

# COVID-19 Vaccine analysis and visualization using Python

## \*Background

The global COVID-19 pandemic has had a profound impact on societies worldwide, resulting in significant health, social, and economic challenges. One of the key strategies to combat the spread of the virus and minimize its impact has been the development and deployment of vaccines. While various COVID-19 vaccines have been developed and made available, the emergence and rapid development of multiple COVID-19 vaccine brands worldwide and successful and efficient administration of these vaccines on a global scale presents a complex challenge.

## \*Objective

To analyze and visualize the global COVID-19 vaccination landscape.

## \*Data Source

Country specific global vaccination data was downloaded from Our World in Data via <https://github.com/owid/covid-19-data/tree/master/public/data>

```
In [1]: # Import libraries

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
In [2]: # Load dataset

data = pd.read_csv("country_vaccinations.csv")
data.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	0.0	0.0
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	NaN	NaN
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	NaN	NaN
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	NaN	NaN
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	NaN	NaN

## \*Exploratory Data Analysis (EDA)

```
In [3]: # Generate descriptive statistics

data.describe()

# Note: If there are missing values in any columns, pandas will automatically exclude these values when calculating the
# descriptive statistics.To calculate descriptive statistics for all column in the df, use include='all' argument

# Generate descriptive statistics for all columns
# data.describe(include='all')
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred
count	4.360700e+04	4.129400e+04	3.880200e+04	3.536200e+04	8.621300e+04	43607.000000	41294.000000	38802.000000
mean	4.592964e+07	1.770508e+07	1.413830e+07	2.705996e+05	1.313055e+05	80.188543	40.927317	33.802000
std	2.246004e+08	7.078731e+07	5.713920e+07	1.212427e+06	7.682388e+05	67.913577	29.290759	23.920000
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	0.000000	0.000000	0.000000
25%	5.264100e+05	3.494642e+05	2.439622e+05	4.668000e+03	9.000000e+02	16.050000	11.370000	8.000000
50%	3.590096e+06	2.187310e+06	1.722140e+06	2.530900e+04	7.343000e+03	67.520000	41.435000	33.802000
75%	1.701230e+07	9.152520e+06	7.559870e+06	1.234925e+05	4.409800e+04	132.735000	67.910000	50.000000
max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.370000	124.760000	100.000000

```
In [4]: # Check date format

pd.to_datetime(data.date)
```

0	2021-02-22
1	2021-02-23
2	2021-02-24
3	2021-02-25
4	2021-02-26
...	...
86507	2022-03-25
86508	2022-03-26
86509	2022-03-27
86510	2022-03-28
86511	2022-03-29
Name: date, Length: 86512, dtype: datetime64[ns]	

```
In [5]: # Count the occurrences/frequencies of countries

data.country.value_counts()
```

Norway	482
Latvia	480
Denmark	476
United States	471
Russia	470
...	...
Bonaire Sint Eustatius and Saba	146
Tokelau	114
Saint Helena	92
Pitcairn	85
Falkland Islands	67
Name: country, Length: 223, dtype: int64	

```
In [6]: # Group the countries unde United Kingdom

data = data[data.country.apply(lambda x: x not in ["England", "Scotland", "Wales", "Northern Ireland"])]
data.country.value_counts()
```

Norway	482
Latvia	480
Denmark	476
United States	471
Canada	470
...	...
Bonaire Sint Eustatius and Saba	146
Tokelau	114
Saint Helena	92
Pitcairn	85
Falkland Islands	67
Name: country, Length: 219, dtype: int64	

```
In [7]: # Explore the vaccines in the data

data.vaccines.value_counts()
```

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	7608
Oxford/AstraZeneca	6022
Oxford/AstraZeneca, Pfizer/BioNTech	4629
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	4491
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech	3564
...	...
Johnson&Johnson, Oxford/AstraZeneca, Sinovac	312
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V	311
Johnson&Johnson, Moderna	251
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing	228
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001	190
Name: vaccines, Length: 84, dtype: int64	

