EASTERN UNIVERSITY

COMPLETE PROJECT

Our project's roadmap from beginning to end is contained in this document. Our data science master's program's capstone project, DTSC-961, uses data from the World Health Organization (WHO) and kaggle.com to analyze COVID-19 statistics from January 22 to July 27, 2020. We used Jupyter notebooks to perform extensive data cleaning and manipulation in Python, and PostgreSQL with PGAdmin4 was used for more complex manipulation. Tableau was used for visualization.

I. DATA MANIPULATION WITH PYTHON: JUPYTER NOTEBOOK

II. DATABASE MANIPULATION AND QUERIES: POSTGRESQL/PGADMIN

III. DATA VISUALISATION : TABLEAU

Kerryl MUSUNGU AJ

DTSC 691 PROJECT

COVID-19 Dataset: Number of Confirmed, Death and Recovered cases every day across the globe from Kaggle.com

Project Overview and Goal

We will be analyzing the Covid-19 datasets for this project and highlighting key details that are pertinent to our study topic. The project aims to monitor several critical variables over a period of time, from January 22, 2020, to July 27, 2020, including the number of cases, the mortality rate, and the recovery rate by geographic location. Instead of discussing the causes of these effects, this project will concentrate on generic research of the COVID-19's effects in various World Health Organization (WHO) regions. We will list the several questions we are attempting to address in this project for greater clarity:

- How the COVID-19 pandemic has changed over time in each geographical area (Coutry, continent)
- The number of cases of COVID-19 per geographical area (country, continent).
- What is the population's overall death rate by geographic area (country, continent)?
- The number of individuals who returned home after recovering from the COvid-19 in each country, continent?
- Assessing the efficacy of treatment modalities by geographical area (nation, continent)
- Analyzing the differences in the efficacy of COVID-19 treatment between the US and other nations
- Where was the COVID-19 pandemic least disruptive? (Nationality, continent)
- Where was the COVID-19 epidemic most prevalent? (nation, Continent)
- And more

You may view our stories in our tableau file, which attempts to address each of the aforementioned questions. We believe that after reading this, you will have a clear understanding of our goals and a solid overview of the project.

We are going to examine several COVID-19 datasets in this notebook. We will examine each dataset independently to analyze any missing values and ensure that the appropriate datatype is given to the appropriate column. We will also remove columns that mostly contain missing values or have worse quality information. Once this procedure is finished, we will analyze which join method is most accurate to combine a table or data set to get the necessary information. And how might we go about building a relational schema?

In [1]:
```
import numpy as np
import pandas as pd
```

1. country_wise_latest

In [2]:
```
df1 = pd.read_csv('Datasets/country_wise_latest.csv')
df1.head()
```

Out[2]:

	Country/Region	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deat / 1 Cas
0	Afghanistan	36263	1269	25198	9796	106	10	18	3.
1	Albania	4880	144	2745	1991	117	6	63	2.
2	Algeria	27973	1163	18837	7973	616	8	749	4.
3	Andorra	907	52	803	52	10	0	0	5.
4	Angola	950	41	242	667	18	1	0	4.

In [3]:
```
list(df1.columns)
```

Out[3]:
```
['Country/Region',
 'Confirmed',
 'Deaths',
 'Recovered',
 'Active',
 'New cases',
 'New deaths',
 'New recovered',
 'Deaths / 100 Cases',
 'Recovered / 100 Cases',
 'Deaths / 100 Recovered',
 'Confirmed last week',
 '1 week change',
 '1 week % increase',
 'WHO Region']
```

In [4]:
```
#df1.describe()
df1.shape
```

Out[4]:
```
(187, 15)
```

In [5]:
```
df1.isnull().sum() #Always checking for missing values. We do not have missing valu
```

Out[5]:

```
Country/Region           0
Confirmed                0
Deaths                   0
Recovered                0
Active                   0
New cases                0
New deaths               0
New recovered            0
Deaths / 100 Cases       0
Recovered / 100 Cases    0
Deaths / 100 Recovered   0
Confirmed last week      0
1 week change            0
1 week % increase        0
WHO Region               0
dtype: int64
```

In [6]:

```
df1_modified = df1.drop(['Confirmed last week', '1 week change', '1 week % increase
                         'Deaths / 100 Recovered' ], axis=1)

df1_modified.rename(columns={'Country/Region':'Country',
                            'New cases':'New_cases',
                            'New deaths':'New_deaths',
                            'New recovered':'New_recovered',
                            'WHO Region':'WHO_Region'}, inplace=True)
```

In [7]:

```
df1_modified = df1_modified.set_index('Country')
df1_modified.head(3)
```

Out[7]:

	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths	New_recover
Country							
Afghanistan	36263	1269	25198	9796	106	10	
Albania	4880	144	2745	1991	117	6	
Algeria	27973	1163	18837	7973	616	8	7

In [8]:

```
df1_modified.to_csv('Country_wise.csv', sep=',', encoding='utf-8')
```

2. covid_19_clan_complete

In [9]:

```
df2 = pd.read_csv('Datasets/covid_19_clean_complete.csv')
df2.head()
```

Out[9]:

	Province_State	Country_Region	Lat	Long	Date	Confirmed	Deaths	Re
0	NaN	Afghanistan	33.93911	67.709953	1/22/2020	0	0	
1	NaN	Albania	41.15330	20.168300	1/22/2020	0	0	
2	NaN	Algeria	28.03390	1.659600	1/22/2020	0	0	
3	NaN	Andorra	42.50630	1.521800	1/22/2020	0	0	
4	NaN	Angola	-11.20270	17.873900	1/22/2020	0	0	

In [10]:

```
list(df2.columns)
```

Out[10]:

```
['Province_State',
 'Country_Region',
 'Lat',
 'Long',
 'Date',
 'Confirmed',
 'Deaths',
 'Recovered',
 'Active',
 'WHO_Region']
```

In [11]:

```
#df1.describe()
df2.shape
```

Out[11]:

```
(49068, 10)
```

Continuously searching for missing values. Many data in the "Province/State" column are missing; in fact, 70.11% of the data in this particular column are missing. Because of this deficiency and the fact that our study is primarily focused on countries, we will omit this particular column.

In [12]:

```
df2.isnull().sum()
```

Out[12]:

```
Province_State    34404
Country_Region        0
Lat                   0
Long                  0
Date                  0
Confirmed             0
Deaths                0
Recovered             0
Active                0
WHO_Region            0
dtype: int64
```

In [13]:

```
df2_modified = df2.drop(['Province_State', 'Lat', 'Long'], axis=1)
df2_modified.rename(columns={'Country_Region':'Country'}, inplace=True)
```

In [14]:
```
df2_modified['a'] = df2_modified['Confirmed'].abs()
df2_modified['b'] = df2_modified['Deaths'].abs()
df2_modified['c'] = df2_modified['Recovered'].abs()
df2_modified['d'] = df2_modified['Active'].abs()
```

In [15]:
```
df2_modified['a'] = df2_modified.a.astype('int64')
df2_modified['b'] = df2_modified.b.astype('int64')
df2_modified['c'] = df2_modified.c.astype('int64')
df2_modified['d'] = df2_modified.d.astype('int64')
```

In [16]:
```
df2_modified = df2_modified.drop(['Confirmed', 'Deaths', 'Recovered', 'Active'], ax
```

In [17]:
```
df2_modified.rename(columns={
    'a':'Confirmed',
    'b':'Deaths',
    'c':'Recovered',
    'd':'Active'}, inplace=True)
```

In [18]:
```
df2_modified = df2_modified[['Country','Date','Confirmed','Deaths','Recovered', 'Ac
```

In [19]:
```
df2_modified = df2_modified.drop_duplicates(subset=['Country','Date'] )
```

In [20]:
```
df2_modified = df2_modified.set_index('Country')
df2_modified.head(3)
```

Out[20]:

	Date	Confirmed	Deaths	Recovered	Active	WHO_Region
Country						
Afghanistan	1/22/2020	0	0	0	0	Eastern Mediterranean
Albania	1/22/2020	0	0	0	0	Europe
Algeria	1/22/2020	0	0	0	0	Africa

In [21]:
```
df2_modified.to_csv('Covid19_clan.csv', sep=',', encoding='utf-8')
```

3. day_wise.csv

In [22]:
```
df3 = pd.read_csv('Datasets/day_wise.csv')
df3.head()
```

Out[22]:

	Date	Confirmed	Deaths	Recovered	Active	New cases	New deaths	New recovered	Deaths / 100 Cases	Recov / C
0	2020-01-22	555	17	28	510	0	0	0	3.06	
1	2020-01-23	654	18	30	606	99	1	2	2.75	
2	2020-01-24	941	26	36	879	287	8	6	2.76	
3	2020-01-25	1434	42	39	1353	493	16	3	2.93	
4	2020-01-26	2118	56	52	2010	684	14	13	2.64	

In [23]:

```
list(df3.columns)
```

Out[23]:

```
['Date',
 'Confirmed',
 'Deaths',
 'Recovered',
 'Active',
 'New cases',
 'New deaths',
 'New recovered',
 'Deaths / 100 Cases',
 'Recovered / 100 Cases',
 'Deaths / 100 Recovered',
 'No. of countries']
```

In [24]:

```
#df3.describe()
df3.shape
```

Out[24]:

```
(188, 12)
```

In [25]:

```
df3.isnull().sum() #Always checking for missing values. We do not have missing valu
```

Out[25]:

```
Date                      0
Confirmed                 0
Deaths                    0
Recovered                 0
Active                    0
New cases                 0
New deaths                0
New recovered             0
Deaths / 100 Cases        0
Recovered / 100 Cases     0
Deaths / 100 Recovered    0
No. of countries          0
dtype: int64
```

In [26]:
```
df3_modified = df3.drop(['Deaths / 100 Cases','Recovered / 100 Cases','Deaths / 100

df3_modified.rename(columns={'New cases':'New_cases',
                            'New deaths':'New_deaths',
                            'New recovered':'New_recovered',
                            'No. of countries':'Number_of_countries'}, inplace=True
```

In [27]:
```
df3_modified = df3_modified.set_index('Date')
df3_modified.head(3)
```

Out[27]:

	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths	New_recovered	N
Date								
2020-01-22	555	17	28	510	0	0	0	
2020-01-23	654	18	30	606	99	1	2	
2020-01-24	941	26	36	879	287	8	6	

In [28]:
```
df3_modified.to_csv('Day_wise.csv', sep=',', encoding='utf-8')
```

4. full_grouped

In [29]:
```
df4 = pd.read_csv('Datasets/full_grouped.csv')
df4.head()
```

Out[29]:

	Date	Country_Region	Confirmed	Deaths	Recovered	Active	New_cases	New_de
0	1/22/2020	Afghanistan	0	0	0	0	0	
1	1/22/2020	Albania	0	0	0	0	0	
2	1/22/2020	Algeria	0	0	0	0	0	
3	1/22/2020	Andorra	0	0	0	0	0	
4	1/22/2020	Angola	0	0	0	0	0	

In [30]:
```
list(df4.columns)
```

Out[30]:
```
['Date',
 'Country_Region',
 'Confirmed',
 'Deaths',
 'Recovered',
 'Active',
 'New_cases',
 'New_deaths',
 'New_recovered',
 'WHO_Region']
```

In [31]:
```
df4.shape
```

Out[31]:
```
(35156, 10)
```

In [32]:
```
df4.isnull().sum() #Always checking for missing values. We do not have missing valu
```

Out[32]:
```
Date              0
Country_Region    0
Confirmed         0
Deaths            0
Recovered         0
Active            0
New_cases         0
New_deaths        0
New_recovered     0
WHO_Region        0
dtype: int64
```

In [33]:
```
df4['New_recovered'] = df4['New_recovered'].abs()
df4['New_cases'] = df4['New_cases'].abs()
df4['New_deaths'] = df4['New_deaths'].abs()
df4['Active'] = df4['Active'].abs()
df4['Recovered'] = df4['Recovered'].abs()
df4['Deaths'] = df4['Deaths'].abs()
df4['Confirmed'] = df4['Confirmed'].abs()
```

In [34]:
```
df4_modified = df4.rename(columns={'Country_Region':'Country'})
```

In [35]:
```
df4_modified = df4_modified.set_index('Country')
df4_modified.head(3)
```

Out[35]:

	Date	Confirmed	Deaths	Recovered	Active	New_cases	New_deaths	N
Country								
Afghanistan	1/22/2020	0	0	0	0	0	0	
Albania	1/22/2020	0	0	0	0	0	0	
Algeria	1/22/2020	0	0	0	0	0	0	

In [36]:
```
df4_modified.to_csv('Full_detail.csv', sep=',', encoding='utf-8')
```

5. usa_country_wise

In [37]:
```
df5 = pd.read_csv('Datasets/usa_county_wise.csv')
df5.head()
```

Out[37]:

	UID	iso2	iso3	code3	FIPS	Admin2	Province_State	Country_Region	
0	16	AS	ASM	16	60.0	NaN	American Samoa	US	-14.27
1	316	GU	GUM	316	66.0	NaN	Guam	US	13.44
2	580	MP	MNP	580	69.0	NaN	Northern Mariana Islands	US	15.09
3	63072001	PR	PRI	630	72001.0	Adjuntas	Puerto Rico	US	18.18
4	63072003	PR	PRI	630	72003.0	Aguada	Puerto Rico	US	18.36

In [38]:
```
list(df5.columns)
```

Out[38]:
```
['UID',
 'iso2',
 'iso3',
 'code3',
 'FIPS',
 'Admin2',
 'Province_State',
 'Country_Region',
 'Lat',
 'Long_',
 'Combined_Key',
 'Date',
 'Confirmed',
 'Deaths']
```

In [39]:
```
df5.shape
```

Out[39]:
```
(627920, 14)
```

In [40]:
```
df5.isnull().sum() #Always checking for missing values.
```

Out[40]:

```
UID                  0
iso2                 0
iso3                 0
code3                0
FIPS              1880
Admin2            1128
Province_State       0
Country_Region       0
Lat                  0
Long_                0
Combined_Key         0
Date                 0
Confirmed            0
Deaths               0
dtype: int64
```

Once again, we will ignore all the following columns ['UID', 'iso2', 'iso3', 'code3', 'FIPS', 'Admin2', 'Long_', 'Lat', 'Country_Region'] because they do not support our investigation according to the objective established and the question we have to answer.

