Data Cleaning and Preparation

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from IPython.core.display import HTML
import geopandas as gpd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_absolute_error

%matplotlib inline
sns.set_style('darkgrid')
plt.rcParams['font.size'] = 14
plt.rcParams['figure.figsize'] = (17, 5)
plt.rcParams['figure.facecolor'] = '#00000000'

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
```

```
df1 = pd.read_csv('/content/absolute-number-of-deaths-from-ambient-particulate-air-pollution.csv')
df2 = pd.read_csv('/content/death-rate-from-air-pollution-per-100000.csv')
df3 = pd.read_csv('/content/death-rates-from-air-pollution.csv')
df4 = pd.read_csv('/content/share-deaths-indoor-pollution.csv')
df5 = pd.read_csv('/content/outdoor-pollution-death-rate.csv')
```

```
# Combine datasets for analysis
datasets = [df1, df2, df3, df4, df5]
df_combined = pd.concat(datasets, axis=0, ignore_index=True)
```

```
for i, df in enumerate(datasets, 1):
    print(f"Dataset {i}:")
    print(df.info())
    print(df.isnull().sum())
    print("-" * 20)
```

```
Dataset 1:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6840 entries, 0 to 6839
Data columns (total 31 columns):
 #   Column                                                                                                  Non-Null Count
---  ------                                                                                                  --------------
 0   Entity                                                                                                  6840 non-null
 1   Code                                                                                                    6150 non-null
 2   Year                                                                                                    6840 non-null
 3   Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: All Ages (Number)    6840 non-null
 4   Deaths - Cause: All causes - Risk: High systolic blood pressure - Sex: Both - Age: All Ages (Number)    6840 non-null
 5   Deaths - Cause: All causes - Risk: Diet high in sodium - Sex: Both - Age: All Ages (Number)              6840 non-null
 6   Deaths - Cause: All causes - Risk: Diet low in whole grains - Sex: Both - Age: All Ages (Number)         6840 non-null
 7   Deaths - Cause: All causes - Risk: Alcohol use - Sex: Both - Age: All Ages (Number)                      6840 non-null
 8   Deaths - Cause: All causes - Risk: Diet low in fruits - Sex: Both - Age: All Ages (Number)               6840 non-null
 9   Deaths - Cause: All causes - Risk: Unsafe water source - Sex: Both - Age: All Ages (Number)             6840 non-null
 10  Deaths - Cause: All causes - Risk: Secondhand smoke - Sex: Both - Age: All Ages (Number)                6840 non-null
 11  Deaths - Cause: All causes - Risk: Low birth weight - Sex: Both - Age: All Ages (Number)                6840 non-null
 12  Deaths - Cause: All causes - Risk: Child wasting - Sex: Both - Age: All Ages (Number)                   6840 non-null
 13  Deaths - Cause: All causes - Risk: Unsafe sex - Sex: Both - Age: All Ages (Number)                      6840 non-null
 14  Deaths - Cause: All causes - Risk: Diet low in nuts and seeds - Sex: Both - Age: All Ages (Number)      6840 non-null
 15  Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: All Ages (Number)  6840 non-null
 16  Deaths - Cause: All causes - Risk: Diet low in vegetables - Sex: Both - Age: All Ages (Number)         6840 non-null
 17  Deaths - Cause: All causes - Risk: Low physical activity - Sex: Both - Age: All Ages (Number)          6840 non-null
 18  Deaths - Cause: All causes - Risk: Smoking - Sex: Both - Age: All Ages (Number)                         6840 non-null
 19  Deaths - Cause: All causes - Risk: High fasting plasma glucose - Sex: Both - Age: All Ages (Number)     6840 non-null
 20  Deaths - Cause: All causes - Risk: Air pollution - Sex: Both - Age: All Ages (Number)                   6840 non-null
 21  Deaths - Cause: All causes - Risk: High body-mass index - Sex: Both - Age: All Ages (Number)            6840 non-null
 22  Deaths - Cause: All causes - Risk: Unsafe sanitation - Sex: Both - Age: All Ages (Number)               6840 non-null
 23  Deaths - Cause: All causes - Risk: No access to handwashing facility - Sex: Both - Age: All Ages (Number)  6840 non-null
 24  Deaths - Cause: All causes - Risk: Drug use - Sex: Both - Age: All Ages (Number)                        6840 non-null
```

```
 25  Deaths - Cause: All causes - Risk: Low bone mineral density - Sex: Both - Age: All Ages (Number)               6840 non-null
 26  Deaths - Cause: All causes - Risk: Vitamin A deficiency - Sex: Both - Age: All Ages (Number)                   6840 non-null
 27  Deaths - Cause: All causes - Risk: Child stunting - Sex: Both - Age: All Ages (Number)                         6840 non-null
 28  Deaths - Cause: All causes - Risk: Discontinued breastfeeding - Sex: Both - Age: All Ages (Number)            6840 non-null
 29  Deaths - Cause: All causes - Risk: Non-exclusive breastfeeding - Sex: Both - Age: All Ages (Number)           6840 non-null
 30  Deaths - Cause: All causes - Risk: Iron deficiency - Sex: Both - Age: All Ages (Number)                       6840 non-null
dtypes: int64(29), object(2)
memory usage: 1.6+ MB
None
Entity                                                                                                            0
Code                                                                                                            690
Year                                                                                                              0
Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: All Ages (Number)              0
Deaths - Cause: All causes - Risk: High systolic blood pressure - Sex: Both - Age: All Ages (Number)              0
Deaths - Cause: All causes - Risk: Diet high in sodium - Sex: Both - Age: All Ages (Number)                       0
Deaths - Cause: All causes - Risk: Diet low in whole grains - Sex: Both - Age: All Ages (Number)                  0
Deaths - Cause: All causes - Risk: Alcohol use - Sex: Both - Age: All Ages (Number)                               0
Deaths - Cause: All causes - Risk: Diet low in fruits - Sex: Both - Age: All Ages (Number)                        0
Deaths - Cause: All causes - Risk: Unsafe water source - Sex: Both - Age: All Ages (Number)                       0
Deaths - Cause: All causes - Risk: Secondhand smoke - Sex: Both - Age: All Ages (Number)                          0
Deaths - Cause: All causes - Risk: Low birth weight - Sex: Both - Age: All Ages (Number)                          0
Deaths - Cause: All causes - Risk: Child wasting - Sex: Both - Age: All Ages (Number)                             0
Deaths - Cause: All causes - Risk: Unsafe sex - Sex: Both - Age: All Ages (Number)                                0
Deaths - Cause: All causes - Risk: Diet low in nuts and seeds - Sex: Both - Age: All Ages (Number)                0
Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: All Ages (Number)  0
```

```
# Print column names for each dataset
for i, df in enumerate(datasets, 1):
    print(f"Dataset {i} - Columns: {df.columns.tolist()}")
    if 'death_rate' in df.columns:
        print(f"Column 'death_rate' found in Dataset {i}")
    else:
        print(f"Column 'death_rate' NOT found in Dataset {i}")
    print("-" * 50)
```

```
Dataset 1 - Columns: ['Entity', 'Code', 'Year', 'Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: All
Column 'death_rate' NOT found in Dataset 1
--------------------------------------------------
Dataset 2 - Columns: ['Entity', 'Code', 'Year', 'Deaths - Cause: All causes - Risk: Air pollution - Sex: Both - Age: Age-standardized (R
Column 'death_rate' NOT found in Dataset 2
--------------------------------------------------
Dataset 3 - Columns: ['Entity', 'Code', 'Year', 'Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both
Column 'death_rate' NOT found in Dataset 3
--------------------------------------------------
Dataset 4 - Columns: ['Entity', 'Code', 'Year', 'Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both
Column 'death_rate' NOT found in Dataset 4
--------------------------------------------------
Dataset 5 - Columns: ['Entity', 'Code', 'Year', 'Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: Age-
Column 'death_rate' NOT found in Dataset 5
--------------------------------------------------
```

```
# Check missing values
print("Missing values per column:")
print(df_combined.isnull().sum())
```

