aqi_visualization_with_plotly (2)

November 29, 2023

1 Introduction:

Explore the global Air Quality Index (AQI) dataset, providing a snapshot of daily air quality in various countries. Uncover insights, tackle seasonality, and delve into geospatial patterns. The AQI scale guides us through potential health impacts, while the 'data_date.csv' file facilitates chronological exploration.

2 Some questions we want to answer

What is the air quality status and AQI values of countries worldwide?

- Explore global AQI values and categorize air quality based on the AQI scale.

Which countries had the highest AQI values on the day before the analysis?

- Identify the top 10 countries with the most elevated AQI values.

Which countries had the lowest AQI values on the day before the analysis?

- Identify the top 10 countries with the lowest AQI values.

What are the AQI values and air quality statuses for major countries?

- Provide an overview of AQI values and air quality for significant nations.

How does air quality vary over time?

- Conduct exploratory data analysis (EDA) with datetime information to identify temporal trends and seasonality.

Can you visualize global air quality on a world map?

Create a world map highlighting countries based on their AQI values and air quality statuses.

3 Importing libraries

[92]:

```
import seaborn as sns
import matplotlib.pyplot as plt
import plotly.express as px
from urllib.request import urlopen
import numpy as np # linear algebra
import pandas as pd # data processing,
import os
px_template = "simple_white"
import json
import time
from datetime import datetime as dt
```

[94]:
```
# from google.colab import drive
# drive.mount('/content/drive')
```

4 Data preprocessing

[152]:
```
df = pd.read_csv("world-ari-quality.csv" )
df.head()
```

[152]:
```
         Date    Country                          Status  AQI Value
0  2022-07-21    Albania                            Good         14
1  2022-07-21    Algeria                        Moderate         65
2  2022-07-21    Andorra                        Moderate         55
3  2022-07-21     Angola  Unhealthy for Sensitive Groups        113
4  2022-07-21  Argentina                        Moderate         63
```

checking shape or size of a dataset

[172]:
```
df.shape
```

[172]:
```
(12986, 4)
```

Column name of a dataset

[103]:
```
df.columns
```

[103]:
```
Index(['Date', 'Country', 'Status', 'AQI Value'], dtype='object')
```

checking unique value number in variables

[106]:
```
df.nunique()
```

[106]:
```
Date          85
Country      142
Status         6
AQI Value    300
dtype: int64
```

describing numerical column of this dataset

[109]:
```
df.describe()
```

```
[109]:           AQI Value
        count  12986.000000
        mean      61.940936
        std       48.481170
        min        1.000000
        25%       29.000000
        50%       53.000000
        75%       83.000000
        max      963.000000
```

information about this dataset

```
[112]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12986 entries, 0 to 12985
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       12986 non-null  object
 1   Country    12986 non-null  object
 2   Status     12986 non-null  object
 3   AQI Value  12986 non-null  int64
dtypes: int64(1), object(3)
memory usage: 405.9+ KB
```

checking missing or null value in this dataset

```
[115]: df.isna().sum()
```

```
[115]: Date         0
       Country      0
       Status       0
       AQI Value    0
       dtype: int64
```