In [41]:

```
df5_modified = df5.drop(['UID','iso2','iso3','code3','FIPS','Admin2','Long_','Lat',
```

Duplicate rows

In [42]:

```
df5_modified = df5_modified.drop_duplicates(subset=['Province_State','Date'] )
df5_modified.rename(columns={'Province_State':'City'}, inplace=True)
```

In [43]:

```
df5_modified = df5_modified.set_index('City')
```

In [44]:

```
df5_modified.head(3)
```

Out[44]:

	Date	Confirmed	Deaths
City			
American Samoa	1/22/20	0	0
Guam	1/22/20	0	0
Northern Mariana Islands	1/22/20	0	0

In [45]:

```
df5_modified.to_csv('USA_wise.csv', sep=',', encoding='utf-8')
```

6. worldometer_data

In [46]:

```
df6 = pd.read_csv('Datasets/worldometer_data.csv')
df6.head()
```

Out[46]:

	Country/Region	Continent	Population	TotalCases	NewCases	TotalDeaths	NewDeat
0	USA	North America	3.311981e+08	5032179	NaN	162804.0	Na
1	Brazil	South America	2.127107e+08	2917562	NaN	98644.0	Na
2	India	Asia	1.381345e+09	2025409	NaN	41638.0	Na
3	Russia	Europe	1.459409e+08	871894	NaN	14606.0	Na
4	South Africa	Africa	5.938157e+07	538184	NaN	9604.0	Na

In [47]:

```
list(df6.columns)
```

Out[47]:

```
['Country/Region',
 'Continent',
 'Population',
 'TotalCases',
 'NewCases',
 'TotalDeaths',
 'NewDeaths',
 'TotalRecovered',
 'NewRecovered',
 'ActiveCases',
 'Serious,Critical',
 'Tot Cases/1M pop',
 'Deaths/1M pop',
 'TotalTests',
 'Tests/1M pop',
 'WHO Region']
```

In [48]:

```
df6.shape
```

Out[48]:

```
(209, 16)
```

In [49]:

```
df6.isnull().sum() #Always checking for missing values.
```

Out[49]:

```
Country/Region        0
Continent             1
Population            1
TotalCases            0
NewCases            205
TotalDeaths          21
NewDeaths           206
TotalRecovered        4
NewRecovered        206
ActiveCases           4
Serious,Critical     87
Tot Cases/1M pop      1
Deaths/1M pop        22
TotalTests           18
Tests/1M pop         18
WHO Region           25
dtype: int64
```

We may remove the columns "NewCases," "NewRecovered," and "NewDeaths" without losing a significant amount of data because it is evident that they are nearly entirely composed of missing values.

In [50]:

```
df6_modified = df6.drop(['NewCases', 'NewRecovered', 'NewDeaths','Tot Cases/1M pop'

df6_modified.rename(columns={'Country/Region':'Country',
                             'Serious,Critical':'Serious_Critical',
                             'WHO Region':'WHO_Region'}, inplace=True)
```

In [51]:

```
df6_modified = df6_modified.set_index('Country')
```

In [52]:

```
df6_modified = df6_modified[df6_modified['Population'].notna()]
df6_modified = df6_modified[df6_modified['TotalDeaths'].notna()]
df6_modified = df6_modified[df6_modified['TotalRecovered'].notna()]
df6_modified = df6_modified[df6_modified['ActiveCases'].notna()]
df6_modified = df6_modified[df6_modified['Serious_Critical'].notna()]
df6_modified = df6_modified[df6_modified['TotalTests'].notna()]
```

In [53]:

```
df6_modified['Population'] = df6_modified.Population.astype('int64')
df6_modified['TotalTests'] = df6_modified.TotalTests.astype('int64')
df6_modified['Serious_Critical'] = df6_modified.Serious_Critical.astype('int64')
df6_modified['ActiveCases'] = df6_modified.ActiveCases.astype('int64')
df6_modified['TotalRecovered'] = df6_modified.TotalRecovered.astype('int64')
df6_modified['TotalDeaths'] = df6_modified.TotalDeaths.astype('int64')
df6_modified['TotalCases'] = df6_modified.TotalCases.astype('int64')
```

In [54]:

```
df6_modified.head(3)
```

Out[54]:

	Continent	Population	TotalCases	TotalDeaths	TotalRecovered	ActiveCases	Ser
Country							
USA	North America	331198130	5032179	162804	2576668	2292707	
Brazil	South America	212710692	2917562	98644	2047660	771258	
India	Asia	1381344997	2025409	41638	1377384	606387	

```
In [55]: df6_modified.to_csv('Worldometer.csv', sep=',', encoding='utf-8')
```

7. owid-covid-data

```
In [56]: df7 = pd.read_csv('Datasets/owid-covid-data.csv')
         df7.head()
```

Out[56]:

	iso_code	continent	location	date	total_cases	new_cases	new_cases_smoothed	tc
0	AFG	Asia	Afghanistan	2020-02-24	1.0	1.0	NaN	
1	AFG	Asia	Afghanistan	2020-02-25	1.0	0.0	NaN	
2	AFG	Asia	Afghanistan	2020-02-26	1.0	0.0	NaN	
3	AFG	Asia	Afghanistan	2020-02-27	1.0	0.0	NaN	
4	AFG	Asia	Afghanistan	2020-02-28	1.0	0.0	NaN	

5 rows × 59 columns

Rather than removing some of the 59 columns in this database, we will select a few that are relevant to our research objective.

```
In [57]: list(df7.columns)
```

```
Out[57]: ['iso_code',
          'continent',
          'location',
          '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_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',
          'new_tests',
          'total_tests',
          'total_tests_per_thousand',
          'new_tests_per_thousand',
          'new_tests_smoothed',
          'new_tests_smoothed_per_thousand',
          'positive_rate',
          'tests_per_case',
          'tests_units',
          'total_vaccinations',
          'people_vaccinated',
          'people_fully_vaccinated',
          'new_vaccinations',
          'new_vaccinations_smoothed',
          'total_vaccinations_per_hundred',
          'people_vaccinated_per_hundred',
          'people_fully_vaccinated_per_hundred',
          'new_vaccinations_smoothed_per_million',
          'stringency_index',
          'population',
          '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']
```

In [58]:
```
df7.shape
```

Out[58]:
```
(91026, 59)
```

In [59]:
```
df7_modified = df7[['continent','date','total_cases','total_deaths','reproduction_r
                    'median_age','aged_65_older','life_expectancy']]
```

In [60]:
```
df7_modified.isnull().sum()
```

Out[60]:
```
continent               4327
date                       0
total_cases             2690
total_deaths           12542
reproduction_rate      17659
total_tests            50153
positive_rate          46582
total_vaccinations     78920
population               604
population_density      6364
median_age              9273
aged_65_older          10197
life_expectancy         4594
dtype: int64
```

In [61]:
```
## We removed the column labeled "total vaccinations" because it contains 86.7% mis
df7_modif = df7_modified.drop(['total_vaccinations'], axis=1)
```

There is a time constraint on our study. We must ensure that DF7 and DF8 are in that rage because that runs from January 22, 2020, to July 27, 2020.

In [62]:
```
df7_mod = df7_modif[df7_modif['date']<'2020-07-28']
df7_mod = df7_mod.drop_duplicates(subset=['continent','date'] )
```

In [63]:
```
df7_mod = df7_mod[df7_mod['continent'].notna()]
df7_mod = df7_mod[df7_mod['total_cases'].notna()]
df7_mod = df7_mod[df7_mod['total_deaths'].notna()]
df7_mod = df7_mod[df7_mod['reproduction_rate'].notna()]
df7_mod = df7_mod[df7_mod['total_tests'].notna()]
df7_mod = df7_mod[df7_mod['total_tests'].notna()]
df7_mod = df7_mod[df7_mod['positive_rate'].notna()]
df7_mod = df7_mod[df7_mod['population'].notna()]
df7_mod = df7_mod[df7_mod['population_density'].notna()]
df7_mod = df7_mod[df7_mod['median_age'].notna()]
df7_mod = df7_mod[df7_mod['aged_65_older'].notna()]
df7_mod = df7_mod[df7_mod['life_expectancy'].notna()]
```

In [64]:
```
df7_mod['total_cases'] = df7_mod.total_cases.astype('int64')
df7_mod['total_deaths'] = df7_mod.total_deaths.astype('int64')
df7_mod['total_tests'] = df7_mod.total_tests.astype('int64')
df7_mod['population'] = df7_mod.population.astype('int64')
```

```
df7_mod['aged_65_older'] = df7_mod.aged_65_older.apply(np.round)
df7_mod['aged_65_older'] = df7_mod.aged_65_older.astype('int64')
```

In [65]:
```
df7_mod['Total_cases'] = df7_mod['total_cases'].abs()
df7_mod['Total_deaths'] = df7_mod['total_deaths'].abs()
df7_mod['Reproduction_rate'] = df7_mod['reproduction_rate'].abs()
df7_mod['Total_tests'] = df7_mod['total_tests'].abs()
df7_mod['Population'] = df7_mod['population'].abs()
df7_mod['Population_density'] = df7_mod['population_density'].abs()
df7_mod['Median_age'] = df7_mod['median_age'].abs()
df7_mod['Aged_65_older'] = df7_mod['aged_65_older'].abs()
df7_mod['Life_expectancy'] = df7_mod['life_expectancy'].abs()
df7_mod['Positive_rate'] = df7_mod['positive_rate'].abs()
```

In [66]:
```
df7_mod = df7_mod.drop(['population','total_cases','positive_rate','total_deaths','
```

In [67]:
```
df7_mod = df7_mod[['continent','date','Total_cases','Total_deaths','Reproduction_ra
```

In [68]:
```
df7_mod = df7_mod.set_index('continent')
df7_mod.head(3)
```

Out[68]:

	date	Total_cases	Total_deaths	Reproduction_rate	Total_tests	Positive_rate	Po
continent							
Europe	2020-03-25	146	5	1.17	922	0.336	
Europe	2020-03-26	174	6	1.14	1025	0.334	
Europe	2020-03-27	186	8	1.08	1127	0.296	

In [69]:
```
df7_mod.to_csv('Covid_data.csv', sep=',', encoding='utf-8')
```

8. covid_19_usa_city

In [70]:
```
df8 = pd.read_csv('Datasets/covid_19_usa_city.csv')
df8.head()
```

Out[70]:

	City	Total Cases	New Cases	Total Deaths	New Deaths	Active Cases	Total Cases /1M pop	Deaths /1M pop	Total Tests	Test /1M po
0	New York	188694	7550.0	9385.0	758	162220	9618.0	478.0	461601.0	23,52
1	New Jersey	61850	3699.0	2350.0	167	58818	6964.0	265.0	126735.0	14,26
2	Michigan	23993	NaN	1392.0		22158	2410.0	140.0	76014.0	7,63
3	Massachusetts	22860	NaN	686.0		21445	3347.0	100.0	108776.0	15,92
4	Pennsylvania	22833	1029.0	507.0	6	21676	1785.0	40.0	124890.0	9,76

In [71]:

```
list(df8.columns)
```

Out[71]:

```
['City',
 'Total Cases',
 'New Cases',
 'Total Deaths',
 'New Deaths',
 'Active Cases',
 'Total Cases /1M pop',
 'Deaths /1M pop',
 'Total Tests',
 'Tests /1M pop',
 'Date']
```

In [72]:

```
df8.shape
```

Out[72]:

```
(9660, 11)
```

In [73]:

```
df8.isnull().sum()
```

```
Out[73]: City                  0
         Total Cases           0
         New Cases          4990
         Total Deaths         20
         New Deaths            0
         Active Cases          0
         Total Cases /1M pop  1288
         Deaths /1M pop     1290
         Total Tests         387
         Tests /1M pop      1288
         Date                  0
         dtype: int64
```

```
In [74]: df8['New Deaths']
```

```
Out[74]: 0       758
         1       167
         2
         3
         4         6
                ...
         9655
         9656
         9657
         9658
         9659
         Name: New Deaths, Length: 9660, dtype: object
```

Even though the 'New Deaths' column appears to have no missing values, upon running the following code, it was discovered that there are actually 5698 blank inputs, which indicates missing data. This means that around 59% of the values in this column are missing. As a result, we have decided to drop this column.

```
In [75]: (df8['New Deaths'].values == ' ').sum()
```

```
Out[75]: 6072
```

It is evident that 20 values, or 0.2% of the data, are missing from the column labeled "Total Deaths." Missing values can be removed without significantly altering the dataset.

```
In [76]: df8 = df8[df8['Total Deaths'].notna()]
         df8 = df8[df8['Total Tests'].notna()]
```

```
In [77]: df8_modified = df8.drop(['New Cases', 'New Deaths', 'Total Cases /1M pop','Deaths /
```

```
In [78]: df8_modified['City'].value_counts()
```

Out[78]:

```
City
Alaska                 161
Vermont                161
Mississippi            161
Minnesota              161
South Carolina         161
                      ...
 Wyoming                 1
 North Dakota            1
Grand Princess Ship      1
US Military              1
Puerto Rico              1
Name: count, Length: 104, dtype: int64
```

In [79]:

```
df8_modified.rename(columns={'Total Cases':'Total_c',
                             'Total Deaths':'Total_d',
                             'Active Cases':'Active_c',
                             'Total Tests':'Total_t'}, inplace=True)

df8_modified = df8_modified[df8_modified['Date'] < '07-28-2020']
df8_modified = df8_modified.set_index('City')
```

Additionally, the float values in the "Total Deaths" column do not make sense, so we are going to convert them to integers.

