Introduction

In [ ]:
```
Data Credit: ourworldindata.org

The COVID-19 pandemic, caused by the novel coronavirus SARS-CoV-2, has had a profound and lasting impact on global health, economies, and societies.
From its emergence in late 2019, the virus rapidly spread across the world, leading to widespread illness, overwhelmed healthcare systems, and tragically, millions of deaths.
Understanding the dynamics of the pandemic, including its spread, severity, and the effectiveness of interventions, has been crucial in guiding public health responses
and mitigating its impact.

Analyzing COVID-19 data has played a pivotal role in this understanding. By collecting and examining data on cases, deaths, hospitalizations, and other relevant metrics,
researchers and policymakers have been able to track the course of the pandemic, identify trends and patterns, and assess the effectiveness of various strategies aimed at
controlling the virus. This data-driven approach has been essential in informing decisions about lockdowns, mask mandates, vaccination campaigns, and other measures
taken to protect public health.

In the following analysis, we will delve into various aspects of COVID-19 data, exploring trends in case numbers, mortality rates, and vaccination coverage across different
regions and populations. By examining this data, we aim to gain insights into the factors that have influenced the pandemic's trajectory, identify disparities in its impact,
and draw lessons that can help us better prepare for future health crises.

PS: ALL ANALYSIS ATTEMPTED OR DONE ARE NOT FOR CONCLUSION BUT RECOMMENDATION AND ARE SUBJECTED TO EXPERTS SCRUTINY.
```

Analysis

Loading libraries and Dataset

In [3]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
data = pd.read_csv("owid-covid-data.csv")
```

In [5]:
```
data.head()
```

Out[5]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_tho
0	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	NaN	0.0	0.0	NaN	...	NaN	37.746	
1	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	NaN	0.0	0.0	NaN	...	NaN	37.746	
2	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	NaN	0.0	0.0	NaN	...	NaN	37.746	
3	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0	NaN	0.0	0.0	NaN	...	NaN	37.746	
4	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0	NaN	0.0	0.0	NaN	...	NaN	37.746	

5 rows × 67 columns

In [6]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 429435 entries, 0 to 429434
Data columns (total 67 columns):
 #   Column                                      Non-Null Count   Dtype  
---  ------                                      --------------   -----  
 0   iso_code                                    429435 non-null  object 
 1   continent                                   402910 non-null  object 
 2   location                                    429435 non-null  object 
 3   date                                        429435 non-null  object 
 4   total_cases                                 411804 non-null  float64
 5   new_cases                                   410159 non-null  float64
 6   new_cases_smoothed                          408929 non-null  float64
 7   total_deaths                                411804 non-null  float64
 8   new_deaths                                  410608 non-null  float64
 9   new_deaths_smoothed                         409378 non-null  float64
 10  total_cases_per_million                     411804 non-null  float64
 11  new_cases_per_million                       410159 non-null  float64
 12  new_cases_smoothed_per_million              408929 non-null  float64
 13  total_deaths_per_million                    411804 non-null  float64
 14  new_deaths_per_million                      410608 non-null  float64
 15  new_deaths_smoothed_per_million             409378 non-null  float64
 16  reproduction_rate                           184817 non-null  float64
 17  icu_patients                                39116 non-null   float64
 18  icu_patients_per_million                    39116 non-null   float64
 19  hosp_patients                               40656 non-null   float64
 20  hosp_patients_per_million                   40656 non-null   float64
 21  weekly_icu_admissions                       10993 non-null   float64
 22  weekly_icu_admissions_per_million           10993 non-null   float64
 23  weekly_hosp_admissions                      24497 non-null   float64
 24  weekly_hosp_admissions_per_million          24497 non-null   float64
 25  total_tests                                 79387 non-null   float64
 26  new_tests                                   75403 non-null   float64
 27  total_tests_per_thousand                    79387 non-null   float64
 28  new_tests_per_thousand                      75403 non-null   float64
 29  new_tests_smoothed                          103965 non-null  float64
 30  new_tests_smoothed_per_thousand             103965 non-null  float64
 31  positive_rate                               95927 non-null   float64
 32  tests_per_case                              94348 non-null   float64
 33  tests_units                                 106788 non-null  object 
 34  total_vaccinations                          85417 non-null   float64
 35  people_vaccinated                           81132 non-null   float64
 36  people_fully_vaccinated                     78061 non-null   float64
 37  total_boosters                              53600 non-null   float64
 38  new_vaccinations                            70971 non-null   float64
 39  new_vaccinations_smoothed                   195029 non-null  float64
 40  total_vaccinations_per_hundred              85417 non-null   float64
 41  people_vaccinated_per_hundred               81132 non-null   float64
 42  people_fully_vaccinated_per_hundred         78061 non-null   float64
 43  total_boosters_per_hundred                  53600 non-null   float64
 44  new_vaccinations_smoothed_per_million       195029 non-null  float64
 45  new_people_vaccinated_smoothed              192177 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  192177 non-null  float64
 47  stringency_index                            196190 non-null  float64
 48  population_density                          360492 non-null  float64
 49  median_age                                  334663 non-null  float64
 50  aged_65_older                               323270 non-null  float64
 51  aged_70_older                               331315 non-null  float64
 52  gdp_per_capita                              328292 non-null  float64
 53  extreme_poverty                             211996 non-null  float64
 54  cardiovasc_death_rate                       328865 non-null  float64
 55  diabetes_prevalence                         345911 non-null  float64
 56  female_smokers                              247165 non-null  float64
 57  male_smokers                                243817 non-null  float64
 58  handwashing_facilities                      161741 non-null  float64
 59  hospital_beds_per_thousand                  290689 non-null  float64
 60  life_expectancy                             390299 non-null  float64
 61  human_development_index                     319127 non-null  float64
 62  population                                  429435 non-null  int64  
 63  excess_mortality_cumulative_absolute        13411 non-null   float64
 64  excess_mortality_cumulative                 13411 non-null   float64
 65  excess_mortality                            13411 non-null   float64
 66  excess_mortality_cumulative_per_million     13411 non-null   float64
dtypes: float64(61), int64(1), object(5)
memory usage: 219.5+ MB
```

In [7]:
```
data['total_cases']
```

Out[7]:
```
0             0.0
1             0.0
2             0.0
3             0.0
4             0.0
            ...
429430    266386.0
429431    266386.0
429432    266386.0
429433    266386.0
429434    266386.0
Name: total_cases, Length: 429435, dtype: float64
```

TOtal cases by population

In [8]:
```
pivot_table = data.pivot_table(index=['continent'],
                               values=['population','total_cases'],
                               aggfunc='sum').reset_index()
pivot_table.sort_values(by='total_cases', ascending=False, inplace=True)
```

In [9]:
```
pivot_table
```

Out[9]:

	continent	population	total_cases
1	Asia	7908539315276	2.521673e+11
2	Europe	1341870119506	2.367567e+11
3	North America	1005451818298	1.270737e+11
5	South America	731413161942	7.348457e+10
0	Africa	2387393435471	1.461689e+10
4	Oceania	75431427341	1.159805e+10

In [10]:
```
"""
Either this data has been tampered with OR we have some wrong continents or countries values because from the population values up there, they are incorrect.

Reason extra mind of Expetise and Gewneral Knowledge is important in analysis. This would have skewed the analysis.

<TYPE 1 and TYPE 2 error/>

"""
```

Out[10]:
```
'\nEither this data has been tampered with OR we have some wrong continents or countries values because from the population values up there, they are incorrect.\n\nReason extra mind o
f Expetise and Gewneral Knowledge is important in analysis. This would have skewed the analysis.\n\n<TYPE 1 and TYPE 2 error/>\n\n'
```

In [11]:
```
data['continent'].unique()
```

Out[11]:
```
array(['Asia', nan, 'Europe', 'Africa', 'Oceania', 'North America',
       'South America'], dtype=object)
```

In [12]:
```
"""
With just 6 continents, we have correct contients.

"""
```

Out[12]:
```
'\nWith just 6 continents, we have correct contients.\n\n'
```

In [13]:
```
data['location'].nunique()
```

Out[13]:
```
255
```

In [14]:
```
"""
LOL. Gotcha. Obviously this is wrong. So the country values are wrong because there are not 255 countries but 195

"""
```

Out[14]:
```
'\nLOL. Gotcha. Obviously this is wrong. So the country values are wrong because there are not 255 countries but 195\n\n'
```

In [15]:
```
data['location'].unique()
```

Out[15]:
```
array(['Afghanistan', 'Africa', 'Albania', 'Algeria', 'American Samoa',
       'Andorra', 'Angola', 'Anguilla', 'Antigua and Barbuda',
       'Argentina', 'Armenia', 'Aruba', 'Asia', 'Australia', 'Austria',
       'Azerbaijan', 'Bahamas', 'Bahrain', 'Bangladesh', 'Barbados',
       'Belarus', 'Belgium', 'Belize', 'Benin', 'Bermuda', 'Bhutan',
       'Bolivia', 'Bonaire Sint Eustatius and Saba',
       'Bosnia and Herzegovina', 'Botswana', 'Brazil',
       'British Virgin Islands', 'Brunei', 'Bulgaria', 'Burkina Faso',
       'Burundi', 'Cambodia', 'Cameroon', 'Canada', 'Cape Verde',
       'Cayman Islands', 'Central African Republic', 'Chad', 'Chile',
       'China', 'Colombia', 'Comoros', 'Congo', 'Cook Islands',
       'Costa Rica', "Cote d'Ivoire", 'Croatia', 'Cuba', 'Curacao',
       'Cyprus', 'Czechia', 'Democratic Republic of Congo', 'Denmark',
       'Djibouti', 'Dominica', 'Dominican Republic', 'East Timor',
       'Ecuador', 'Egypt', 'El Salvador', 'England', 'Equatorial Guinea',
       'Eritrea', 'Estonia', 'Eswatini', 'Ethiopia', 'Europe',
       'European Union (27)', 'Faroe Islands', 'Falkland Islands', 'Fiji',
       'Finland', 'France', 'French Guiana', 'French Polynesia', 'Gabon',
       'Gambia', 'Georgia', 'Germany', 'Ghana', 'Gibraltar', 'Greece',
       'Greenland', 'Grenada', 'Guadeloupe', 'Guam', 'Guatemala',
       'Guernsey', 'Guinea', 'Guinea-Bissau', 'Guyana', 'Haiti',
       'High-income countries', 'Honduras', 'Hong Kong', 'Hungary',
       'Iceland', 'India', 'Indonesia', 'Iran', 'Iraq', 'Ireland',
       'Isle of Man', 'Israel', 'Italy', 'Jamaica', 'Japan', 'Jersey',
       'Jordan', 'Kazakhstan', 'Kenya', 'Kiribati', 'Kosovo', 'Kuwait',
       'Kyrgyzstan', 'Laos', 'Latvia', 'Lebanon', 'Lesotho', 'Liberia',
       'Libya', 'Liechtenstein', 'Lithuania', 'Low-income countries',
       'Lower-middle-income countries', 'Luxembourg', 'Macao',
       'Madagascar', 'Malawi', 'Malaysia', 'Maldives', 'Mali', 'Malta',
       'Marshall Islands', 'Martinique', 'Mauritania', 'Mauritius',
       'Mayotte', 'Mexico', 'Micronesia (country)', 'Moldova', 'Monaco',
       'Mongolia', 'Montenegro', 'Montserrat', 'Morocco', 'Mozambique',
       'Myanmar', 'Namibia', 'Nauru', 'Nepal', 'Netherlands',
       'New Caledonia', 'New Zealand', 'Nicaragua', 'Niger', 'Nigeria',
       'Niue', 'North America', 'North Korea', 'North Macedonia',
       'Northern Cyprus', 'Northern Ireland', 'Northern Mariana Islands',
       'Norway', 'Oceania', 'Oman', 'Pakistan', 'Palau', 'Palestine',
       'Panama', 'Papua New Guinea', 'Paraguay', 'Peru', 'Philippines',
       'Pitcairn', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',
       'Reunion', 'Romania', 'Russia', 'Rwanda', 'Saint Barthelemy',
       'Saint Helena', 'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Martin (French part)', 'Saint Pierre and Miquelon',
       'Saint Vincent and the Grenadines', 'Samoa', 'San Marino',
       'Sao Tome and Principe', 'Saudi Arabia', 'Scotland', 'Senegal',
       'Serbia', 'Seychelles', 'Sierra Leone', 'Singapore',
       'Sint Maarten (Dutch part)', 'Slovakia', 'Slovenia',
       'Solomon Islands', 'Somalia', 'South Africa', 'South America',
       'South Korea', 'South Sudan', 'Spain', 'Sri Lanka', 'Sudan',
       'Suriname', 'Sweden', 'Switzerland', 'Syria', 'Taiwan',
       'Tajikistan', 'Tanzania', 'Thailand', 'Togo', 'Tokelau', 'Tonga',
       'Trinidad and Tobago', 'Tunisia', 'Turkey', 'Turkmenistan',
       'Turks and Caicos Islands', 'Tuvalu', 'Uganda', 'Ukraine',
       'United Arab Emirates', 'United Kingdom', 'United States',
       'United States Virgin Islands', 'Upper-middle-income countries',
       'Uruguay', 'Uzbekistan', 'Vanuatu', 'Vatican', 'Venezuela',
       'Vietnam', 'Wales', 'Wallis and Futuna', 'Western Sahara', 'World',
       'Yemen', 'Zambia', 'Zimbabwe'], dtype=object)
```

In [16]:
```
"""
Realised from above that we have 255 countries listed in the location column which is wrong because we have just 195 regisgtered countries of the world.
So, we'll try to find the wrong inputs out.
```

```
"""
```

Out[16]:
```
"\nRealised from above that we have 255 countries listed in the location column which is wrong because we have just 195 regisgtered countries of the world.\nSo, we'll try to find the wrong inputs out.\n\n\n"
```

In [17]:
```
"""
There are 195 registered known countries but we have 255 here, meaning we have about 60 wrong input here.
"""
```

Out[17]:
```
'\nThere are 195 registered known countries but we have 255 here, meaning we have about 60 wrong input here.\n'
```

In [18]:
```
"""
So I checked online.Luckily I was able to get a list of countries in excel format.I am happy, makes my work faster and easier.

"""
```

Out[18]:
```
'\nSo I checked online.Luckily I was able to get a list of countries in excel format.I am happy, makes my work faster and easier.\n\n'
```

In [19]:
```
real_countries = pd.read_excel('WorldCountriesList.xlsx')

real_countries
```

Out[19]:

	Country	Abreviation	Capital City	Continent
0	Afghanistan	AFG	Kabul	Asia
1	Albania	ALB	Tirana	Europe
2	Algeria	DZA	Algiers	Africa
3	Andorra	AND	Andorra la Vella	Europe
4	Angola	AGO	Luanda	Africa
...	...	...	...	...
190	Venezuela	VEN	Caracas	S. America
191	Vietnam	VNM	Hanoi	Asia
192	Yemen	YEM	Sana'a	Asia
193	Zambia	ZMB	Lusaka	Africa
194	Zimbabwe	ZWE	Harare	Africa

195 rows × 4 columns

In [20]:
```
not_country = list(set(data['location'].unique()) - set(real_countries['Country']))

not_country
```

Out[20]:
```
['Wallis and Futuna',
 'Bermuda',
 'Martinique',
 'Aruba',
 'Montserrat',
 'Wales',
 'Hong Kong',
 'Asia',
 'United States Virgin Islands',
 'World',
 'Vatican',
 'Isle of Man',
 'Northern Cyprus',
 'East Timor',
 'Cayman Islands',
 'Guam',
 'Oceania',
 'North America',
 'Northern Ireland',
 "Cote d'Ivoire",
 'Mayotte',
 'Pitcairn',
 'Saint Pierre and Miquelon',
 'Saint Barthelemy',
 'Northern Mariana Islands',
 'Gibraltar',
 'Greenland',
 'Micronesia (country)',
 'Saint Vincent and the Grenadines',
 'Sao Tome and Principe',
 'European Union (27)',
 'South America',
 'New Caledonia',
 'British Virgin Islands',
 'Tokelau',
 'Western Sahara',
 'Jersey',
 'French Polynesia',
 'Saint Martin (French part)',
 'Niue',
 'Turks and Caicos Islands',
 'Upper-middle-income countries',
 'England',
 'American Samoa',
 'Low-income countries',
 'Falkland Islands',
 'Anguilla',
 'Guernsey',
 'Cape Verde',
 'Palestine',
 'Czechia',
 'Saint Helena',
 'Democratic Republic of Congo',
 'Guadeloupe',
 'Scotland',
 'Sint Maarten (Dutch part)',
 'Bonaire Sint Eustatius and Saba',
 'Puerto Rico',
 'High-income countries',
 'Kosovo',
 'Saint Kitts and Nevis',
 'Lower-middle-income countries',
 'Faroe Islands',
 'Africa',
 'Taiwan',
 'Cook Islands',
 'Curacao',
 'French Guiana',
 'Reunion',
 'Europe',
 'Macao']
```

In [21]:
```
len(not_country)
```

Out[21]:
```
71
```

In [22]:
```
"""
So we were able to pick about 71 locations that are either not a country at all or wrongly spelt as to 60 exoected.
So by chance we have 11 more which is about 5.6% of the whole real known countries. So we can just ignore since it's
relatively insignificant.

"""
```

Out[22]:
```
"\nSo we were able to pick about 71 locations that are either not a country at all or wrongly spelt as to 60 exoected.\nSo by chance we have 11 more which is about 5.6% of the whole r
eal known countries. So we can just ignore since it's\nrelatively insignificant.\n\n"
```

Droping rows with the wrong location value

In [23]:
```
for country in not_country:
    data.drop(list(data[data['location']==country].index), inplace=True)
#    for index in not_country_index:
#        data.drop(index)


#data.drop(list(data[data['location']==country].index), 0)
```

In [24]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 308071 entries, 0 to 429434
Data columns (total 67 columns):
 #   Column                                      Non-Null Count   Dtype
---  ------                                      --------------   -----
 0   iso_code                                    308071 non-null  object
 1   continent                                   308071 non-null  object
 2   location                                    308071 non-null  object
 3   date                                        308071 non-null  object
 4   total_cases                                 308016 non-null  float64
 5   new_cases                                   306375 non-null  float64
 6   new_cases_smoothed                          305455 non-null  float64
 7   total_deaths                                308016 non-null  float64
 8   new_deaths                                  306823 non-null  float64
 9   new_deaths_smoothed                         305903 non-null  float64
 10  total_cases_per_million                     308016 non-null  float64
 11  new_cases_per_million                       306375 non-null  float64
 12  new_cases_smoothed_per_million              305455 non-null  float64
 13  total_deaths_per_million                    308016 non-null  float64
 14  new_deaths_per_million                      306823 non-null  float64
 15  new_deaths_smoothed_per_million             305903 non-null  float64
 16  reproduction_rate                           173540 non-null  float64
 17  icu_patients                                33199 non-null   float64
 18  icu_patients_per_million                    33199 non-null   float64
 19  hosp_patients                               33826 non-null   float64
 20  hosp_patients_per_million                   33826 non-null   float64
 21  weekly_icu_admissions                       9375 non-null    float64
 22  weekly_icu_admissions_per_million           9375 non-null    float64
 23  weekly_hosp_admissions                      18048 non-null   float64
 24  weekly_hosp_admissions_per_million          18048 non-null   float64
 25  total_tests                                 73425 non-null   float64
 26  new_tests                                   68808 non-null   float64
 27  total_tests_per_thousand                    73425 non-null   float64
 28  new_tests_per_thousand                      68808 non-null   float64
 29  new_tests_smoothed                          95037 non-null   float64
 30  new_tests_smoothed_per_thousand             95037 non-null   float64
 31  positive_rate                               88077 non-null   float64
 32  tests_per_case                              86684 non-null   float64
 33  tests_units                                 96923 non-null   object
 34  total_vaccinations                          59535 non-null   float64
 35  people_vaccinated                           55845 non-null   float64
 36  people_fully_vaccinated                     53170 non-null   float64
 37  total_boosters                              33764 non-null   float64
 38  new_vaccinations                            47692 non-null   float64
 39  new_vaccinations_smoothed                   149897 non-null  float64
 40  total_vaccinations_per_hundred              59535 non-null   float64
 41  people_vaccinated_per_hundred               55845 non-null   float64
 42  people_fully_vaccinated_per_hundred         53170 non-null   float64
 43  total_boosters_per_hundred                  33764 non-null   float64
 44  new_vaccinations_smoothed_per_million       149897 non-null  float64
 45  new_people_vaccinated_smoothed              147421 non-null  float64
 46  new_people_vaccinated_smoothed_per_hundred  147421 non-null  float64
 47  stringency_index                            182390 non-null  float64
 48  population_density                          304723 non-null  float64
 49  median_age                                  293005 non-null  float64
 50  aged_65_older                               291331 non-null  float64
 51  aged_70_older                               291331 non-null  float64
 52  gdp_per_capita                              296353 non-null  float64
 53  extreme_poverty                             197580 non-null  float64
 54  cardiovasc_death_rate                       298027 non-null  float64
 55  diabetes_prevalence                         306397 non-null  float64
 56  female_smokers                              239437 non-null  float64
 57  male_smokers                                236089 non-null  float64
 58  handwashing_facilities                      152347 non-null  float64
 59  hospital_beds_per_thousand                  277939 non-null  float64
 60  life_expectancy                             308071 non-null  float64
 61  human_development_index                     298027 non-null  float64
 62  population                                  308071 non-null  int64
 63  excess_mortality_cumulative_absolute        11469 non-null   float64
 64  excess_mortality_cumulative                 11469 non-null   float64
 65  excess_mortality                            11469 non-null   float64
 66  excess_mortality_cumulative_per_million     11469 non-null   float64
dtypes: float64(61), int64(1), object(5)
memory usage: 159.8+ MB
```

In [25]:
```
"""
About 28% of the records were drop. That may seem like a bad judgement call but I'd appreciate any other way to go about it.

"""
```

Out[25]:
```
"\nAbout 28% of the records were drop. That may seem like a bad judgement call but I'd appreciate any other way to go about it.\n\n"
```

In [ ]:
```

```

Checking for Null values by columns

In [26]:
```
data.isna().sum()
```

Out[26]:
```
iso_code                                      0
continent                                     0
location                                      0
date                                          0
total_cases                                  55
                                            ...
population                                    0
excess_mortality_cumulative_absolute     296602
excess_mortality_cumulative              296602
excess_mortality                         296602
excess_mortality_cumulative_per_million  296602
Length: 67, dtype: int64
```

In [27]:
```
isna = pd.DataFrame(data.isna().sum(), columns=['Null values']).reset_index()
isna
```

Out[27]:

	index	Null values
0	iso_code	0
1	continent	0
2	location	0
3	date	0
4	total_cases	55
...	...	...
62	population	0
63	excess_mortality_cumulative_absolute	296602
64	excess_mortality_cumulative	296602
65	excess_mortality	296602
66	excess_mortality_cumulative_per_million	296602

67 rows × 2 columns

In [28]:
```
# Number of columns with null values

len(isna[isna['Null values']>0])
```

Out[28]: 61

In [29]:
```
"""
FYI, By dropping those rows up there, I was 2 columns short from numbers of columns with Null values.
I saw this because I had run the analysis before without catching the wron countries


"""
```

Out[29]:
```
'\nFYI, By dropping those rows up there, I was 2 columns short from numbers of columns with Null values.\nI saw this because I had run the analysis before without catching the wron co
untries\n\n\n'
```

In [30]:
```
dict = pd.DataFrame(data.dtypes, columns=['ColumnType']).reset_index()
dict
```

Out[30]:

	index	ColumnType
0	iso_code	object
1	continent	object
2	location	object
3	date	object
4	total_cases	float64
...	...	...
62	population	int64
63	excess_mortality_cumulative_absolute	float64
64	excess_mortality_cumulative	float64
65	excess_mortality	float64
66	excess_mortality_cumulative_per_million	float64

67 rows × 2 columns

Categorical Features, cleaninng and imputation

In [31]:
```
dict['ColumnType'].unique()
```

Out[31]:
```
array([dtype('O'), dtype('float64'), dtype('int64')], dtype=object)
```

In [32]:
```
dict['ColumnType'].value_counts()
```

Out[32]:
```
ColumnType
float64    61
object      5
int64       1
Name: count, dtype: int64
```

In [33]:
```
categorical_columns = data.select_dtypes(include=['object'])
numerical_columns = data.select_dtypes(exclude=['object'])
```

In [34]:
```
categorical_columns
```

Out[34]:

	iso_code	continent	location	date	tests_units
0	AFG	Asia	Afghanistan	2020-01-05	NaN
1	AFG	Asia	Afghanistan	2020-01-06	NaN
2	AFG	Asia	Afghanistan	2020-01-07	NaN
3	AFG	Asia	Afghanistan	2020-01-08	NaN
4	AFG	Asia	Afghanistan	2020-01-09	NaN
...	...	...	...	...	...
429430	ZWE	Africa	Zimbabwe	2024-07-31	NaN
429431	ZWE	Africa	Zimbabwe	2024-08-01	NaN
429432	ZWE	Africa	Zimbabwe	2024-08-02	NaN
429433	ZWE	Africa	Zimbabwe	2024-08-03	NaN
429434	ZWE	Africa	Zimbabwe	2024-08-04	NaN

308071 rows × 5 columns

In [35]:
```
categorical_columns.isna().sum()
```

Out[35]:
```
iso_code            0
continent           0
location            0
date                0
tests_units    211148
dtype: int64
```

In [36]:
```
"""
Before addressing the wrong countries value, I had Null values in a my cartegorical columns but now just test_units column alone.
"""
```

Out[36]:
```
'\nBefore addressing the wrong countries value, I had Null values in a my cartegorical columns but now just test_units column alone.\n'
```

In [37]:
```
#Convert date into datetime format

data['date'] = pd.to_datetime(data['date'])
```

In [38]:
```
data['continent'].value_counts(normalize=True)
```

Out[38]:
```
continent
Africa           0.271691
Asia             0.244583
Europe           0.233722
North America    0.114123
Oceania          0.070662
South America    0.065219
Name: proportion, dtype: float64
```

In [39]:
```
data['tests_units'].value_counts(normalize=True)
```

Out[39]:
```
tests_units
tests performed    0.738535
people tested      0.154215
samples tested     0.098573
units unclear      0.008677
Name: proportion, dtype: float64
```

In [40]:
```
(data['tests_units'].isna().sum()/len(data))*100
```

Out[40]:
```
68.53874593843628
```

In [41]:
```
data['tests_units'].fillna('tests performed', inplace=True)
```

In [42]:
```
data[categorical_columns.columns].isna().sum()
```

Out[42]:
```
iso_code       0
continent      0
location       0
date           0
tests_units    0
dtype: int64
```

Numerical Features, cleaning, imputation, Stats, Correlation and Visualization

In [43]:
```
for col in numerical_columns.columns:
    data[col].fillna(data[col].mean(),inplace= True)
```

In [44]:
```
data[numerical_columns.columns].isna().sum()
```

Out[44]:
```
total_cases                                0
new_cases                                  0
new_cases_smoothed                         0
total_deaths                               0
new_deaths                                 0
                                          ..
population                                 0
excess_mortality_cumulative_absolute       0
excess_mortality_cumulative                0
excess_mortality                           0
excess_mortality_cumulative_per_million    0
Length: 62, dtype: int64
```

In [45]:
```
data.isna().sum()
```

Out[45]:
```
iso_code                                0
continent                               0
location                                0
date                                    0
total_cases                             0
                                       ..
population                              0
excess_mortality_cumulative_absolute    0
excess_mortality_cumulative             0
excess_mortality                        0
excess_mortality_cumulative_per_million 0
Length: 67, dtype: int64
```

In [46]:
```
data.head()
```

Out[46]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_tho
0	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
1	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
2	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
3	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
4	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	

5 rows × 67 columns

In [47]:
```
data.describe()
```

Out[47]:

	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed_per_
count	308071	3.080710e+05	3.080710e+05	3.080710e+05	3.080710e+05	308071.000000	308071.000000	308071.000000	308071.000000	308071
mean	2022-04-20 14:25:34.905914112	2.294454e+06	2.502989e+03	2.510398e+03	2.575100e+04	22.790257	22.856527	97769.124267	105.345363	105
min	2020-01-01 00:00:00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000	0.000000	0
25%	2021-02-26 00:00:00	1.122600e+04	0.000000e+00	1.430000e-01	1.250000e+02	0.000000	0.000000	1606.816000	0.000000	0
50%	2022-04-21 00:00:00	1.156670e+05	0.000000e+00	2.100000e+01	1.461000e+03	0.000000	0.143000	20155.080000	0.000000	2
75%	2023-06-13 00:00:00	9.708600e+05	0.000000e+00	4.150000e+02	1.104300e+04	0.000000	4.143000	123379.875000	0.000000	53
max	2024-08-14 00:00:00	1.034368e+08	4.047548e+07	5.782211e+06	1.193165e+06	47687.000000	6812.429000	763598.600000	226617.450000	32373
std	NaN	8.798102e+06	9.659652e+04	3.643659e+04	9.242362e+04	354.207142	132.205300	152347.634459	1203.655824	444

8 rows × 63 columns

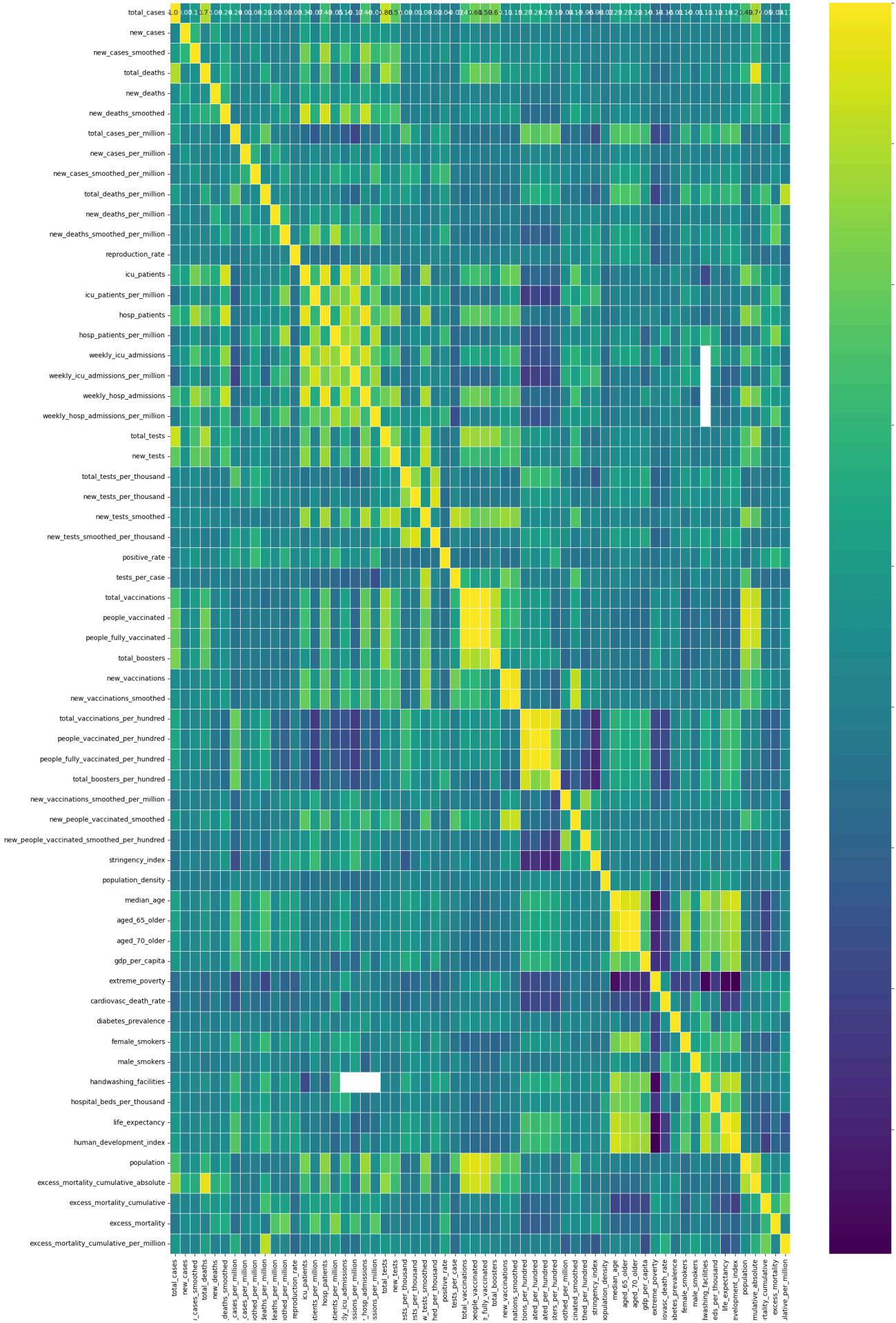
In [48]:
```
numerical_columns.corr()
```

Out[48]:

	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	total_cases_per_million	new_cases_per_million	new_cases_smoothed
total_cases	1.000000	0.071470	0.189431	0.790568	0.089498	0.239519	0.238183	0.014882	
new_cases	0.071470	1.000000	0.377204	0.052644	0.298965	0.110590	0.007722	0.168049	
new_cases_smoothed	0.189431	0.377204	1.000000	0.139516	0.109473	0.293182	0.020348	0.058503	
total_deaths	0.790568	0.052644	0.139516	1.000000	0.128163	0.343106	0.124878	0.006805	
new_deaths	0.089498	0.298965	0.109473	0.128163	1.000000	0.373247	-0.008698	0.109577	
...	...	...	...	...	...	...	...	...	...
population	0.488764	0.054619	0.145017	0.384724	0.090958	0.244035	-0.079390	-0.010335	
excess_mortality_cumulative_absolute	0.762411	0.316022	0.339193	0.934399	0.363806	0.390182	0.024064	-0.031965	
excess_mortality_cumulative	0.053236	0.028481	0.034082	0.229938	0.181702	0.191255	-0.005735	-0.028057	
excess_mortality	-0.039347	0.080473	0.092995	0.001559	0.293509	0.313856	-0.122786	0.104592	
excess_mortality_cumulative_per_million	0.166131	0.006054	0.009670	0.263697	0.015262	0.021865	0.335403	-0.014098	

62 rows × 62 columns

In [49]:
```
#Correlation matrix
plt.figure(figsize=(20, 30))
#fig.set_size_inches(5,15)
corr= numerical_columns.corr()
sns.heatmap(corr, annot=True, cmap= 'viridis',fmt='.2f', linewidths=0.5)
plt.tight_layout()
plt.show()
```


total_cases
new_cases
new_cases_smoothed
total_deaths
new_deaths
new_deaths_smoothed
total_cases_per_million
new_cases_per_million
new_cases_smoothed_per_million
total_deaths_per_million
new_deaths_per_million
new_deaths_smoothed_per_million
reproduction_rate
icu_patients
icu_patients_per_million
hosp_patients
hosp_patients_per_million
weekly_icu_admissions
weekly_icu_admissions_per_million
weekly_hosp_admissions
weekly_hosp_admissions_per_million
total_tests
new_tests
total_tests_per_thousand
new_tests_per_thousand
new_tests_smoothed
new_tests_smoothed_per_thousand
positive_rate
tests_per_case
total_vaccinations
people_vaccinated
people_fully_vaccinated
total_boosters
new_vaccinations
new_vaccinations_smoothed
total_vaccinations_per_hundred
people_vaccinated_per_hundred
people_fully_vaccinated_per_hundred
total_boosters_per_hundred
new_vaccinations_smoothed_per_million
new_people_vaccinated_smoothed
new_people_vaccinated_smoothed_per_hundred
stringency_index
population_density
median_age
aged_65_older
aged_70_older
gdp_per_capita
extreme_poverty
cardiovasc_death_rate
diabetes_prevalence
female_smokers
male_smokers
handwashing_facilities
hospital_beds_per_thousand
life_expectancy
human_development_index
population
excess_mortality_cumulative_absolute
excess_mortality_cumulative
excess_mortality
excess_mortality_cumulative_per_million
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

In [50]:
```
#Corelation of numerical features against tital cases

corr_list = []

for col in numerical_columns:
    corr_list.append(data['total_cases'].corr(data[col]))

corr_df = pd.DataFrame(corr_list, numerical_columns.columns, columns=['Correlation Value']).reset_index()
```

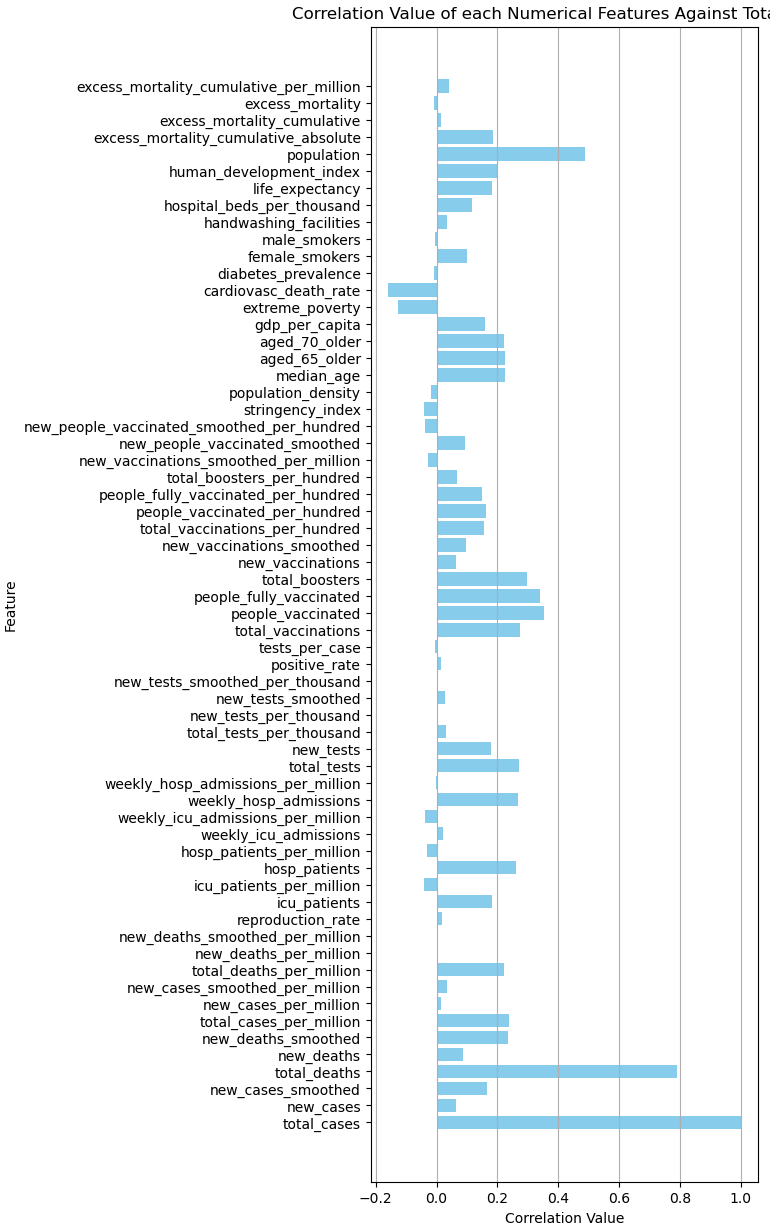
In [51]:
```
corr_df
```

Out[51]:

	index	Correlation Value
0	total_cases	1.000000
1	new_cases	0.062438
2	new_cases_smoothed	0.165476
3	total_deaths	0.790568
4	new_deaths	0.087376
...	...	...
57	population	0.488248
58	excess_mortality_cumulative_absolute	0.185522
59	excess_mortality_cumulative	0.012954
60	excess_mortality	-0.009575
61	excess_mortality_cumulative_per_million	0.040426

62 rows × 2 columns

In [52]:
```
plt.figure(figsize=(5, 15))
plt.barh(numerical_columns.columns, corr_df['Correlation Value'], color='skyblue')
plt.xlabel('Correlation Value')
plt.ylabel('Feature')
plt.title('Correlation Value of each Numerical Features Against Total Cases')
plt.grid(axis='x')
plt.show()
```



In [53]:
```
#Corelation of numerical features against tital deaths

corr_list = []

for col in numerical_columns:
    corr_list.append(data['total_deaths'].corr(data[col]))

corr_df = pd.DataFrame(corr_list, numerical_columns.columns, columns=['Correlation Value']).reset_index()
```

In [54]:
```
corr_df
```

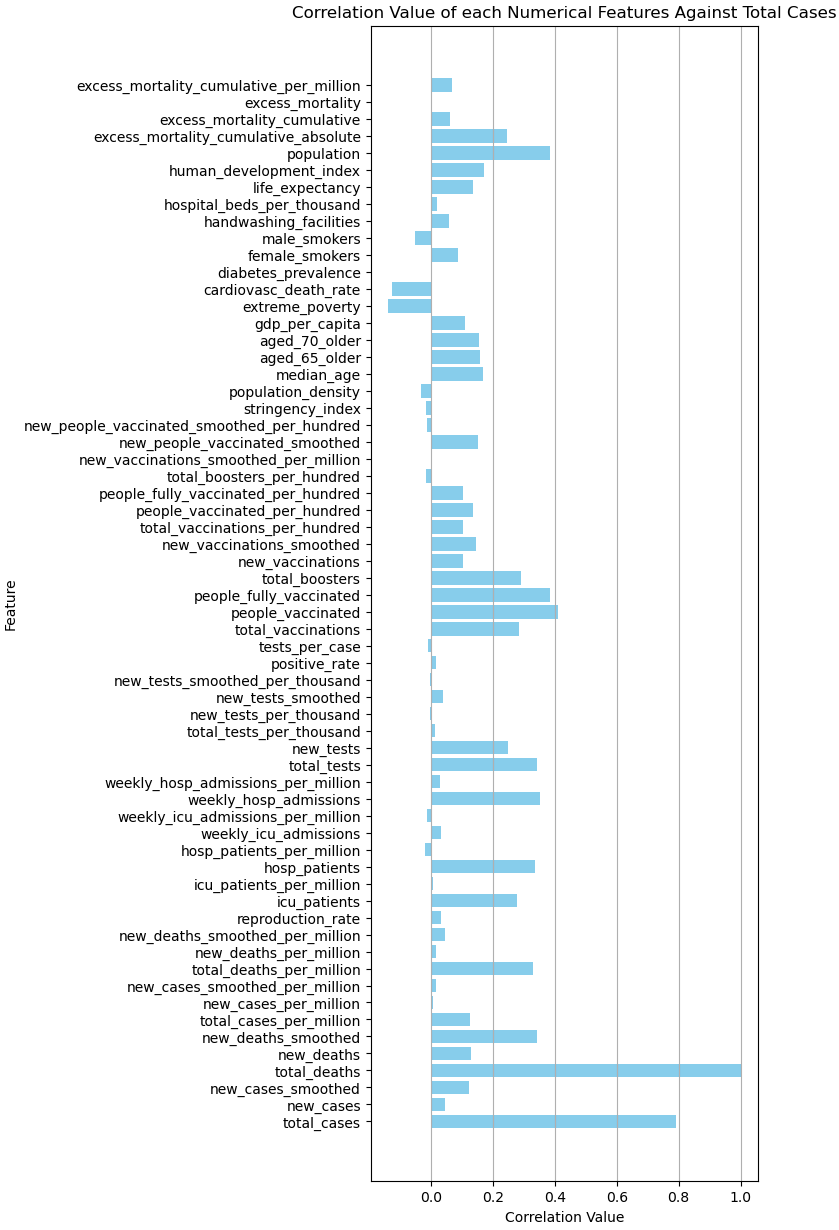
Out[54]:

	index	Correlation Value
0	total_cases	0.790568
1	new_cases	0.046334
2	new_cases_smoothed	0.122779
3	total_deaths	1.000000
4	new_deaths	0.127661
...	...	...
57	population	0.384317
58	excess_mortality_cumulative_absolute	0.245181
59	excess_mortality_cumulative	0.060334
60	excess_mortality	0.000409
61	excess_mortality_cumulative_per_million	0.069193

62 rows × 2 columns

In [55]:
```
plt.figure(figsize=(5, 15))
plt.barh(numerical_columns.columns, corr_df['Correlation Value'], color='skyblue')
```

```
plt.xlabel('Correlation Value')
plt.ylabel('Feature')
plt.title('Correlation Value of each Numerical Features Against Total Cases')
plt.grid(axis='x')
plt.show()
```



In [ ]:

First Pivot Table of Total Cases by Continents and Population together

In [56]:
```
continents = []
population = []

for continent in data['continent'].unique():
    continents.append(continent)
    population.append(data[data['continent']==continent]['population'].unique().sum())

pivot_table_now = pd.DataFrame({'continent':continents, 'population':population})
    #print(continent)
```

In [57]:
```
pivot_table_now.sort_values(by=['population'], ascending=False, inplace=True)

pivot_table_now
```

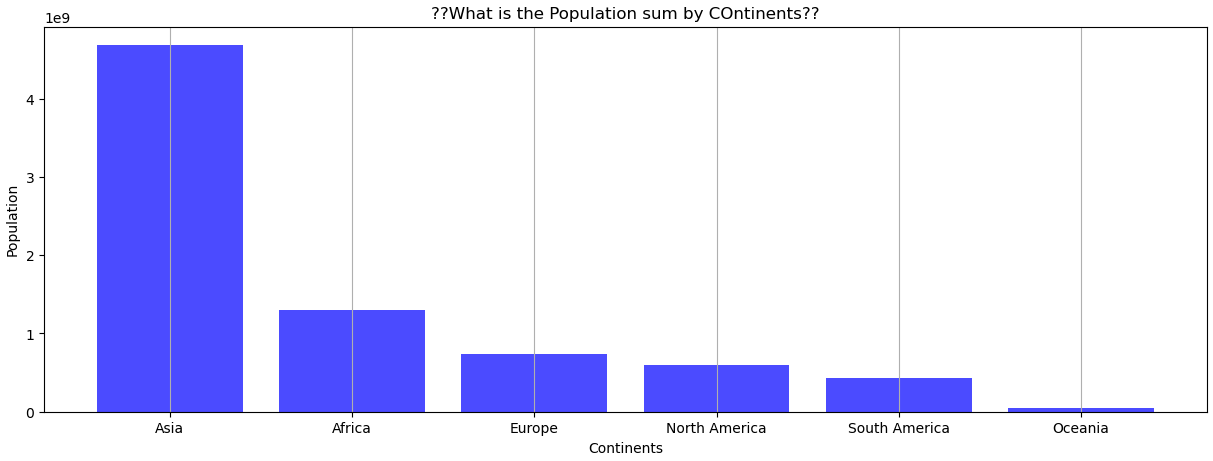
Out[57]:

	continent	population
0	Asia	4682786577
2	Africa	1296863714
1	Europe	734789904
3	North America	595352510
4	South America	436508310
5	Oceania	44030306

In [58]:
```
"""
So after so much long ghurdle we were able to come close to the real life dataq we have online.
Yaaaayyyyy


"""
```

Out[58]:
```
'\nSo after so much long ghurdle we were able to come close to the real life dataq we have online.\nYaaaayyyyy\n\n\n'
```

In [59]:
```
# Vizualization of Population by continents

plt.figure(figsize=(15, 5))
#plt.bar(pivot_table['continent'], pivot_table['total_cases'], color='blue', alpha=0.7)
plt.bar(pivot_table_now['continent'], pivot_table_now['population'], color='blue', alpha=0.7)
plt.xlabel('Continents')
plt.ylabel('Population')
plt.title('??What is the Population sum by COntinents??')
plt.grid(axis='x')
plt.show()
```



I am trying to plot the maximum value of total cases for each year, hue by continent

In [60]:
```
"""
Since the total case count only changes with time and is a recurring count, that is, same patient could be counted the following day.

SO the pitale_table result below will be wrong.

"""
```

Out[60]:
```
'\nSince the total case count only changes with time and is a recurring count, that is, same patient could be counted the following day.\n\nSO the pitale_table result below will be wr
ong.\n\n'
```

In [61]:
```
pivot_table_total_cases = data.pivot_table(index='continent',
                                           values='total_cases',
                                           aggfunc='sum'

).sort_values(by=['total_cases'], ascending=False)

pivot_table_total_cases
```

Out[61]:

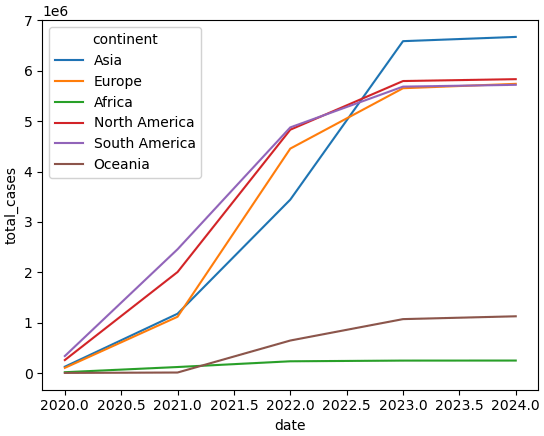
continent	total_cases
Asia	2.514045e+11
Europe	2.314803e+11
North America	1.253728e+11
South America	7.339164e+10
Africa	1.383996e+10
Oceania	1.136559e+10

In [ ]:
```
"""
So we can't use MEAN as the estimator/aggregation because total_cases is a cummulative frequency (THINK OUTLIERS). I mean, the number of cases at any instant is a cummulative
of the previous days, So I decided to use MAX instead as it will give the last recorded number of cases.