## \*Data Preparation

```
In [8]: # Create a new DataFrame by only selecting the vaccine and country columns to explore which vaccine is taken by which country

df = data[["vaccines", "country"]]
df.head()
```

	vaccines	country
0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	Afghanistan
1	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	Afghanistan
2	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	Afghanistan
3	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	Afghanistan
4	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech	Afghanistan

```
In [9]: # Check how many countries are taking each of the vaccines mentioned in this data

dict_ = {}
for i in df.vaccines.unique():
    dict_[i] = [df["country"][j] for j in df[df["vaccines"]==i].index]
```

```
vaccines = {}
for key, value in dict_.items():
    vaccines[key] = set(value)
for i, j in vaccines.items():
    print(f"{i}>:{j}")

Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>{'Cameroon', 'Namibia', 'Afghanistan', 'Trinidad and Tobago', 'Belize'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V:>{'Azerbaijan', 'Oman', 'Bosnia and Herzegovina', 'Albania'}
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Zimbabwe', 'Oman'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech:>{'Jersey', 'Sint Maarten (Dutch part)', 'Sweden', 'Guernsey', 'Australia', 'Finland', 'Japan', 'Andorra', 'United Kingdom', 'Fiji', 'Isle of Man'}
Oxford/AstraZeneca:>{'Liberia', 'Angola', 'Saint Vincent and the Grenadines', 'Samoa', 'Kiribati', 'Nauru', 'Montserrat', 'Tuvalu', 'Falkland Islands', 'Pitcairn', 'Solomon Islands', 'Democratic Republic of Congo', 'Mali', 'Saint Helena', 'Vanuatu', 'Papua New Guinea', 'Sao Tome and Principe', 'Togo', 'Nigeria'}
Oxford/AstraZeneca, Pfizer/BioNTech:>{'Anguilla', 'Saint Lucia', 'Bermuda', 'Cayman Islands', 'Kosovo', 'New Zealand', 'Panama', 'Saint Kitts and Nevis', 'Gibraltar', 'Costa Rica', 'Saudi Arabia'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>{'Antigua and Barbuda'}
CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>{'Argentina'}
Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Armenia'}
Pfizer/BioNTech:>{'Aruba', 'Cook Islands', 'New Caledonia', 'Monaco', 'Tokelau', 'Niue', 'Turks and Caicos Islands'}
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech:>{'Germany', 'Italy', 'Austria', 'Netherlands', 'Slovenia', 'South Korea', 'Czechia', 'Lithuania'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech:>{'Bahamas', 'Eswatini', 'Grenada'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik V:>{'Bahrain'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Bangladesh'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>{'Maldives', 'Peru', 'Barbados', 'Dominica', 'Suriname'}
Sinopharm/Beijing, Sputnik V:>{'Kyrgyzstan', 'Belarus'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech:>{'Ireland', 'Iceland', 'Romania', 'Canada', 'Croatia', 'Luxembourg', 'Cyprus', 'Portugal', 'Estonia', 'Spain', 'Bulgaria', 'Jamaica', 'Malta', 'Poland', 'France', 'Belgium', 'Greece'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>{'Benin', 'Brazil'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>{'Cape Verde', 'Bhutan'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>{'Cote d'Ivoire', 'Morocco', 'Bolivia', 'Moldova'}
Moderna, Pfizer/BioNTech:>{'Bonaire Sint Eustatius and Saba', 'Norway', 'Qatar', 'Faeroe Islands', 'Israel', 'Curacao'}
Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>{'Botswana'}
Johnson&Johnson, Oxford/AstraZeneca:>{'Malawi', 'South Sudan', 'British Virgin Islands'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing:>{'Kuwait', 'Nepal', 'Brunei', 'Kenya'}
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing:>{'Gambia', 'Senegal', 'Madagascar', 'Lesotho', 'Mozambique', 'Zambia', 'Burkina Faso'}
Sinopharm/Beijing:>{'Chad', 'Burundi', 'Equatorial Guinea'}
