In [1]:
```
# To start, I got all of the libraries I am going to use for the project and imported my datasets and checked to make sure
# everything was there and imported properly. I also went ahead and converted a date column to datetime to help later.

import pandas as pd
import numpy as np
import seaborn as sb
```

In [2]:
```
wqdf = pd.read_csv(r'C:\Users\mccut\OneDrive\Desktop\Drinking_Water_Quality_Distribution_Monitoring_Data.csv')
wqdf['Sample Date'] = wqdf['Sample Date'].astype('datetime64[ns]')
wqdf.head()
```

```
C:\Users\mccut\anaconda3\lib\site-packages\IPython\core\interactiveshell.py:3444: DtypeWarning: Columns (7) have mixed type
s.Specify dtype option on import or set low_memory=False.
  exec(code_obj, self.user_global_ns, self.user_ns)
```

Out[2]:

	Sample Number	Sample Date	Sample Time	Sample Site	Sample class	Residual Free Chlorine (mg/L)	Turbidity (NTU)	Fluoride (mg/L)	Coliform (Quanti-Tray) (MPN /100mL)	E.coli(Quanti-Tray) (MPN/100mL)
0	202120243	2021-07-01	10:31	23650	Compliance	0.22	0.84	NaN	<1	<1
1	202120244	2021-07-01	09:54	29550	Compliance	0.69	0.81	NaN	<1	<1
2	202120245	2021-07-01	07:52	50200	Operational	0.55	0.77	NaN	<1	<1
3	202120246	2021-07-01	08:12	50250	Compliance	0.87	0.81	NaN	<1	<1
4	202120247	2021-07-01	08:31	50300	Operational	0.80	0.84	NaN	<1	<1

In [3]:
```
wqdf.dtypes
```

Out[3]:
```
Sample Number                          int64
Sample Date                   datetime64[ns]
Sample Time                           object
Sample Site                           object
Sample class                          object
Residual Free Chlorine (mg/L)        float64
Turbidity (NTU)                       object
Fluoride (mg/L)                       object
Coliform (Quanti-Tray) (MPN /100mL)   object
E.coli(Quanti-Tray) (MPN/100mL)       object
dtype: object
```

In [4]:
```
wcdf = pd.read_csv(r'C:\Users\mccut\OneDrive\Desktop\Water_Consumption_in_the_City_of_New_York.csv')
wcdf.head()
```

Out[4]:

	Year	New York City Population	NYC Consumption(Million gallons per day)	Per Capita(Gallons per person per day)
0	1979	7,102,100	1,512	213
1	1980	7,071,639	1,506	213
2	1981	7,089,241	1,309	185
3	1982	7,109,105	1,382	194
4	1983	7,181,224	1,424	198

In [5]:
```
wcdf.dtypes
```

Out[5]:
```
Year                                    int64
New York City Population               object
NYC Consumption(Million gallons per day)    object
Per Capita(Gallons per person per day)      int64
dtype: object
```

In [6]:
```
# I am going to start cleaning with the water consumption dataframe, as it should definitely be the easier of the two and
# won't take as long. Since the water quality dataset only starts from the year 2015, there is no point to keep any of the
# data before that in the years/rows. The next part I would like to drop is the per capita gallons, as I do not really think
# it pertains to the data I want and would just be in the way when visualizing the data.

wcdf.drop(wcdf.index[0:36], inplace=True)
wcdf.drop(['Per Capita(Gallons per person per day)'], axis=1, inplace=True)
wcdf.dropna()
wcdf.head()
```

Out[6]:

	Year	New York City Population	NYC Consumption(Million gallons per day)
36	2015	8,736,703	1,009
37	2016	8,794,605	1,002
38	2017	8,815,448	990.2
39	2018	8,826,472	1,008
40	2019	8,824,887	987.4

In [7]:
```
# The next dataset has more cleaning that needs to be done, and from the head at the beginning, there were three columns I
# wanted to check: flouride, ecoli, and coliform. These seemed to be plagued with duplicates and null values.

wqdf.isna().sum()
```

Out[7]:
```
Sample Number                          0
Sample Date                            0
Sample Time                            0
Sample Site                            0
Sample class                           0
Residual Free Chlorine (mg/L)          2
Turbidity (NTU)                        1
Fluoride (mg/L)                   103160
Coliform (Quanti-Tray) (MPN /100mL)    60
E.coli(Quanti-Tray) (MPN/100mL)        60
dtype: int64
```