SEE WRONG PLOT BELOW
"""
```

In [62]:
```
sns.lineplot(data=data,
                x=data['date'].dt.year,
                y='total_cases',
             hue='continent',
             estimator='mean',
             ci=None
               )
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\760828855.py:1: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[62]: <Axes: xlabel='date', ylabel='total_cases'>



In [ ]:
```
"""
So we check for yearly and use the last record for each year.

"""
```

In [63]:
```
nth = 1
while nth <= len(data['date'].dt.year.unique()):
    if nth == 1:
        year_20 = data[data['date'].dt.year==2020][['continent', 'total_cases']].groupby(by=['continent']).max().rename(columns={'total_cases':'year_20'})
        nth+=1
    elif nth == 2:
        year_21 = data[data['date'].dt.year==2021][['continent', 'total_cases']].groupby(by=['continent']).max().rename(columns={'total_cases':'year_21'})
        nth+=1
    elif nth == 3:
        year_22 = data[data['date'].dt.year==2022][['continent', 'total_cases']].groupby(by=['continent']).max().rename(columns={'total_cases':'year_22'})
        nth+=1
    elif nth == 4:
        year_23 = data[data['date'].dt.year==2023][['continent', 'total_cases']].groupby(by=['continent']).max().rename(columns={'total_cases':'year_23'})
        nth+=1
    elif nth == 5:
        year_24 = data[data['date'].dt.year==2024][['continent', 'total_cases']].groupby(by=['continent']).max().rename(columns={'total_cases':'year_24'})
        nth+=1
```

In [64]:
```
total_cases_by_continent = pd.concat([year_20,year_21,year_22,year_23,year_24], axis=1)

total_cases_by_continent
```

Out[64]:

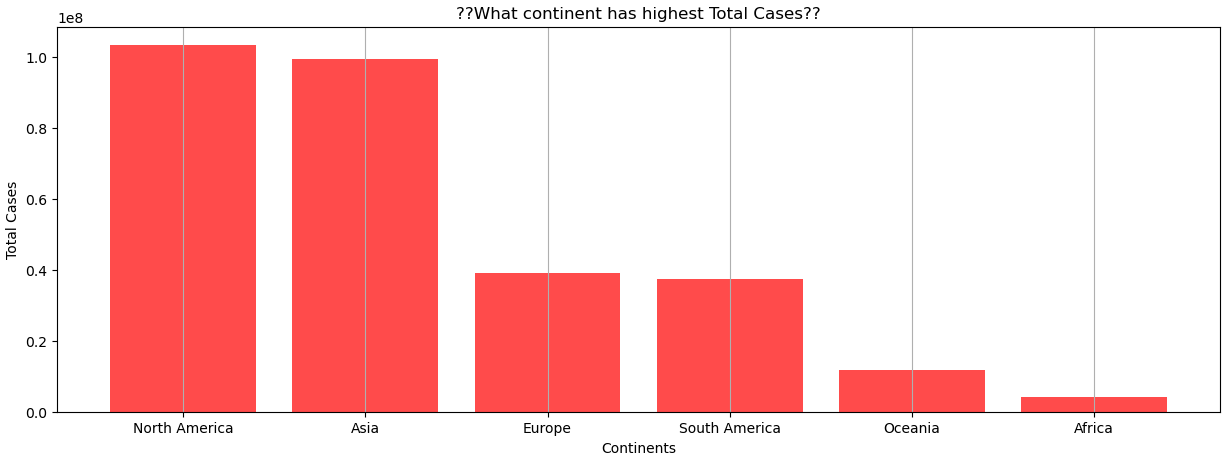
continent	year_20	year_21	year_22	year_23	year_24
Africa	994911.0	3407937.0	4048232.0	4072636.0	4072765.0
Asia	10187850.0	34786802.0	62445650.0	99322727.0	99373219.0
Europe	3050248.0	12574779.0	37989547.0	38997490.0	38997490.0
North America	18890446.0	51878860.0	99019493.0	103436829.0	103436829.0
Oceania	28296.0	366607.0	10694041.0	11721535.0	11861161.0
South America	7448560.0	22230737.0	36124337.0	37511921.0	37511921.0

In [65]:
```
cum_total_cases = pd.DataFrame(total_cases_by_continent.max(axis=1)).rename(columns={0:'sum_total_cases'}).sort_values(by=['sum_total_cases'], ascending=False).reset_index()
cum_total_cases
```

Out[65]:

	continent	sum_total_cases
0	North America	103436829.0
1	Asia	99373219.0
2	Europe	38997490.0
3	South America	37511921.0
4	Oceania	11861161.0
5	Africa	4072765.0

In [66]:
```
# Vizualization of Total cases by continents

plt.figure(figsize=(15, 5))
plt.bar(cum_total_cases['continent'], cum_total_cases['sum_total_cases'], color='red', alpha=0.7)
#plt.bar(pivot_table['continent'], pivot_table['population'], color='red', alpha=0.7)
plt.xlabel('Continents')
plt.ylabel('Total Cases')
plt.title('??What continent has highest Total Cases??')
plt.grid(axis='x')
plt.show()
```

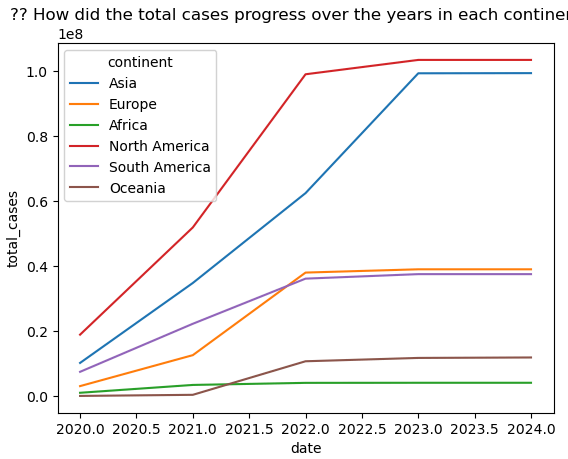


In [ ]:
```
"""
SO I TRIED TO CONFIRM IF TRULY NORTH AMARICA HAD THE HIEST RECORD SO FAR BUT IT WAS INDICATED TO BE ASIA.
SO I BACK TRACED TO OUr YEARLY TABLE UP AND REALISED THAT EVEN IN 2020, NORTH AMERICA HAD MORE CASES THAN ASIA.
THOUGH THE DIFFERENCE BETWEEN THE TWO CONTINENTS IS ABOUT 4M.
THE PICK IS THAT ASIA-CHINA WAS THE BEGINING OF COVID BUT WE COULD LATER INVESTIGATE BETWEEN NORTH AMERICA AND ASIA COUNTRIES JUST FOR 2020.

BUT WE'LL LOOK AWAY, ASSUMING THIS DATA HAD BEEN TWEAKED OR HAVE SOME ERROR INPUT OR I AM WRONG SOME WHERE.
PLEASE FEEL FREE TO COMMENT AND POINT IT OUT AND WE'LL LEARN TOGETHER.


"""
```

In [67]:
```
# How cases in each continents progressed over the years

sns.lineplot(data=data,
                x=data['date'].dt.year,
                y='total_cases',
             hue='continent',
             estimator='max',
             ci=None
               ).set_title('?? How did the total cases progress over the years in each continent??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\1022548101.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[67]:
```
Text(0.5, 1.0, '?? How did the total cases progress over the years in each continent??')
```



In [ ]:
```
"""
So in support of our data table we deducted, we have this plot using max as the estimator aggregator as the parameter.
"""
```

In [ ]:
```
"""
Especially in 2022, ASIA and NORTH AMERICA seems to have a significant rise in total cases.
Also it can be seen that from around 2022 there has been approximately no increase in the record meaning the new cases is near 0 for continents like
EUROPE, SOUTH AMERICA, OCEANIA and AFRICA. But ASIA and NORTH AMERICA still struggled till around 2023 before they could contain the outbreak.

You should ask WHY?

"""
```

In [ ]:
```
"""
So I'll maerge the population of each continent and the cummulative total cases recorded into a single table so that we can visualize and see better.

"""
```

In [68]:
```
cases_by_pop = pd.merge(pivot_table_now,cum_total_cases, on='continent').sort_values(by=['sum_total_cases'], ascending=False)
cases_by_pop
```

Out[68]:

	continent	population	sum_total_cases
3	North America	595352510	103436829.0
0	Asia	4682786577	99373219.0
2	Europe	734789904	38997490.0
4	South America	436508310	37511921.0
5	Oceania	44030306	11861161.0
1	Africa	1296863714	4072765.0

In [69]:
```
# Saving table above to excel for further investigation
cases_by_pop.to_csv('pivot.csv')
```

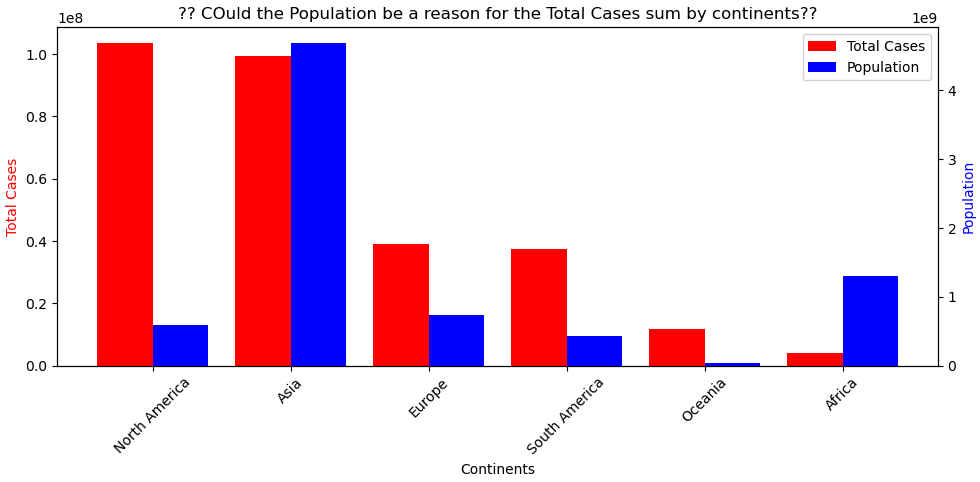
In [70]:
```
excel_sheet = pd.read_csv('pivot.csv')
excel_sheet
```

Out[70]:

	Unnamed: 0	continent	population	sum_total_cases
0	3	North America	595352510	103436829.0
1	0	Asia	4682786577	99373219.0
2	2	Europe	734789904	38997490.0
3	4	South America	436508310	37511921.0
4	5	Oceania	44030306	11861161.0
5	1	Africa	1296863714	4072765.0

In [ ]:
```
"""
Looing at this data another way, that is relative to the population of the continents,
OCEANINA seems to be more affected because 1 out of every 4 people were sick and in
ASIA 1 out of every 48 people were sick.
And of course AFRICA seems relativelhy coolwith 1 out of every 319 people sick.


"""
```

In [71]:
```
# Ploting Total cases and Population by continents for easy comparison

fig, ax = plt.subplots(figsize = (10, 5))
plt.title('?? COuld the Population be a reason for the Total Cases sum by continents??')

ax2 = ax.twinx()

X_axis = np.arange(len(cases_by_pop['continent']))
ax.bar(X_axis - 0.2, cases_by_pop['sum_total_cases'], 0.4, label = 'Total Cases', color='red')
ax2.bar(X_axis + 0.2, cases_by_pop['population'], 0.4, label = 'Population', color='blue')

ax.set_xticks(X_axis, cases_by_pop['continent'], rotation=45)
ax.set_xlabel('Continents')
ax.set_ylabel('Total Cases', color='red')

ax2.set_ylabel('Population', color='blue')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



In [ ]:
```
"""
From above it can be well observed that the population desity of the continents played a role in the number of cases as in ASIA which is reasonable giving that
the COVID started with them.
But something more attracting is the NORTH AMERICA significantly or exponential cases relative to their population and another thing is AFRICA inverse correlation.

So we'll first check in to ASIA to see what happened then we'll fly to NORTH AMERICA before heading to AFRICA.


"""
```

In [ ]:
```
"""



"""
```

Investigating into ASIA

In [ ]:
```
"""
ASIA WE ARE!!!!

First, I want to check how their cases progressed over the years.

"""
```

In [77]:
```
# Recall our yearly table

total_cases_by_continent = total_cases_by_continent.reset_index()
total_cases_by_continent
```

Out[77]:

	continent	year_20	year_21	year_22	year_23	year_24
0	Africa	994911.0	3407937.0	4048232.0	4072636.0	4072765.0
1	Asia	10187850.0	34786802.0	62445650.0	99322727.0	99373219.0
2	Europe	3050248.0	12574779.0	37989547.0	38997490.0	38997490.0
3	North America	18890446.0	51878860.0	99019493.0	103436829.0	103436829.0
4	Oceania	28296.0	366607.0	10694041.0	11721535.0	11861161.0
5	South America	7448560.0	22230737.0	36124337.0	37511921.0	37511921.0

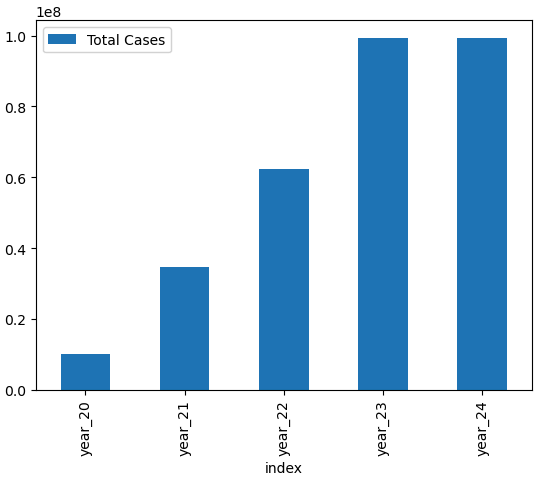
In [78]:
```
# For ASIA alone

total_cases_by_continent[total_cases_by_continent['continent']=='Asia']
```

Out[78]:

	continent	year_20	year_21	year_22	year_23	year_24
1	Asia	10187850.0	34786802.0	62445650.0	99322727.0	99373219.0

In [79]:
```
# For ASIA alone

total_cases_by_continent[total_cases_by_continent['continent']=='Asia'].transpose().reset_index().drop(0,axis=0).rename(columns={1:'Total Cases'}).plot(kind='bar',x='index',y=1)
```

Out[79]:
```
<Axes: xlabel='index'>
```



In [80]:
```
((34786802.0-10187850.0)/10187850.0)*100
```

Out[80]:
```
241.45381017584674
```

In [81]:
```
((62445650.0-34786802.0)/34786802.0)*100
```

Out[81]:
```
79.50960252109405
```

In [82]:
```
((99322727.0-62445650.0)/62445650.0)*100
```

Out[82]:
```
59.05467714724725
```

In [83]:
```
((99373219.0-99322727.0)/99322727.0)*100
```

Out[83]:
```
0.05083630053774097
```

In [ ]:
```
"""
Now we look deeper into the continent to check for the countries in ASIA
```

```
I confirmed the countries listed in our data under ASIA and the countries in the real world, we are 3 countries short.
"""
```

In [84]:
```
data_asia = data[data['continent']=='Asia']
data_asia
```

Out[84]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_p
0	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
1	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
2	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
3	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
4	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
426082	YEM	Asia	Yemen	2024-07-31	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426083	YEM	Asia	Yemen	2024-08-01	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426084	YEM	Asia	Yemen	2024-08-02	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426085	YEM	Asia	Yemen	2024-08-03	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426086	YEM	Asia	Yemen	2024-08-04	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	

75349 rows × 67 columns

In [85]:
```
# Pivot table to see the total cases by continents

pivot_table_asia = data_asia.pivot_table(index=['location'],
                            values=['total_cases','population'],
                            aggfunc='max').reset_index()
pivot_table_asia.sort_values(by='total_cases', ascending=False)
```

Out[85]:

	location	population	total_cases
8	China	1425887360	99373219.0
10	India	1417173120	45041748.0
34	South Korea	51815808	34571873.0
15	Japan	123951696	33803572.0
39	Turkey	85341248	17004718.0
43	Vietnam	98186856	11624000.0
12	Iran	88550568	7627863.0
11	Indonesia	275501344	6829399.0
22	Malaysia	33938216	5309410.0
14	Israel	9449000	4841558.0
38	Thailand	71697024	4799180.0
30	Philippines	115559008	4173631.0
33	Singapore	5637022	3006155.0
13	Iraq	44496124	2465545.0
4	Bangladesh	171186368	2051348.0
9	Georgia	3744385	1863615.0
16	Jordan	11285875	1746997.0
29	Pakistan	235824864	1580631.0
17	Kazakhstan	19397998	1504370.0
21	Lebanon	5489744	1239904.0
41	United Arab Emirates	9441138	1067030.0
24	Mongolia	3398373	1011489.0
26	Nepal	30547586	1003450.0
32	Saudi Arabia	36408824	841469.0
2	Azerbaijan	10358078	835757.0
3	Bahrain	1472237	696614.0
35	Sri Lanka	21832150	672798.0
18	Kuwait	4268886	667290.0
25	Myanmar	54179312	642885.0
31	Qatar	2695131	514524.0
1	Armenia	2780472	452273.0
28	Oman	4576300	399449.0
6	Brunei	449002	347723.0
0	Afghanistan	41128772	235214.0
20	Laos	7529477	219060.0
23	Maldives	523798	186694.0
42	Uzbekistan	34627648	175081.0
7	Cambodia	16767851	139319.0
19	Kyrgyzstan	6630621	88953.0
5	Bhutan	782457	62697.0
36	Syria	22125242	57423.0
37	Tajikistan	9952789	17786.0
44	Yemen	33696612	11945.0
27	North Korea	26069416	0.0
40	Turkmenistan	6430777	0.0

In [ ]:
```
"""
In ASIA China had the highest record, follow by India. But down in the list North Korea and Turkmenistan has 0 record which could account for NORTH AMERICA leading
ASIA in total cases.

So it is safe to say our data is fair.
"""
```

In [ ]:
```
"""
Further more, relative to India that followed China on the table, there is a wdie a wide gap of over 54M cases.
The gap seems too wide, so lets see what happened in China.

"""
```

In [86]:
```
data_china = data[data['location']=='China']
data_china
```

Out[86]:

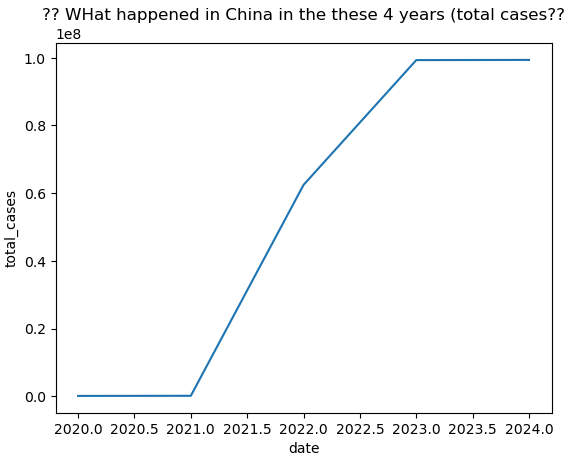
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_per_th
73670	CHN	Asia	China	2020-01-05	1.0	1.0	2510.397659	0.0	0.0	22.856527	...	48.4	51.894546	
73671	CHN	Asia	China	2020-01-06	1.0	0.0	2510.397659	0.0	0.0	22.856527	...	48.4	51.894546	
73672	CHN	Asia	China	2020-01-07	1.0	0.0	2510.397659	0.0	0.0	22.856527	...	48.4	51.894546	
73673	CHN	Asia	China	2020-01-08	1.0	0.0	2510.397659	0.0	0.0	22.856527	...	48.4	51.894546	
73674	CHN	Asia	China	2020-01-09	1.0	0.0	2510.397659	0.0	0.0	22.856527	...	48.4	51.894546	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
75339	CHN	Asia	China	2024-07-31	99371132.0	0.0	300.429000	122289.0	0.0	1.286000	...	48.4	51.894546	
75340	CHN	Asia	China	2024-08-01	99371132.0	0.0	300.429000	122289.0	0.0	1.286000	...	48.4	51.894546	
75341	CHN	Asia	China	2024-08-02	99371132.0	0.0	300.429000	122289.0	0.0	1.286000	...	48.4	51.894546	
75342	CHN	Asia	China	2024-08-03	99371132.0	0.0	300.429000	122289.0	0.0	1.286000	...	48.4	51.894546	
75343	CHN	Asia	China	2024-08-04	99373219.0	2087.0	298.143000	122304.0	15.0	2.143000	...	48.4	51.894546	

1674 rows × 67 columns

In [89]:
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_asia[data_asia['location']=='China'],
             x=data_asia['date'].dt.year,
             y='total_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in China in the these 4 years (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\378021079.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_asia[data_asia['location']=='China'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[89]: Text(0.5, 1.0, '?? WHat happened in China in the these 4 years (total cases??')

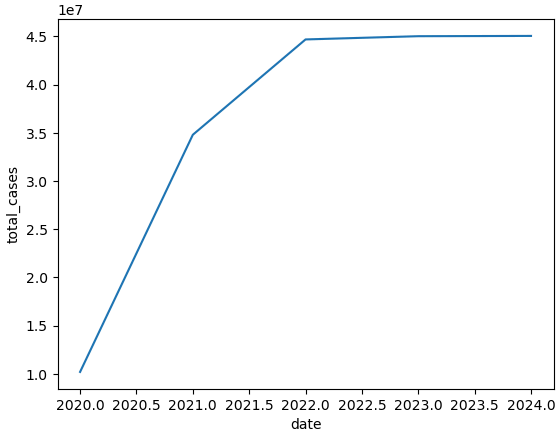


In [91]:
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_asia[data_asia['location']=='India'],
             x=data_asia['date'].dt.year,
             y='total_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in India in the these 4 years (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\3374676466.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_asia[data_asia['location']=='India'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[91]: Text(0.5, 1.0, '?? WHat happened in India in the these 4 years (total cases??')

?? WHat happened in India in the these 4 years (total cases??

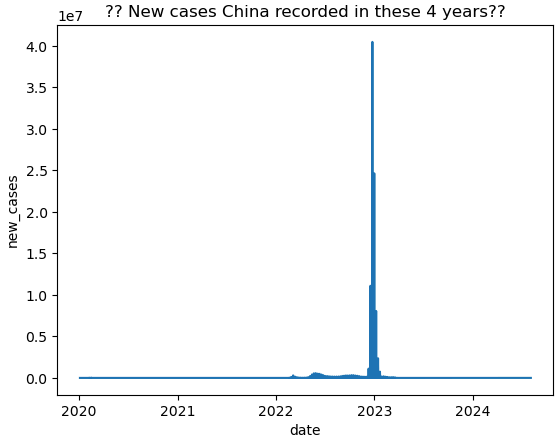


In [ ]:
```
"""
Around 2022 to 2023, there was a significant increase in total cases in China.
Let's see the new cases flow.

"""
```

In [92]:
```
# New cases rate in China

# How cases in each continents progressed over the years

sns.lineplot(data=data_china,
                x=data_china['date'],
                y='new_cases',
#             hue='continent',
#             estimator='sum',
             ci=None
               ).set_title('?? New cases China recorded in these 4 years??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\3433645661.py:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_china,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[92]:
```
Text(0.5, 1.0, '?? New cases China recorded in these 4 years??')
```

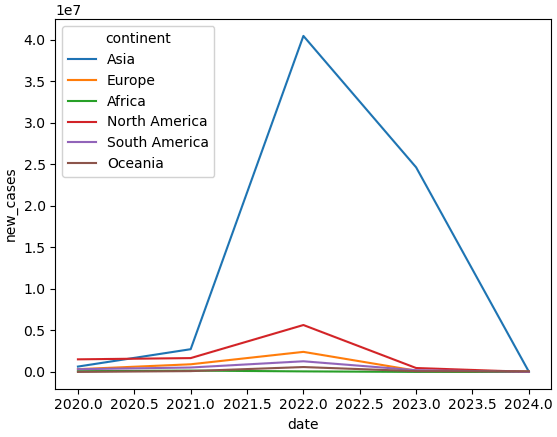


In [ ]:
```
"""
2023 seems so much bad in China.

"""
```

In [93]:
```
# NEW CASES

sns.lineplot(data=data,
                x=data['date'].dt.year,
                y='new_cases',
             hue='continent',
             estimator='max',
             ci=None
               )
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\1479782706.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

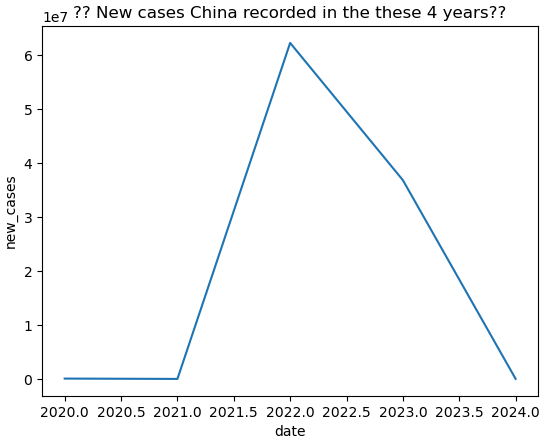
Out[93]: <Axes: xlabel='date', ylabel='new_cases'>



In [94]:
```
# New cases rate in China

# How cases in each continents progressed over the years

sns.lineplot(data=data_china,
                x=data_china['date'].dt.year,
                y='new_cases',
#             hue='continent',
             estimator='sum',
             ci=None
               ).set_title('?? New cases China recorded in the these 4 years??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\391012660.py:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_china,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[94]: Text(0.5, 1.0, '?? New cases China recorded in the these 4 years??')



In [ ]:
```
"""
It was China that increased the new cases erecord of ASIA in 2022 and 2023

"""
```

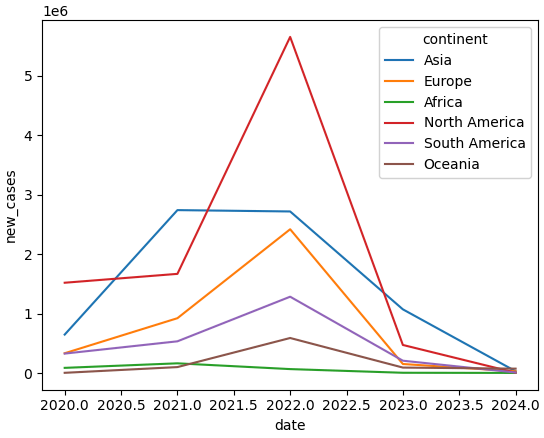
In [ ]:
```
"""
Over laying this immediate plot and the previous one, it shows that every day in 2023, many new cases were recorded but at the end of the year it was not as many as 2022.

We'll check what happened in 2023, why there was higher daily record of new cases.
"""
```

In [95]:
```
# NEW CASES without CHINA

sns.lineplot(data=data[data['location']!='China'],
                x=data['date'].dt.year,
                y='new_cases',
             hue='continent',
             estimator='max',
```

```
            ci=None
              )
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\3026413998.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data[data['location']!='China'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[95]: <Axes: xlabel='date', ylabel='new_cases'>

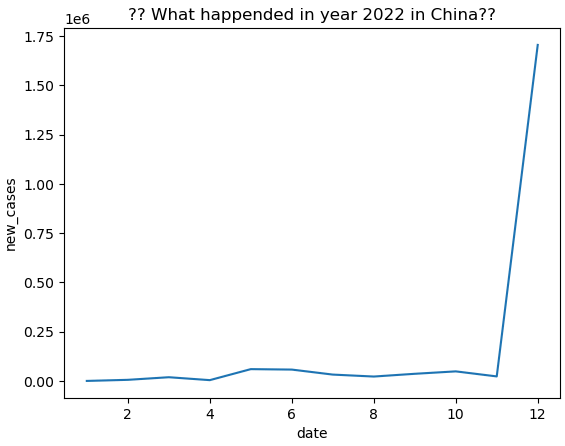


In [ ]:
```
"""
Apparently NORTH AMERICA ALSO HAD A SPIKE IN NEW CASES IN 2022.

"""
```

In [96]:
```
# New cases rate in China

# How cases in each continents progressed over the years

sns.lineplot(data=data_china,
               x=data_china[data_china['date'].dt.year==2022]['date'].dt.month,
               y='new_cases',
#             hue='continent',
#             estimator='sum',
             ci=None
              ).set_title('?? What happended in year 2022 in China??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\1468045646.py:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_china,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

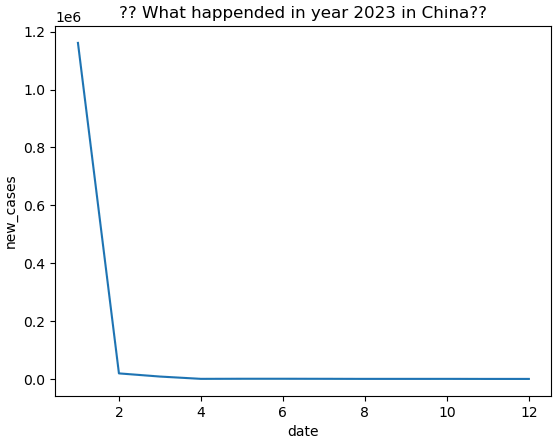
Out[96]: Text(0.5, 1.0, '?? What happended in year 2022 in China??')



In [97]:
```
# New cases rate in China

# How cases in each continents progressed over the years

sns.lineplot(data=data_china,
               x=data_china[data_china['date'].dt.year==2023]['date'].dt.month,
               y='new_cases',
#             hue='continent',
#             estimator='sum',
             ci=None
              ).set_title('?? What happended in year 2023 in China??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\288144419.py:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_china,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[97]: Text(0.5, 1.0, '?? What happended in year 2023 in China??')

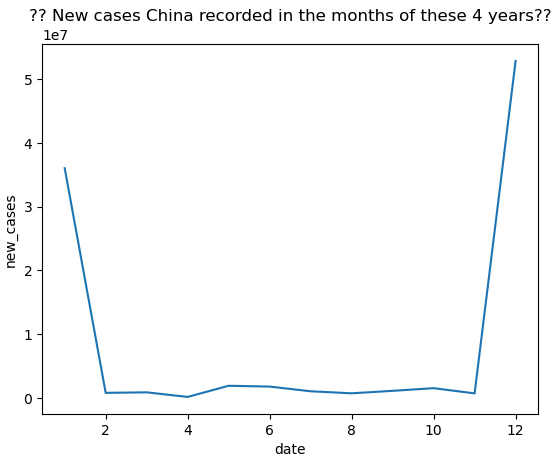


In [ ]:
```
"""
So the most new cases were recorded in December and JAnuary.

December 2022 to January 2023::: Festive periods which most likely there was lot of travelling and gathering which must have increased the numbers of people per square meter,
hence increasing the spread of the virus.
Also, as confirmed, it wad this period there was new variants of COVID which was faster spreading.

Below PLot says it much too.
"""
```

In [98]:
```
# New cases rate in China

# How cases in each continents progressed over the years

sns.lineplot(data=data_china,
               x=data_china['date'].dt.month,
               y='new_cases',
#              hue='continent',
             estimator='sum',
             ci=None
               ).set_title('?? New cases China recorded in the months of these 4 years??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\664135755.py:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_china,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[98]: Text(0.5, 1.0, '?? New cases China recorded in the months of these 4 years??')



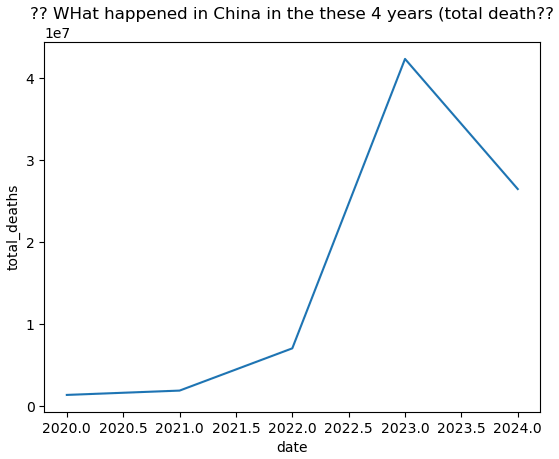
In [ ]:
```
"""
So around December 2022 to January 2023, there was a large increase in the numbers of new caese in China.


You can check the numbers in the table to confirm.

So we can say China caused the high increase in the ASIA record and it was marjorly as December and January.
```

```
"""
```

In [99]:
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_asia[data_asia['location']=='China'],
                x=data_asia['date'].dt.year,
                y='total_deaths',
#             hue='continent',
            estimator='sum',
            ci=None
              ).set_title('?? WHat happened in China in the these 4 years (total death??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\983588013.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_asia[data_asia['location']=='China'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[99]:
```
Text(0.5, 1.0, '?? WHat happened in China in the these 4 years (total death??')
```



In [ ]:
```
"""
Mortality rate increase about the same years too, 2022 and 2023.
Shall we check our correlation chart of death cases again.

"""
```

In [100…
```
# CORRELATION FOR ASIA ALONE

corr_list = []

for col in numerical_columns:
    corr_list.append(data_asia['total_deaths'].corr(data_asia[col]))

corr_df = pd.DataFrame(corr_list, numerical_columns.columns, columns=['Correlation Value']).reset_index()
```

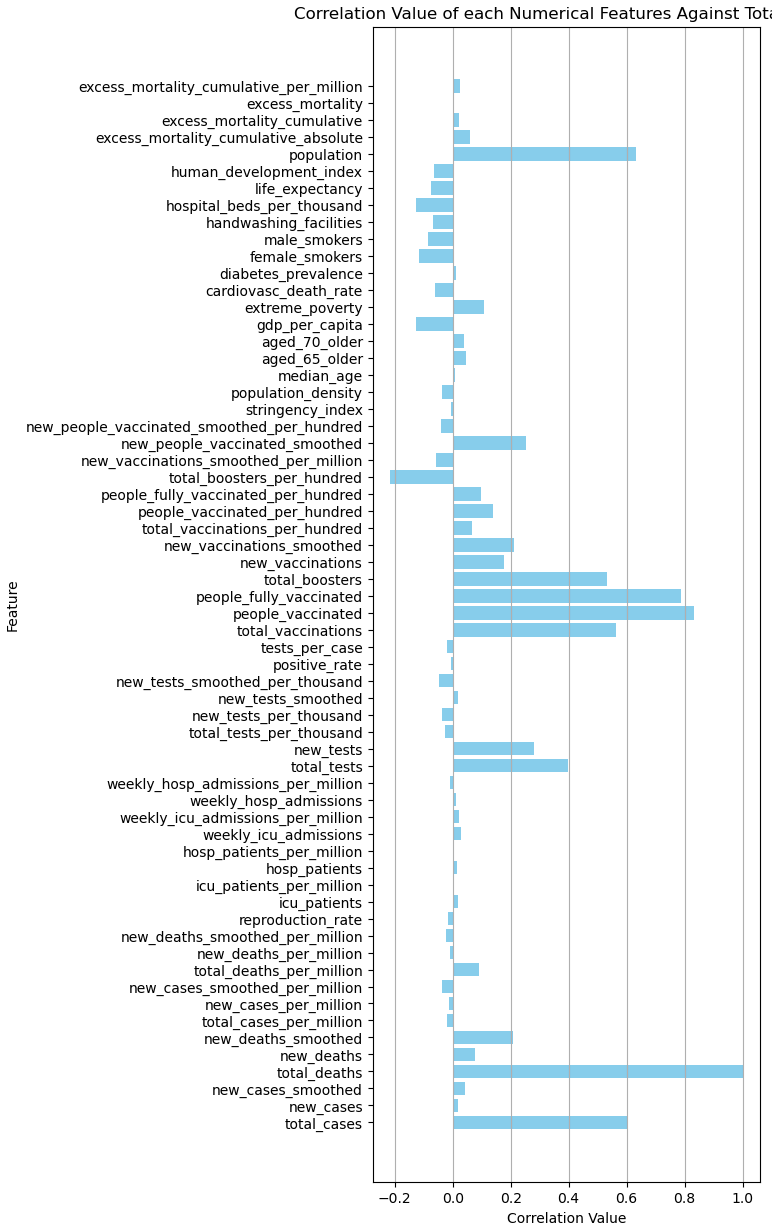
In [101…
```
corr_df
```

Out[101…

	index	Correlation Value
0	total_cases	0.602996
1	new_cases	0.016518
2	new_cases_smoothed	0.043726
3	total_deaths	1.000000
4	new_deaths	0.078338
...	...	...
57	population	0.631206
58	excess_mortality_cumulative_absolute	0.060708
59	excess_mortality_cumulative	0.021877
60	excess_mortality	0.003912
61	excess_mortality_cumulative_per_million	0.025656

62 rows × 2 columns

In [102…
```
plt.figure(figsize=(5, 15))
plt.barh(numerical_columns.columns, corr_df['Correlation Value'], color='skyblue')
plt.xlabel('Correlation Value')
plt.ylabel('Feature')
plt.title('Correlation Value of each Numerical Features Against Total Death')
plt.grid(axis='x')
plt.show()
```



In [ ]:
```
"""
IS IT WORTH NOTHING THE SIGNIFICANT CORRELATION VALUE OF OVER 0.65 BETWEEN TOTAL DEATH IN ASIA AND NUMBERS VACCINATED???

We could check CAUSALITY to confirm that.

"""
```

In [103…
```
# OLS

#data_asia = data_asia[numerical_columns]

data_asia
```

Out[103…

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	...	male_smokers	handwashing_facilities	hospital_beds_p
0	AFG	Asia	Afghanistan	2020-01-05	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
1	AFG	Asia	Afghanistan	2020-01-06	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
2	AFG	Asia	Afghanistan	2020-01-07	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
3	AFG	Asia	Afghanistan	2020-01-08	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
4	AFG	Asia	Afghanistan	2020-01-09	0.0	0.0	2510.397659	0.0	0.0	22.856527	...	32.655006	37.746	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
426082	YEM	Asia	Yemen	2024-07-31	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426083	YEM	Asia	Yemen	2024-08-01	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426084	YEM	Asia	Yemen	2024-08-02	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426085	YEM	Asia	Yemen	2024-08-03	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	
426086	YEM	Asia	Yemen	2024-08-04	11945.0	0.0	0.000000	2159.0	0.0	0.000000	...	29.200000	49.542	

75349 rows × 67 columns

In [104…

```
# OLS
import statsmodels.api as sm

X = data_asia.drop(['iso_code','continent','location','date','total_deaths','tests_units'], axis=1)
X = sm.add_constant(X)
y = data_asia['total_deaths']
```

In [105…

```
# Split your data set into 80/20 for train/test datasets

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)
```

In [106…

```
# create a fitted model & print the summary

y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()
```

In [107…

```
print(lm.summary(xname=list(X.columns)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.845
Model:                            OLS   Adj. R-squared:                  0.844
Method:                 Least Squares   F-statistic:                     5361.
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        00:17:56   Log-Likelihood:            -7.0099e+05
No. Observations:               60279   AIC:                         1.402e+06
Df Residuals:                   60217   BIC:                         1.403e+06
Df Model:                          61
Covariance Type:            nonrobust
=============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------
const                                     2.199e+04   4954.082      4.440      0.000    1.23e+04    3.17e+04
total_cases                                  0.0016   1.64e-05     99.441      0.000       0.002       0.002
new_cases                                   -0.0020      0.001     -1.905      0.057      -0.004    5.79e-05
new_cases_smoothed                          -0.0337      0.002    -15.097      0.000      -0.038      -0.029
new_deaths                                  -0.9345      0.348     -2.683      0.007      -1.617      -0.252
new_deaths_smoothed                         26.1218      1.041     25.100      0.000      24.082      28.162
total_cases_per_million                     -0.0582      0.001    -48.439      0.000      -0.061      -0.056
new_cases_per_million                        0.3203      0.146      2.187      0.029       0.033       0.607
new_cases_smoothed_per_million               0.4339      0.422      1.028      0.304      -0.393       1.261
total_deaths_per_million                    15.6387      0.248     63.044      0.000      15.152      16.125
new_deaths_per_million                     -12.0837     38.482     -0.314      0.754     -87.508      63.340
new_deaths_smoothed_per_million           -798.4554    112.059     -7.125      0.000   -1018.091    -578.820
reproduction_rate                          564.3465    394.611      1.430      0.153    -209.091    1337.785
icu_patients                               -10.3201      3.210     -3.215      0.001     -16.611      -4.029
icu_patients_per_million                   320.2259    148.281      2.160      0.031      29.595     610.857
hosp_patients                                1.9189      0.244      7.862      0.000       1.441       2.397
hosp_patients_per_million                 -114.2897     17.142     -6.667      0.000    -147.888     -80.691
weekly_icu_admissions                      152.6398     11.148     13.692      0.000     130.789     174.490
weekly_icu_admissions_per_million        -1789.7627    365.819     -4.892      0.000   -2506.770   -1072.756
weekly_hosp_admissions                      -1.8626      0.425     -4.385      0.000      -2.695      -1.030
weekly_hosp_admissions_per_million          99.6344     14.461      6.890      0.000      71.292     127.977
total_tests                              7.874e-06   2.26e-06      3.486      0.000    3.45e-06    1.23e-05
new_tests                                    0.0120      0.001     16.984      0.000       0.011       0.013
total_tests_per_thousand                     0.7517      0.192      3.917      0.000       0.376       1.128
new_tests_per_thousand                      74.7395     57.136      1.308      0.191     -37.247     186.726
new_tests_smoothed                          -0.0047      0.000    -22.907      0.000      -0.005      -0.004
new_tests_smoothed_per_thousand           -216.0801     72.824     -2.967      0.003    -358.815     -73.345
positive_rate                            1.089e+04   2227.667      4.890      0.000    6526.298    1.53e+04
tests_per_case                               0.0499      0.006      8.634      0.000       0.039       0.061
total_vaccinations                       -2.402e-06   7.03e-07     -3.415      0.001   -3.78e-06   -1.02e-06
people_vaccinated                            0.0004   4.57e-06     84.785      0.000       0.000       0.000
people_fully_vaccinated                  7.346e-05   5.04e-06     14.578      0.000    6.36e-05    8.33e-05
total_boosters                              -0.0004   7.58e-06    -54.136      0.000      -0.000      -0.000
new_vaccinations                            -0.0083      0.000    -18.343      0.000      -0.009      -0.007
new_vaccinations_smoothed                    0.0051      0.001      9.543      0.000       0.004       0.006
total_vaccinations_per_hundred              -47.5486      8.607     -5.524      0.000     -64.418     -30.679
people_vaccinated_per_hundred               -12.6772     25.805     -0.491      0.623     -63.255      37.901
people_fully_vaccinated_per_hundred         -45.1078     23.590     -1.912      0.056     -91.344       1.128
total_boosters_per_hundred                  261.4317     17.529     14.914      0.000     227.074     295.789
new_vaccinations_smoothed_per_million        -0.3447      0.079     -4.366      0.000      -0.499      -0.190
new_people_vaccinated_smoothed               0.0185      0.001     27.409      0.000       0.017       0.020
new_people_vaccinated_smoothed_per_hundred -2600.5797   1189.578     -2.186      0.029   -4932.157    -269.002
stringency_index                            -71.9758      6.631    -10.855      0.000     -84.972     -58.979
population_density                           -0.1819      0.122     -1.493      0.135      -0.421       0.057
median_age                                -1475.9977     58.640    -25.170      0.000   -1590.933   -1361.063
aged_65_older                               251.6258    161.559      1.557      0.119     -65.030     568.282
aged_70_older                             -1150.0640    248.368     -4.630      0.000   -1636.866    -663.262
gdp_per_capita                               -0.1063      0.010    -10.975      0.000      -0.125      -0.087
extreme_poverty                              81.4181     23.275      3.498      0.000      35.798     127.038
cardiovasc_death_rate                       -46.6813      1.278    -36.518      0.000     -49.187     -44.176
diabetes_prevalence                       -2903.5876     69.142    -41.994      0.000   -3039.107   -2768.068
female_smokers                             -175.6157     31.423     -5.589      0.000    -237.205    -114.026
male_smokers                                -40.3471     12.669     -3.185      0.001     -65.179     -15.515
handwashing_facilities                     -210.4667      7.468    -28.182      0.000    -225.104    -195.829
hospital_beds_per_thousand                -2231.9399     70.651    -31.591      0.000   -2370.417   -2093.463
life_expectancy                           -1079.2477     71.252    -15.147      0.000   -1218.903    -939.593
human_development_index                   1.828e+05   3408.342     53.645      0.000    1.76e+05    1.9e+05
population                                3.344e-05   8.57e-07     39.002      0.000    3.18e-05    3.51e-05
excess_mortality_cumulative_absolute         0.2374      0.012     19.662      0.000       0.214       0.261
excess_mortality_cumulative                529.2040     88.191      6.001      0.000     356.350     702.058
excess_mortality                           -28.3002     34.658     -0.817      0.414     -96.230      39.630
excess_mortality_cumulative_per_million     -5.7211      0.595     -9.620      0.000      -6.887      -4.555
==============================================================================
Omnibus:                    25825.233   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         10480877.857
Skew:                           0.749   Prob(JB):                         0.00
Kurtosis:                      67.581   Cond. No.                     2.12e+10
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.12e+10. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:
```
"""
From above we can see that some variables have effect and some does not but before we try to get insights, let's fine tune the model some bit
by looking at the statistical tests and dropping variables with p-value greater than 0.05.

new_cases
new_cases_smoothed_per_million
new_deaths_per_million
reproduction_rate
new_tests_per_thousand
people_vaccinated_per_hundred
people_fully_vaccinated_per_hundred
population_density
aged_65_older
excess_mortality


"""
```

In [108…
```
# OLS
import statsmodels.api as sm

X = data_asia.drop(['iso_code','continent','location','date','total_deaths','tests_units',
                    'new_cases', 'new_cases_smoothed_per_million', 'new_deaths_per_million',
                   'reproduction_rate', 'new_tests_per_thousand', 'people_vaccinated_per_hundred',
```

```
                  'people_fully_vaccinated_per_hundred', 'population_density', 'aged_65_older',
                  'excess_mortality'], axis=1)
X = sm.add_constant(X)
y = data_asia['total_deaths']


# Split your data set into 80/20 for train/test datasets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)


# create a fitted model & print the summary
y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()


print(lm.summary(xname=list(X.columns)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.844
Model:                            OLS   Adj. R-squared:                  0.844
Method:                 Least Squares   F-statistic:                     6411.
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        00:20:05   Log-Likelihood:            -7.0100e+05
No. Observations:               60279   AIC:                             1.402e+06
Df Residuals:                   60227   BIC:                             1.403e+06
Df Model:                          51
Covariance Type:            nonrobust
=============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------
const                                     2.168e+04   4763.456      4.552      0.000    1.23e+04     3.1e+04
total_cases                                  0.0016   1.62e-05    100.818      0.000       0.002       0.002
new_cases_smoothed                          -0.0344      0.002    -16.358      0.000      -0.038      -0.030
new_deaths                                  -1.1890      0.319     -3.729      0.000      -1.814      -0.564
new_deaths_smoothed                         26.2751      1.033     25.444      0.000      24.251      28.299
total_cases_per_million                     -0.0586      0.001    -49.849      0.000      -0.061      -0.056
new_cases_per_million                        0.2749      0.124      2.216      0.027       0.032       0.518
total_deaths_per_million                    15.6675      0.241     64.927      0.000      15.195      16.140
new_deaths_smoothed_per_million           -783.5372    100.384     -7.805      0.000    -980.291    -586.783
icu_patients                               -10.1066      3.197     -3.162      0.002     -16.372      -3.841
icu_patients_per_million                   281.0037    147.313      1.908      0.056      -7.729     569.737
hosp_patients                                1.8989      0.244      7.793      0.000       1.421       2.377
hosp_patients_per_million                 -112.2742     17.101     -6.565      0.000    -145.792     -78.757
weekly_icu_admissions                      149.5561     10.972     13.630      0.000     128.050     171.062
weekly_icu_admissions_per_million        -1702.3561    361.577     -4.708      0.000   -2411.049    -993.664
weekly_hosp_admissions                       -1.8315      0.419     -4.366      0.000      -2.654      -1.009
weekly_hosp_admissions_per_million          102.1898     14.186      7.204      0.000      74.386     129.994
total_tests                                7.877e-06   2.26e-06      3.489      0.000    3.45e-06    1.23e-05
new_tests                                    0.0121      0.001     17.183      0.000       0.011       0.013
total_tests_per_thousand                     0.7961      0.188      4.233      0.000       0.427       1.165
new_tests_smoothed                          -0.0047      0.000    -23.120      0.000      -0.005      -0.004
new_tests_smoothed_per_thousand           -153.5457     57.816     -2.656      0.008    -266.865     -40.226
positive_rate                             1.163e+04   2155.755      5.393      0.000    7399.819    1.59e+04
tests_per_case                               0.0489      0.006      8.472      0.000       0.038       0.060
total_vaccinations                        -1.727e-06   6.43e-07     -2.688      0.007   -2.99e-06   -4.68e-07
people_vaccinated                            0.0004   4.37e-06     88.803      0.000       0.000       0.000
people_fully_vaccinated                   7.113e-05   4.82e-06     14.762      0.000    6.17e-05    8.06e-05
total_boosters                              -0.0004   7.56e-06    -54.340      0.000      -0.000      -0.000
new_vaccinations                            -0.0083      0.000    -18.305      0.000      -0.009      -0.007
new_vaccinations_smoothed                    0.0050      0.001      9.356      0.000       0.004       0.006
total_vaccinations_per_hundred              -66.0848      3.836    -17.228      0.000     -73.603     -58.566
total_boosters_per_hundred                  273.7898     16.269     16.829      0.000     241.902     305.677
new_vaccinations_smoothed_per_million        -0.3460      0.078     -4.434      0.000      -0.499      -0.193
new_people_vaccinated_smoothed               0.0186      0.001     27.641      0.000       0.017       0.020
new_people_vaccinated_smoothed_per_hundred -2623.2963   1188.307     -2.208      0.027   -4952.383    -294.210
stringency_index                            -69.5250      6.426    -10.820      0.000     -82.119     -56.931
median_age                                -1480.8306     57.540    -25.736      0.000   -1593.610   -1368.051
aged_70_older                              -785.9085     96.138     -8.175      0.000    -974.340    -597.477
gdp_per_capita                               -0.1029      0.009    -10.949      0.000      -0.121      -0.084
extreme_poverty                              70.5324     22.728      3.103      0.002      25.986     115.079
cardiovasc_death_rate                       -46.5354      1.266    -36.750      0.000     -49.017     -44.054
diabetes_prevalence                       -2902.3367     66.307    -43.771      0.000   -3032.299   -2772.374
female_smokers                             -173.0724     30.636     -5.649      0.000    -233.120    -113.025
male_smokers                                -40.7574     12.501     -3.260      0.001     -65.260     -16.255
handwashing_facilities                     -210.7695      7.447    -28.304      0.000    -225.365    -196.174
hospital_beds_per_thousand                -2218.5272     66.236    -33.495      0.000   -2348.349   -2088.705
life_expectancy                           -1063.4675     68.905    -15.434      0.000   -1198.522    -928.413
human_development_index                    1.811e+05   3298.685     54.915      0.000    1.75e+05    1.88e+05
population                                 3.359e-05    8.5e-07     39.510      0.000    3.19e-05    3.53e-05
excess_mortality_cumulative_absolute          0.2378      0.012     19.698      0.000       0.214       0.261
excess_mortality_cumulative                 497.2624     79.695      6.240      0.000     341.060     653.465
excess_mortality_cumulative_per_million      -5.6341      0.584     -9.655      0.000      -6.778      -4.490
==============================================================================
Omnibus:                    25855.141   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         10464118.445
Skew:                           0.753   Prob(JB):                         0.00
Kurtosis:                      67.529   Cond. No.                     2.05e+10
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.05e+10. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:
```
"""
Furthermore,
icu_patients_per_million
"""
```

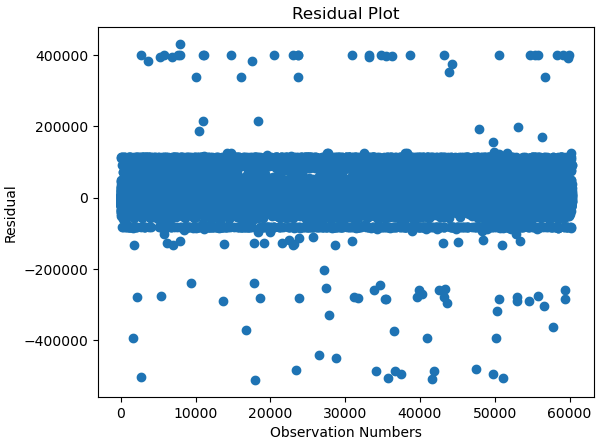
In [109…
```
# OLS
import statsmodels.api as sm

X = data_asia.drop(['iso_code','continent','location','date','total_deaths','tests_units',
                    'new_cases', 'new_cases_smoothed_per_million', 'new_deaths_per_million',
                  'reproduction_rate', 'new_tests_per_thousand', 'people_vaccinated_per_hundred',
                  'people_fully_vaccinated_per_hundred', 'population_density', 'aged_65_older',
                  'excess_mortality', 'icu_patients_per_million'], axis=1)
X = sm.add_constant(X)
```

```
y = data_asia['total_deaths']