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac:>{'Somalia', 'Cambodia'}
Covaxin, Oxford/AstraZeneca:>{'Central African Republic'}
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>{'Ecuador', 'Chile'}
CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001:>{'China'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>{'Colombia', 'Uganda', 'Ukraine'}
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing:>{'Mauritius', 'Comoros'}
Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V:>{'Congo'}
Abdala, Soberana Plus, Soberana02:>{'Cuba'}
Johnson&Johnson, Moderna, Pfizer/BioNTech:>{'Liechtenstein', 'United States', 'Switzerland', 'Denmark'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Guinea', 'Djibouti', 'Egypt'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Georgia', 'Dominican Republic', 'El Salvador'}
Covaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac:>{'Ethiopia'}
Johnson&Johnson, Pfizer/BioNTech:>{'French Polynesia', 'South Africa'}
Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>{'Gabon'}
Oxford/AstraZeneca, Sputnik V:>{'Ghana'}
Moderna:>{'Greenland', 'Wallis and Futuna'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>{'Guatemala'}
Oxford/AstraZeneca, Sinopharm/Beijing:>{'Niger', 'Sierra Leone', 'Mauritania', 'Myanmar', 'Guinea-Bissau'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>{'Guyana', 'Sri Lanka'}
Johnson&Johnson, Moderna:>{'Haiti'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>{'Honduras'}
Pfizer/BioNTech, Sinovac:>{'Hong Kong'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>{'Jordan', 'Hungary'}
Covaxin, Oxford/AstraZeneca, Sputnik V:>{'India'}
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Indonesia'}
COVIRan Barekat, Covaxin, FAKHRVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik V:>{'Iran'}
QazVac, Sinopharm/Beijing, Sputnik V:>{'Kazakhstan'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V:>{'Laos'}
Johnson&Johnson, Moderna, Novavax, Pfizer/BioNTech:>{'Latvia'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Libya', 'North Macedonia'}
Pfizer/BioNTech, Sinopharm/Beijing:>{'Macao'}
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Malaysia'}
CanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V:>{'Mexico'}
Abdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik V:>{'Nicaragua'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac:>{'Northern Cyprus', 'Timor', 'Uruguay'}
CanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Pakistan'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V:>{'Palestine', 'Philippines'}
Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Paraguay'}
EpiVacCorona, Sputnik V:>{'Russia'}
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V:>{'Rwanda', 'Tunisia'}
Pfizer/BioNTech, Sputnik V:>{'San Marino'}
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V:>{'Seychelles'}
Moderna, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Singapore'}
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V:>{'Slovakia'}
Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Sudan'}
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V:>{'Syria'}
Medigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V:>{'Tajikistan'}
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing:>{'Tanzania'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac:>{'Thailand'}
Pfizer/BioNTech, Sinovac, Turkovac:>{'Turkey'}
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001:>{'Turkmenistan'}
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V:>{'United Arab Emirates'}
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V, ZF2001:>{'Uzbekistan'}
Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik V:>{'Venezuela'}
Abdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V:>{'Vietnam'}
Johnson&Johnson, Oxford/AstraZeneca, Sinovac:>{'Yemen'}
```