checking unique values of column country

```
[118]: df.Country.unique()
```

```
[118]: array(['Albania', 'Algeria', 'Andorra', 'Angola', 'Argentina', 'Armenia',
              'Australia', 'Austria', 'Azerbaijan', 'Bahrain', 'Bangladesh',
              'Belarus', 'Belgium', 'Belize', 'Bermuda', 'Bolivia',
              'Bosnia and Herzegovina', 'Brazil', 'Brunei', 'Bulgaria',
              'Burkina Faso', 'Cambodia', 'Canada', 'Cape Verde',
              'Cayman Islands', 'Central African Republic', 'Chad', 'Chile',
              'China', 'Colombia', 'Costa Rica', 'Croatia', 'Cyprus',
              'Czech Republic', 'Denmark', 'Dominican Republic', 'Ecuador',
              'Egypt', 'El Salvador', 'Estonia', 'Ethiopia', 'Finland', 'France',
              'French Guiana', 'Gabon', 'Gambia', 'Georgia', 'Germany', 'Ghana',
              'Gibraltar', 'Greece', 'Grenada', 'Guadeloupe', 'Guam',
              'Guatemala', 'Honduras', 'Hong Kong', 'Hungary', 'Iceland',
              'India', 'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Israel', 'Italy',
              'Ivory Coast', 'Japan', 'Jersey', 'Jordan', 'Kazakhstan', 'Kenya',
              'Kosovo', 'Kuwait', 'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon',
              'Liberia', 'Liechtenstein', 'Lithuania', 'Luxembourg', 'Macao',
              'Macedonia', 'Madagascar', 'Malaysia', 'Malta', 'Martinique',
              'Mexico', 'Moldova', 'Monaco', 'Mongolia', 'Montenegro', 'Myanmar',
              'Nepal', 'Netherlands', 'New Caledonia', 'New Zealand', 'Nigeria',
              'Norway', 'Pakistan', 'Palestinian Territory', 'Peru',
              'Philippines', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',
              'Reunion', 'Romania', 'Russia', 'San Marino', 'Saudi Arabia',
              'Senegal', 'Serbia', 'Singapore', 'Slovakia', 'Slovenia',
              'South Africa', 'South Korea', 'Spain', 'Sri Lanka', 'Sudan',
              'Sweden', 'Switzerland', 'Taiwan', 'Tajikistan', 'Thailand',
              'Togo', 'Trinidad and Tobago', 'Turkey', 'Turkmenistan', 'Uganda',
              'Ukraine', 'United Arab Emirates',
              'United Kingdom of Great Britain and Northern Ireland',
              'United States of America', 'Uzbekistan', 'Vatican', 'Venezuela',
              'Vietnam', 'Zambia'], dtype=object)
```

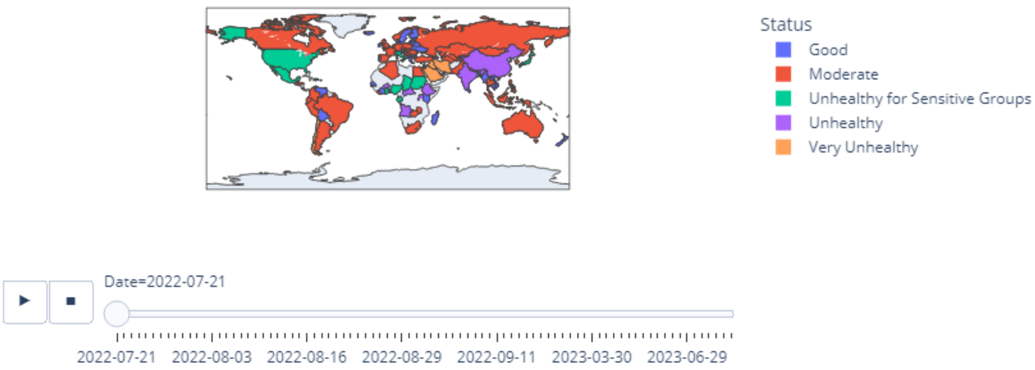
changing name of 2 countries

[121]:
```
df["Country"]= df["Country"].replace({'United Kingdom of Great Britain and␣
 ↪Northern Ireland':"Great Britain",'United States of America':"USA"})
```