# Split your data set into 80/20 for train/test datasets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)


# create a fitted model & print the summary
y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()


print(lm.summary(xname=list(X.columns)))
```

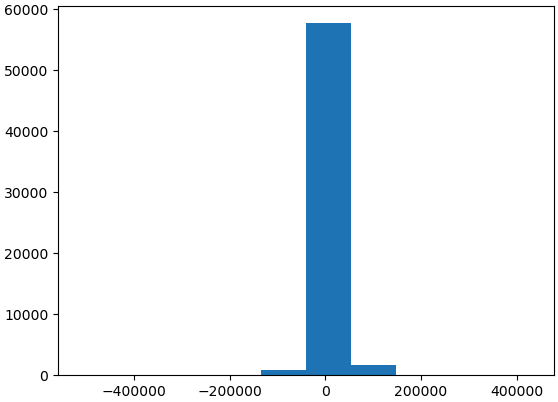
```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.844
Model:                            OLS   Adj. R-squared:                  0.844
Method:                 Least Squares   F-statistic:                     6539.
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        00:21:21   Log-Likelihood:            -7.0101e+05
No. Observations:               60279   AIC:                         1.402e+06
Df Residuals:                   60228   BIC:                         1.403e+06
Df Model:                          50
Covariance Type:            nonrobust
================================================================================================================
                                                   coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------------------------
const                                       2.212e+04   4758.192      4.648      0.000    1.28e+04    3.14e+04
total_cases                                    0.0016   1.61e-05    101.117      0.000       0.002       0.002
new_cases_smoothed                            -0.0344      0.002    -16.375      0.000      -0.039      -0.030
new_deaths                                    -1.1868      0.319     -3.722      0.000      -1.812      -0.562
new_deaths_smoothed                           26.2762      1.033     25.444      0.000      24.252      28.300
total_cases_per_million                       -0.0586      0.001    -49.832      0.000      -0.061      -0.056
new_cases_per_million                          0.2602      0.124      2.102      0.036       0.018       0.503
total_deaths_per_million                      15.6636      0.241     64.912      0.000      15.191      16.137
new_deaths_smoothed_per_million             -772.8279    100.230     -7.711      0.000    -969.278    -576.378
icu_patients                                  -4.8280      1.600     -3.017      0.003      -7.964      -1.692
hosp_patients                                  1.6239      0.196      8.266      0.000       1.239       2.009
hosp_patients_per_million                    -97.8949     15.350     -6.378      0.000    -127.981     -67.809
weekly_icu_admissions                        140.6659      9.934     14.161      0.000     121.196     160.136
weekly_icu_admissions_per_million          -1405.2795    326.325     -4.306      0.000   -2044.878    -765.681
weekly_hosp_admissions                        -1.8736      0.419     -4.473      0.000      -2.695      -1.053
weekly_hosp_admissions_per_million           106.8204     13.977      7.643      0.000      79.426     134.215
total_tests                                7.889e-06   2.26e-06      3.494      0.000    3.46e-06    1.23e-05
new_tests                                      0.0121      0.001     17.169      0.000       0.011       0.013
total_tests_per_thousand                       0.8269      0.187      4.413      0.000       0.460       1.194
new_tests_smoothed                            -0.0047      0.000    -23.083      0.000      -0.005      -0.004
new_tests_smoothed_per_thousand             -161.6300     57.662     -2.803      0.005    -274.647     -48.613
positive_rate                              1.149e+04   2154.709      5.334      0.000    7270.943    1.57e+04
tests_per_case                                 0.0490      0.006      8.488      0.000       0.038       0.060
total_vaccinations                         -1.691e-06   6.42e-07     -2.633      0.008   -2.95e-06   -4.32e-07
people_vaccinated                              0.0004   4.37e-06     88.787      0.000       0.000       0.000
people_fully_vaccinated                     7.106e-05   4.82e-06     14.747      0.000    6.16e-05    8.05e-05
total_boosters                                -0.0004   7.56e-06    -54.314      0.000      -0.000      -0.000
new_vaccinations                              -0.0083      0.000    -18.311      0.000      -0.009      -0.007
new_vaccinations_smoothed                      0.0050      0.001      9.353      0.000       0.004       0.006
total_vaccinations_per_hundred               -66.2237      3.835    -17.267      0.000     -73.741     -58.706
total_boosters_per_hundred                   274.3696     16.267     16.867      0.000     242.487     306.252
new_vaccinations_smoothed_per_million         -0.3395      0.078     -4.355      0.000      -0.492      -0.187
new_people_vaccinated_smoothed                 0.0186      0.001     27.625      0.000       0.017       0.020
new_people_vaccinated_smoothed_per_hundred -2654.4026   1188.221     -2.234      0.025   -4983.320    -325.485
stringency_index                             -69.1829      6.423    -10.771      0.000     -81.772     -56.593
median_age                                 -1486.3302     57.469    -25.863      0.000   -1598.970   -1373.690
aged_70_older                               -774.7373     95.962     -8.073      0.000    -962.823    -586.652
gdp_per_capita                                -0.1029      0.009    -10.957      0.000      -0.121      -0.085
extreme_poverty                               71.3654     22.724      3.140      0.002      26.826     115.905
cardiovasc_death_rate                        -46.5200      1.266    -36.738      0.000     -49.002     -44.038
diabetes_prevalence                        -2896.3942     66.236    -43.729      0.000   -3026.216   -2766.572
female_smokers                               -174.4684     30.628     -5.696      0.000    -234.500    -114.437
male_smokers                                  -40.6564     12.501     -3.252      0.001     -65.159     -16.154
handwashing_facilities                       -210.3243      7.443    -28.257      0.000    -224.913    -195.736
hospital_beds_per_thousand                 -2226.2237     66.114    -33.672      0.000   -2355.807   -2096.640
life_expectancy                            -1068.8403     68.849    -15.524      0.000   -1203.785    -933.896
human_development_index                     1.813e+05   3297.271     54.996      0.000    1.75e+05    1.88e+05
population                                  3.354e-05    8.5e-07     39.467      0.000    3.19e-05    3.52e-05
excess_mortality_cumulative_absolute           0.2379      0.012     19.713      0.000       0.214       0.262
excess_mortality_cumulative                  499.3267     79.690      6.266      0.000     343.135     655.518
excess_mortality_cumulative_per_million       -5.6318      0.584     -9.650      0.000      -6.776      -4.488
==============================================================================
Omnibus:                    25854.727   Durbin-Watson:                   1.994
Prob(Omnibus):                  0.000   Jarque-Bera (JB):         10449449.729
Skew:                           0.753   Prob(JB):                         0.00
Kurtosis:                      67.484   Cond. No.                     2.04e+10
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.04e+10. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [333…

```
# plot to check homoscedasticity  and Normality
plt.plot(lm.resid,'o')
plt.title('Residual Plot')
plt.ylabel('Residual')
plt.xlabel('Observation Numbers')
plt.show()
plt.hist(lm.resid,)
#plt.hist()
```



Out[333… (array([1.4000e+01, 7.0000e+00, 3.0000e+01, 1.0000e+00, 8.4900e+02,
5.7693e+04, 1.6400e+03, 7.0000e+00, 0.0000e+00, 3.8000e+01]),
array([-510871.33853117, -416705.10981711, -322538.88110304,
-228372.65238897, -134206.4236749 , -40040.19496084,
54126.03375323, 148292.2624673 , 242458.49118136,
336624.71989543, 430790.9486095]),
<BarContainer object of 10 artists>)



In [ ]:
```
"""
With the most Residual plot as along a straight, we can say it's quite safe to draw conclusion from this model.
Our Durbin-Watson almost 2 shows no autocorrelation but our Condition no. too high shows possibility of multicollinearity
which is quite fair because some predictors are just ratio of another.

"""
```

In [ ]:
```
"""
With our model accuracy up tp .84 and no redundant variable or p-value greater than .05.
We can look at the coefficient and pick some variables that stands out.

Since we are trying to look at the variables that contributed to increase in the death rate, so we'll pick predictors with
positive coefficients instead.

new_deaths_smoothed                          26.2762
total_deaths_per_million                     15.6636
weekly_icu_admissions                       140.6659
weekly_hosp_admissions_per_million          106.8204
new_tests                                     0.0121
total_tests_per_thousand                      0.8269
positive_rate                           1.149e+04
tests_per_case                                0.0490
people_vaccinated                             0.0004
new_vaccinations_smoothed                     0.0050
total_boosters_per_hundred                  274.3696
new_people_vaccinated_smoothed                0.0186
extreme_poverty                              71.3654
human_development_index                 1.813e+05
population                              3.354e-05
excess_mortality_cumulative                 499.3267



PS::: Do you know how to automatically get the variables out without manually copying. lol. I am lazy.
please let me know.
"""
```

In [ ]:
```
"""
For me, what stands out most are

extreme_poverty                              71.3654
human_development_index                 1.813e+05
total_boosters_per_hundred                  274.3696

Others seems to be CASUALITY EFFECT!


"""
```

In [111…
```
# Pivot table to see the total cases by continents
indicators = ['population','total_cases','total_deaths','extreme_poverty',
              'human_development_index','total_boosters_per_hundred']
```

```
pivot_all = data_asia.pivot_table(index=['location'],
                                  values=indicators,
                                  aggfunc='max').reset_index()
pivot_all.sort_values(by='total_deaths', ascending=False)
```

Out[111…

	location	extreme_poverty	human_development_index	population	total_boosters_per_hundred	total_cases	total_deaths
10	India	21.200000	0.645000	1417173120	36.983361	45041748.0	533623.0
11	Indonesia	5.700000	0.718000	275501344	36.983361	6829399.0	162059.0
12	Iran	0.200000	0.783000	88550568	36.983361	7627863.0	146837.0
8	China	0.700000	0.761000	1425887360	57.990000	99373219.0	122304.0
39	Turkey	0.200000	0.820000	85341248	48.540000	17004718.0	101419.0
15	Japan	13.128331	0.919000	123951696	141.720000	33803572.0	74694.0
30	Philippines	13.128331	0.718000	115559008	36.983361	4173631.0	66864.0
43	Vietnam	2.000000	0.704000	98186856	59.050000	11624000.0	43206.0
22	Malaysia	0.100000	0.810000	33938216	50.640000	5309410.0	37351.0
34	South Korea	0.200000	0.916000	51815808	79.760000	34571873.0	35934.0
38	Thailand	0.100000	0.777000	71697024	44.830000	4799180.0	34715.0
29	Pakistan	4.000000	0.557000	235824864	36.983361	1580631.0	30656.0
4	Bangladesh	14.800000	0.632000	171186368	40.080000	2051348.0	29499.0
13	Iraq	2.500000	0.674000	44496124	36.983361	2465545.0	25375.0
25	Myanmar	6.400000	0.583000	54179312	36.983361	642885.0	19494.0
17	Kazakhstan	0.100000	0.825000	19397998	36.983361	1504370.0	19072.0
9	Georgia	4.200000	0.812000	3744385	36.983361	1863615.0	17150.0
35	Sri Lanka	0.700000	0.782000	21832150	37.650000	672798.0	16907.0
16	Jordan	0.100000	0.729000	11285875	36.983361	1746997.0	14122.0
14	Israel	0.500000	0.919000	9449000	61.030000	4841558.0	12707.0
26	Nepal	15.000000	0.602000	30547586	36.983361	1003450.0	12031.0
21	Lebanon	13.128331	0.744000	5489744	36.983361	1239904.0	10947.0
2	Azerbaijan	13.128331	0.756000	10358078	36.983361	835757.0	10353.0
32	Saudi Arabia	13.128331	0.854000	36408824	36.983361	841469.0	9646.0
1	Armenia	1.800000	0.776000	2780472	36.983361	452273.0	8777.0
0	Afghanistan	13.128331	0.511000	41128772	36.983361	235214.0	7998.0
28	Oman	13.128331	0.813000	4576300	36.983361	399449.0	4628.0
36	Syria	13.128331	0.567000	22125242	36.983361	57423.0	3163.0
7	Cambodia	13.128331	0.594000	16767851	64.310000	139319.0	3056.0
18	Kuwait	13.128331	0.806000	4268886	36.983361	667290.0	2570.0
41	United Arab Emirates	13.128331	0.890000	9441138	54.430000	1067030.0	2349.0
44	Yemen	18.800000	0.470000	33696612	36.983361	11945.0	2159.0
24	Mongolia	0.500000	0.737000	3398373	36.983361	1011489.0	2136.0
33	Singapore	13.128331	0.938000	5637022	78.770000	3006155.0	2024.0
3	Bahrain	13.128331	0.852000	1472237	68.510000	696614.0	1536.0
19	Kyrgyzstan	1.400000	0.697000	6630621	36.983361	88953.0	1024.0
42	Uzbekistan	13.128331	0.720000	34627648	47.400000	175081.0	1016.0
31	Qatar	13.128331	0.848000	2695131	70.680000	514524.0	690.0
20	Laos	22.700000	0.613000	7529477	36.983361	219060.0	671.0
23	Maldives	13.128331	0.740000	523798	36.983361	186694.0	316.0
6	Brunei	13.128331	0.838000	449002	75.830000	347723.0	179.0
37	Tajikistan	4.800000	0.668000	9952789	53.480000	17786.0	125.0
5	Bhutan	1.500000	0.654000	782457	81.110000	62697.0	21.0
27	North Korea	13.128331	0.724345	26069416	36.983361	0.0	0.0
40	Turkmenistan	13.128331	0.715000	6430777	59.350000	0.0	0.0

In [119…

```
# TOP 5 by death cases

top_5 = pivot_all.sort_values(by='total_deaths', ascending=False).head()
top_5
```

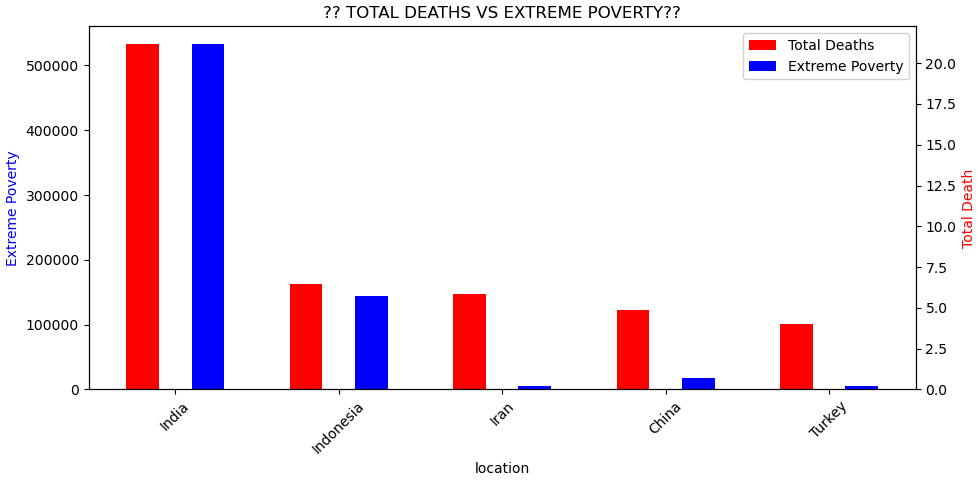
Out[119…

	location	extreme_poverty	human_development_index	population	total_boosters_per_hundred	total_cases	total_deaths
10	India	21.2	0.645	1417173120	36.983361	45041748.0	533623.0
11	Indonesia	5.7	0.718	275501344	36.983361	6829399.0	162059.0
12	Iran	0.2	0.783	88550568	36.983361	7627863.0	146837.0
8	China	0.7	0.761	1425887360	57.990000	99373219.0	122304.0
39	Turkey	0.2	0.820	85341248	48.540000	17004718.0	101419.0

In [137…

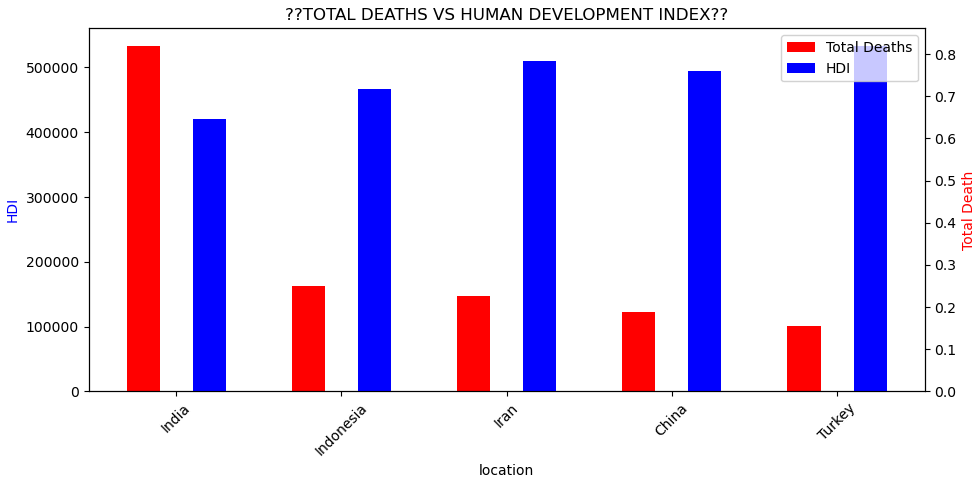
```
# Ploting Total cases and Population by continents for easy comparison

fig, ax = plt.subplots(figsize = (10, 5))
plt.title('?? TOTAL DEATHS VS EXTREME POVERTY??')

ax2 = ax.twinx()
```

```
X_axis = np.arange(len(top_5['location']))
ax.bar(X_axis - 0.2, top_5['total_deaths'], 0.2, label = 'Total Deaths', color='red')
ax2.bar(X_axis + 0.2, top_5['extreme_poverty'], 0.2, label = 'Extreme Poverty', color='blue')


ax.set_xticks(X_axis, top_5['location'], rotation=45)
ax.set_xlabel('location')
ax.set_ylabel('Extreme Poverty', color='blue')

ax2.set_ylabel('Total Death', color='red')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



In [138…

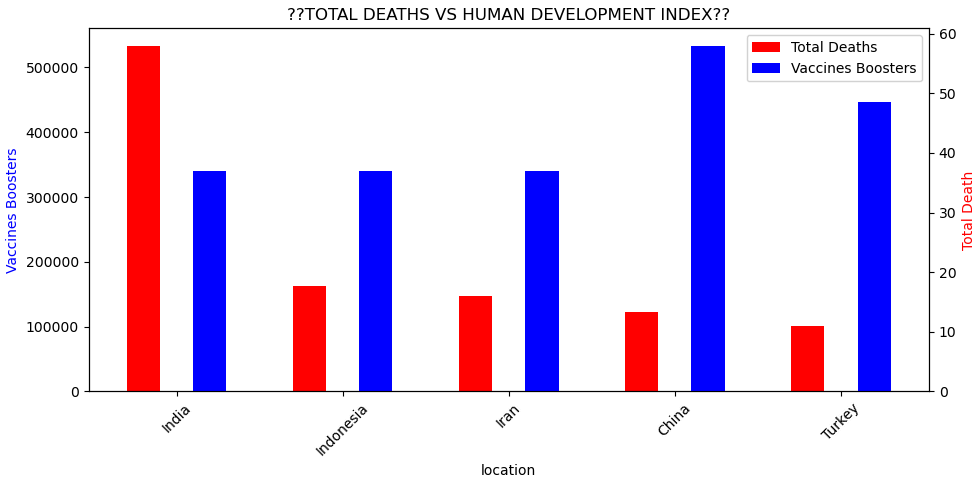
```
# Ploting Total cases and Population by continents for easy comparison

fig, ax = plt.subplots(figsize = (10, 5))
plt.title('??TOTAL DEATHS VS HUMAN DEVELOPMENT INDEX??')

ax2 = ax.twinx()


X_axis = np.arange(len(top_5['location']))
ax.bar(X_axis - 0.2, top_5['total_deaths'], 0.2, label = 'Total Deaths', color='red')
ax2.bar(X_axis + 0.2, top_5['human_development_index'], 0.2, label = 'HDI', color='blue')


ax.set_xticks(X_axis, top_5['location'], rotation=45)
ax.set_xlabel('location')
ax.set_ylabel('HDI', color='blue')

ax2.set_ylabel('Total Death', color='red')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



In [139…

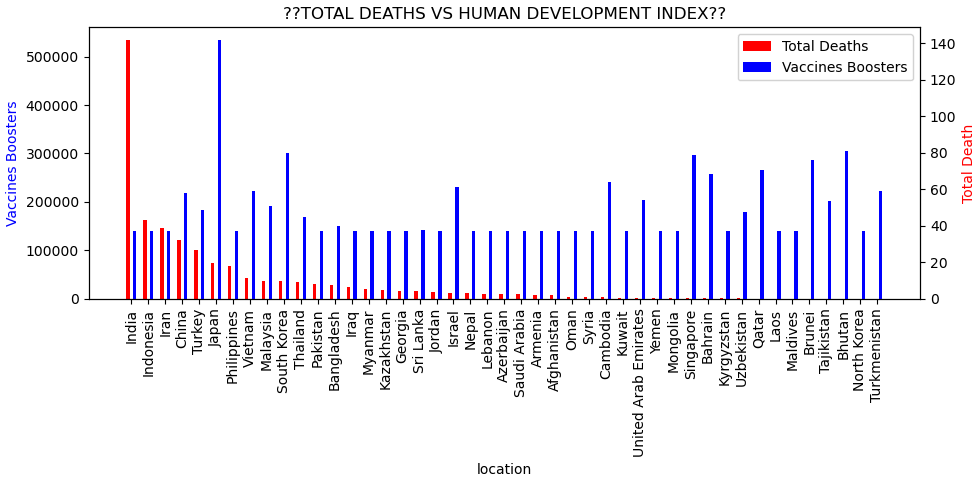
```
# Ploting Total cases and Population by continents for easy comparison
```

```
fig, ax = plt.subplots(figsize = (10, 5))
plt.title('??TOTAL DEATHS VS HUMAN DEVELOPMENT INDEX??')

ax2 = ax.twinx()


X_axis = np.arange(len(top_5['location']))
ax.bar(X_axis - 0.2, top_5['total_deaths'], 0.2, label = 'Total Deaths', color='red')
ax2.bar(X_axis + 0.2, top_5['total_boosters_per_hundred'], 0.2, label = 'Vaccines Boosters', color='blue')


ax.set_xticks(X_axis, top_5['location'], rotation=45)
ax.set_xlabel('location')
ax.set_ylabel('Vaccines Boosters', color='blue')

ax2.set_ylabel('Total Death', color='red')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



In [142…

```
# Ploting Total cases and Population by continents for easy comparison

fig, ax = plt.subplots(figsize = (10, 5))
plt.title('??TOTAL DEATHS VS HUMAN DEVELOPMENT INDEX??')

ax2 = ax.twinx()


X_axis = np.arange(len(pivot_all.sort_values(by='total_deaths', ascending=False)['location']))
ax.bar(X_axis - 0.2, pivot_all.sort_values(by='total_deaths', ascending=False)['total_deaths'], 0.2, label = 'Total Deaths', color='red')
ax2.bar(X_axis + 0.2, pivot_all.sort_values(by='total_deaths', ascending=False)['total_boosters_per_hundred'], 0.2, label = 'Vaccines Boosters', color='blue')


ax.set_xticks(X_axis, pivot_all.sort_values(by='total_deaths', ascending=False)['location'], rotation=90)
ax.set_xlabel('location')
ax.set_ylabel('Vaccines Boosters', color='blue')

ax2.set_ylabel('Total Death', color='red')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



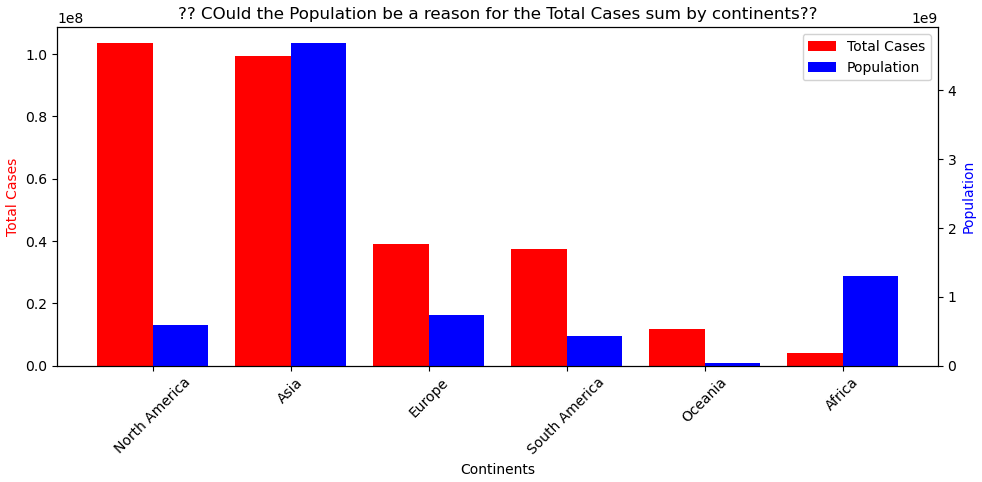
In [354…
```
"""

Though China had the higest cases,they only fall as 4th position in total death cases as India has the highest death in ASIA.

India has the second index in extreme poverty with measure of 21 and Laos with highest of 22 sits on far below in the number of death cases.
So we can say Poverty contributed to India high rate of death coupled with thier high total cases. But that seems to be contrary to their over rthe average HDI.
This shows that India didn't handle the crisis as comparedto China definitaely not because of HDI but because of poverty. China had much more cases than they do but
was able to prevent death.

India has the second population, second largest total casers, definitly the population affected how quick they were able to ocntain the outbreak. But given their poverty
rating, it seems otherwise. Wth India second ranking in poverty means that they most likely lacked proper health facilities and cares to quickly savaged the out break.

Speaking of advantages, low population, high HDI number and low extreme poverty measure, IRAN still managed to have higher death number than China,
which shows a subtle space for other factors not recorded like government policies and the masses reaction and compliances.



"""
```

Out[354… ''

In [ ]:

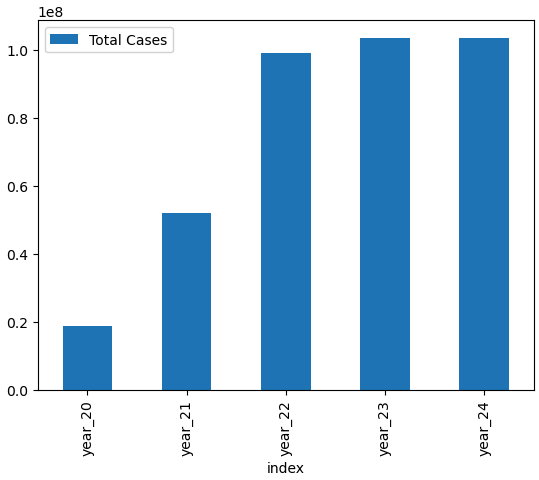
Investigating into NORTH AMAERICA

In [143…
```
# RECALL PLOT


# Ploting Total cases and Population by continents for easy comparison

fig, ax = plt.subplots(figsize = (10, 5))
plt.title('?? COuld the Population be a reason for the Total Cases sum by continents??')

ax2 = ax.twinx()

X_axis = np.arange(len(cases_by_pop['continent']))
ax.bar(X_axis - 0.2, cases_by_pop['sum_total_cases'], 0.4, label = 'Total Cases', color='red')
ax2.bar(X_axis + 0.2, cases_by_pop['population'], 0.4, label = 'Population', color='blue')

ax.set_xticks(X_axis, cases_by_pop['continent'], rotation=45)
ax.set_xlabel('Continents')
ax.set_ylabel('Total Cases', color='red')

ax2.set_ylabel('Population', color='blue')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



In [144…
```
# For NOth America alone

total_cases_by_continent[total_cases_by_continent['continent']=='North America'].transpose().reset_index().drop(0,axis=0).rename(columns={3:'Total Cases'}).plot(kind='bar',x='index',y=
```

Out[144…
```
<Axes: xlabel='index'>
```



In [ ]:
```
"""
We can see thaty NORTH AMARICA also has a spike in total cases in 2022 before being able to control the outbreak.
"""
```

In [145…
```
data_north = data[data['continent']=='North America']
data_north
```

Out[145…

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	…	male_smokers	handwashing_facilities	hospital_beds_p
13392	ATG	North America	Antigua and Barbuda	2020-01-05	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	32.655006	51.894546	
13393	ATG	North America	Antigua and Barbuda	2020-01-06	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	32.655006	51.894546	
13394	ATG	North America	Antigua and Barbuda	2020-01-07	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	32.655006	51.894546	
13395	ATG	North America	Antigua and Barbuda	2020-01-08	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	32.655006	51.894546	
13396	ATG	North America	Antigua and Barbuda	2020-01-09	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	32.655006	51.894546	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
405120	USA	North America	United States	2024-07-31	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.600000	51.894546	
405121	USA	North America	United States	2024-08-01	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.600000	51.894546	
405122	USA	North America	United States	2024-08-02	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.600000	51.894546	
405123	USA	North America	United States	2024-08-03	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.600000	51.894546	
405124	USA	North America	United States	2024-08-04	103436829.0	2502.989441	2510.397659	1193165.0	619.0	88.429000	…	24.600000	51.894546	

35158 rows × 67 columns

In [146…
```
# Pivot table to see the total cases by continents

pivot_table_north = data_north.pivot_table(index=['location'],
                                values=['total_cases','population'],
                                aggfunc='max').reset_index()
pivot_table_north.sort_values(by='total_cases', ascending=False)
```

Out[146…

	location	population	total_cases
20	United States	338289856	103436829.0
15	Mexico	127504120	7619458.0
4	Canada	38454328	4819055.0
11	Guatemala	17843914	1250371.0
5	Costa Rica	5180836	1234701.0
6	Cuba	11212198	1113662.0
17	Panama	4408582	1044821.0
8	Dominican Republic	11228821	661103.0
13	Honduras	10432858	472896.0
9	El Salvador	6336393	201920.0
19	Trinidad and Tobago	1531043	191496.0
14	Jamaica	2827382	157181.0
2	Barbados	281646	108582.0
3	Belize	405285	71414.0
1	Bahamas	409989	39127.0
12	Haiti	11585003	34456.0
18	Saint Lucia	179872	30282.0
10	Grenada	125459	19693.0
16	Nicaragua	6948395	16185.0
7	Dominica	72758	16047.0
0	Antigua and Barbuda	93772	9106.0

In [ ]:
```
"""
United states has the highest population and highest total cases.
"""
```

In [ ]:
```
"""
So what happened in USA???
"""
```

In [148…
```
data_usa = data_north[data_north['location']=='United States']
data_usa
```

Out[148…

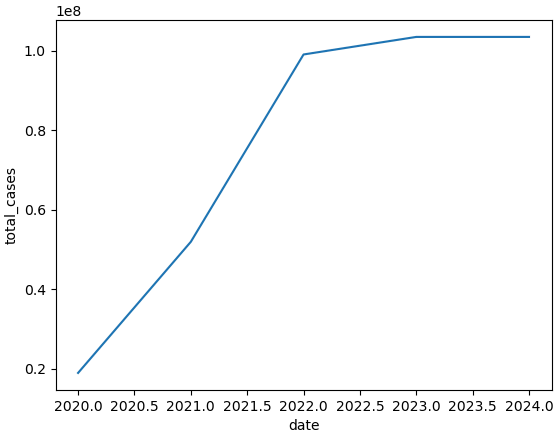
	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	…	male_smokers	handwashing_facilities	hospital_beds_p
403451	USA	North America	United States	2020-01-05	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403452	USA	North America	United States	2020-01-06	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403453	USA	North America	United States	2020-01-07	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403454	USA	North America	United States	2020-01-08	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403455	USA	North America	United States	2020-01-09	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
405120	USA	North America	United States	2024-07-31	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405121	USA	North America	United States	2024-08-01	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405122	USA	North America	United States	2024-08-02	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405123	USA	North America	United States	2024-08-03	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405124	USA	North America	United States	2024-08-04	103436829.0	2502.989441	2510.397659	1193165.0	619.0	88.429000	…	24.6	51.894546	

1674 rows × 67 columns

In [150…
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_usa[data_usa['location']=='United States'],
             x=data_usa['date'].dt.year,
             y='total_cases',
#             hue='continent',
            estimator='max',
            ci=None).set_title('?? WHat happened in United States in the these 4 years (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\3984615277.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_usa[data_usa['location']=='United States'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[150… `Text(0.5, 1.0, '?? WHat happened in United States in the these 4 years (total cases??')`

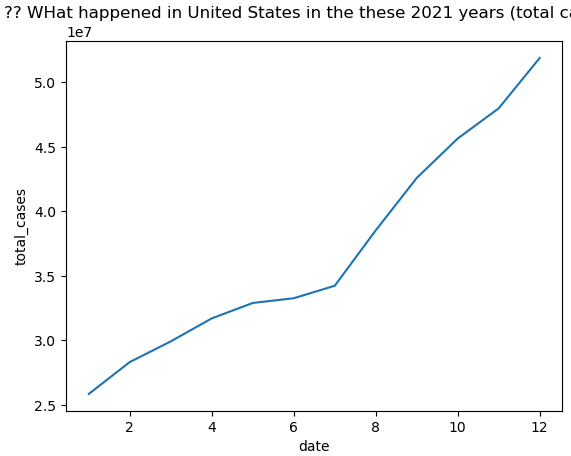


In [ ]:
```
"""
Again in USA, there was a rise in 2021 and 2022 as we can see from the plot above.
Let's see what happened in the year 2021

"""
```

In [154…
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_usa[data_usa['location']=='United States'],
                x=data_usa[data_usa['date'].dt.year==2021]['date'].dt.month,
                y='total_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in United States in the these 2021 years (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\2089035710.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_usa[data_usa['location']=='United States'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

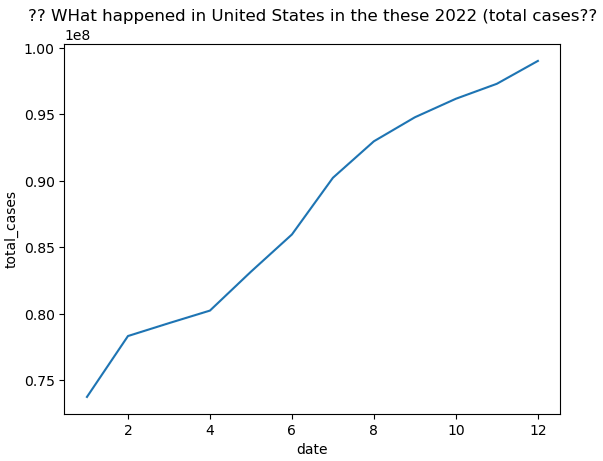
Out[154… `Text(0.5, 1.0, '?? WHat happened in United States in the these 2021 years (total cases??')`



In [155…
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_usa[data_usa['location']=='United States'],
                x=data_usa[data_usa['date'].dt.year==2022]['date'].dt.month,
                y='total_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in United States in the these 2022 (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\536285881.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_usa[data_usa['location']=='United States'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[155… Text(0.5, 1.0, '?? WHat happened in United States in the these 2022 (total cases??')

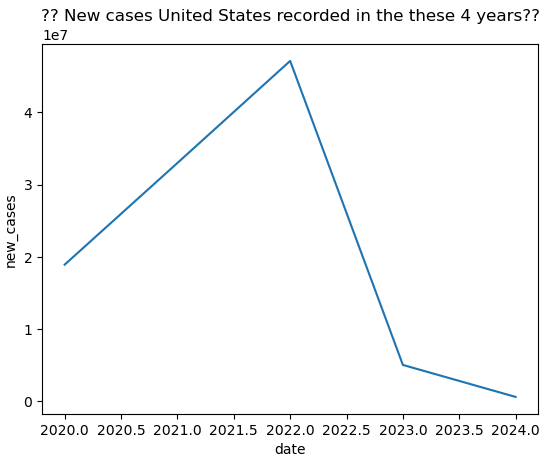


In [ ]:
```
"""
There was rather a steady rise of total cases in USA in 2022 to 2023
"""
```

In [156…
```
# New cases rate in United states

# How cases in each continents progressed over the years

sns.lineplot(data=data_usa,
                x=data_usa['date'].dt.year,
                y='new_cases',
#             hue='continent',
             estimator='sum',
             ci=None
               ).set_title('?? New cases United States recorded in the these 4 years??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\2329416879.py:5: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_usa,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[156… Text(0.5, 1.0, '?? New cases United States recorded in the these 4 years??')



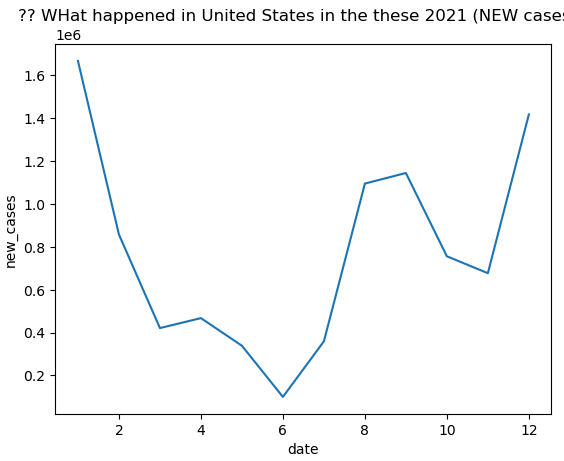
In [ ]:
```
"""
2022 seems to be the highest spike of new cases in USA.
"""
```

In [158…
```
# How cases in each continents progressed over the years

sns.lineplot(data=data_usa[data_usa['location']=='United States'],
                x=data_usa[data_usa['date'].dt.year==2021]['date'].dt.month,
                y='new_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in United States in the these 2021 (NEW cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\706277533.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_usa[data_usa['location']=='United States'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[158…

```
Text(0.5, 1.0, '?? WHat happened in United States in the these 2021 (NEW cases??')
```



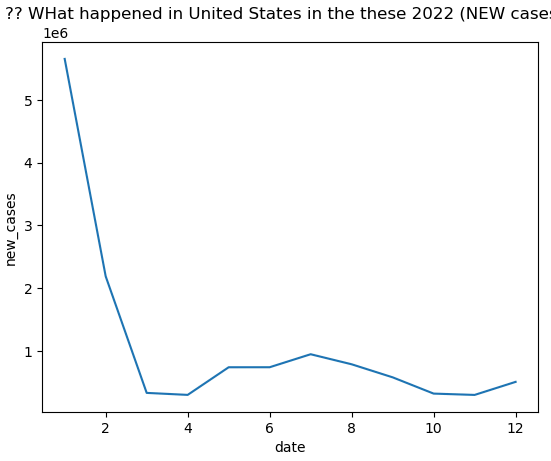
In [157…

```
# How cases in each continents progressed over the years

sns.lineplot(data=data_usa[data_usa['location']=='United States'],
                x=data_usa[data_usa['date'].dt.year==2022]['date'].dt.month,
                y='new_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in United States in the these 2022 (NEW cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\997000894.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_usa[data_usa['location']=='United States'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[157…

```
Text(0.5, 1.0, '?? WHat happened in United States in the these 2022 (NEW cases??')
```



In [ ]:

```
"""
Starting from mid year 2021 to the end of the year, new cases were recorded.
From 2022 January, the new cases record declined.

WHY??
Following the decline in new cases from begining of the year, I guess the socail distancing policy and nose mask was relaxed.
Also the new variant was hitting high too at this period.


"""
```

In [160…

```
#Pivot table to see the total deaths in NORTH AMERICA

pivot_table_north = data_north.pivot_table(index=['location'],
                            values=['total_deaths','population'],
                            aggfunc='max').reset_index()
pivot_table_north.sort_values(by='total_deaths', ascending=False)
```

Out[160…

	location	population	total_deaths
20	United States	338289856	1193165.0
15	Mexico	127504120	334551.0
4	Canada	38454328	55282.0
11	Guatemala	17843914	20203.0
13	Honduras	10432858	11114.0
5	Costa Rica	5180836	9372.0
17	Panama	4408582	8748.0
6	Cuba	11212198	8530.0
19	Trinidad and Tobago	1531043	4390.0
8	Dominican Republic	11228821	4384.0
9	El Salvador	6336393	4230.0
14	Jamaica	2827382	3611.0
12	Haiti	11585003	860.0
1	Bahamas	409989	849.0
3	Belize	405285	688.0
2	Barbados	281646	593.0
18	Saint Lucia	179872	410.0
16	Nicaragua	6948395	245.0
10	Grenada	125459	238.0
0	Antigua and Barbuda	93772	146.0
7	Dominica	72758	74.0

In [ ]:
```
"""
US loves top board...lol..
Highest total cases, highest death cases too both in North America and in the world.


So I'll be investigating their toal_cases and _total_deaths, trying to see if we can pick some pridictors.

"""
```

In [178…
```
# Starting with Correlation of TOTAL CASES

# CORRELATION FOR United States alone

corr_list = []

for col in numerical_columns:
    corr_list.append(data_usa['total_cases'].corr(data_usa[col]))

corr_df = pd.DataFrame(corr_list, numerical_columns.columns, columns=['Correlation Value']).reset_index()
```

```
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\numpy\lib\function_base.py:2897: RuntimeWarning: invalid value encountered in divide
  c /= stddev[:, None]
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\numpy\lib\function_base.py:2898: RuntimeWarning: invalid value encountered in divide
  c /= stddev[None, :]
```

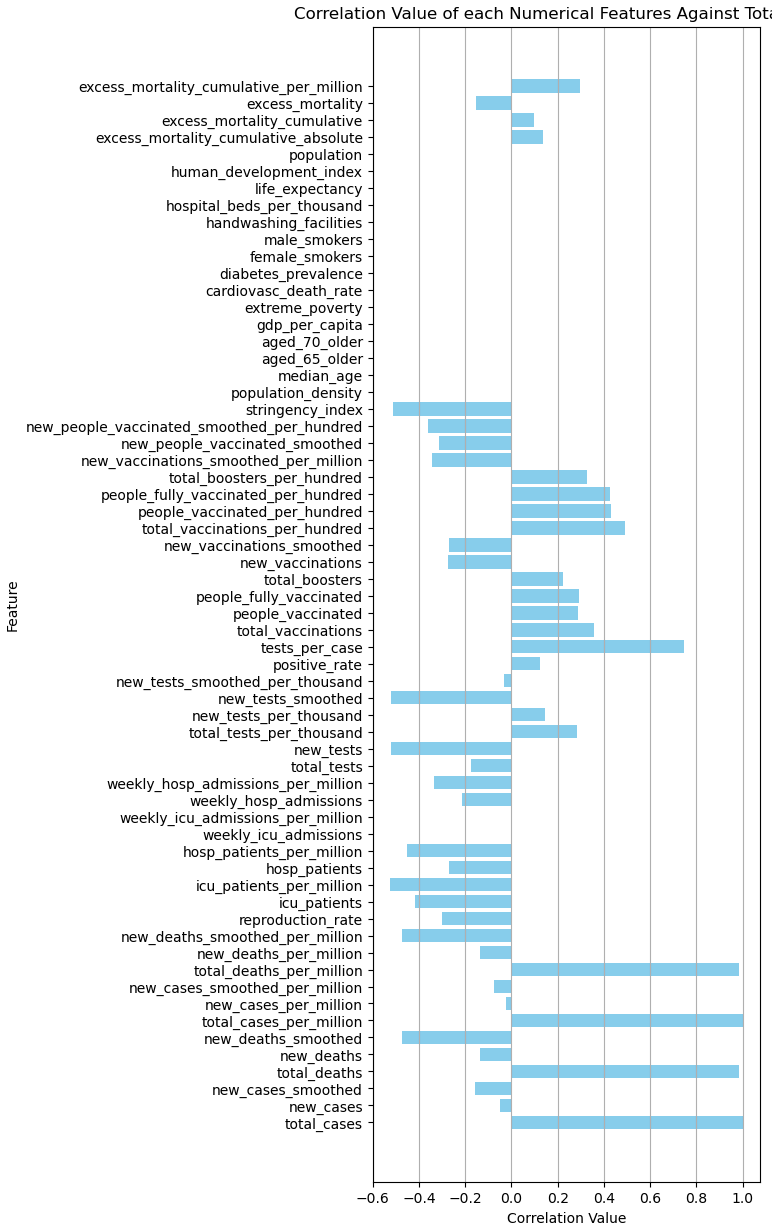
In [179…
```
corr_df
```

Out[179…

	index	Correlation Value
0	total_cases	1.000000
1	new_cases	-0.051436
2	new_cases_smoothed	-0.156821
3	total_deaths	0.983652
4	new_deaths	-0.134387
...	...	...
57	population	NaN
58	excess_mortality_cumulative_absolute	0.138194
59	excess_mortality_cumulative	0.099079
60	excess_mortality	-0.152951
61	excess_mortality_cumulative_per_million	0.296010

62 rows × 2 columns

In [180…
```
plt.figure(figsize=(5, 15))
plt.barh(numerical_columns.columns, corr_df['Correlation Value'], color='skyblue')
plt.xlabel('Correlation Value')
plt.ylabel('Feature')
plt.title('Correlation Value of each Numerical Features Against Total Death')
plt.grid(axis='x')
plt.show()
```



In [ ]:
```
"""
Some variables do not have a corre;ation value because we had RUntimeWarninbg while calculating our correlation value,
which means some calculation resulted in divisor equals 0.

So we'll progress with OLS

"""
```

In [181…
```
data_usa
```

Out[181…

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	…	male_smokers	handwashing_facilities	hospital_beds_p
403451	USA	North America	United States	2020-01-05	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403452	USA	North America	United States	2020-01-06	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403453	USA	North America	United States	2020-01-07	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403454	USA	North America	United States	2020-01-08	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
403455	USA	North America	United States	2020-01-09	0.0	0.000000	2510.397659	0.0	0.0	22.856527	…	24.6	51.894546	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
405120	USA	North America	United States	2024-07-31	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405121	USA	North America	United States	2024-08-01	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405122	USA	North America	United States	2024-08-02	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405123	USA	North America	United States	2024-08-03	103436829.0	2502.989441	2510.397659	1192546.0	0.0	88.429000	…	24.6	51.894546	
405124	USA	North America	United States	2024-08-04	103436829.0	2502.989441	2510.397659	1193165.0	619.0	88.429000	…	24.6	51.894546	

1674 rows × 67 columns

In [183…

```
# OLS
import statsmodels.api as sm

X = data_usa.drop(['iso_code','continent','location','date','tests_units','total_cases'], axis=1)
X = sm.add_constant(X)
y = data_usa['total_cases']


# Split your data set into 80/20 for train/test datasets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)


# create a fitted model & print the summary
y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()


print(lm.summary(xname=list(X.columns)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                1.551e+16
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        15:08:40   Log-Likelihood:                -2647.7
No. Observations:                1339   AIC:                             5385.
Df Residuals:                    1294   BIC:                             5619.
Df Model:                          44
Covariance Type:            nonrobust
================================================================================================
                                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------------
new_cases                         0.0022      0.004      0.566      0.572      -0.005       0.010
new_cases_smoothed               -0.0022      0.004     -0.570      0.569      -0.010       0.005
total_deaths                      0.2245      0.478      0.469      0.639      -0.714       1.163
new_deaths                        0.6800      1.350      0.504      0.615      -1.968       3.328
new_deaths_smoothed               0.3015      0.528      0.571      0.568      -0.734       1.337
total_cases_per_million         341.5340   1.09e-05   3.13e+07      0.000     341.534     341.534
new_cases_per_million            -0.7373      1.303     -0.566      0.572      -3.293       1.819
new_cases_smoothed_per_million    0.7414      1.299      0.571      0.568      -1.807       3.290
total_deaths_per_million        -76.6872    163.392     -0.469      0.639    -397.230     243.856
new_deaths_per_million         -232.2361    461.070     -0.504      0.615   -1136.763     672.291
new_deaths_smoothed_per_million -102.9554    180.306     -0.571      0.568    -456.680     250.770
reproduction_rate                 0.1377      0.255      0.541      0.589      -0.362       0.638
icu_patients                     -0.2176      0.545     -0.399      0.690      -1.287       0.852
icu_patients_per_million         73.5932    184.463      0.399      0.690    -288.286     435.472
hosp_patients                     0.0270      0.068      0.399      0.690      -0.106       0.160
hosp_patients_per_million        -9.1389     22.912     -0.399      0.690     -54.088      35.810
weekly_icu_admissions            -0.0258      0.056     -0.462      0.644      -0.135       0.084
weekly_icu_admissions_per_million 5.196e-07   2.6e-06      0.200      0.842   -4.58e-06    5.62e-06
weekly_hosp_admissions           1.828e-05   4.12e-05      0.444      0.657   -6.25e-05    9.91e-05
weekly_hosp_admissions_per_million  -0.0116      0.021     -0.541      0.588      -0.054       0.030
total_tests                      5.549e-07   2.63e-06      0.211      0.833   -4.6e-06    5.71e-06
new_tests                         -0.0002      0.001     -0.211      0.833      -0.002       0.001
total_tests_per_thousand          -0.1874      0.886     -0.211      0.833      -1.926       1.551
new_tests_per_thousand            51.3348    242.780      0.211      0.833    -424.950     527.620
new_tests_smoothed               2.115e-06   2.49e-06      0.848      0.396   -2.78e-06    7.01e-06
new_tests_smoothed_per_thousand   -0.3348      0.794     -0.422      0.673      -1.893       1.223
positive_rate                      4.5605      2.623      1.739      0.082      -0.585       9.706
tests_per_case                     0.0002      0.001      0.289      0.773      -0.001       0.002
total_vaccinations              -3.464e-06   3.7e-06     -0.936      0.350   -1.07e-05    3.8e-06
people_vaccinated                6.975e-06   6.82e-06      1.023      0.306    -6.4e-06    2.04e-05
people_fully_vaccinated          6.559e-07   6.53e-06      0.100      0.920   -1.22e-05    1.35e-05
total_boosters                  -2.596e-09    3.4e-09     -0.764      0.445   -9.27e-09    4.07e-09
new_vaccinations                 7.994e-08   2.68e-07      0.298      0.766   -4.47e-07    6.07e-07
new_vaccinations_smoothed           0.0001   3.07e-05      4.541      0.000    7.92e-05       0.000
total_vaccinations_per_hundred      11.4943     12.293      0.935      0.350     -12.622      35.611
people_vaccinated_per_hundred      -23.0942     22.631     -1.020      0.308     -67.491      21.303
people_fully_vaccinated_per_hundred  -2.2274     21.680     -0.103      0.918     -44.759      40.304
total_boosters_per_hundred        -0.0033      0.018     -0.189      0.850      -0.038       0.031
new_vaccinations_smoothed_per_million  -0.0465      0.010     -4.563      0.000      -0.066      -0.026
new_people_vaccinated_smoothed    -0.0003   7.22e-05     -4.486      0.000      -0.000      -0.000
new_people_vaccinated_smoothed_per_hundred  1078.2119    239.914      4.494      0.000     607.550    1548.874
stringency_index                  -0.0129      0.012     -1.045      0.296      -0.037       0.011
population_density             -2.012e-09   3.5e-09     -0.575      0.565   -8.87e-09    4.85e-09
median_age                     -4.116e-09   6.72e-09     -0.613      0.540   -1.73e-08    9.06e-09
aged_65_older                  -1.273e-09   2.34e-09     -0.545      0.586   -5.86e-09    3.31e-09
aged_70_older                   1.991e-11   2.26e-10      0.088      0.930   -4.22e-10    4.62e-10
gdp_per_capita                 -1.385e-06    5.1e-06     -0.271      0.786   -1.14e-05    8.63e-06
extreme_poverty                -8.555e-11   1.35e-10     -0.636      0.525    -3.5e-10    1.79e-10
cardiovasc_death_rate          -5.758e-09    1.5e-08     -0.385      0.700   -3.51e-08    2.36e-08
diabetes_prevalence             2.966e-10   6.53e-10      0.455      0.649   -9.83e-10    1.58e-09
female_smokers                  1.944e-09   1.71e-09      1.139      0.255   -1.41e-09    5.29e-09
male_smokers                    3.684e-10   4.28e-10      0.862      0.389    -4.7e-10    1.21e-09
handwashing_facilities          2.857e-10   1.31e-09      0.218      0.827   -2.28e-09    2.85e-09
hospital_beds_per_thousand     -3.107e-10    4.1e-10     -0.758      0.449   -1.12e-09    4.94e-10
life_expectancy                -5.238e-09   1.03e-08     -0.510      0.610   -2.54e-08    1.49e-08
human_development_index        -5.953e-11   1.77e-10     -0.337      0.736   -4.06e-10    2.87e-10
population                      2.841e-08   5.93e-08      0.479      0.632   -8.79e-08    1.45e-07
excess_mortality_cumulative_absolute  2.034e-07   5.87e-07      0.346      0.729   -9.49e-07    1.36e-06
excess_mortality_cumulative       -0.0047      0.052     -0.091      0.928      -0.107       0.097
excess_mortality                   0.0168      0.026      0.655      0.513      -0.034       0.067
excess_mortality_cumulative_per_million  -0.0001      0.000     -0.389      0.698      -0.001       0.000
==============================================================================
Omnibus:                      461.192   Durbin-Watson:                   1.928
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7262.222
Skew:                           1.163   Prob(JB):                         0.00
Kurtosis:                      14.169   Cond. No.                     1.04e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 4.36e-12. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

In [ ]:
```
"""
Droping variables with p-value greater than .05


Dropping all features except::::

total_cases_per_million
new_vaccinations_smoothed
new_vaccinations_smoothed_per_million
new_people_vaccinated_smoothed
new_people_vaccinated_smoothed_per_hundred

"""
```

In [188…
```
# OLS
import statsmodels.api as sm

X = data_usa.drop(['iso_code','continent','location','date','tests_units','total_cases'], axis=1)[[
                    'total_cases_per_million','new_vaccinations_smoothed','new_vaccinations_smoothed_per_million',
                    'new_people_vaccinated_smoothed','new_people_vaccinated_smoothed_per_hundred'
]]

X = sm.add_constant(X)
y = data_usa['total_cases']
```

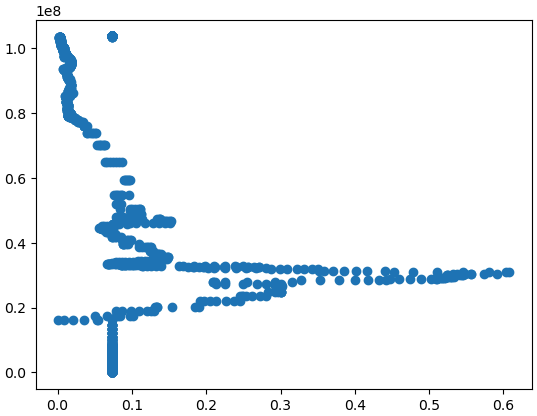
```
# Split your data set into 80/20 for train/test datasets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)


# create a fitted model & print the summary
y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()


print(lm.summary(xname=list(X.columns)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       1.000
Model:                            OLS   Adj. R-squared:                  1.000
Method:                 Least Squares   F-statistic:                 1.352e+17
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        15:26:21   Log-Likelihood:                -2673.9
No. Observations:                1339   AIC:                             5360.
Df Residuals:                    1333   BIC:                             5391.
Df Model:                           5
Covariance Type:            nonrobust
=============================================================================================================
                                                coef    std err          t      P>|t|      [0.025      0.975]
-------------------------------------------------------------------------------------------------------------
const                                       -0.6521      0.150     -4.348      0.000      -0.946      -0.358
total_cases_per_million                    341.5340    4.5e-07   7.59e+08      0.000     341.534     341.534
new_vaccinations_smoothed                    0.0001   3.02e-05      4.604      0.000    7.98e-05       0.000
new_vaccinations_smoothed_per_million       -0.0459      0.010     -4.585      0.000      -0.066      -0.026
new_people_vaccinated_smoothed              -0.0003   7.16e-05     -4.631      0.000      -0.000      -0.000
new_people_vaccinated_smoothed_per_hundred  1099.4922    237.589      4.628      0.000     633.404    1565.581
==============================================================================
Omnibus:                      470.223   Durbin-Watson:                   1.904
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             7690.334
Skew:                           1.183   Prob(JB):                         0.00
Kurtosis:                      14.500   Cond. No.                     4.29e+09
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.29e+09. This might indicate that there are
strong multicollinearity or other numerical problems.
```

In [ ]:
```
"""
Of all the features left, we have the positive features:::

total_cases_per_million                     341.5340. This feature is just a multiple of our total_cases variable, so it is not mean much.

new_people_vaccinated_smoothed_per_hundred  1099.4922


So we'll plot new_people_vaccinated_smoothed_per_hundred  and total_cases side by side
"""
```

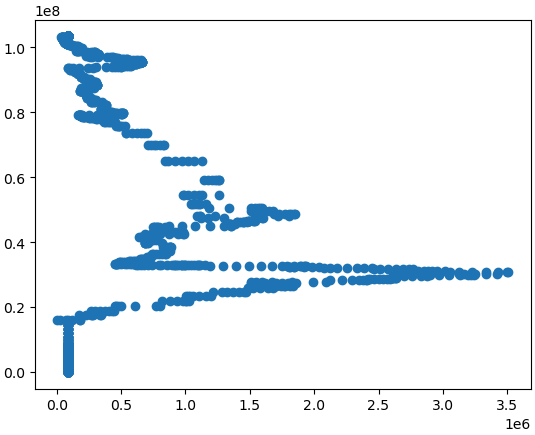
In [211…
```
plt.scatter(x = data_usa['new_people_vaccinated_smoothed_per_hundred'],
            y = data_usa['total_cases'])
```

Out[211…
```
<matplotlib.collections.PathCollection at 0x170b7722a10>
```



In [213…
```
plt.scatter(x = data_usa['new_vaccinations_smoothed'],
            y = data_usa['total_cases'])
```

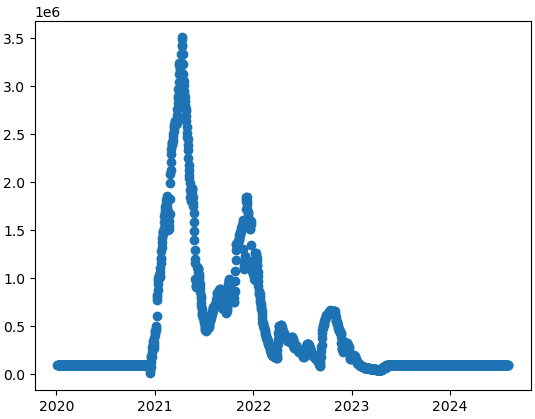
Out[213…
```
<matplotlib.collections.PathCollection at 0x170b7aa0490>
```



In [ ]:
```
"""
Somthing about the pattern of these plots indicates that at some period the vaccine had a positive slope with the total cases and at another time
it had a negative slope.