```
Missing values per column:
Entity                                                                                                        0
Code                                                                                                       3450
Year                                                                                                          0
Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: All Ages (Number)       27360
Deaths - Cause: All causes - Risk: High systolic blood pressure - Sex: Both - Age: All Ages (Number)       27360
Deaths - Cause: All causes - Risk: Diet high in sodium - Sex: Both - Age: All Ages (Number)                27360
Deaths - Cause: All causes - Risk: Diet low in whole grains - Sex: Both - Age: All Ages (Number)           27360
Deaths - Cause: All causes - Risk: Alcohol use - Sex: Both - Age: All Ages (Number)                        27360
Deaths - Cause: All causes - Risk: Diet low in fruits - Sex: Both - Age: All Ages (Number)                 27360
Deaths - Cause: All causes - Risk: Unsafe water source - Sex: Both - Age: All Ages (Number)               27360
Deaths - Cause: All causes - Risk: Secondhand smoke - Sex: Both - Age: All Ages (Number)                   27360
Deaths - Cause: All causes - Risk: Low birth weight - Sex: Both - Age: All Ages (Number)                   27360
Deaths - Cause: All causes - Risk: Child wasting - Sex: Both - Age: All Ages (Number)                      27360
Deaths - Cause: All causes - Risk: Unsafe sex - Sex: Both - Age: All Ages (Number)                         27360
Deaths - Cause: All causes - Risk: Diet low in nuts and seeds - Sex: Both - Age: All Ages (Number)         27360
Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: All Ages (Number)  27360
Deaths - Cause: All causes - Risk: Diet low in vegetables - Sex: Both - Age: All Ages (Number)            27360
Deaths - Cause: All causes - Risk: Low physical activity - Sex: Both - Age: All Ages (Number)             27360
Deaths - Cause: All causes - Risk: Smoking - Sex: Both - Age: All Ages (Number)                           27360
Deaths - Cause: All causes - Risk: High fasting plasma glucose - Sex: Both - Age: All Ages (Number)       27360
Deaths - Cause: All causes - Risk: Air pollution - Sex: Both - Age: All Ages (Number)                     27360
Deaths - Cause: All causes - Risk: High body-mass index - Sex: Both - Age: All Ages (Number)              27360
Deaths - Cause: All causes - Risk: Unsafe sanitation - Sex: Both - Age: All Ages (Number)                 27360
```

```
Deaths - Cause: All causes - Risk: No access to handwashing facility - Sex: Both - Age: All Ages (Number)                  27360
Deaths - Cause: All causes - Risk: Drug use - Sex: Both - Age: All Ages (Number)                                          27360
Deaths - Cause: All causes - Risk: Low bone mineral density - Sex: Both - Age: All Ages (Number)                          27360
Deaths - Cause: All causes - Risk: Vitamin A deficiency - Sex: Both - Age: All Ages (Number)                              27360
Deaths - Cause: All causes - Risk: Child stunting - Sex: Both - Age: All Ages (Number)                                    27360
Deaths - Cause: All causes - Risk: Discontinued breastfeeding - Sex: Both - Age: All Ages (Number)                        27360
Deaths - Cause: All causes - Risk: Non-exclusive breastfeeding - Sex: Both - Age: All Ages (Number)                       27360
Deaths - Cause: All causes - Risk: Iron deficiency - Sex: Both - Age: All Ages (Number)                                   27360
Deaths - Cause: All causes - Risk: Air pollution - Sex: Both - Age: Age-standardized (Rate)                               20520
Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: Age-standardized (Rate)    27360
Deaths - Cause: All causes - Risk: Ambient particulate matter pollution - Sex: Both - Age: Age-standardized (Rate)        27360
Deaths - Cause: All causes - Risk: Ambient ozone pollution - Sex: Both - Age: Age-standardized (Rate)                     27360
Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: Age-standardized (Percent) 27360
Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: Age-standardized (Rate)               27360
dtype: int64
```

```
# Impute missing values with the median for numerical columns
def impute_missing_values(df):
    for column in df.select_dtypes(include=['float64', 'int64']).columns:
        median_value = df[column].median()
        df[column].fillna(median_value, inplace=True)
    return df

# Apply the function to all datasets
datasets = [impute_missing_values(df) for df in datasets]

# Verify missing values after handling
for i, df in enumerate(datasets, 1):
    print(f"Dataset {i} - Missing values after imputation:")
    print(df.isnull().sum())
    print("-" * 20)
```

```
--------------------
taset 3 - Missing values after imputation:
city                                                                                                    0
de                                                                                                    690
ar                                                                                                      0
ths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: Age-standardized (Rate)     0
ths - Cause: All causes - Risk: Ambient particulate matter pollution - Sex: Both - Age: Age-standardized (Rate)         0
ths - Cause: All causes - Risk: Air pollution - Sex: Both - Age: Age-standardized (Rate)                                 0
ths - Cause: All causes - Risk: Ambient ozone pollution - Sex: Both - Age: Age-standardized (Rate)                       0
```

```
df[column].fillna(median_value, inplace=True)
ython-input-12-d452020527c8>:5: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignme
e behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting value

 example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].met
```

```
trdf1 = df1.rename(columns = {'Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: All Ages (Number)':'Total

apoh = trdf1[['Entity','Year','Total Deaths for Air Pollution','Total Deaths for Outdoor Air Pollution','Total Deaths for Household Air Poll

drapph = df2.rename(columns = {'Deaths - Cause: All causes - Risk: Air pollution - Sex: Both - Age: Age-standardized (Rate)':'Death Rate frc

drapmp = df3.rename(columns={'Deaths - Cause: All causes - Risk: Ambient particulate matter pollution - Sex: Both - Age: Age-standardized (R

dhapsfp = df4.rename(columns={'Deaths - Cause: All causes - Risk: Household air pollution from solid fuels - Sex: Both - Age: Age-standardiz

dfoapp = df5.rename(columns={'Deaths - Cause: All causes - Risk: Outdoor air pollution - OWID - Sex: Both - Age: Age-standardized (Rate)':'D
```

```
merap = apoh.merge(drapph, on=['Entity','Year'], how='left').drop('Code', axis=1)

merap119_tr = merap[merap.Year == 2019].sort_values('Total Deaths for Air Pollution', ascending=False).drop([6629,2129,6719,1649,6749,5579,5

merap119hd = merap119_tr[['Entity','Year','Total Deaths for Air Pollution']].reset_index().drop('index', axis=1).head(20)

merap119tl = merap119_tr[['Entity','Year','Total Deaths for Air Pollution']].reset_index().drop('index', axis=1).tail(20).sort_values('Total

merap219_tr = merap[merap.Year == 2019].sort_values('Total Deaths for Outdoor Air Pollution', ascending=False).drop([6629,2129,6719,1649,674

merap219hd = merap219_tr[['Entity','Year','Total Deaths for Outdoor Air Pollution']].reset_index().drop('index', axis=1).head(20)

merap219tl = merap219_tr[['Entity','Year','Total Deaths for Outdoor Air Pollution']].reset_index().drop('index', axis=1).tail(20).sort_value

merap319_tr = merap[merap.Year == 2019].sort_values('Total Deaths for Household Air Pollution from Solid Fuels', ascending=False).drop([6629

merap319hd = merap319_tr[['Entity','Year','Total Deaths for Household Air Pollution from Solid Fuels']].reset_index().drop('index', axis=1).