5 Data Analysis and Visualization

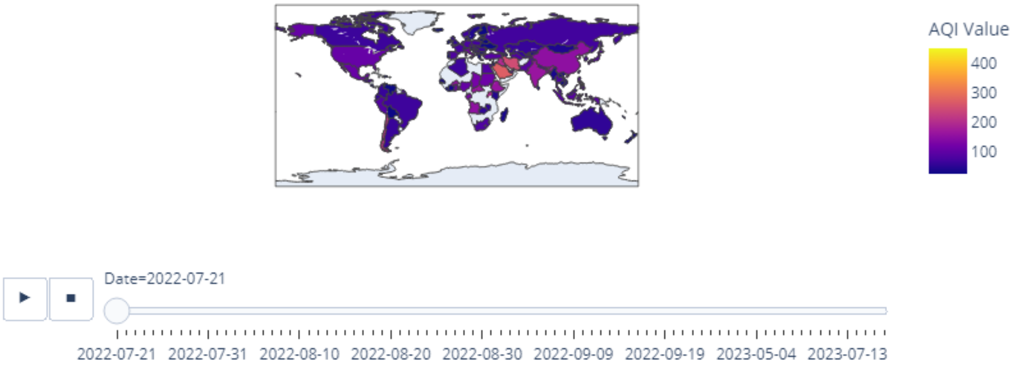
date wise AQI status of countries on world map

[136]:
```
fig = px.choropleth(df,locations = "Country", locationmode='country names',␣
 ↪color="Status", animation_frame="Date",range_color= [25,450])
fig.show()
```



date wise AQI value of countries on world map

[212]:
```
fig = px.choropleth(df,locations = "Country", locationmode='country names',␣
  ↪color="AQI Value", animation_frame="Date",range_color= [25,450])
fig.show()
```



[ ]:
```

```

6 what are Highest and Lowest AQI value of Countries ?

setting Date column on index

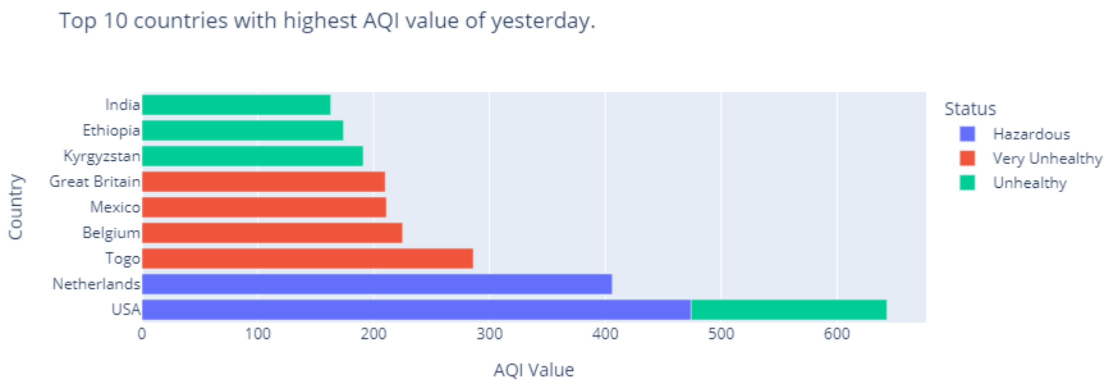
[148]:
```
df = df.set_index("Date")
```

Top 10 countries with highest AQI value of yesterday.

[42]:
```
a = df.iloc[-141: ,:].sort_values(ascending = False ,by = "AQI Value")[:10]
a
```

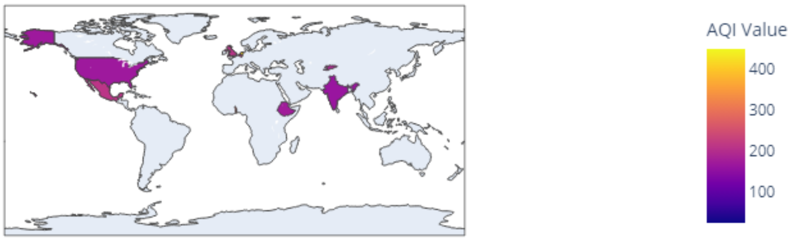
[42]:
```
                    Country          Status  AQI Value
Date
2023-08-10              USA       Hazardous        474
2023-08-10      Netherlands       Hazardous        406
2023-08-10             Togo  Very Unhealthy        286
2023-08-10          Belgium  Very Unhealthy        225
2023-08-10           Mexico  Very Unhealthy        211
2023-08-10    Great Britain  Very Unhealthy        210
2023-08-10       Kyrgyzstan       Unhealthy        191
2023-08-10         Ethiopia       Unhealthy        174
2023-08-10              USA       Unhealthy        169
2023-08-10            India       Unhealthy        163
```