"""
```

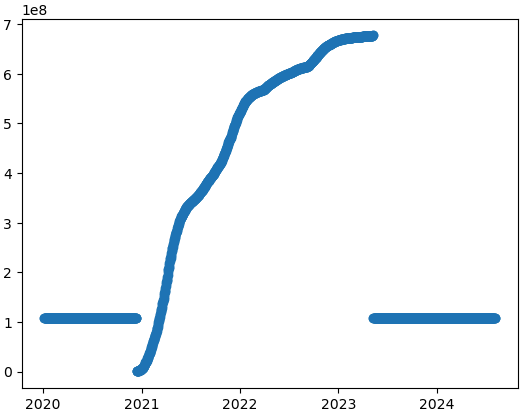
In [214…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['new_vaccinations_smoothed'])
```

Out[214… <matplotlib.collections.PathCollection at 0x170b7aaf990>



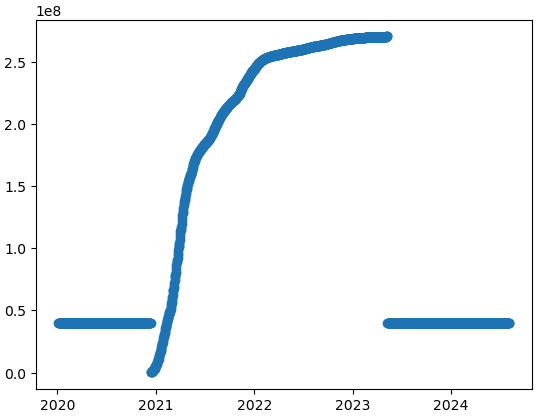
In [215…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['total_vaccinations'])
```

Out[215… <matplotlib.collections.PathCollection at 0x170b7b16d90>



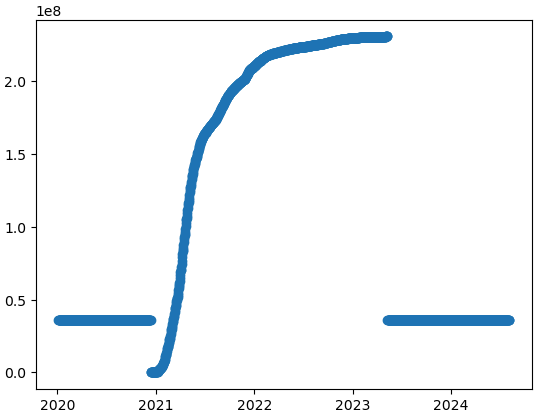
In [216…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['people_vaccinated'])
```

Out[216… <matplotlib.collections.PathCollection at 0x170b7b97e10>



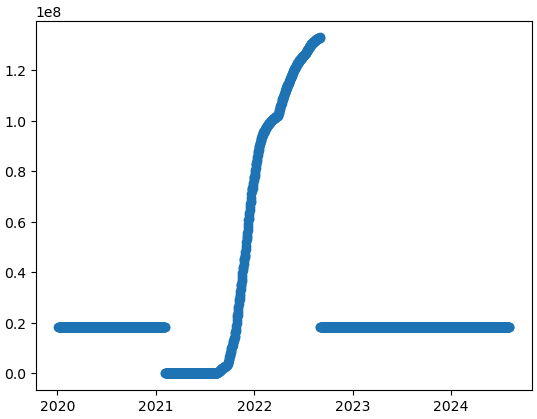
In [217…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['people_fully_vaccinated'])
```

Out[217…
```
<matplotlib.collections.PathCollection at 0x170b7b920d0>
```



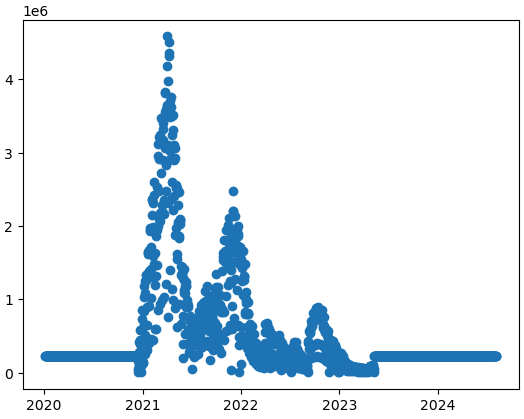
In [218…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['total_boosters'])
```

Out[218…
```
<matplotlib.collections.PathCollection at 0x170b7b93450>
```



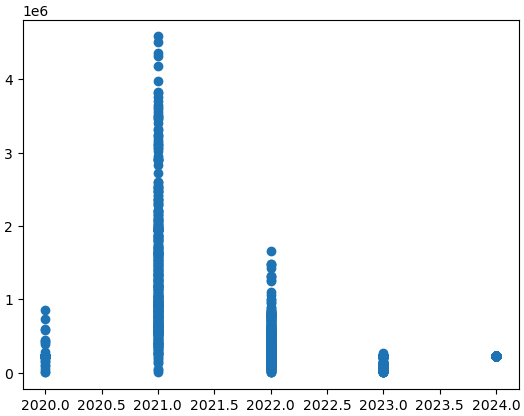
In [219…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['new_vaccinations'])
```

Out[219…
```
<matplotlib.collections.PathCollection at 0x170b7b909d0>
```



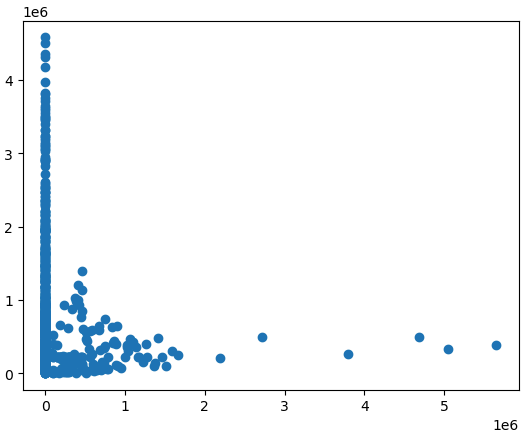
In [225…
```
plt.scatter(x = data_usa['date'].dt.year,
            y = data_usa['new_vaccinations'])
```

Out[225…
```
<matplotlib.collections.PathCollection at 0x170c56310d0>
```



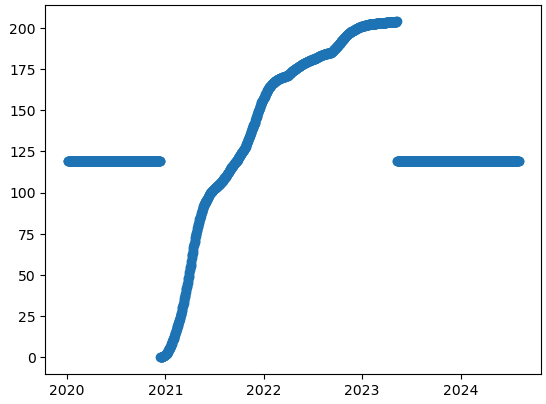
In [226…
```
plt.scatter(x = data_usa['new_cases'],
            y = data_usa['new_vaccinations'])
```

Out[226…
```
<matplotlib.collections.PathCollection at 0x170c569fa10>
```



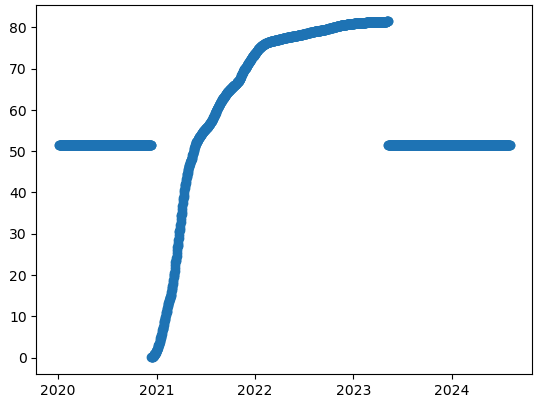
In [220…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['total_vaccinations_per_hundred'])
```

Out[220…
```
<matplotlib.collections.PathCollection at 0x170b8390310>
```



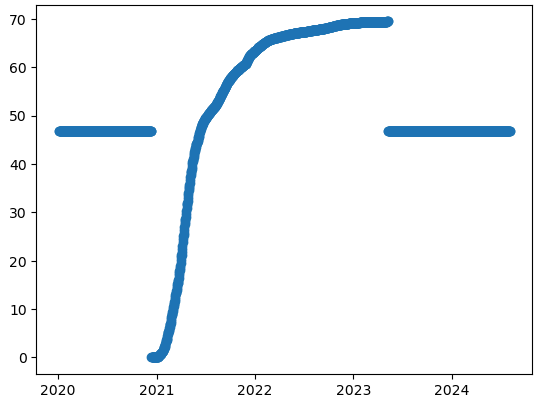
In [221…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['people_vaccinated_per_hundred'])
```

Out[221…
```
<matplotlib.collections.PathCollection at 0x170b83bf9d0>
```



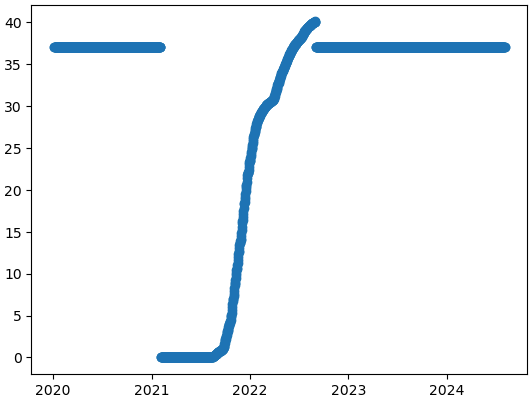
In [222…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['people_fully_vaccinated_per_hundred'])
```

Out[222…
```
<matplotlib.collections.PathCollection at 0x170b84652d0>
```



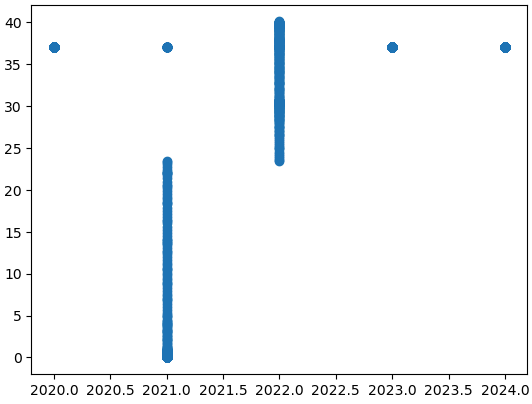
In [223…
```
plt.scatter(x = data_usa['date'],
            y = data_usa['total_boosters_per_hundred'])
```

Out[223…
```
<matplotlib.collections.PathCollection at 0x170c53f9c10>
```



In [224…
```
plt.scatter(x = data_usa['date'].dt.year,
            y = data_usa['total_boosters_per_hundred'])
```

Out[224…
```
<matplotlib.collections.PathCollection at 0x170c5467850>
```

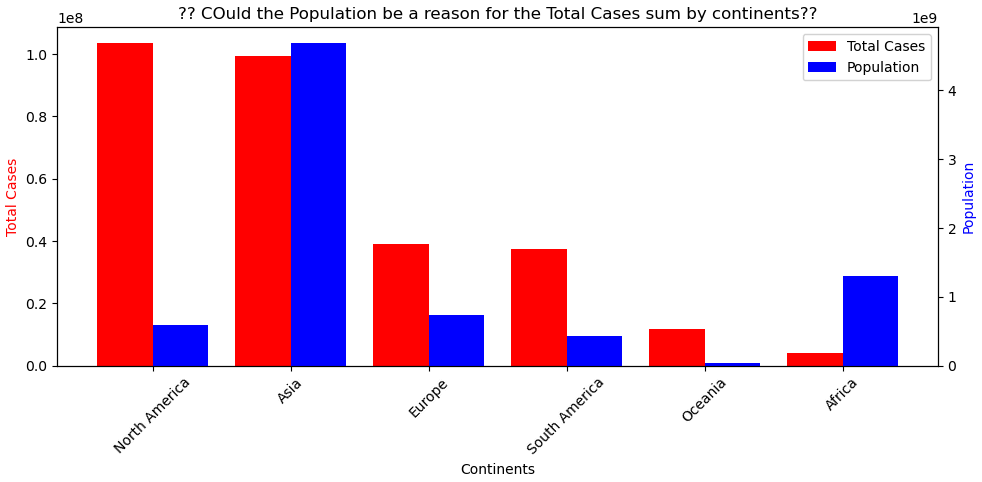


In [ ]:
```
"""
1. At the end of 2020, all charts shows there was a discontinuity in the use of vaccines.
2. There was high increase of use of vaccine towards the end of 2021, which was around when high record of new cases was recorded.
   lol...you can do CAUSALITY test with the date column to check if there is a cause and effect there.
   But if I am going to judge by the time I have, there was surge in US total cases in the mid 2021 because as at 2021 January, they
   had stopped administering the vaccine.
   Good luck to them.
3. Which makes sense because we can't say US lacks the facilities to contain the virus nor they have extreme poverty level like the India.

"""
```

In [ ]:
```
"""
With this, I think I am going to stop on North America now and move to Africa.
"""
```

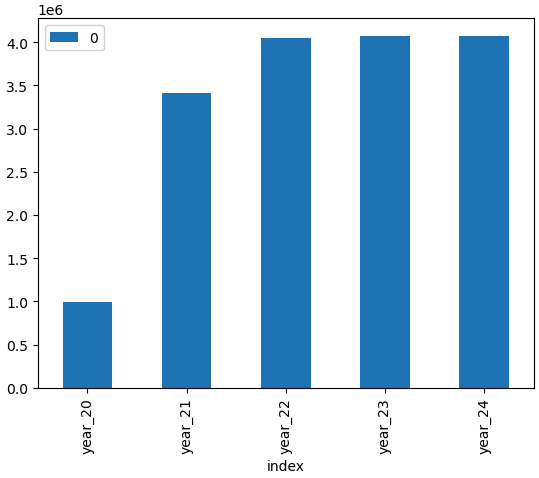
Investigating AFRICA

In [227…
```
# RECALL PLOT


# Ploting Total cases and Population by continents for easy comparison

fig, ax = plt.subplots(figsize = (10, 5))
plt.title('?? COuld the Population be a reason for the Total Cases sum by continents??')

ax2 = ax.twinx()

X_axis = np.arange(len(cases_by_pop['continent']))
ax.bar(X_axis - 0.2, cases_by_pop['sum_total_cases'], 0.4, label = 'Total Cases', color='red')
ax2.bar(X_axis + 0.2, cases_by_pop['population'], 0.4, label = 'Population', color='blue')

ax.set_xticks(X_axis, cases_by_pop['continent'], rotation=45)
ax.set_xlabel('Continents')
ax.set_ylabel('Total Cases', color='red')

ax2.set_ylabel('Population', color='blue')

tick1, label1 = ax.get_legend_handles_labels()
tick2, label2 = ax2.get_legend_handles_labels()
tick = tick1 + tick2
label = label1 + label2
plt.legend(tick, label, loc='upper right')

plt.tight_layout()

plt.show()
```



In [230…
```
# For Africa alone

total_cases_by_continent[total_cases_by_continent['continent']=='Africa'].transpose().reset_index().drop(0,axis=0).rename(columns={3:'Total Cases'}).plot(kind='bar',x='index',y=0)
```

Out[230…
```
<Axes: xlabel='index'>
```



In [ ]:
```
"""
Seems Africa was able to quikly contain the outbreak. Froom 2022 till 2024, they had a steady approximate numbers.

"""
```

In [231…
```
data_africa = data[data['continent']=='Africa']
data_africa
```

Out[231…

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	…	male_smokers	handwashing_facilities	hospital_beds_per
5022	DZA	Africa	Algeria	2020-01-05	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	30.4	83.741	
5023	DZA	Africa	Algeria	2020-01-06	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	30.4	83.741	
5024	DZA	Africa	Algeria	2020-01-07	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	30.4	83.741	
5025	DZA	Africa	Algeria	2020-01-08	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	30.4	83.741	
5026	DZA	Africa	Algeria	2020-01-09	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	30.4	83.741	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
429430	ZWE	Africa	Zimbabwe	2024-07-31	266386.0	0.0	0.000000	5740.0	0.0	0.000000	…	30.7	36.791	
429431	ZWE	Africa	Zimbabwe	2024-08-01	266386.0	0.0	0.000000	5740.0	0.0	0.000000	…	30.7	36.791	
429432	ZWE	Africa	Zimbabwe	2024-08-02	266386.0	0.0	0.000000	5740.0	0.0	0.000000	…	30.7	36.791	
429433	ZWE	Africa	Zimbabwe	2024-08-03	266386.0	0.0	0.000000	5740.0	0.0	0.000000	…	30.7	36.791	
429434	ZWE	Africa	Zimbabwe	2024-08-04	266386.0	0.0	0.000000	5740.0	0.0	0.000000	…	30.7	36.791	

83700 rows × 67 columns

In [232…
```
# Pivot table to see the total cases by continents

pivot_table_africa = data_africa.pivot_table(index=['location'],
```

```
                            values=['total_cases','population'],
                            aggfunc='max').reset_index()
pivot_table_africa.sort_values(by='total_cases', ascending=False)
```

Out[232…

	location	population	total_cases
41	South Africa	59893884	4072765.0
31	Morocco	37457976	1279115.0
46	Tunisia	12356116	1153361.0
12	Egypt	110990096	516023.0
25	Libya	6812344	507269.0
16	Ethiopia	123379928	501193.0
48	Zambia	20017670	349842.0
22	Kenya	54027484	344106.0
3	Botswana	2630300	330696.0
30	Mauritius	1299478	328167.0
0	Algeria	44903228	272139.0
35	Nigeria	218541216	267188.0
49	Zimbabwe	16320539	266386.0
32	Mozambique	32969520	233843.0
33	Namibia	2567024	172533.0
47	Uganda	47249588	172154.0
19	Ghana	33475870	172062.0
36	Rwanda	13776702	133264.0
6	Cameroon	27914542	125246.0
1	Angola	35588996	107481.0
37	Senegal	17316452	89485.0
27	Malawi	20405318	89168.0
15	Eswatini	1201680	75356.0
26	Madagascar	29611718	68567.0
43	Sudan	46874200	63993.0
29	Mauritania	4736146	63872.0
5	Burundi	12889583	54569.0
38	Seychelles	107135	51886.0
17	Gabon	2388997	49051.0
44	Tanzania	65497752	43230.0
45	Togo	8848700	39530.0
20	Guinea	13859349	38572.0
23	Lesotho	2305826	36138.0
28	Mali	22593598	33166.0
2	Benin	13352864	28036.0
40	Somalia	17597508	27334.0
10	Congo	5970430	25227.0
4	Burkina Faso	22673764	22139.0
42	South Sudan	10913172	18823.0
13	Equatorial Guinea	1674916	17130.0
11	Djibouti	1120851	15690.0
7	Central African Republic	5579148	15441.0
18	Gambia	2705995	12627.0
14	Eritrea	3684041	10189.0
21	Guinea-Bissau	2105580	9614.0
34	Niger	26207982	9518.0
9	Comoros	836783	9109.0
24	Liberia	5302690	8090.0
39	Sierra Leone	8605723	7979.0
8	Chad	17723312	7702.0

In [237…

```
4072765.0-1279115.0
```

Out[237…

```
2793650.0
```

In [ ]:

```
"""
Though overall Africa had the the relative low number of 1 out of 370+ persons. But South Africa still had high nunbers that ia over 200% of the immediate country.
So let's check into South Africa to see if we can learn anything from there.

"""
```

In [239…

```
data_south_africa = data_africa[data_africa['location']=='South Africa']
data_south_africa
```

Out[239…

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	total_deaths	new_deaths	new_deaths_smoothed	…	male_smokers	handwashing_facilities	hospital_beds_per_
355890	ZAF	Africa	South Africa	2020-01-05	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	33.2	43.993	
355891	ZAF	Africa	South Africa	2020-01-06	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	33.2	43.993	
355892	ZAF	Africa	South Africa	2020-01-07	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	33.2	43.993	
355893	ZAF	Africa	South Africa	2020-01-08	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	33.2	43.993	
355894	ZAF	Africa	South Africa	2020-01-09	0.0	0.0	2510.397659	0.0	0.0	22.856527	…	33.2	43.993	
…	…	…	…	…	…	…	…	…	…	…	…	…	…	
357559	ZAF	Africa	South Africa	2024-07-31	4072765.0	0.0	0.286000	102595.0	0.0	0.000000	…	33.2	43.993	
357560	ZAF	Africa	South Africa	2024-08-01	4072765.0	0.0	0.286000	102595.0	0.0	0.000000	…	33.2	43.993	
357561	ZAF	Africa	South Africa	2024-08-02	4072765.0	0.0	0.286000	102595.0	0.0	0.000000	…	33.2	43.993	
357562	ZAF	Africa	South Africa	2024-08-03	4072765.0	0.0	0.286000	102595.0	0.0	0.000000	…	33.2	43.993	
357563	ZAF	Africa	South Africa	2024-08-04	4072765.0	0.0	0.000000	102595.0	0.0	0.000000	…	33.2	43.993	

1674 rows × 67 columns

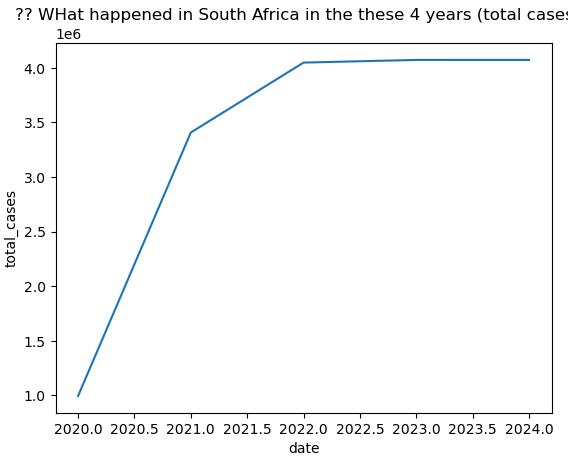
In [241…

```
# How cases in each continents progressed over the years

sns.lineplot(data=data_africa[data_africa['location']=='South Africa'],
             x=data_africa['date'].dt.year,
             y='total_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in South Africa in the these 4 years (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\4097159013.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_africa[data_africa['location']=='South Africa'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[241…

```
Text(0.5, 1.0, '?? WHat happened in South Africa in the these 4 years (total cases??')
```



In [ ]:

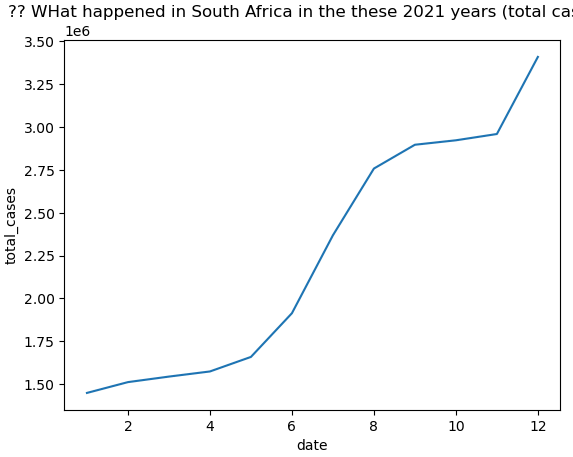
```
"""
Lets see their new cases in 2021

"""
```

In [242…

```
# How cases in each continents progressed over the years

sns.lineplot(data=data_africa[data_africa['location']=='South Africa'],
             x=data_africa[data_africa['date'].dt.year==2021]['date'].dt.month,
             y='total_cases',
#             hue='continent',
             estimator='max',
             ci=None).set_title('?? WHat happened in South Africa in the these 2021 years (total cases??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\420807065.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_africa[data_africa['location']=='South Africa'],
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[242… Text(0.5, 1.0, '?? WHat happened in South Africa in the these 2021 years (total cases??')

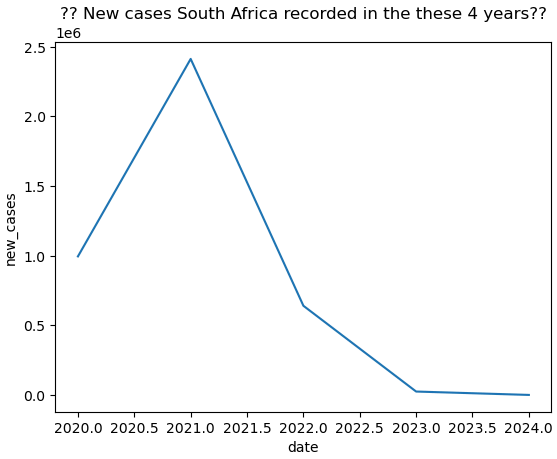


In [ ]:
```
"""
Generally, 2021 was a rise for most countries in the world.
SOuth Africa had increase starting from around April, 2021 till the end of the year.

Let's see the new_cases record

"""
```

In [244…
```
# New cases rate in United statesSouth Africa
# How cases in each continents progressed over the years

sns.lineplot(data=data_south_africa,
                x=data_south_africa['date'].dt.year,
                y='new_cases',
#             hue='continent',
             estimator='sum',
             ci=None
               ).set_title('?? New cases South Africa recorded in the these 4 years??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\1594702977.py:4: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data_south_africa,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[244… Text(0.5, 1.0, '?? New cases South Africa recorded in the these 4 years??')



In [ ]:
```
"""
Whatsoever they did to have low record of new cases, they were serious with it because right from 2021 up to date. their new cases record had declined.

Maybe the same thing the whole Africa continents were doing...lol...let's try to find out.
We'll plunge directly into the Whole Africa dataset to sniff for their recipe..lol..


"""
```

In [245…
```
# Starting with Correlation of TOTAL CASES

# CORRELATION FOR United States alone

corr_list = []

for col in numerical_columns:
    corr_list.append(data_africa['total_cases'].corr(data_africa[col]))

corr_df = pd.DataFrame(corr_list, numerical_columns.columns, columns=['Correlation Value']).reset_index()
```

```
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\numpy\lib\function_base.py:2897: RuntimeWarning: invalid value encountered in divide
  c /= stddev[:, None]
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\numpy\lib\function_base.py:2898: RuntimeWarning: invalid value encountered in divide
  c /= stddev[None, :]
```

In [246…

```
corr_df
```

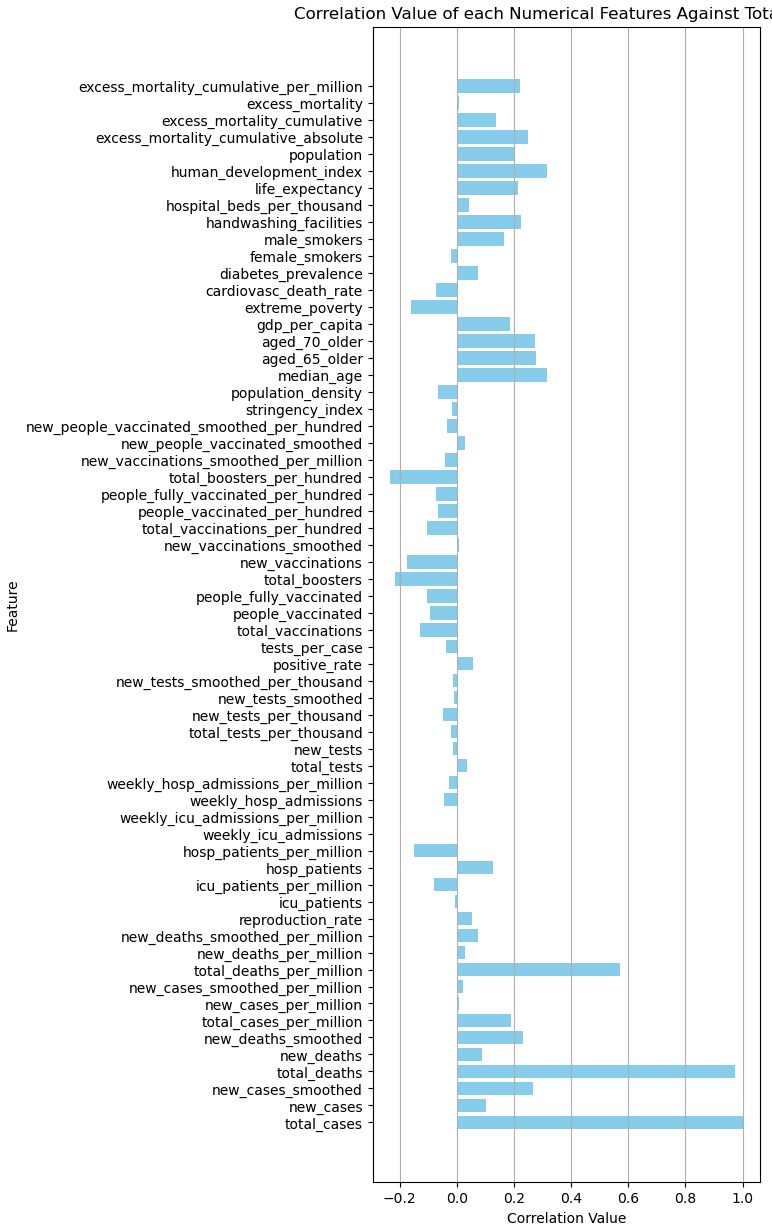
Out[246…

	index	Correlation Value
0	total_cases	1.000000
1	new_cases	0.101434
2	new_cases_smoothed	0.265549
3	total_deaths	0.973711
4	new_deaths	0.087073
...	...	...
57	population	0.202501
58	excess_mortality_cumulative_absolute	0.250515
59	excess_mortality_cumulative	0.138216
60	excess_mortality	0.007942
61	excess_mortality_cumulative_per_million	0.220581

62 rows × 2 columns

In [247…

```
plt.figure(figsize=(5, 15))
plt.barh(numerical_columns.columns, corr_df['Correlation Value'], color='skyblue')
plt.xlabel('Correlation Value')
plt.ylabel('Feature')
plt.title('Correlation Value of each Numerical Features Against Total Death')
plt.grid(axis='x')
plt.show()
```



In [ ]:
```
"""
Nothing spectacular. We'll proceedto OLS

"""
```

In [249…
```
# OLS
import statsmodels.api as sm

X = data_africa.drop(['iso_code','continent','location','date','tests_units','total_cases'], axis=1)
X = sm.add_constant(X)
y = data_africa['total_cases']


# Split your data set into 80/20 for train/test datasets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)


# create a fitted model & print the summary
y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()


print(lm.summary(xname=list(X.columns)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.964
Model:                            OLS   Adj. R-squared:                  0.964
Method:                 Least Squares   F-statistic:                 3.007e+04
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        22:59:58   Log-Likelihood:            -8.6010e+05
No. Observations:               66960   AIC:                         1.720e+06
Df Residuals:                   66900   BIC:                         1.721e+06
Df Model:                          59
Covariance Type:            nonrobust
================================================================================================================
                                                   coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------------------------
new_cases                                       0.6543      0.305      2.147      0.032       0.057       1.251
new_cases_smoothed                             17.4518      0.895     19.510      0.000      15.699      19.205
total_deaths                                   38.0370      0.047    817.329      0.000      37.946      38.128
new_deaths                                     -5.0521     14.033     -0.360      0.719     -32.558      22.453
new_deaths_smoothed                          -406.2354     47.469     -8.558      0.000    -499.275    -313.196
total_cases_per_million                         0.2725      0.012     22.034      0.000       0.248       0.297
new_cases_per_million                          -1.3164      1.686     -0.781      0.435      -4.622       1.989
new_cases_smoothed_per_million                 -57.3666      4.511    -12.718      0.000     -66.207     -48.526
total_deaths_per_million                       -13.2467      1.677     -7.898      0.000     -16.534      -9.959
new_deaths_per_million                        -153.8222    213.166     -0.722      0.471    -571.626     263.982
new_deaths_smoothed_per_million             -6983.1077    626.353    -11.149      0.000   -8210.758   -5755.457
reproduction_rate                            6789.8080   1187.272      5.719      0.000    4462.755    9116.861
icu_patients                                 -414.3211     65.770     -6.300      0.000    -543.231    -285.411
icu_patients_per_million                    2.457e+04   3077.112      7.985      0.000    1.85e+04    3.06e+04
hosp_patients                                 -54.3221      5.718     -9.500      0.000     -65.530     -43.114
hosp_patients_per_million                    2731.4587    211.425     12.919      0.000    2317.065    3145.852
weekly_icu_admissions                       -1679.2740     49.398    -33.995      0.000   -1776.094   -1582.454
weekly_icu_admissions_per_million             -45.2082      1.330    -33.995      0.000     -47.815     -42.602
weekly_hosp_admissions                        -22.8237     21.402     -1.066      0.286     -64.772      19.125
weekly_hosp_admissions_per_million           1590.4919   1296.172      1.227      0.220    -950.005    4130.989
total_tests                                     0.0015      0.000      3.520      0.000       0.001       0.002
new_tests                                      -1.3859      0.163     -8.498      0.000      -1.705      -1.066
total_tests_per_thousand                      -16.8806     10.869     -1.553      0.120     -38.183       4.422
new_tests_per_thousand                      2.773e+04   3621.380      7.656      0.000    2.06e+04    3.48e+04
new_tests_smoothed                             -0.0716      0.024     -3.042      0.002      -0.118      -0.025
new_tests_smoothed_per_thousand              6473.0332   1158.719      5.586      0.000    4201.945    8744.121
positive_rate                               -1.781e+04   9742.345     -1.828      0.068   -3.69e+04    1283.735
tests_per_case                                  0.6020      1.058      0.569      0.569      -1.472       2.676
total_vaccinations                             -0.0012      0.000     -9.516      0.000      -0.001      -0.001
people_vaccinated                               0.0018      0.000      5.683      0.000       0.001       0.002
people_fully_vaccinated                         0.0008      0.000      2.250      0.024       0.000       0.002
total_boosters                                 -0.0005      0.001     -0.735      0.462      -0.002       0.001
new_vaccinations                                0.1608      0.013     12.637      0.000       0.136       0.186
new_vaccinations_smoothed                      -0.0904      0.014     -6.482      0.000      -0.118      -0.063
total_vaccinations_per_hundred                788.1136    122.063      6.457      0.000     548.871    1027.356
people_vaccinated_per_hundred                -527.1442    279.974     -1.883      0.060   -1075.893      21.605
people_fully_vaccinated_per_hundred          -494.7322    292.326     -1.692      0.091   -1067.692      78.227
total_boosters_per_hundred                    -63.8376    346.877     -0.184      0.854    -743.717     616.042
new_vaccinations_smoothed_per_million          -2.1623      0.533     -4.057      0.000      -3.207      -1.118
new_people_vaccinated_smoothed                  0.0679      0.025      2.735      0.006       0.019       0.117
new_people_vaccinated_smoothed_per_hundred  -3.54e+04   8208.523     -4.313      0.000   -5.15e+04   -1.93e+04
stringency_index                             -141.7633     22.269     -6.366      0.000    -185.411     -98.115
population_density                             -1.0654      3.925     -0.271      0.786      -8.757       6.627
median_age                                  1.092e+04    296.134     36.879      0.000    1.03e+04    1.15e+04
aged_65_older                              -4400.4477   1793.086     -2.454      0.014   -7914.895    -886.001
aged_70_older                               1.929e+04   2802.640      6.881      0.000    1.38e+04    2.48e+04
gdp_per_capita                                 -4.6106      0.149    -30.927      0.000      -4.903      -4.318
extreme_poverty                              -475.3384     24.334    -19.534      0.000    -523.034    -427.643
cardiovasc_death_rate                        -161.7300      7.351    -22.001      0.000    -176.138    -147.322
diabetes_prevalence                         -9330.7460    165.096    -56.517      0.000   -9654.333   -9007.159
female_smokers                               4192.1680    113.868     36.816      0.000    3968.986    4415.350
male_smokers                                 -781.7847     51.576    -15.158      0.000    -882.874    -680.696
handwashing_facilities                      -1358.6993     26.869    -50.567      0.000   -1411.363   -1306.035
hospital_beds_per_thousand                  -4534.7005    381.545    -11.885      0.000   -5282.528   -3786.873
life_expectancy                              2244.8326    129.045     17.396      0.000    1991.905    2497.760
human_development_index                      1.64e+05   9196.288     17.836      0.000    1.46e+05    1.82e+05
population                                     0.0004   1.26e-05     32.495      0.000       0.000       0.000
excess_mortality_cumulative_absolute          -1.0236      0.068    -14.972      0.000      -1.158      -0.890
excess_mortality_cumulative                -8034.7866    787.193    -10.207      0.000   -9577.685   -6491.889
excess_mortality                            1415.3921    237.159      5.968      0.000     950.560    1880.224
excess_mortality_cumulative_per_million      101.0221      5.902     17.116      0.000      89.454     112.590
==============================================================================
Omnibus:                    25482.965   Durbin-Watson:                   2.005
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           345578.761
Skew:                           1.454   Prob(JB):                         0.00
Kurtosis:                      13.743   Cond. No.                     1.04e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 9.17e-12. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

In [ ]:
```
"""
Again we remove every feature with p-value over .05

new_deaths
new_cases_per_million
new_deaths_per_million
weekly_hosp_admissions
weekly_hosp_admissions_per_million
total_tests_per_thousand
positive_rate
tests_per_case
people_fully_vaccinated
total_boosters
people_vaccinated_per_hundred
people_fully_vaccinated_per_hundred
total_boosters_per_hundred
population_density
aged_65_older


"""
```

In [250…
```
# OLS
import statsmodels.api as sm
```

```
X = data_africa.drop(['iso_code','continent','location','date','tests_units','total_cases','new_deaths','new_cases_per_million','new_deaths_per_million',
                      'weekly_hosp_admissions','weekly_hosp_admissions_per_million','total_tests_per_thousand','positive_rate','tests_per_case',
                      'people_fully_vaccinated','total_boosters','people_vaccinated_per_hundred','people_fully_vaccinated_per_hundred',
                      'total_boosters_per_hundred','population_density','aged_65_older'], axis=1)
X = sm.add_constant(X)
y = data_africa['total_cases']


# Split your data set into 80/20 for train/test datasets
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=.80, random_state=1)


# create a fitted model & print the summary
y_train = np.array(y_train, dtype=np.float64)
X_train = np.array(X_train, dtype=np.float64)

lm = sm.OLS(y_train, X_train).fit()


print(lm.summary(xname=list(X.columns)))
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.964
Model:                            OLS   Adj. R-squared:                  0.964
Method:                 Least Squares   F-statistic:                 4.029e+04
Date:                Sun, 09 Feb 2025   Prob (F-statistic):               0.00
Time:                        23:10:13   Log-Likelihood:            -8.6013e+05
No. Observations:               66960   AIC:                          1.720e+06
Df Residuals:                   66915   BIC:                          1.721e+06
Df Model:                          44
Covariance Type:            nonrobust
================================================================================================================
                                                   coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------------------------------------
new_cases                                       0.4015      0.197      2.036      0.042       0.015       0.788
new_cases_smoothed                             17.3026      0.817     21.169      0.000      15.701      18.905
total_deaths                                   38.0626      0.046    833.902      0.000      37.973      38.152
new_deaths_smoothed                          -414.6784     45.906     -9.033      0.000    -504.654    -324.703
total_cases_per_million                         0.2743      0.012     22.417      0.000       0.250       0.298
new_cases_smoothed_per_million                 -58.7431      4.245    -13.838      0.000     -67.063     -50.423
total_deaths_per_million                       -13.4458      1.648     -8.159      0.000     -16.676     -10.216
new_deaths_smoothed_per_million              -7373.5713    581.235    -12.686      0.000   -8512.792   -6234.350
reproduction_rate                             6363.3198   1172.784      5.426      0.000    4064.665    8661.975
icu_patients                                  -419.9232     65.025     -6.458      0.000    -547.373    -292.474
icu_patients_per_million                       2.48e+04   3043.798      8.147      0.000    1.88e+04    3.08e+04
hosp_patients                                  -52.1091      5.571     -9.353      0.000     -63.029     -41.189
hosp_patients_per_million                     2639.7187    202.300     13.049      0.000    2243.211    3036.227
weekly_icu_admissions                        -1691.7606     31.677    -53.407      0.000   -1753.847   -1629.674
weekly_icu_admissions_per_million              -45.5465      0.853    -53.409      0.000     -47.218     -43.875
total_tests                                     0.0009      0.000      8.092      0.000       0.001       0.001
new_tests                                      -1.2930      0.155     -8.318      0.000      -1.598      -0.988
new_tests_per_thousand                       2.537e+04   3384.827      7.496      0.000    1.87e+04    3.2e+04
new_tests_smoothed                             -0.0623      0.020     -3.104      0.002      -0.102      -0.023
new_tests_smoothed_per_thousand              6094.0516   1115.043      5.465      0.000    3908.568    8279.535
total_vaccinations                             -0.0008   5.74e-05    -13.524      0.000      -0.001      -0.001
people_vaccinated                               0.0014      0.000     11.852      0.000       0.001       0.002
new_vaccinations                                0.1518      0.013     12.093      0.000       0.127       0.176
new_vaccinations_smoothed                      -0.0883      0.014     -6.345      0.000      -0.116      -0.061
total_vaccinations_per_hundred                383.3869     41.504      9.237      0.000     302.039     464.734
new_vaccinations_smoothed_per_million          -2.2713      0.529     -4.296      0.000      -3.308      -1.235
new_people_vaccinated_smoothed                  0.0739      0.025      2.982      0.003       0.025       0.122
new_people_vaccinated_smoothed_per_hundred -3.809e+04   8154.026     -4.671      0.000   -5.41e+04   -2.21e+04
stringency_index                             -136.3909     22.221     -6.138      0.000    -179.945     -92.837
median_age                                   1.069e+04    274.606     38.920      0.000    1.01e+04    1.12e+04
aged_70_older                                1.285e+04   1155.448     11.120      0.000    1.06e+04    1.51e+04
gdp_per_capita                                 -4.5205      0.139    -32.558      0.000      -4.793      -4.248
extreme_poverty                              -490.6995     23.622    -20.773      0.000    -536.999    -444.400
cardiovasc_death_rate                        -157.9494      6.837    -23.101      0.000    -171.351    -144.548
diabetes_prevalence                          -9514.5869    130.754    -72.767      0.000   -9770.865   -9258.309
female_smokers                                4218.1485    112.782     37.401      0.000    3997.095    4439.202
male_smokers                                  -787.9595     51.504    -15.299      0.000    -888.907    -687.012
handwashing_facilities                       -1359.1758     26.373    -51.537      0.000   -1410.867   -1307.485
hospital_beds_per_thousand                   -4349.3319    374.603    -11.611      0.000   -5083.553   -3615.110
life_expectancy                               2362.1590    122.435     19.293      0.000    2122.187    2602.131
human_development_index                       1.596e+05   8985.922     17.758      0.000    1.42e+05    1.77e+05
population                                      0.0004   1.25e-05     32.627      0.000       0.000       0.000
excess_mortality_cumulative_absolute           -1.0223      0.068    -15.084      0.000      -1.155      -0.889
excess_mortality_cumulative                 -8063.9413    786.865    -10.248      0.000   -9606.197   -6521.686
excess_mortality                             1402.1995    228.020      6.149      0.000     955.281    1849.118
excess_mortality_cumulative_per_million       101.3858      5.871     17.270      0.000      89.879     112.892
==============================================================================
Omnibus:                    25500.727   Durbin-Watson:                   2.005
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           345855.966
Skew:                           1.455   Prob(JB):                         0.00
Kurtosis:                      13.747   Cond. No.                     1.04e+16
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The smallest eigenvalue is 8.19e-12. This might indicate that there are
strong multicollinearity problems or that the design matrix is singular.
```

In [ ]:

```
"""
We have all our features less than .05 now.

Before looking at thefeatures that are obviously Causatives of COVID, some features caught my attention like
gdp_per_capita, extreme_poverty, these coefficent shows that due to the not-enough-financial status of some Africa
countries, it reduced their exposure to the disease...lol..After all if a man does not have money, he should guard his health.
Hence, the strigency_index coeeficient which shoes that an avaerage African was well complaint to the lock down policies. See the
 handwashing_facilities, hospital_beds_per_thousand, coeefficient which are negative as well.
Consequently, weekly_icu_admissions are high showing strict complaince to sending victims to hospitals straight up.
It is worth noting that in spite of getting low vaccination as we can see the coeeficients of various vaccination measures are low, Africa still managed to contain the outbreak as earl
Apart from other possible measures has weather condition that could have alter downward the rate of spread of the virus as  well as less immigration from other countries outside the co


Another odd number I thought I could wrap my head around is the coefficient of  diabetes_prevalence....lol..especially
diabetes_prevalence...I think some times there is blessing is some bad things....lol..my point is diabetes could have kept some people close to the hospital which in turn had kept them
```

```
"""
```

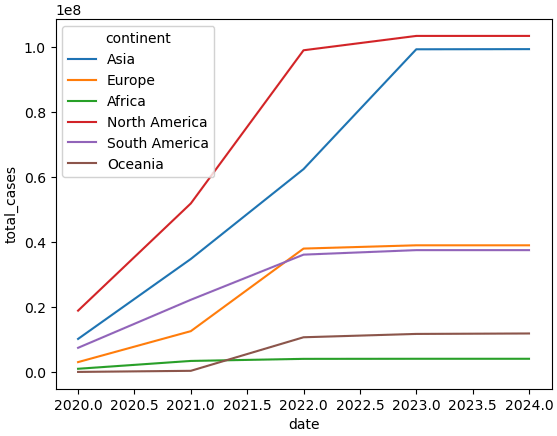
Conclusion

In [251…
```
# How cases in each continents progressed over the years

sns.lineplot(data=data,
                x=data['date'].dt.year,
                y='total_cases',
            hue='continent',
            estimator='max',
            ci=None
              ).set_title('?? How did the total cases progress over the years in each continent??')
```

```
C:\Users\OLADOYINBO BABATUNDE\AppData\Local\Temp\ipykernel_11964\1022548101.py:3: FutureWarning:

The `ci` parameter is deprecated. Use `errorbar=None` for the same effect.

  sns.lineplot(data=data,
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\OLADOYINBO BABATUNDE\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert in
f values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[251…
```
Text(0.5, 1.0, '?? How did the total cases progress over the years in each continent??')
```

?? How did the total cases progress over the years in each continent??



In [ ]:
```
"""
I think whart happend in Africa speaks averagely for the other continents that were able to quickly contain the virus as early as 2021, 2022
like Oceania.

So I'll be ending the analysis here.
It;s been a great journey with you.

Feel free to continue and share insights.

THANK YOU.