In [80]:

```
df8_modified['Total_d'] = df8_modified.Total_d.astype('int64')
df8_modified['Total_t'] = df8_modified.Total_t.astype('int64')
df8_modified['Total_c'] = df8_modified.Total_c.astype('int64')
df8_modified['Active_c'] = df8_modified.Active_c.astype('int64')
```

An analysis of the data reveals the presence of negative values in the columns 'Total Cases', 'Total Deaths', 'Active Cases', and 'Total Tests'. Such values are anomalous and cannot be valid entries. It is reasonable to assume that they are input errors. In order to rectify these errors, a conversion of the negative values to positive is necessary.

In [81]:

```
df8_modified['Total_deaths'] = df8_modified['Total_d'].abs()
df8_modified['Total_cases'] = df8_modified['Total_c'].abs()
df8_modified['Total_tests'] = df8_modified['Total_t'].abs()
df8_modified['Active_cases'] = df8_modified['Active_c'].abs()
```

In [82]:

```
df8_modified = df8_modified.drop(['Total_d', 'Total_t', 'Total_c', 'Active_c'], axi
```

In [83]:

```
df8_modified = df8_modified[['Total_cases', 'Total_deaths' , 'Total_tests', 'Active
```

In [84]:

```
df8_modified.head(3)
```

Out[84]:

	Total_cases	Total_deaths	Total_tests	Active_cases	Date
City					
New York	188694	9385	461601	162220	04-12-2020
New Jersey	61850	2350	126735	58818	04-12-2020
Michigan	23993	1392	76014	22158	04-12-2020

In [85]:

```
df8_modified['Date'].value_counts()
```

Out[85]:

```
Date
07-27-2020    58
06-24-2020    58
07-03-2020    58
07-02-2020    58
06-30-2020    58
              ..
04-17-2020    55
04-16-2020    55
04-15-2020    55
04-14-2020    55
04-13-2020    54
Name: count, Length: 101, dtype: int64
```

In [86]:

```
df8_modified.to_csv('USA_city_Covid19.csv', sep=',', encoding='utf-8')
```

In [ ]:

In [ ]:

EASTERN UNIVERSITY

COMPLETE PROJECT PDF

Databases_import.sql

```
CREATE TABLE Country_wise(
        Country varchar(50) NOT NULL,
        Confirmed int CHECK(Confirmed=0 OR Confirmed>0),
        Deaths int CHECK( Deaths=0 OR Deaths>0),
        Recovered int CHECK( Recovered=0 OR Recovered>0),
        Active int CHECK( Active=0 OR Active>0),
        New_cases int CHECK( New_cases=0 OR New_cases>0),
        New_deaths int CHECK( New_deaths=0 OR New_deaths>0),
        New_recovered int CHECK( New_recovered=0 OR New_recovered>0),
        WHO_Region varchar(50),
        CONSTRAINT Country_wise_pkey PRIMARY KEY (Country)
);

CREATE TABLE Day_wise(
        Date date NOT NULL,
        Confirmed int  CHECK( Confirmed=0 OR Confirmed>0),
        Deaths int CHECK( Deaths=0 OR Deaths>0),
        Recovered int CHECK( Recovered=0 OR Recovered>0),
        Active int CHECK( Active=0 OR Active>0),
        New_cases int CHECK( New_cases=0 OR New_cases>0),
        New_deaths int CHECK( New_deaths=0 OR New_deaths>0),
        New_recovered int CHECK( New_recovered=0 OR New_recovered>0),
        Number_of_countries smallint CHECK( Number_of_countries=0 OR
Number_of_countries>0),
        CONSTRAINT Day_wise_pkey PRIMARY KEY (Date)
);

CREATE TABLE Full_detail(
        Country varchar(50) NOT NULL,
        Date date NOT NULL,
        Confirmed int CHECK( Confirmed=0 OR Confirmed>0),
        Deaths int  CHECK( Deaths=0 OR Deaths>0),
        Recovered int CHECK( Recovered=0 OR Recovered>0),
        Active int CHECK( Active=0 OR Active>0),
        New_cases int CHECK( New_cases=0 OR New_cases>0),
        New_deaths int CHECK( New_deaths=0 OR New_deaths>0),
        New_recovered int CHECK( New_recovered=0 OR New_recovered>0),
        WHO_Region varchar(50),
        CONSTRAINT Full_detail_pkey PRIMARY KEY (Country,Date)
);

CREATE TABLE Covid19_clan(
        Country varchar(50) NOT NULL,
        Date date NOT NULL,
        Confirmed int CHECK( Confirmed=0 OR Confirmed>0),
        Deaths int CHECK( Deaths=0 OR Deaths>0),
        Recovered int CHECK( Recovered=0 OR Recovered>0),
        Active int CHECK( Active=0 OR Active>0),
        WHO_Region varchar(50),
        CONSTRAINT Covid19_clan_pkey PRIMARY KEY (Country,Date, WHO_Region),
        CONSTRAINT Covid19_clan_fkey FOREIGN KEY (Country,Date, WHO_Region)
REFERENCES Full_detail (Country,Date)
                ON DELETE SET NULL
);

CREATE TABLE USA_wise(
        City varchar(50) NOT NULL,
        Date date NOT NULL,
        Confirmed int CHECK( Confirmed=0 OR Confirmed>0),
        Deaths int CHECK( Deaths=0 OR Deaths>0),
        CONSTRAINT USA_wise_pkey PRIMARY KEY (City,Date)
);

CREATE TABLE Worldometer(
        Country varchar(50) NOT NULL,
        Continent varchar(30),
        Population bigint CHECK( Population=0 OR Population>0),
        TotalCases bigint CHECK( TotalCases=0 OR TotalCases>0),
        TotalDeaths bigint CHECK( TotalDeaths=0 OR TotalDeaths>0),
        TotalRecovered bigint CHECK( TotalRecovered=0 OR TotalRecovered>0),
        ActiveCases bigint CHECK( ActiveCases=0 OR ActiveCases>0),
        Serious_Critical int CHECK( Serious_Critical=0 OR Serious_Critical>0),
        TotalTests bigint CHECK( TotalTests=0 OR TotalTests>0),
        WHO_Region varchar(30),
        CONSTRAINT Worldometer_pkey PRIMARY KEY (Country)
);

CREATE TABLE USA_city_Covid19(
        City varchar(50) NOT NULL,
        Total_cases int CHECK( Total_cases=0 OR Total_cases>0),
        Total_deaths int CHECK( Total_deaths=0 OR Total_deaths>0),
        Active_cases int CHECK( Active_cases=0 OR Active_cases>0),
        Total_tests int CHECK( Total_tests=0 OR Total_tests>0),
        Date date,
        CONSTRAINT USA_city_Covid19_pkey PRIMARY KEY (City, Date)
);

CREATE TABLE Covid_data(
        continent varchar(50) NOT NULL,
        date date NOT NULL,
        total_cases int CHECK( total_cases=0 OR total_cases>0),
        total_deaths int CHECK( total_deaths=0 OR total_deaths>0),
        reproduction_rate decimal CHECK( reproduction_rate=0 OR reproduction_rate>0),
        total_tests int CHECK( total_tests=0 OR total_tests>0),
        positive_rate numeric CHECK( positive_rate=0 OR positive_rate>0),
        population bigint CHECK( population=0 OR population>0),
        population_density decimal CHECK( population_density=0 OR population_density>0),
        median_age numeric CHECK( median_age=0 OR median_age>0),
        aged_65_older numeric CHECK( aged_65_older=0 OR aged_65_older>0),
        life_expectancy decimal CHECK( life_expectancy=0 OR life_expectancy>0),
        CONSTRAINT Covid_data_pkey PRIMARY KEY (continent, date)
);
```

Import_dataset.sql

```
COPY Day_wise
FROM 'E:\DTSC_691_PROJECT\Day_wise.csv'
WITH (FORMAT CSV, HEADER);

COPY Full_detail
FROM 'E:\DTSC_691_PROJECT\Full_detail.csv'
WITH (FORMAT CSV, HEADER);

COPY Usa_city_Covid19
FROM 'E:\DTSC_691_PROJECT\Usa_city_Covid19.csv'
WITH (FORMAT CSV, HEADER);

COPY Usa_wise
```

```
FROM 'E:\DTSC_691_PROJECT\Usa_wise.csv'
WITH (FORMAT CSV, HEADER);

COPY Worldometer
FROM 'E:\DTSC_691_PROJECT\Worldometer.csv'
WITH (FORMAT CSV, HEADER);

COPY Country_wise
FROM 'E:\DTSC_691_PROJECT\Country_wise.csv'
WITH (FORMAT CSV, HEADER);

COPY Covid19_clan
FROM 'E:\DTSC_691_PROJECT\Covid19_clan.csv'
WITH (FORMAT CSV, HEADER);

COPY Covid_data
FROM 'E:\DTSC_691_PROJECT\Covid_data.csv'
WITH (FORMAT CSV, HEADER);
```

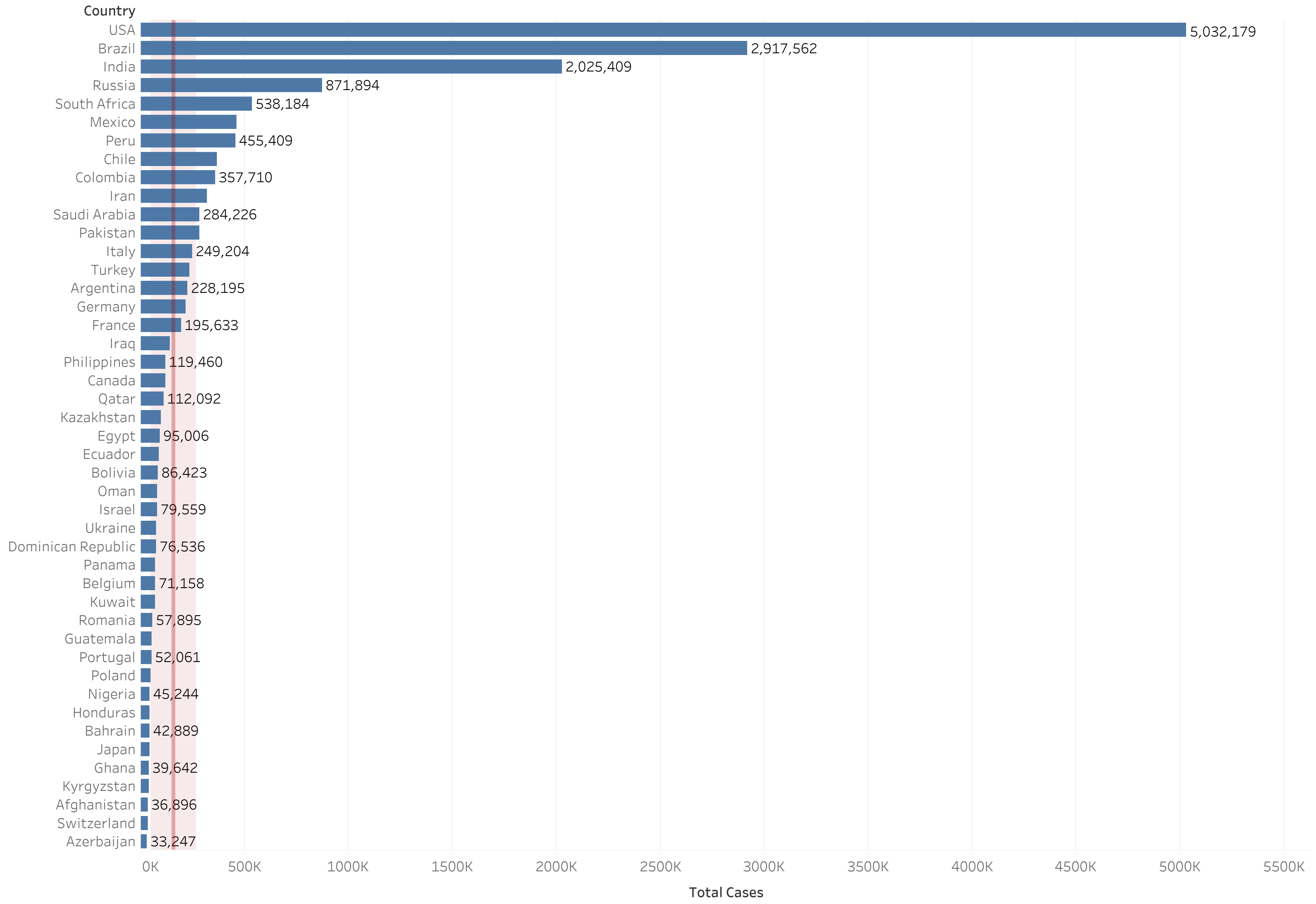
Database_queries.sql

```
-- Let's start addressing our business question or study questions

-- The number of cases of COVID-19 per geographical area:
-- Note: the table worldometer doesn't include China, however, the table country_wise does

-- 1. Continent
SELECT SUM(totalcases) AS cases,continent  FROM public.worldometer
        GROUP BY continent
                ORDER BY cases DESC;

-- 3. Country
SELECT SUM(totalcases) AS cases, country FROM public.worldometer
        GROUP BY country
                ORDER BY cases DESC;

SELECT SUM(confirmed) AS cases,country  FROM public.country_wise
        GROUP BY country
                ORDER BY cases DESC;
```

```
-- The number of deaths of COVID-19 per geographical area:
-- 1. Continent
SELECT SUM(totaldeaths) AS deaths, continent FROM public.worldometer
        GROUP BY continent
                ORDER BY deaths DESC;

-- 2. Country:
-- Note: the table worldometer doesn't include China, however, the table country_wise does

SELECT SUM(totaldeaths) AS deaths,country  FROM public.worldometer
        GROUP BY country
                ORDER BY deaths DESC;

SELECT SUM(deaths) AS deaths,country  FROM public.country_wise
        GROUP BY country
                ORDER BY deaths DESC;

-- The number of individuals who returned home after recovering from the Covid-19 in each
country, continent
-- Continent
SELECT SUM(totalrecovered) AS recovered, continent FROM public.worldometer
        GROUP BY continent
                ORDER BY recovered DESC;

-- Country
SELECT SUM(totalrecovered) AS recovered,country  FROM public.worldometer
        GROUP BY country
                ORDER BY recovered DESC;


SELECT SUM(recovered) AS recovered,country  FROM public.country_wise
        GROUP BY country
                ORDER BY recovered DESC;

-- Assessing the efficacy of treatment modalities by geographical area nation, continent

-- Continent
SELECT continent, ROUND((SUM(totalrecovered) * 100)/SUM(totalcases),2)  AS
efficacy_pourcentage
        FROM public.worldometer
                GROUP BY continent
                        ORDER BY efficacy_pourcentage DESC;
```