[46]:
```
px.bar(a , x = "AQI Value", y = "Country" , color = "Status" , title = "Top 10␣
 ↪countries with highest AQI value of yesterday.")
```



[47]:
```
fig = px.choropleth(a,locations = "Country", locationmode='country names',
                    color="AQI Value",hover_name= "Status",range_color= [25,450]
                    ,title = "Top 10 countries with highest AQI value of␣
 ↪yesterday.")
fig.show()
```

Top 10 countries with highest AQI value of yesterday.

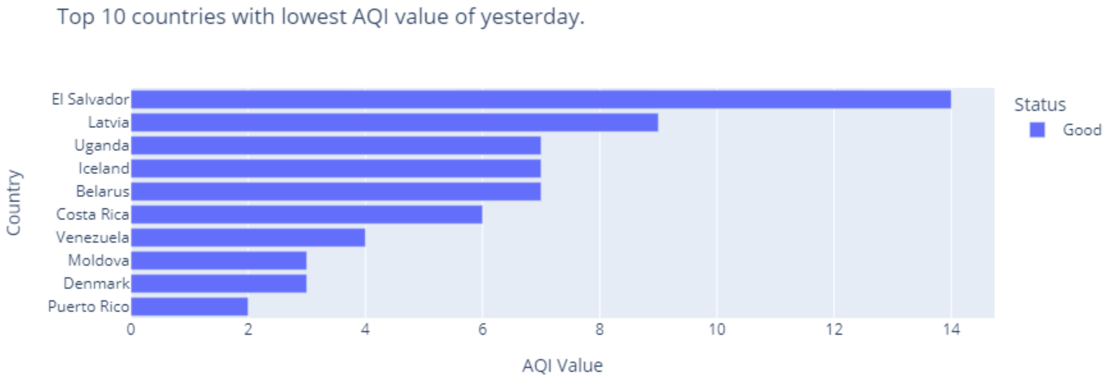


Top 10 countries with lowest AQI value of yesterday.

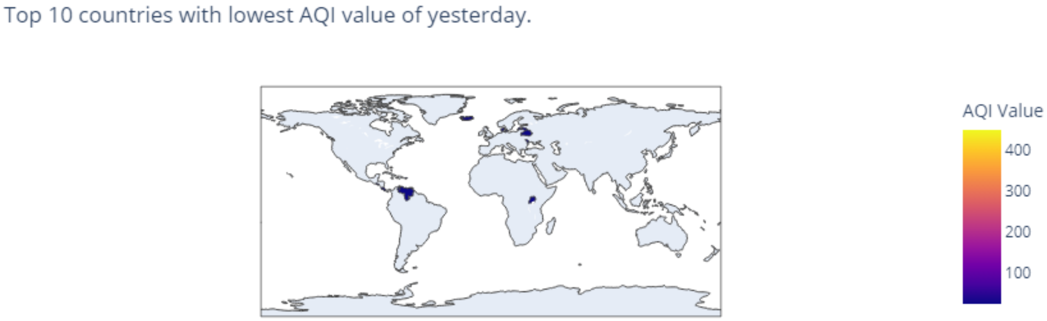
```
[49]: b = df.iloc[-141: ,:].sort_values(ascending = True ,by = "AQI Value")[:10]
      b
```

```
[49]:                 Country Status  AQI Value
      Date
      2023-08-10  Puerto Rico   Good          2
      2023-08-10      Denmark   Good          3
      2023-08-10      Moldova   Good          3
      2023-08-10    Venezuela   Good          4
      2023-08-10   Costa Rica   Good          6
      2023-08-10      Belarus   Good          7
      2023-08-10      Iceland   Good          7
      2023-08-10       Uganda   Good          7
      2023-08-10       Latvia   Good          9
      2023-08-10  El Salvador   Good         14
```

```
[53]: px.bar(b , x = "AQI Value", y = "Country" , color = "Status" , title = "Top 10␣
        ↪countries with lowest AQI value of yesterday.")
```



```
[54]: fig = px.choropleth(b,locations = "Country", locationmode='country names',␣
        ↪color="AQI Value"
                          ,hover_name= "Status",range_color= [25,450]
                          ,title = "Top 10 countries with lowest AQI value of␣
        ↪yesterday.")
      fig.show()
```