"""
```

Analysis moltivation : Elma Fortunate.....fortunatecreations.com

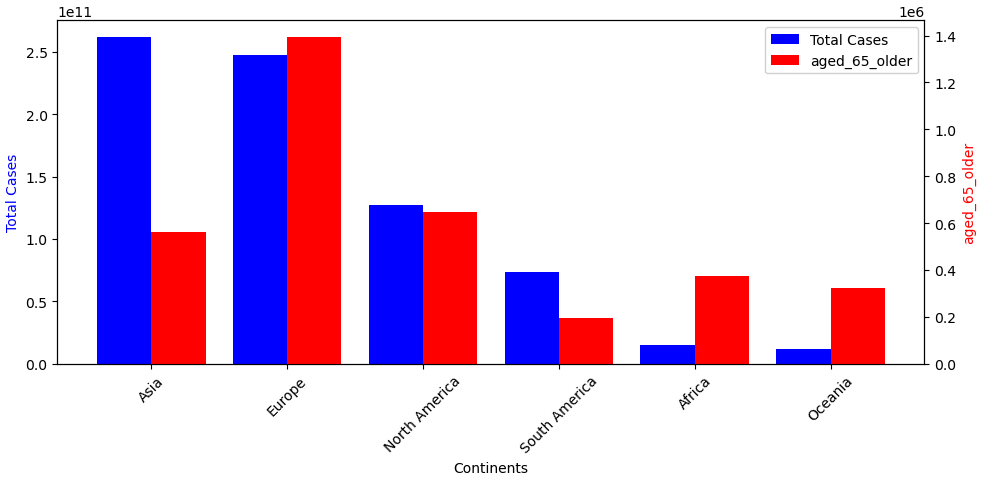
Data credit : ourworldindata

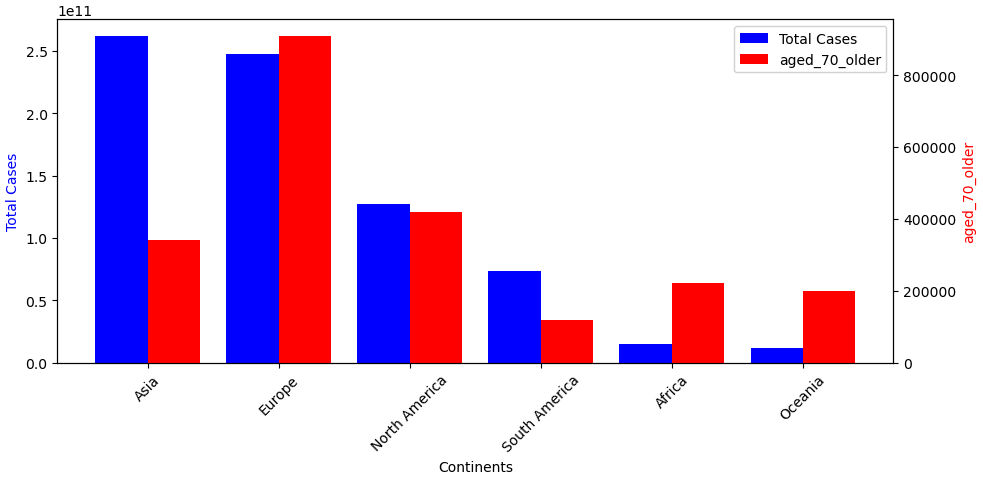
PLEASE NOTE ALL ANALYSIS, INSIGHTS AND RECOMMENDATIONS ARE BASED ON THE ANALYSIS AND DATA HERE.

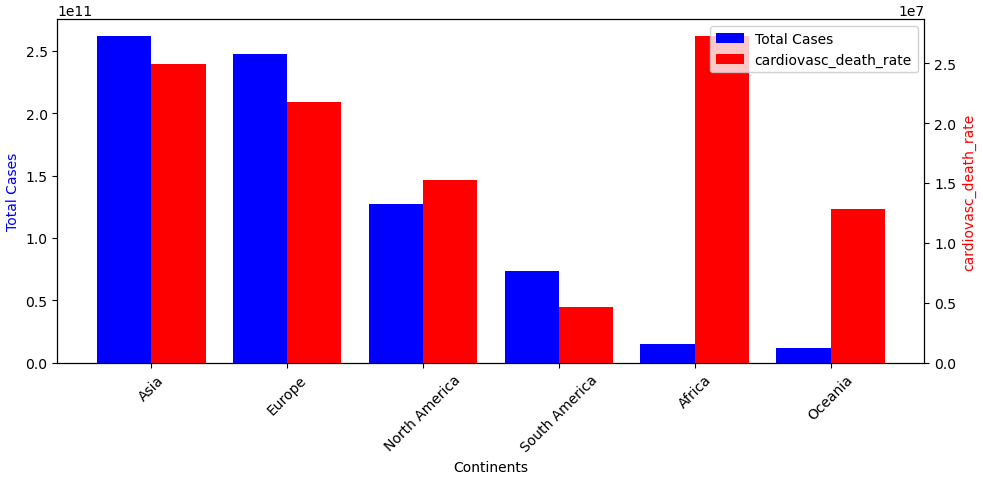
Extras

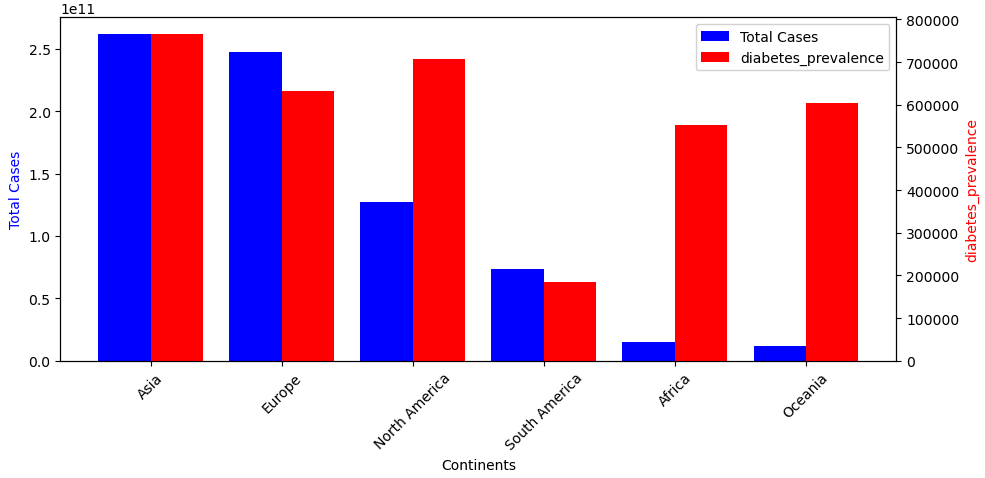
In [382…
```
for feature in pivot_table3.columns.drop('continent'):
    fig, ax = plt.subplots(figsize = (10, 5))
#     plt.title(feature, 'and Population by Continents')

    ax2 = ax.twinx()

    X_axis = np.arange(len(pivot_table3['continent']))
    ax.bar(X_axis - 0.2, pivot_table3['total_cases'], 0.4, label = 'Total Cases', color='blue')
    ax2.bar(X_axis + 0.2, pivot_table3[feature], 0.4, label = feature, color='red')

    ax.set_xticks(X_axis, pivot_table3['continent'], rotation=45)
    ax.set_xlabel('Continents')
    ax.set_ylabel('Total Cases', color='blue')

    ax2.set_ylabel(feature, color='red')

    tick1, label1 = ax.get_legend_handles_labels()
    tick2, label2 = ax2.get_legend_handles_labels()
    tick = tick1 + tick2
    label = label1 + label2
    plt.legend(tick, label, loc='upper right')