merap419_tr = merap[merap.Year == 2019].sort_values('Death Rate from Air Pollution Per 100000', ascending=False).drop([6629,2129,6719,1649,6

merap419hd = merap419_tr[['Entity','Year','Death Rate from Air Pollution Per 100000']].reset_index().drop('index', axis=1).head(20)

merap419tl = merap419_tr[['Entity','Year','Death Rate from Air Pollution Per 100000']].reset_index().drop('index', axis=1).tail(20).sort_val

merap19re = merap.loc[[1649,5579,5489,3719,5669,59,2009,3149,4229,2129]].reset_index().drop('index', axis=1)

merapwd = merap[merap.Entity == 'World'].reset_index().drop('index', axis=1)

merapdf1_tr = merap.pivot_table(values = 'Total Deaths for Air Pollution',index = ['Entity'], columns = 'Year')
merapdf1 = merapdf1_tr.drop(['African Region (WHO)','East Asia & Pacific (WB)','Eastern Mediterranean Region (WHO)','Europe & Central Asia (
merapdf1.sort_values(list(merapdf1_tr.columns),inplace=True)
merapdf1 = merapdf1.sort_index()

merapdf2_tr = merap.pivot_table(values = 'Total Deaths for Outdoor Air Pollution',index = ['Entity'], columns = 'Year')
merapdf2 = merapdf2_tr.drop(['African Region (WHO)','East Asia & Pacific (WB)','Eastern Mediterranean Region (WHO)','Europe & Central Asia (
merapdf2.sort_values(list(merapdf2_tr.columns),inplace=True)
merapdf2 = merapdf2.sort_index()

merapdf3_tr = merap.pivot_table(values = 'Total Deaths for Household Air Pollution from Solid Fuels',index = ['Entity'], columns = 'Year')
merapdf3 = merapdf3_tr.drop(['African Region (WHO)','East Asia & Pacific (WB)', 'World Bank Low Income','Eastern Mediterranean Region (WHO)'
merapdf3.sort_values(list(merapdf3_tr.columns),inplace=True)
merapdf3 = merapdf3.sort_index()
```

```
drapmp1_tr = drapmp[drapmp.Year == 2019].sort_values('Deaths Rate for Ambient Particulate Matter Pollution', ascending=False).drop([6629,212

drapmp1hd = drapmp1_tr[['Entity','Year','Deaths Rate for Ambient Particulate Matter Pollution']].reset_index().drop('index', axis=1).head(20

drapmp1tl = drapmp1_tr[['Entity','Year','Deaths Rate for Ambient Particulate Matter Pollution']].reset_index().drop('index', axis=1).tail(20

drapmp2_tr = drapmp[drapmp.Year == 2019].sort_values('Deaths Rate for Ambient Ozone Pollution', ascending=False).drop([6629,2129,6719,1649,6

drapmp2hd = drapmp2_tr[['Entity','Year','Deaths Rate for Ambient Ozone Pollution']].reset_index().drop('index', axis=1).head(20)

drapmp2tl = drapmp2_tr[['Entity','Year','Deaths Rate for Ambient Ozone Pollution']].reset_index().drop('index', axis=1).tail(20).sort_values
```

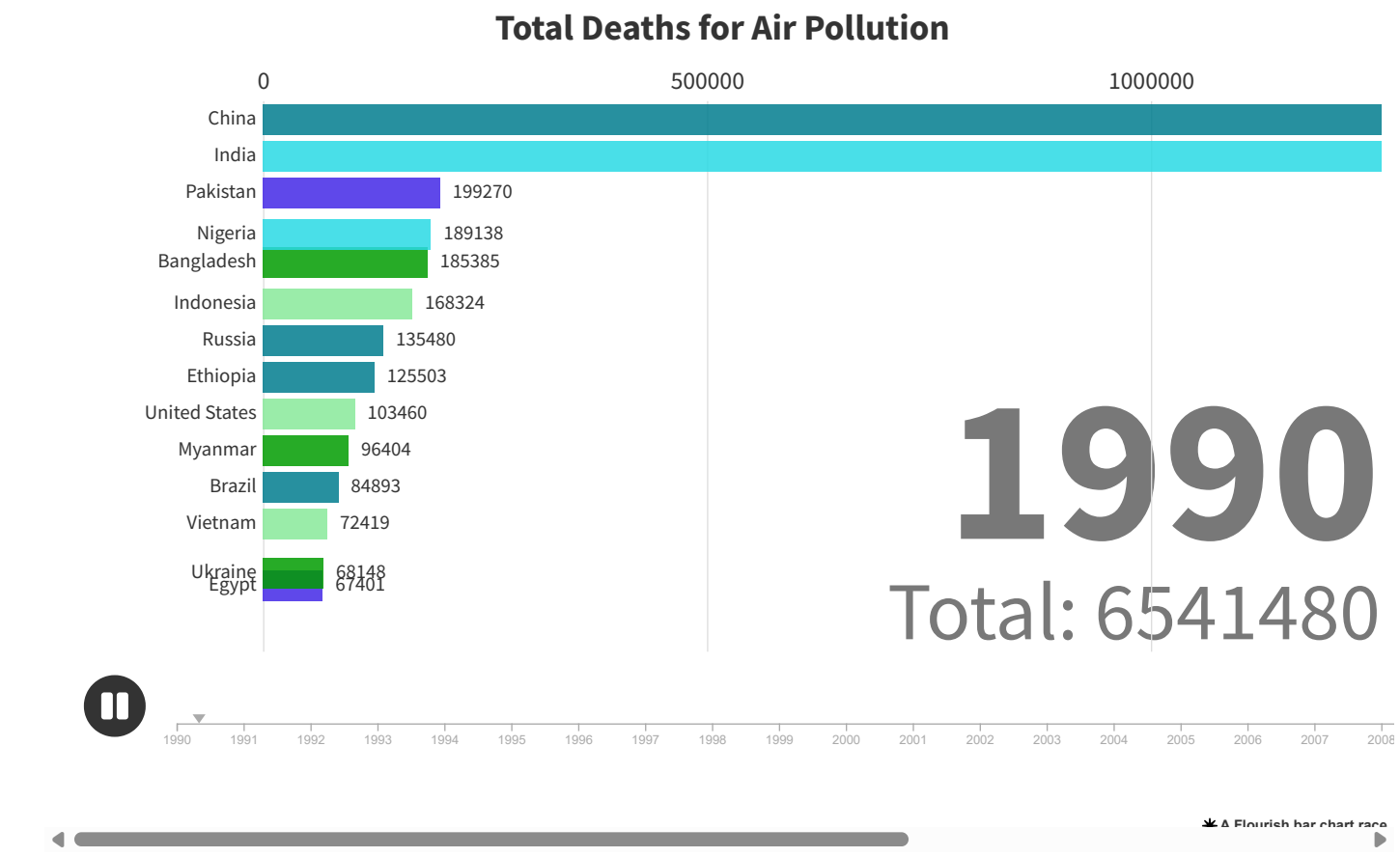
```
drapmpre = drapmp.loc[[1649,5579,5489,3719,5669,59,2009,3149,4229,2129]].reset_index().drop('index', axis=1)

drapmpwd = drapmp[drapmp.Entity == 'World'].reset_index().drop('index', axis=1)

drapmp1_tr = drapmp.pivot_table(values = 'Deaths Rate for Ambient Particulate Matter Pollution',index = ['Entity'], columns = 'Year')
drapmp1 = drapmp1_tr.drop(['African Region (WHO)','East Asia & Pacific (WB)', 'World Bank Low Income','Eastern Mediterranean Region (WHO)','
drapmp1.sort_values(list(drapmp1_tr.columns),inplace=True)
drapmp1 = drapmp1.sort_index()

drapmp2_tr = drapmp.pivot_table(values = 'Deaths Rate for Ambient Ozone Pollution',index = ['Entity'], columns = 'Year')
drapmp2 = drapmp2_tr.drop(['African Region (WHO)','East Asia & Pacific (WB)', 'World Bank Low Income','Eastern Mediterranean Region (WHO)','
drapmp2.sort_values(list(drapmp2_tr.columns),inplace=True)
drapmp2 = drapmp2.sort_index()
```

```
dhapsfp19 = dhapsfp[dhapsfp.Year == 2019].sort_values('Deaths for Household Air Pollution from Solid Fuels (Percent)', ascending=False).drop

dhapsfpre = dhapsfp.loc[[1649,5579,5489,3719,5669,59,2009,3149,4229,2129]].reset_index().drop(['index','Code'], axis=1)

dhapsfpwd = dhapsfp[dhapsfp.Entity == 'World'].reset_index().drop(['index','Code'], axis=1)
```