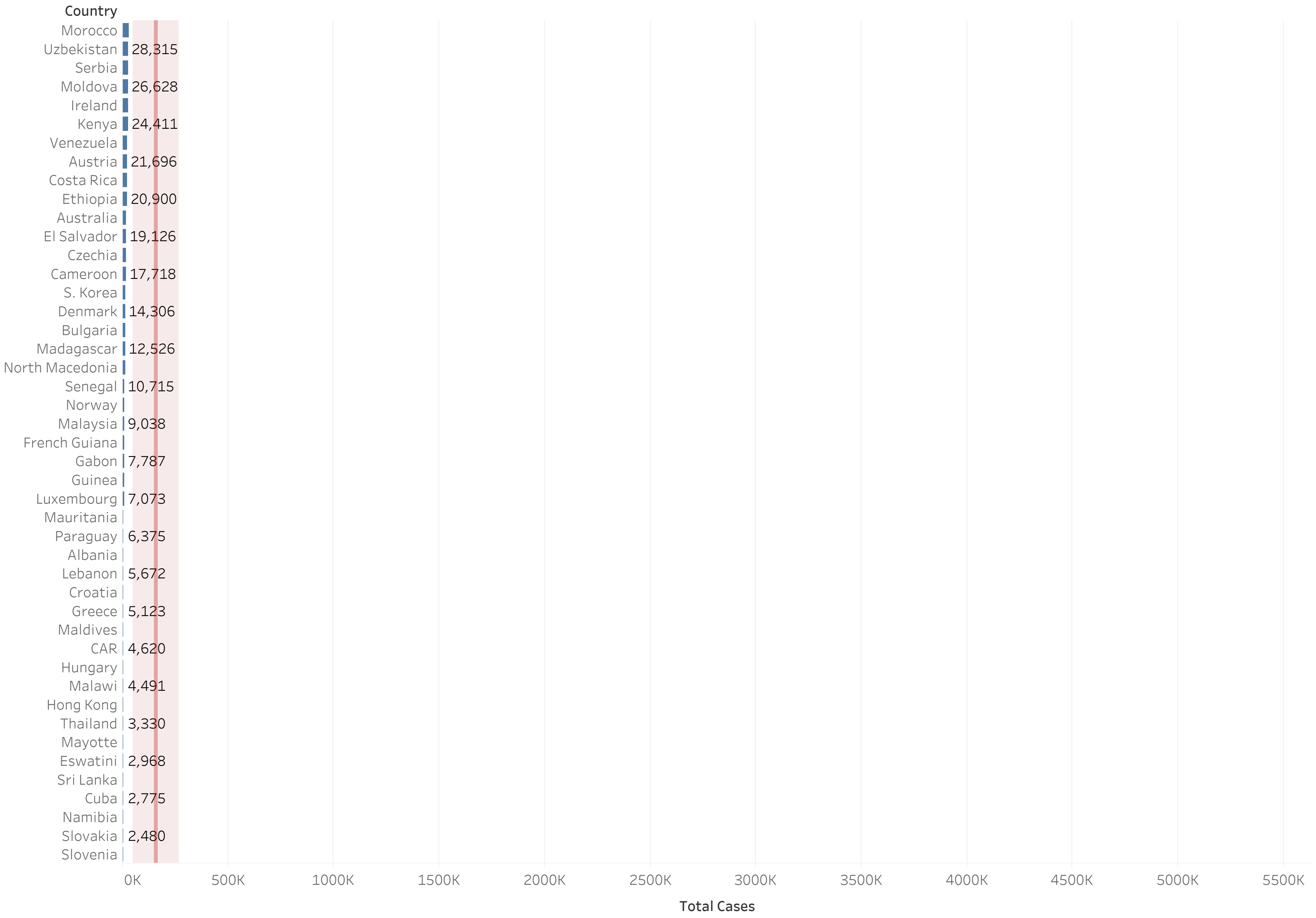
```
-- Country
SELECT country, ROUND((SUM(totalrecovered) * 100)/SUM(totalcases),2)  AS
efficacy_pourcentage
        FROM public.worldometer
                GROUP BY country
                        ORDER BY efficacy_pourcentage DESC;

-- Active cases of Covid-19 per day 2020-01-22 to 2020-07-27
-- Worldwide
SELECT date, active FROM public.covid19_clan
        WHERE country != 'US'
                GROUP BY date, active
                        ORDER BY date;

-- United States
SELECT date, active FROM public.covid19_clan
        WHERE country = 'US'
                GROUP BY date, active
                        ORDER BY date;

-- United States
SELECT date, SUM(confirmed) AS confirm FROM public.usa_wise
        GROUP BY date
                ORDER BY date

SELECT date, SUM(deaths) AS death FROM public.usa_wise
        GROUP BY date
                ORDER BY date;
```

Total cases by Country



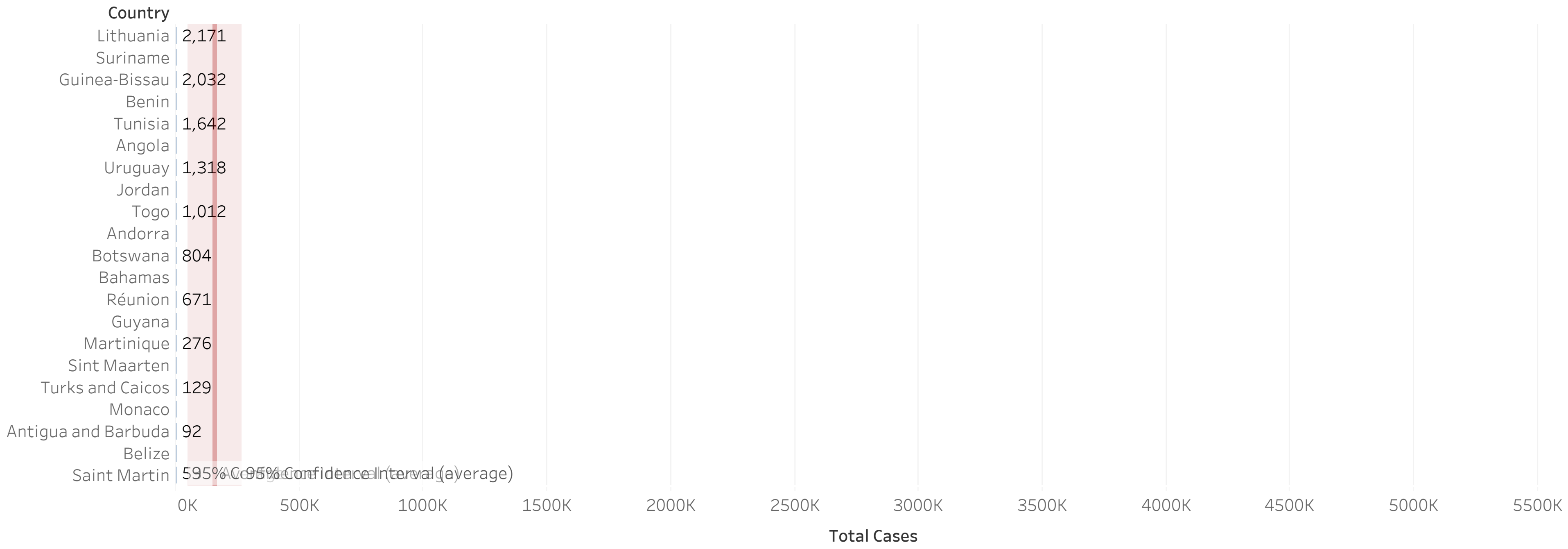
List of all confirmed cases of COVID19 in each country, ordered by country from 01-22-2020 to 07-27-2020 Expect China

Total cases by Country



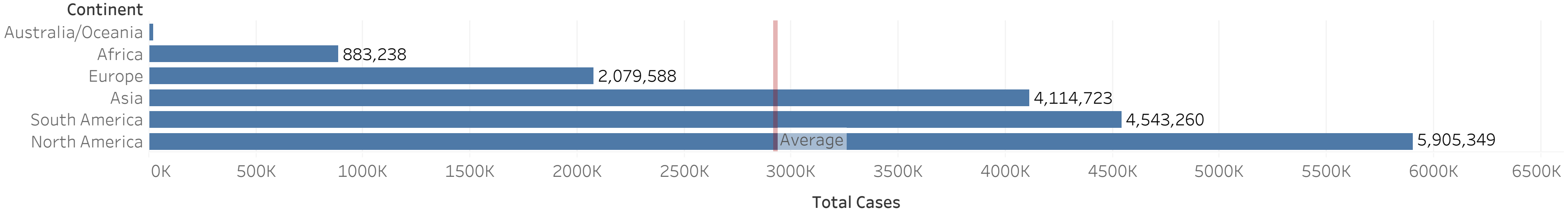
List of all confirmed cases of COVID19 in each country, ordered by country from 01-22-2020 to 07-27-2020 Expect China

Total cases by Country



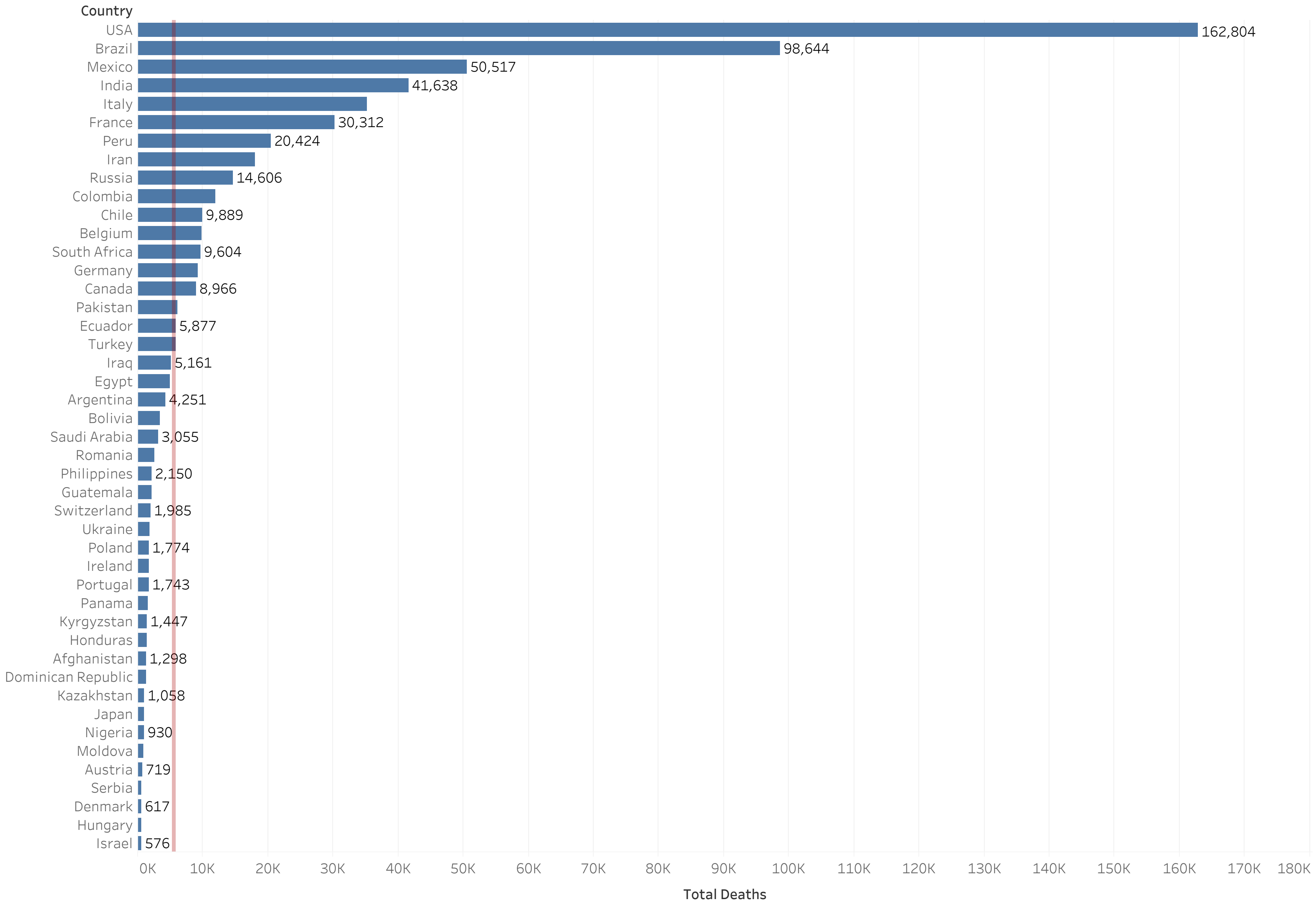
List of all confirmed cases of COVID19 in each country, ordered by country from 01-22-2020 to 07-27-2020 Expect China