    plt.tight_layout()
```

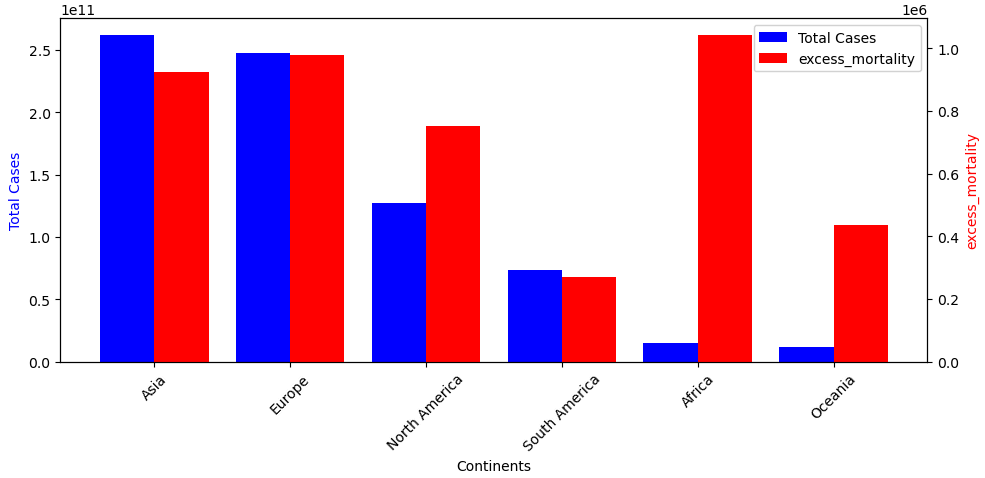
```
plt.show()
```

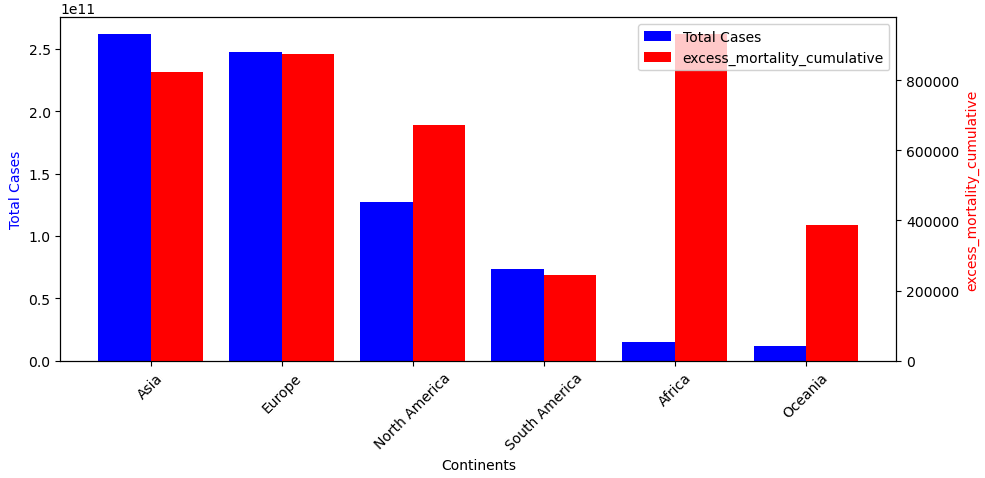

1e11
1e6
Total Cases
aged_65_older
Total Cases
aged_65_older
Asia
Europe
North America
South America
Africa
Oceania
Continents

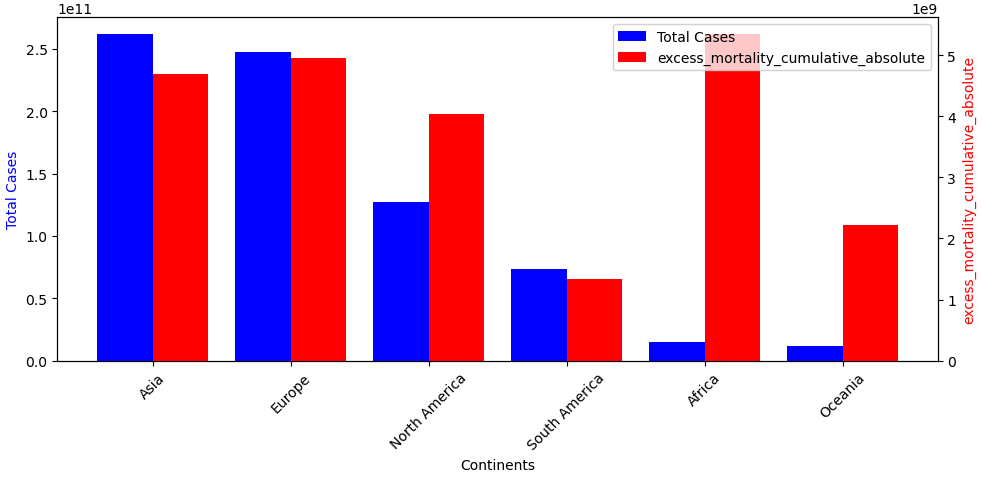

1e11
Total Cases
aged_70_older
Total Cases
aged_70_older
Asia
Europe
North America
South America
Africa
Oceania
Continents

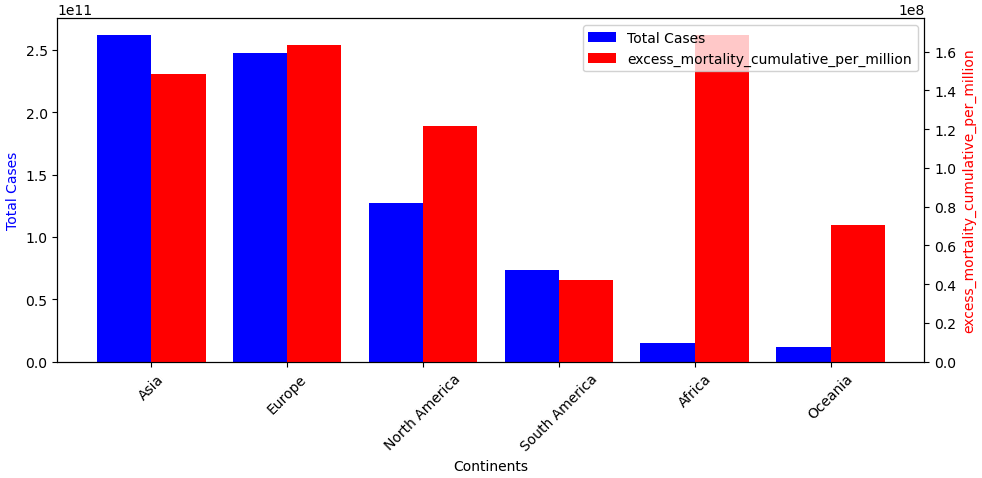

1e11
1e7
Total Cases
cardiovasc_death_rate
Total Cases
cardiovasc_death_rate
Asia
Europe
North America
South America
Africa
Oceania
Continents

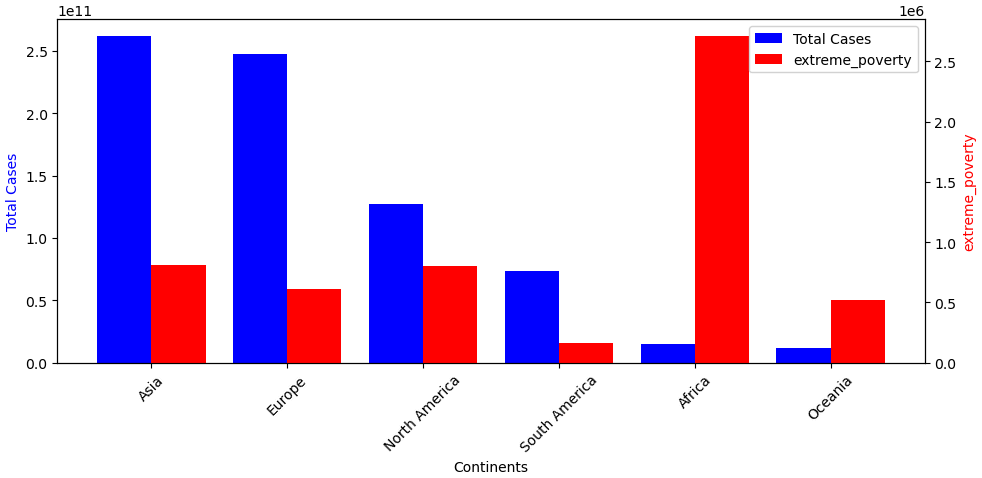

1e11
Total Cases
diabetes_prevalence
Total Cases
diabetes_prevalence
Asia
Europe
North America
South America
Africa
Oceania
Continents

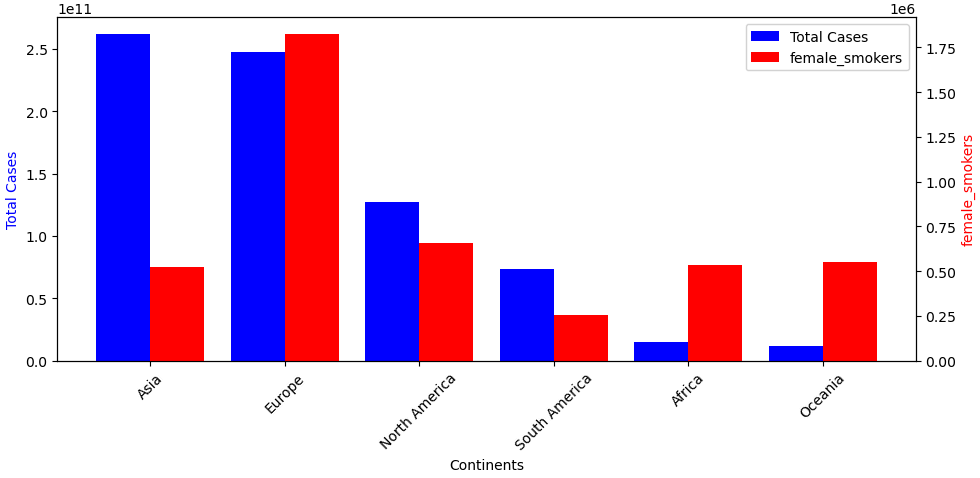

1e11
1e6
Total Cases
excess_mortality
Total Cases
excess_mortality
Asia
Europe
North America
South America
Africa
Oceania
Continents

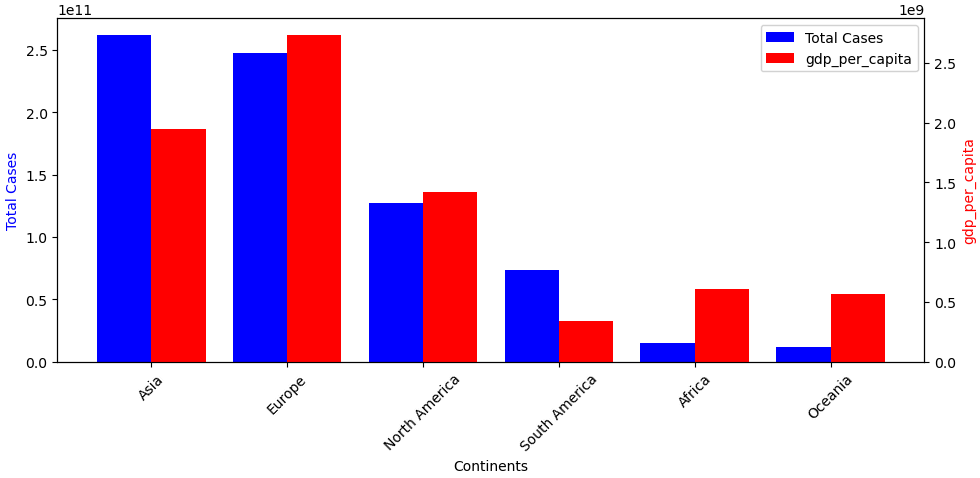

1e11
Total Cases
excess_mortality_cumulative
Total Cases
excess_mortality_cumulative
Asia
Europe
North America
South America
Africa
Oceania
Continents

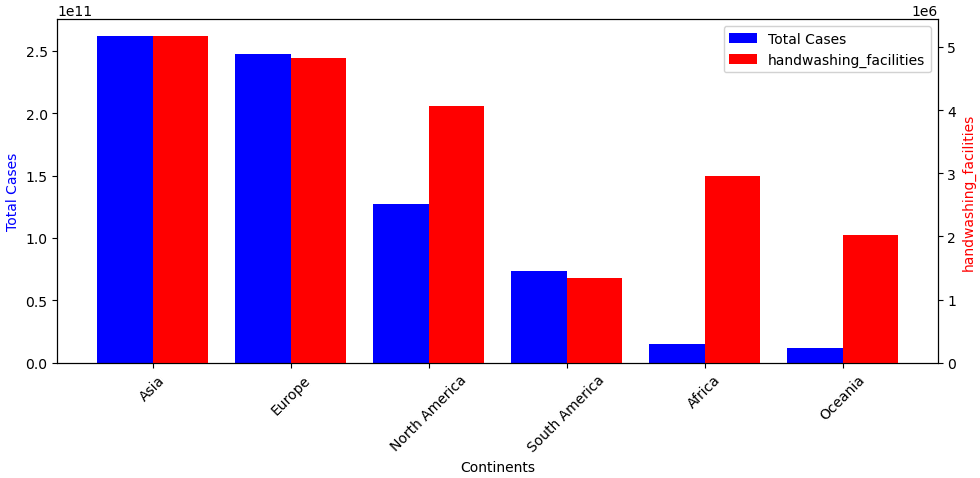

1e11
1e9
Total Cases
excess_mortality_cumulative_absolute
Total Cases
excess_mortality_cumulative_absolute
Asia
Europe
North America
South America
Africa
Oceania
Continents

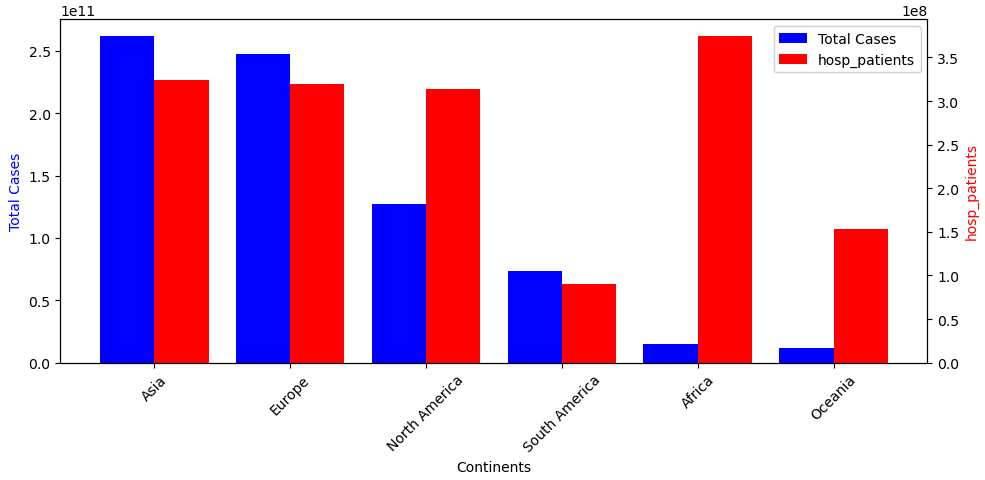

1e11
1e8
Total Cases
excess_mortality_cumulative_per_million
Total Cases
excess_mortality_cumulative_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

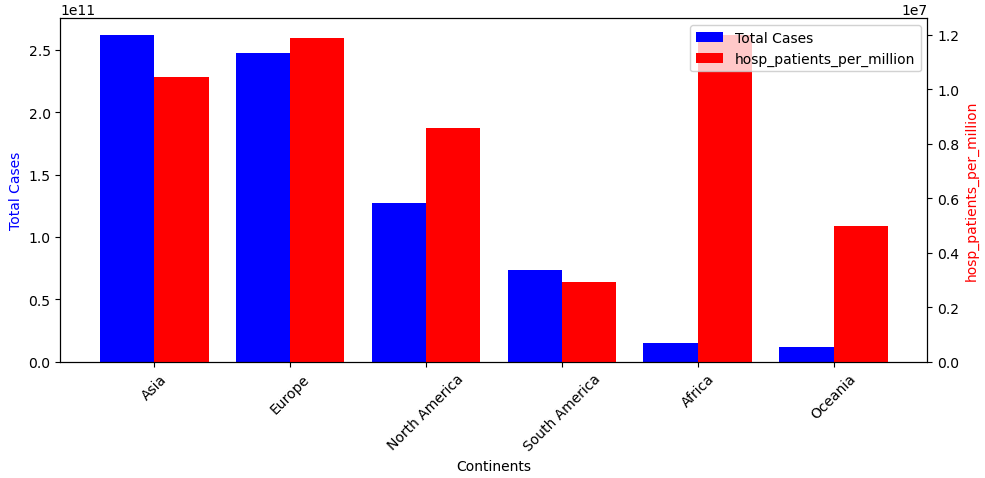

1e11
1e6
Total Cases
extreme_poverty
Total Cases
extreme_poverty
Asia
Europe
North America
South America
Africa
Oceania
Continents

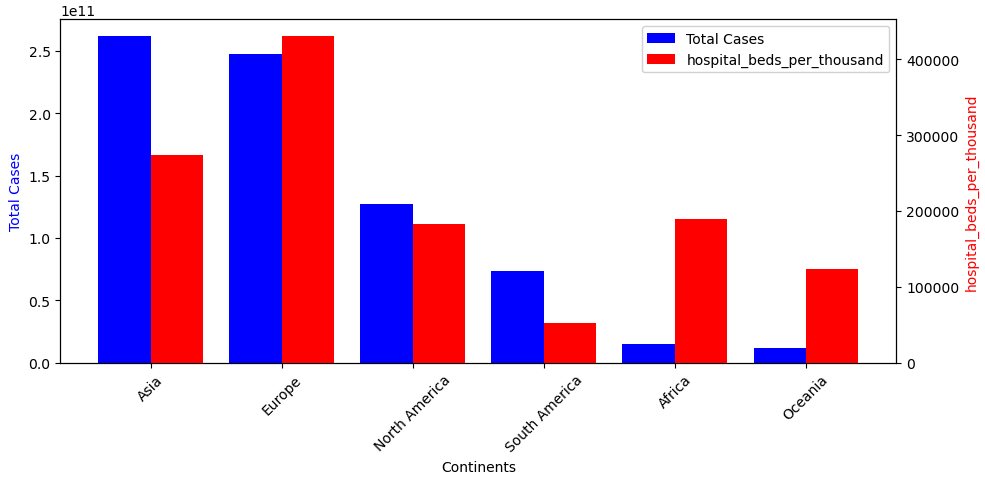

1e11
1e6
Total Cases
female_smokers
Total Cases
female_smokers
Asia
Europe
North America
South America
Africa
Oceania
Continents

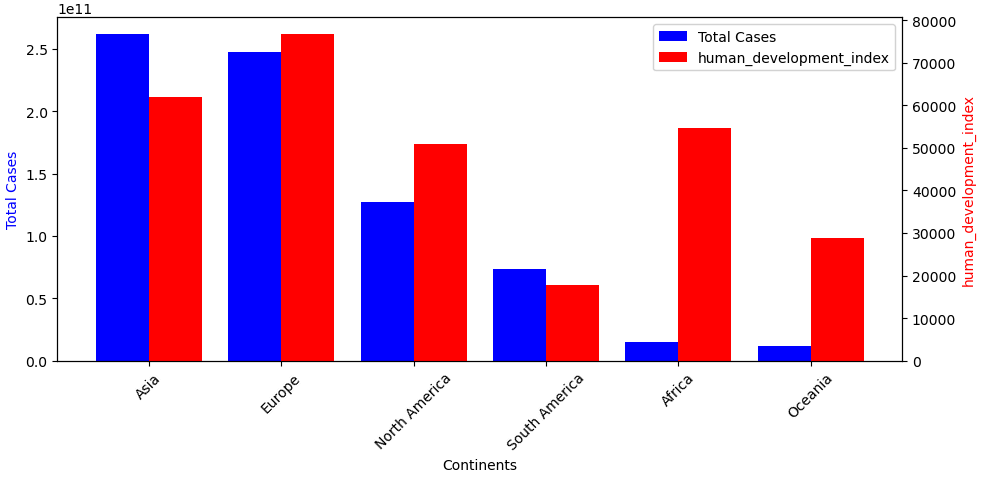

1e11
1e9
Total Cases
gdp_per_capita
Total Cases
gdp_per_capita
Asia
Europe
North America
South America
Africa
Oceania
Continents

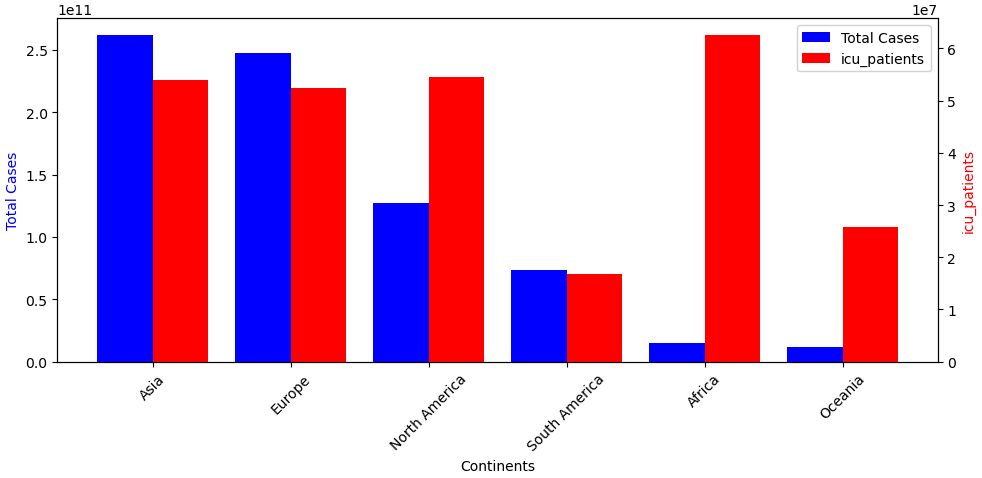

1e11
1e6
Total Cases
handwashing_facilities
Total Cases
handwashing_facilities
Asia
Europe
North America
South America
Africa
Oceania
Continents

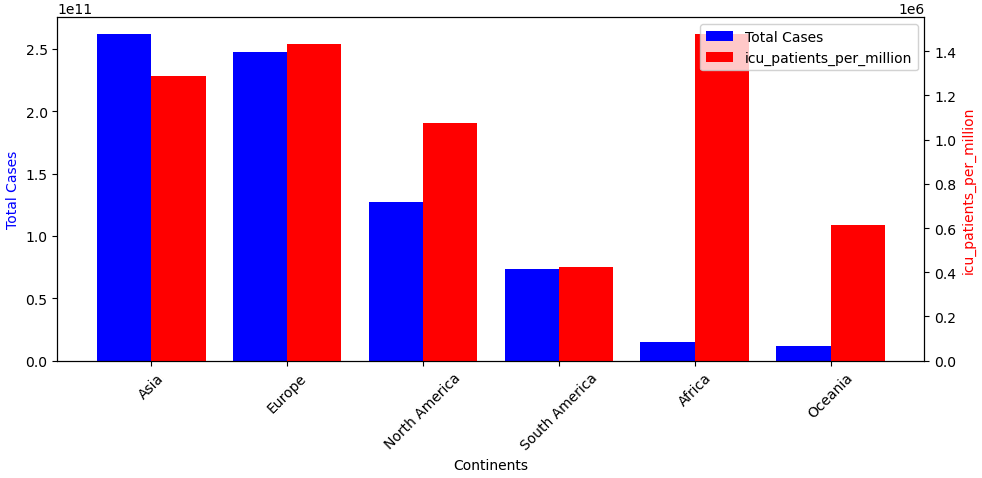

1e11
1e8
Total Cases
hosp_patients
Total Cases
hosp_patients
Asia
Europe
North America
South America
Africa
Oceania
Continents

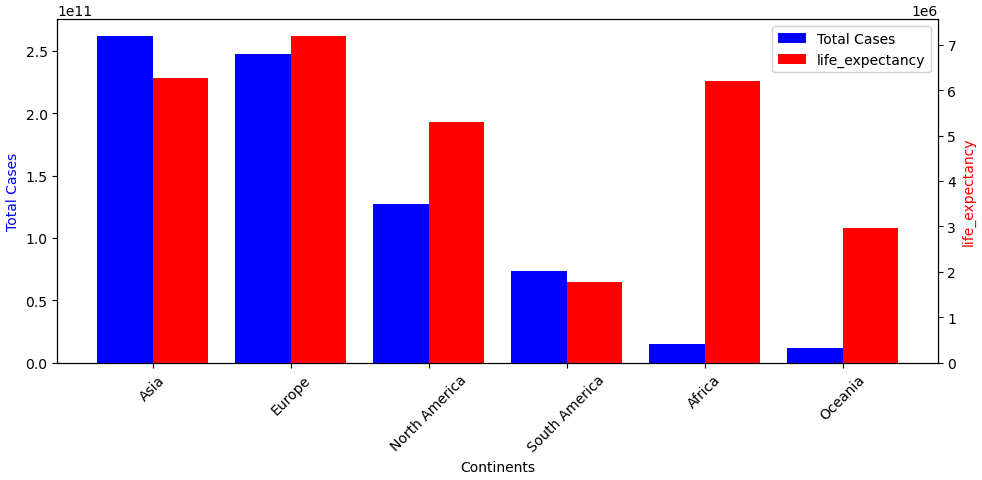

1e11
1e7
Total Cases
hosp_patients_per_million
Total Cases
hosp_patients_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

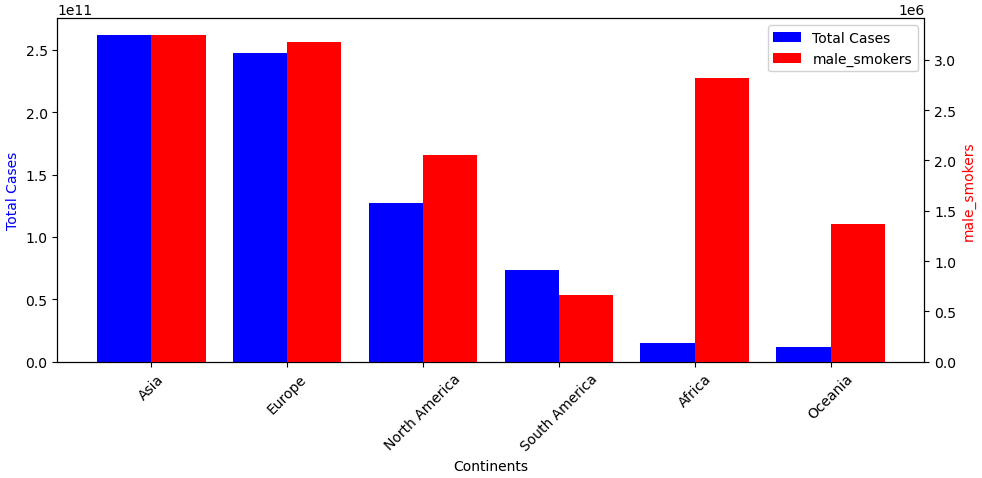

1e11
Total Cases
hospital_beds_per_thousand
Total Cases
hospital_beds_per_thousand
Asia
Europe
North America
South America
Africa
Oceania
Continents

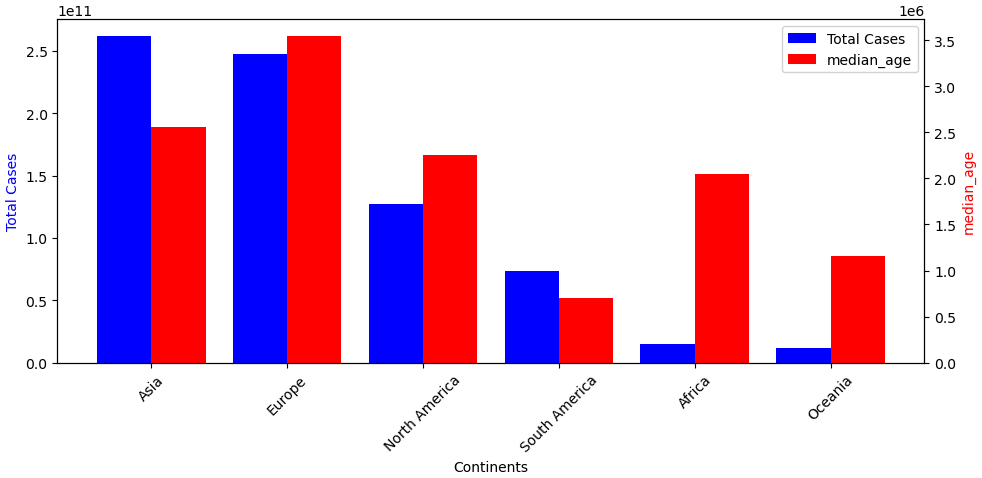

1e11
Total Cases
human_development_index
Total Cases
human_development_index
Asia
Europe
North America
South America
Africa
Oceania
Continents

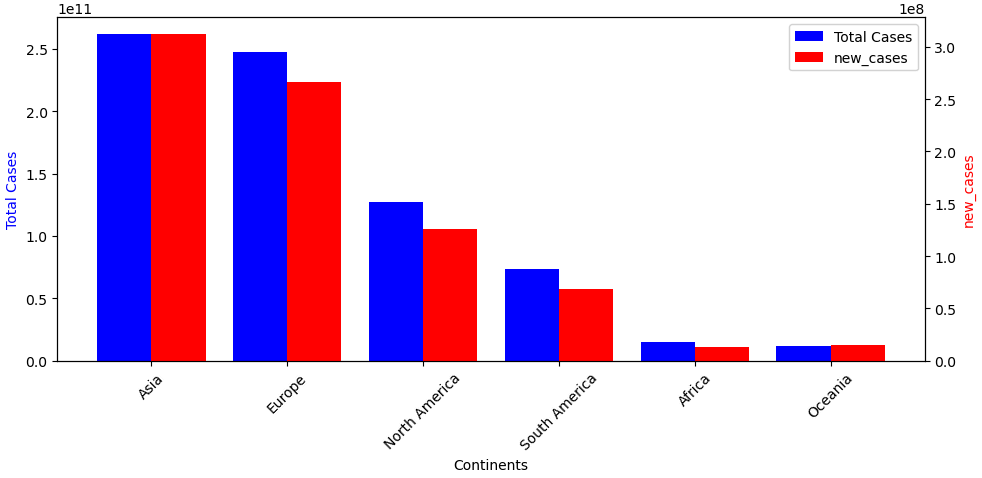

1e11
1e7
Total Cases
icu_patients
Total Cases
icu_patients
Asia
Europe
North America
South America
Africa
Oceania
Continents

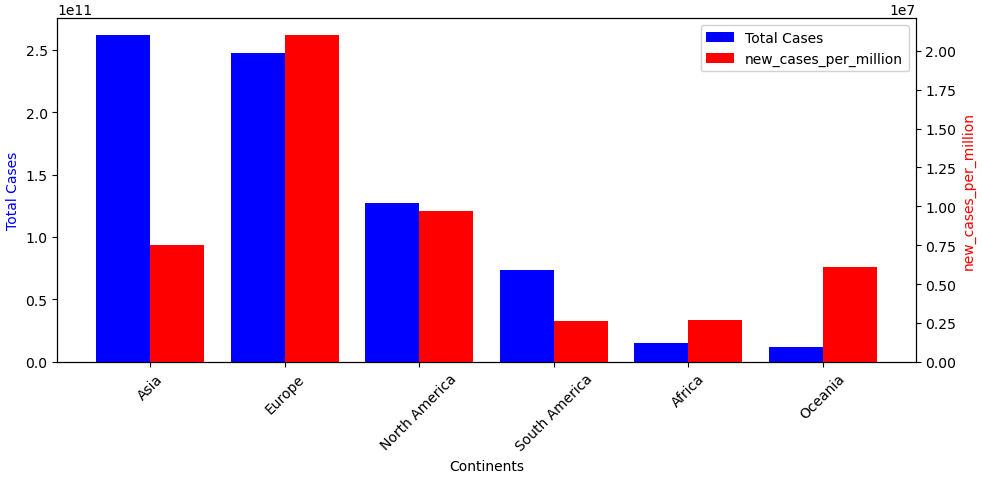

1e11
1e6
Total Cases
icu_patients_per_million
Total Cases
icu_patients_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

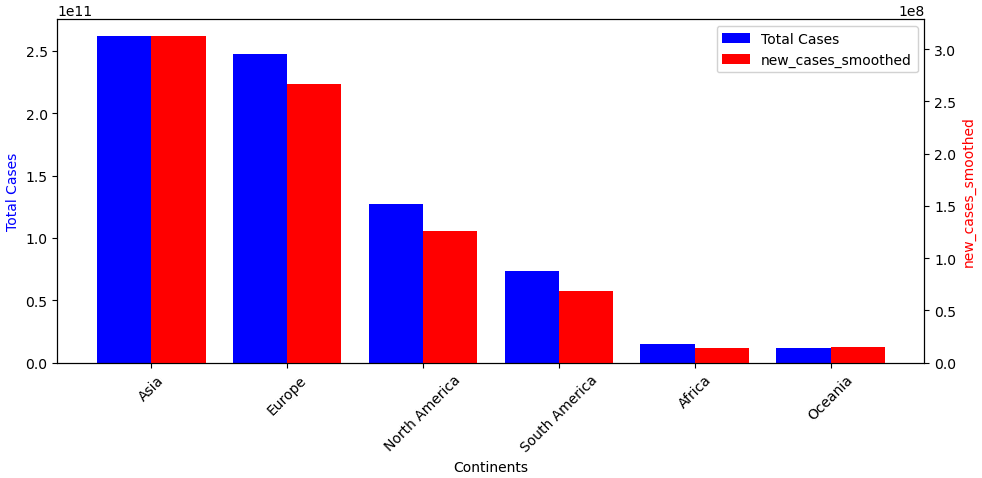

1e11
1e6
Total Cases
life_expectancy
Total Cases
life_expectancy
Asia
Europe
North America
South America
Africa
Oceania
Continents

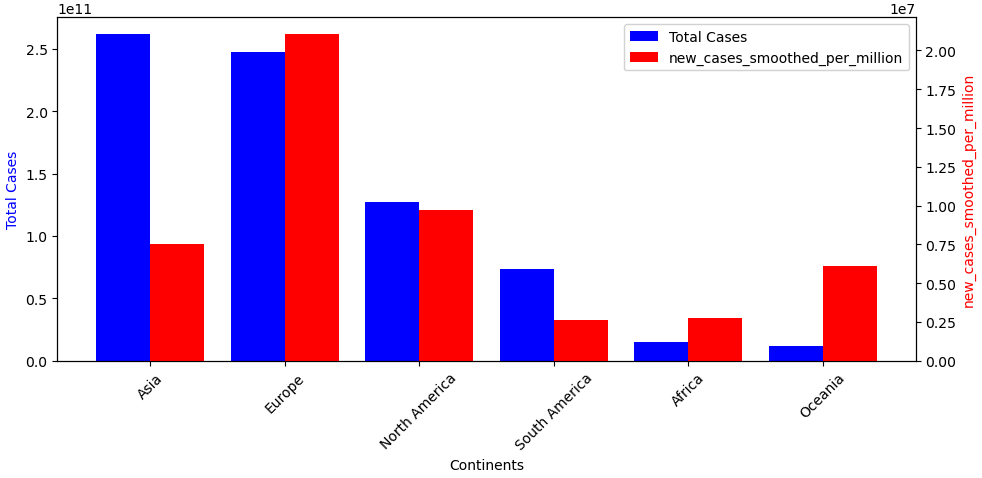

1e11
1e6
Total Cases
male_smokers
Total Cases
male_smokers
Asia
Europe
North America
South America
Africa
Oceania
Continents

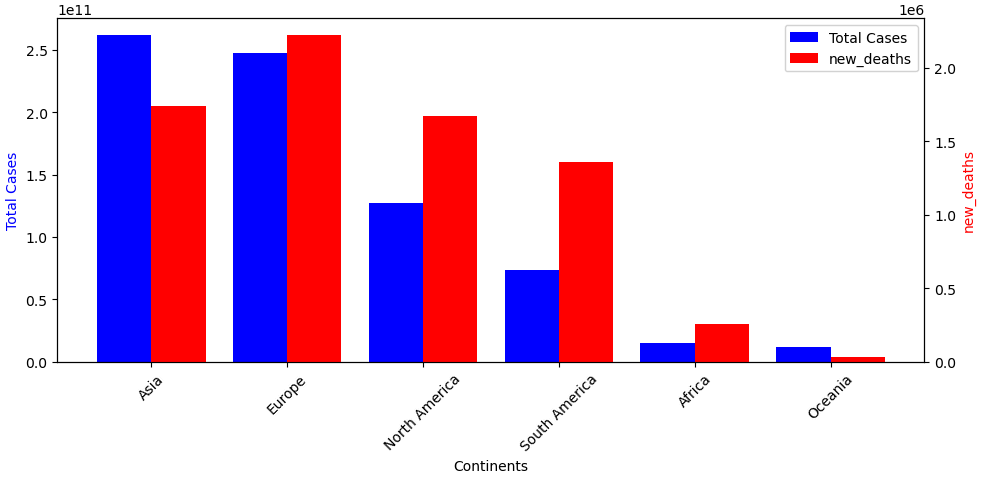

1e11
1e6
Total Cases
median_age
Total Cases
median_age
Asia
Europe
North America
South America
Africa
Oceania
Continents

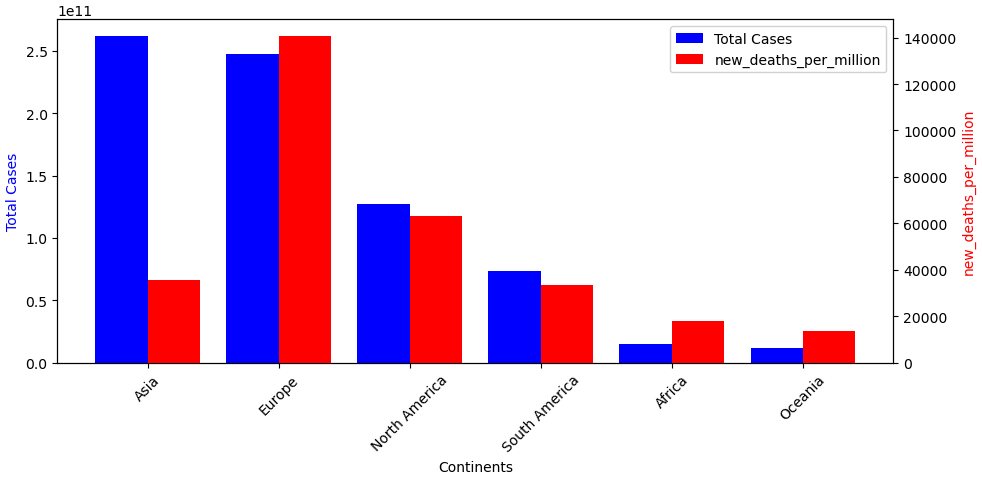

1e11
1e8
Total Cases
new_cases
Total Cases
new_cases
Asia
Europe
North America
South America
Africa
Oceania
Continents

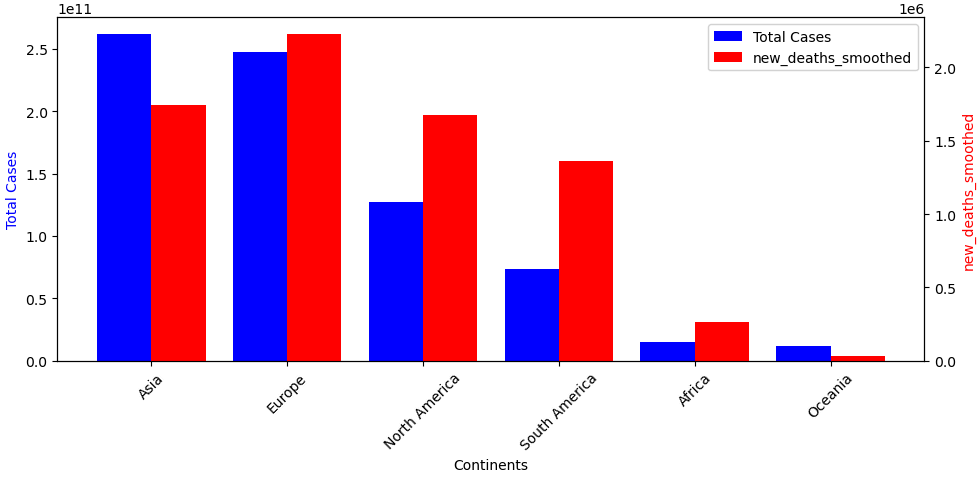

1e11
1e7
Total Cases
new_cases_per_million
Total Cases
new_cases_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

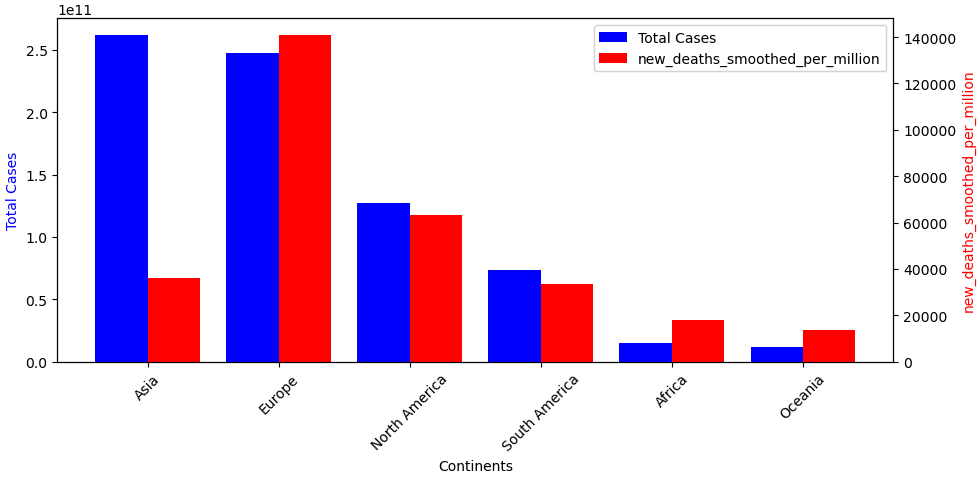

1e11
1e8
Total Cases
new_cases_smoothed
Total Cases
new_cases_smoothed
Asia
Europe
North America
South America
Africa
Oceania
Continents

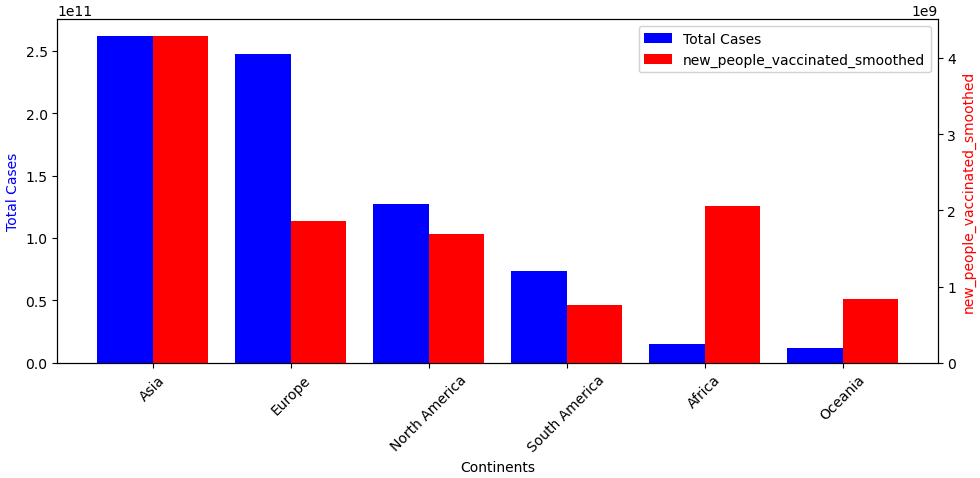

1e11
1e7
Total Cases
new_cases_smoothed_per_million
Total Cases
new_cases_smoothed_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

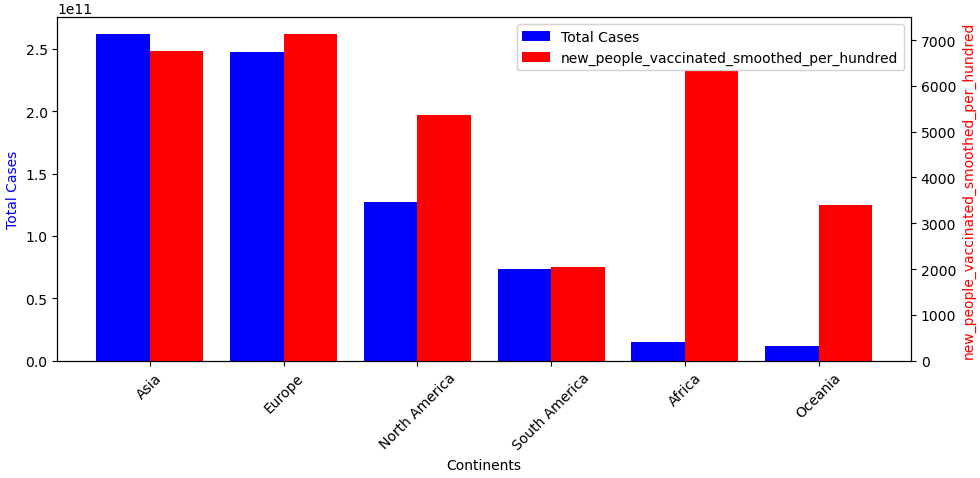

1e11
1e6
Total Cases
new_deaths
Total Cases
new_deaths
Asia
Europe
North America
South America
Africa
Oceania
Continents

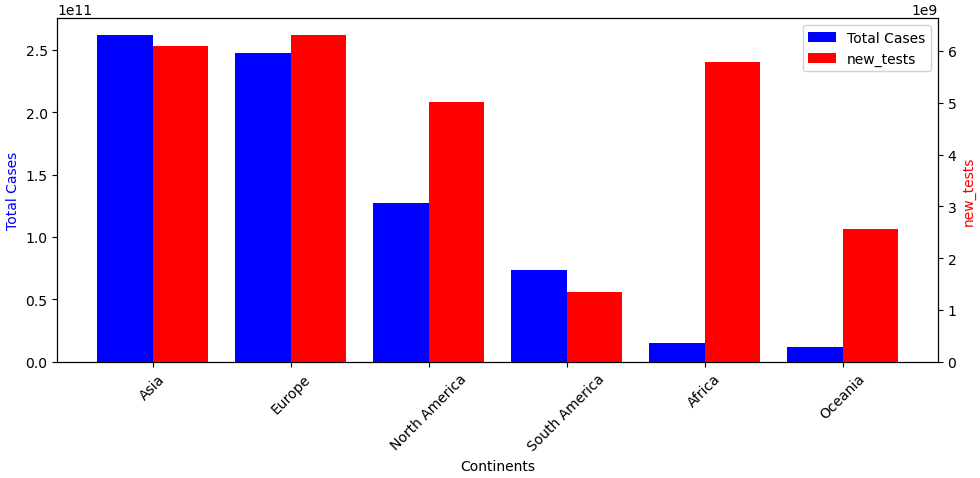

1e11
Total Cases
new_deaths_per_million
Total Cases
new_deaths_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

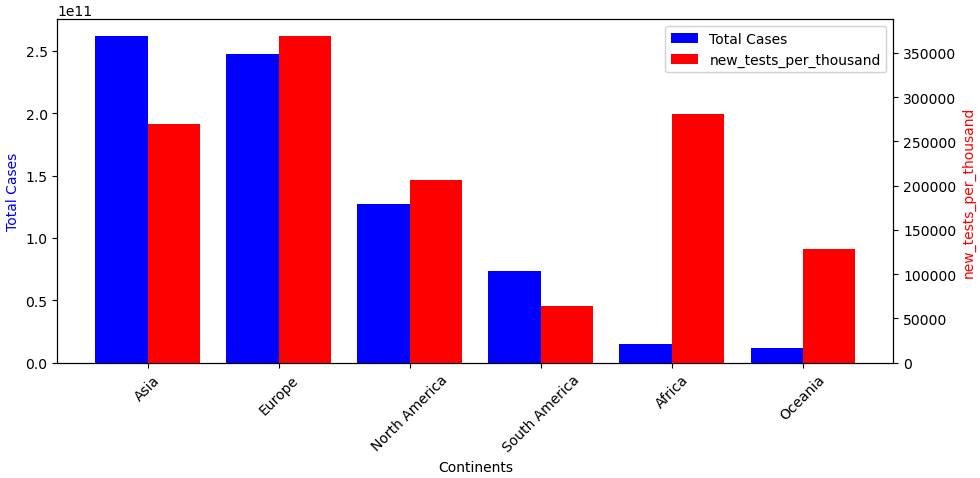

1e11
1e6
Total Cases
new_deaths_smoothed
Total Cases
new_deaths_smoothed
Asia
Europe
North America
South America
Africa
Oceania
Continents

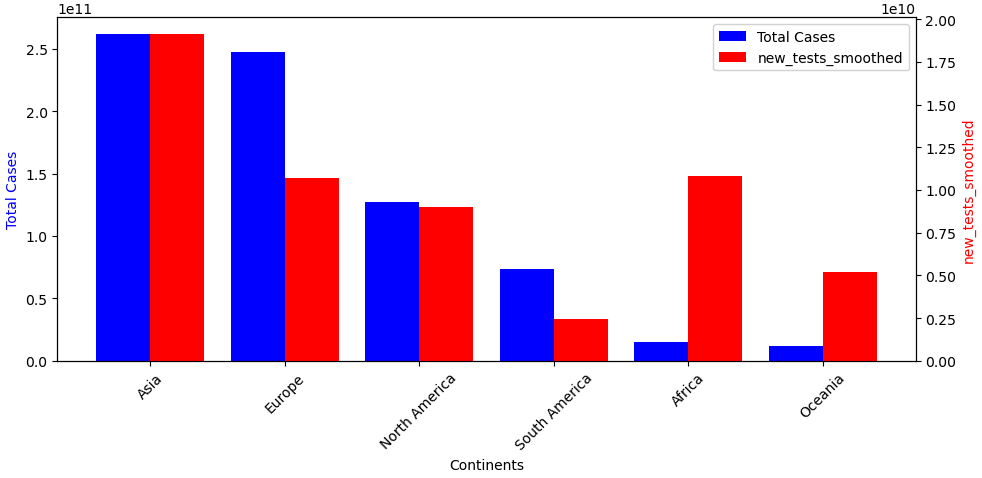

1e11
Total Cases
new_deaths_smoothed_per_million
Total Cases
new_deaths_smoothed_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

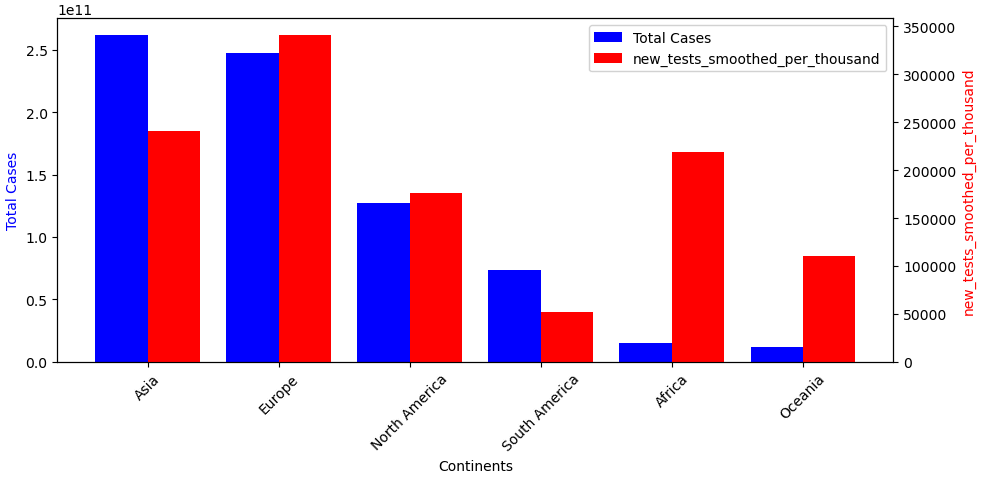

1e11
1e9
Total Cases
new_people_vaccinated_smoothed
Total Cases
new_people_vaccinated_smoothed
Asia
Europe
North America
South America
Africa
Oceania
Continents

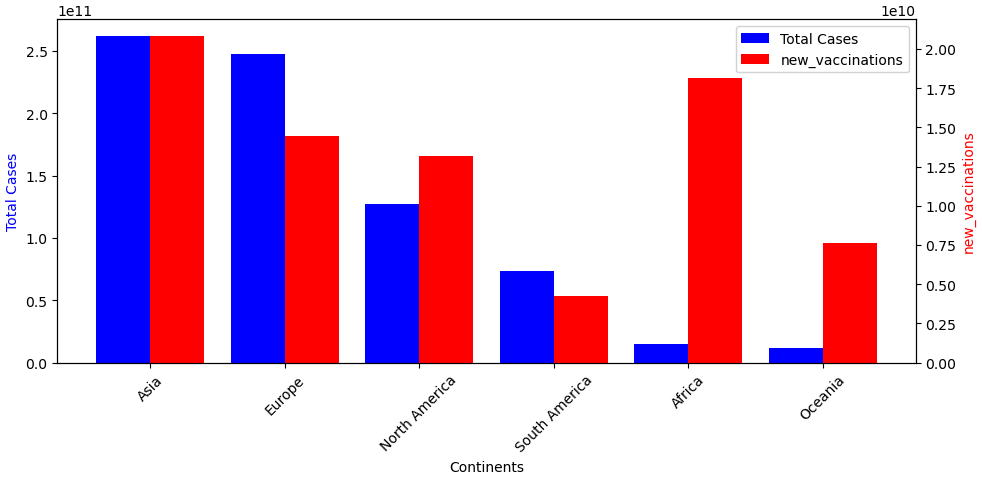

1e11
Total Cases
new_people_vaccinated_smoothed_per_hundred
Total Cases
new_people_vaccinated_smoothed_per_hundred
Asia
Europe
North America
South America
Africa
Oceania
Continents

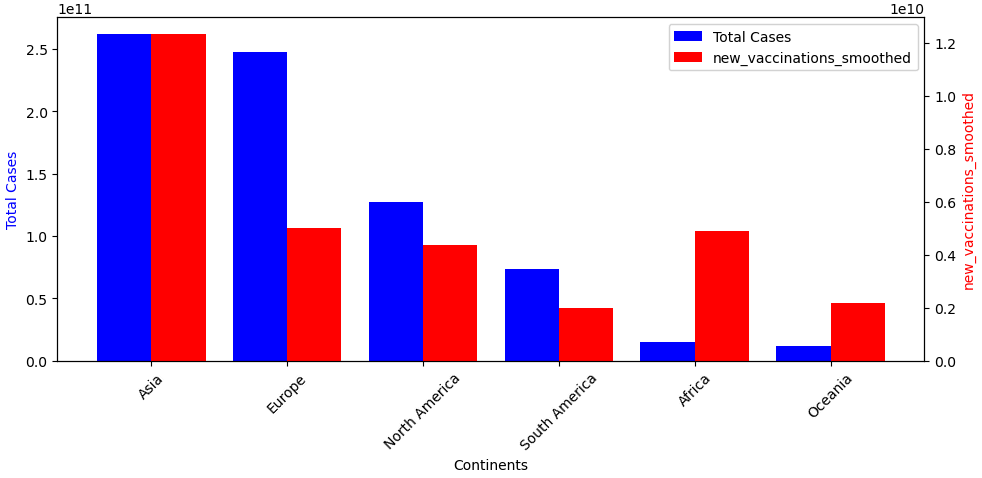

1e11
1e9
Total Cases
new_tests
Total Cases
new_tests
Asia
Europe
North America
South America
Africa
Oceania
Continents

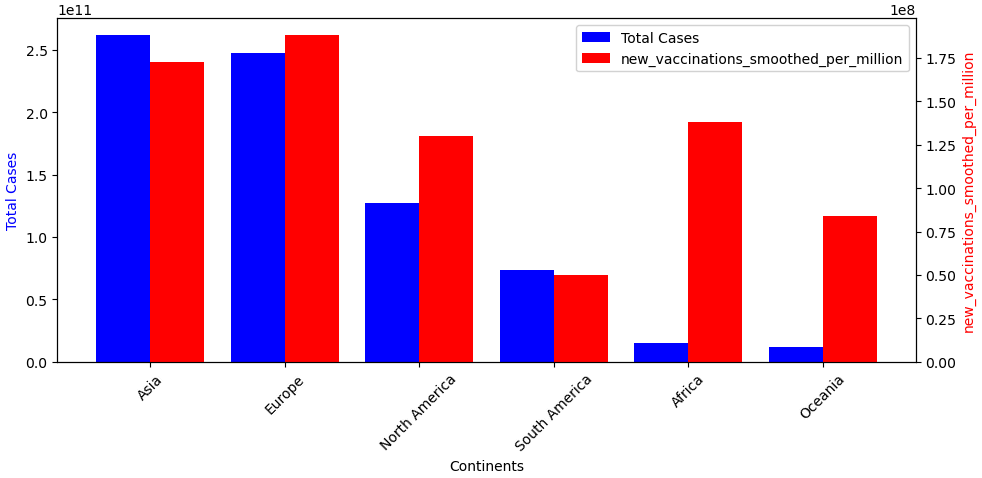

1e11
Total Cases
new_tests_per_thousand
Total Cases
new_tests_per_thousand
Asia
Europe
North America
South America
Africa
Oceania
Continents

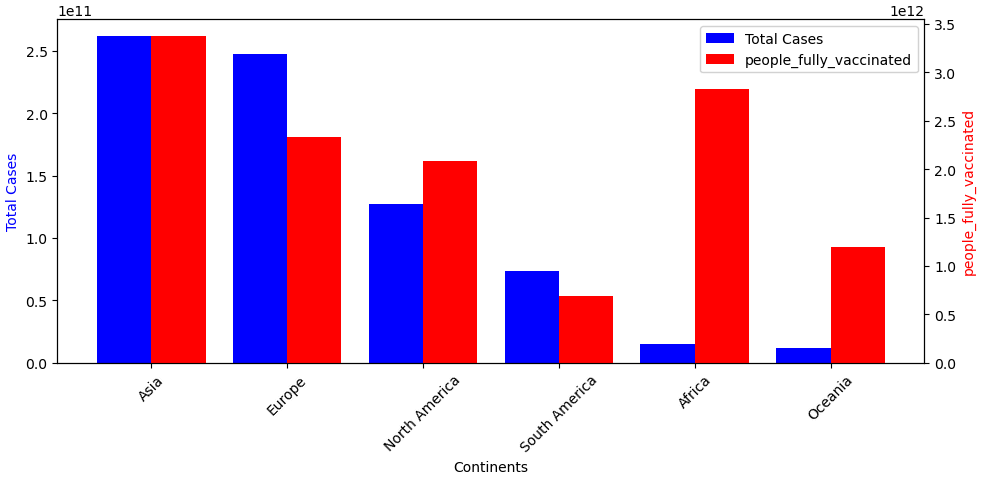

1e11
1e10
Total Cases
new_tests_smoothed
Total Cases
new_tests_smoothed
Asia
Europe
North America
South America
Africa
Oceania
Continents

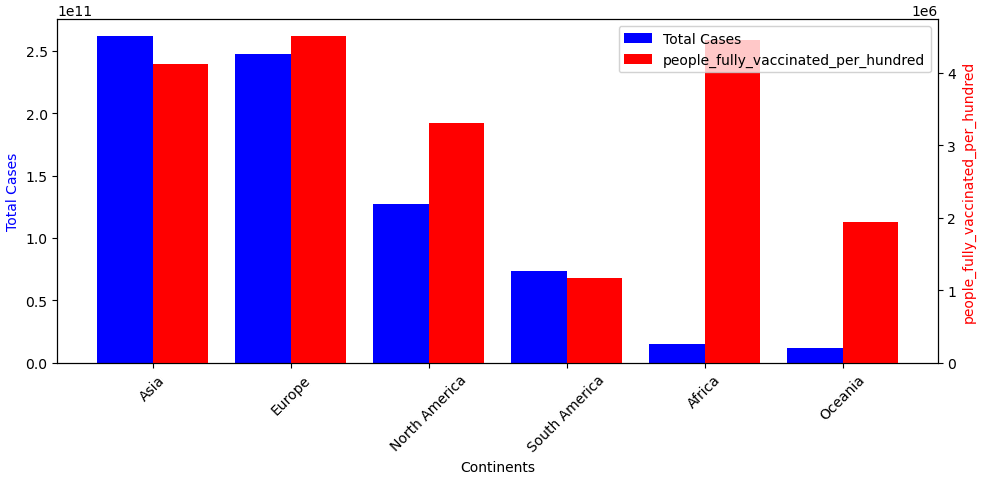

1e11
Total Cases
new_tests_smoothed_per_thousand
Total Cases
new_tests_smoothed_per_thousand
Asia
Europe
North America
South America
Africa
Oceania
Continents

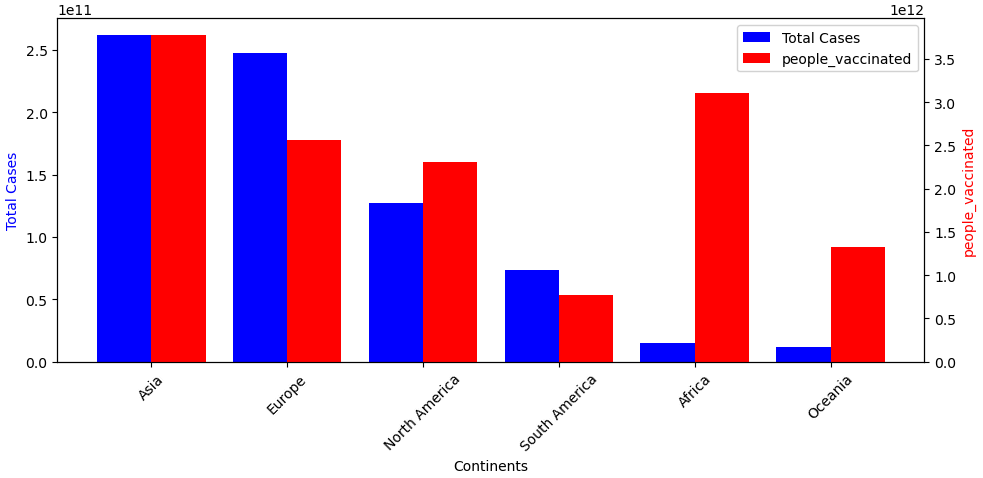

1e11
1e10
Total Cases
new_vaccinations
Total Cases