```
dfoapphd = dfoapp[dfoapp.Year == 2019].sort_values('Death for Outdoor Air Pollution - (Per 100K)', ascending=False).drop([6629,2129,6719,164

dfoapptl = dfoapp[dfoapp.Year == 2019].sort_values('Death for Outdoor Air Pollution - (Per 100K)', ascending=False).drop([6629,2129,6719,164

dfoappre = dfoapp.loc[[1649,5579,5489,3719,5669,59,2009,3149,4229,2129]].reset_index().drop(['index','Code'], axis=1)

dfoappwd = dfoapp[dfoapp.Entity == 'World'].reset_index().drop(['index','Code'], axis=1)
```

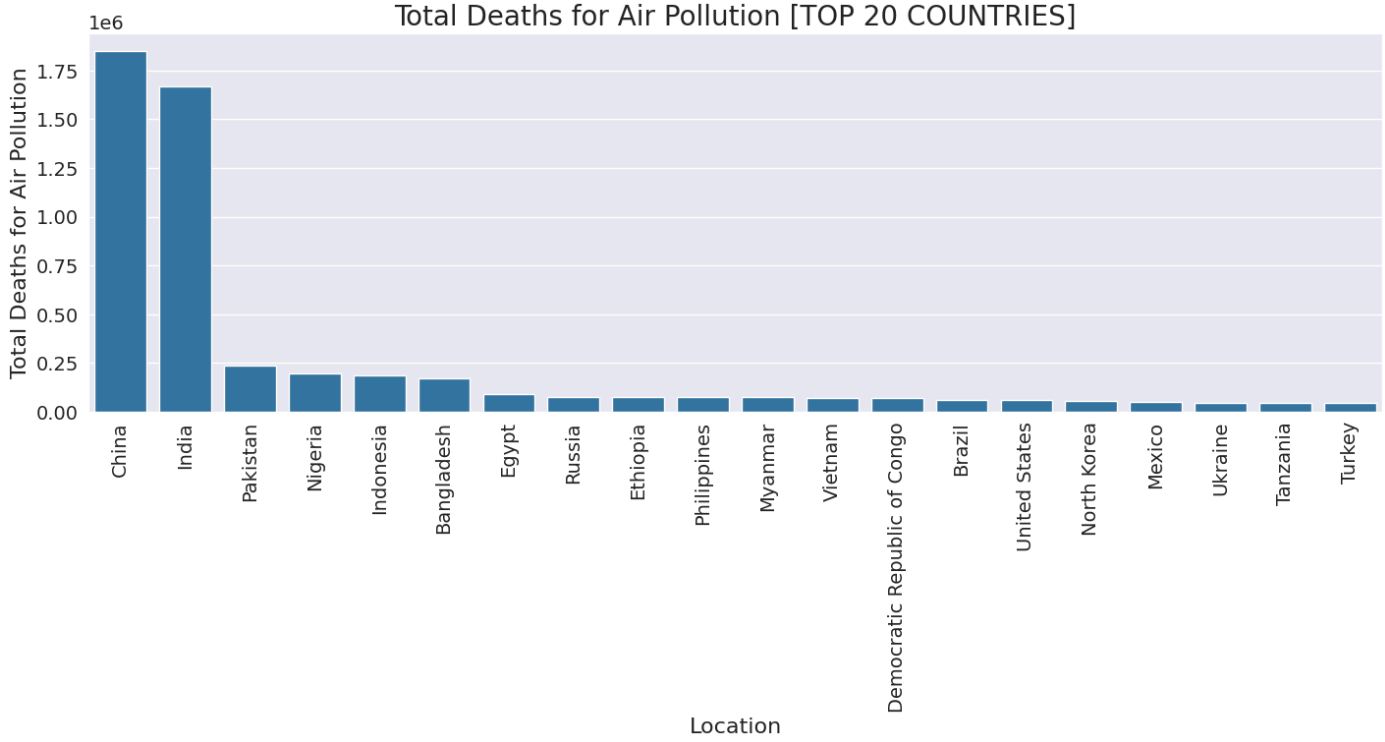
Data Visualizations

```
HTML('''<div class="flourish-embed flourish-bar-chart-race" data-src="visualisation/13185255"><script src="https://public.flourish.studio/re
```




```
sns.barplot(x='Entity', y='Total Deaths for Air Pollution', data=merap119hd)

plt.xlabel('Location', fontsize=16)
plt.ylabel('Total Deaths for Air Pollution', fontsize=16)
plt.yticks(rotation=90)
```

```
plt.xticks(rotation=90)
plt.title('Total Deaths for Air Pollution [TOP 20 COUNTRIES]', fontsize=20)

plt.show();
```

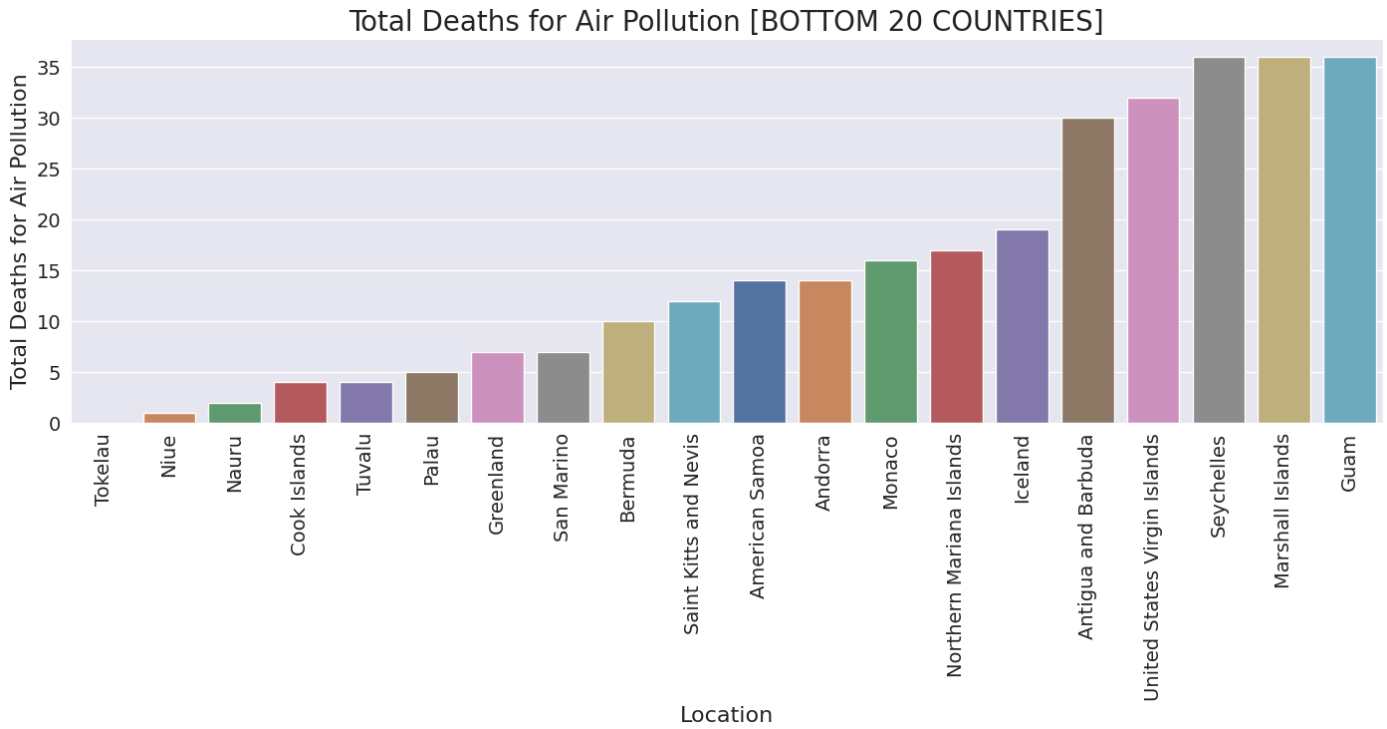



```
merap119hd
```

	Entity	Year	Total Deaths for Air Pollution
0	China	2019	1848274
1	India	2019	1667331
2	Pakistan	2019	235657
3	Nigeria	2019	197567
4	Indonesia	2019	186267
5	Bangladesh	2019	173515
6	Egypt	2019	91663
7	Russia	2019	77516
8	Ethiopia	2019	77020
9	Philippines	2019	74783
10	Myanmar	2019	74544
11	Vietnam	2019	71701
12	Democratic Republic of Congo	2019	69503
13	Brazil	2019	60915
14	United States	2019	60229
15	North Korea	2019	55154
16	Mexico	2019	48332
17	Ukraine	2019	46129
18	Tanzania	2019	45491
19	Turkey	2019	44166