Total cases by Continent



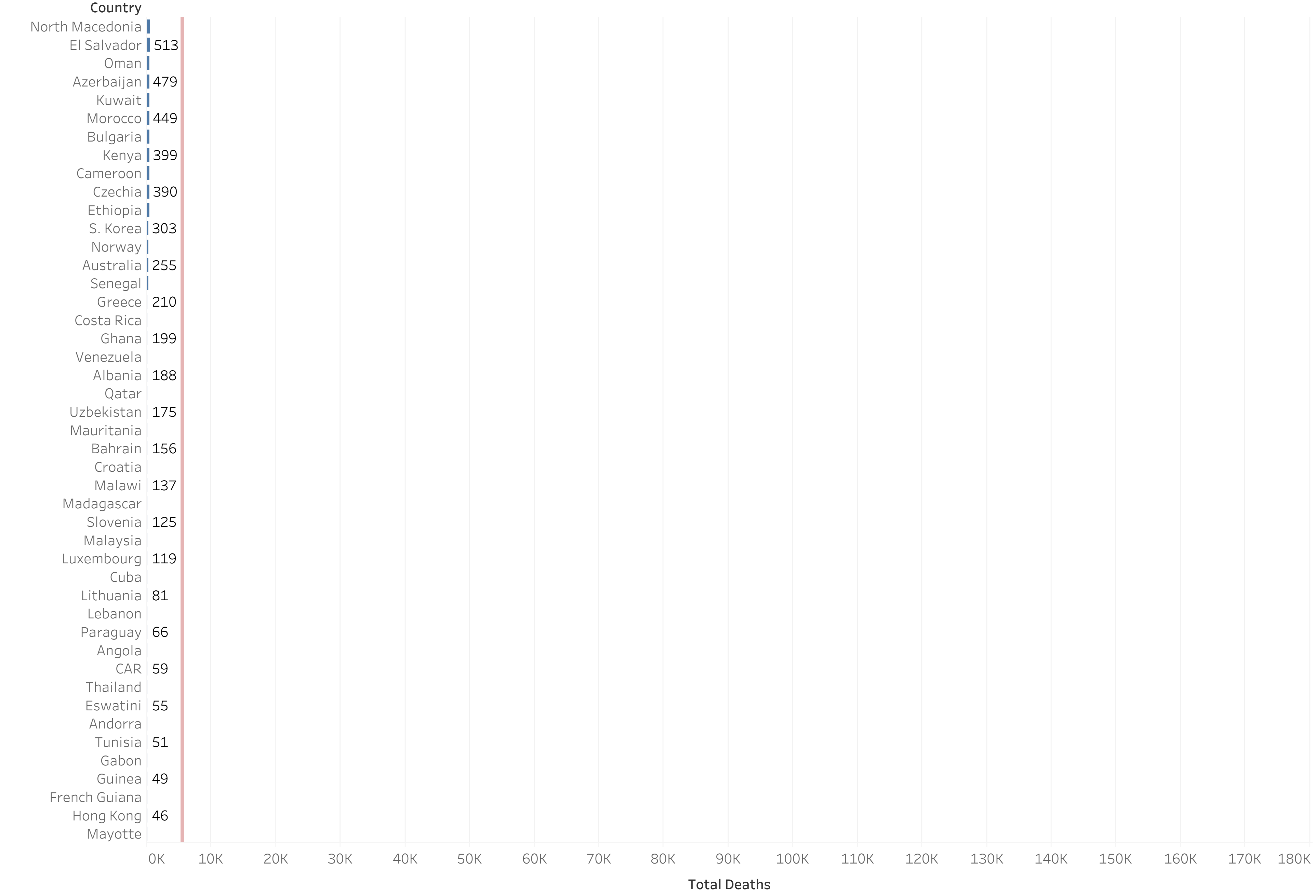
COVID19 cases confirmed on each continent, arranged by continent. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing from Asia

Total deaths by Country



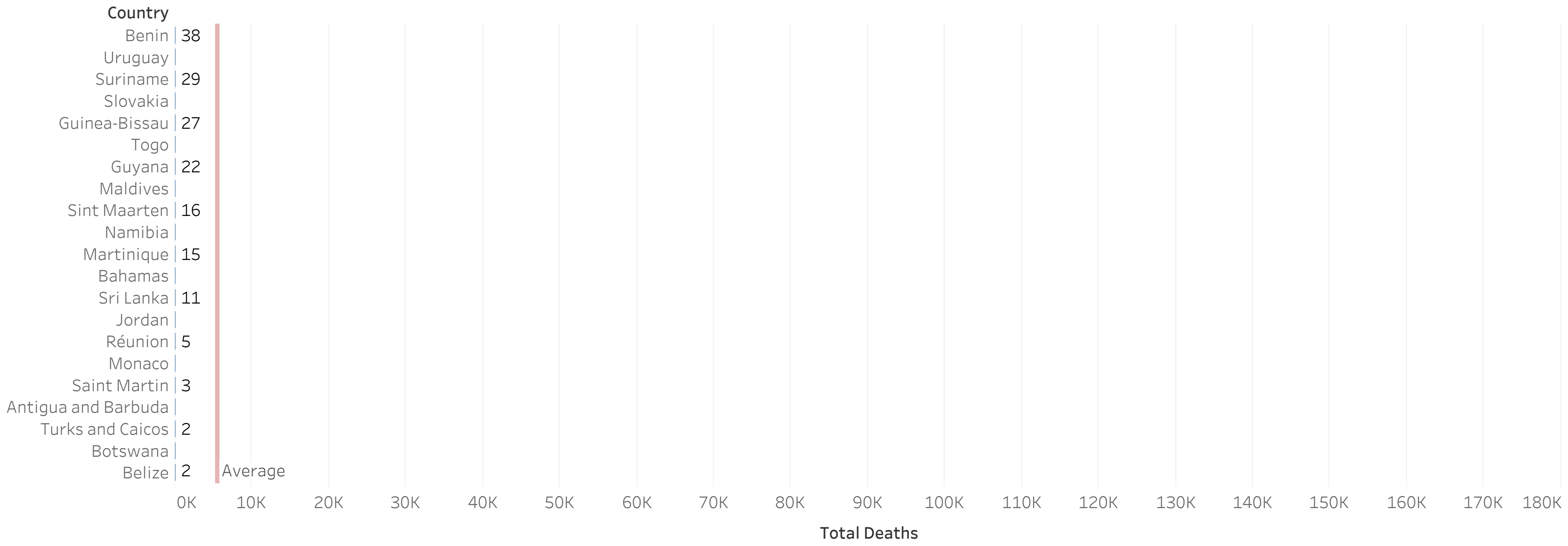
COVID19 deaths on each Country, arranged by Country. from 01-22-2020 to 07-27-2020. Note: China's statistics are missing.

Total deaths by Country



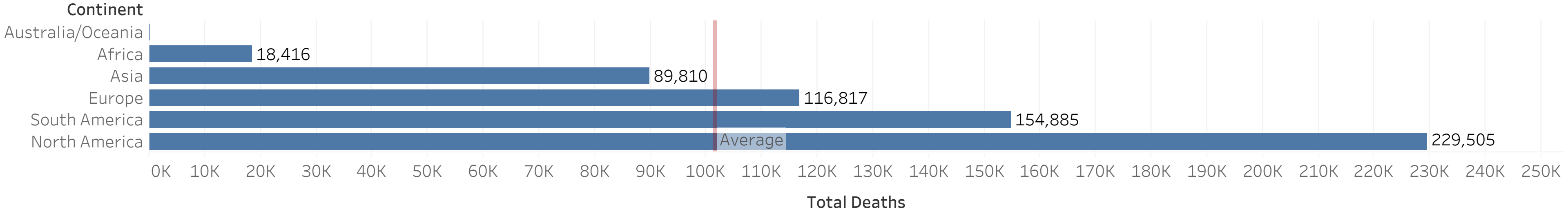
COVID19 deaths on each Country, arranged by Country. from 01-22-2020 to 07-27-2020. Note: China's statistics are missing.

Total deaths by Country



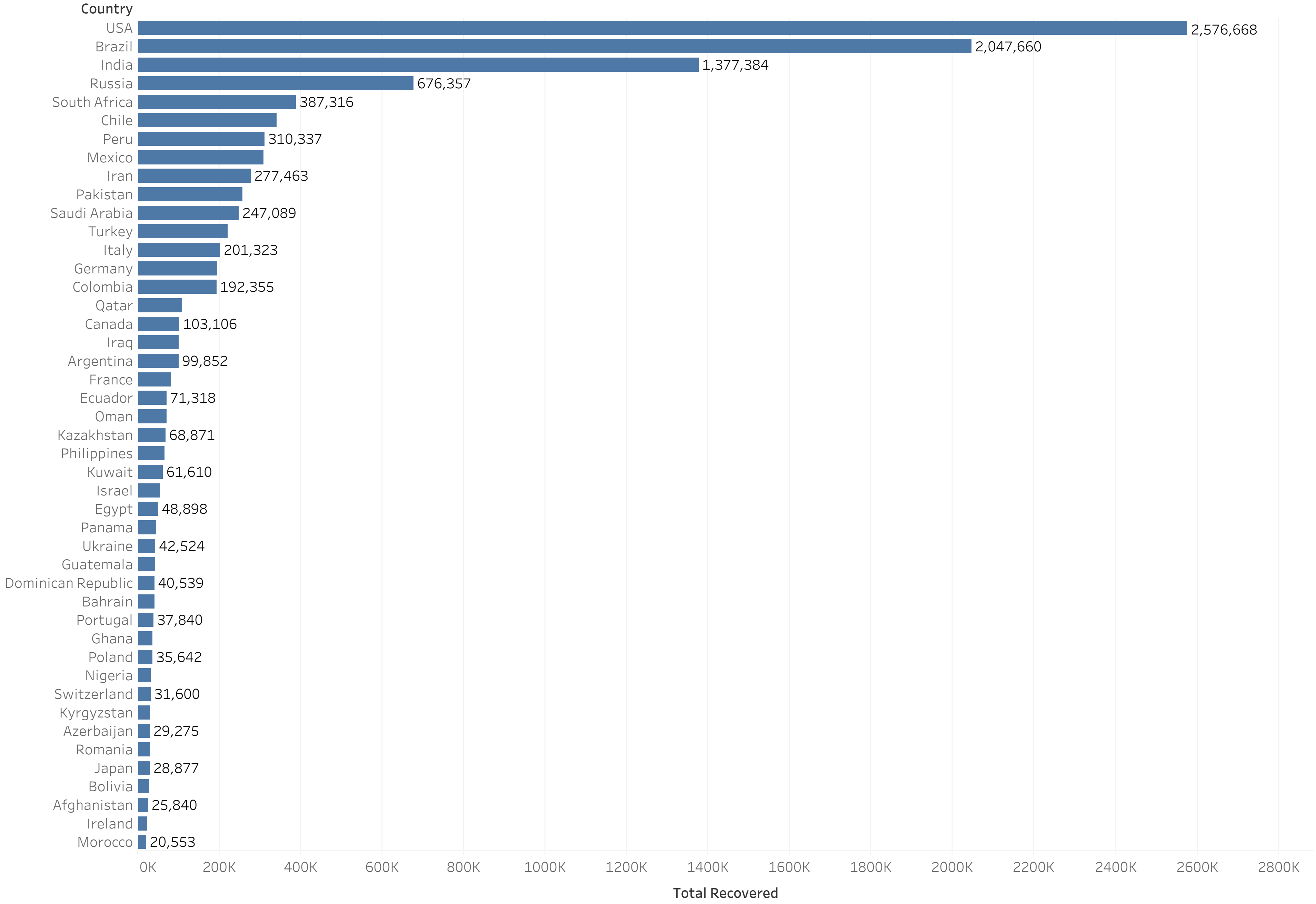
COVID19 deaths on each Country, arranged by Country. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing.

Total deaths by Continent



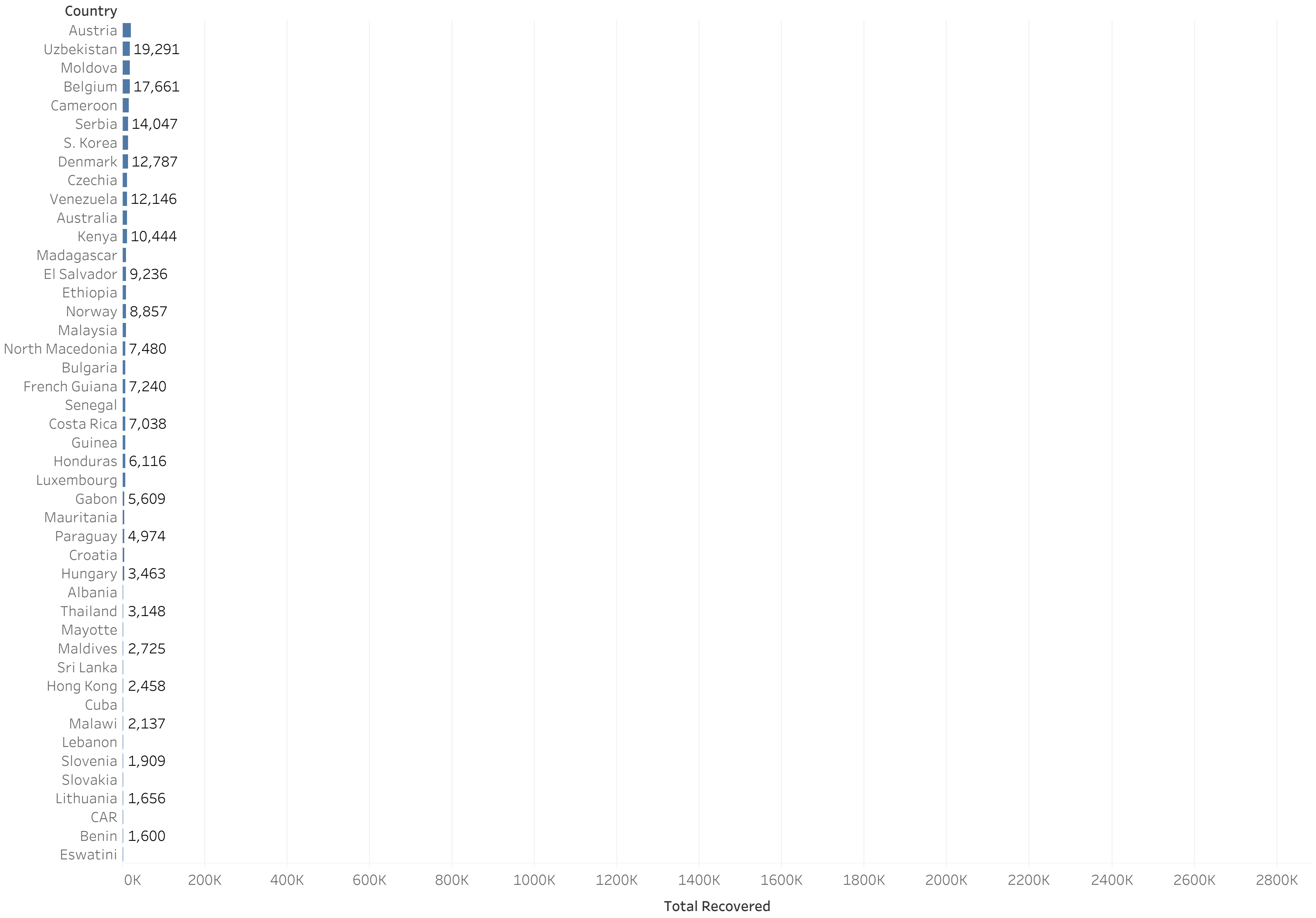
COVID19 deaths on each Continent, arranged by Continent. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing.

Total Recovered by Country



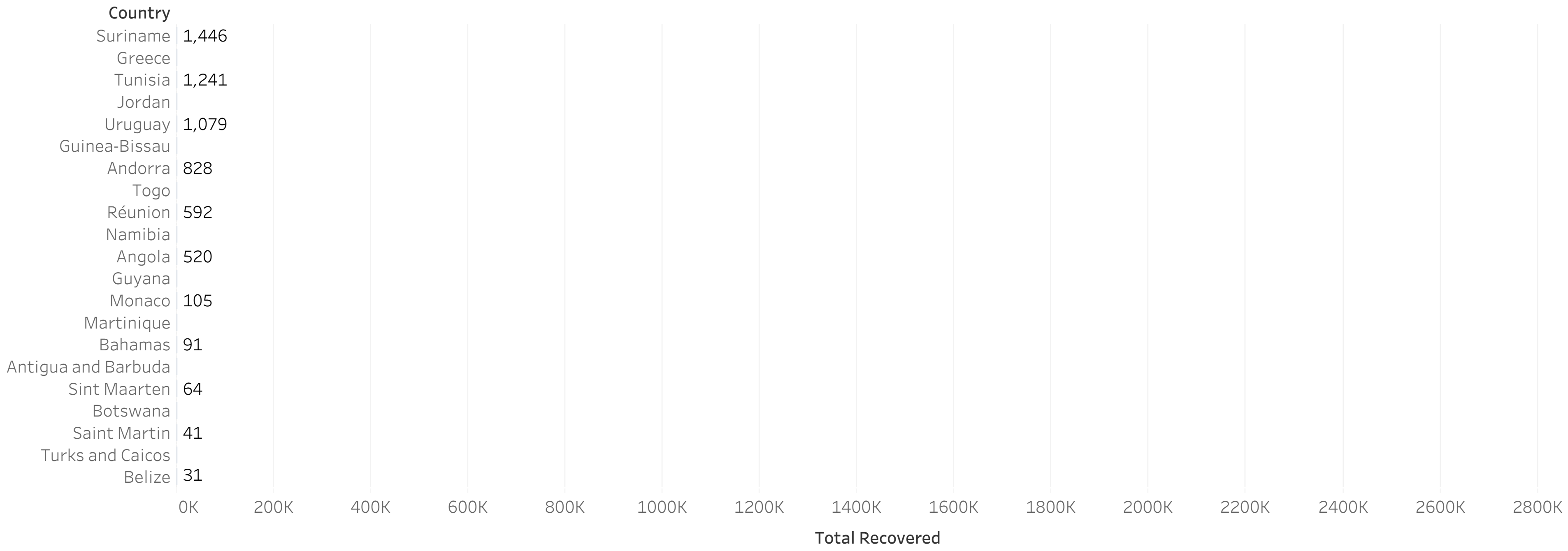
COVID19 Total Recovered on each Country. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing.

Total Recovered by Country



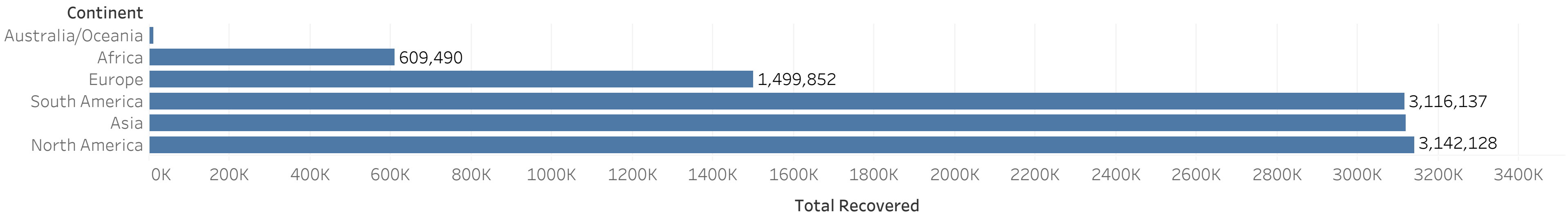
COVID19 Total Recovered on each Country. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing.

Total Recovered by Country



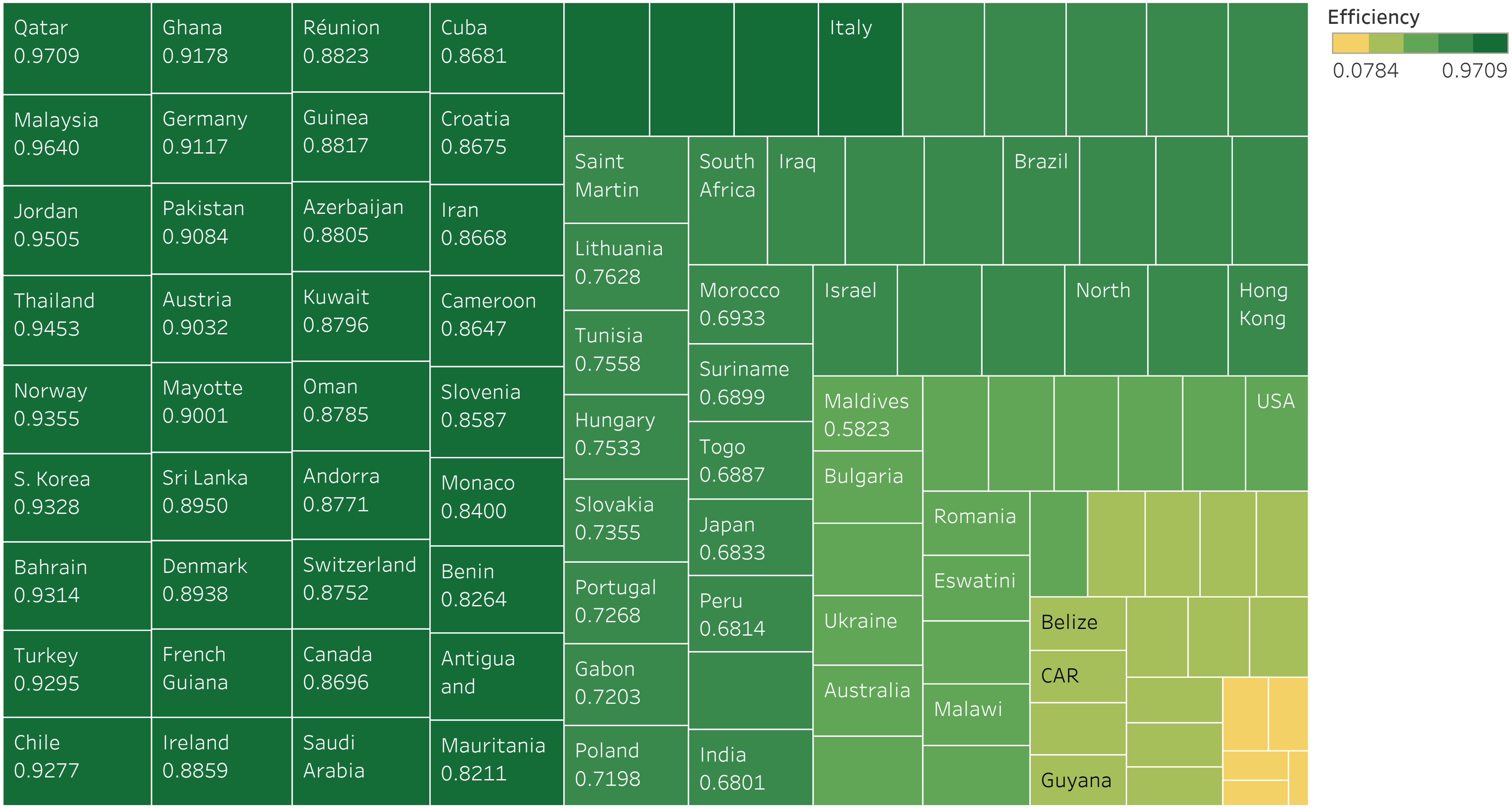
COVID19 Total Recovered on each Country. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing.

Total Recovered by Continent



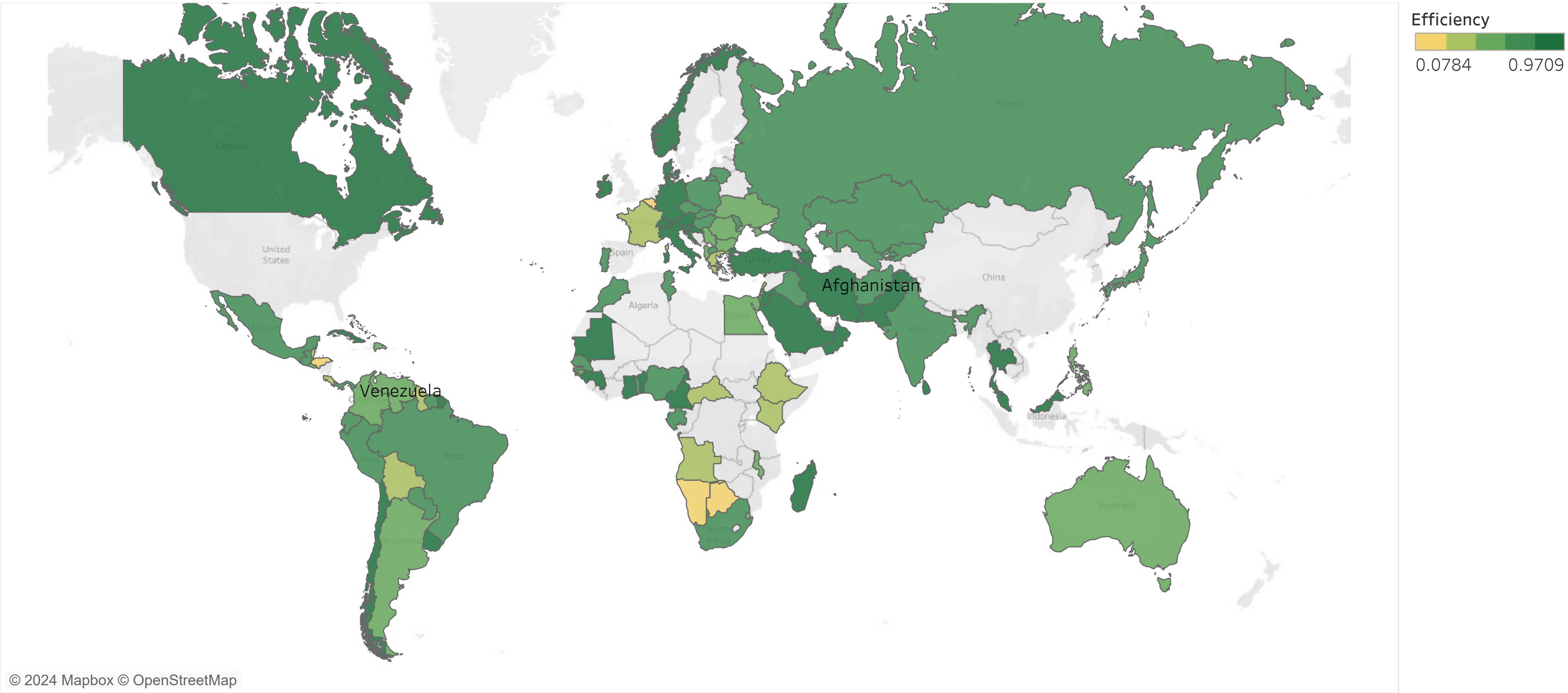
COVID19 Total Recovered on each Continent. from 01-22-2020 to 07-27-2020. Note: China’s statistics are missing.