new_vaccinations
Asia
Europe
North America
South America
Africa
Oceania
Continents

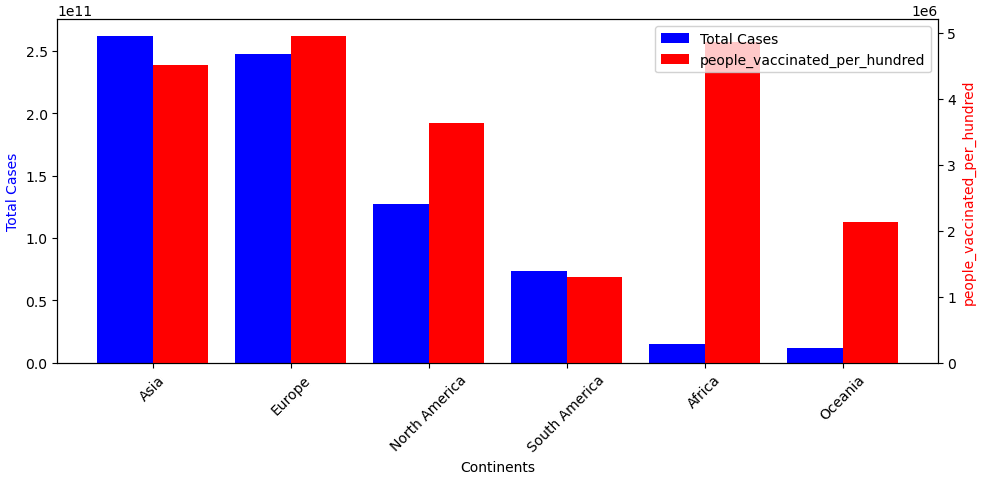

1e11
1e10
Total Cases
new_vaccinations_smoothed
Total Cases
new_vaccinations_smoothed
Asia
Europe
North America
South America
Africa
Oceania
Continents

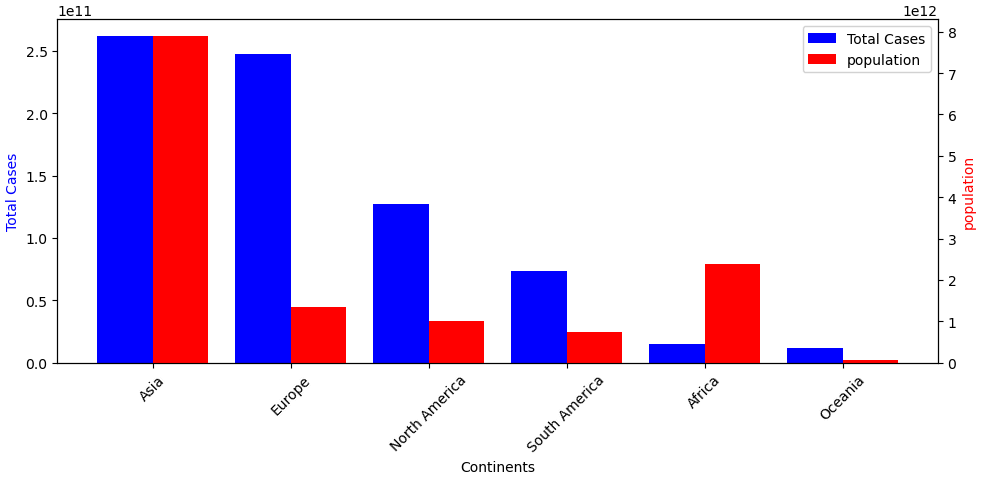

1e11
1e8
Total Cases
new_vaccinations_smoothed_per_million
Total Cases
new_vaccinations_smoothed_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

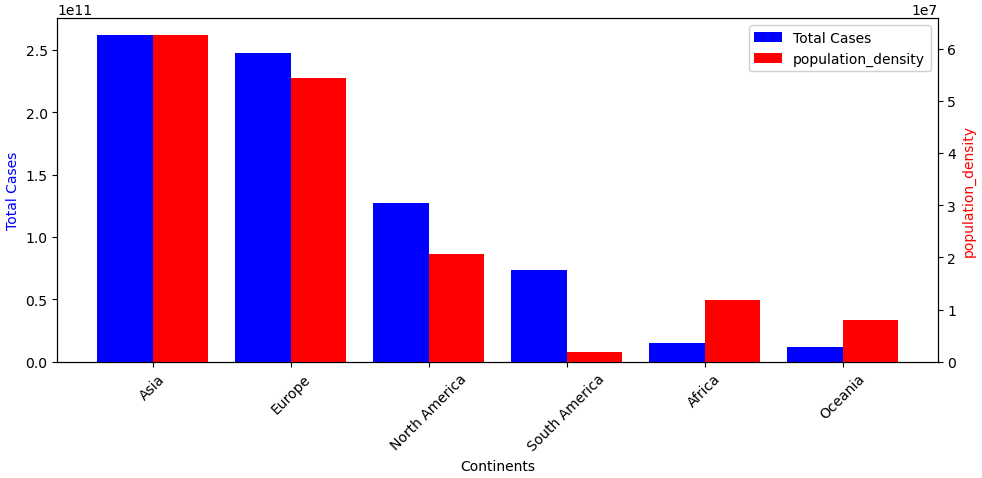

1e11
1e12
Total Cases
people_fully_vaccinated
Total Cases
people_fully_vaccinated
Asia
Europe
North America
South America
Africa
Oceania
Continents

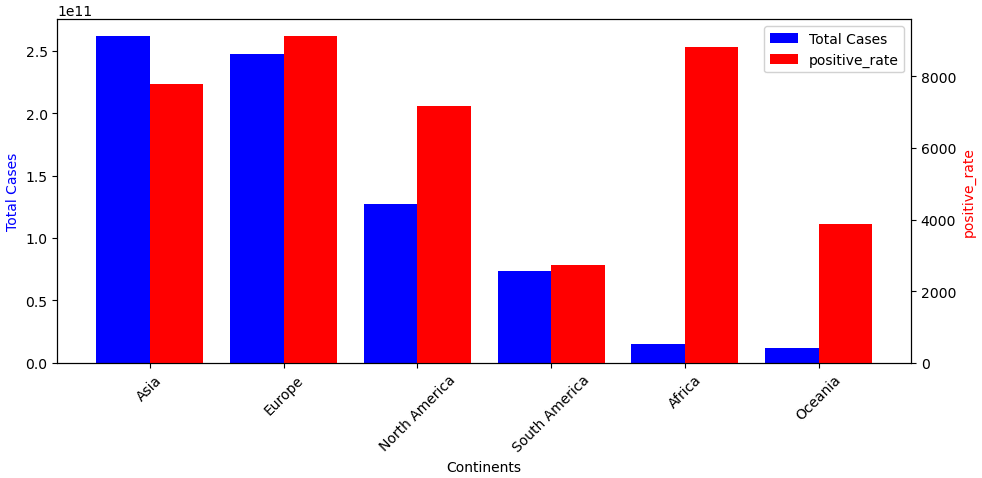

1e11
1e6
Total Cases
people_fully_vaccinated_per_hundred
Total Cases
people_fully_vaccinated_per_hundred
Asia
Europe
North America
South America
Africa
Oceania
Continents

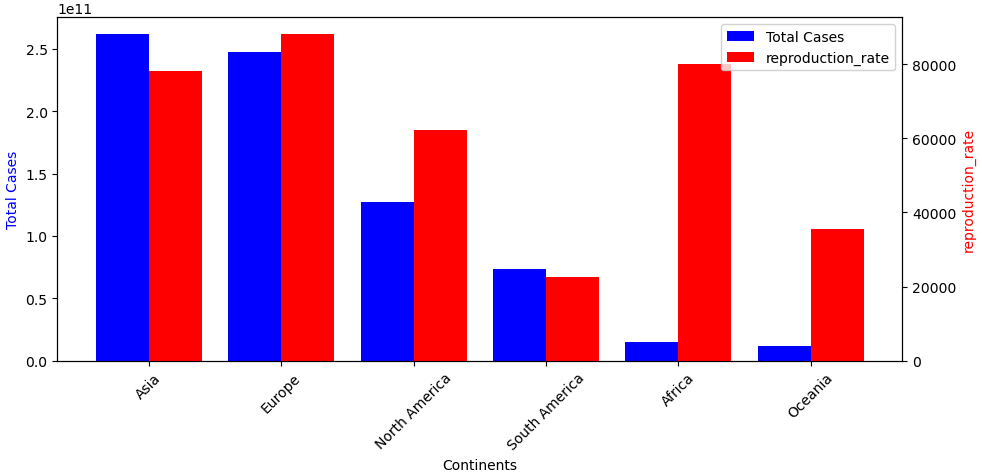

1e11
1e12
Total Cases
people_vaccinated
Total Cases
people_vaccinated
Asia
Europe
North America
South America
Africa
Oceania
Continents

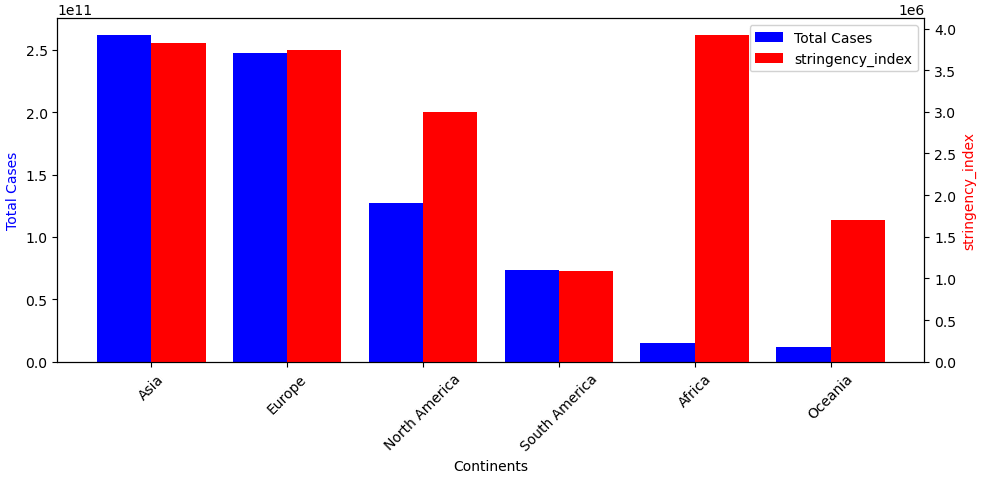

1e11
1e6
Total Cases
people_vaccinated_per_hundred
Total Cases
people_vaccinated_per_hundred
Asia
Europe
North America
South America
Africa
Oceania
Continents

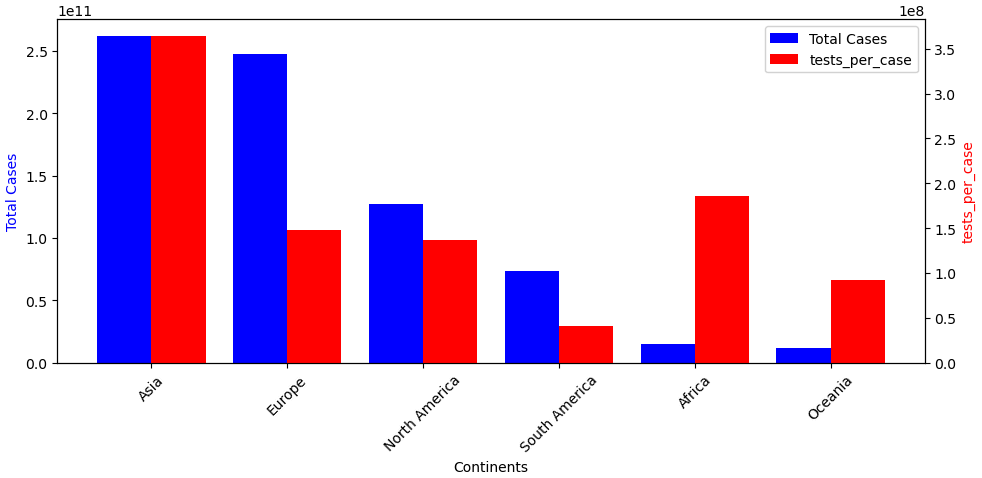

1e11
1e12
Total Cases
population
Total Cases
population
Asia
Europe
North America
South America
Africa
Oceania
Continents

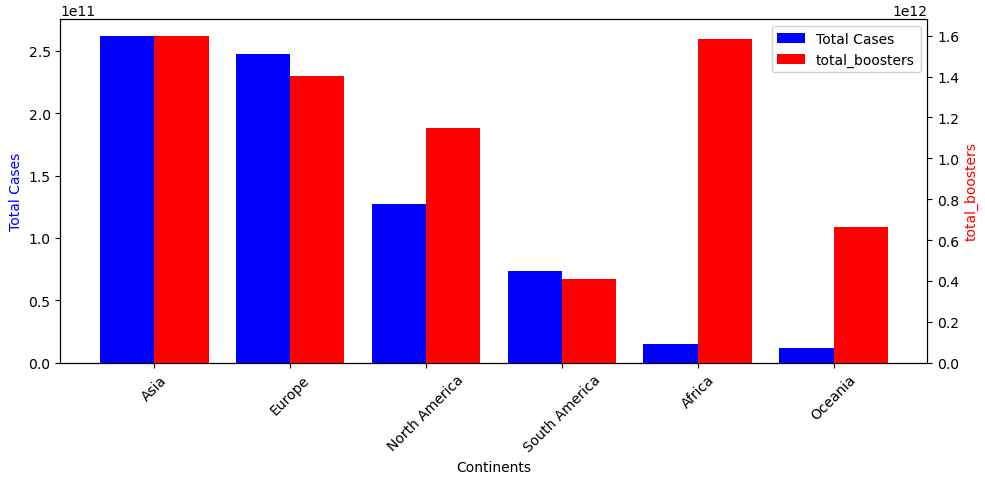

1e11
1e7
Total Cases
population_density
Total Cases
population_density
Asia
Europe
North America
South America
Africa
Oceania
Continents

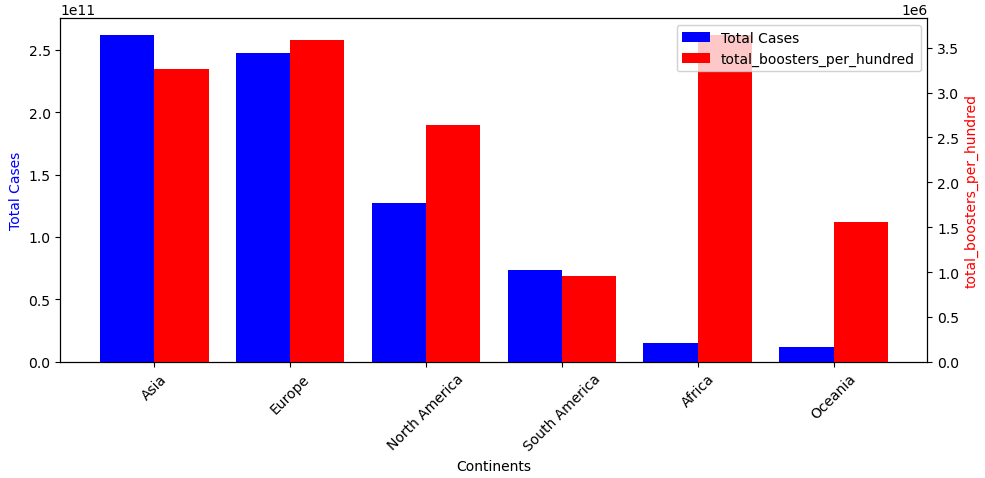

1e11
Total Cases
positive_rate
Total Cases
positive_rate
Asia
Europe
North America
South America
Africa
Oceania
Continents

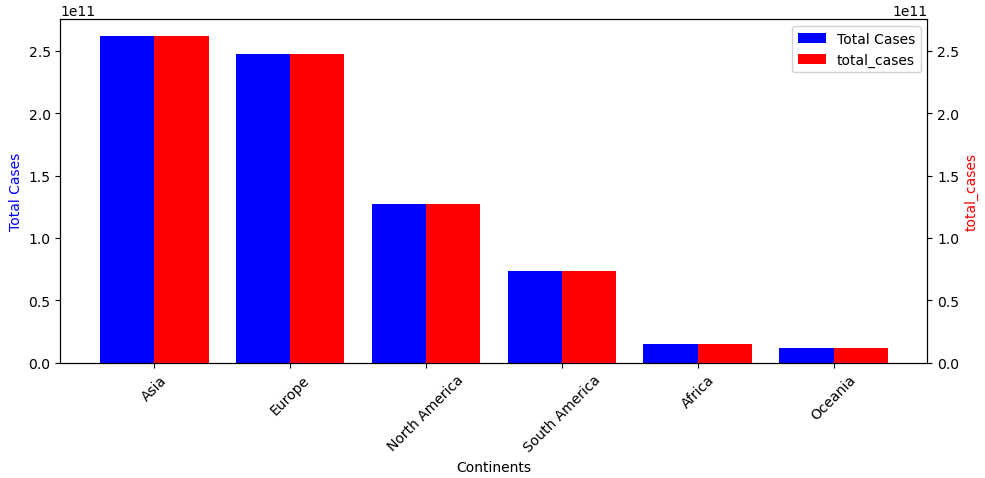

1e11
Total Cases
reproduction_rate
Total Cases
reproduction_rate
0.0
0.5
1.0
1.5
2.0
2.5
0
20000
40000
60000
80000
Asia
Europe
North America
South America
Africa
Oceania
Continents

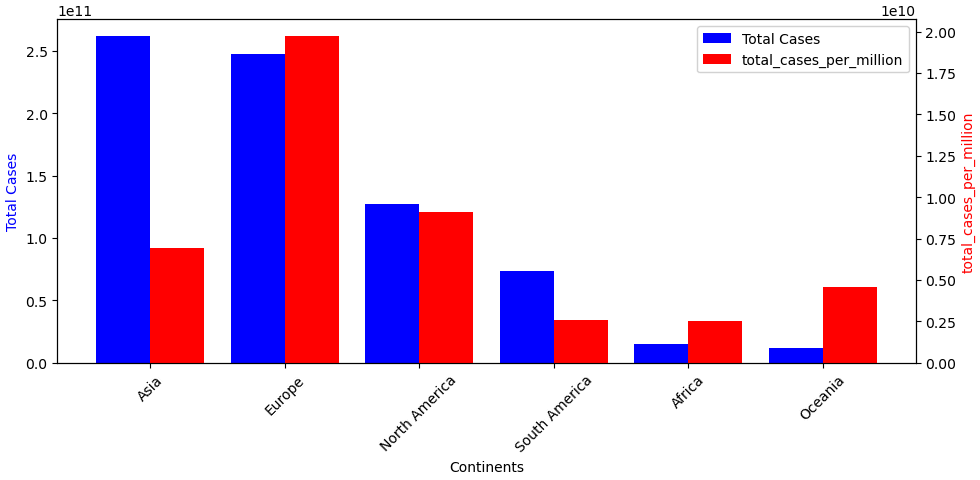

1e11
1e6
Total Cases
stringency_index
Total Cases
stringency_index
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Asia
Europe
North America
South America
Africa
Oceania
Continents

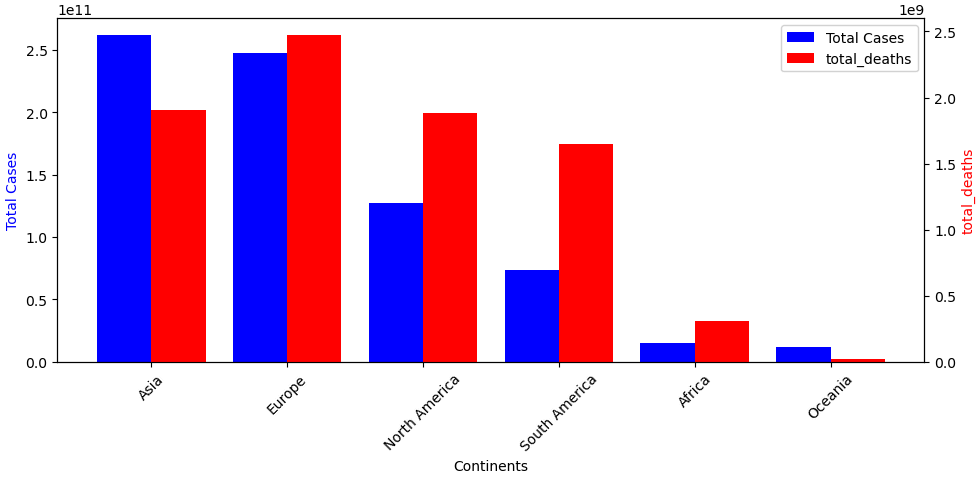

1e11
1e8
Total Cases
tests_per_case
Total Cases
tests_per_case
0.0
0.5
1.0
1.5
2.0
2.5
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
Asia
Europe
North America
South America
Africa
Oceania
Continents

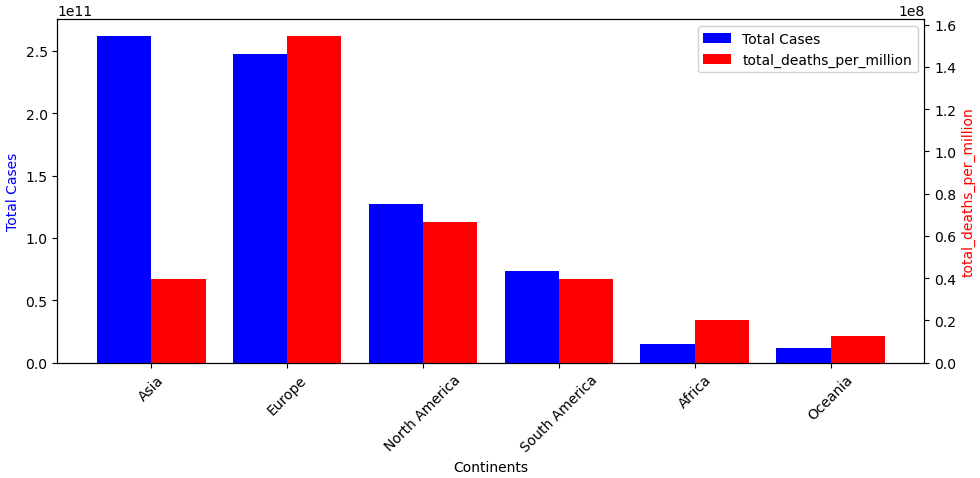

1e11
1e12
Total Cases
total_boosters
Total Cases
total_boosters
Asia
Europe
North America
South America
Africa
Oceania
Continents

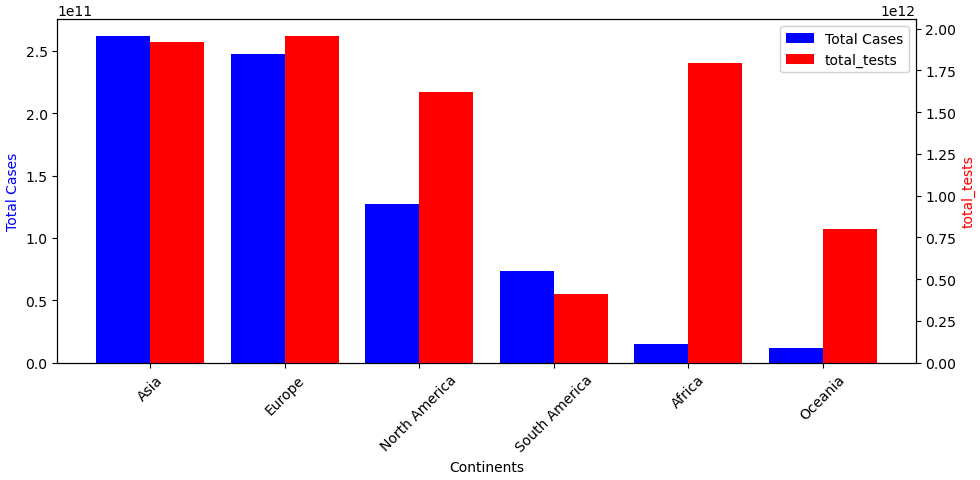

1e11
1e6
Total Cases
total_boosters_per_hundred
Total Cases
total_boosters_per_hundred
Asia
Europe
North America
South America
Africa
Oceania
Continents

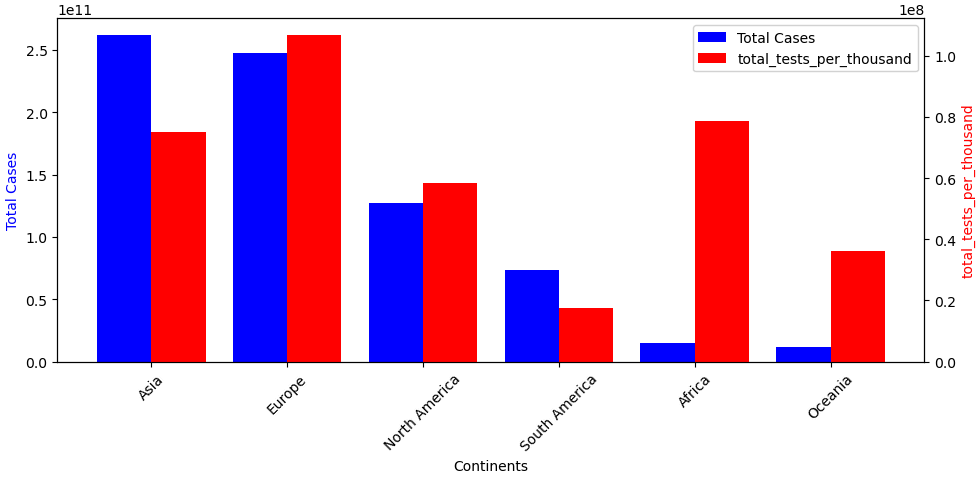

1e11
1e11
Total Cases
total_cases
Total Cases
total_cases
Asia
Europe
North America
South America
Africa
Oceania
Continents

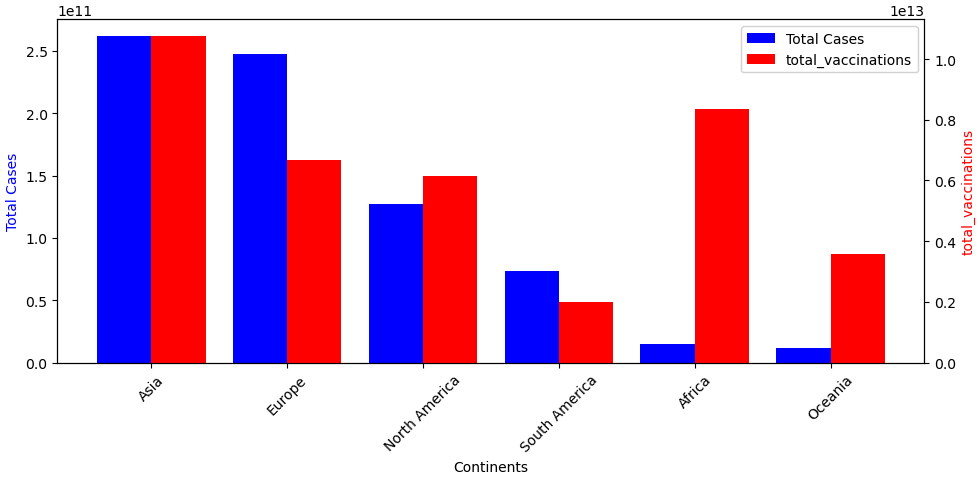

1e11
1e10
Total Cases
total_cases_per_million
Total Cases
total_cases_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

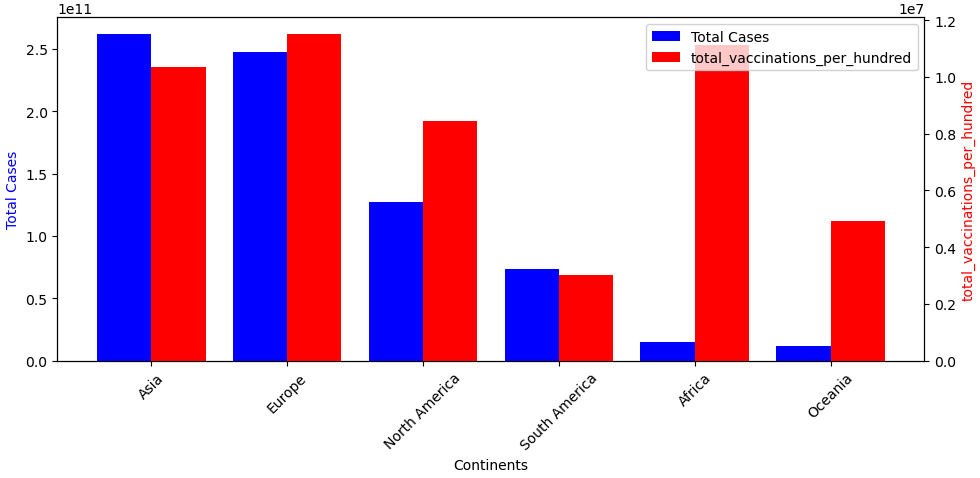

1e11
1e9
Total Cases
total_deaths
Total Cases
total_deaths
Asia
Europe
North America
South America
Africa
Oceania
Continents

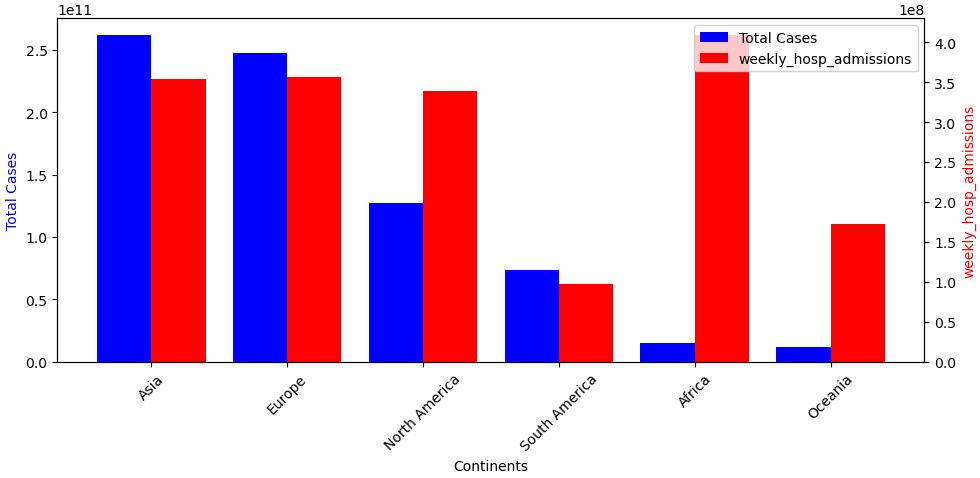

1e11
1e8
Total Cases
total_deaths_per_million
Total Cases
total_deaths_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents

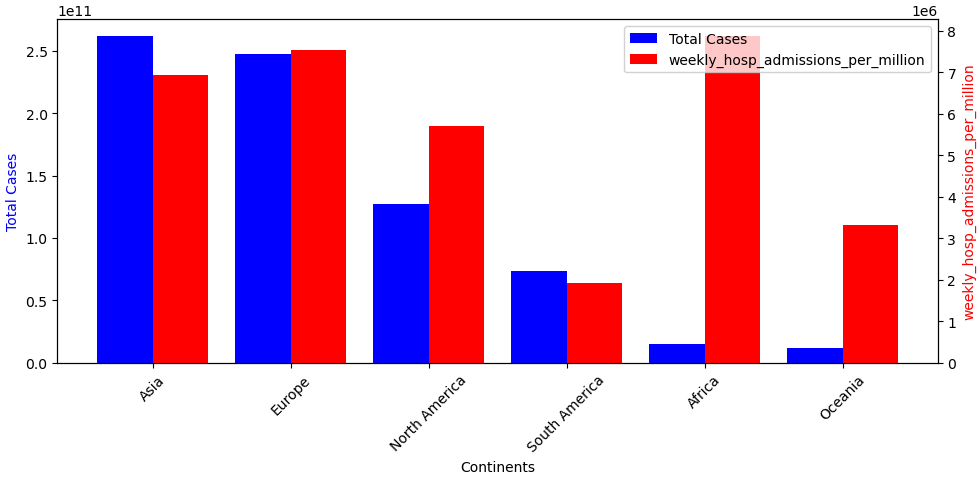

1e11
1e12
Total Cases
total_tests
Total Cases
total_tests
Asia
Europe
North America
South America
Africa
Oceania
Continents

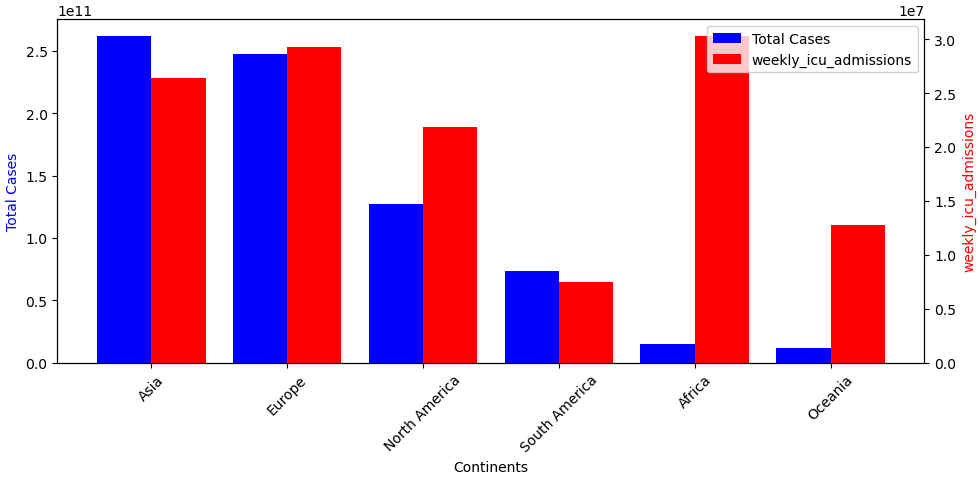

1e11
1e8
Total Cases
total_tests_per_thousand
Total Cases
total_tests_per_thousand
Asia
Europe
North America
South America
Africa
Oceania
Continents

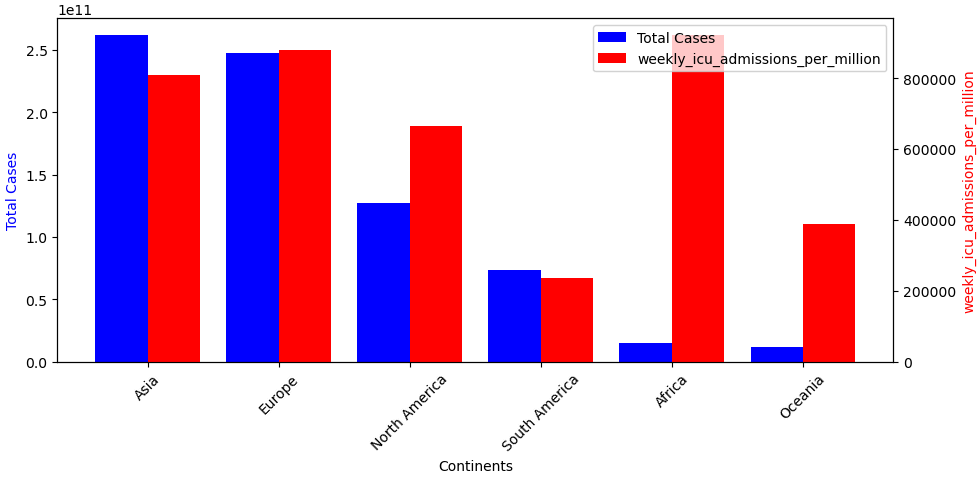

1e11
1e13
Total Cases
total_vaccinations
Total Cases
total_vaccinations
Asia
Europe
North America
South America
Africa
Oceania
Continents


1e11
1e7
Total Cases
total_vaccinations_per_hundred
Total Cases
total_vaccinations_per_hundred
Asia
Europe
North America
South America
Africa
Oceania
Continents


1e11
1e8
Total Cases
weekly_hosp_admissions
Total Cases
weekly_hosp_admissions
Asia
Europe
North America
South America
Africa
Oceania
Continents


1e11
1e6
Total Cases
weekly_hosp_admissions_per_million
Total Cases
weekly_hosp_admissions_per_million
Asia
Europe
North America
South America
Africa
Oceania
Continents





Of a indicators, Africa seems to have a high extreme value for all majority.

Observed that Continents with high Population Density which is supported by the Reproduction rate also have a positvie correlation with the numbers of total cases in such cintinents such as Asia, Europe and North America. Also, they appear to be top 3 continents with the high Life Expectancy value Europe being the highest in GDP per capita, follow by Asia and North America.

Now, to try to investigate the Death Cases in all continents in order to determine the reason.

In [388…
```
pivot_table3.sort_values(by='total_deaths', ascending=False, inplace=True)
```

In [389…
```
pivot_table3
```

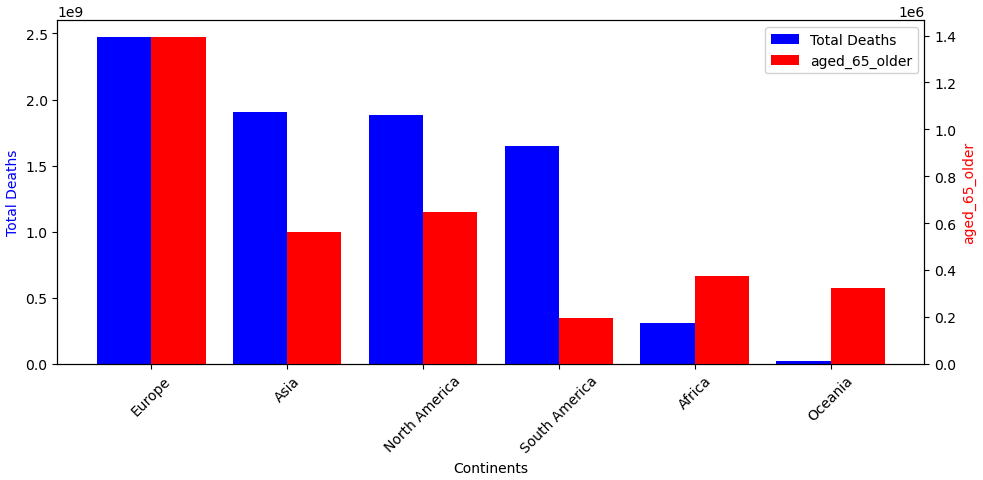
Out[389…

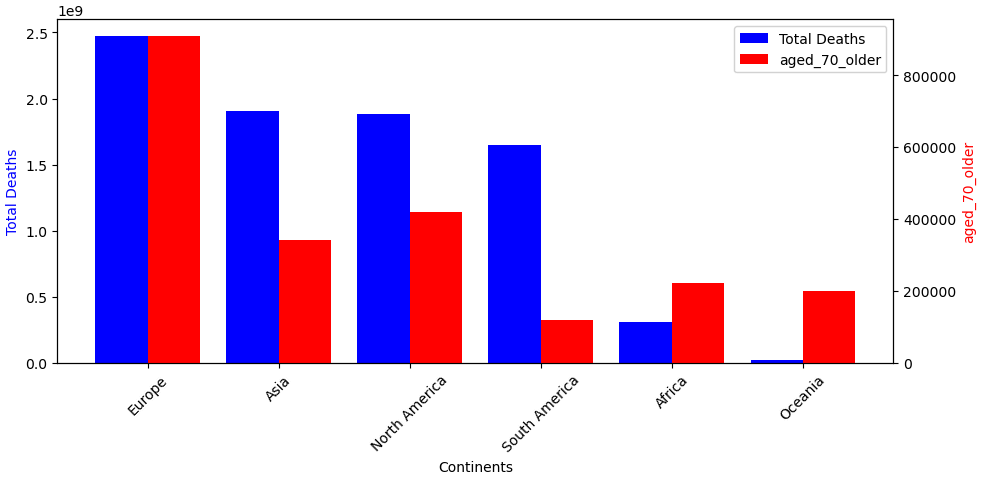
	continent	aged_65_older	aged_70_older	cardiovasc_death_rate	diabetes_prevalence	excess_mortality	excess_mortality_cumulative	excess_mortality_cumulative_absolute	excess_mortality_cumulative_pe
2	Europe	1.395855e+06	909993.920310	2.172748e+07	631621.180159	9.787496e+05	874408.172543	4.950803e+09	1.63
1	Asia	5.625091e+05	341966.806955	2.493296e+07	766978.792132	9.234700e+05	823918.277747	4.683814e+09	1.48
3	North America	6.488190e+05	419445.933264	1.522693e+07	707941.314224	7.519322e+05	671499.749125	4.030934e+09	1.21
5	South America	1.953924e+05	118241.812751	4.650284e+06	185572.446845	2.689018e+05	244398.833366	1.338279e+09	4.20
0	Africa	3.753080e+05	221929.446751	2.730064e+07	551857.456548	1.043096e+06	933224.795318	5.352761e+09	1.68
4	Oceania	3.210038e+05	199397.384260	1.285304e+07	603439.753957	4.357844e+05	387543.002048	2.225569e+09	7.04

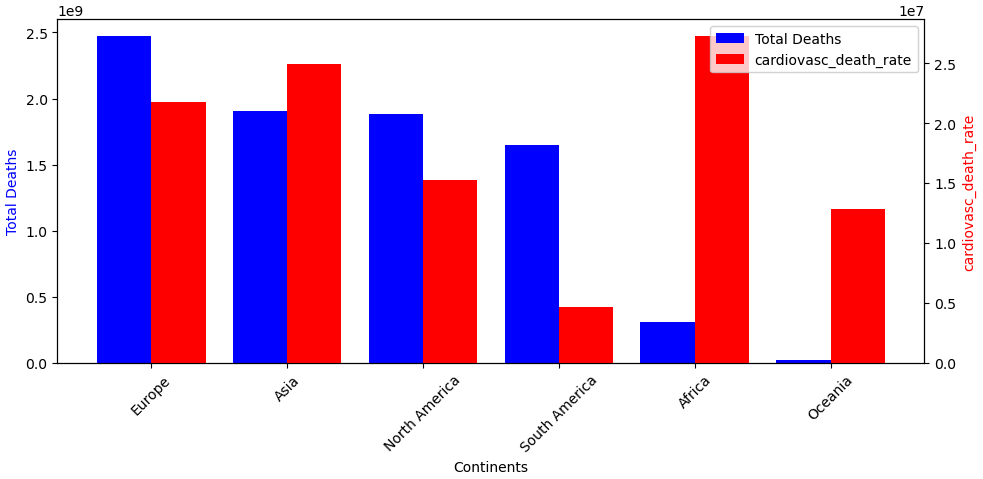
6 rows × 63 columns

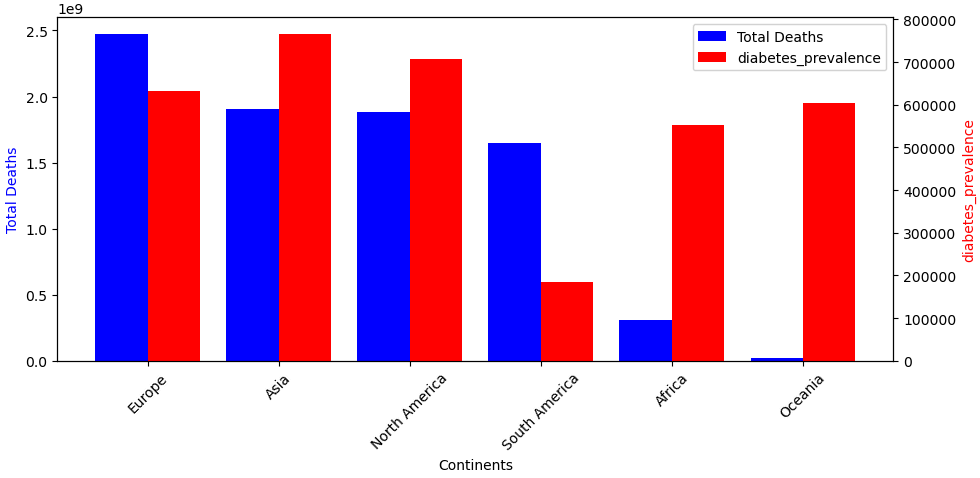
In [390…
```
for feature in pivot_table3.columns.drop('continent'):
    fig, ax = plt.subplots(figsize = (10, 5))
#     plt.title(feature, 'and Population by Continents')

    ax2 = ax.twinx()

