In [1]:
```
import pandas as pd
import os
import numpy as np
import multiprocessing as mp
import geopandas as gpd
import matplotlib.pyplot as plt
import folium
from pprint import pprint
from datetime import date
from dateutil import relativedelta
```

```
/Users/lorenzo/opt/anaconda3/lib/python3.9/site-packages/pandas/core/
computation/expressions.py:21: UserWarning: Pandas requires version '
2.8.4' or newer of 'numexpr' (version '2.8.3' currently installed).
  from pandas.core.computation.check import NUMEXPR_INSTALLED
/Users/lorenzo/opt/anaconda3/lib/python3.9/site-packages/pandas/core/
arrays/masked.py:60: UserWarning: Pandas requires version '1.3.6' or
newer of 'bottleneck' (version '1.3.5' currently installed).
  from pandas.core import (
```

In [4]:
```
dict_var = {
    'PWSID' : object,
    'VIOLATION_ID' : object,
    'VIOLATION_CODE' : object,
    'IS_HEALTH_BASED_IND' : object,
    'CONTAMINANT_CODE' : object,
    'IS_MAJOR_VIOL_IND' : object,
    'VIOLATION_STATUS' : object,
    'PUBLIC_NOTIFICATION_TIER' : object,
    'VIOL_FIRST_REPORTED_DATE' : object
}
```

In [5]:
```
SDWA_VIOL = pd.read_csv("/Volumes/T7/Water Project/SDWA_latest_downloa
                        iterator=True, chunksize = 100000, usecols=["PW
                                                    "VIOLATION_ID",
                                                    "VIOLATION_CODE",
                                                    'IS_HEALTH_BASED_I
                                                    'CONTAMINANT_CODE'
                                                    'IS_MAJOR_VIOL_IND
                                                    'VIOLATION_STATUS'
                                                    'PUBLIC_NOTIFICATI
                                                    'VIOL_FIRST_REPORT
    ## Clean based on Column and Value
df_clean = pd.concat([chunk[chunk['VIOLATION_STATUS'] == "Unaddressed"
```

In [6]:
```
# Define the file path
file_path = "/Volumes/T7/Water Project/SDWA_latest_downloads/SDWA_VIOL

# Define the chunk size
chunk_size = 10000  # Adjust the chunk size based on your memory capac

# Create an empty set to store unique PWSID values
unique_pwsid = set()

# Iterate over the file in chunks
for chunk in pd.read_csv(file_path, chunksize=chunk_size):
    # Update the set with unique PWSID values from the current chunk
    unique_pwsid.update(chunk['PWSID'].unique())

# Convert the set to a list if you need a list of unique PWSID values
unique_pwsid_list = list(unique_pwsid)
```

In [7]:
```
len(unique_pwsid_list)
```

Out[7]: 254666

In [8]:
```
UA_VIOL = df_clean.drop_duplicates()
```

In [9]:
```
UA_PWSID = list(UA_VIOL['PWSID'].unique())
```

In [10]:
```
len(UA_PWSID)
```

Out[10]: 16630

In [11]:
```
data = UA_VIOL[UA_VIOL["PWSID"] == "WV9938038"]
scoring_dict = {"3" :1,
                    "2" : 5,
                    "1" : 10}
data['SCORE_TIER'] = data['PUBLIC_NOTIFICATION_TIER'].map(scoring_dict
data["YEARS_SINCE"] = 2024 -data['VIOL_FIRST_REPORTED_DATE'].str[-4:].
total_score = np.sum(data["SCORE_TIER"]) + np.max(data["YEARS_SINCE"])
```

```
/var/folders/fj/58nmvrz11g517ghvh__5bmb40000gn/T/ipykernel_20883/1254
13760.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  data['SCORE_TIER'] = data['PUBLIC_NOTIFICATION_TIER'].map(scoring_d
ict)
/var/folders/fj/58nmvrz11g517ghvh__5bmb40000gn/T/ipykernel_20883/1254
13760.py:6: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  data["YEARS_SINCE"] = 2024 -data['VIOL_FIRST_REPORTED_DATE'].str[-
4:].astype(int)
```

In [12]:
```
data
```

Out[12]:

IND	CONTAMINANT_CODE	IS_MAJOR_VIOL_IND	VIOLATION_STATUS	PUBLIC_NOTIFICATION_TIE
N	7500	NaN	Unaddressed	
N	7500	NaN	Unaddressed	
N	7500	NaN	Unaddressed	
N	7500	NaN	Unaddressed	
N	7500	NaN	Unaddressed	
N	7500	NaN	Unaddressed	

In [13]:
```
total_score
```

Out[13]: 21

In [14]:
```
def score_PWS(PWSID):
    data = UA_VIOL[UA_VIOL["PWSID"] == PWSID]
    scoring_dict = {"3" :1,
                    "2" : 5,
                    "1" : 10}
    data['SCORE_TIER'] = data['PUBLIC_NOTIFICATION_TIER'].map(scoring_
    data["YEARS_SINCE"] = 2024 -data['VIOL_FIRST_REPORTED_DATE'].str[-
    total_score = np.sum(data["SCORE_TIER"]) + np.max(data["YEARS_SINC
    return PWSID, total_score
```

In [15]:
```
score_dict = {}
for i in UA_PWSID:
    PW, SC = score_PWS(i)
    score_dict[PW] = SC
```

```
See the caveats in the documentation: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  data['SCORE_TIER'] = data['PUBLIC_NOTIFICATION_TIER'].map(scoring_d
ict)
/var/folders/fj/58nmvrz11g517ghvh__5bmb40000gn/T/ipykernel_20883/2908
987832.py:7: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  data["YEARS_SINCE"] = 2024 -data['VIOL_FIRST_REPORTED_DATE'].str[-
4:].astype(int)
/var/folders/fj/58nmvrz11g517ghvh__5bmb40000gn/T/ipykernel_20883/2908
987832.py:6: SettingWithCopyWarning:
```

In [32]:
```
Score_DF = pd.DataFrame({"PWSID" :unique_pwsid_list, "SCORE": [0]*len(
```

In [33]:
```
Score_DF["SCORE"] = Score_DF["PWSID"].map(score_dict)
```

In [35]:
```
Score_DF['SCORE'] = Score_DF['SCORE'].fillna(0)
```

In [36]:
```
PWS_data = pd.read_csv("/Volumes/T7/Water Project/SDWA_latest_download
                       usecols = ["PWSID", "PWS_NAME", "POPULATION_SERV
```

In [37]:
```
Geo_Data = pd.read_csv("/Volumes/T7/Water Project/SDWA_latest_download
                       usecols = ["PWSID", "AREA_TYPE_CODE", "STATE_SER
```

In [38]:
```
SCORED  = pd.merge(PWS_data, Score_DF, on= "PWSID", how= 'right')
```

In [39]:
```
SCORED_GEOS = pd.merge(SCORED, Geo_Data, on= "PWSID", how= 'left')
```

In [40]:
```
#Score_DF.to_csv("/Volumes/T7/Water Project/Scored_PWSID.csv")
SCORED_GEOS
```

3	CA3800848	SKATELAND	30.0	0.0	CN
4	ND1811323	LARSONS DRIVE INN	30.0	0.0	CN
...	...	...	...	...	...
373935	MD1101127	MARKET BASKET	32.0	0.0	ZC
373936	TX1840052	MARY'S CREEK DAY CAMP	100.0	0.0	CN
373937	MI2068803	DOLLAR GENERAL #18891 - PLAINWELL	300.0	0.0	CN
373938	RI2389723	OAR & BLOCK ISLAND BOAT BASIN, THE	150.0	0.0	CN

In [41]:
```
SCORED_GEOS[SCORED_GEOS["PWSID"] == "NC0465143"]
```

Out[41]:

	PWSID	PWS_NAME	POPULATION_SERVED_COUNT	SCORE	AREA_TYPE_CODE	STAT
0	NC0465143	TIMBERLYNN VILLAGE	75.0	0.0	CN	
1	NC0465143	TIMBERLYNN VILLAGE	75.0	0.0	CT	

In [42]:
```
Geo_Data[Geo_Data["PWSID"] == "NC0465143"]
```

Out[42]:

	PWSID	AREA_TYPE_CODE	STATE_SERVED	ZIP_CODE_SERVED	COUNTY_SERVED
193401	NC0465143	CN	NaN	NaN	New Hanove
193404	NC0465143	CT	NC	NaN	NaN

In [43]:
```
SCORED_GEOS.to_csv("/Volumes/T7/Water Project/Scored_PWSID_Final.csv")
```