```
sns.barplot(x='Entity', y='Total Deaths for Air Pollution', data=merap119tl, palette='deep')

plt.xlabel('Location', fontsize=16)
plt.ylabel('Total Deaths for Air Pollution', fontsize=16)
plt.xticks(rotation=90)
plt.title('Total Deaths for Air Pollution [BOTTOM 20 COUNTRIES]', fontsize=20)

plt.show();
```

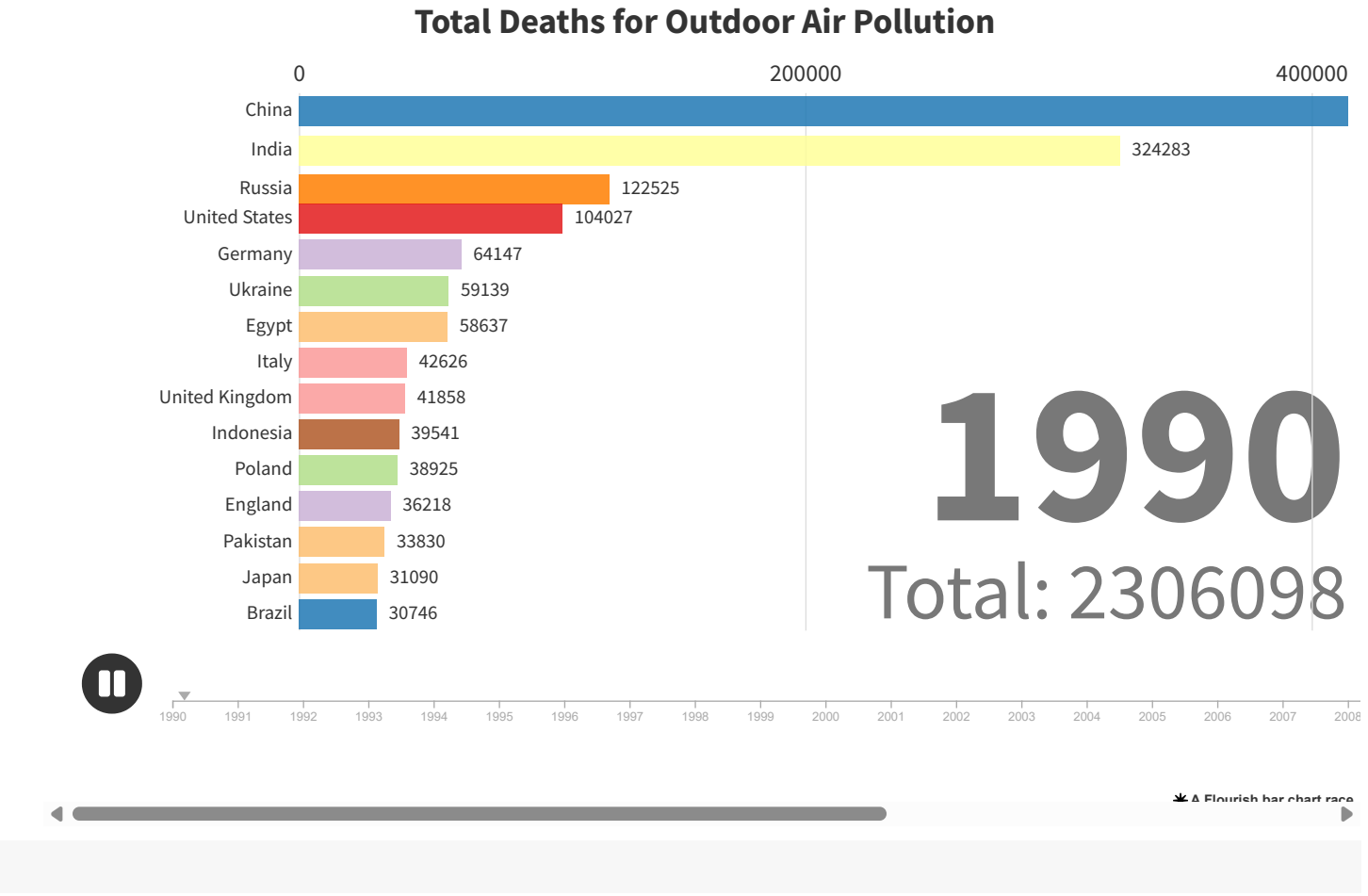
```
<ipython-input-12-76e0a3e72490>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x='Entity', y='Total Deaths for Air Pollution', data=merap119tl, palette='deep')
```



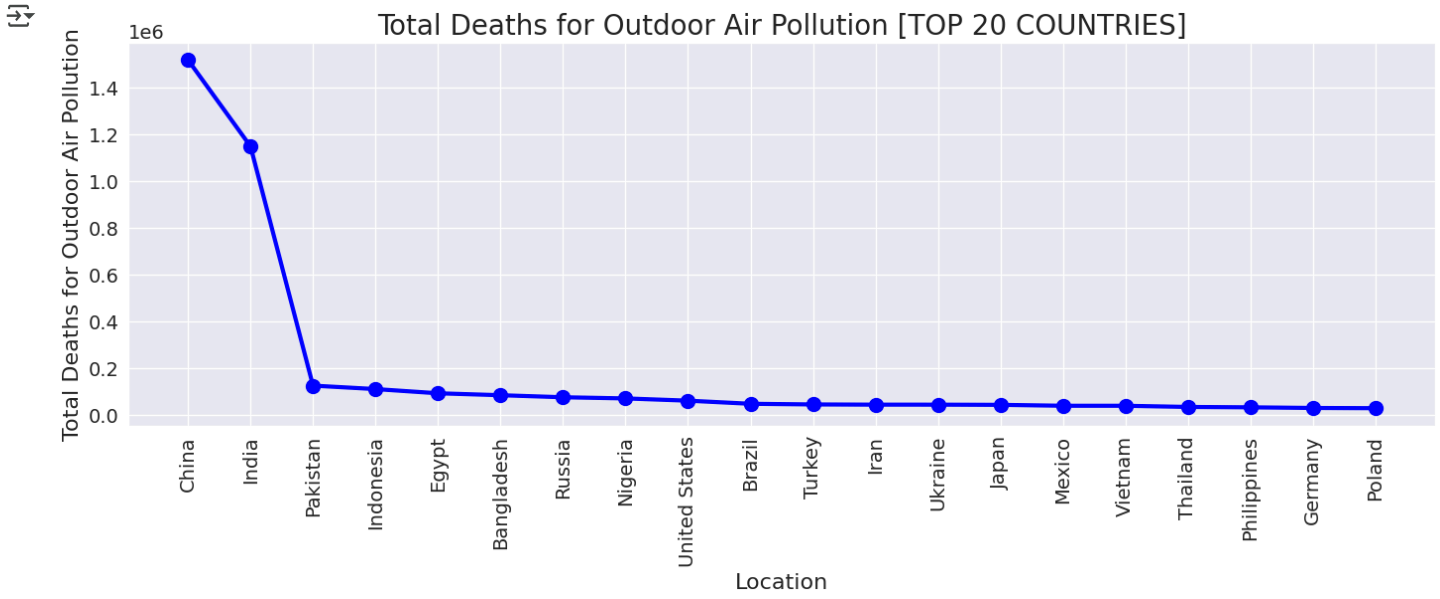

```
merap119tl
```

	Entity	Year	Total Deaths for Air Pollution
207	Tokelau	2019	0
206	Niue	2019	1
205	Nauru	2019	2
203	Cook Islands	2019	4
204	Tuvalu	2019	4
202	Palau	2019	5
201	Greenland	2019	7
200	San Marino	2019	7
199	Bermuda	2019	10
198	Saint Kitts and Nevis	2019	12
197	American Samoa	2019	14
196	Andorra	2019	14
195	Monaco	2019	16
194	Northern Mariana Islands	2019	17
193	Iceland	2019	19
192	Antigua and Barbuda	2019	30
191	United States Virgin Islands	2019	32
188	Seychelles	2019	36
189	Marshall Islands	2019	36
190	Guam	2019	36