Efficiency rate by Country Treemap



The efficiency rate, or survival rate, is the ratio of individuals who recovered from the COVID-19 pandemic. Total Recovered/Total Cases.

Efficiency rate by Country Except US



© 2024 Mapbox © OpenStreetMap

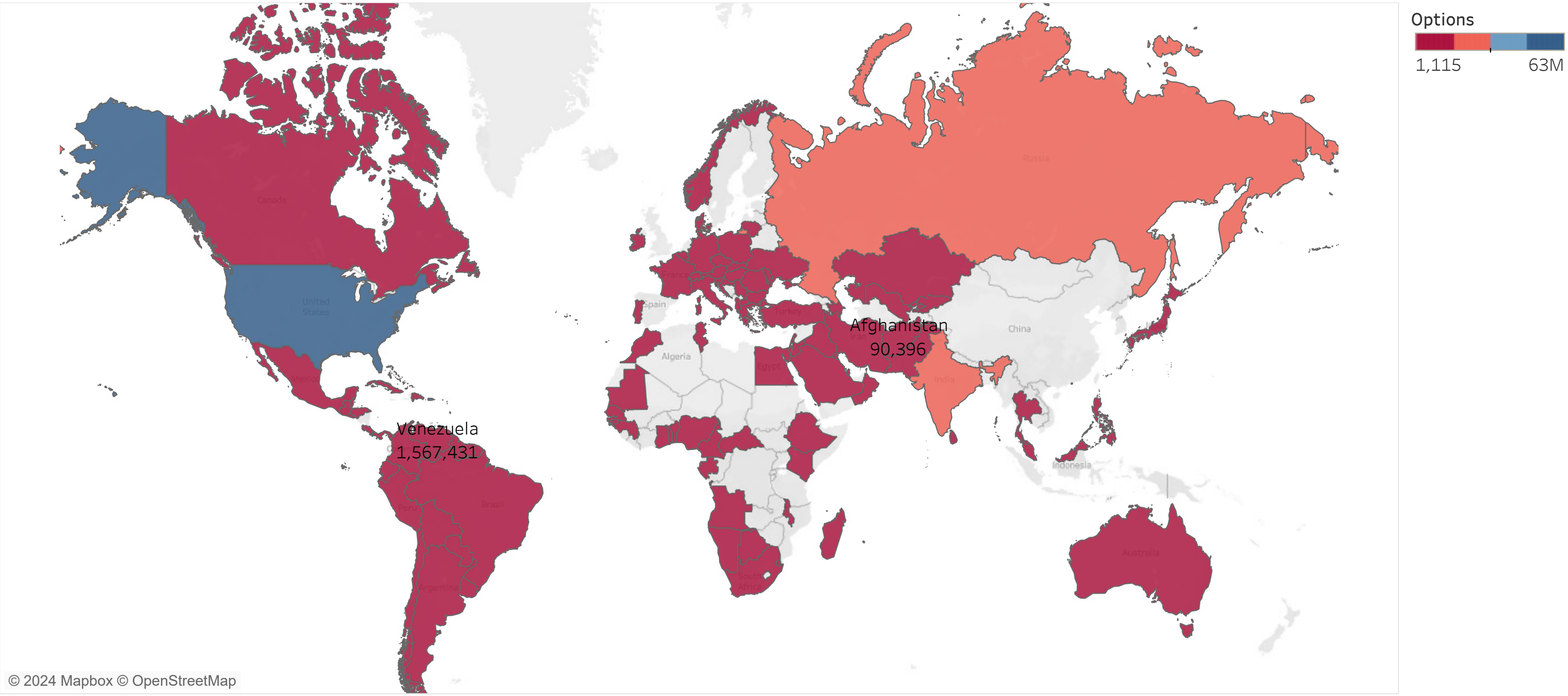
The efficiency rate, or survival rate, is the ratio of individuals who recovered from the COVID-19 pandemic. Total Recovered/Total Cases.

Efficiency rate USA

Efficiency
0.5120
0.5120
USA
Canada
Mexico
Colombia
Peru
Brazil
Argentina
France
Spain
Algeria
Egypt
Turkey
Iran
India
China
Russia
Japan
Indonesia
Australia
South Africa
© 2024 Mapbox © OpenStreetMap

The efficiency rate, or survival rate, in the United States

Worldwide Stat



The efficiency rate, or survival rate, is the ratio of individuals who recovered from the COVID-19 pandemic. Total Recovered/Total Cases.

Total Deaths

326,876

Sum of Deaths. The data is filtered on Continent, which keeps 6 of 6 members.

Total Recovered

5,244,297

Sum of Recovered. The data is filtered on Continent, which keeps 6 of 6 members.

Total Cases

7,611,676

Sum of Confirmed. The data is filtered on Continent, which keeps 6 of 6 members.

Total Tests

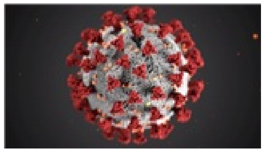
224,401,477

Sum of Total Tests. The data is filtered on Continent, which keeps 6 of 6 members.

Data source: WHO via

kaggles.com as of July 27, 2020

COVID 19 Pandemic Statistics



Total Cases	Total Recovered	Total Deaths	Total Tests
7,611,676	5,244,297	326,876	224,401,477