7 Top 10 Major Countries of the world

```
[182]: major_countries = df[df['Country'].isin(['India',␣
         ↪'USA','China','Brazil','France','Russia','Great␣
         ↪Britain','Japan','Germany','South Africa'])].reset_index()
       major_countries
```

[182]:

```
     index       Date       Country                          Status  \
0       17 2022-07-21        Brazil                        Moderate
1       28 2022-07-21         China                       Unhealthy
2       42 2022-07-21        France                        Moderate
3       47 2022-07-21       Germany                        Moderate
4       59 2022-07-21         India                       Unhealthy
..     …          …           …                             …
739  12897 2023-08-10       Germany  Unhealthy for Sensitive Groups
740  12908 2023-08-10         India                       Unhealthy
741  12916 2023-08-10         Japan                        Moderate
742  12957 2023-08-10        Russia                        Moderate
743  12963 2023-08-10  South Africa                       Unhealthy

     AQI Value
0           67
1          160
2           72
3           77
4          162
..         …
739        145
740        163
741         70
742         57
743        158

[744 rows x 5 columns]
```

8 importing datetime

[185]:

```python
import datetime as dt
```

converting object dtype of Date column into datetime dtype

[188]:

```python
major_countries["Date"] = pd.to_datetime(major_countries["Date"])
```

making new columns from Date column : Month , week , day

[191]:

```python
major_countries["month"] = major_countries["Date"].dt.month
major_countries["day"] = major_countries["Date"].dt.day
major_countries["week"] = major_countries["Date"].dt.week


major_countries.head()
```

```
/tmp/ipykernel_160/1978257230.py:3: FutureWarning:
```

```
Series.dt.weekofyear and Series.dt.week have been deprecated. Please use
Series.dt.isocalendar().week instead.
```

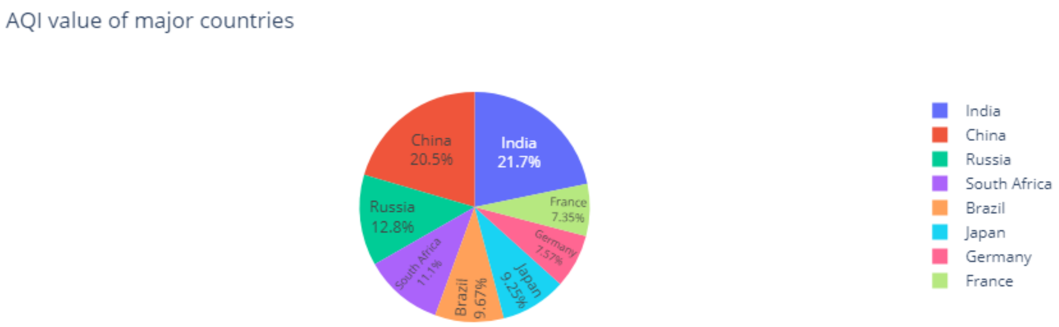
```
[191]:    index       Date  Country    Status  AQI Value  month  day  week
       0     17 2022-07-21   Brazil  Moderate         67      7   21    29
       1     28 2022-07-21    China Unhealthy        160      7   21    29
       2     42 2022-07-21   France  Moderate         72      7   21    29
       3     47 2022-07-21  Germany  Moderate         77      7   21    29
       4     59 2022-07-21    India Unhealthy        162      7   21    29
```

```
[193]: #checking information about dataset
       major_countries.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 744 entries, 0 to 743
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   index      744 non-null    int64
 1   Date       744 non-null    datetime64[ns]
 2   Country    744 non-null    object
 3   Status     744 non-null    object
 4   AQI Value  744 non-null    int64
 5   month      744 non-null    int64
 6   day        744 non-null    int64
 7   week       744 non-null    int64
dtypes: datetime64[ns](1), int64(5), object(2)
memory usage: 46.6+ KB
```