```
HTML('''<div class="flourish-embed flourish-bar-chart-race" data-src="visualisation/13185444"><script src="https://public.flourish.studio/re
```



```
plt.plot(merap219hd['Entity'], merap219hd['Total Deaths for Outdoor Air Pollution'], 'o-b', lw=3, ms=10)

plt.xlabel('Location', fontsize=16)
plt.ylabel('Total Deaths for Outdoor Air Pollution', fontsize=16)
plt.xticks(rotation=90)
plt.title('Total Deaths for Outdoor Air Pollution [TOP 20 COUNTRIES]', fontsize=20)

plt.show();
```

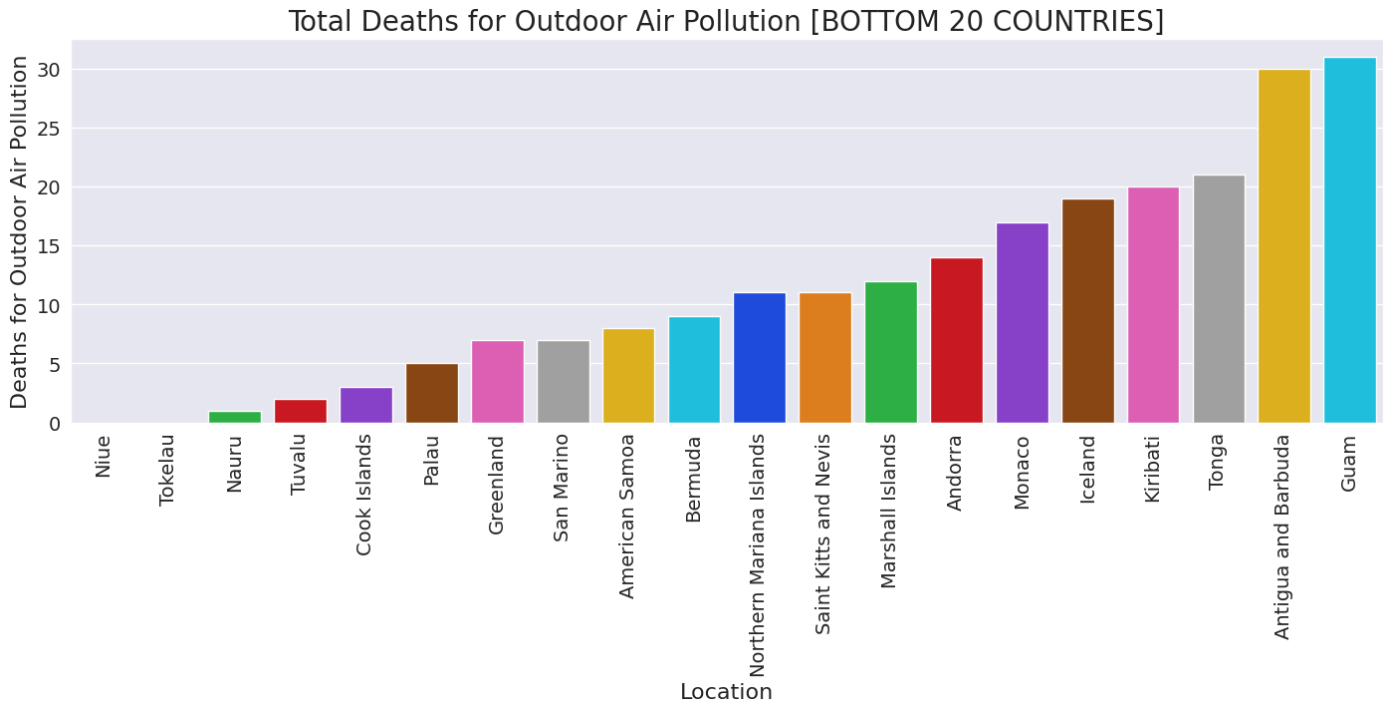



```
merap219hd
```

	Entity	Year	Total Deaths for Outdoor Air Pollution
0	China	2019	1516904
1	India	2019	1147669
2	Pakistan	2019	124913
3	Indonesia	2019	110127
4	Egypt	2019	92169
5	Bangladesh	2019	83912
6	Russia	2019	75166
7	Nigeria	2019	70181
8	United States	2019	60572
9	Brazil	2019	47257
10	Turkey	2019	44457
11	Iran	2019	43536
12	Ukraine	2019	43458
13	Japan	2019	42742
14	Mexico	2019	38854
15	Vietnam	2019	38776
16	Thailand	2019	33772
17	Philippines	2019	32138
18	Germany	2019	29401
19	Poland	2019	28426

```
sns.barplot(x='Entity', y='Total Deaths for Outdoor Air Pollution', data=merap219tl, palette='bright')

plt.xlabel('Location', fontsize=16)
plt.ylabel('Deaths for Outdoor Air Pollution', fontsize=16)
plt.xticks(rotation=90)
plt.title('Total Deaths for Outdoor Air Pollution [BOTTOM 20 COUNTRIES]', fontsize=20)