    X_axis = np.arange(len(pivot_table3['continent']))
    ax.bar(X_axis - 0.2, pivot_table3['total_deaths'], 0.4, label = 'Total Deaths', color='blue')
    ax2.bar(X_axis + 0.2, pivot_table3[feature], 0.4, label = feature, color='red')

    ax.set_xticks(X_axis, pivot_table3['continent'], rotation=45)
    ax.set_xlabel('Continents')
    ax.set_ylabel('Total Deaths', color='blue')

    ax2.set_ylabel(feature, color='red')

    tick1, label1 = ax.get_legend_handles_labels()
    tick2, label2 = ax2.get_legend_handles_labels()
    tick = tick1 + tick2
    label = label1 + label2
    plt.legend(tick, label, loc='upper right')

    plt.tight_layout()
```

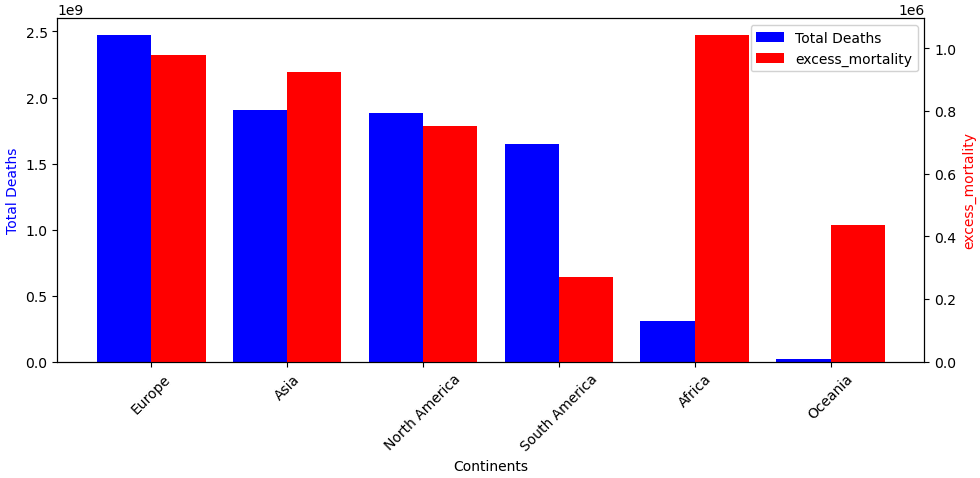
```
plt.show()
```

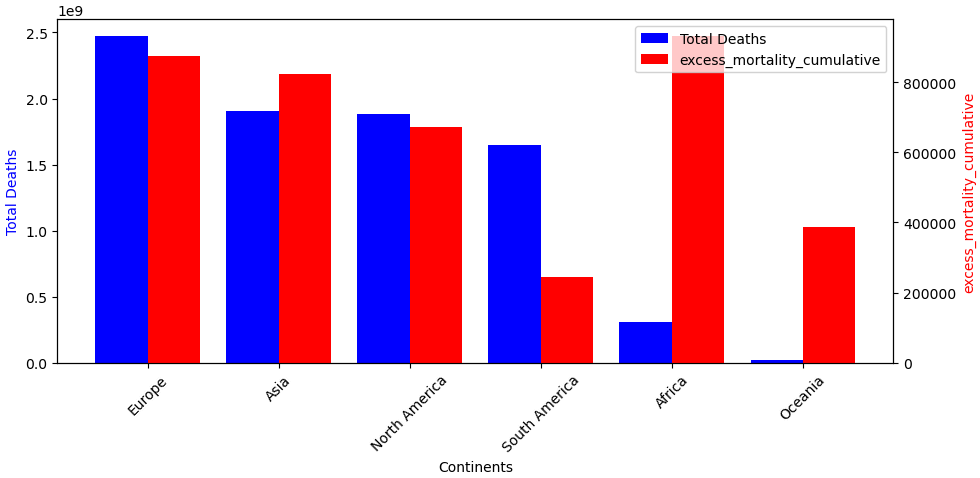

1e9
1e6
Total Deaths
aged_65_older
Total Deaths
aged_65_older
Europe
Asia
North America
South America
Africa
Oceania
Continents

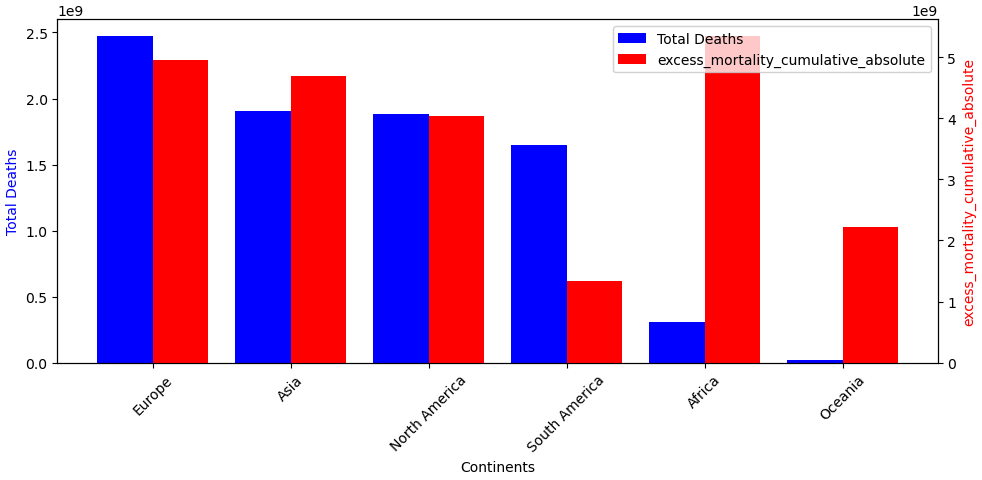

1e9
Total Deaths
aged_70_older
Total Deaths
aged_70_older
Europe
Asia
North America
South America
Africa
Oceania
Continents

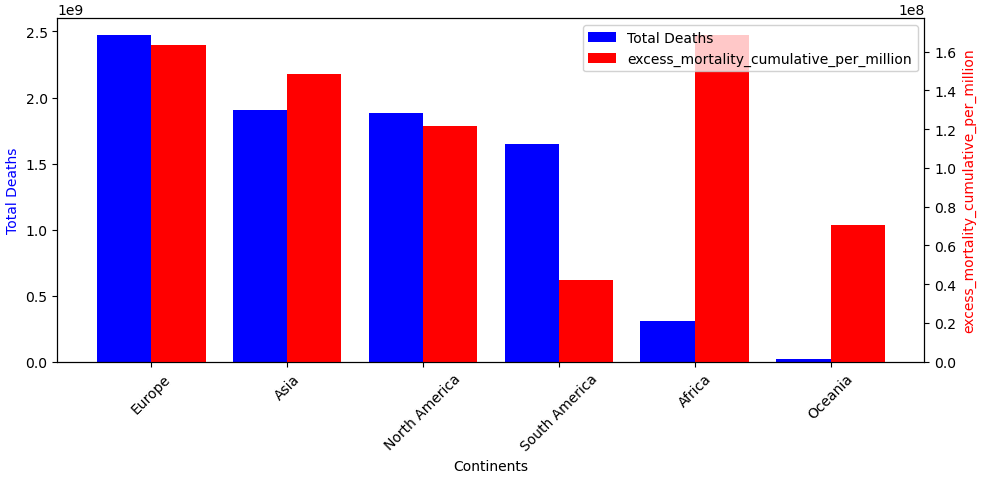

1e9
1e7
Total Deaths
cardiovasc_death_rate
Total Deaths
cardiovasc_death_rate
Europe
Asia
North America
South America
Africa
Oceania
Continents

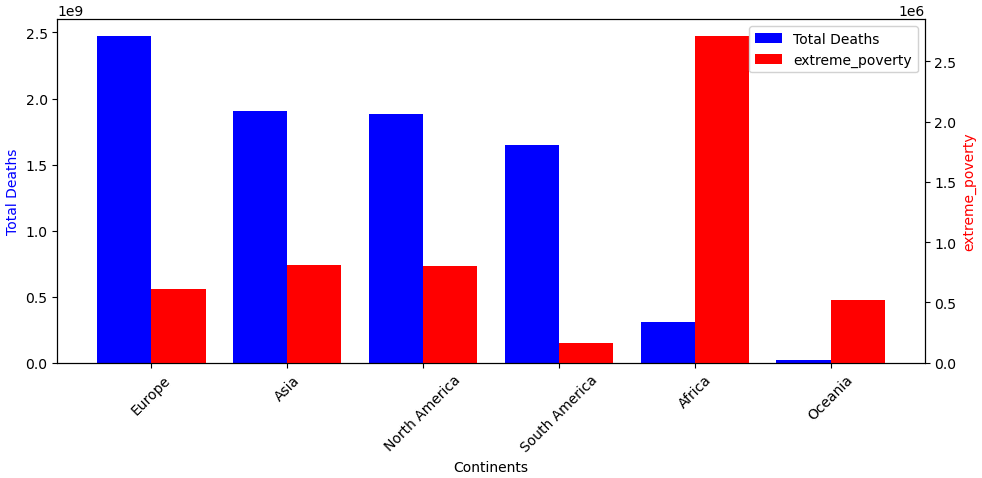

1e9
Total Deaths
diabetes_prevalence
Total Deaths
diabetes_prevalence
Europe
Asia
North America
South America
Africa
Oceania
Continents

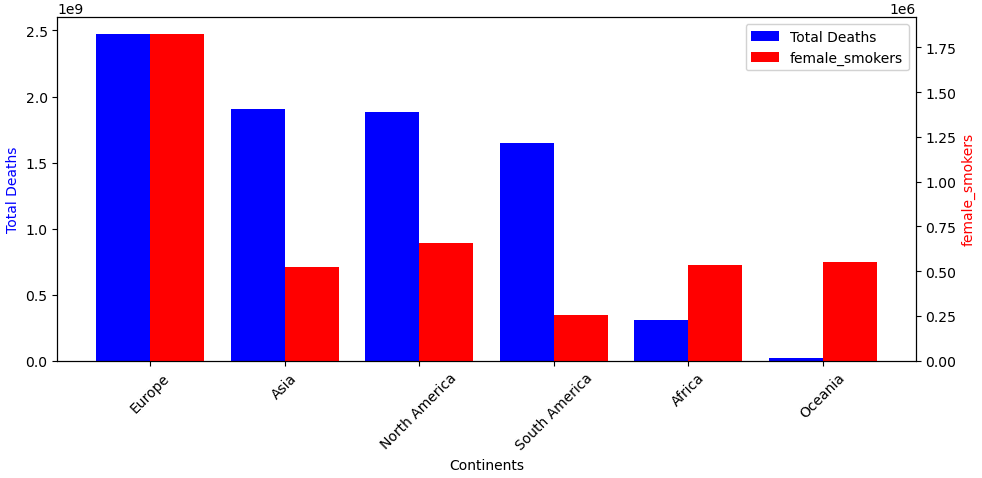

1e9
1e6
Total Deaths
excess_mortality
Total Deaths
excess_mortality
Europe
Asia
North America
South America
Africa
Oceania
Continents

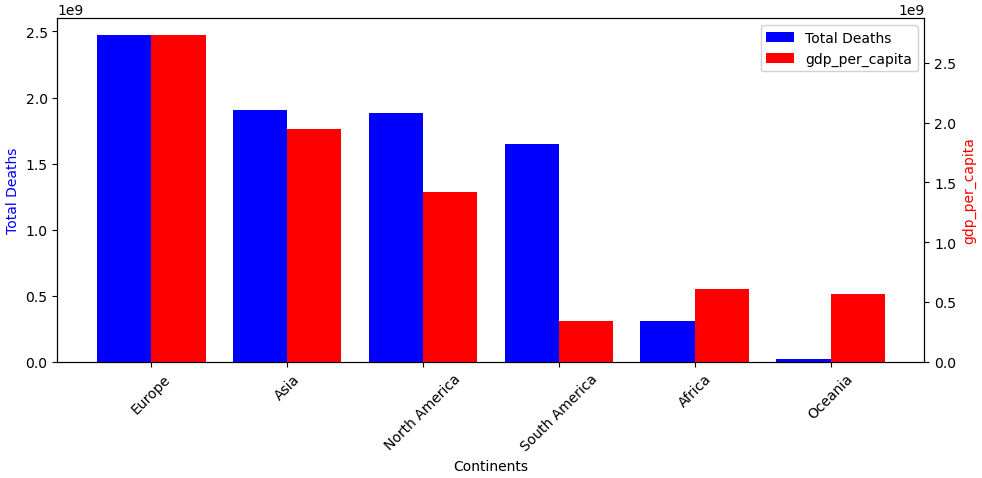

1e9
Total Deaths
excess_mortality_cumulative
Total Deaths
excess_mortality_cumulative
Europe
Asia
North America
South America
Africa
Oceania
Continents

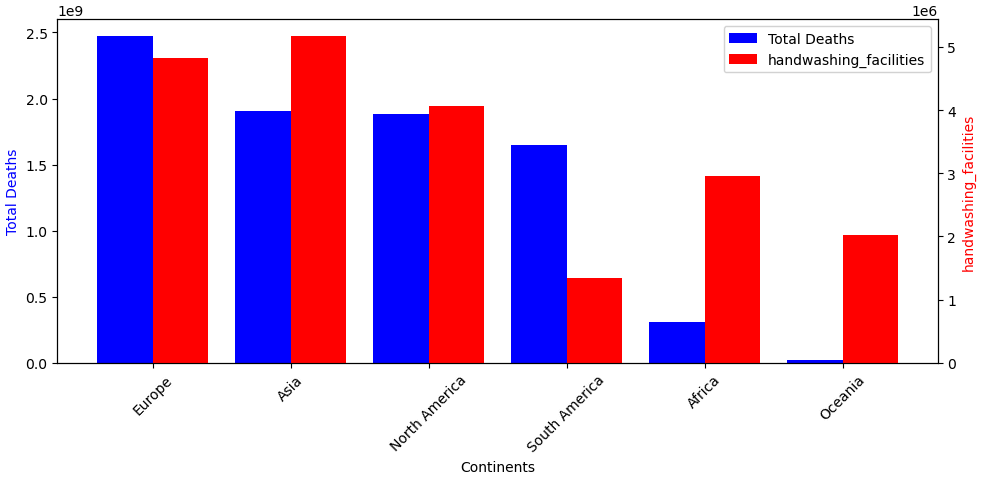

1e9
1e9
Total Deaths
excess_mortality_cumulative_absolute
Total Deaths
excess_mortality_cumulative_absolute
Europe
Asia
North America
South America
Africa
Oceania
Continents

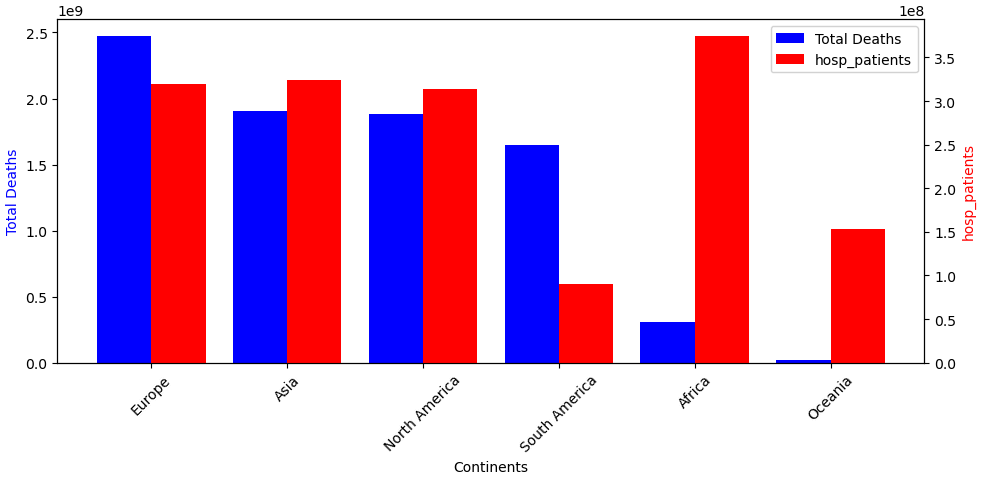

1e9
1e8
Total Deaths
excess_mortality_cumulative_per_million
Total Deaths
excess_mortality_cumulative_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

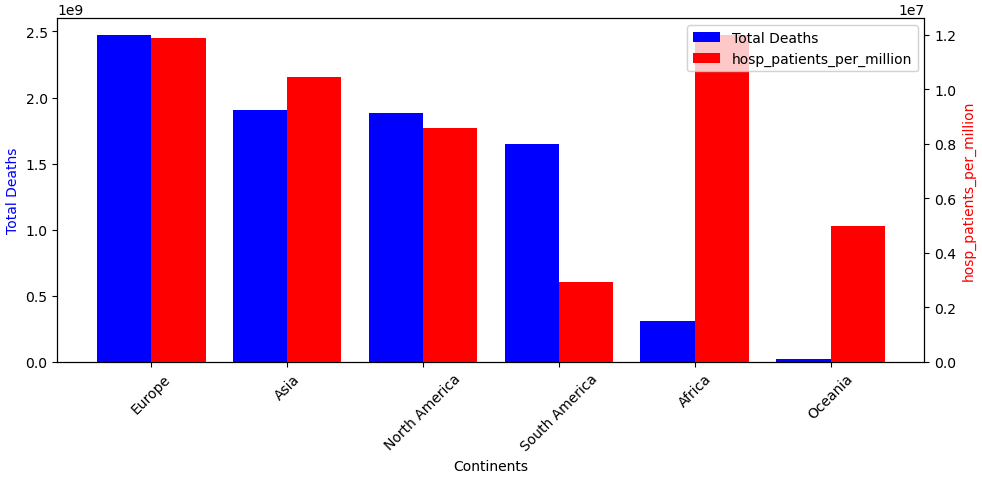

1e9
1e6
Total Deaths
extreme_poverty
Total Deaths
extreme_poverty
Europe
Asia
North America
South America
Africa
Oceania
Continents

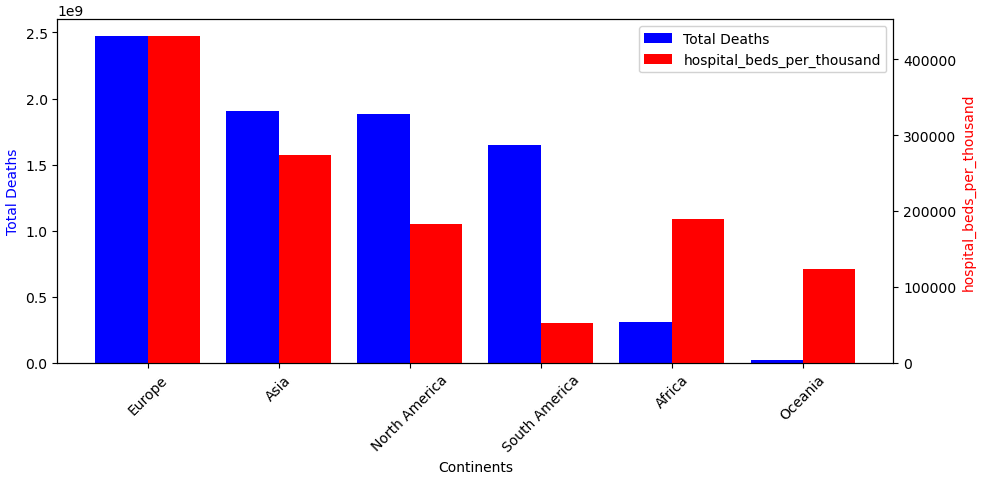

1e9
1e6
Total Deaths
female_smokers
Total Deaths
female_smokers
Europe
Asia
North America
South America
Africa
Oceania
Continents

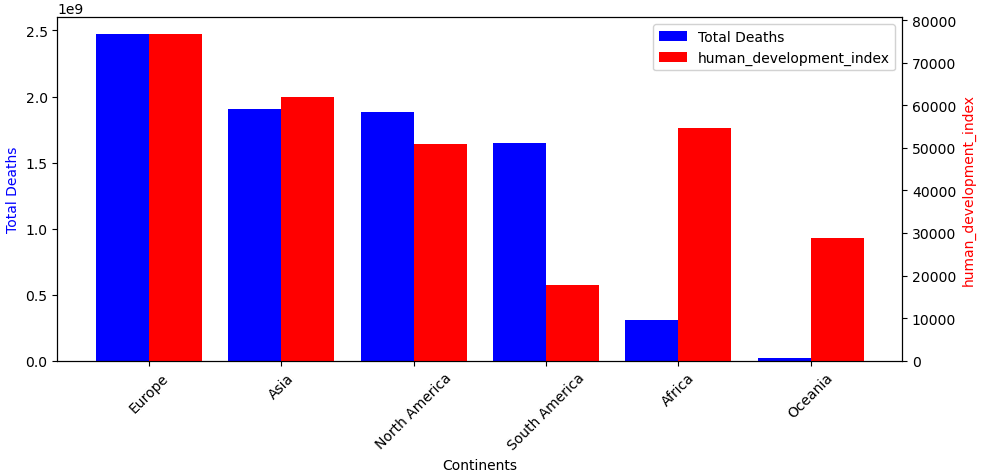

1e9
1e9
Total Deaths
gdp_per_capita
Total Deaths
gdp_per_capita
Europe
Asia
North America
South America
Africa
Oceania
Continents

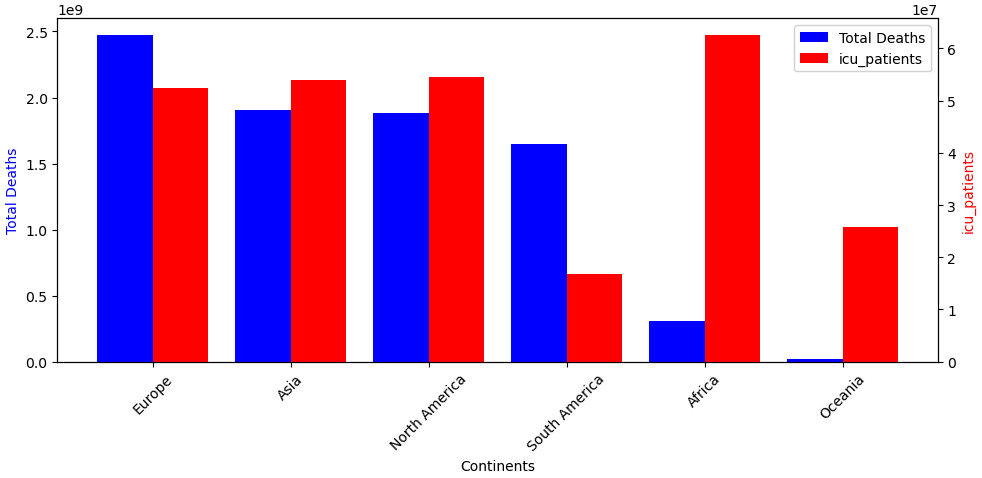

1e9
1e6
Total Deaths
handwashing_facilities
Total Deaths
handwashing_facilities
Europe
Asia
North America
South America
Africa
Oceania
Continents

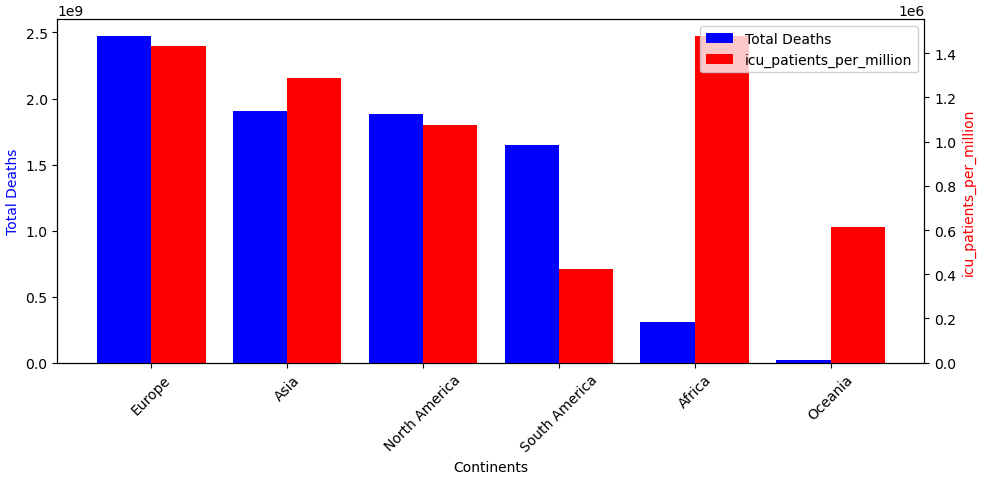

1e9
1e8
Total Deaths
hosp_patients
Total Deaths
hosp_patients
Europe
Asia
North America
South America
Africa
Oceania
Continents

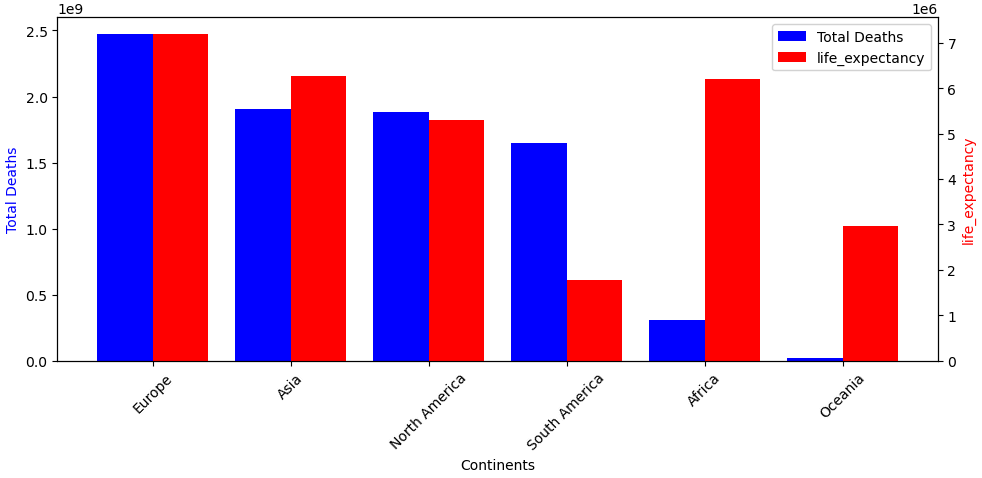

1e9
1e7
Total Deaths
hosp_patients_per_million
Total Deaths
hosp_patients_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

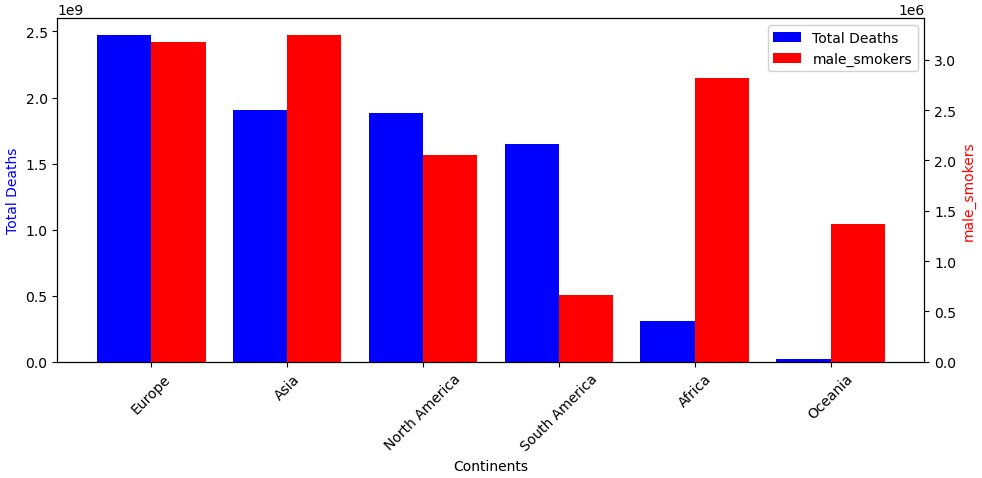

1e9
Total Deaths
hospital_beds_per_thousand
Total Deaths
hospital_beds_per_thousand
Europe
Asia
North America
South America
Africa
Oceania
Continents

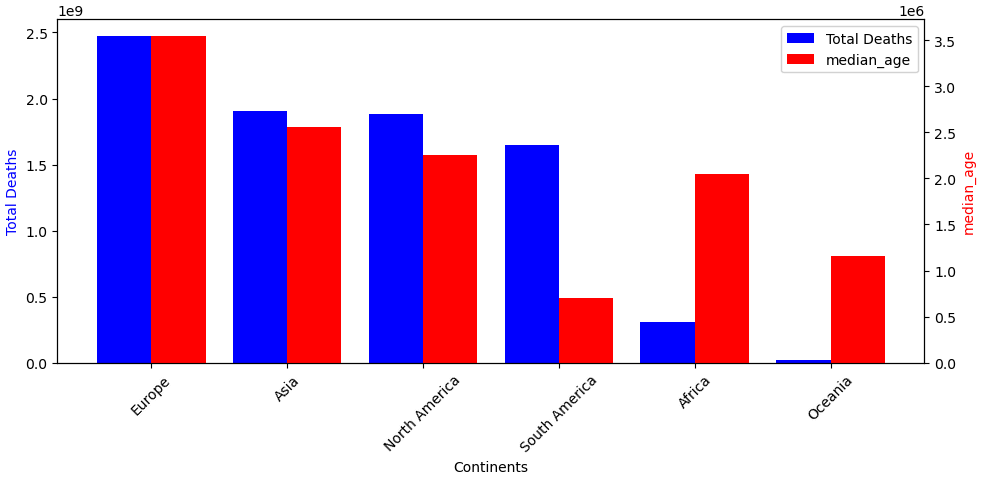

1e9
Total Deaths
human_development_index
Total Deaths
human_development_index
Europe
Asia
North America
South America
Africa
Oceania
Continents

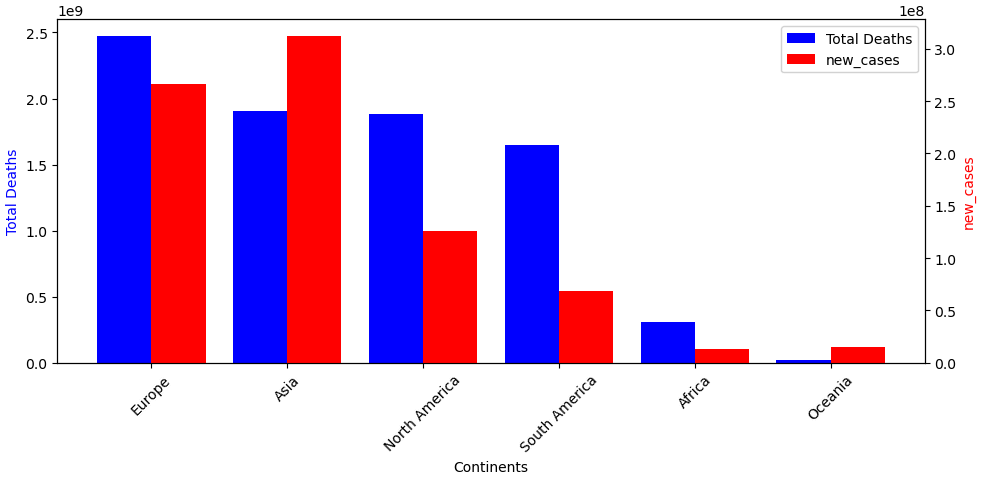

1e9
1e7
Total Deaths
icu_patients
Total Deaths
icu_patients
Europe
Asia
North America
South America
Africa
Oceania
Continents

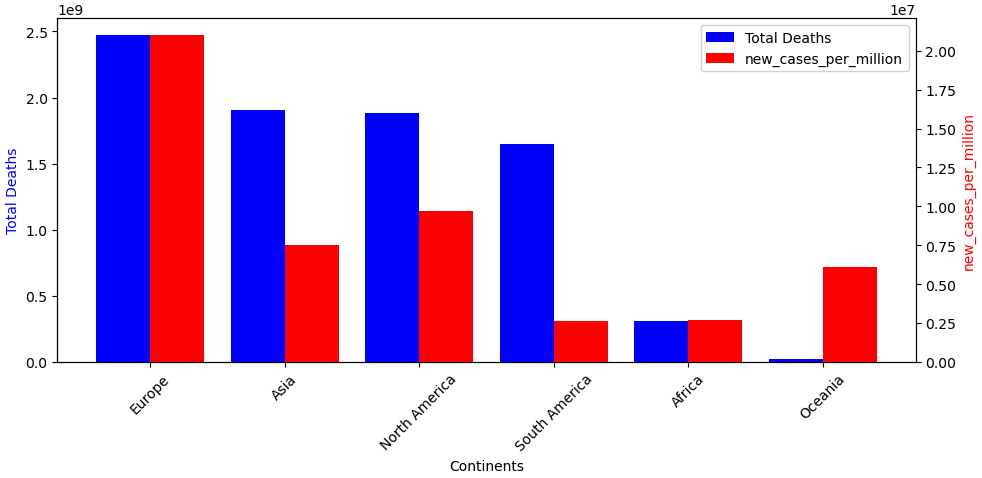

1e9
1e6
Total Deaths
icu_patients_per_million
Total Deaths
icu_patients_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

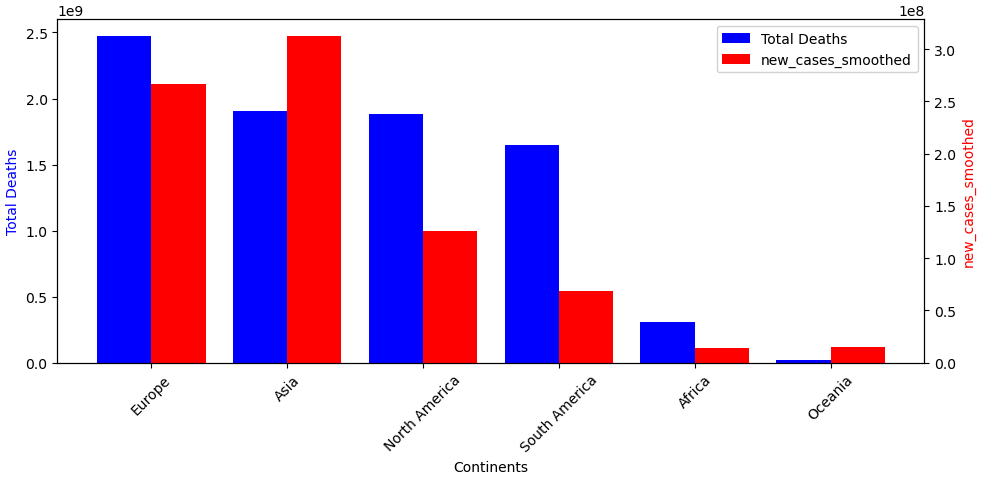

1e9
1e6
Total Deaths
life_expectancy
Total Deaths
life_expectancy
Europe
Asia
North America
South America
Africa
Oceania
Continents

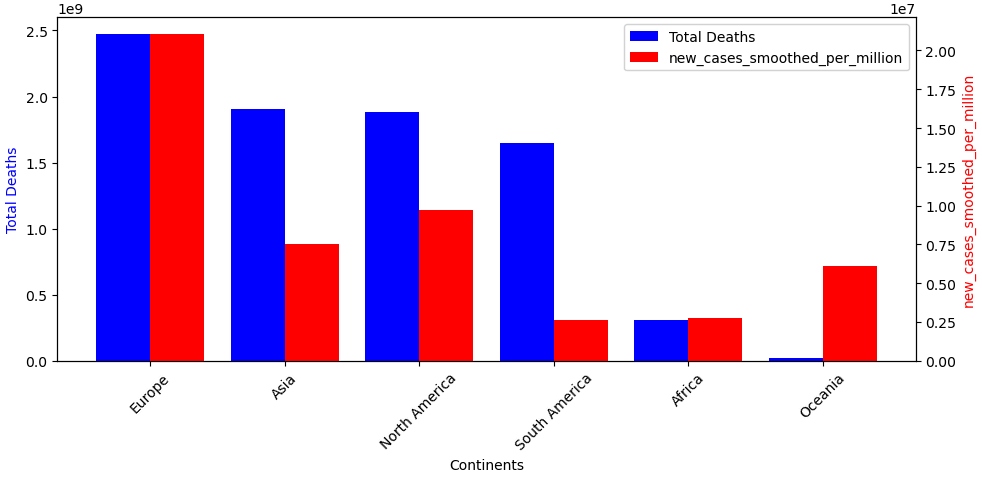

1e9
1e6
Total Deaths
male_smokers
Total Deaths
male_smokers
Europe
Asia
North America
South America
Africa
Oceania
Continents

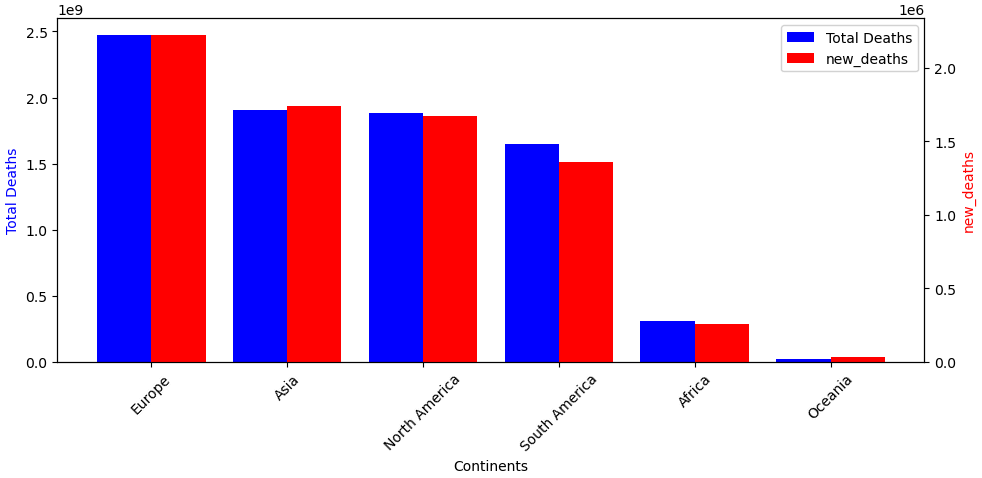

1e9
1e6
Total Deaths
median_age
Total Deaths
median_age
Europe
Asia
North America
South America
Africa
Oceania
Continents

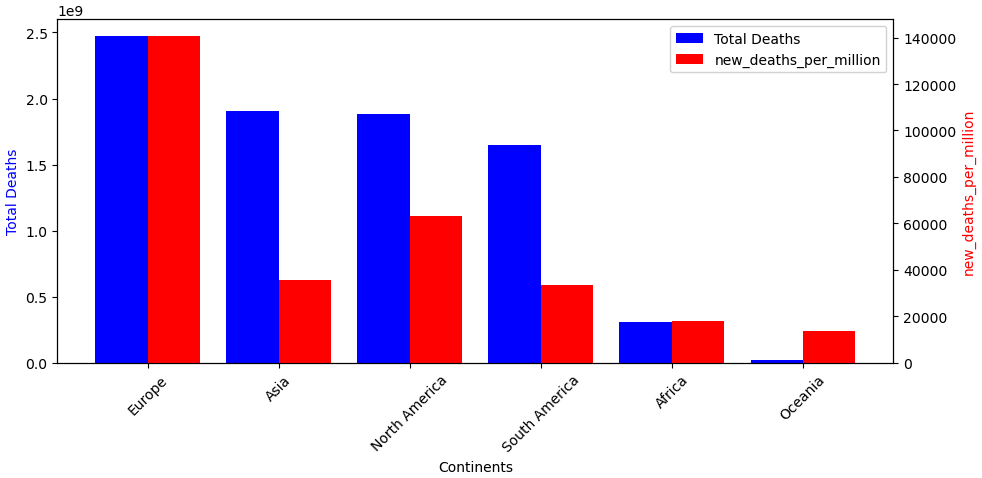

1e9
1e8
Total Deaths
new_cases
Total Deaths
new_cases
Europe
Asia
North America
South America
Africa
Oceania
Continents

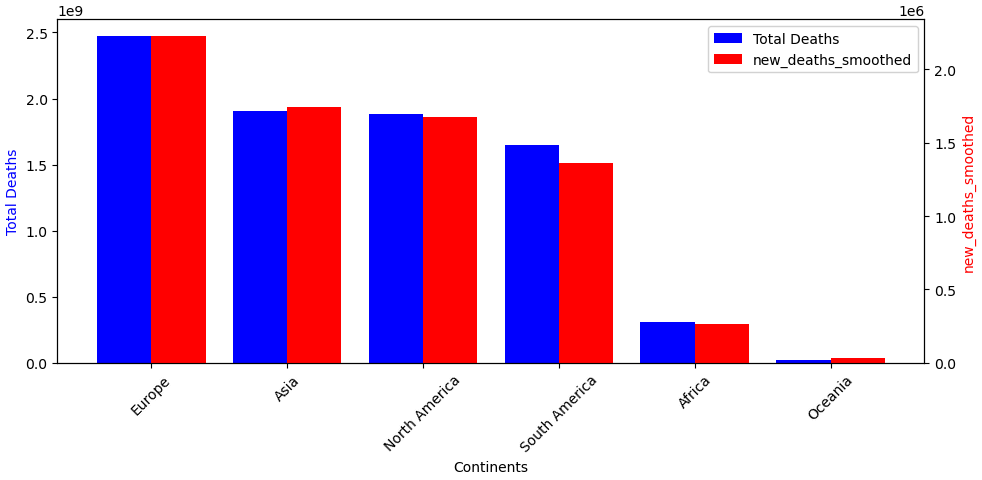

1e9
1e7
Total Deaths
new_cases_per_million
Total Deaths
new_cases_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

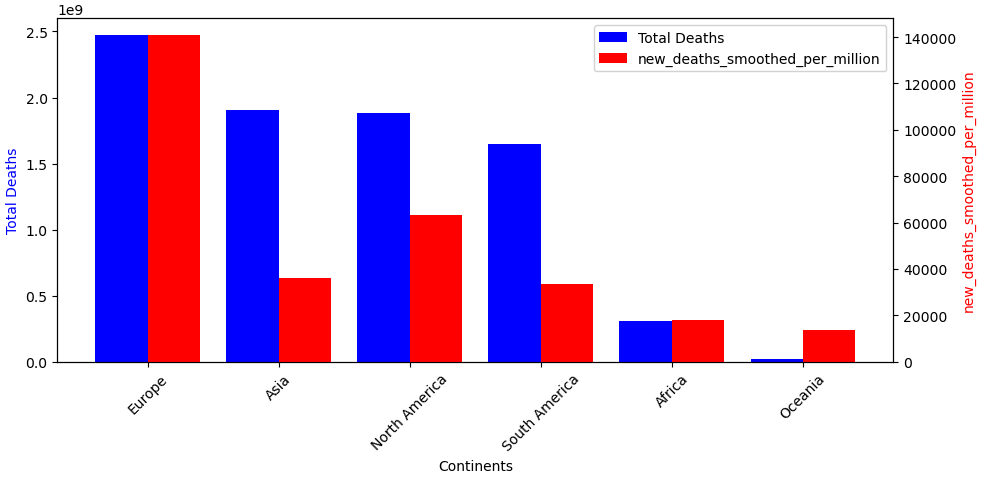

1e9
1e8
Total Deaths
new_cases_smoothed
Total Deaths
new_cases_smoothed
Europe
Asia
North America
South America
Africa
Oceania
Continents

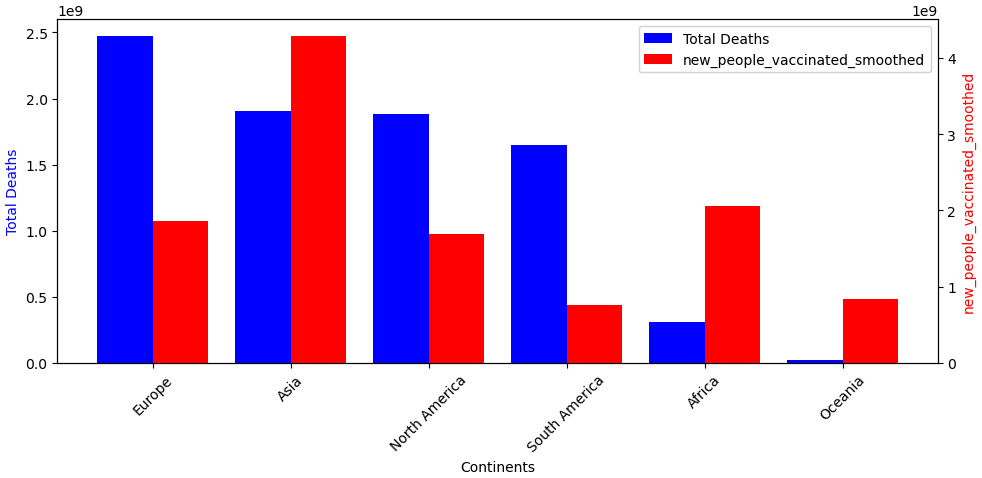

1e9
1e7
Total Deaths
new_cases_smoothed_per_million
Total Deaths
new_cases_smoothed_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

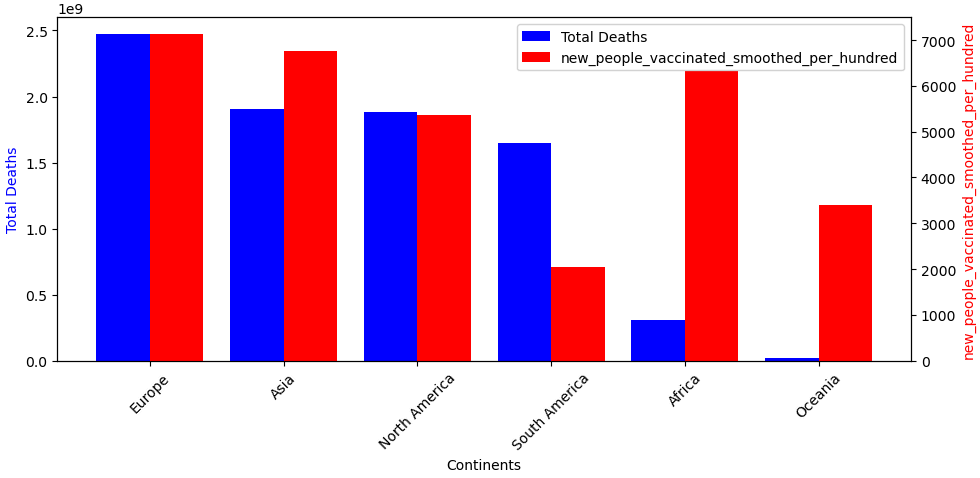

1e9
1e6
Total Deaths
new_deaths
Total Deaths
new_deaths
Europe
Asia
North America
South America
Africa
Oceania
Continents

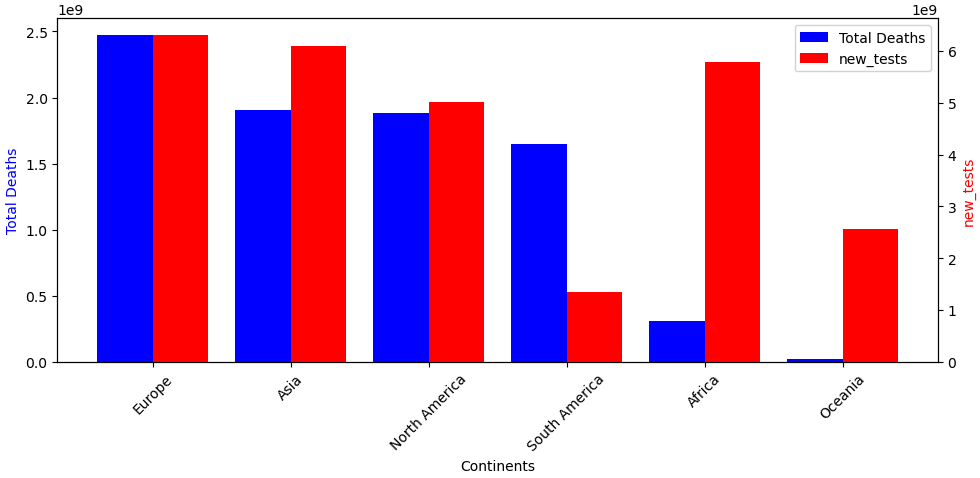

1e9
Total Deaths
new_deaths_per_million
Total Deaths
new_deaths_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

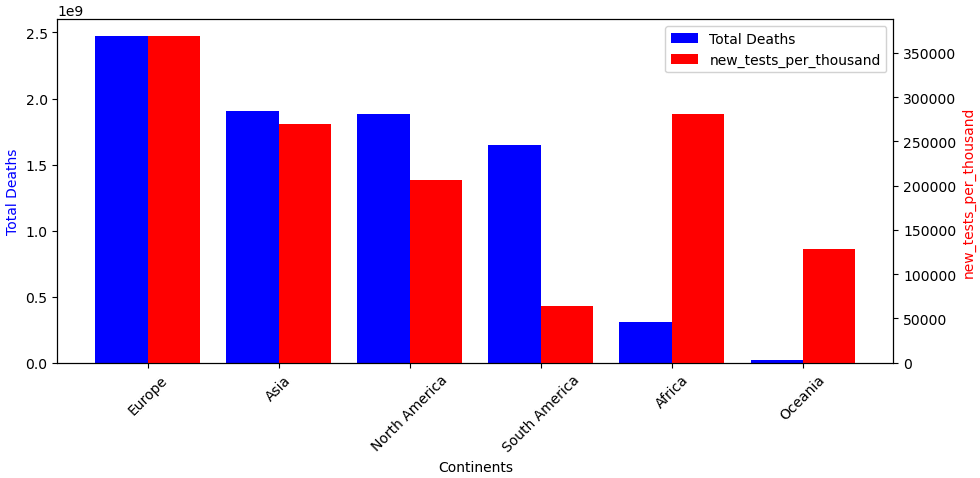

1e9
1e6
Total Deaths
new_deaths_smoothed
Total Deaths
new_deaths_smoothed
Europe
Asia
North America
South America
Africa
Oceania
Continents

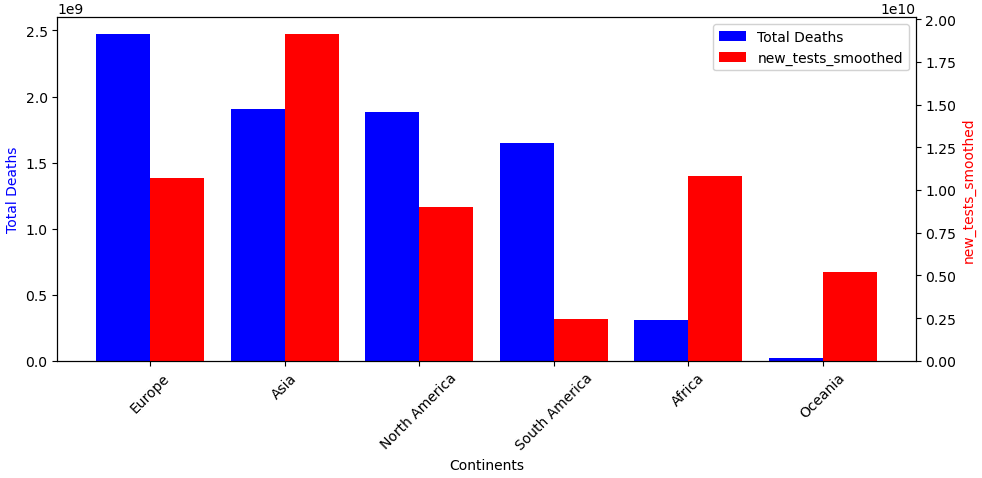

1e9
Total Deaths
new_deaths_smoothed_per_million
Total Deaths
new_deaths_smoothed_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

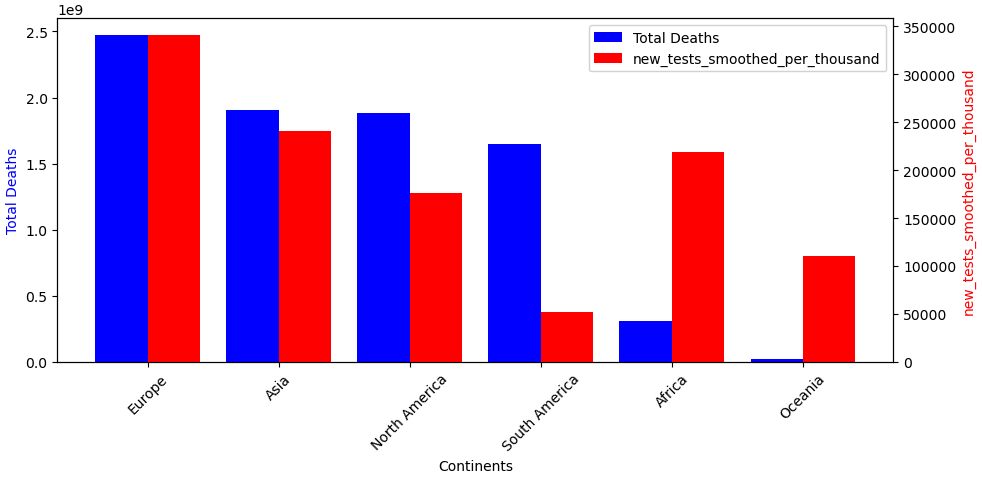

1e9
1e9
Total Deaths
new_people_vaccinated_smoothed
Total Deaths
new_people_vaccinated_smoothed
Europe
Asia
North America
South America
Africa
Oceania
Continents

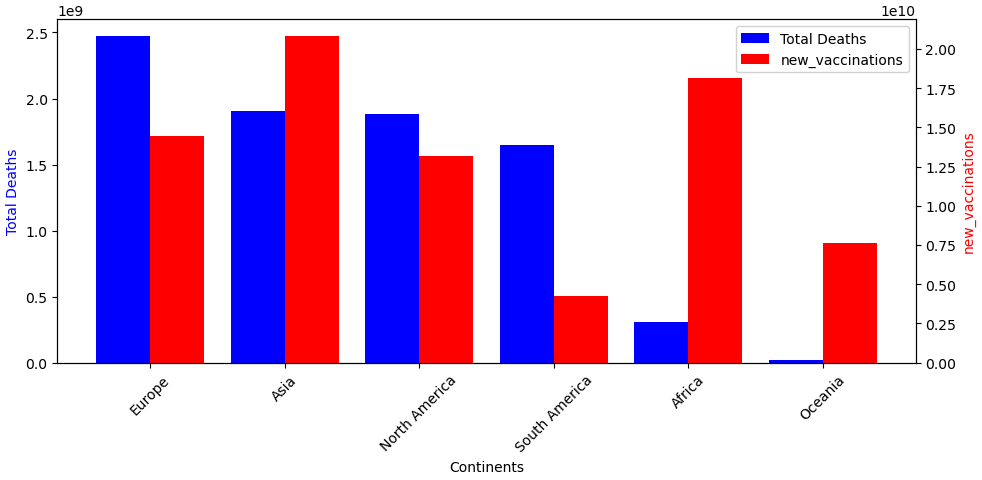

1e9
Total Deaths
new_people_vaccinated_smoothed_per_hundred
Total Deaths
new_people_vaccinated_smoothed_per_hundred
Europe
Asia
North America
South America
Africa
Oceania
Continents

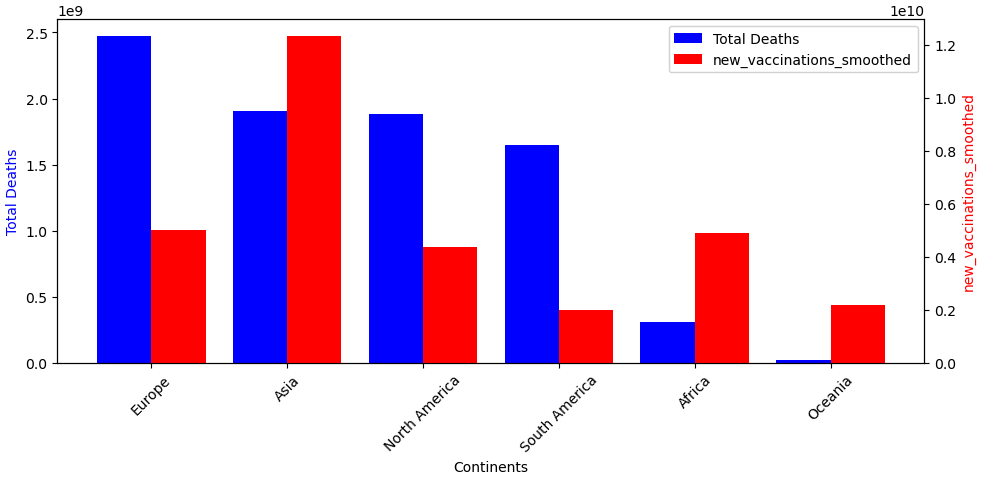

1e9
1e9
Total Deaths
new_tests
Total Deaths
new_tests
Europe
Asia
North America
South America
Africa
Oceania
Continents

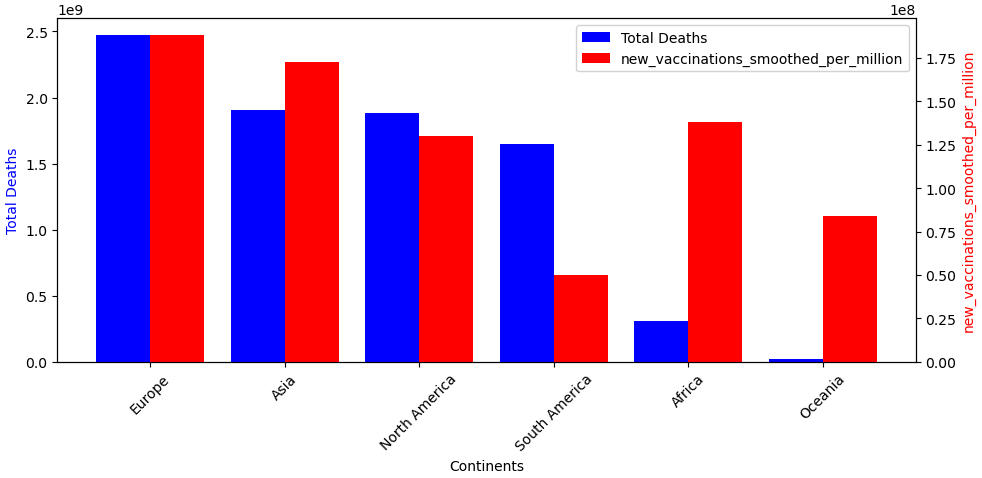

1e9
Total Deaths
new_tests_per_thousand
Total Deaths
new_tests_per_thousand
Europe
Asia
North America
South America
Africa
Oceania
Continents

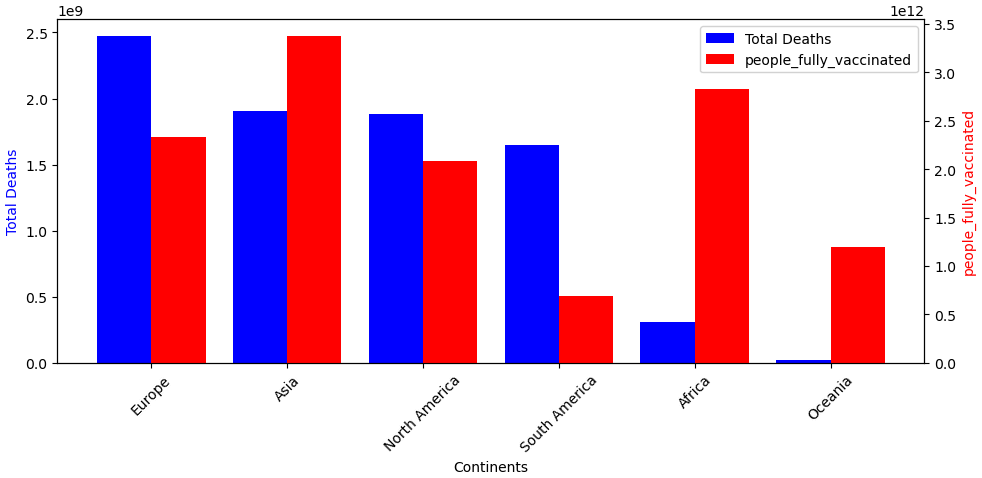

1e9
1e10
Total Deaths
new_tests_smoothed
Total Deaths
new_tests_smoothed
Europe
Asia
North America
South America
Africa
Oceania
Continents

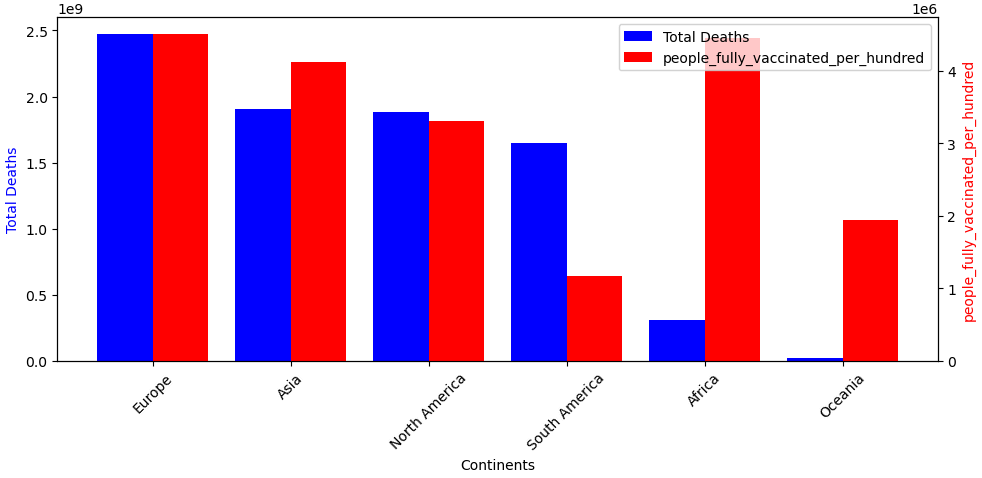

1e9
Total Deaths
new_tests_smoothed_per_thousand
Total Deaths
new_tests_smoothed_per_thousand
Europe
Asia
North America
South America
Africa
Oceania
Continents

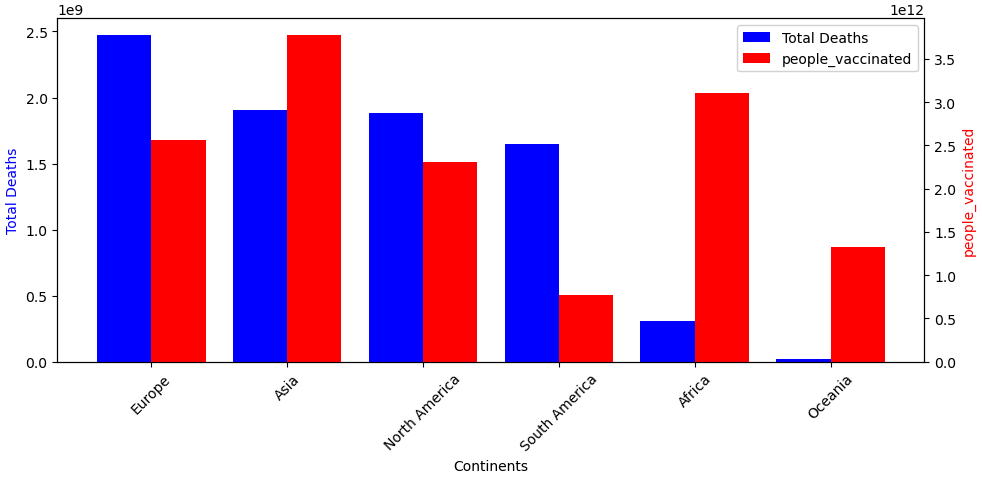

1e9
1e10
Total Deaths
new_vaccinations
Total Deaths
new_vaccinations
Europe
Asia
North America
South America
Africa
Oceania
Continents

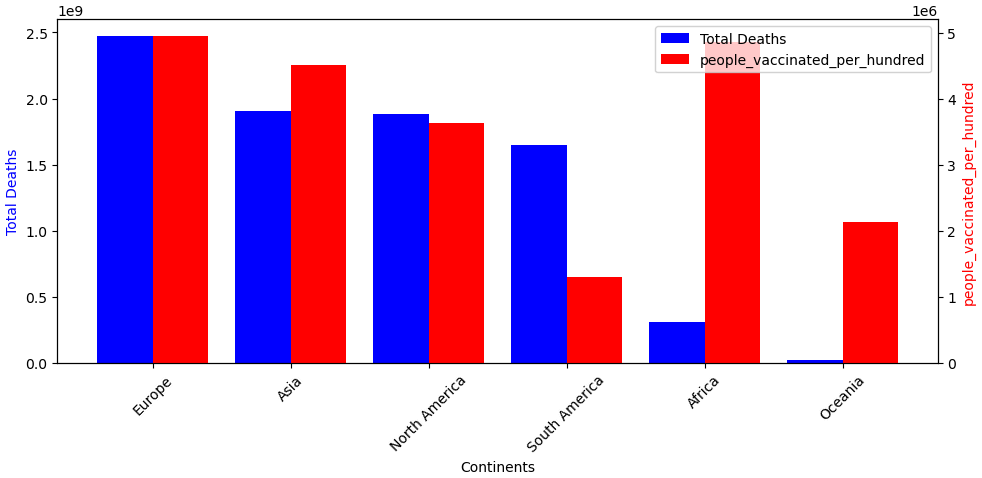

1e9
1e10
Total Deaths
new_vaccinations_smoothed
Total Deaths
new_vaccinations_smoothed
Europe
Asia
North America
South America
Africa
Oceania
Continents

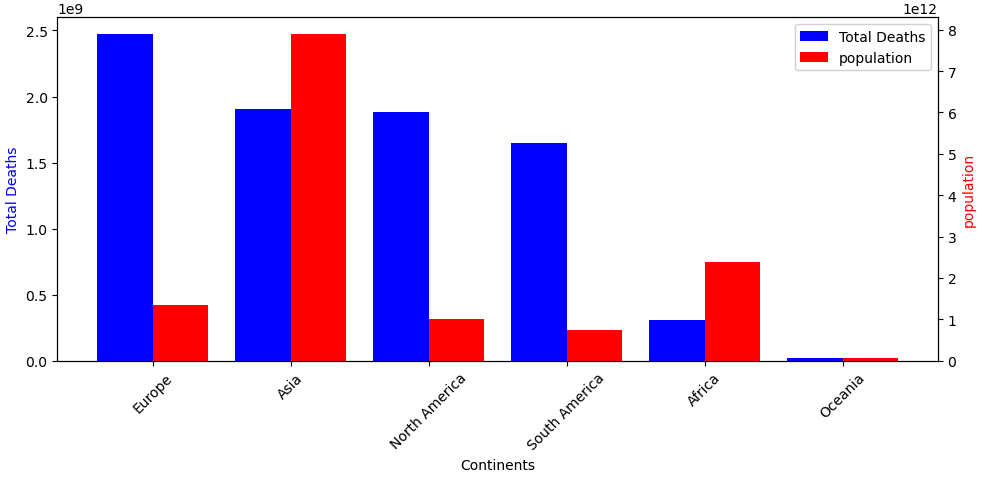

1e9
1e8
Total Deaths
new_vaccinations_smoothed_per_million
Total Deaths
new_vaccinations_smoothed_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

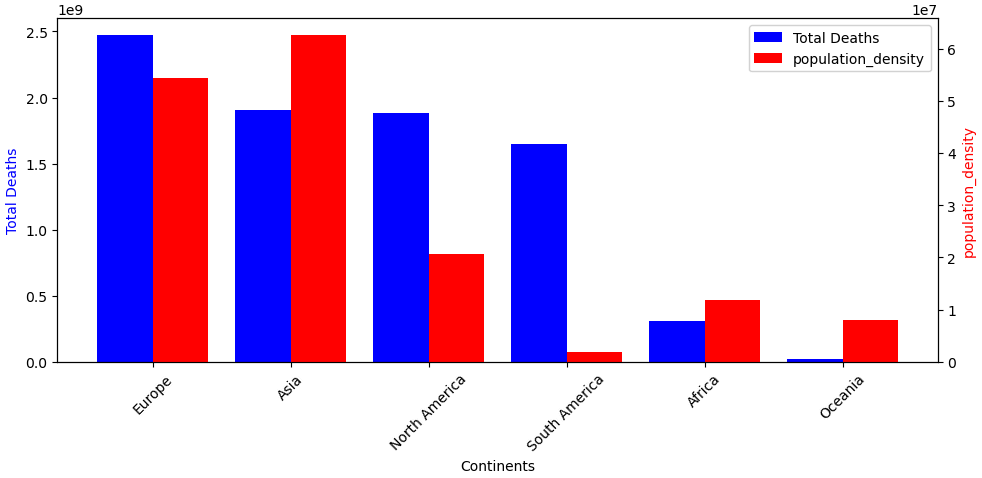

1e9
1e12
Total Deaths
people_fully_vaccinated
Total Deaths
people_fully_vaccinated
Europe
Asia
North America
South America
Africa
Oceania
Continents

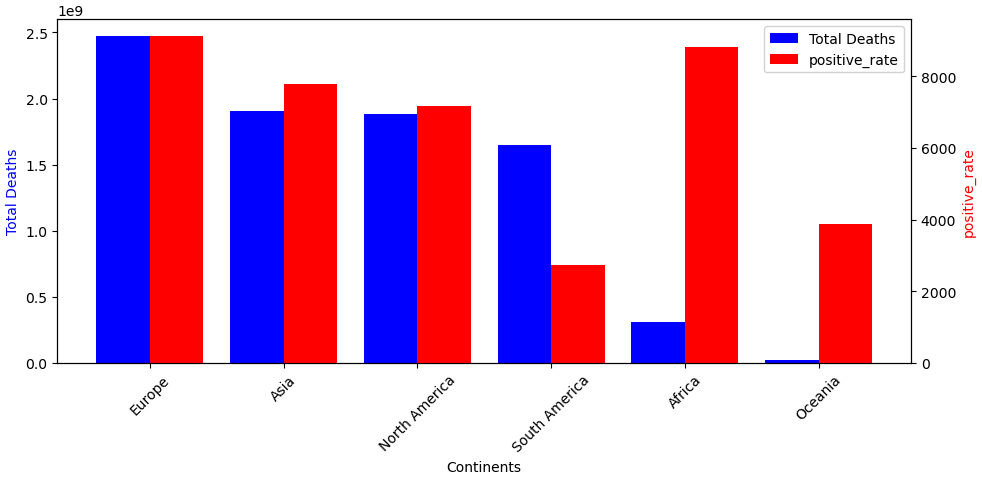

1e9
1e6
Total Deaths
people_fully_vaccinated_per_hundred
Total Deaths
people_fully_vaccinated_per_hundred
Europe
Asia
North America
South America
Africa
Oceania
Continents

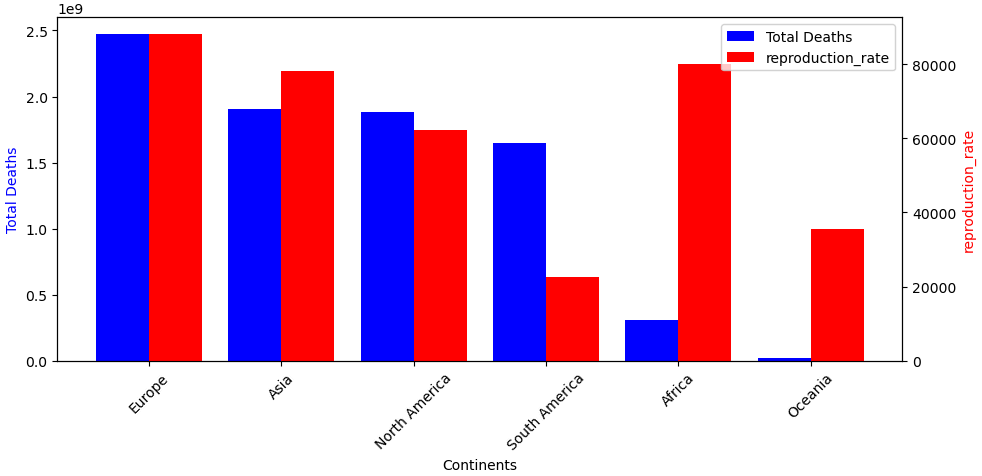

1e9
1e12
Total Deaths
people_vaccinated
Total Deaths
people_vaccinated
Europe
Asia
North America
South America
Africa
Oceania
Continents

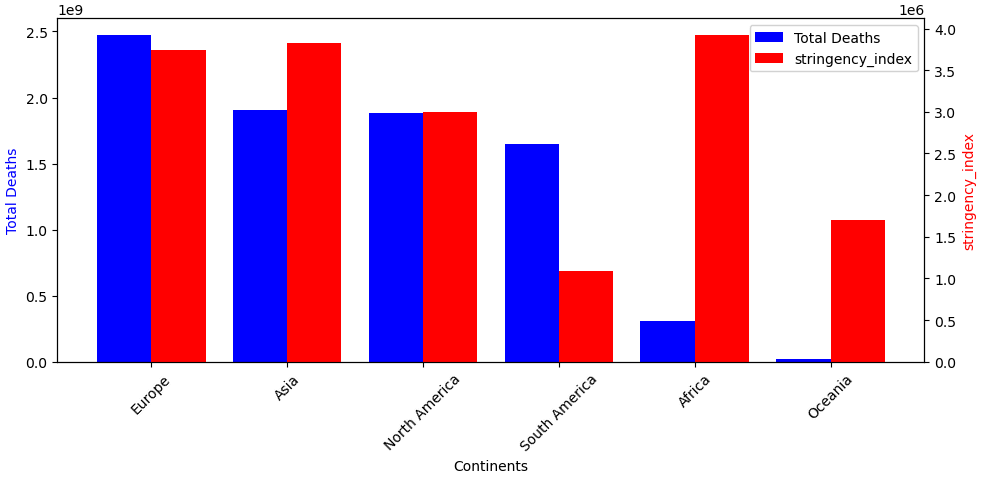

1e9
1e6
Total Deaths
people_vaccinated_per_hundred
Total Deaths
people_vaccinated_per_hundred
Europe
Asia
North America
South America
Africa
Oceania
Continents

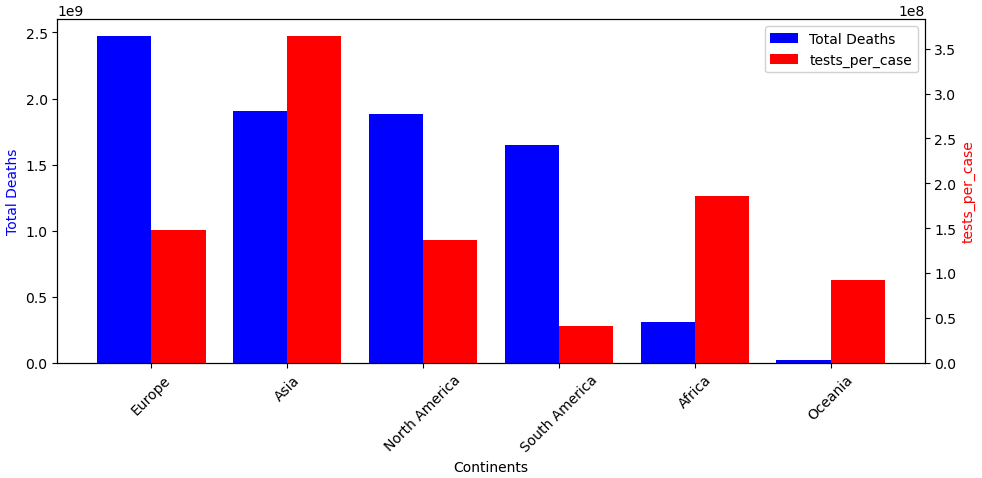

1e9
1e12
Total Deaths
population
Total Deaths
population
Europe
Asia
North America
South America
Africa
Oceania
Continents

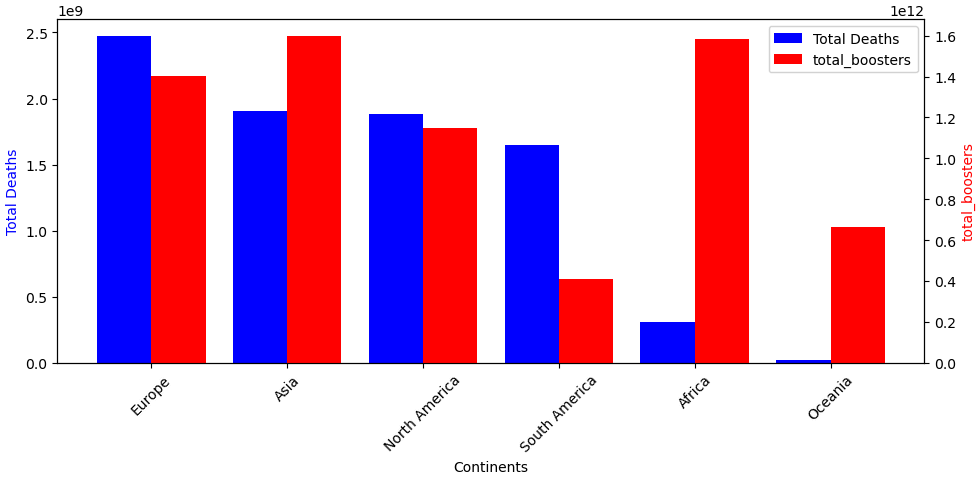

1e9
1e7
Total Deaths
population_density
Total Deaths
population_density
Europe
Asia
North America
South America
Africa
Oceania
Continents

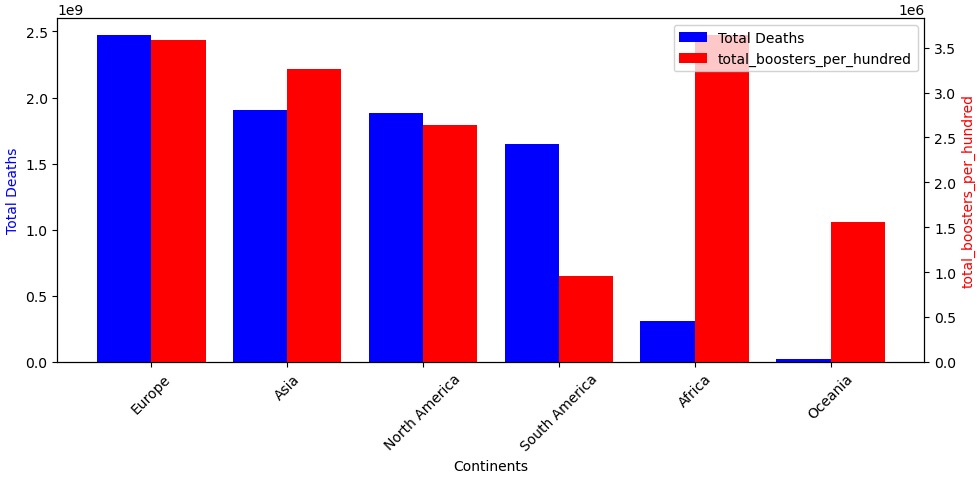

1e9
Total Deaths
positive_rate
Total Deaths
positive_rate
Europe
Asia
North America
South America
Africa
Oceania
Continents

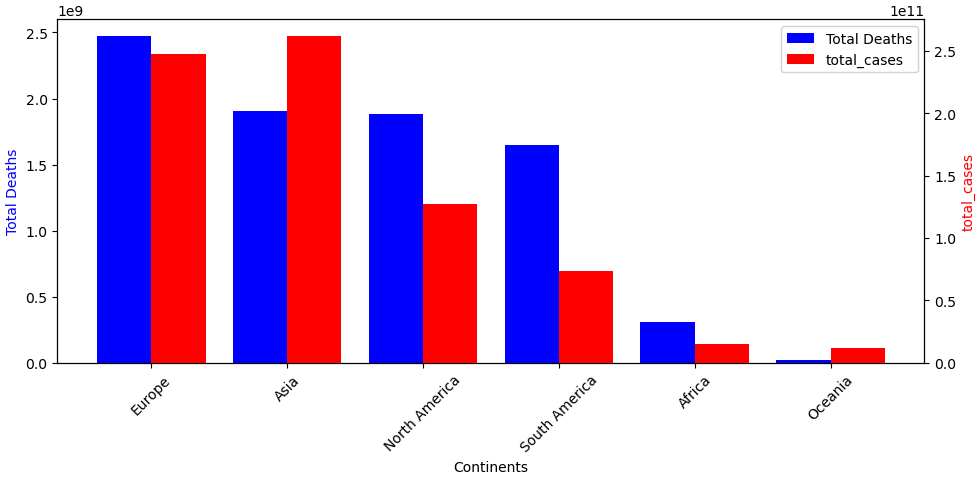

1e9
Total Deaths
reproduction_rate
Total Deaths
reproduction_rate
Europe
Asia
North America
South America
Africa
Oceania
Continents

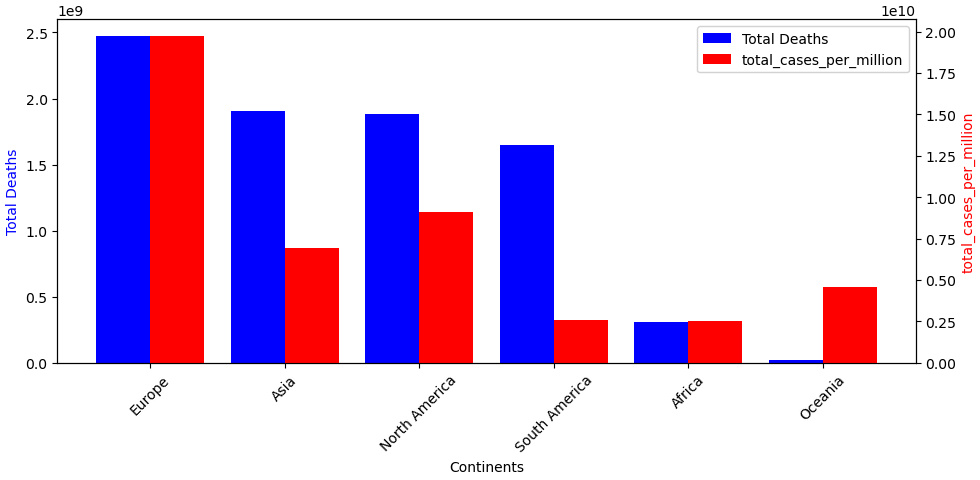

1e9
1e6
Total Deaths
stringency_index
Total Deaths
stringency_index
Europe
Asia
North America
South America
Africa
Oceania
Continents

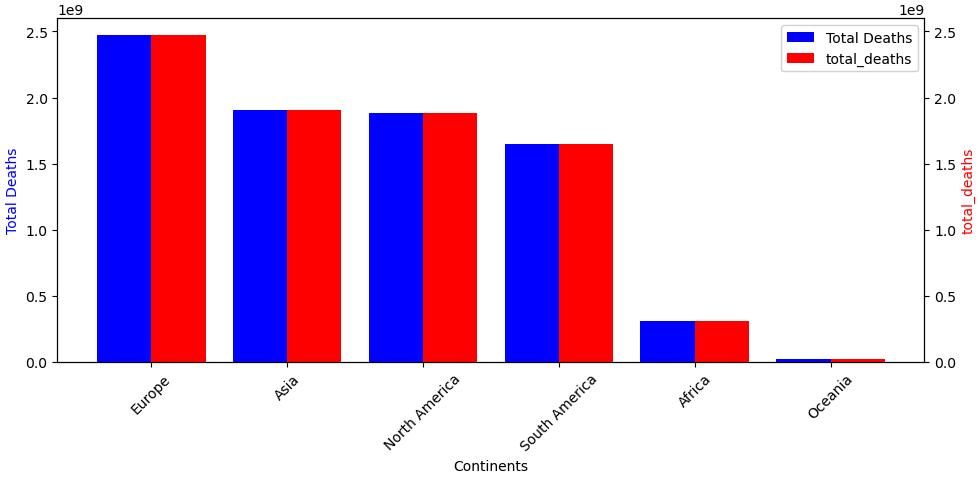

1e9
1e8
Total Deaths
tests_per_case
Total Deaths
tests_per_case
Europe
Asia
North America
South America
Africa
Oceania
Continents

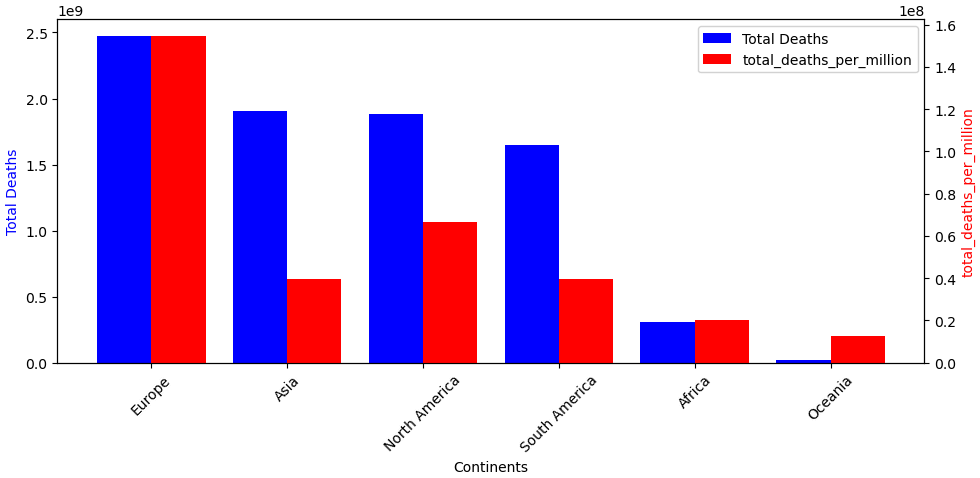

1e9
1e12
Total Deaths
total_boosters
Europe
Asia
North America
South America
Africa
Oceania
Continents

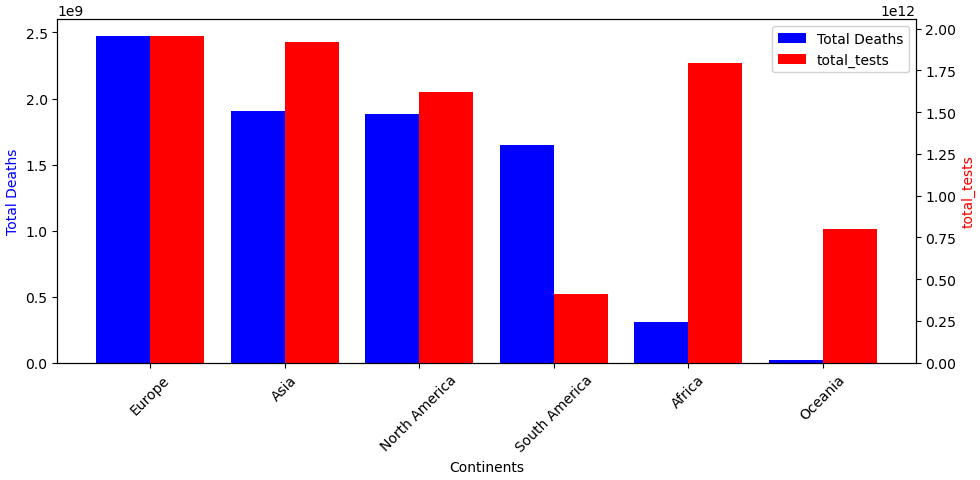

1e9
1e6
Total Deaths
total_boosters_per_hundred
Europe
Asia
North America
South America
Africa
Oceania
Continents

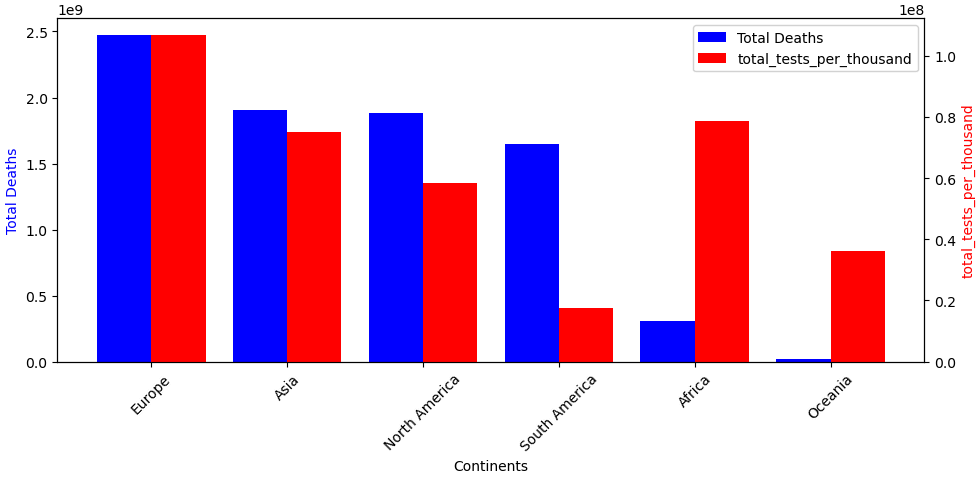

1e9
1e11
Total Deaths
total_cases
Europe
Asia
North America
South America
Africa
Oceania
Continents

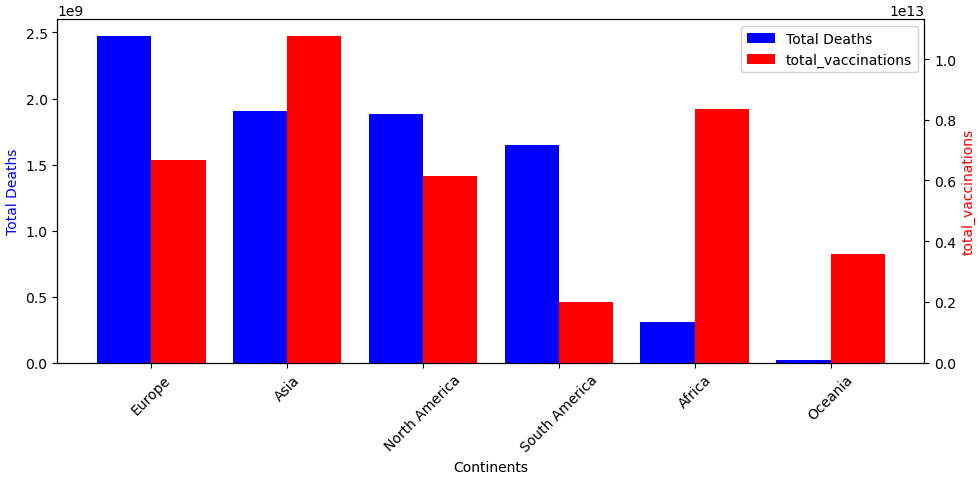

1e9
1e10
Total Deaths
total_cases_per_million
Total Deaths
total_cases_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

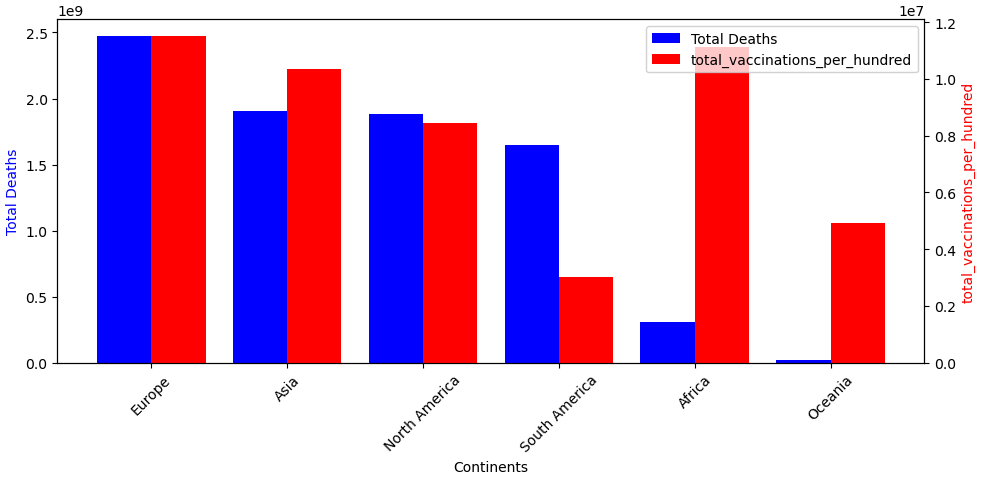

1e9
1e9
Total Deaths
total_deaths
Total Deaths
total_deaths
Europe
Asia
North America
South America
Africa
Oceania
Continents

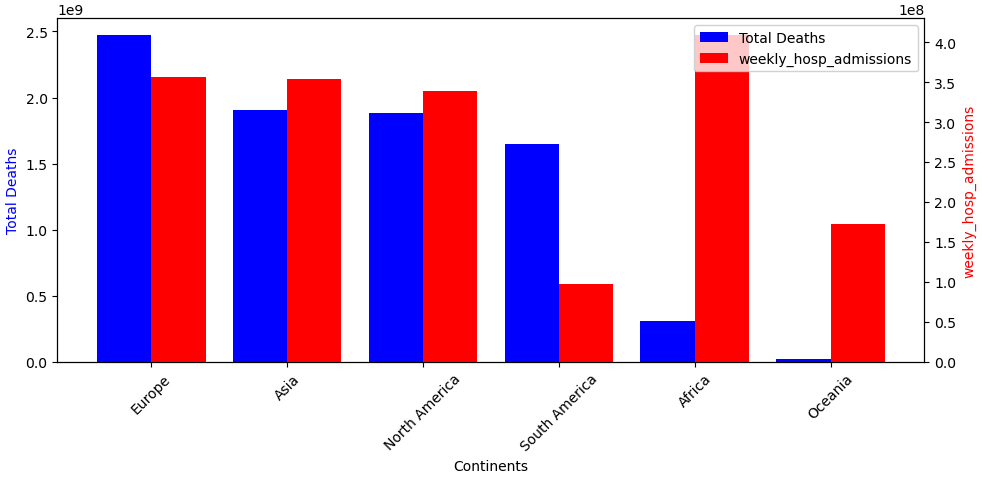

1e9
1e8
Total Deaths
total_deaths_per_million
Total Deaths
total_deaths_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents

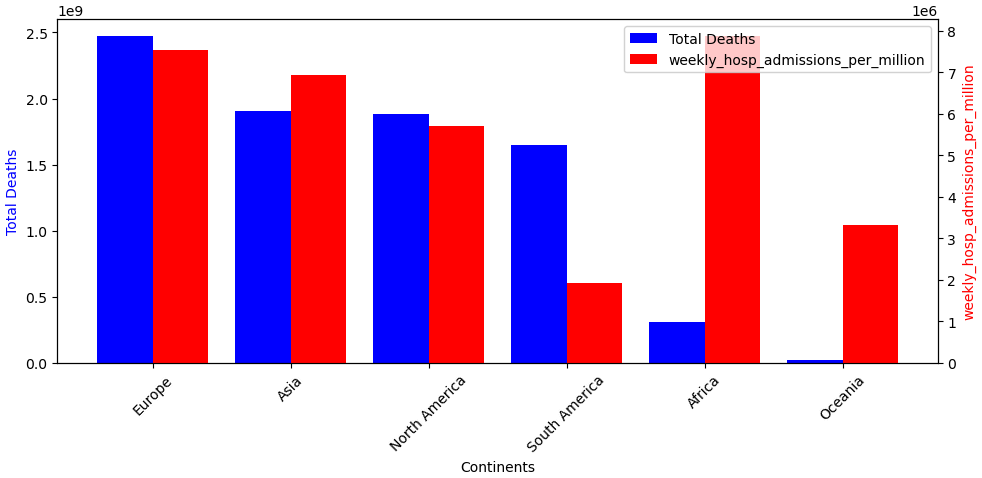

1e9
1e12
Total Deaths
total_tests
Total Deaths
total_tests
Europe
Asia
North America
South America
Africa
Oceania
Continents

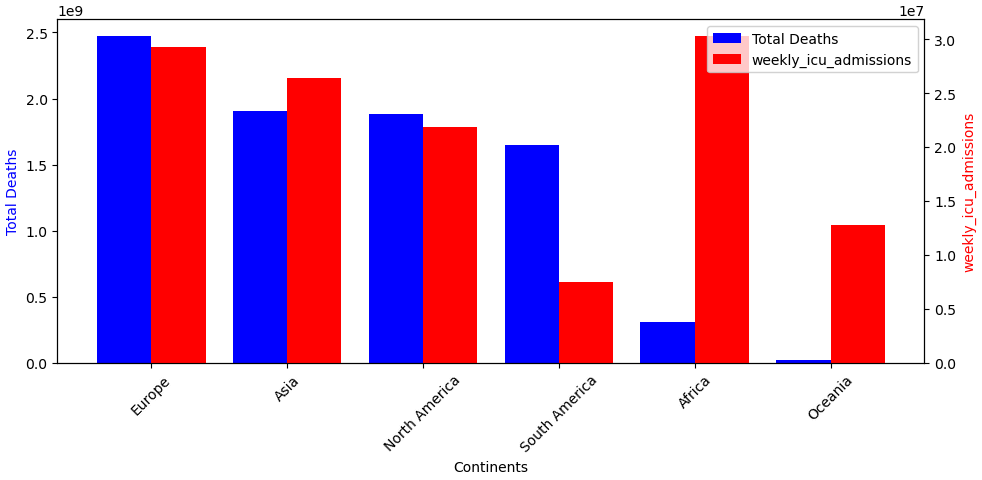

1e9
1e8
Total Deaths
total_tests_per_thousand
Total Deaths
total_tests_per_thousand
Europe
Asia
North America
South America
Africa
Oceania
Continents

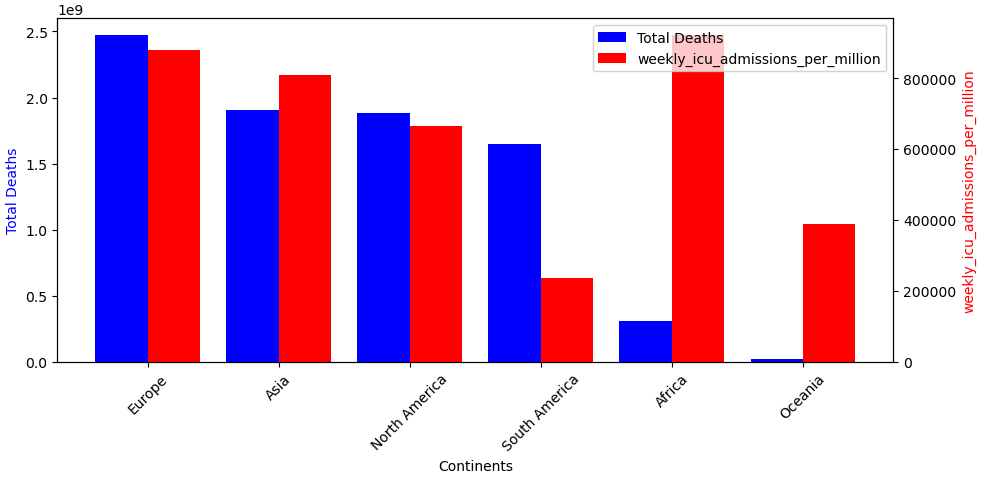

1e9
1e13
Total Deaths
total_vaccinations
Total Deaths
total_vaccinations
Europe
Asia
North America
South America
Africa
Oceania
Continents


1e9
1e7
Total Deaths
total_vaccinations_per_hundred
Total Deaths
total_vaccinations_per_hundred
Europe
Asia
North America
South America
Africa
Oceania
Continents


1e9
1e8
Total Deaths
weekly_hosp_admissions
Total Deaths
weekly_hosp_admissions
Europe
Asia
North America
South America
Africa
Oceania
Continents


1e9
1e6
Total Deaths
weekly_hosp_admissions_per_million
Total Deaths
weekly_hosp_admissions_per_million
Europe
Asia
North America
South America
Africa
Oceania
Continents





DEDICATION: For the Lord giveth wisdom: out of his mouth cometh knowledge and understanding.

ANALYST: oladoyinbobabatundematthew@gmail.com