## \*Data Visualization

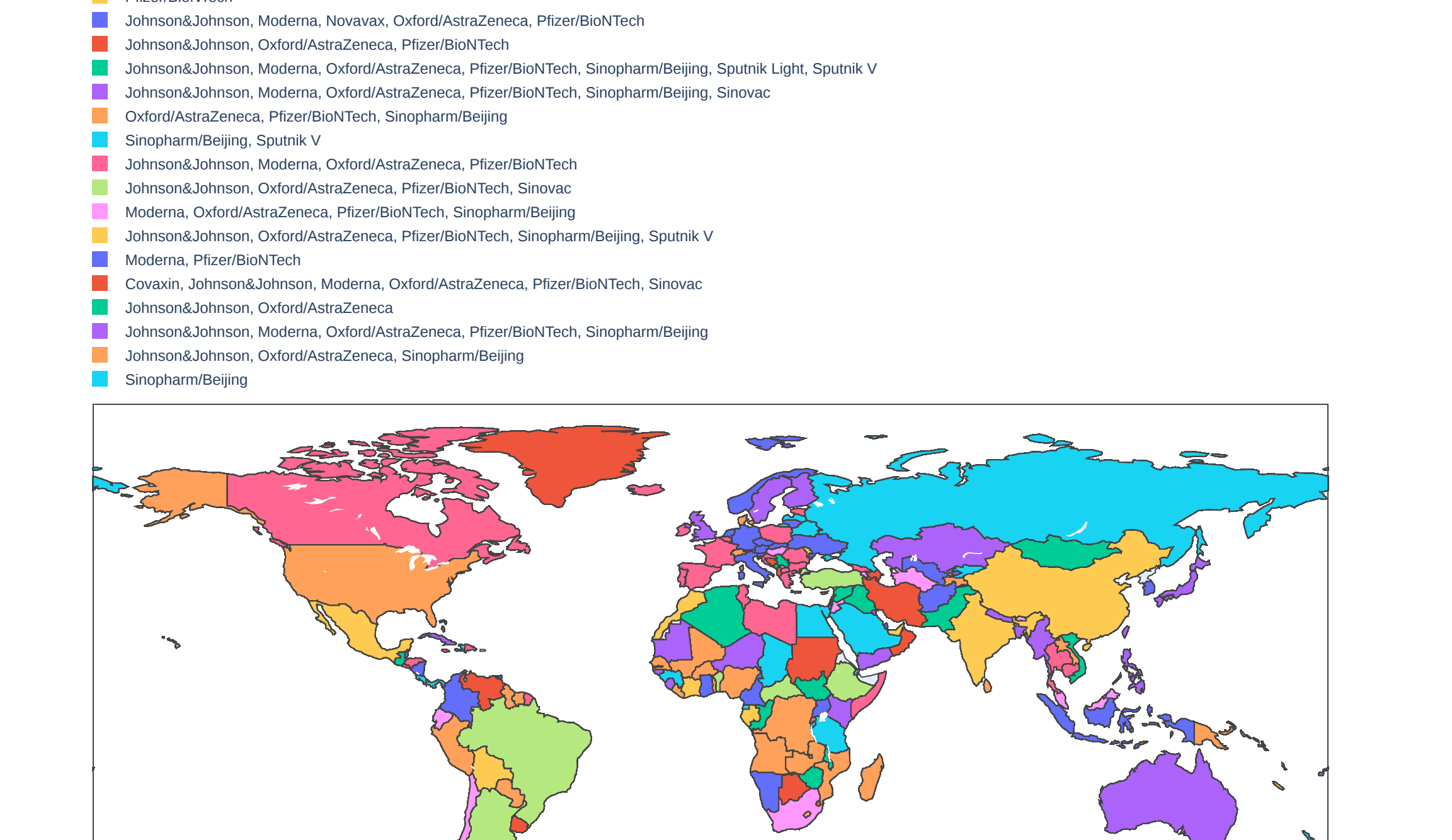
```
In [10]: # Visualize this data to have a look at what combination of vaccines every country is using

import plotly.express as px

vaccine_map = px.choropleth(data, locations='iso_code', color='vaccines')
vaccine_map.update_layout(height=1000, margin={'r': 0, 't': 0, 'l': 0, 'b': 0})

# Position the legend below the map
vaccine_map.update_layout(legend=dict(orientation="h", y=1.02, yanchor="bottom"))

vaccine_map.show()
```



End