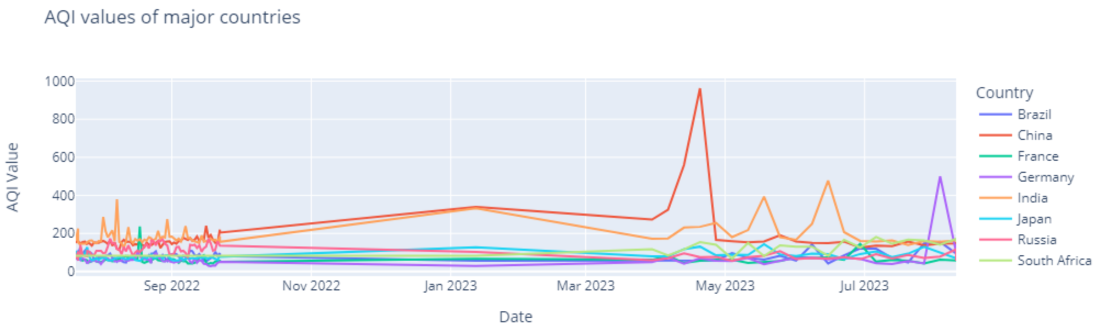
Pie chart of major countries Average AQI values over all dates.

```
[196]: fig = px.pie(major_countries , names = "Country" ,values = "AQI Value", title =␣
         ↪"AQI value of major countries", hover_data = ["Status"] )
       fig.update_traces(textposition='inside', textinfo='percent+label')
```



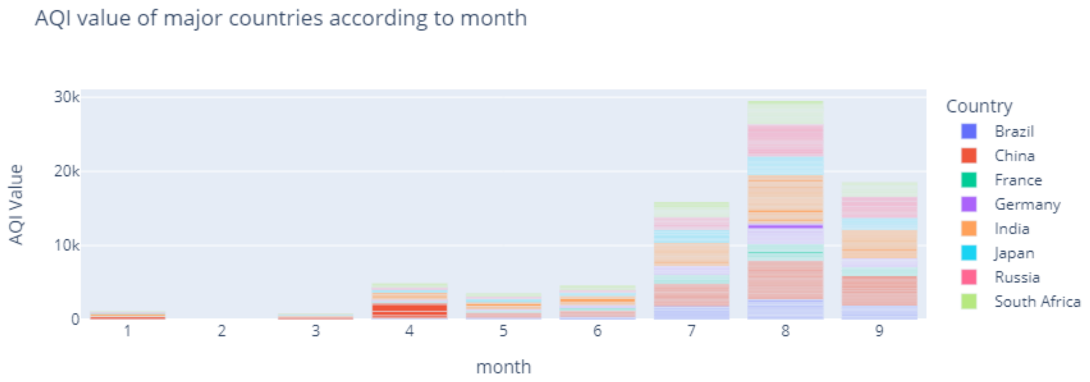
line plot of major countries AQI value

```
[204]: px.line(major_countries ,x = 'Date', y = 'AQI Value', color = "Country" , title␣
         ↪= "AQI values of major countries")
```



bar plot of major countries AQI value

```
[ ]: px.bar(major_countries ,x = 'month', y = 'AQI Value', color = "Country" , title␣
       ↪= "AQI value of major countries according to month" )
```



9 Reference

- https://plotly.com/python/bar-charts/
- https://plotly.com/python/choropleth-maps/
- https://plotly.com/python/line-charts/

- https://medium.com/@sawsanyusuf/data-visualization-with-python-10-choropleth-maps-df7ab3118c3a -https://medium.com/@mleblog/creating-a-choropleth-map-using-geopandas-and-financial-data-c9839a51c187
- https://towardsdatascience.com/tagged/choropleth-map

[ ]: