Transformations on API Source

Kaylynn Mosier

12 May 2024

In [203…
```
# Load required packages
import json
import requests
import pandas as pd
```

API key

I have saved my API key in a json file to keep in secret. I used json.dump() to do this. I have omitted this section of code from the assignment to keep my key private.

In [204…
```
# Opening JSON file that stores API key
with open("OpenWeaterAPIKey.json", "r") as openfile:
    # Reading from JSON file
    json_object = json.load(openfile)

# Saving API key as variable
api_key = json_object['api_key']
```

Testing Air Quality API

Before I begin to write my functions, I want to check that my API requests are working correctly.

In [205…
```
# Saves base URL
# Latitude and longitude of city along with start and end date of observations will be added as variables to this url
base_url_test = "http://api.openweathermap.org/data/2.5/air_pollution?"
```

In [206…
```
api_key_test = '7db0bf338907c0e424aee048b1d8369a'
lat_test = '36.19' # Latitude of test city
lon_test = '-94.49' # Longitude of test city
```

In [207…
```
# Constructs final url with all need information
url_test = base_url_test + 'lat=' + lat_test + '&lon=' + lon_test + '&appid=' + api_key_test
# Prints url to confirm concatenation has worked correctly
url_test
```

Out[207… `'http://api.openweathermap.org/data/2.5/air_pollution?lat=36.19&lon=-94.49&appid=7db0bf338907c0e424aee048b1d8369a'`

In [208…
```
# Submit a request to the API
response_test = requests.get(url_test).json()
```

In [209…
```
# Print response from API
response_test
```

Out[209…
```
{'coord': {'lon': -94.49, 'lat': 36.19},
 'list': [{'main': {'aqi': 1},
   'components': {'co': 185.25,
    'no': 0,
    'no2': 1.33,
    'o3': 38.62,
    'so2': 0.46,
    'pm2_5': 2.1,
    'pm10': 3.38,
    'nh3': 1.77},
   'dt': 1717034216}]}
```

In [210…
```
# Find quality rating
quality_rating_test = response_test['list'][0]['main']['aqi']
# Find concentration carbon monoxide
concentration_CO_test = response_test['list'][0]['components']['co']

print("Air quality rating: {}".format(quality_rating_test))
print("Concentration Carbon Monoxide: {}".format(concentration_CO_test))
```

```
Air quality rating: 1
Concentration Carbon Monoxide: 185.25
```

Looks like my API calls are working correctly and I can access the data I need to!

Constructing DataFrame of API data

In [211…
```
# List of cities that API requests need to be made on
# This list is from my Milestone 2 data and will be used to merge the dataframes together
cities_list = ['Algiers', 'Bujumbura', 'Cotonou', 'Bangui', 'Brazzaville',
       'Cairo', 'Addis Ababa', 'Libreville', 'Banjul', 'Conakry',
       'Bissau', 'Abidjan', 'Nairobi', 'Rabat', 'Antananarivo',
       'Nouakchott', 'Lilongwe', 'Maputo', 'Windhoek', 'Niamey', 'Lagos',
       'Dakar', 'Freetown', 'Capetown', 'Lome', 'Tunis', 'Dar Es Salaam',
       'Kampala', 'Lusaka', 'Dhaka', 'Beijing', 'Chengdu', 'Guangzhou',
       'Shanghai', 'Shenyang', 'Hong Kong', 'Bombay (Mumbai)', 'Calcutta',
       'Chennai (Madras)', 'Delhi', 'Jakarta', 'Osaka', 'Sapporo',
       'Tokyo', 'Almaty', 'Bishkek', 'Vientiane', 'Kuala Lumpur',
       'Ulan-bator', 'Rangoon', 'Katmandu', 'Pyongyang', 'Islamabad',
       'Karachi', 'Manila', 'Singapore', 'Seoul', 'Colombo', 'Taipei',
       'Dusanbe', 'Bangkok', 'Ashabad', 'Tashkent', 'Hanoi', 'Brisbane',
       'Canberra', 'Melbourne', 'Perth', 'Sydney', 'Auckland', 'Tirana',
       'Vienna', 'Minsk', 'Brussels', 'Sofia', 'Zagreb', 'Nicosia',
       'Prague', 'Copenhagen', 'Helsinki', 'Paris', 'Bordeaux', 'Bonn',
       'Frankfurt', 'Hamburg', 'Munich', 'Tbilisi', 'Athens', 'Budapest',
       'Reykjavik', 'Dublin', 'Milan', 'Rome', 'Riga', 'Skopje',
       'Amsterdam', 'Oslo', 'Warsaw', 'Lisbon', 'Bucharest', 'Moscow',
       'Yerevan', 'Pristina', 'Bratislava', 'Barcelona', 'Bilbao',
       'Madrid', 'Stockholm', 'Bern', 'Geneva', 'Zurich', 'Kiev',
       'Belfast', 'London', 'Belgrade', 'Manama', 'Tel Aviv', 'Amman',
       'Kuwait', 'Beirut']
```

In [212…
```
def get_coordinates(city):
    """
    Function to find latitude and longitude of cities using OpenWeatherMap Geocoding API

    Returns latitude and longitude of city

    """
    geocoding_base = "http://api.openweathermap.org/geo/1.0/direct?" # Base URL for geocoding API
    # api_key = '7db0bf338907c0e424aee048b1d8369a'
    limit = '1' # Finds information on only the first city
    geocoding_url = geocoding_base + 'q=' + city + '&limit=' + limit + '&appid=' + api_key # Final URL for geocoding API
    geocoding_response = requests.get(geocoding_url).json() # Submites request to API
    lat = geocoding_response[0]['lat'] # Finds Latitude
    lon = geocoding_response[0]['lon'] # Finds Longitude
```

```
    return lat, lon
```

In [213…
```
def get_air_quality(lat, lon):
    """
    Function that finds Air Quality data from the OpenWeatherMap Air Quality API

    Returns response from API

    """

    base_url = "http://api.openweathermap.org/data/2.5/air_pollution?"
    # api_key = '7db0bf338907c0e424aee048b1d8369a'
    url = base_url + 'lat=' + str(lat) + '&lon=' + str(lon) + '&appid=' + api_key

    # Submits request to API
    response = requests.get(url).json()

    return response
```

In [214…
```
def build_dataframe(cities):
    """
    Takes a list of city names

    Returns a DataFrame with needed air quality information from OpenWeatherMap Air Quality API

    """
    # Define an empty dictionary with keys
    city_dict={'City':[], 'Quality Rating':[], 'Concentration CO':[], 'Concentration NO':[],
               'Concentration NO2':[], 'Concentration NH3':[], 'Concentration O3':[], 'Concentration SO2':[],
               'Concentration PM2.5':[], 'Concentration PM10':[], 'DateTime':[], 'Lat':[], 'Lon':[]}


    # This section of code parses required data and appends it to the dictionary
    for city in cities:
        lat, lon = get_coordinates(city)
        response = get_air_quality(lat, lon)
        city_dict['City'].append(city)
        city_dict['Lat'].append(lat)
        city_dict['Lon'].append(lon)
        city_dict['Quality Rating'].append(response['list'][0]['main']['aqi'])
        city_dict['Concentration CO'].append(response['list'][0]['components']['co'])
        city_dict['Concentration NO'].append(response['list'][0]['components']['no'])
        city_dict['Concentration NO2'].append(response['list'][0]['components']['no2'])
        city_dict['Concentration NH3'].append(response['list'][0]['components']['nh3'])
        city_dict['Concentration O3'].append(response['list'][0]['components']['o3'])
        city_dict['Concentration SO2'].append(response['list'][0]['components']['so2'])
        city_dict['Concentration PM2.5'].append(response['list'][0]['components']['pm2_5'])
        city_dict['Concentration PM10'].append(response['list'][0]['components']['pm10'])
        city_dict['DateTime'].append(response['list'][0]['dt'])

    return pd.DataFrame(city_dict)
```

In [215…
```
# Construct a dataframe of all API information using list of cities in Milestone 2
air_pollution_data = build_dataframe(cities_list)
air_pollution_data
```

Out[215…

	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Lat	Lon
0	Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	1717034229	36.775361	3.060188
1	Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	1717034230	-3.363812	29.367503
2	Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	1717034231	6.367695	2.425251
3	Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	1717034233	4.390715	18.550913
4	Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	1717034234	-4.269441	15.271226
…	…	…	…	…	…	…	…	…	…	…	…	…	…
115	Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	1717034359	26.223504	50.582244
116	Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	1717034360	32.085300	34.781806
117	Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	1717034361	31.951569	35.923963
118	Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	1717034362	29.379653	47.973417
119	Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	1717034363	33.895920	35.478430

120 rows × 13 columns

Transformation 1- Change date type of DateTime column

In [216…
```
# Check data types of dataframe columns
air_pollution_data.dtypes
```

Out[216…
```
City                  object
Quality Rating         int64
Concentration CO     float64
Concentration NO     float64
Concentration NO2    float64
Concentration NH3    float64
Concentration O3     float64
Concentration SO2    float64
Concentration PM2.5  float64
Concentration PM10   float64
DateTime               int64
Lat                  float64
Lon                  float64
dtype: object
```

In [217…
```
# Imports datetime package
from datetime import datetime
```

In [218…
```
# Uses to_datetime to transform column to datetime format
air_pollution_data['DateTime'] = pd.to_datetime(air_pollution_data['DateTime'], unit='s')
air_pollution_data
```

Out[218…

	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Lat	Lon
0	Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	36.775361	3.060188
1	Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	-3.363812	29.367503
2	Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	6.367695	2.425251
3	Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	4.390715	18.550913
4	Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	-4.269441	15.271226
...	...	...	...	...	...	...	...	...	...	...	...	...	...
115	Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	26.223504	50.582244
116	Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	32.085300	34.781806
117	Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	31.951569	35.923963
118	Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	29.379653	47.973417
119	Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	33.895920	35.478430

120 rows × 13 columns

Transformation 2- Add column for decode of Quality Rating

Using information from the API documentation, the values in the quality rating column are equilavent to qualitative ratings.

```
1 Good
2 Fair
3 Moderate
4 Poor
5 Very Poor
```

In [219…

```
# Duplicate quality rating values in a new column titled qualitative name
air_pollution_data['Qualitative Name'] = air_pollution_data['Quality Rating']
air_pollution_data
```

Out[219…

	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Lat	Lon	Qualitative Name
0	Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	36.775361	3.060188	1
1	Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	-3.363812	29.367503	3
2	Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	6.367695	2.425251	1
3	Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	4.390715	18.550913	1
4	Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	-4.269441	15.271226	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115	Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	26.223504	50.582244	4
116	Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	32.085300	34.781806	3
117	Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	31.951569	35.923963	2
118	Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	29.379653	47.973417	4
119	Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	33.895920	35.478430	2

120 rows × 14 columns

In [220…

```
# Replace numerical values with qualitative name
air_pollution_data['Qualitative Name'] = air_pollution_data['Qualitative Name'].replace([1, 2, 3, 4, 5], ['Good', 'Fair', 'Moderate', 'Poor', 'Very Poor'])
air_pollution_data
```

Out[220…

	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Lat	Lon	Qualitative Name
0	Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	36.775361	3.060188	Good
1	Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	-3.363812	29.367503	Moderate
2	Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	6.367695	2.425251	Good
3	Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	4.390715	18.550913	Good
4	Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	-4.269441	15.271226	Good
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
115	Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	26.223504	50.582244	Poor
116	Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	32.085300	34.781806	Moderate
117	Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	31.951569	35.923963	Fair
118	Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	29.379653	47.973417	Poor
119	Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	33.895920	35.478430	Fair

120 rows × 14 columns

Transformation 3- Set index

In [221…

```
air_pollution_data = air_pollution_data.set_index(air_pollution_data['City']) # Sets index as City
air_pollution_data = air_pollution_data.drop('City', axis=1) # Drops city column
air_pollution_data
```

Out[221…

City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Lat	Lon	Qualitative Name
Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	36.775361	3.060188	Good
Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	-3.363812	29.367503	Moderate
Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	6.367695	2.425251	Good
Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	4.390715	18.550913	Good
Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	-4.269441	15.271226	Good
…	…	…	…	…	…	…	…	…	…	…	…	…	…
Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	26.223504	50.582244	Poor
Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	32.085300	34.781806	Moderate
Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	31.951569	35.923963	Fair
Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	29.379653	47.973417	Poor
Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	33.895920	35.478430	Fair

120 rows × 13 columns

Transformation 4- Drop Lat and Lon columns

None of my other datasets include latitude and longitude inforamation. This information was useful at first to ensure my functions were working correctly, but since the city is already included as a column, this information a redundent.

In [222…

```
air_pollution_data = air_pollution_data.drop(['Lat', 'Lon'], axis=1) # Drop Lat and Lon columns
air_pollution_data
```

Out[222...

City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name
Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	Good
Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	Moderate
Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	Good
Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	Good
Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	Good
...	...	...	...	...	...	...	...	...	...	...	...
Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	Poor
Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	Moderate
Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	Fair
Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	Poor
Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	Fair

120 rows × 11 columns

Transformation 5- Add country column

Multiple countries can have the same city names. To make sure comparisons to my other datasets are correct, it is useful to know the country each city is in. This will require an additional request to the Geocoding API.

In [223...

```
def get_country(city):

    """
    Function to find country of cities using OpenWeatherMap Geocoding API

    Returns country code

    """
    geocoding_base = "http://api.openweathermap.org/geo/1.0/direct?" # Base URL for geocoding API
    api_key = '7db0bf338907c0e424aee048b1d8369a'
    limit = '1'
    geocoding_url = geocoding_base + 'q=' + city + '&limit=' + limit + '&appid=' + api_key # Final URL for geocoding API
    geocoding_response = requests.get(geocoding_url).json() # Submits request to API
    country_code = geocoding_response[0]['country'] # Finds latitude

    return country_code
```

In [224...

```
def build_city_country_dict(cities):
    """
    Takes a list of city names

    Returns a dictionary with city names and country code
    """

    city_country_dict = {'City':[], 'Country Code':[]} # Creates empty dictionary to store data

    for city in cities_list:
        city_country_dict['City'].append(city) # Appends city to dictionary
        country_code = get_country(city) # Calls function to get country_code
        city_country_dict['Country Code'].append(country_code) # Appends country_code to dictionary

    return pd.DataFrame(city_country_dict)
```

In [225...

```
# Create dictionary of cities and countries
city_country_df = build_city_country_dict(cities_list)
```

In [226...

```
city_country_df = city_country_df.set_index(city_country_df['City']) # Setting index
city_country_df = city_country_df.drop('City', axis=1) # Dropping city column because it is now the index
city_country_df
```

Out[226...

City	Country Code
Algiers	DZ
Bujumbura	BI
Cotonou	BJ
Bangui	CF
Brazzaville	CG
...	...
Manama	BH
Tel Aviv	IL
Amman	JO
Kuwait	KW
Beirut	LB

120 rows × 1 columns

In [227...

```
# Join dataframes together based on index
air_pollution_data = air_pollution_data.join(city_country_df)
```

In [228...

```
air_pollution_data
```

Out[228…

City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name	Country Code
Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	Good	DZ
Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	Moderate	BI
Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	Good	BJ
Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	Good	CF
Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	Good	CG
…	…	…	…	…	…	…	…	…	…	…	…	…
Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	Poor	BH
Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	Moderate	IL
Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	Fair	JO
Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	Poor	KW
Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	Fair	LB

120 rows × 12 columns

Transformation 6- Add column for country name using additional dataset

In [229…

```
# Open dataset that contains country name and alpha2 code
# Found at: https://www.kaggle.com/datasets/emolodov/country-codes-alpha2-alpha3
country_codes_data = pd.read_csv("C:/Users/kayly/OneDrive/Desktop/MSDS/DSC540/Tem Project/CountryCodes.csv")
country_codes_data
```

Out[229…

	country	alpha2	alpha3	numeric
0	Afghanistan	AF	AFG	4
1	Albania	AL	ALB	8
2	Algeria	DZ	DZA	12
3	American Samoa	AS	ASM	16
4	Andorra	AD	AND	20
…	…	…	…	…
244	Western Sahara	EH	ESH	732
245	Yemen	YE	YEM	887
246	Zambia	ZM	ZMB	894
247	Zimbabwe	ZW	ZWE	716
248	Åland Islands	AX	ALA	248

249 rows × 4 columns

In [230…

```
# Drop columns that are not needed
country_codes_data = country_codes_data.drop(['alpha3', 'numeric'], axis=1)
country_codes_data
```

Out[230…

	country	alpha2
0	Afghanistan	AF
1	Albania	AL
2	Algeria	DZ
3	American Samoa	AS
4	Andorra	AD
…	…	…
244	Western Sahara	EH
245	Yemen	YE
246	Zambia	ZM
247	Zimbabwe	ZW
248	Åland Islands	AX

249 rows × 2 columns

In [231…

```
# Rename columns
country_codes_data.rename(columns={'alpha2':'Country Code', 'country':'Country'}, inplace=True)
country_codes_data
```

Out[231…

	Country	Country Code
0	Afghanistan	AF
1	Albania	AL
2	Algeria	DZ
3	American Samoa	AS
4	Andorra	AD
...	...	...
244	Western Sahara	EH
245	Yemen	YE
246	Zambia	ZM
247	Zimbabwe	ZW
248	Åland Islands	AX

249 rows × 2 columns

In [232…
```
# Set index
country_codes_data.set_index('Country Code', inplace=True)
country_codes_data
```

Out[232…

Country Code	Country
AF	Afghanistan
AL	Albania
DZ	Algeria
AS	American Samoa
AD	Andorra
...	...
EH	Western Sahara
YE	Yemen
ZM	Zambia
ZW	Zimbabwe
AX	Åland Islands

249 rows × 1 columns

In [234…
```
air_pollution_data = air_pollution_data.reset_index()
air_pollution_data.set_index('Country Code', inplace=True)
air_pollution_data
```

Out[234…

Country Code	index	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name
DZ	0	Algiers	1	240.33	0.01	39.07	0.43	23.60	5.54	4.29	8.38	2024-05-30 01:57:09	Good
BI	1	Bujumbura	3	647.54	0.00	4.16	2.41	10.46	0.48	27.18	60.28	2024-05-30 01:57:10	Moderate
BJ	2	Cotonou	1	343.80	0.00	0.51	1.30	36.84	0.57	3.66	9.32	2024-05-30 01:57:11	Good
CF	3	Bangui	1	263.69	0.00	0.11	0.07	0.15	0.00	0.50	1.16	2024-05-30 01:57:13	Good
CG	4	Brazzaville	1	357.15	0.00	0.91	0.81	18.95	0.30	6.04	16.23	2024-05-30 01:57:14	Good
...	...	...	...	...	...	...	...	...	...	...	...	...	...
BH	115	Manama	4	260.35	0.00	11.48	0.65	91.55	29.56	54.10	173.35	2024-05-30 01:59:19	Poor
IL	116	Tel Aviv	3	185.25	0.00	0.34	0.00	110.15	0.64	10.57	23.45	2024-05-30 01:59:20	Moderate
JO	117	Amman	2	180.24	0.00	6.00	0.96	60.08	3.55	13.34	33.41	2024-05-30 01:59:21	Fair
KW	118	Kuwait	4	230.31	0.00	1.95	0.35	92.98	2.12	35.86	127.23	2024-05-30 01:59:22	Poor
LB	119	Beirut	2	196.93	0.00	2.87	0.00	98.71	3.43	13.17	25.16	2024-05-30 01:59:23	Fair

120 rows × 13 columns

In [235…
```
# Join datasets on index (country code)
air_pollution_data = air_pollution_data.join(country_codes_data)
```

In [236…
```
air_pollution_data = air_pollution_data.reset_index()
air_pollution_data
```

Out[236…

	Country Code	index	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name	Country
0	AL	70	Tirana	1	208.62	0.00	3.90	2.95	40.41	0.30	8.70	10.92	2024-05-30 01:58:36	Good	Albania
1	AM	101	Yerevan	1	175.24	0.00	6.86	5.00	38.62	0.60	6.12	15.66	2024-05-30 01:59:07	Good	Armenia
2	AT	71	Vienna	1	216.96	0.00	7.28	3.01	28.61	0.66	5.24	6.72	2024-05-30 01:58:36	Good	Austria
3	AU	64	Brisbane	2	247.00	1.72	6.94	0.58	62.94	5.54	2.28	3.93	2024-05-30 01:58:29	Fair	Australia
4	AU	65	Canberra	1	223.64	0.09	0.39	0.13	47.92	0.23	3.46	3.73	2024-05-30 01:58:32	Good	Australia
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
115	US	23	Capetown	2	196.93	0.05	0.34	0.00	87.98	0.45	0.79	2.25	2024-05-30 01:57:34	Fair	United States of America (the)
116	UZ	62	Tashkent	1	367.17	23.69	35.30	3.77	11.36	2.92	9.00	14.01	2024-05-30 01:58:28	Good	Uzbekistan
117	VN	63	Hanoi	3	827.79	1.30	25.71	9.12	9.66	13.95	34.47	47.49	2024-05-30 01:50:24	Moderate	Viet Nam
118	XK	102	Pristina	1	203.61	0.01	6.26	4.18	18.06	1.73	8.37	11.71	2024-05-30 01:59:08	Good	NaN
119	ZM	28	Lusaka	2	1415.25	0.00	6.34	11.65	37.19	6.14	11.48	19.94	2024-05-30 01:57:44	Fair	Zambia

120 rows × 15 columns

In [237…
```
# Check for NA values
air_pollution_data.isna().sum()
```

Out[237…
```
Country Code           0
index                  0
City                   0
Quality Rating         0
Concentration CO       0
Concentration NO       0
Concentration NO2      0
Concentration NH3      0
Concentration O3       0
Concentration SO2      0
Concentration PM2.5    0
Concentration PM10     0
DateTime               0
Qualitative Name       0
Country                2
dtype: int64
```

There are two countries with NA values for the Country Name column.

In [238…
```
# Find rows with NA values
air_pollution_data[air_pollution_data['Country'].isna()]
```

Out[238…

	Country Code	index	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name	Country
86	NA	18	Windhoek	1	216.96	0.00	0.40	0.20	33.62	0.87	7.79	10.34	2024-05-30 01:57:29	Good	NaN
118	XK	102	Pristina	1	203.61	0.01	6.26	4.18	18.06	1.73	8.37	11.71	2024-05-30 01:59:08	Good	NaN

The country names in rows 86 and 118 are, I will drop these rows.

In [239…
```
air_pollution_data.set_index('Country Code', inplace=True)
air_pollution_data = air_pollution_data.drop(['NA', 'XK'])
air_pollution_data
```

Out[239…

Country Code	index	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name	Country
AL	70	Tirana	1	208.62	0.00	3.90	2.95	40.41	0.30	8.70	10.92	2024-05-30 01:58:36	Good	Albania
AM	101	Yerevan	1	175.24	0.00	6.86	5.00	38.62	0.60	6.12	15.66	2024-05-30 01:59:07	Good	Armenia
AT	71	Vienna	1	216.96	0.00	7.28	3.01	28.61	0.66	5.24	6.72	2024-05-30 01:58:36	Good	Austria
AU	64	Brisbane	2	247.00	1.72	6.94	0.58	62.94	5.54	2.28	3.93	2024-05-30 01:58:29	Fair	Australia
AU	65	Canberra	1	223.64	0.09	0.39	0.13	47.92	0.23	3.46	3.73	2024-05-30 01:58:32	Good	Australia
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
UG	27	Kampala	5	1628.88	0.09	7.45	6.59	2.48	2.27	98.01	132.54	2024-05-30 01:57:42	Very Poor	Uganda
US	23	Capetown	2	196.93	0.05	0.34	0.00	87.98	0.45	0.79	2.25	2024-05-30 01:57:34	Fair	United States of America (the)
UZ	62	Tashkent	1	367.17	23.69	35.30	3.77	11.36	2.92	9.00	14.01	2024-05-30 01:58:28	Good	Uzbekistan
VN	63	Hanoi	3	827.79	1.30	25.71	9.12	9.66	13.95	34.47	47.49	2024-05-30 01:50:24	Moderate	Viet Nam
ZM	28	Lusaka	2	1415.25	0.00	6.34	11.65	37.19	6.14	11.48	19.94	2024-05-30 01:57:44	Fair	Zambia

118 rows × 14 columns

In [240…
```
# Previous drop successfully dropped needed row
air_pollution_data[air_pollution_data['Country'].isna()]
```

Out[240…

Country Code	index	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name	Country

In [245…
```
air_pollution_data = air_pollution_data.drop('index', axis=1)
air_pollution_data
```

Out[245…

Country	Country Code	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name
Albania	AL	Tirana	1	208.62	0.00	3.90	2.95	40.41	0.30	8.70	10.92	2024-05-30 01:58:36	Good
Armenia	AM	Yerevan	1	175.24	0.00	6.86	5.00	38.62	0.60	6.12	15.66	2024-05-30 01:59:07	Good
Austria	AT	Vienna	1	216.96	0.00	7.28	3.01	28.61	0.66	5.24	6.72	2024-05-30 01:58:36	Good
Australia	AU	Brisbane	2	247.00	1.72	6.94	0.58	62.94	5.54	2.28	3.93	2024-05-30 01:58:29	Fair
Australia	AU	Canberra	1	223.64	0.09	0.39	0.13	47.92	0.23	3.46	3.73	2024-05-30 01:58:32	Good
...	...	...	...	...	...	...	...	...	...	...	...	...	...
Uganda	UG	Kampala	5	1628.88	0.09	7.45	6.59	2.48	2.27	98.01	132.54	2024-05-30 01:57:42	Very Poor
United States of America (the)	US	Capetown	2	196.93	0.05	0.34	0.00	87.98	0.45	0.79	2.25	2024-05-30 01:57:34	Fair
Uzbekistan	UZ	Tashkent	1	367.17	23.69	35.30	3.77	11.36	2.92	9.00	14.01	2024-05-30 01:58:28	Good
Viet Nam	VN	Hanoi	3	827.79	1.30	25.71	9.12	9.66	13.95	34.47	47.49	2024-05-30 01:50:24	Moderate
Zambia	ZM	Lusaka	2	1415.25	0.00	6.34	11.65	37.19	6.14	11.48	19.94	2024-05-30 01:57:44	Fair

118 rows × 13 columns

Final Dataset

In [246…
```
air_pollution_data = air_pollution_data.reset_index()
air_pollution_data.set_index('Country', inplace=True)
air_pollution_data
```

Out[246…

Country	Country Code	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name
Albania	AL	Tirana	1	208.62	0.00	3.90	2.95	40.41	0.30	8.70	10.92	2024-05-30 01:58:36	Good
Armenia	AM	Yerevan	1	175.24	0.00	6.86	5.00	38.62	0.60	6.12	15.66	2024-05-30 01:59:07	Good
Austria	AT	Vienna	1	216.96	0.00	7.28	3.01	28.61	0.66	5.24	6.72	2024-05-30 01:58:36	Good
Australia	AU	Brisbane	2	247.00	1.72	6.94	0.58	62.94	5.54	2.28	3.93	2024-05-30 01:58:29	Fair
Australia	AU	Canberra	1	223.64	0.09	0.39	0.13	47.92	0.23	3.46	3.73	2024-05-30 01:58:32	Good
...	...	...	...	...	...	...	...	...	...	...	...	...	...
Uganda	UG	Kampala	5	1628.88	0.09	7.45	6.59	2.48	2.27	98.01	132.54	2024-05-30 01:57:42	Very Poor
United States of America (the)	US	Capetown	2	196.93	0.05	0.34	0.00	87.98	0.45	0.79	2.25	2024-05-30 01:57:34	Fair
Uzbekistan	UZ	Tashkent	1	367.17	23.69	35.30	3.77	11.36	2.92	9.00	14.01	2024-05-30 01:58:28	Good
Viet Nam	VN	Hanoi	3	827.79	1.30	25.71	9.12	9.66	13.95	34.47	47.49	2024-05-30 01:50:24	Moderate
Zambia	ZM	Lusaka	2	1415.25	0.00	6.34	11.65	37.19	6.14	11.48	19.94	2024-05-30 01:57:44	Fair

118 rows × 13 columns

Writing final table to CSV file

In [242…
```
import csv
```

In [247…
```
# Writing dataframe to a csv file
air_pollution_data.to_csv('AirPollutionData', sep=',', encoding='utf-8', index=True)
```

In [244…
```
# Checking that writing to file worked correctly
csvFile = pd.read_csv("C:/Users/kayly/OneDrive/Desktop/MSDS/DSC540/Tem Project/AirPollutionData")
csvFile
```

Out [244...

	Country	Country Code	index	City	Quality Rating	Concentration CO	Concentration NO	Concentration NO2	Concentration NH3	Concentration O3	Concentration SO2	Concentration PM2.5	Concentration PM10	DateTime	Qualitative Name
0	Albania	AL	70	Tirana	1	208.62	0.00	3.90	2.95	40.41	0.30	8.70	10.92	2024-05-30 01:58:36	Good
1	Armenia	AM	101	Yerevan	1	175.24	0.00	6.86	5.00	38.62	0.60	6.12	15.66	2024-05-30 01:59:07	Good
2	Austria	AT	71	Vienna	1	216.96	0.00	7.28	3.01	28.61	0.66	5.24	6.72	2024-05-30 01:58:36	Good
3	Australia	AU	64	Brisbane	2	247.00	1.72	6.94	0.58	62.94	5.54	2.28	3.93	2024-05-30 01:58:29	Fair
4	Australia	AU	65	Canberra	1	223.64	0.09	0.39	0.13	47.92	0.23	3.46	3.73	2024-05-30 01:58:32	Good
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
113	Uganda	UG	27	Kampala	5	1628.88	0.09	7.45	6.59	2.48	2.27	98.01	132.54	2024-05-30 01:57:42	Very Poor
114	United States of America (the)	US	23	Capetown	2	196.93	0.05	0.34	0.00	87.98	0.45	0.79	2.25	2024-05-30 01:57:34	Fair
115	Uzbekistan	UZ	62	Tashkent	1	367.17	23.69	35.30	3.77	11.36	2.92	9.00	14.01	2024-05-30 01:58:28	Good
116	Viet Nam	VN	63	Hanoi	3	827.79	1.30	25.71	9.12	9.66	13.95	34.47	47.49	2024-05-30 01:50:24	Moderate
117	Zambia	ZM	28	Lusaka	2	1415.25	0.00	6.34	11.65	37.19	6.14	11.48	19.94	2024-05-30 01:57:44	Fair

118 rows × 15 columns

Ethical implications

The API I used provided thorough clean data that did not require much manipulation. I changed the DateTime information into a more human-readable format. Other than that, I did not alter any data provided by the API, I basically just re-formatted the dataframe to be more compatable with my other data sources. I do not see any additional risk being added or created by my transformations. As far as I am aware, there are no legal or regulatory guidelines for my topic.

The API documentation did note that each country has their own standard for air quality ratings. This means that a quality rating of Medium in the UK may be a different rating in the US. It is not clear if the API returns a quality rating for each country based on that countrys' standards or by a chosen set of standards. It makes more sense that a chosen set of standards would be used (for example, apply US standards to all countries) because this allows for normalization of the quality ratings. If this is not the case, it would be difficult to compare quality ratings accross countries. I chose to assume all cities are being held to the same standard. This could be a risky assumption because it could lead to false correlations and incorrect conclusions.

I got my data from OpenWeatherMap which is a well known and reputable API source. I do not have concerns about the quality of validity of my data. I also do not have ethical concerns about sourcing of the data.