In [8]:
```
wqdf['Coliform (Quanti-Tray) (MPN /100mL)'].duplicated().sum()
```

Out[8]: 118627

In [9]:
```
wqdf['E.coli(Quanti-Tray) (MPN/100mL)'].duplicated().sum()
```

Out[9]: 118670

In [10]:
```
wqdf.tail()
```

Out[10]:

	Sample Number	Sample Date	Sample Time	Sample Site	Sample class	Residual Free Chlorine (mg/L)	Turbidity (NTU)	Fluoride (mg/L)	Coliform (Quanti-Tray) (MPN /100mL)	E.coli(Quanti-Tray) (MPN/100mL)
118669	202228015	2022-09-30	12:00	32750	Compliance	0.37	0.54	NaN	<1	<1
118670	202228017	2022-09-30	11:21	33850	Compliance	0.39	0.57	NaN	<1	<1
118671	202228018	2022-09-30	08:05	3SC26	Operational	0.82	0.55	NaN	<1	<1
118672	202228045	2022-09-29	10:31	35350	Compliance	0.10	0.52	NaN	<1	<1
118673	202228133	2022-09-30	09:58	33150	Compliance	0.53	0.51	NaN	<1	<1

In [11]:
```
# The only rows on this one I care about for the
# data are the sample dates and the various items they tested for. I also want to exclude the flouride row, as around 87% of
# it is entirely null values, and it doesn't relate to the data/topic I am going for as a ton of everyday food and such
# has flouride in it, so it showing up in water is not going to really have any negative effects.

wqdf.drop(['Sample Number', 'Sample Time', 'Sample Site', 'Sample class', 'Fluoride (mg/L)'], axis=1, inplace=True)
wqdf.dropna()
wqdf.head()
```

Out[11]:

	Sample Date	Residual Free Chlorine (mg/L)	Turbidity (NTU)	Coliform (Quanti-Tray) (MPN /100mL)	E.coli(Quanti-Tray) (MPN/100mL)
0	2021-07-01	0.22	0.84	<1	<1
1	2021-07-01	0.69	0.81	<1	<1
2	2021-07-01	0.55	0.77	<1	<1
3	2021-07-01	0.87	0.81	<1	<1
4	2021-07-01	0.80	0.84	<1	<1

In [12]:
```
# The next big problem with this dataset is the duplicates. I gave the tail above to show how many datapoints are in this
# set, and with only 3 being not a duplicate ecoli and 46 for coliform, they do not feel like they would be too relevant.
# This is due to the sheer scale of the data and the various points that each have unique data or are at least linked to a
# unique date making those feel like they would be irrelevant. I personally think the graphs would be useless in my data
# visualization, so I am going to drop these as well. Within the data dictionary that this set also provided, it said that
# the "<1" value in them meant it was either 0 or not detected at all, backing up the fact this data is not relevant and can
# be safely dropped.

wqdf.drop(['Coliform (Quanti-Tray) (MPN /100mL)', 'E.coli(Quanti-Tray) (MPN/100mL)'], axis=1, inplace=True)
wqdf.head()
```

Out[12]:

	Sample Date	Residual Free Chlorine (mg/L)	Turbidity (NTU)
0	2021-07-01	0.22	0.84
1	2021-07-01	0.69	0.81
2	2021-07-01	0.55	0.77
3	2021-07-01	0.87	0.81
4	2021-07-01	0.80	0.84

In [13]:
```
# The last big cleaning with this dataset will be the date, which I have to get to match the consumption dataset via
# converting it to a year. This can be done relatively easily, as I have converted the sample date to a datetime and plan to
# extract the year and create a new column in order for them to be able to join effectively.

wqdf['Year'] = wqdf['Sample Date'].dt.year
wqdf.head()
```

Out[13]:

	Sample Date	Residual Free Chlorine (mg/L)	Turbidity (NTU)	Year
0	2021-07-01	0.22	0.84	2021
1	2021-07-01	0.69	0.81	2021
2	2021-07-01	0.55	0.77	2021
3	2021-07-01	0.87	0.81	2021
4	2021-07-01	0.80	0.84	2021