plt.show();
```

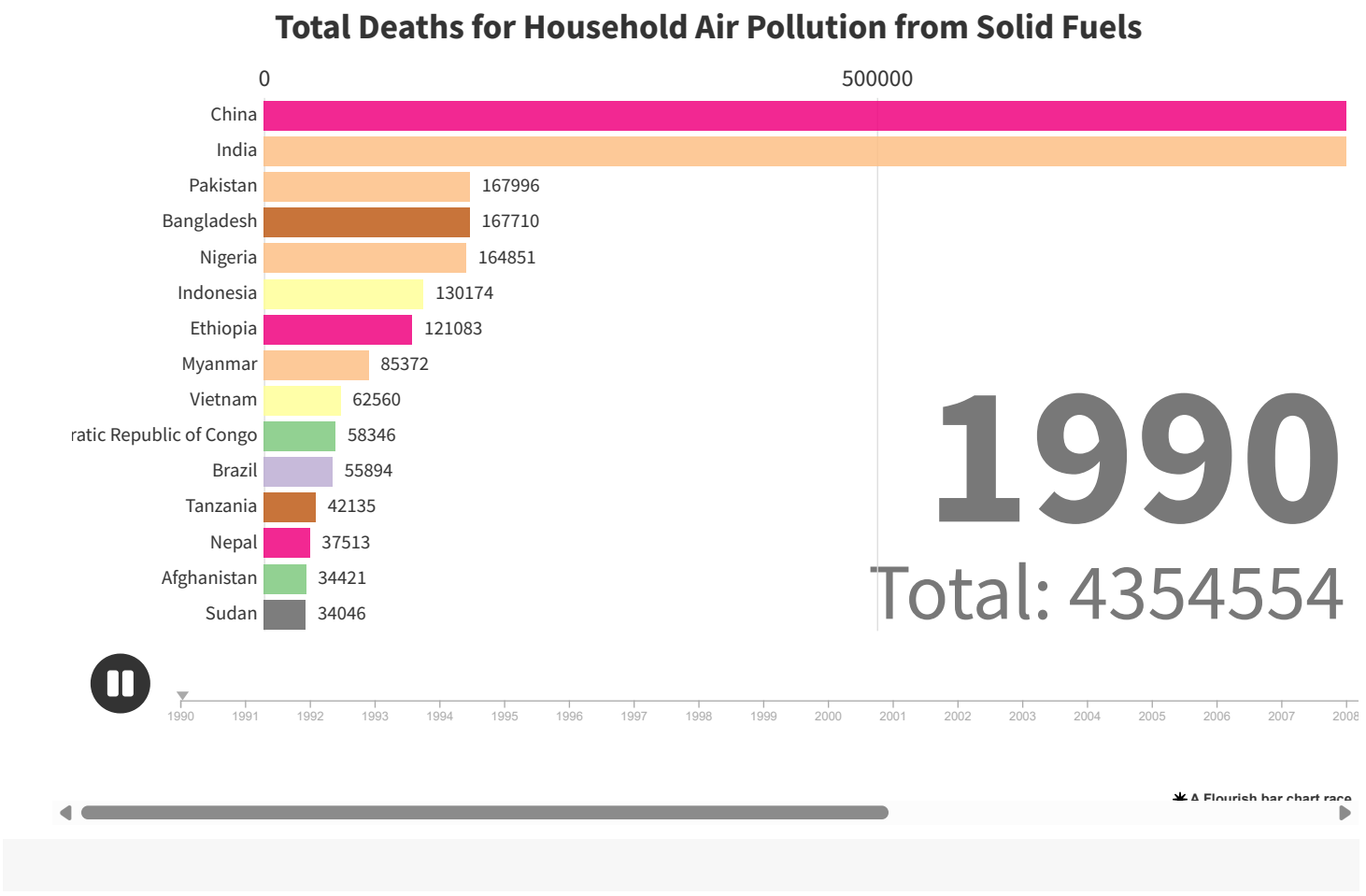
```
<ipython-input-17-98fb721ae7c6>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x='Entity', y='Total Deaths for Outdoor Air Pollution', data=merap219tl, palette='bright')
```



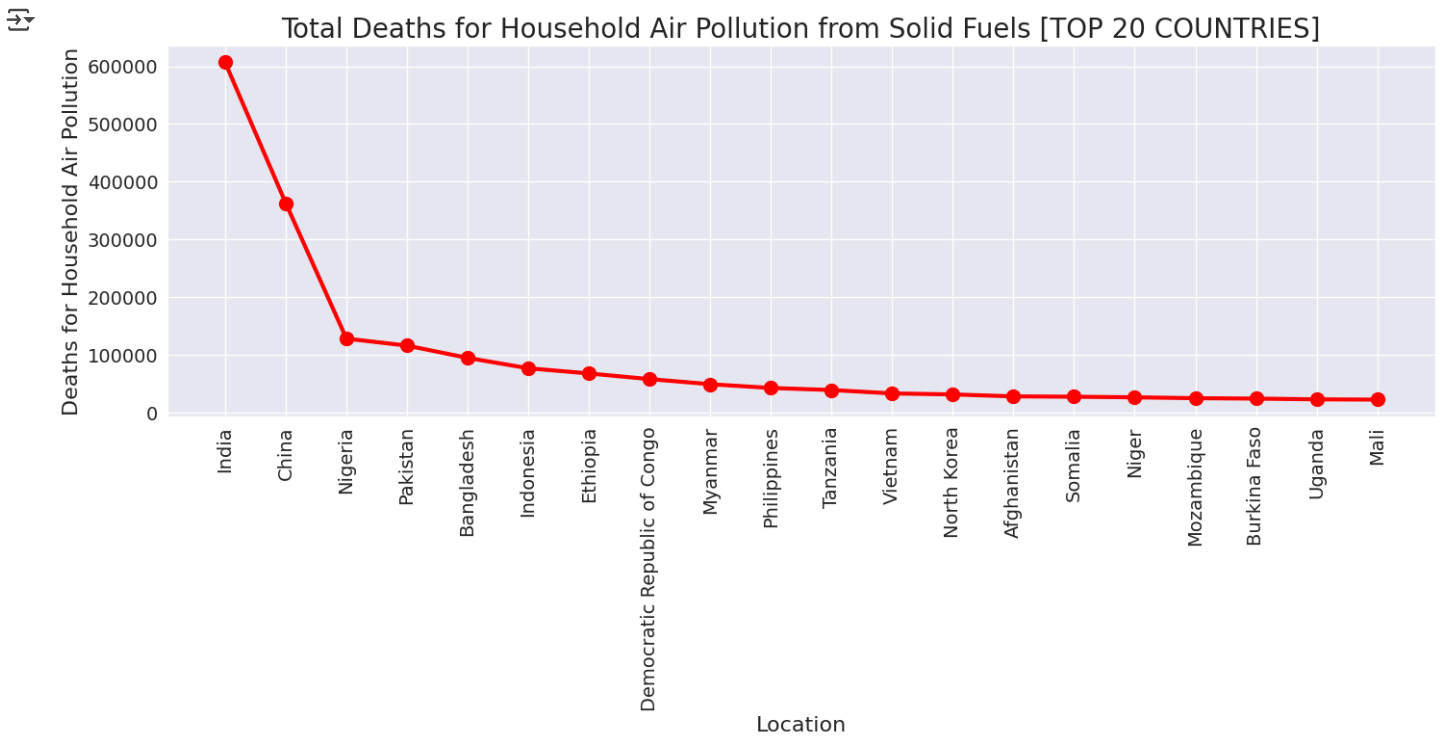

```
merap219tl
```

	Entity	Year	Total Deaths for Outdoor Air Pollution
206	Niue	2019	0
207	Tokelau	2019	0
205	Nauru	2019	1
204	Tuvalu	2019	2
203	Cook Islands	2019	3
202	Palau	2019	5
201	Greenland	2019	7
200	San Marino	2019	7
199	American Samoa	2019	8
198	Bermuda	2019	9
197	Northern Mariana Islands	2019	11
196	Saint Kitts and Nevis	2019	11
195	Marshall Islands	2019	12
194	Andorra	2019	14
193	Monaco	2019	17
192	Iceland	2019	19
191	Kiribati	2019	20
190	Tonga	2019	21
189	Antigua and Barbuda	2019	30
188	Guam	2019	31

```
HTML('''<div class="flourish-embed flourish-bar-chart-race" data-src="visualisation/13185561"><script src="https://public.flourish.studio/re
```



```
plt.plot(merap319hd['Entity'], merap319hd['Total Deaths for Household Air Pollution from Solid Fuels'], 'o-r', lw=3, ms=10)

plt.xlabel('Location', fontsize=16)
plt.ylabel('Deaths for Household Air Pollution', fontsize=16)
plt.xticks(rotation=90)
plt.title('Total Deaths for Household Air Pollution from Solid Fuels [TOP 20 COUNTRIES]', fontsize=20)

