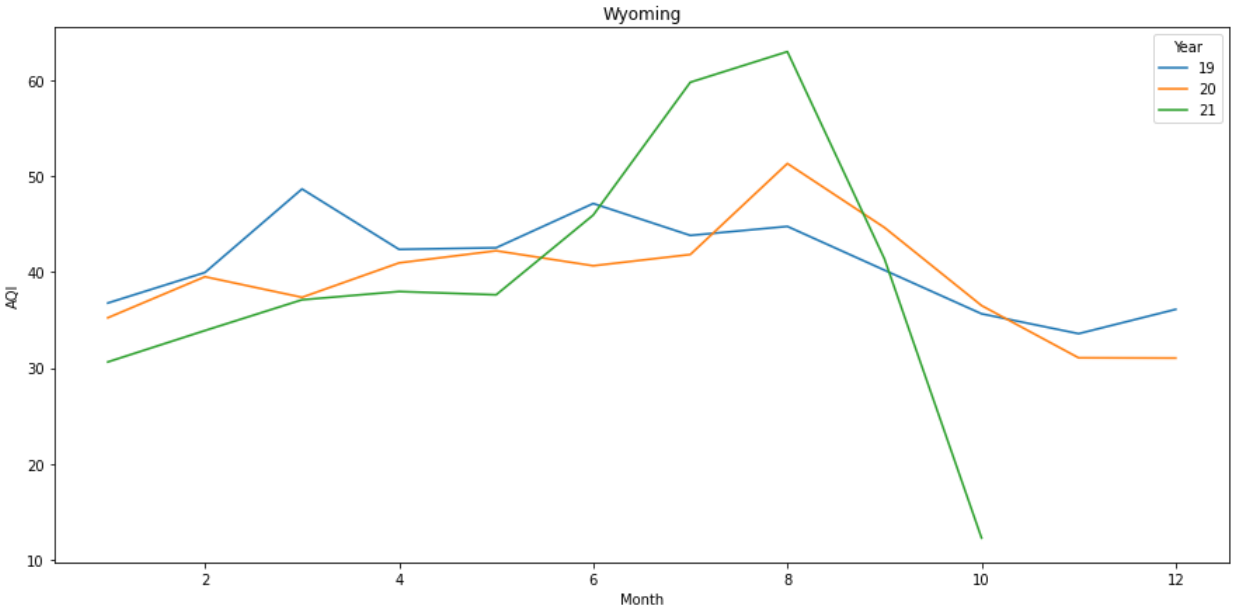

Virgin Islands
AQI
Month
Year
19
20
21


Virginia
AQI
Month
Year
19
20
21


Washington
Year
19
20
21
AQI
Month
90
80
70
60
50
40
30
20
2
4
6
8
10
12


West Virginia
Year
19
20
21
AQI
Month
40
35
30
25
20
15
10
5
2
4
6
8
10
12


Wisconsin
Year
19
20
21
AQI
Month
50
45
40
35
30
2
4
6
8
10
12



Did the lockdown result in a noticable improvement in air quality? Most notably in what regions?

Ans:

After analyzing the AQI data for all the states of the United States, it was found that the states: {Alaska}, {Mexico}, {District of Columbia}, {Hawaii}, {Idaho}, {Illinois}, {Indiana, {Iowa}, {Michigan}, {Minnesota}, {Montana}, {Nebraska}, {Puerto Rico}, {Tennessee} had low AQI levels during the lockdown period of February-2020 - April-2020. However, after the lockdown period, there was a sudden increase in AQI levels in all the listed states. This suggests that the lockdown measures implemented in the state had a positive impact on reducing air pollution during the lockdown period, but that the levels returned to normal or increased after the lockdown period.

{Finall Summary}

Metro States: {Alabama, Arizona, Arkansas, California, Colorado, Connecticut, Delaware, Florida, Georgia, Kansas, Kentucky, Louisiana, Maine, Maryland, Massachusetts, Mississippi, Missouri, Nevada, New Hampshire, New Jersey, New Mexico, New York, North Carolina, North Dakota, Ohio, Oklahoma, Oregon, Pennsylvania, Rhode Island, South Carolina, Texas, Utah, Vermont, Virginia, Washington, Wyoming, West Virginia}

The AQI levels in metropolitan states and big county were found to be relatively same as well as mi-nute low compared to previous years, but not as low as in other states during the lockdown period. This is likely due to the high population density and number of industries present in these metropolitan areas, which contribute to ongoing air pollution even during a lockdown period. Despite this, it is still notable that the AQI levels were lower than in previous years, indicating that some measures to reduce air pollution may have been

successful in these cities. Overall, while more work needs to be done to further reduce air pollution in metropolitan areas, the data suggests that progress is being made.