Continent

☑ Africa

☑ Asia

☑ Australia/Oceania

☑ Europe

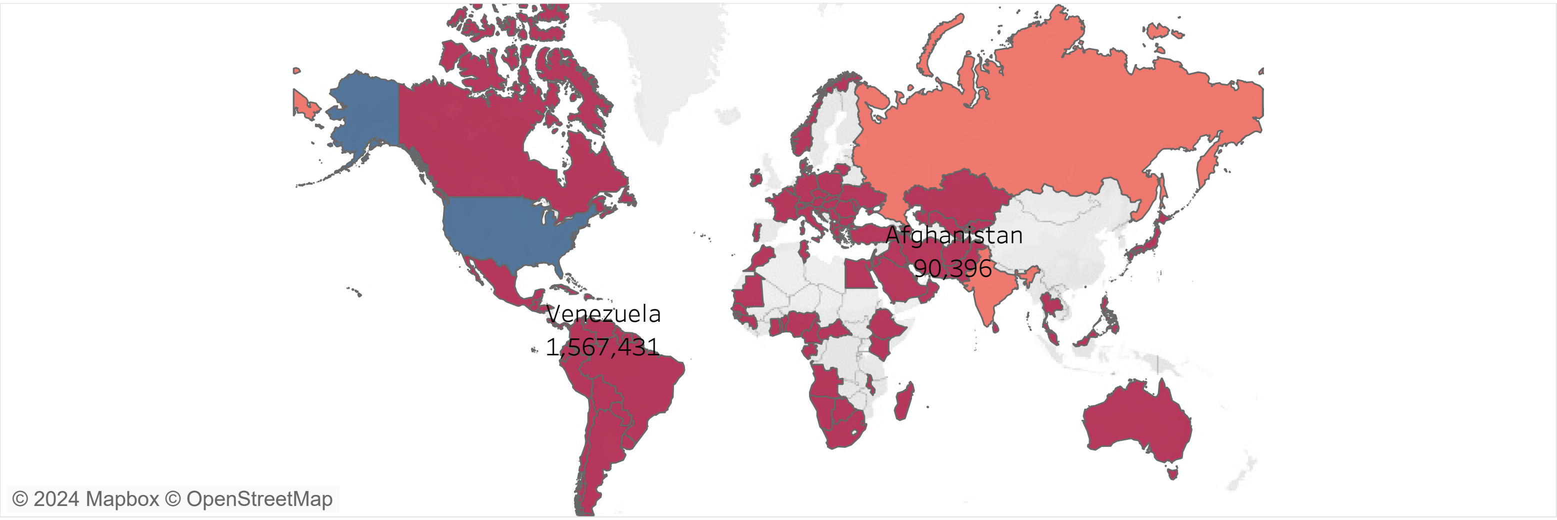
Options

1,115 63,139,605

Option

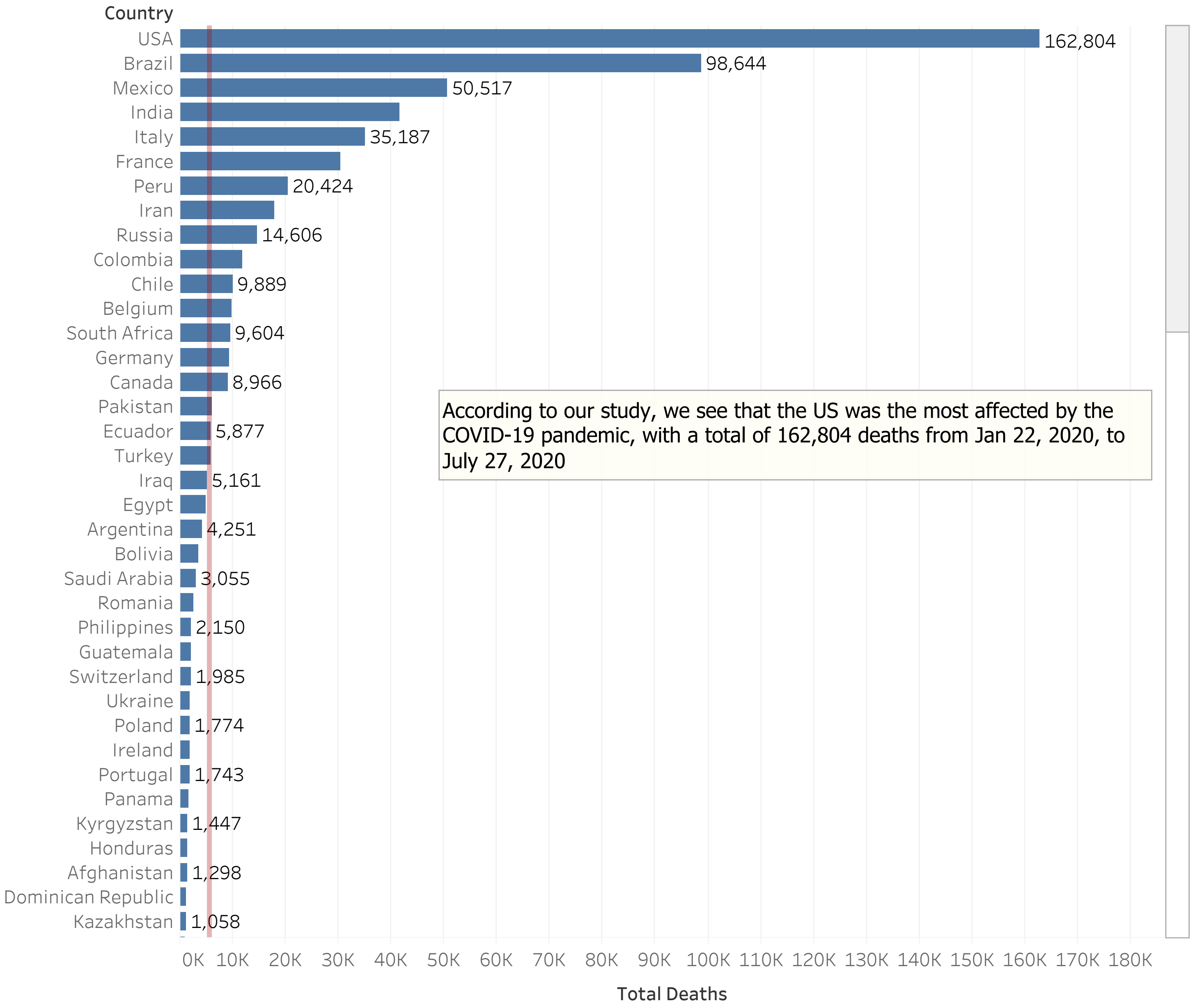
Total Tests

Worldwide Stat



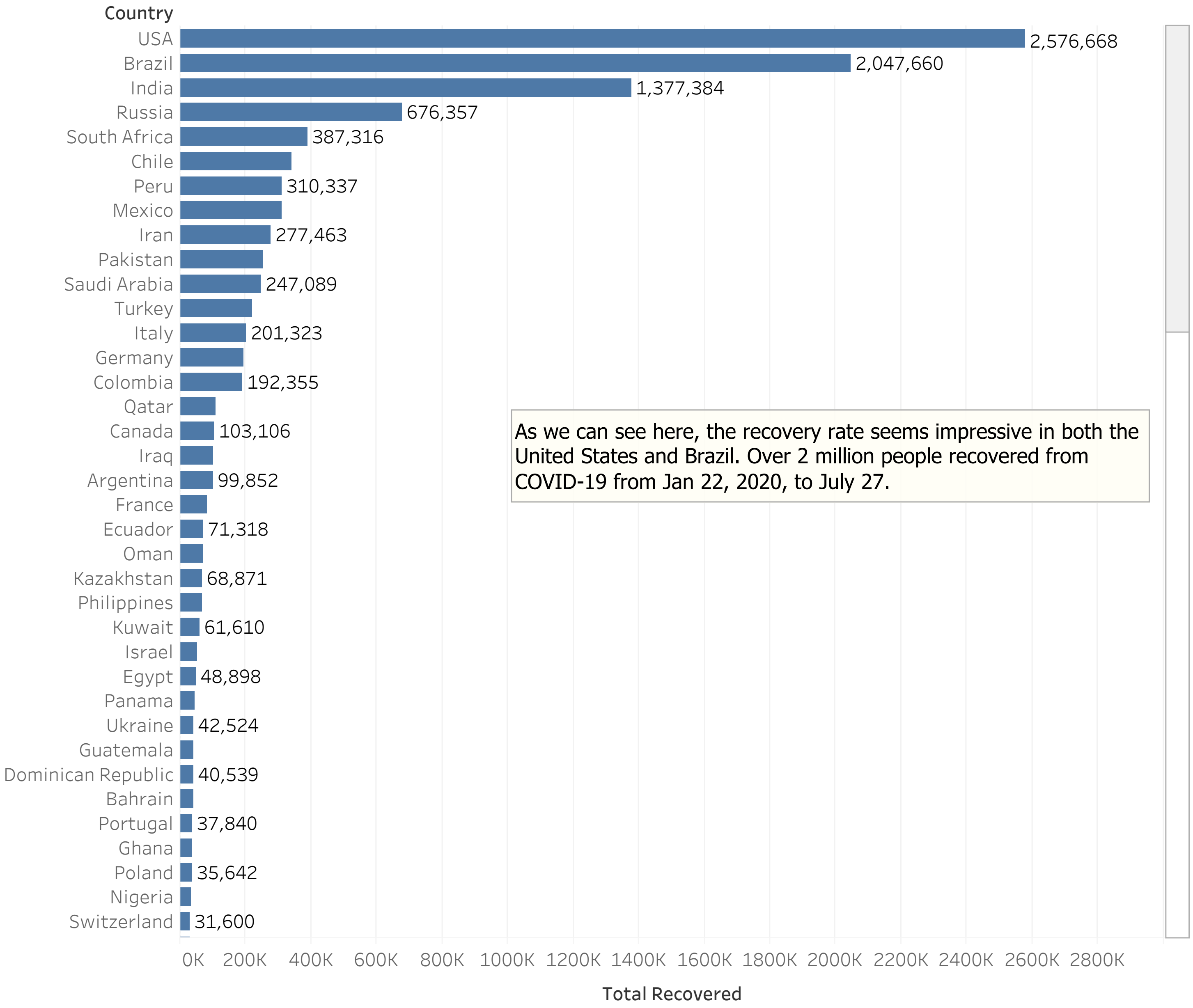
Data source: WHO via kaggles.com as of July 27, 2020

COVID-19 Story



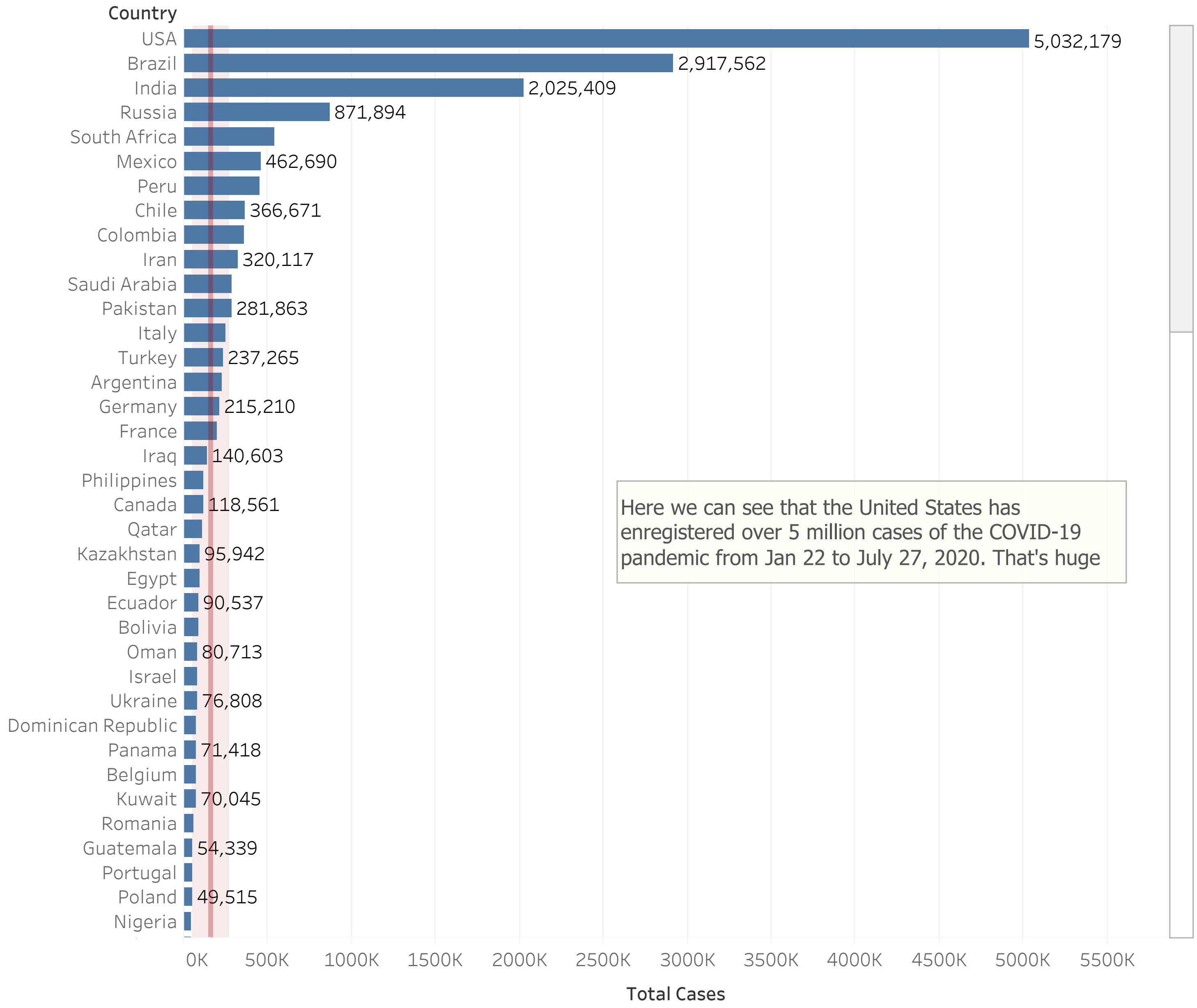
According to our study, we see that the US was the most affected by the COVID-19 pandemic, with a total of 162,804 deaths from Jan 22, 2020, to July 27, 2020

COVID-19 Story



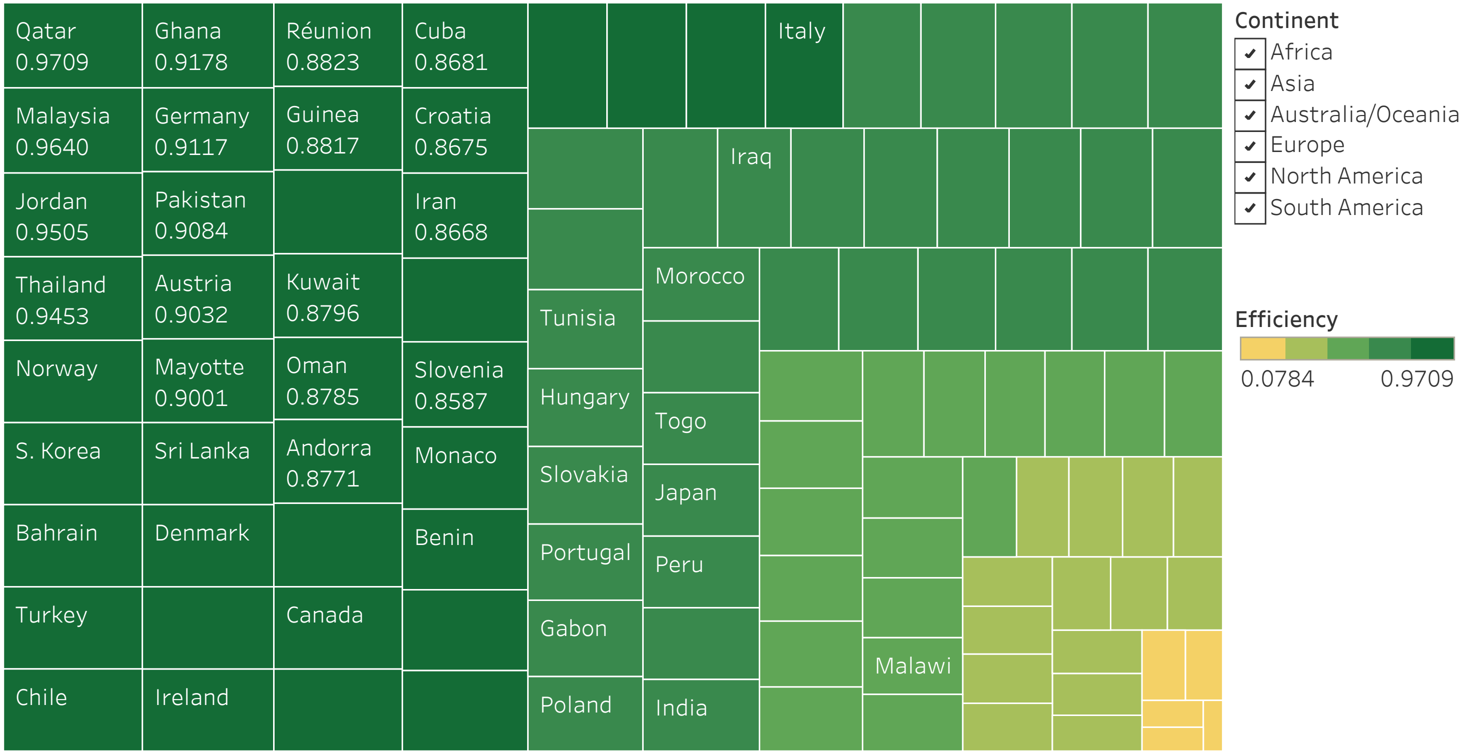
As we can see here, the recovery rate seems impressive in both the United States and Brazil. Over 2 million people recovered from COVID-19 from Jan 22, 2020, to July 27.

COVID-19 Story



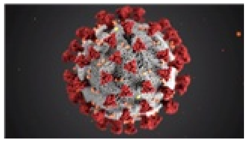
Here we can see that the United States has enregistered over 5 million cases of the COVID-19 pandemic from Jan 22 to July 27, 2020. That's huge

COVID-19 Story



COVID-19 Story

COVID 19 Pandemic Statistics



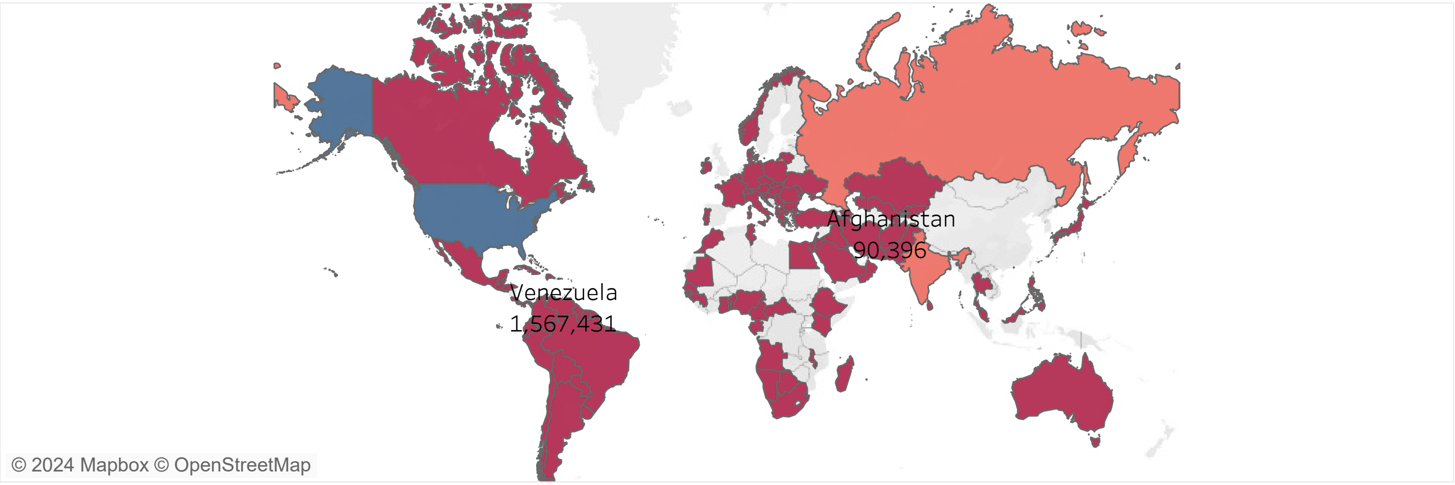
Total Cases	Total Recovered	Total Deaths	Total Tests
7,611,676	5,244,297	326,876	224,401,477

Continent
☑ Africa
☑ Asia
☑ Australia/Oceania
☑ Europe

Options
1,115 🟥🟥🟦🟦 63,139,605

Option
Total Tests

Worldwide Stat



© 2024 Mapbox © OpenStreetMap

Data source: WHO via kaggles.com as of July 27, 2020