plt.show();
```

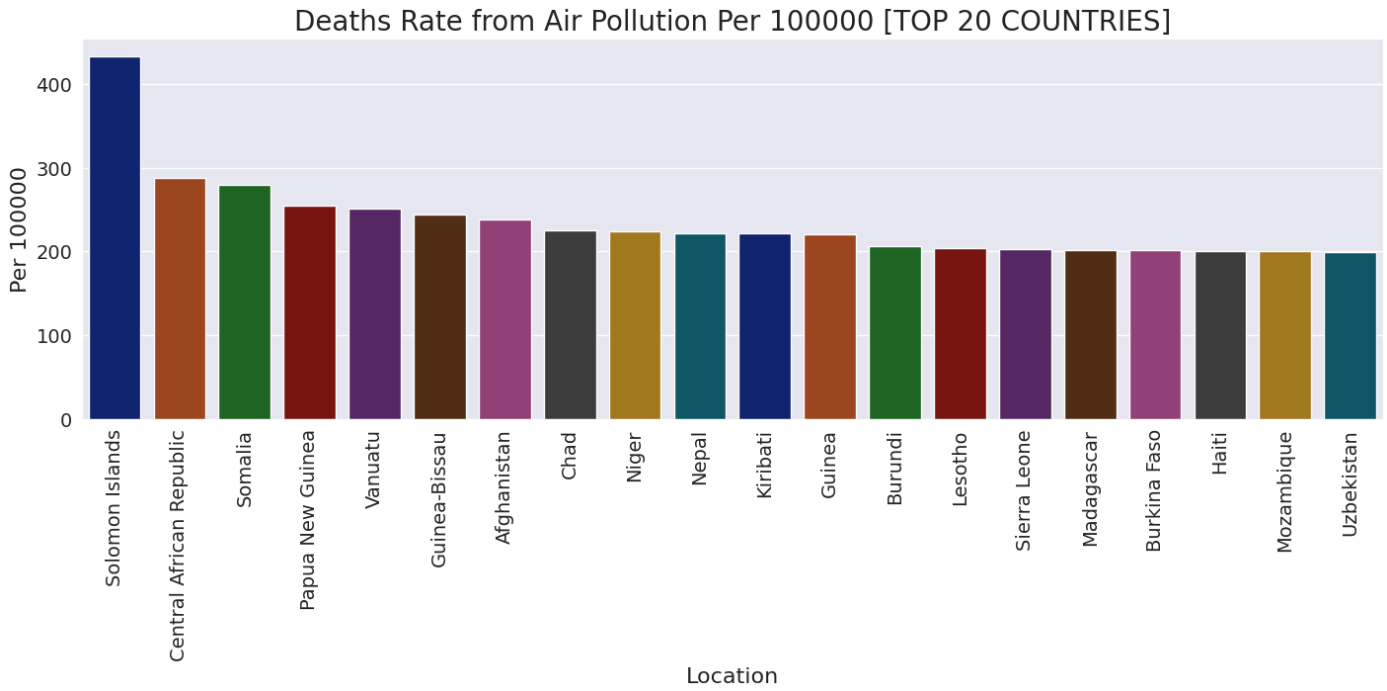



```
merap319hd
```

	Entity	Year	Total Deaths for Household Air Pollution from Solid Fuels
0	India	2019	606890
1	China	2019	363029
2	Nigeria	2019	128259
3	Pakistan	2019	116090
4	Bangladesh	2019	94789
5	Indonesia	2019	76867
6	Ethiopia	2019	67827
7	Democratic Republic of Congo	2019	58038
8	Myanmar	2019	49223
9	Philippines	2019	42675
10	Tanzania	2019	39165
11	Vietnam	2019	33247
12	North Korea	2019	31515
13	Afghanistan	2019	28168
14	Somalia	2019	27553
15	Niger	2019	26507
16	Mozambique	2019	25017
17	Burkina Faso	2019	24303
18	Uganda	2019	23001
19	Mali	2019	22580

```
sns.barplot(x='Entity', y='Death Rate from Air Pollution Per 100000', data=merap419hd, palette='dark')

plt.xlabel('Location', fontsize=16)
plt.ylabel('Per 100000', fontsize=16)
plt.xticks(rotation=90)

plt.title('Deaths Rate from Air Pollution Per 100000 [TOP 20 COUNTRIES]', fontsize=20)

plt.show();
```

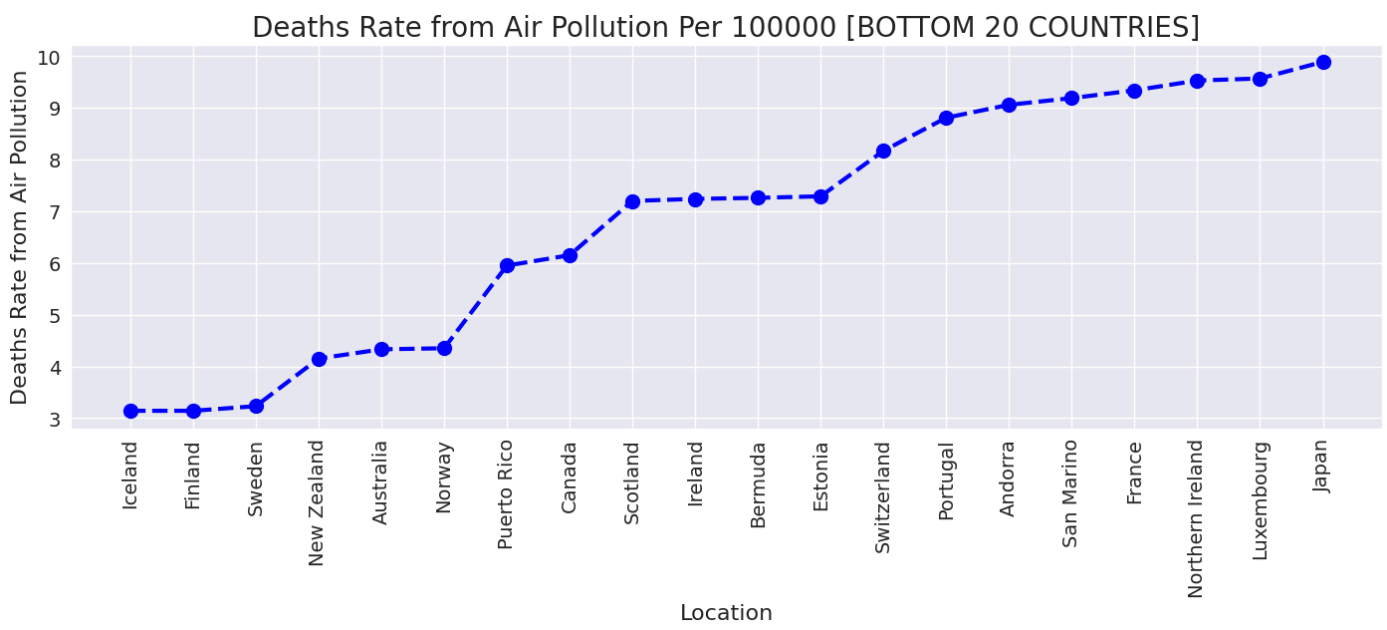
```
<ipython-input-23-4388bce0f2e7>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `legend

  sns.barplot(x='Entity', y='Death Rate from Air Pollution Per 100000', data=merap419hd, palette='dark')
```




```
merap419hd
```

	Entity	Year	Death Rate from Air Pollution Per 100000
0	Solomon Islands	2019	432.93
1	Central African Republic	2019	287.26
2	Somalia	2019	280.00
3	Papua New Guinea	2019	254.16
4	Vanuatu	2019	250.75
5	Guinea-Bissau	2019	243.93
6	Afghanistan	2019	238.33
7	Chad	2019	224.69
8	Niger	2019	223.49
9	Nepal	2019	222.00
10	Kiribati	2019	221.92
11	Guinea	2019	219.99
12	Burundi	2019	206.00
13	Lesotho	2019	204.11
14	Sierra Leone	2019	203.13
15	Madagascar	2019	201.04
16	Burkina Faso	2019	201.03
17	Haiti	2019	200.68
18	Mozambique	2019	200.13
19	Uzbekistan	2019	199.39

```
plt.plot(merap419tl['Entity'], merap419tl['Death Rate from Air Pollution Per 100000'], 'o--b', lw=3, ms=10)

plt.xlabel('Location', fontsize=16)
plt.ylabel('Deaths Rate from Air Pollution', fontsize=16)
plt.xticks(rotation=90)

plt.title('Deaths Rate from Air Pollution Per 100000 [BOTTOM 20 COUNTRIES]', fontsize=20)

plt.show();
```




```
merap419tl
```

	Entity	Year	Death Rate from Air Pollution Per 100000
206	Iceland	2019	3.14
207	Finland	2019	3.14
205	Sweden	2019	3.23
204	New Zealand	2019	4.15
203	Australia	2019	4.33
202	Norway	2019	4.35
201	Puerto Rico	2019	5.95
200	Canada	2019	6.15
199	Scotland	2019	7.20
198	Ireland	2019	7.24
197	Bermuda	2019	7.26
196	Estonia	2019	7.29
195	Switzerland	2019	8.17
194	Portugal	2019	8.81
193	Andorra	2019	9.06
192	San Marino	2019	9.19
191	France	2019	9.34
190	Northern Ireland	2019	9.53
189	Luxembourg	2019	9.57
188	Japan	2019	9.89

```
plt.scatter(merap19re['Entity'], merap19re['Total Deaths for Air Pollution'], s=150)
plt.scatter(merap19re['Entity'], merap19re['Total Deaths for Outdoor Air Pollution'], s=150)
plt.scatter(merap19re['Entity'], merap19re['Total Deaths for Household Air Pollution from Solid Fuels'], s=150)
```

```
plt.xlabel('Region', fontsize=16)
plt.ylabel('Total Deaths', fontsize=16)
plt.xticks(rotation=90)
plt.title('Total Deaths Over Time - REGION', fontsize=20)

plt.legend(['Deaths for Air Pollution', 'Deaths for Outdoor Air Pollution', 'Deaths for Household Air Pollution']
```