In [14]:
```
# After cleaning and creating a merging point for both datasets, I went ahead and merged them into a new dataframe.

nycwaterdf = pd.merge(wcdf, wqdf, on='Year')
nycwaterdf
```

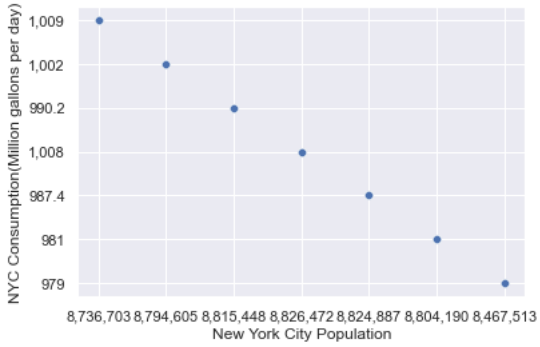
Out[14]:

	Year	New York City Population	NYC Consumption(Million gallons per day)	Sample Date	Residual Free Chlorine (mg/L)	Turbidity (NTU)
0	2015	8,736,703	1,009	2015-01-02	0.65	0.87
1	2015	8,736,703	1,009	2015-01-01	0.58	0.96
2	2015	8,736,703	1,009	2015-01-01	0.71	0.94
3	2015	8,736,703	1,009	2015-01-01	0.79	0.93
4	2015	8,736,703	1,009	2015-01-01	0.77	0.93
...	...	...	...	...	...	...
108497	2021	8,467,513	979	2021-06-30	0.39	0.83
108498	2021	8,467,513	979	2021-06-30	0.70	0.88
108499	2021	8,467,513	979	2021-06-30	0.62	0.87
108500	2021	8,467,513	979	2021-06-30	1.06	0.85
108501	2021	8,467,513	979	2021-06-30	0.90	0.78

108502 rows × 6 columns

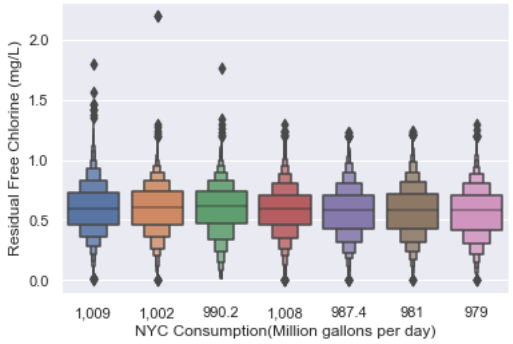
In [15]:
```
# I would like to preface that these graphs might look a bit funky due to the amount of datapoints I have in the dataframe.

sb.set(style="darkgrid")
sb.scatterplot(x='New York City Population', y='NYC Consumption(Million gallons per day)', data=nycwaterdf)
```

Out[15]: <AxesSubplot:xlabel='New York City Population', ylabel='NYC Consumption(Million gallons per day)'>



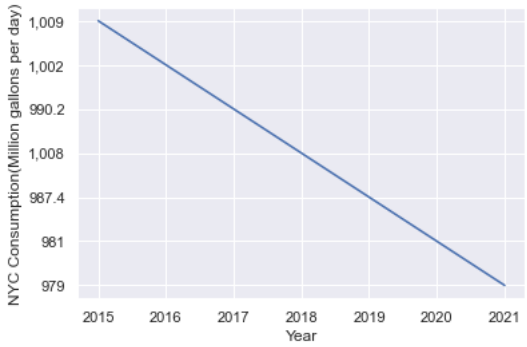
In [16]:
```
sb.boxenplot(x='NYC Consumption(Million gallons per day)', y='Residual Free Chlorine (mg/L)', data=nycwaterdf)
```

Out[16]: <AxesSubplot:xlabel='NYC Consumption(Million gallons per day)', ylabel='Residual Free Chlorine (mg/L)'>



In [17]:
```
sb.lineplot(x='Year', y='NYC Consumption(Million gallons per day)', data=nycwaterdf)
```

Out[17]: <AxesSubplot:xlabel='Year', ylabel='NYC Consumption(Million gallons per day)'>



In [18]:
```
sb.displot(nycwaterdf, x='Residual Free Chlorine (mg/L)')
```

Out[18]:
```
<seaborn.axisgrid.FacetGrid at 0x206a7b9a2e0>
```